Elżbieta Hajnicz

Przegląd formalnych metod semantycznych

Nr 965

Warszawa, grudzień 2003

Streszczenie

Niniejsze opracowanie zawiera przegląd trzech formalizmów, służących do reprezentacji semantyki j. naturalnych, opartych na teorii modeli. Są to: *Semantyka Montague*, *Teoria Reprezentacji Dyskursu* (ang. *Discourse Reprezention Theory* (DRT)) oraz *Semantyka oparta na zasobach leksykalnych* (ang. *Lexical Resource Semantics* (LRS)). Dla każdego z tych rozwiązań przedstawiamy formalizm logiczny będący jego podstawą, przykładową gramatykę dla (ograniczonego) fragmentu j. angielskiego oraz zbiór zasad translacji rozbiorów gramatycznych uzyskanych w wybranej gramatyce na właściwy formalizm. Na koniec prezentowane jest porównanie prezentowanych podejść.

Słowa kluczowe: semantyka j. naturalnego, formalizmy logiczne, podejście teoriomodelowe

Abstract

A survey of fomal semantic methods

This report contains a survey of three formalisms serving to represent natural languege semantics and based on model theory. We *choose Montague Semantics*, *Discourse Reprezention Theory* (DRT) and *Lexical Resource Semantics* (LRS). For every solution, we present logical formalism it is based on, an exemplary grammar for a (limited) fragment of english and a set of rules of translation of syntactic analysis obtained in a chosen grammar onto the appropriate formalism. Finally, we present a comparison of the solutions under presentation.

Keywords: natural language semantics, logic formalisms, modeltheoretic approach

Spis treści

1 Wstęp

Inżynieria lingwistyczna jest obecnie coraz szybciej rozwijającą się dziedziną wiedzy. Stale rosnące zasoby tekstów w wielu różnych językach naturalnych zapisanych w postaci elektronicznej, zwłaszcza zasoby dostępne za pomocą globalnej sieci Internetu, wymagają coraz doskonalszych i szybszych technik ich automatycznego przetwarzania.

Jednym z istotnych ograniczeń istniejących technik, wykorzystywanych na przykład do automatycznego tłumaczenia, wyszukiwania czy anotacji tekstów, jest ograniczone wykorzystanie informacji semantycznych zawartych w przetwarzanych tekstach. Aby móc tę słabość przezwyciężyć, niezbędne jest opracowywanie i rozwijanie metod semantyki formalnej. Chociaż semantyka z pozoru zdaje się niezależna od języka, w jakim dany tekst jest zapisany, w celu przekształcenia wybranego ciągu zdań na jego reprezentację semantyczną niezbędne jest dysponowanie jego rozbiorem syntaktycznym, a do tego potrzebna jest gramatyka formalna jak najbardziej zależna od języka. Rzecz jasna istnieją formalizmy gramatyczne (np. *Definite Clause Grammars* (DCG), *Generalized Phrased Structured Grammar* (GPSG), *Head-driven Phrased Structured Grammar* (HPSG)) w których można zapisać różne gramatyki dla odmiennych języków. Podobnie rzecz się ma z formalizmami semantycznymi: można sformułować reguły przekształcające wyrażenia dowolnego języka (a właściwie ich rozbiorów gramatycznych uzyskanych za pomocą wybranej gramatyki tego języka) na formuły języka logicznego stanowiącego podstawę reprezentacji semantycznej.

W niniejszej pracy zaprezentowany zostanie przegląd najbardziej znanych i najszerzej stosowanych formalizmów semantycznych dla języka naturalnego. Mnogość istniejących rozwiązań nie pozwala na omówienie ich wszystkich. Ostatecznie wybrane zostały trzy formalizmy. Zaczniemy od *semantyki Montague*. Formalizmu tego nie można pominąć, gdyż stanowi punkt odniesienia dla większości późniejszych podejść. Kolejne dwa formalizmy, *Discourse Reprezention Theory* (DRT) oraz *Lexical Resource Semantics* (LRS), umieściliśmy w tym omówieniu głównie dlatego, że planujemy oprzeć opracowywanie semantyki dla języka polskiego bazujące na gramatyce HPSG stworzonej w naszym zespole [Przepiórkowski i in., 2002] właśnie na jednym z tych dwóch formalizmów.

Ponieważ jednak każdy z tych formalizmów podlegał intensywnym zmianom i rozszerzeniom, napisano wiele prac na temat każadego z nich, niejednokrotnie poświęconych analizie pojedynczych zjawisk lingwistycznych, niniejsza praca w najmniejszym stopniu nie pretenduje do omówienia żadnego z tych formalizmów w całości. Przeciwnie, staraliśmy się przedstawić podstawowe ich wersje oraz w jaki sposób ujmują one najważniejsze zjawiska lingwistyczne. Z tego względu pominięte zostały rozważania dotyczące reprezentacji liczby mnogiej (a więc obiektów zbiorowych), przymiotników, przysłówków oraz czasu gramatycznego i aspektu czasowników.

2 Semantyka Montague

Jeden z najwcześniejszych i jednocześnie najbardziej znanych systemów, w których sformułowany został formalny, usystematyzowany sposób opisu semantyki języków, zaproponowany został w szeregu kolejnych prac przez Montague (1970a, 1970b, 1973). Co nie mniej ważne, jest to pierwsza formalizacja semantyki języka naturalnego oparta na teorii modeli, co obecnie jest metodą najbardziej rozpowszechnioną. Ponieważ dla celów tej semantyki opracowany został nie tylko formalny język jej zapisu, lecz także odpowiadający jej specyficzny zestaw reguł składniowych dla j. angielskiego (tak by każdej regule składniowej odpowiadała właściwa reguła semantyczna), niejednokrotnie mówi się nie tyle o *semantyce Montague*, ile wręcz o *gramatyce Montague*.

Niniejsze omówienie zostało opracowane nie w oparciu o oryginalne prace Montague, tylko na podstawie wyczerpującego omówienia zawartych w nich idei, przedstawionego w pracy *Introduction to Montague Semantics* autorstwa D.R. Dowty'ego, R.E. Walla i St. Petersa (1981).

Zaczniemy od przedstawienia logiki PTQ stanowiącej podstawę omawianej semantyki.y

2.1 Logika IL

Logika IL (skrót od *Intensional logic*) jest to formalizm modalny (operatory modalne dotyczą zarówno czasu, jak i wiedzy o świecie) rzędu ω oparty na teorii typów. Potrzeba reprezentowania zmienności wiedzy względem czasu wydaje się oczywista; jej uzasadnieniem jest choćby konieczność uwzględnienia czasu gramatycznego zdań. Natomiast wprowadzenie dodatkowego operatora modalnego motywowane było spostrzeżeniem, że prawdziwość czy też *konieczność* logiczna (*tautologiczność*) nie wyczerpuje pojęcia konieczności związanej z posiadaną wiedzą o świecie. Przykładem jest różnica między zdaniami *Hysperus jest tą samą planetą co Hysperus* oraz *Hysperus jest tą samą planetą co Fosforus*. Prawdziwość (czy też *konieczność*) drugiego z nich wynika z faktu, że w obu przypadkach chodzi o Wenus (o czym jednak starożytni Babilończycy nie wiedzieli, stąd dwie nazwy). Innym przykładem jest zdanie *Kawaler to nieżonaty mężczyzna*, którego prawdziwość wynika z kolei z wiedzy o języku.

2.1.1 Syntaktyka

Definicja 2.1. Zbiór typów $\mathcal{T}$ jest to najmniejszy zbiór taki, że:

- $t, e \in \mathcal{T}$ (typy proste);
- jeżeli $a, b \in \mathcal{T}$, to $\langle s, a\rangle, \langle a, b\rangle \in \mathcal{T}$.

Typ prosty e określa indywidua (stałe bądź zmienne indywiduowe), zaś typ t ma charakter prawdziwościowy (określa formuły). Zauważmy ponadto, że typ $\langle s, a\rangle$ zawiera symbol s (pochodzacy od wyrazu *sens*), który nie jest typem. Wynika to ze specyficznej koncepcji Montague, który postanowił przenieść semantyczne ze swej natury pojęcie *intensji* na poziom syntaktyczny.

Dopiero dysponując zbiorem typów $\mathcal{T}$ możemy określić alfabet języka omawianej logiki. Składają się nań:

- Niepusty, przeliczalny zbiór zmiennych V_a dla każdego typu a;
- Niepusty, przeliczalny zbiór stałych C_a dla każdego typu a;
- wyróżniony symbol $=$;
- symbole spójników logicznych $\neg$, $\rightarrow$ oraz kwantyfikatora ogólnego $\forall$;
- symbol λ;
- symbole operatorów modalnych $\mathbf{F}$, $\mathbf{P}$, $\Box$;
- symbole intensji $\hat{}$ oraz ekstensji $\check{}$.

Ponadto zbiór wszystkich zmiennych (a więc sumę zbiorów V_a) będziemy oznaczać przez V, zaś zbiór wszystkich stałych przez C.

Ze względu na istnienie hierachii typów nie wystarczy wydzielić zbioru termów i zbioru formuł. Zamiast tego wprowadzony został zbiór *wyrażeń znaczących* $\mathcal{E}$ (ang. *meaningful expressions*), podzielony na zbiory wyrażeń znaczących danego typu. Rzecz jasna formuły są to wyrażenia typu $\mathcal{E}_t$, zaś za typowe termy można uznać wyrażenia typu $\mathcal{E}_e$.

Definicja 2.2. Niech $a, b \in \mathcal{T}$ będą dowolnymi typami. Zbiór *wyrażeń znaczących* $\mathcal{E}$ jest to zestaw takich (najmniejszych) zbiorów $\mathcal{E}_a$, dla dowolnego $a \in \mathcal{T}$, że:

- $V_a \subseteq \mathcal{E}_a$;
- $C_a \subseteq \mathcal{E}_a$;
- jeżeli $\alpha \in \mathcal{E}_a$ oraz $u \in V_b$, to $\lambda u\alpha \in \mathcal{E}_{\langle b,a\rangle}$;
- jeżeli $\alpha \in \mathcal{E}_{\langle a,b\rangle}$ oraz $\beta \in \mathcal{E}_a$, to $\alpha(\beta) \in \mathcal{E}_b$;
- jeżeli $\alpha, \beta \in \mathcal{E}_a$, to $\alpha = \beta \in \mathcal{E}_t$;
- jeżeli $\phi, \psi \in \mathcal{E}_t$, to $\neg\phi \in \mathcal{E}_t$ oraz $\phi \to \psi \in \mathcal{E}_t$;
- jeżeli $\phi \in \mathcal{E}_t$ oraz $u \in V$, to $\forall u\phi \in \mathcal{E}_t$;
- jeżeli $\phi \in \mathcal{E}_t$, to $\mathbf{F}\,\phi, \mathbf{P}\,\phi, \Box\phi \in \mathcal{E}_t$;
- jeżeli $\alpha \in \mathcal{E}_a$, to $\hat{\ }\alpha \in \mathcal{E}_{\langle s,a\rangle}$;
- jeżeli $\alpha \in \mathcal{E}_{\langle s,a\rangle}$, to $\check{\ }\alpha \in \mathcal{E}_a$.

Ponadto w standardowy sposób definiujemy pozostałe spójniki logiczne, operatory modalne oraz kwantyfikator egzystencjalny $\exists$.

$$
\begin{array}{lll}
\varphi \ \vee \ \psi & \equiv_{def} & \neg\varphi \ \to \ \psi, \\
\varphi \ \& \ \psi & \equiv_{def} & \neg(\varphi \ \to \ \neg\psi), \\
\varphi \ \leftrightarrow \ \psi & \equiv_{def} & \neg((\varphi \ \to \ \psi) \ \to \ \neg(\psi \ \to \ \varphi)), \\
\exists x \ \varphi & \equiv_{def} & \neg\forall x \ \neg\varphi, \\
\mathbf{G}\,\varphi & \equiv_{def} & \neg\mathbf{F}\,\neg\varphi, \\
\mathbf{H}\,\varphi & \equiv_{def} & \neg\mathbf{P}\,\neg\varphi, \\
\Diamond\varphi & \equiv_{def} & \neg\Box\neg\varphi.
\end{array}
$$

Ponieważ typy złożone definiowane są wyłącznie w postaci par uporządkowanych, a nie na przykład trójek lub czwórek, funkcje bądź predykaty mogą mieć tylko jeden argument. Z tego względu predykat n-argumentowy $P(u_1, u_2 ..., u_n)$ definiujemy rekurencyjnie jako $[P'(u_2, ..., u_n)](u_1)$. Tak więc predykaty jednoargumentowe są typu $\langle e, t\rangle$, dwuargumentowe są typu $\langle e, \langle e, t\rangle\rangle$, trzyargumentowe są typu $\langle e, \langle e, \langle e, t\rangle\rangle\rangle$, itd. W szczególności, operator negacji $\neg$ jest typu $\langle t, t\rangle$, zaś operator implikacji ma typ $\langle t, \langle t, t\rangle\rangle$. Konwencja odwrotnej kolejności aplikacji argumentów pochodzi z omawianej pracy.

2.1.2 Semantyka

Przedstawiony w powyższym rozdziale język posiada następującą semantykę, będącą przypadkiem tzw. *semantyki możliwych światów* Kripke'go [Kripke, 1959; 1963; 1965] dla logik modalnych.

Definicja 2.3. Modelem $\mathcal{M}$ nazywamy piątkę uporządkowaną $\langle A, W, T, <, F \rangle$ taką, że A, W, T są to niepuste zbiory indywiduów (obiektów), możliwych (mega)światów i momentów czasu, odpowiednio; $<$ jest liniowym porządkiem na T, zaś F jest funkcją przyporządkowującą znaczenia (sens) wyrażeniom języka, czyli interpretacją.
Niech $a, b \in \mathcal{T}$. Zbiór D_a *możliwych denotacji* typu a definiowany jest następująco:

- $D_e = A$;
- $D_t = \{ 0, 1 \}$;
- $D_{\langle a,b \rangle} = D_b^{D_a}$;
- $D_{\langle s,a \rangle} = D_a^{W \times T}$.

Wartościowanie g jest funkcją (a właściwie zbiorem funkcji g_a) taką, że każdej zmiennej $u \in V_a$ przypisywana jest wartość z D_a. Zbiór wszystkich wartościowań oznaczany będzie przez $\mathcal{G}$. Ponadto przez g_u^e oznaczane będzie wartościowanie uzyskane z g przez przypisanie zmiennej u wartości e, zaś przez $\mathcal{G}_u^g$ — zbiór wszystkich wartościowań różniących się od g jedynie wartością zmiennej u.

Zauważmy, że w proponowanym podejściu pojęcia *(mega)światów* i *momentów czasu* są niezależne, innymi słowy dysponujemy zbiorem $|W|$ osi czasu, a każda taka oś tworzy pojedynczy (mega)świat, gdy tymczasem *moment czasu* stanowi przekrój przez wszystkie osie, zbiór „wariantów" rzeczywistości w danym momencie. Poszczególne „faktyczne" światy, czyli elementy takiej osi, identyfikowane są jako para $\langle \mathtt{w}, \mathtt{t} \rangle$. Aby pozostawać w zgodności ze zwyczajową terminologią, jako *świat* będziemy określać parę $\langle \mathtt{w}, \mathtt{t} \rangle$, $\mathtt{t}$ będzie *momentem*, zaś $\mathtt{w}$ *osią* czasu. Zwróćmy przy tym uwagę, że takie podejście oznacza, że upływ czasu we wszystkich „mega-światach" jest taki sam.

Zgodnie z powyższą definicją, wartościowanie zmiennych jest niezależne od osi i momentów czasu. Nie dotyczy to jednak stałych indywiduowych (typu e), których wartość może ulegać zmianie, podobnie jak stałych funkcyjnych czy predykatywnych. Uzasadnieniem takiego rozwiązania jest istnienie takich stałych, jak *Król Anglii*, których wartość ulega zmianie w czasie.

Istotną cechą interpretacji wyrażeń w logice klasycznej jest ich *kompozycyjność* (ang. *compositionality*). Własność ta oznacza, że wartość (ekstensja, denotacja) każdego (złożonego) wyrażenia jest funkcją wartości jego składników. Okazuje się jednak, że w przypadku wyrażeń modalnych bynajmniej tak nie jest, gdyż wartość wyrażenia zależy od wartości jego składników względem światów różnych od bieżącego.[1] Aby nie tracić tej własności, Montague wprowadza pojęcie *intensji* wyrażenia, będącej zbiorem wszystkich *ekstensji* tego wyrażenia dla wszystkich światów, czyli formalnie biorąc funkcją przyporządkowującą wyrażeniom ich ekstensje dla poszczególnych światów. Pojęcie to umożliwia nam rozpatrywanie potencjalnej wartości danego wyrażenia w całej jego zmienności, a nie tylko w konkretnym (bieżącym) świecie.

Montague wprowadza także pojęcie *sensu* typu a, oznaczany przez S_a. *Sens* (ang. *sense*) jest to wartość, jaką potencjalnie mogą przyjmować wyrażenia (typu a); tak więc $S_a = D_{\langle s,a \rangle}$.

Ponieważ w omawianym formalizmie brak jest wyraźnego podziału wyrażeń na termy i formuły, wygodniej jest mówić o *wartości semantycznej* wyrażeń zamiast o ich *spełnialności*.

Definicja 2.4. Niech $a, b \in \mathcal{T}$ będą dowolnymi typami. *Wartością semantyczną* wyrażenia α w modelu $\mathcal{M}$, osi czasu $\mathtt{w} \in W$, momencie $\mathtt{t} \in T$ i przy wartościowaniu $g \in \mathcal{G}$ (piszemy $[\![\alpha]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g}$) jest:

$$
\begin{array}{lcll}
[\![\alpha]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} & = & [F(\alpha)](\langle \mathtt{w}, \mathtt{t} \rangle) & \text{dla dowolnego } \alpha \in C; \\
[\![\alpha]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} & = & g(\alpha) & \text{dla dowolnego } \alpha \in V; \\
[\![\alpha(\beta)]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} & = & [\![\alpha]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g}([\![\beta]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g}) & \text{dla dowolnego } \alpha \in \mathcal{E}_{\langle b,a \rangle} \text{ i } \beta \in \mathcal{E}_a; \\
[\![\lambda u \alpha]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} & = & h & \text{dla dowolnego } \alpha \in \mathcal{E}_a,\ u \in V_b, \text{ gdzie } h\colon\ D_b \longrightarrow D_a \\
 & & & \text{jest taką funkcją, że } h(x) = [\![\alpha]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g_u^x}, \text{ gdzie } x \in D_b; \\
[\![{}^\frown\alpha]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} & = & h & \text{dla dowolnego } \alpha \in \mathcal{E}_a, \text{ gdzie } h\colon\ W \times T \longrightarrow D_a \\
 & & & \text{jest taką funkcją, że } h(\langle \mathtt{w}', \mathtt{t}' \rangle) = [\![\alpha]\!]^{\mathcal{M},\mathtt{w}',\mathtt{t}',\mathtt{g}} \\
 & & & \text{dla dowolnych } \mathtt{w}' \in W \text{ i } \mathtt{t}' \in T;
\end{array}
$$

[1] Nie należy mylić tego pojęcia z pojęciem *lokalności*, określającym brak zależności wartości wyrażenia od innych wyrażeń nie będących jego składowymi. Własność ta jest w logikach modalnych zachowana, znika dopiero w przypadku logik niemonotonicznych.

$$
\begin{array}{lll}
[\![{}^\frown\alpha]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} & = & [\![\alpha]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g}(\langle \mathtt{w},\mathtt{t}\rangle) \qquad \text{dla dowolnego } \alpha \in \mathcal{E}_{\langle s,a\rangle};\\
[\![\alpha=\beta]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} & = 1 & \text{wtw, gdy } [\![\alpha]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} \text{ jest równe } [\![\beta]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} \text{ dla dowolnych } \alpha,\beta \in \mathcal{E}_a;\\
[\![\neg\phi]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} & = 1 & \text{wtw, gdy } [\![\phi]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} = 0 \text{ dla dowolnego } \phi \in \mathcal{E}_t;\\
[\![\phi\to\psi]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} & = 1 & \text{wtw, gdy } [\![\phi]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} = 0 \text{ lub } [\![\psi]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} = 1 \text{ dla dowolnych } \phi,\psi \in \mathcal{E}_t;\\
[\![\forall u\phi]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} & = 1 & \text{wtw, gdy } [\![\phi]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g'} = 1 \text{ dla dowolnych } \phi \in \mathcal{E}_t,\ u \in V \text{ oraz } g' \in \mathcal{G}^g_u;\\
[\![\Box\phi]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} & = 1 & \text{wtw, gdy } [\![\phi]\!]^{\mathcal{M},\mathtt{w}',\mathtt{t}',g} = 1 \text{ dla dowolnych } \mathtt{w}' \in W,\ \mathtt{t}' \in T \text{ oraz}\\
& & \qquad\qquad \text{dla dowolnego } \phi \in \mathcal{E}_t;\\
\left[\!\!\left[\mathtt{F}\,\phi\right]\!\!\right]^{\mathcal{M},\mathtt{w},\mathtt{t},g} & = 1 & \text{wtw, gdy istnieje } \mathtt{t}' \in T \text{ takie, że } \mathtt{t} < \mathtt{t}' \text{ oraz } [\![\phi]\!]^{\mathcal{M},\mathtt{w},\mathtt{t}',g} = 1,\\
& & \qquad\qquad \text{dla dowolnego } \phi \in \mathcal{E}_t;\\
\left[\!\!\left[\mathtt{P}\,\phi\right]\!\!\right]^{\mathcal{M},\mathtt{w},\mathtt{t},g} & = 1 & \text{wtw, gdy istnieje } \mathtt{t}' \in T \text{ takie, że } \mathtt{t}' < \mathtt{t} \text{ oraz } [\![\phi]\!]^{\mathcal{M},\mathtt{w},\mathtt{t}',g} = 1,\\
& & \qquad\qquad \text{dla dowolnego } \phi \in \mathcal{E}_t.
\end{array}
$$

Jak można by oczekiwać, operatory **F** i **P** interpretowane są w ramach bieżącej osi czasu. Zauważmy jednak, że formuła $\Box\phi$ jest spełniona wtedy i tylko wtedy, gdy formuła ϕ spełniona jest we wszystkich możliwych światach. Autorzy nie definiują żadnej relacji dostępu dla światów (lub, innymi słowy, jest to relacja totalna).[2]

Definicja 2.5. Powiemy, że formuła ϕ jest prawdziwa w modelu $\mathcal{M}$ w osi czasu $\mathtt{w}$ i momencie $\mathtt{t}$ (piszemy $[\![\phi]\!]^{\mathcal{M},\mathtt{w},\mathtt{t}} = 1$) wtedy i tylko wtedy, gdy $[\![\phi]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} = 1$ dla każdego $g \in \mathcal{G}$. Powiemy, że formuła ϕ jest prawdziwa w modelu $\mathcal{M}$ (piszemy $[\![\phi]\!]^{\mathcal{M}} = 1$) wtedy i tylko wtedy, gdy $[\![\phi]\!]^{\mathcal{M},\mathtt{w},\mathtt{t}} = 1$ dla każdego $\mathtt{w} \in W$ i $\mathtt{t} \in T$. Powiemy, że formuła ϕ jest prawdziwa (jest tautologią; piszemy $[\![\phi]\!] = 1$) wtedy i tylko wtedy, gdy $[\![\phi]\!]^{\mathcal{M}} = 1$ dla każdego modelu $\mathcal{M}$.
Ponadto *intensją* wyrażenia α typu $a \in \mathcal{T}$ względem modelu $\mathcal{M}$ i wartościowania $g \in \mathcal{G}$ (piszemy $[\![\phi]\!]^{\mathcal{M},g}_{\not{e}}$) jest taka funkcja $h\colon W \times T \longrightarrow D_a$, że $h(\langle \mathtt{w},\mathtt{t}\rangle) = [\![\phi]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},\mathtt{g}}$ dla dowolnych $\mathtt{w} \in W$ i $\mathtt{t} \in T$.

Funkcja interpretacji F przypisuje stałym różnego typu ich intensje zgodnie z tym typem. W ten sam sposób można zdefiniować wartości operatorów logicznych, choć one rzecz jasna nie zależą od funkcji F. Dla przykładu podamy odpowiednie wartości dla negacji i implikacji w tabeli 2.1.

$$
[\![\neg]\!] = \begin{bmatrix} 1 & \longrightarrow & 0 \\ 0 & \longrightarrow & 1 \end{bmatrix}
\qquad\qquad
[\![\to]\!] = \begin{bmatrix} 1 & \longrightarrow & \begin{bmatrix} 1 & \longrightarrow & 1 \\ 0 & \longrightarrow & 1 \end{bmatrix} \\ 0 & \longrightarrow & \begin{bmatrix} 1 & \longrightarrow & 0 \\ 0 & \longrightarrow & 1 \end{bmatrix} \end{bmatrix}
$$

Tabela 2.1: Wartość semantyczna operatorów negacji i implikacji

Ponieważ symbol $=$ może być używany do porównywania wyrażeń dowolnego typu, może zostać zastosowany również do stwierdzenia identyczności intensji dwóch wyrażeń. Zauważmy przy tym, że stwierdzenie ${}^\frown\alpha = {}^\frown\beta$ jest zazwyczaj mocniejsze niż zwykłe $\alpha = \beta$, gdyż dotyczą wartości dla wszystkich światów, a nie wybranego. Zwróćmy ponadto uwagę, że w przypadku formuł symbol równości $=$ jest tożsamy z operatorem równoważności $\leftrightarrow$.

Przedstawimy teraz kilka sugestii terminologicznych, które przydadzą się w dalszych rozważaniach. Po pierwsze, intensje obiektów (typu e), czyli wyrażenia typu $\langle s, e\rangle$, będą określane mianem *pojęć prostych* (ang. *individual concept*). Po drugie, intensje formuł, czyli wyrażenia typu $\langle s, t\rangle$, będą określane mianem *stwierdzeń* (ang. *proposition*). Następnie, intensje predykatów jednoargumentowych typu $\langle e, t\rangle$ (wyrażenia typu $\langle s, \langle e, t\rangle\rangle$) będą nazywane *własnościami obiektów* (ang. *properties of individuals*). Ogólnie, *własnościami* wyrażeń typu $a \in \mathcal{T}$ będą określane wyrażenia typu $\langle s, \langle a, t\rangle\rangle$. Podobnie, intensje

[2] W pracy Dowty i in. (1981) pada w pewnym miejscu (s.131, 136) uwaga, że takie podejście odpowiada systemowi S5. Nie jest to prawda, gdyż system S5 oznacza jedynie, że relacja dostępności światów jest relacją równoważności, niekoniecznie totalną. Na przykład powiązanie ze sobą jedynie światów „jednoczesnych" dalej byłoby zgodne z systemem S5.

wyrażeń typu $\langle e, \langle e, t \rangle \rangle$ (a więc predykatów dwuargumentowych) będziemy nazywać intensją relacji między obiektami (ang. *relation-in-intention between individuals*), i pojęcie to z kolei uogólnia się na predykaty dwuargumentowe typu $\langle a, \langle b, t \rangle \rangle$ o argumentach typu $a, b \in \mathcal{T}$.

Rozważmy ponadto wyrażenie ξ typu $\langle s, \langle e, t \rangle \rangle$ (czyli *własność obiektów*) oraz pewną stałą indywiduową α (typu e). Zamiar powiązania tych wyrażeń w celu uzyskania własności ξ obiektu α wydaje się naturalny i zrozumiały. Jednak nie jest możliwe dokonanie tego bezpośrednio, możemy jedynie określić ekstensję tego wyrażenia. W tym celu definiujemy $\xi\{\alpha\}$ jako ${}^{\vee}\xi(\alpha)$ dla dowolnych $\xi \in \mathcal{E}_{\langle s, \langle a, t \rangle \rangle}$ oraz $\alpha \in \mathcal{E}_a$, gdzie $a \in \mathcal{T}$ na oznaczenie faktu, że stała α ma własność ξ.

2.2 Gramatyka PTQ

Gramatyka PTQ (skrót pochodzi od tytułu pracy Montague *The proper treatment of quantification in ordinary English*) opisuje, zgodnie ze słowami samego Montague, „fragment języka angielskiego". Syntaktyka tego języka skonstruowana jest za pomocą rekurencyjnych definicji określających, w jaki sposób frazy złożone konstruowane są z prostszych. Reguły składniowe nie tylko konkatenują wyrażenia wejściowe, lecz także dokonują transformacji danych wejściowych. Należy przy tym zauważyć, że składnia ta może zostać przełożona na równoważną gramatykę ze strukturą frazową i prostym składnikiem transformacyjnym.

Niezmiernie istotną cechą podejścia Montague jest fakt, że nie przypisuje on teoriomodelowej semantyki zdaniom j. angielskiego opisywanym przez jego gramatykę bezpośrednio, a jedynie dokonuje ich translacji na język logiki IL omówionej w poprzednim rozdziale. Dopiero semantyczna reprezentacja wyrażeń tegoż języka daje w rezultacie właściwy semantyczny opis tych zdań. Zasada ta stosowana jest także w innych formalizmach semantycznych opartych na teorii modeli, w szczególności w tych omawianych w niniejszym opracowaniu.

Wspomniana translacja stanowi rygorystycznie sformalizowaną procedurę, która nie tylko zachowuje strukturę składniową każdego analizowanego zdania angielskiego, ale także spełnia następujące warunki:

1. Każde angielskie wyrażenie proste jest tłumaczone na dokładnie jedno wyrażenie IL (niekoniecznie proste).

2. Istnieje ścisła odpowieniość pomiędzy kategoriami j. angielskiego i typami IL (dotycząca wszystkich wyrażeń danej kategorii).

3. Dla każdej reguły składniowej j. angielskiego istnieje odpowiadająca jej reguła translacyjna, która tworzy odpowiednie wyrażenie języka IL, zachowując przy tym odpowiedniość kategorii językowych i typów.

Omawiana gramatyka sformułowana została dla języka angielskiego. Analizowane wyrażenia tego języka (wyrazy, zdania) nie będą tłumaczone na język polski, zakładamy bowiem znajomość tego języka wśród czytelników niniejszego opracowania wystarczającą do zrozumienia przytoczonych tu wyrażeń.

2.2.1 Kategorie składniowe dla języka angielskiego w gramatyce PTQ

Montague wprowadza formalną, rekurencyjną definicję kategorii składniowych.

Definicja 2.6. Kategorią składniową jest dowolny symbol, spełniający poniższe warunki:

- symbole e i t są kategoriami składniowymi;[3]
- jeśli A i B są kategoriami składniowymi, to A/B, $A//B$ są także kategoriami składniowymi.

[3] Proste kategorie składniowe oznaczane są dokładnie tymi samymi symbolami co odpowiadające im typy proste, co jest rozwiązaniem odrobinę nieformalnym, choć nie prowadzi do konfuzji.

System ten oparty jest na *gramatykach kategorialnych* opracowanych przez Ajdukiewicza (1935). W gramatykach kategorialnych wyrażenie kategorii A/B (lub $A//B$) złożone z wyrażeniem kategorii B daje wyrażenie kategorii A (zapisujemy $A/B \cdot B = A$). Rozróżnienie kategorii A/B i $A//B$ podyktowane było koniecznością uwzględnienia dwóch różnych kategorii o takiej samej budowie (przypadki trzech analogicznych kategorii nie występują). Rzecz jasna, kategorie takie tłumaczą się na ten sam typ języka IL. Zauważmy ponadto, że złożone symbole kategorii j. angielskiego czytane są „z prawa na lewo”, gdy tymczasem złożone typy języka logicznego czytane są „z lewa na prawo”.

W tabeli 2.2 przedstawione są kategorie faktycznie używane w gramatyce PTQ, wraz z zaproponowanymi przez Mongtague nazwami.

Nazwa Kategorii	Definicja Kategorii	Opis Kategorii	Wyrażenia podstawowe
e		brak	brak
t		zdanie	brak
IV	t/e	fraza czasownikowa oraz czasownik nieprzechodni	run, walk, talk, rise, change
T	t/IV	fraza rzeczownikowa oraz nazwa własna	John, Mary, Bill, ninety, $he_0, he_1, ...$
TV	IV/T	czasownik przechodni	find, lose, eat, love, date, be, seek, conceive
IAV	IV/IV	przysłówek przyczasownikowy	rapidly, slowly, voluntarily, allegedly
CN	$t//e$	rzeczownik pospolity	man, woman, park, fish, pan, unicorn, price, temperature
t/t		przysłówek zdaniowy	necessarily
IAV/T		przyimek	in, about
IV/t		czasownik o dopełnieniu zdaniowym	believe, assert
$IV//IV$		czasownik o dopełnieniu bezokolicznikowym	try, wish
DET	T/CN	rodzajnik	every, the, a(n)

Tabela 2.2: Kategorie składniowe gramatyki PTQ

Wszystkie kategorie PTQ różne od wymienionych powyżej są puste. Zauważmy, że kategoria IV zawiera zarówno czasowniki nieprzechodnie, jak i frazy czasownikowe. Podobna sytuacja dotyczy kategorii T. Kategoria e nie zawiera wyrażeń podstawowych, gdyż nazwy własne takie jak *John* nie są traktowane jako indywiduum *John*, lecz zbiór jego własności. W związku z tym nazwy własne należą do tej samej kategorii co kwantyfikowane frazy typu *every man*. Ponadto „indeksowane” zaimki $\mathtt{he}_1, \mathtt{he}_2$ (w pozycji dopełnienia zapisywane jako $\mathtt{him}_1, \mathtt{him}_2$) pełnią rolę zmiennych tego typu (quasi-indywidua). Są one używane przy tworzeniu zaimków anaforycznych, rozwiązywaniu niejednoznaczności zakresu (ang. *scope ambiguity*) oraz w konstrukcjach względnych (ang. *relative clauses*). Natomiast rzeczowniki pospolite (np. *man*) należą do odrębnej kategorii *CN*, gdyż dopiero wraz z rodzajnikiem (*a man, every man)*) tworzą frazę rzeczownikową kategorii T.

Wszystkie wyrażenia podstawowe dowolnego typu zostały wymienione w tabeli 2.2. Zbiór wyrażeń podstawowych kategorii A oznaczany będzie przez B_A; zbiór wszystkich wyrażeń tej kategorii oznaczany będzie przez P_A. Zestaw ten jest wystarczający dla celów demonstracyjnych, a bez trudu można sobie wyobrazić znacznie bogatszy słownik, który każdemu wyrazowi przyporządkowuje odpowiednią kategorię. Zauważmy jednak, że niezależnie od zaniedbywalnych ograniczeń czysto słownikowych, „fragment” j. angielskiego opisywany przez PTQ jest dość ograniczony, na przykład pominięte zostały przymiotniki.

Montague uważał, że do określenia ekstensji dowolnego wyrażenia złożonego niezbędne są nie tyle ekstensje, ile wręcz intensje jego składników. Jednak Bennett (1974) pokazał, że w przypadku kategorii IV i CN było to założenie niepotrzebne. Za Dowty i in. (1981) prezentujemy odpowiednio zmodyfikowane podejście. Wówczas (pusta) kategoria e staje się zbędna.

Reguła, zgodnie z którą każdej kategorii składniowej PTQ przypisywany jest dokładnie jeden typ, może zostać wyrażona za pomocą następującej funkcji f:

Definicja 2.7. Niech A i B będą dowolnymi kategoriami składniowymi. Funkcją przyporządkowującą dowolnej kategorii składniowej odpowiedni typ z $\mathcal{T}$ jest taka funkcja f, że:

- $f(t) = t$;
- $f(CN) = F(IV) = \langle e, t\rangle$;
- $f(A/B) = f(A//B) = \langle\langle s, F(B)\rangle, f(A)\rangle$ dla dowolnych kategorii składniowych A i B.

Typy odpowiadające poszczególnym kategoriom syntaktycznym przedstawione są tabeli 2.3.

Nazwa kategorii	Definicja kategorii	Typ odpowiadający danej kategorii	Wartość semantyczna
t	t	t	wartość logiczna
IV	t/e	$\langle e, t\rangle$	zbiór indywiduów
CN	$t//e$	$\langle e, t\rangle$	zbiór indywiduów
T	$t/(t/e)$	$\langle\langle s, \langle e, t\rangle\rangle, t\rangle$	zbiór własności indywiduów
IAV	IV/IV	$\langle\langle s, \langle e, t\rangle\rangle, \langle e, t\rangle\rangle$	funkcja z własności indywiduów do zbiorów indywiduów
TV	$IV/(t/IV)$	$\langle\langle s, \langle\langle s, \langle e, t\rangle\rangle, t\rangle\rangle, \langle e, t\rangle\rangle$	funkcja z własności własności indywiduów do zbiorów indywiduów
DET	$(t/IV)/CN$	$\langle\langle s, \langle e, t\rangle\rangle, \langle\langle s, \langle e, t\rangle\rangle, t\rangle\rangle$	funkcja z własności indywiduów do zbiorów własności indywiduów
t/t	t/t	$\langle\langle s, t\rangle, t\rangle$	zbiór stwierdzeń
IV/t	IV/t	$\langle\langle s, t\rangle, \langle e, t\rangle\rangle$	funkcja ze stwierdzeń do zbioru indywiduów
$IV//IV$	$IV//IV$	$\langle\langle s, \langle e, t\rangle\rangle, \langle e, t\rangle\rangle$	funkcja z własności indywiduów do zbiorów indywiduów
IAV/T	$(IV/IV)/T$	$\langle\langle s, \langle\langle s, \langle e, t\rangle\rangle, t\rangle\rangle,$ $\langle\langle s, \langle e, t\rangle\rangle, \langle e, t\rangle\rangle\rangle$	funkcja z własności własności indywiduów do funkcji z własności indywiduów do zbiorów indywiduów

Tabela 2.3: Typy odpowiadające poszczególnym kategoriom syntaktycznym

2.2.2 Najważniejsze zasady translacji

Jak już wspominaliśmy, ważną, jeśli nie najważniejszą cechą gramatyki PTQ jest występowanie translacji na język IL stanowiącej podstawę do określenia wartości semantycznej analizowanych wyrażeń j. angielskiego. Wyrażenia podstawowe przekładane są z j. angielskiego na język IL za pomocą funkcji $\mathfrak{g}$; funkcję tę można rzecz jasna rozszerzyć na wyrażenia o dowolnym stopniu złożoności. Przyjęto konwencję notacyjną, że $\mathfrak{g}(\alpha) = \alpha'$ dla dowolnego wyrażenia (podstawowego) α.

Niektóre spośród wyrażeń podstawowych wymienionych w tabeli 2.2 przekładane są na stałe języka IL (oczywiście typu właściwego dla danej kategorii). Dotyczy to w szczególności czasowników nieprzechodnich z kategorii IV oraz rzeczowników pospolitych z kategorii CN. Tak więc $\mathfrak{g}(\mathsf{walk}) = \mathsf{walk}'$, zaś $\mathfrak{g}(\mathsf{man}) = \mathsf{man}'$, gdzie zarówno walk', jak i man' są stałymi typu $\langle e, t\rangle$. Podobnie rzecz się ma dla czasowników o dopełnieniu zdaniowym z kategorii IV/t ($\mathfrak{g}(\mathsf{believe}) = \mathsf{believe}'$), przy czym $\mathsf{believe}'$ jest stałą typu $\langle\langle s, t\rangle, \langle e, t\rangle\rangle$. W ten sam sposób traktowane są czasowniki przechodnie (dając stałe typu $\langle\langle s, \langle\langle s, \langle e, t\rangle\rangle, t\rangle\rangle, \langle e, t\rangle\rangle$), w tym także o dopełnieniu bezokolicznikowym (typu $\langle\langle s, \langle e, t\rangle\rangle, \langle e, t\rangle\rangle$), z wyjątkiem czasownika be, który przekładany jest następująco:

$$\mathfrak{g}(\mathsf{be}) = \lambda\mathcal{P}\,\lambda x\,\mathcal{P}\{{}^\frown\lambda y\,[x = y]\}.$$

Także przysłówki przyczasownikowe translowane są na stałe (typu $\langle\langle s, \langle e, t\rangle\rangle, \langle e, t\rangle\rangle$), gdy tymczasem przysłówek zdaniowy necessarily posiada następującą interpretację w IL:

$$\mathfrak{g}(\textsf{necessarily}) = \lambda p\,[\Box\check{}p].$$

Translacje nazw własnych i rodzajników

W przeciwieństwie do omówionych powyżej, pozostałe wyrażenia podstawowe posiadają dość złożone translacje. I tak, nazwy własne John, Mary, Bill $\in B_T$ (jako zbiory własności indywiduów) posiadają następujące translacje (typu $\langle\langle s, \langle e, t\rangle\rangle, t\rangle$, a więc samo P jest typu $\langle s, \langle e, t\rangle\rangle$):

$$\mathfrak{g}(\textsf{John}) = \lambda P\,[P\{j\}] \qquad \mathfrak{g}(\textsf{Mary}) = \lambda P\,[P\{m\}] \qquad \mathfrak{g}(\textsf{Bill}) = \lambda P\,[P\{b\}]$$

Dotyczy to również „indeksowanych" zaimków $\texttt{he}_1, \texttt{he}_2 \in B_T$, przeto ich translacje na IL wyglądają dość podobnie:

$$\mathfrak{g}(\texttt{he}_n) = \lambda P\,[P\{x_n\}], \text{ gdzie } x_n \in V_e.$$

Zgodnie z def. 2.4 $[\![\lambda P[P\{c\}]]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} = h\colon D^{W\times T}_{\langle e,t\rangle} \longrightarrow \{0,1\}$, gdzie $h(Q) = [\![Q(c)]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g^Q_P}(\langle \mathtt{w}, \mathtt{t}\rangle)$. Tak więc jest to funkcja charakterystyczna (zbiór) własności, a właściwie ich ekstensji (typu $\langle e, t\rangle$). Typ ten określa nie tylko czasowniki nieprzechodnie, będące relacjami jednoargumentowymi (I rzędu), lecz także rzeczowniki pospolite, będące zbiorami indywiduów (gdyż pojęcia te są równoważne). W tym właśnie znaczeniu nazwa własna bądź fraza rzeczownikowa są ≫zbiorami własności≪ (posiadanych przez stałą indywiduową c typu e). Tak więc własnością stałej j „denotującej" John może być zarówno fakt bycia mężczyzną ($\textsf{man}'(j)$) czy studentem, jak i fakt, że obiekt ten idzie ($\textsf{walk}'(j)$) czy śpiewa. Zwyczajowo jedynie te pierwsze uznajemy za *własności* obiektu (zwane przez niektórych autorów *faktami*), podobnie jak stwierdzenia, że ktoś jest wysoki, dowcipny lub ma zielone oczy [Allen, 1984; Artale i Franconi, 1994; Galton, 1990; McDermott, 1982].[4] Jednakże zgodnie z koncepcją Montagu'e semantyka nazw własnych polega na reprezentowaniu całej wiedzy występującej w modelu na temat denotującego ją (w danym świecie) obiektu, można by rzec ≫całą jego historię≪.

Następnie, rodzajniki (jako funkcje z własności indywiduów do zbiorów własności indywiduów), mają następujące translacje (P i Q są znowu typu $\langle s, \langle e, t\rangle\rangle$):

$$\begin{array}{lcl} \mathfrak{g}(\textsf{every}) & = & \lambda P\,[\lambda Q\,\forall x\ [P\{x\} \rightarrow Q\{x\}]], \\ \mathfrak{g}(\textsf{the}) & = & \lambda P\,[\lambda Q\,\exists y\ [\forall x\ [P(x) \leftrightarrow x = y]\ \&\ Q\{y\}]], \\ \mathfrak{g}(\textsf{a}) & = & \lambda P\,[\lambda Q\,\exists x\ [P\{x\}\ \&\ Q\{x\}]]. \end{array}$$

Intuicyjnie, rodzajniki używane są w zwrotach typu *Każdy (ten, pewien) ktoś coś_robi*, przy czym P reprezentuje tego kogoś, zaś Q wykonywaną przez niego czynność. Zauważmy, że warunkiem dla the jest istnienie dokładnie jednego y spełniającego $P(y)$. Jest to niezmiernie mocny warunek, na szczęście semantyka możliwych światów powoduje, że ma być to dokładnie jeden taki obiekt w pewnym określonym świecie, co nieco osłabia jego siłę.

Te skomplikowane formuły zostaną wyjaśnione przy opisie reguł syntaktycznych, w których są używane. Zanim to nastąpi, chcielibyśmy zwrócić uwagę na pewne istotne własności takich translacji.

Po pierwsze, musimy wyróżnić dwa rodzaje „zbiorów własności indywiduów", w zależności od tego, czy są one reprezentowane przez stałe czy zmienne. Oczywiste jest, że dwie tożsame stałe mają te same własności (w danym świecie):

$$\Box \Big[a = b \ \leftrightarrow\ \lambda P\,[P\{a\}] = \lambda P\,[P\{b\}]\Big].$$

Natomiast dwie tożsame zmienne, jako że mają to samo wartościowanie we wszystkich światach, mają tożsame także intensje własności:

$$\forall x\ \forall y\ \Box \Big[x = y \ \leftrightarrow\ \hat{}\lambda P\,[P\{x\}] = \hat{}\lambda P\,[P\{y\}]\Big],$$

co dla stałych ewidentnie nie zachodzi. Oczywiście w banalny sposób wynika z tego także:

$$\forall x\ \forall y\ \Box\ [x = y \ \leftrightarrow\ \lambda P\,[P\{x\}] = \lambda P\,[P\{y\}]].$$

[4] Rozróżnienie między *własnością* a *faktem* bywa czasem subtelne. Stwierdzenie typu *Jan mieszka w Warszawie.* jest ewidentnie *faktem*, choć trudno uznać je za *własność* Jana. We wspomnianych pracach omawiana była jednak czysto temporalna charakterystyka tych pojęć, a z tej perspektywy niczym sie one nie różnią.

Definicja 2.8. Powiemy, że *zbiór własności* jest *niesprzeczny*, jeżeli istnieje indywiduum, które posiada każdą własność z tego zbioru. Powiemy, że zbiór ten jest *maksymalny*, jeżeli dodanie doń jakiejkolwiek własności powoduje utratę niesprzeczności. Maksymalny niesprzeczny zbiór własności danego *indywiduum* będziemy nazywali *sublimacją* tego indywiduum. Dla zbioru indywiduów α, zbiór własności posiadanych przez wszystkie jego elementy (a więc przecięcie ich sublimacji) nazwiemy *sublimacją uniwersalną* α, zaś zbiór własności posiadanych przez którykolwiek jego element (a więc sumę ich sublimacji) będziemy nazywać *sublimacją egzystencjalną* α. Sublimację indywiduum opisuje wyrażenie $\lambda P\,[P\{x\}]$, sublimację uniwersalną wyrażenie $\lambda P_\alpha\,[\lambda Q\,\forall x\;[P_\alpha\{x\} \to Q\{x\}]]$, zaś sublimację egzystencjalną wyrażenie $\lambda P_\alpha\,[\lambda Q\,\exists x\;[P_\alpha\{x\}\;\&\;Q\{x\}]]$, gdzie P_α jest funkcją charakterystyczną zbioru α.

Zbiór wszystkich fraz kategorii A oznaczany będzie przez P_A. Pierwsza reguła syntaktyczna PTQ wiąże zbiory podstawowe i frazy danego typu.

S1 $B_A \subseteq P_A$ dla dowolnej kategorii A.

2.2.3 Podstawowe reguły syntaktyczne i semantyczne

Pozostałe reguły syntaktyczne dotyczą wyrażeń złożonych. Każda taka reguła zawiera *operację strukturalną* (oznaczaną symbolem F_m) specyfikującą, jak wyrażenia należące do kategorii wejściowych mają zostać przekształcone na kategorię wyjściową. Zasadą jest, że każdej regule syntaktycznej Sn odpowiada reguła translacji Tn wyrażenia wyjściowego na odpowiednie wyrażenie języka IL.

Reguła „rodzajnik-rzeczownik pospolity"

Reguła ta łączy rodzajnik (kategorii *DET*) z rzeczownikiem pospolitym (kategorii *CN*), tworząc frazę rzeczownikową (kategorii T).

S2 Jeśli $\delta \in P_{DET}$ oraz $\xi \in P_{CN}$, to $F_2(\delta,\xi) = \delta\xi \in P_T$.[5]

Z regułą **S2** związana jest następująca reguła translacji:

T2 Jeśli $\delta \in P_{DET}$ oraz $\xi \in P_{CN}$, to $\mathfrak{g}(F_2(\delta,\xi)) = \mathfrak{g}(\delta)(^\frown\mathfrak{g}(\xi))$.

Przykład *2.1.* Rozważmy wyrażenia podstawowe $\mathsf{every} \in B_{DET} \subseteq P_{DET}$ oraz $\mathsf{man} \in B_{CN} \subseteq P_{CN}$ (dzięki zastosowaniu reguły **S1**). Wówczas $\mathsf{every\ man} \in P_T$ na podstawie reguły **T2**.
Zgodnie z regułą **T2**, fraza *every man* przekładana jest na język IL jako

$$\begin{array}{lcl} \mathfrak{g}\,(\mathsf{every})(^\frown\mathfrak{g}(\mathsf{man})) & \equiv & \lambda P\left[\lambda Q\,\forall x\;[P\{x\} \to Q\{x\}]\right](^\frown\mathsf{man}') \quad \equiv \\ \lambda Q\,\forall x\;[^\frown\mathsf{man}'\{x\} \to Q\{x\}] & \equiv & \lambda Q\,\forall x\;[\mathsf{man}'(x) \to Q\{x\}]. \end{array}$$

Zauważmy, że wyrażenie to w ogromnym stopniu przypomina translacje wyrażeń podstawowych kategorii T, czyli nazw własnych (po uwzględnieniu różnic wynikających z charakteru rodzajnika every). I o to właśnie chodzi, przecież przedstawiciele tej samej kategorii występują w tych samych miejscach w zdaniu i podlegają analogicznym przekształceniom podczas translacji.

Reguła „podmiot-orzeczenie"

Reguła ta łączy frazę rzeczownikową (kategorii T) i frazę czasownikową (kategorii IV) w zdanie (kategorii t), dokonując przy tym kontroli zgody podmiotu z orzeczeniem. Zauważmy, że rozważany fragent j. angielskiego nie zawiera rzeczowników w liczbie mnogiej, uwzględnienie ich dodatkowo skomplikowałoby regułę.

S4 Jeśli $\alpha \in P_T$ oraz $\delta \in P_{IV}$, to $F_4(\alpha,\delta) = \alpha\delta' \in P_t$, gdzie δ' powstaje z δ poprzez zastąpienie pierwszego występującego w nim czasownika $\beta \in B_{IV} \cup B_{TV} \cup B_{IV/t} \cup B_{IV/IV}$ jego formą w trzeciej osobie liczby pojedynczej.[6]

T4 Jeśli $\alpha \in P_T$ oraz $\delta \in P_{IV}$, to $\mathfrak{g}(F_4(\alpha,\delta)) = \mathfrak{g}(\alpha)(^\frown\mathfrak{g}(\delta))$.

[5] W oryginale reguła zawiera także warunek, że jeśli $\delta = \mathsf{a}$ zaś ξ rozpoczyna się samogłoską, to δ zastępowane jest $\delta' = \mathsf{an}$. Zauważmy jednak, że w zbiorze B_{CN} brak rzeczownika, do którego mógłby odnosić się ten warunek.

Przykład *2.2.* Rozważmy wyrażenia podstawowe $\mathsf{John} \in B_T \subseteq P_T$ oraz $\mathsf{walk} \in B_{IV} \subseteq P_{IV}$ (dzięki zastosowaniu reguły **S1**). Wówczas zdanie $\mathsf{John\ walks} \in P_t$ zgodnie z regułą **S4**.

Metoda syntaktycznego konstruowania frazy może być przedstawiona za pomocą *drzewa analizy.* Wszystkie liście takiego drzewa stanowią wyrażenia podstawowe. Pozostałe węzły etykietowane są przez wyrażenia utworzone z węzłów synów za pomocą pewnej reguły syntaktycznej. Numer operacji strukturalnej występującej w tej regule umieszczany jest po prawej stronie etykiety. Drzewa analizy niezmiernie przypominają drzewa struktur frazowych w gramatykach transformacyjnych, jednak w tym wypadku faktycznie odzwierciedlają „historię wyprowadzenia" (ang. *derivation history*). Drzewa analizy okażą się szczególnie przydatne w przypadku zdań niejednoznacznych semantycznie.

John walks, **4**

John walk

Zgodnie z regułą **T4**, zdanie *John walks* przekładane jest na język IL jako

$$\mathfrak{g}\,(\mathsf{John})(\hat{}\,\mathfrak{g}(\mathsf{walk})) \;\equiv\; \lambda P[P\{j\}](\hat{}\,\mathsf{walk}') \;\equiv\; \hat{}\,\mathsf{walk}'\{j\} \;\equiv\; \mathsf{walk}'(j).$$

Na tym przykładzie widać, jak skomplikowane formuły będące efektem translacji na język IL dają się przekształcić na równoważne im proste i „naturalne" formuły tego języka.

A oto przykład zdania wymagającego zastosowania obu powyższych reguł.

Przykład *2.3.* Analiza zdania *Every man talks.* zawierającego wyrażenia podstawowe $\mathsf{every} \in B_{DET} \subseteq P_{DET}$, $\mathsf{man} \in B_{CN} \subseteq P_{CN}$ oraz $\mathsf{talk} \in B_{IV} \subseteq P_{IV}$ przedstawiona jest za pomocą poniższego drzewa.

every man talks, **4**

every man, **2** talk

every man

Zdanie *Every man talks.* przekładane jest na język IL jako

$$\begin{array}{lcl} \mathfrak{g}\,(\mathsf{Every\ man})(\hat{}\,\mathfrak{g}(\mathsf{talk})) & \equiv & \lambda Q\,\forall x\,[\mathsf{man}'(x) \to Q\{x\}](\hat{}\,\mathsf{talk}') \quad \equiv \\ \forall x\,[\mathsf{man}'(x) \to \hat{}\,\mathsf{talk}'\{x\}] & \equiv & \forall x\,[\mathsf{man}'(x) \to \mathsf{talk}'(x)]. \end{array}$$

Zauważmy, że przekłady zdań *John talks* czy *Every man talks* są reprezentowane w sposób odpowiadający angielskiemu czasowi ciągłemu (ang. *present continuous*), tzn. *John is talking* (właśnie teraz). Jest to szczegół który Montague zignorował prawdopodobnie dla prostoty; w języku polskim kwestia ta zresztą się nie pojawia.

Reguła „przysłówek zdaniowy–zdanie"

Reguła ta łączy przysłówki zdaniowe (kategorii t/t) ze zdaniem (kategorii t), dając w rezultacie (zmodyfikowane) zdanie.

S9 Jeśli $\delta \in P_{t/t}$ oraz $\phi \in P_t$, to $F_6(\delta, \phi) = \delta\ \phi \in P_t$.

T9 Jeśli $\delta \in P_{t/t}$ oraz $\phi \in P_t$, to $\mathfrak{g}(F_6(\delta, \phi)) = \mathfrak{g}(\delta)(\hat{}\,\mathfrak{g}(\phi))$.

Przykład *2.4.* Weźmy zdanie *Necessarily John walks.* i rozważmy dwa jego rozbiory.

necessarily John walks, **6**

necessarily *John walks,* **4**

John walk

necessarily John walks, **10,** 6

John *necessarily* $\mathtt{he}_6$ *walks,* **6**

necessarily $\mathtt{he}_6$ *walks,* **4**

$\mathtt{he}_6$ walk

Pierwszy rozbiór przekładany jest na język IL w prosty sposób jako:

$$\begin{array}{lcl}
\mathfrak{g}\ (\textit{necessarily John walks}) & \equiv & \mathfrak{g}\,(\mathsf{necessarily})\ ({}^\wedge\mathfrak{g}(\textit{John walks})) \quad \equiv \\
\lambda p\,[\Box{}^\vee p]({}^\wedge\mathsf{walk}'\,(j)) & \equiv & \Box{}^{\vee\wedge}\mathsf{walk}'\,(j) \quad \equiv \quad \Box\,\mathsf{walk}'\,(j)
\end{array}$$

Z kolei drugi rozbiór jest odrobinę bardziej skomplikowany.

$$\begin{array}{lcl}
\mathfrak{g}\ (\textit{necessarily John walks}) & \equiv & \mathfrak{g}\,(\mathsf{John})\ ({}^\wedge\lambda x_6\,\mathfrak{g}(\textit{necessarily}\ \texttt{he}_6\ \textit{walks})) \quad \equiv \\
\lambda P\,[P\{j\}]\left({}^\wedge\lambda x_6\,\lambda p\,[\Box{}^\vee p]({}^\wedge\mathsf{walk}'\,(x_6))\right) & \equiv & {}^\wedge\lambda x_6\,\Box\left[{}^{\vee\wedge}\mathsf{walk}'\,(x_6)\right]\{j\} \quad \equiv \\
\Box\,[\lambda x_6\,\mathsf{walk}'\,(x_6)]\,(j) & \equiv & \Box\,\mathsf{walk}'\,(j)
\end{array}$$

Ostatnie dwa przekształcenia związane są z przekroczeniem bariery operatora $\Box$. Chociaż jednak nie znamy aksjomatyzacji logiki IL, możemy bezpiecznie założyć, że zawiera ona aksjomaty (lub dające się z nich wywieść tautologie) $\Box\,(A \to B) \ \to\ (\Box\,A \to \Box\,B)$ oraz $(\lambda x\,\Box\,[\varphi(x)])(y) \ \leftrightarrow\ \Box\,[\lambda x\,[\varphi(x)]\,(y)]$. Ten ostatni warunkowany jest faktem, że wartościowanie dowolnej zmiennej jest stałe we wszystkich światach, i można bez trudu sprawdzić, że obie strony równoważności mają tę samą wartość semantyczną. Jednak aksjomat ten nie dotyczy stałych.

Aby wyrażenia języka IL będące wynikiem translacji $\mathfrak{g}$ odpowiednich drzew rozbioru fraz j. angielskiego w gramatyce PTQ stanowiły ich interpretację semantyczną zgodną z intuicją, Montague wprowadza specjalne *postulaty* (ang. *meaning postulates*). Są to aksjomaty specyficzne tworzące pewną teorię IL, umożliwiającą „sensowną” interpretację fraz PTQ.[7] Tak więc aby uzyskać „naturalną” interpretację $\Box\,\mathsf{walk}'\,(j)$, Montague wprowadza postulat MP2.

MP2 $\exists x\,\Box\,(x = \alpha)$, gdzie $\alpha \in \{\mathsf{j}, \mathsf{m}, \mathsf{b}\}$.

<u>Reguła „przysłówek przyczasownikowy–fraza czasownikowa”</u>

Reguła ta łączy przysłówki przyczasownikowe (kategorii *IAV*; typu $\langle\langle s, \langle e,t\rangle\rangle, \langle e,t\rangle\rangle$) z frazą czasownikową, dając w rezultacie (zmodyfikowaną) frazę czasownikową.

S10 Jeśli $\delta \in P_{IAV}$ oraz $\beta \in P_{IV}$, to $F_7(\delta,\beta) = \delta\ \beta \in P_{IV}$.

T10 Jeśli $\delta \in P_{IAV}$ oraz $\beta \in P_{IV}$, to $\mathfrak{g}F_7(\delta,\beta) = \mathfrak{g}(\beta)({}^\wedge\mathfrak{g}(\delta))$.

<u>Przykład</u> *2.5.* Dla zdania *John walks slowly.* uzyskujemy drzewo rozbioru widoczne po lewej, które następnie interpretujemy w sposób przedstawiony po prawej.

John walks slowly, **4**

John   walk slowly, **7**

walk   slowly

$$\begin{array}{lcl}
\mathfrak{g}\ (\textit{John walks slowly}) & \equiv & \mathfrak{g}\,(\mathsf{John})\ ({}^\wedge\mathfrak{g}(\mathsf{walk\ slowly})) \quad \equiv \\
\lambda P\,[P\{j\}]\left({}^\wedge\mathfrak{g}(\mathsf{slowly})\,\mathfrak{g}(\mathsf{walk})\right) & \equiv & {}^\wedge\mathsf{slowly}'\,({}^\wedge\mathsf{walk}')\,\{j\} \quad \equiv \\
\mathsf{slowly}'\,({}^\wedge\mathsf{walk}')\,(j) & &
\end{array}$$

Montague zauważa, że jeżeli *John walks slowly.*, to w szczególności *John walks.*, ale jeśli *John walks allegedly.*, to takiego wniosku nie możemy już wyprowadzić. Wprowadza więc kolejny postulat mający zastosowanie do tych przysłówków, które nie poddają w wątpliwość zajścia samej czynności.

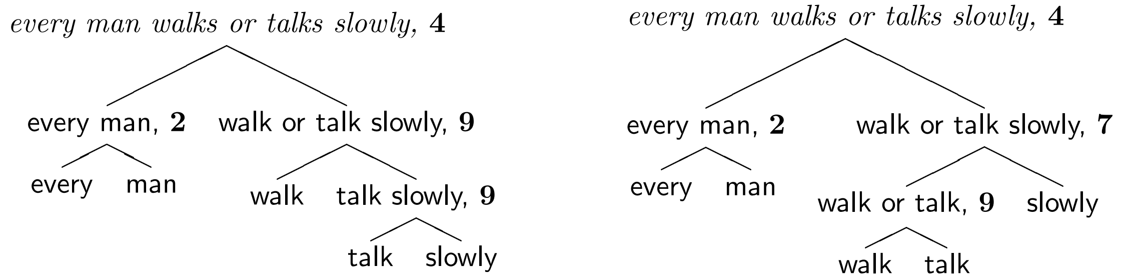
MP10 $\forall x\,\forall P\,\Box\,[\gamma\,(P(x)) \ \to\ P\{x\}]$, gdzie $\gamma \in \{\mathsf{slowly}, \mathsf{rapidly}\}$.

Przysłówki mogą być aplikowane do dowolnej frazy czasownikowej; np. zdanie *John walks slowly allegedly.* zostanie przetranslowane na IL jako $\mathsf{allegedly}'\,({}^\wedge\mathsf{slowly}'\,({}^\wedge\mathsf{walk}'))\,(j)$. Poniżej przedstawimy przykład, w którym przysłówek wprowadza niejednoznaczność (syntaktyczną i semantyczną).

---

[7] Aksjomaty formalizmu języka IL w ogóle nie zostały podane w pracy [Dowty i in., 1981], nie mówiąc o dowodzie jego pełności; por. [Montague, 1970c].

Przykład *2.6.* Dwa rozbiory zdania *Every man walks or talks slowly.* przedstawione są poniżej.



Pierwszy przekład tego zdania na język IL jest następujący:

$$\begin{aligned}
&\mathfrak{g}\,(\textit{Every man walks or talks slowly}) && \equiv\\
&\mathfrak{g}\,(\textsf{Every man})(\hat{}\,\mathfrak{g}(\textsf{walk or talk slowly})) && \equiv\\
&\lambda P\,\forall x\,[\textsf{man}'(x) \to P\{x\}]\left(\hat{}\lambda y\,[\mathfrak{g}(\textsf{walk})(y) \vee \mathfrak{g}(\textsf{talk slowly})(y)]\right) && \equiv\\
&\forall x\,\left[\textsf{man}'(x) \;\to\; \hat{}\lambda y\,[\textsf{walk}'(y) \vee \textsf{slowly}'\hat{}(\textsf{talk}')(y)]\,\{x\}\right] && \equiv\\
&\forall x\,[\textsf{man}'(x) \;\to\; \textsf{walk}'(x) \vee \textsf{slowly}'\hat{}(\textsf{talk}')(x)]
\end{aligned}$$

Natomiast druga interpretacja, w równym stopniu zgodna z intuicją, przebiega jak następuje:

$$\begin{aligned}
&\mathfrak{g}\,(\textit{Every man walks or talks slowly}) && \equiv\\
&\mathfrak{g}\,(\textsf{Every man})(\hat{}\,\mathfrak{g}(\textsf{walk or talk slowly})) && \equiv\\
&\lambda P\,\forall x\,[\textsf{man}'(x) \to P\{x\}]\left(\hat{}\,\mathfrak{g}(\textsf{slowly})\left(\hat{}\,\mathfrak{g}(\textsf{walk or talk})\right)\right) && \equiv\\
&\forall x\left[\textsf{man}'(x) \;\to\; \left[\hat{}\textsf{slowly}'\left(\hat{}\lambda y\,[\textsf{walk}'(y) \vee \textsf{talk}'(y)]\right)\right]\{x\}\right] && \equiv\\
&\forall x\left[\textsf{man}'(x) \;\to\; \textsf{slowly}'\left(\hat{}\lambda y\,[\textsf{walk}'(y) \vee \textsf{talk}'(y)]\right)(x)\right] && \equiv\\
&\forall x\left[\textsf{man}'(x) \;\to\; \textsf{slowly}'\left(\hat{}[\textsf{walk}'(x) \vee \textsf{talk}'(x)]\right)\right]
\end{aligned}$$

Reguły spójnikowe

Reguły te łączą dwa zdania (kategorii t) za pomocą spójników and oraz or w zdanie złożone.

S11a Jeśli $\phi, \psi \in P_t$, to $F_8(\phi, \psi) = \phi$ and $\psi \in P_t$.
S11b Jeśli $\phi, \psi \in P_t$, to $F_9(\phi, \psi) = \phi$ or $\psi \in P_t$.
T11a Jeśli $\phi, \psi \in P_t$, to $\mathfrak{g}(F_8(\phi, \psi)) = \mathfrak{g}(\phi)\ \&\ \mathfrak{g}(\psi)$.
T11b Jeśli $\phi, \psi \in P_t$, to $\mathfrak{g}(F_9(\phi, \psi)) = \mathfrak{g}(\phi) \ \vee\ \mathfrak{g}(\psi)$.

Przykład *2.7.* Z dwóch zdań z przykładów 2.2 i 2.3 *John walks.* oraz *Every man walks.* (oba kategorii t) reguła **S11a** tworzy zdanie *John walks and every man walks.*, zaś reguła **T11a** przekłada je na formułę $\textsf{walk}'(j)\ \&\ \forall x\,[\textsf{man}'(x) \to \textsf{walk}'(x)]$.

Podobne reguły sformułowane zostały dla kordynacji fraz czasownikowych (kategorii IV) oraz fraz rzeczownikowych (kategorii T).

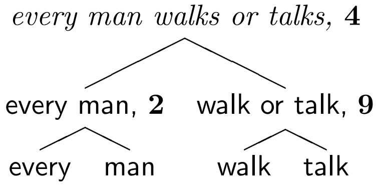
S12a Jeśli $\delta, \gamma \in P_{IV}$, to $F_8(\delta, \gamma) = \delta$ and $\gamma \in P_{IV}$.
S12b Jeśli $\delta, \gamma \in P_{IV}$, to $F_9(\delta, \gamma) = \delta$ or $\gamma \in P_{IV}$.
S13 Jeśli $\alpha, \beta \in P_T$, to $F_9(\alpha, \beta) = \alpha$ or $\beta \in P_T$.

Dla fraz rzeczownikowych spójnik *and* został pominięty, gdyż zdanie takie wymagałoby użycia czasownika w liczbie mnogiej, a przypadek taki nie jest w tej prostej gramtyce rozważany. Zauważmy, że przekłady takich fraz na IL są nieco bardziej skomplikowane niż w przypadku zdań.

T12a Jeśli $\delta, \gamma \in P_{IV}$, to $\mathfrak{g}(F_8(\delta, \gamma)) = \lambda x\,[\mathfrak{g}(\delta)(x)\ \&\ \mathfrak{g}(\gamma)(x)]$.
T12b Jeśli $\delta, \gamma \in P_{IV}$, to $\mathfrak{g}(F_9(\delta, \gamma)) = \lambda x\,[\mathfrak{g}(\delta)(x) \ \vee\ \mathfrak{g}(\gamma)(x)]$.
T13 Jeśli $\alpha, \beta \in P_T$, to $\mathfrak{g}(F_9(\alpha, \beta)) = \lambda P\,[\mathfrak{g}(\alpha)(P) \ \vee\ \mathfrak{g}(\beta)(P)]$.

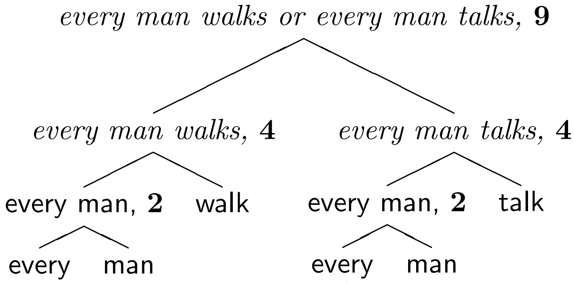
Powyższe reguły są potrzebne, gdyż nie wystarczy sprowadzić skoordynowanej frazy czasownikowej czy rzeczownikowej do zdań połączonych odpowiednim spójnikiem, gdyż może to prowadzić do zmiany semantyki w przypadku fraz rzeczownikowych zawierających rodzajniki (kwantyfikatory), niezależnie od faktu, że proste zdania typu *John walks and talks.* oraz *John walks and John talks.* są równoznaczne.

Przykład *2.8.* Analiza zdania *Every man walks or talks.* zawierającego wyrażenia podstawowe $\mathsf{every} \in B_{DET} \subseteq P_{DET}$, $\mathsf{man} \in B_{CN} \subseteq P_{CN}$ oraz $\mathsf{walk}, \mathsf{talk} \in B_{IV} \subseteq P_{IV}$ przedstawiona jest za pomocą poniższego drzewa, co jest przekładane na język IL w następujący sposób:



$$\begin{aligned}
&\mathfrak{g}\,(\textit{Every man walks or talks}) && \equiv\\
&\mathfrak{g}\,(\mathsf{Every\ man})(\hat{}\,\mathfrak{g}(\mathsf{walk\ or\ talk})) && \equiv\\
&\lambda Q\,\forall x\,[\mathsf{man}'(x) \rightarrow Q\{x\}]\left(\hat{}\lambda y\,[\mathfrak{g}(\mathsf{walk})(y) \vee \mathfrak{g}(\mathsf{talk})(y)]\right) && \equiv\\
&\forall x\,\left[\mathsf{man}'(x) \rightarrow \hat{}\lambda y\,[\mathsf{walk}'(y) \vee \mathsf{talk}'(y)]\,\{x\}\right] && \equiv\\
&\forall x\,[\mathsf{man}'(x) \rightarrow \mathsf{walk}'(x) \vee \mathsf{talk}'(x)].
\end{aligned}$$

Natomiast analiza zdania *Every man walks or every man talks.* zawierającego te same wyrażenia podstawowe przedstawiona jest za pomocą poniższego drzewa.



Zdanie to przekładane jest na język IL w następujący sposób.

$$\begin{aligned}
&\mathfrak{g}\,(\textit{Every man walks or every man talks}) && \equiv\\
&\mathfrak{g}(\textit{Every man walks}) \vee \mathfrak{g}(\textit{Every man talks}) && \equiv\\
&\forall x\,[\mathsf{man}'(x) \rightarrow \mathsf{walk}'(x)] \vee \forall x\,[\mathsf{man}'(x) \rightarrow \mathsf{talk}'(x)].
\end{aligned}$$

Jest ewidentne, że formuły będące rezultatem translacji tych zdań na język IL nie są logicznie równoważne.

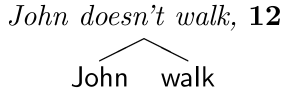
Reguła negacji

Negacja wprowadzana jest w gramatyce PTQ w sposób możliwie najprostszy — reguła S17 jest „zanegowanym” odpowiednikiem reguły S4. Tak więc reguła ta łączy frazę rzeczownikową (kategorii T) i frazę czasownikową (kategorii IV) w zdanie (kategorii t), tyle że uwzględniając negację.

S17 Jeśli $\alpha \in P_T$ oraz $\delta \in P_{IV}$, to $F_{12}(\alpha, \delta) = \alpha\delta' \in P_t$, gdzie δ' powstaje z δ poprzez zastąpienie pierwszego występującego w nim czasownika $\beta \in B_{IV} \cup B_{TV} \cup B_{IV/t} \cup B_{IV/IV}$ jego zanegowaną formą trzeciej osoby liczby pojedynczej.

T17 Jeśli $\alpha \in P_T$ oraz $\delta \in P_{IV}$, to $\mathfrak{g}(F_{12}(\alpha, \delta)) = \neg\mathfrak{g}(\alpha)(\hat{}\,\mathfrak{g}(\delta))$.

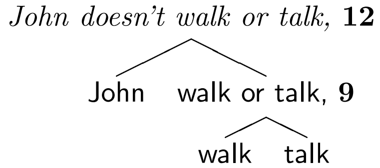
Przykład *2.9.* Rozważmy ponownie wyrażenia podstawowe $\mathsf{John} \in B_T \subseteq P_T$ oraz $\mathsf{walk} \in B_{IV} \subseteq P_{IV}$. Wówczas zdanie *John doesn't walk* $\in P_t$ zgodnie z regułą **S17**, a jego drzewo rozbioru prezentowane jest obok. Zgodnie z regułą **T17**, zdanie *John does'n walk.* przekładane jest na język IL jako:



$$\neg\mathfrak{g}\,(\mathsf{John})(\hat{}\,\mathfrak{g}(\mathsf{walk})) \equiv \neg\lambda P[P\{j\}]\,(\hat{}\,\mathsf{walk}') \equiv \neg\hat{}\,\mathsf{walk}'\,\{j\} \equiv \neg\mathsf{walk}'(j).$$

Zdanie *John walks or doesn't talk.* nie daje się zaakceptować w gramatyce PTQ, gdyż nie można do niego zastosować ani reguły S4, ani S12b. Rozważymy więc inne zdanie z negacją i spójnikiem.

Przykład *2.10.* Zdanie *John doesn't walk or talk.* ma następujący rozkład w PTQ i interpretację w IL:



$$\begin{aligned}
&\mathfrak{g}\,(\textit{John doesn't walk or talk}) && \equiv\\
&\neg\lambda P\,[P\{j\}]\left(\hat{}\,\mathfrak{g}(\mathsf{walk\ or\ talk})\right) && \equiv\\
&\neg\hat{}\lambda x\,[\mathsf{walk}'(x) \vee \mathsf{talk}'(x)]\,\{j\} && \equiv\\
&\neg[\mathsf{walk}'(j) \vee \mathsf{talk}'(j)] \equiv \neg\mathsf{walk}'(j)\ \&\ \neg\mathsf{talk}'(j)
\end{aligned}$$

Także dołączenie przysłówków do zdań zawierających negację nie sprawia żadnych kłopotów: ponieważ reguła S17 łączy frazę rzeczownikową i czasownikową w zdanie, nie grozi użycie jej i reguły S10 w niepoprawnej kolejności. Tak więc banalnie zdanie *John doesn't walk slowly.* translowane jest na formułę IL $\neg[\mathsf{slowly}'\,({}^\frown\mathsf{walk}')\,(j)]$.

Zauważmy, że reguły S4 i S17 sformułowane są opisowo, w sposób zgoła nieformalny, bez sprecyzowania sposobu przekształcania czasowników na ich formę zanegowaną. Tak więc akceptowane są zarówno formy *does not* δ jak i *doesn't* δ, a dla czasownika be odpwoiednio *is not* i *isn't*, bez wskazania różnic między tymi fundamentalnie różnymi przekształceniami.

2.2.4 Reguły wykorzystujące zaimki

W gramatyce PTQ Montague zaimki pełnią bardzo istotną rolę. Są one wykorzystywane (w sposób nieco sztuczny) nawet w zdaniach, w których *de facto* żadne zaimki nie występują!

Reguła wiązania zaimków anaforycznych

Reguła ta zamienia pierwsze wystąpienie „indeksowanego" zaimka $\mathtt{he}_n$ na frazę rzeczownikową (kategorii T), zaś pozostałe wystąpienia na odpowiednie formy (zgodność rodzaju) „zwykłego" zaimka. W ten sposób dokonywane jest wiązanie zaimka z odpowiadającym mu rzeczownikiem. Zauważmy, że operacja strukturalna **S14** posiada dodatkowy indeks identyfikujący „indeksowany" zaimek, który jest w danej instancji reguły zastępowany. Dotyczy to wszystkich reguł, w których występują takie zaimki.

S14 Jeśli $\alpha \in P_T$ oraz $\phi \in P_t$, to $F_{10,n}(\alpha, \phi) \in P_t$, gdzie:

$$F_{10,n}(\alpha,\phi) = \begin{cases} \text{powstaje z } \phi \text{ przez zastąpienie pierwszego wystąpienia } \mathtt{he}_n/\mathtt{him}_n \\ \text{przez } \alpha \text{ oraz pozostałych wystąpień } \mathtt{he}_n/\mathtt{him}_n \text{ przez } \mathsf{he/him,\ she/her} & \text{dla } \alpha \neq \mathtt{he}_k \\ \text{lub } \mathsf{it} \text{ w zależności od rodzaju pierwszego } \beta \in B_{CN} \cup B_T \text{ w } \alpha. \\ \\ \text{powstaje z } \phi \text{ przez zastąpienie wszystkich wystąpień } \mathtt{he}_n/\mathtt{him}_n & \text{dla } \alpha = \mathtt{he}_k. \\ \text{przez } \mathtt{he}_k/\mathtt{him}_k \end{cases}$$

Dzięki takiemu sformułowaniu reguły ma ona zastosowanie do wszystkich zdań i fraz rzeczownikowych. Na przykład:

$F_{10,4}(\mathsf{Mary},\ \mathtt{he}_4\ \mathsf{walks\ and}\ \mathtt{he}_4\ \mathsf{talks}) = \mathsf{Mary\ walks\ and\ she\ talks},$

$F_{10,4}(\mathtt{he}_5,\ \mathtt{he}_4\ \mathsf{walks\ and}\ \mathtt{he}_4\ \mathsf{talks}) = \mathtt{he}_5\ \mathsf{walks\ and}\ \mathtt{he}_5\ \mathsf{talks},$

zaś $F_{10,5}(\mathsf{Mary},\ \mathtt{he}_4\ \mathsf{walks\ and}\ \mathtt{he}_4\ \mathsf{talks}) = \mathtt{he}_4\ \mathsf{walks\ and}\ \mathtt{he}_4\ \mathsf{talks}.$

Z regułą **S14** związana jest następująca reguła translacji:

T14 Jeśli $\alpha \in P_T$ oraz $\phi \in P_t$, to $\mathfrak{g}(F_{10,n}(\alpha,\phi)) = \mathfrak{g}(\alpha)[{}^\frown\lambda x_n\, \mathfrak{g}(\phi)]$.

Przykład *2.11.* Analiza zdania *A woman walks and she talks.* przedstawiona jest za pomocą poniższego drzewa, przekładanego na język IL jako:

$$\begin{array}{ll}
\mathfrak{g}\ (a\ woman\ walks\ and\ she\ talks) & \equiv \\
\mathfrak{g}\,(\mathsf{a\ woman})\left({}^\frown\lambda x_2\, \mathfrak{g}(\mathtt{he}_2\ \mathsf{walks\ and}\ \mathtt{he}_2\ \mathsf{talks})\right) & \equiv \\
\lambda Q\, \exists x\ [\mathsf{woman}'(x)\ \&\ Q\{x\}]\left({}^\frown\lambda x_2\, (\mathfrak{g}(\mathtt{he}_2\ \mathsf{walks})\ \&\ \mathfrak{g}(\mathtt{he}_2\ \mathsf{talks}))\right) & \equiv \\
\exists x \left[\mathsf{woman}'(x)\ \&\ {}^\frown\lambda x_2 \left(\mathfrak{g}(\mathtt{he}_2)[{}^\frown\mathfrak{g}(\mathsf{walk})]\ \&\ \mathfrak{g}(\mathtt{he}_2)[{}^\frown\mathfrak{g}(\mathsf{talk})]\right)\{x\}\right] & \equiv \\
\exists x \left[\mathsf{woman}'(x)\ \&\ {}^\frown\lambda x_2 \left(\lambda P\,[P\{x_2\}]({}^\frown\mathsf{walk}')\ \&\ \lambda P\,[P\{x_2\}]({}^\frown\mathsf{talk}')\right)\{x\}\right] & \equiv \\
\exists x \left[\mathsf{woman}'(x)\ \&\ {}^\frown\lambda x_2 \left(\mathsf{walk}'(x_2)\ \&\ \mathsf{talk}'(x_2)\right)\{x\}\right] & \equiv \\
\exists x\ [\mathsf{woman}'(x)\ \&\ \mathsf{walk}'(x)\ \&\ \mathsf{talk}'(x)] &
\end{array}$$

A woman walks and she talks, **10,** 2
- a woman, **2**
  - a
  - woman
- $\mathtt{he}_2$ *walks and* $\mathtt{he}_2$ *talks*, **8**
  - $\mathtt{he}_2$ *walks*, **4**
    - $\mathtt{he}_2$
    - walk
  - $\mathtt{he}_2$ *talks*, **4**
    - $\mathtt{he}_2$
    - talk

W regule **S14** nie ma żadnych warunków na wystąpienia „indeksowanych” zaimków w zdaniu. Tak więc reguła ta może być zastosowana dla każdej frazy rzeczownikowej, co zwiększa liczbę analiz zdań, niejednokrotnie posiadających identyczną translację na język IL.

Przykład *2.12.* Zdanie *John walks.* z przykładu 2.2 może posiadać następujące drzewo analizy.

John walks, **10,** 1
- John
- $\mathtt{he}_1$ *walks*, **4**
  - $\mathtt{he}_1$
  - walk

Takiej analizie omawianego zdania odpowiada następujący przekład na język IL:

$$\begin{array}{lclc}
\mathfrak{g}\ (\mathit{John\ walks}) & \equiv & \mathfrak{g}\,(\mathsf{John})\ ({}^\frown\lambda x_1\, \mathfrak{g}(\mathtt{he}_1\ \mathsf{walks})) & \equiv \\
\lambda P\,[P\{j\}]\left({}^\frown\lambda x_1\, \mathfrak{g}(\mathtt{he}_1)\ ({}^\frown\mathfrak{g}(\mathsf{walk}))\right) & \equiv & {}^\frown\lambda x_1 \left(\lambda R\,[R\{x_1\}]\,({}^\frown\mathsf{walk}')\right)\ \{j\} & \equiv \\
\lambda x_1\,({}^\frown\mathsf{walk}'\,\{x_1\})\,(j) & \equiv & \mathsf{walk}'(j) &
\end{array}$$

Jak widać, translacja ta jest równoważna do przedstawionej w przykładzie 2.2.

Zdania zawierające czasowniki przechodnie (a więc wymagające dopełnień), mają w gramatyce PTQ dość skomplikowaną i niejednoznaczną semantykę, co niejednokrotnie wymaga stosowania reguły **S14**. Dlatego odpowiednie reguły omówimy dopiero teraz.

Reguła „orzeczenie-dopełnienie rzeczownikowe”

Reguła ta łączy czasowniki przechodnie (kategorii *TV*) z frazą rzeczownikową (kategorii *T*), tworząc frazę czasownikową (kategorii *IV*).

S5 Jeśli $\delta \in P_{TV}$ oraz $\beta \in P_T$, to $F_5(\delta,\beta) \in P_{IV}$, gdzie:

$$F_5(\delta,\beta) = \begin{cases} \delta\ \mathtt{him}_n & \text{dla } \beta = \mathtt{he}_n, \\ \delta\,\beta & \text{dla } \beta \neq \mathtt{he}_n. \end{cases}$$

T5 Jeśli $\delta \in P_{TV}$ oraz $\beta \in P_T$, to $\mathfrak{g}(F_5(\delta,\beta)) = \mathfrak{g}(\delta)({}^\frown\mathfrak{g}(\beta))$.

Translacja frazy rzeczownikowej jest typu $\langle\langle s,\langle e,t\rangle\rangle,t\rangle$, stąd jej intensja jest typu $d = \langle s,\langle\langle s,\langle e,t\rangle\rangle,t\rangle\rangle$, czyli jest to *własność własności obiektu*. Jak łatwo zauważyć, translacja czasownika przechodniego jest typu $\langle d,\langle e,t\rangle\rangle$, a więc wyrażenie uzyskiwane za pomocą funkcji F_5 wiążącej czasownik z dopełnieniem jest typu $\langle e,t\rangle$, a powiązanie go z kolei z obiektem reprezentującym podmiot (typu e) daje w rezultacie zdanie. Wyrażenie typu d może mieć w szczególności postać ${}^\frown\lambda P\,\varphi\,[P\{x\}]$, gdzie $P \in V_{\langle s,\langle e,t\rangle\rangle}$, $x \in V_e$ oraz $\varphi \in \mathcal{E}_t$. Wszystkie translacje dopełnień w rozważanych przykładach zdań zawierających czasowniki przechodnie mają taką właśnie postać.

Przykład *2.13.* Analiza zdań *John loves Mary.* oraz *John doesn't love Mary.* widnieje poniżej.

John loves Mary, **4**
- John
- love Mary, **5**
  - love
  - Mary

John doesn't love Mary, **4**
- John
- doesn't love Mary, **12**
  - love
  - Mary

Pierwsze ze zdań przekładane jest na język IL jako

$$
\begin{array}{lcll}
\mathfrak{g}\ (\textit{John loves Mary}) & \equiv & \mathfrak{g}\,(\mathsf{John})\ ({}^\frown\mathfrak{g}(\mathsf{love\ Mary})) & \equiv\\
\lambda P\,[P\{j\}]\left({}^\frown\mathfrak{g}(\mathsf{love})\,\mathfrak{g}(\mathsf{Mary})\right) & \equiv & {}^\frown\mathsf{love}'\left({}^\frown\lambda P\,[P\{m\}]\right)\{j\} & \equiv\\
\mathsf{love}'\left(j,{}^\frown\lambda P\,[P\{m\}]\right) & \equiv & & \\
\lambda y\,\lambda x\left[\mathsf{love}'\left({}^\frown\lambda P\,[P\{y\}]\right)(x)\right](m)(j) & \equiv & \mathsf{love}'_\star(j,m) &
\end{array}
$$

zaś drugie jako:

$$
\begin{array}{lcll}
\mathfrak{g}\ (\textit{John doesn't love Mary}) & \equiv & \neg\mathfrak{g}\ (\mathsf{John})\ ({}^\frown\mathfrak{g}(\mathsf{love\ Mary})) & \equiv\\
\neg\lambda P\,[P\{j\}]\left({}^\frown\mathfrak{g}(\mathsf{love})\,\mathfrak{g}(\mathsf{Mary})\right) & \equiv & \neg{}^\frown\mathsf{love}'\left({}^\frown\lambda P\,[P\{m\}]\right)\{\mathrm{j}\} & \equiv\\
\neg\mathsf{love}'\left(j,{}^\frown\lambda P\,[P\{m\}]\right) & \equiv & & \\
\neg\lambda y\,\lambda x\left[\mathsf{love}'\left({}^\frown\lambda P\,[P\{y\}]\right)(x)\right](m)(j) & \equiv & \neg\mathsf{love}'_\star(j,m) &
\end{array}
$$

Jak wiadomo, nazwy własne reprezentowane są jako zbiory własności, czyli odpowiadają sublimacji indywiduów. Montague utrzymuje, że stwierdzenie czegokolwiek o sublimacji indywiduum jest równoznaczne ze stwierdzniem tego o indywiduum jako takim. Pogląd ten służy mu do uproszczenia wyrażenia $\mathsf{love}'\left(j,{}^\frown\lambda P\,[P\{m\}]\right)$ w sposób zaprezentowany w powyższym przykładzie. Czyni to za pomocą następującej definicji:

Definicja 2.9. Dla dowolnego wyrażenia $\delta \in \mathcal{E}_{f(IV)}$ (odpowiadającego czasownikom przechodnim; typu $\langle\langle s,\langle\langle s,\langle e,t\rangle\rangle,t\rangle\rangle,\langle e,t\rangle\rangle$) definiujemy wyrażenie $\delta_\star$ (typu $\langle e,\langle e,t\rangle\rangle$) w następujący sposób:

$$\delta_\star \quad \equiv_{def} \quad \lambda y\,\lambda x\left[\delta\left({}^\frown\lambda P\,[P\{y\}]\right)(x)\right].$$

Tak więc chociaż żadna własność posiadana przez Mary nie jest w translacji z powyższego przykładu realizowana, (jedyną relewantną własnością jest *bycie kochaną przez Johna*), przez co w końcowej wersji „zwykłej” translacji pozostaje wyrażenie $\lambda P\,[P\{m\}]$, ostatecznie po zastosowaniu powyższej definicji uzyskujemy prostą formułę zgodną z intuicją.

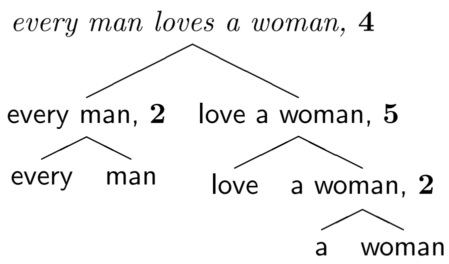
Rozważmy teraz nieco bardziej skomplikowane zdanie.

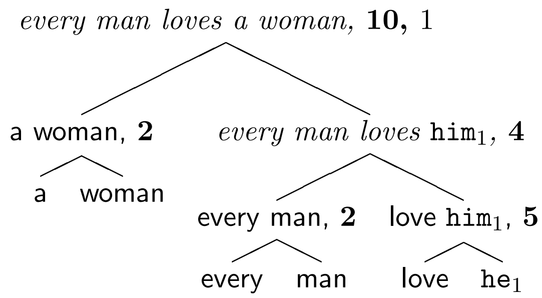
<u>Przykład</u> *2.14.* Zdanie *Every man loves a woman.* posiada dwa podstawowe rozbiory znacząco różniące się między sobą, których drzewa prezentujemy poniżej. Zaczniemy od translacji odpowiadającej drugiemu rozbiorowi.

$$
\begin{array}{ll}
\mathfrak{g}\ (\textit{every man loves a woman}) & \equiv\\
\mathfrak{g}\,(\mathsf{a\ woman})\,({}^\frown\lambda x_1\,\mathfrak{g}(\textit{every man loves }\mathtt{him}_1)) & \equiv\\
\lambda Q\,\exists x\,[\mathsf{woman}'(x)\ \&\ Q\{x\}]\left(\lambda x_1\,{}^\frown\mathfrak{g}(\mathsf{every\ man})\left({}^\frown\mathfrak{g}(\mathsf{love}\ \mathtt{him}_1)\right)\right) & \equiv\\
\exists x\left[\mathsf{woman}'(x)\ \&\ \lambda x_1\,\lambda P\,\forall y\ [\mathsf{man}'(y)\to P\{y\}]\left({}^\frown\mathsf{love}'\left({}^\frown\mathfrak{g}(\mathtt{he}_1)\right)\right)(x)\right] & \equiv\\
\exists x\left[\mathsf{woman}'(x)\ \&\ \lambda x_1\,\forall y\left[\mathsf{man}'(y)\to\mathsf{love}'\left({}^\frown\lambda R\,[R\{x_1\}]\right)(y)\right](x)\right] & \equiv\\
\exists x\left[\mathsf{woman}'(x)\ \&\ \forall y\left[\mathsf{man}'(y)\to\mathsf{love}'\left({}^\frown\lambda R\,[R\{x\}]\right)(y)\right]\right] & \equiv\\
\exists x\left[\mathsf{woman}'(x)\ \&\ \forall y\left[\mathsf{man}'(y)\to\lambda u\,\lambda v\left[\mathsf{love}'\left({}^\frown\lambda R\,[R\{u\}]\right)(v)\right](x)(y)\right]\right] & \equiv\\
\exists x\left[\mathsf{woman}'(x)\ \&\ \forall y\ [\mathsf{man}'(y)\to\mathsf{love}'_\star\,(y,x)]\right] &
\end{array}
$$

Jak widać, dla uzyskania wyniku zgodnego z intuicją znów niezbędne było zastosowanie definicji 2.9. Translacja pierwszego rozbioru na język IL jest, o dziwo, jeszcze trudniejsza.

$$
\begin{aligned}
&\mathfrak{g}\ (\textit{every man loves a woman}) && \equiv\\
&\mathfrak{g}\,(\mathsf{every\ man})\left({}^{\frown}\mathfrak{g}(\mathsf{loves\ a\ woman})\right) && \equiv\\
&\lambda P\,\forall y\,[\mathsf{man}'(y) \to P\{y\}]\left({}^{\frown}\mathsf{love}'\left({}^{\frown}\mathfrak{g}(\mathsf{a\ woman})\right)\right) && \equiv\\
&\forall y\left[\mathsf{man}'(y)\ \to\ {}^{\frown}\mathsf{love}'\left({}^{\frown}\lambda Q\,\exists x\,(\mathsf{woman}'(x)\ \&\ Q\{x\})\right)\{y\}\right] && \equiv\\
&\forall y\left[\mathsf{man}'(y)\ \to\ \mathsf{love}'\left(y, {}^{\frown}\lambda Q\,\exists x\,(\mathsf{woman}'(x)\ \&\ Q\{x\})\right)\right]
\end{aligned}
$$

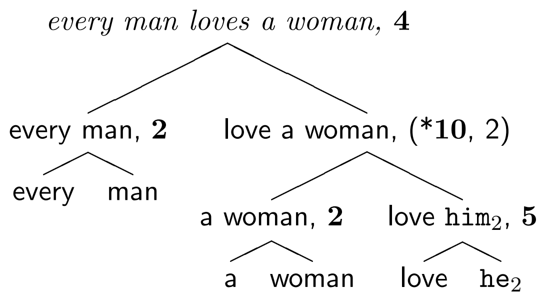




Ponieważ argument stałej predykatywnej love' nie jest postaci ${}^{\frown}\lambda P\,[P\{y\}]$, nie możemy zastosować doń definicji 2.9 bezpośrednio. Możemy natomiast posłużyć się postulatem MP1 (patrz poniżej), dzięki czemu uzyskujemy formułę dającą się przekształcić do oczekiwanej postaci:

$$
\begin{aligned}
&\forall y\left[\mathsf{man}'(y) \to {}^{\frown}\lambda Q\,\exists x\,[\mathsf{woman}'(x)\ \&\ Q\{x\}]\left\{{}^{\frown}\lambda v\,[\mathsf{love}'_{\star}(y,v)]\right\}\right] && \equiv\\
&\forall y\left[\mathsf{man}'(y) \to \lambda Q\,\exists x\,[\mathsf{woman}'(x)\ \&\ Q\{x\}]\left({}^{\frown}\lambda v\,[\mathsf{love}'_{\star}(y,v)]\right)\right] && \equiv\\
&\forall y\left[\mathsf{man}'(y) \to \exists x\left[\mathsf{woman}'(x)\ \&\ {}^{\frown}\lambda v\,[\mathsf{love}'_{\star}(y,v)]\{x\}\right]\right] && \equiv\\
&\forall y\left[\mathsf{man}'(y) \to \exists x\ [\mathsf{woman}'(x)\ \&\ \mathsf{love}'_{\star}(y,x)]\right]
\end{aligned}
$$

Zauważmy przy okazji, że rozbiór rozważanego zdania przedstawiony na drzewie obok, który mógłby dać bardziej „naturalną” translację od przedstawionej powyżej, nie jest akceptowany przez rozważaną gramatykę, gdyż funkcja $F_{10,n}$ z reguły S14 może być aplikowana wyłącznie do wyrażeń typu t, zaś love a woman ewidentnie do tej kategorii nie należy.



Wspomniany postulat można sformułować następująco:

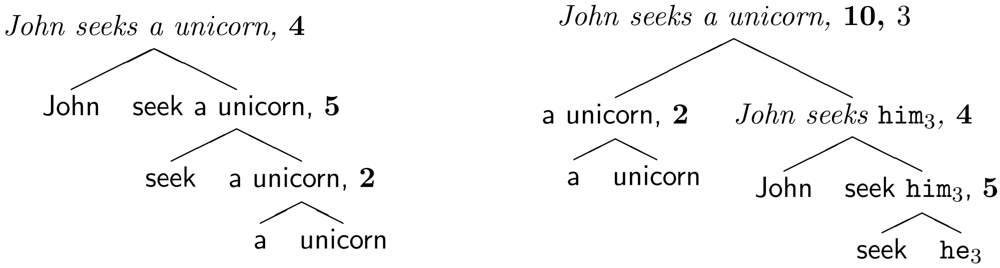
$$\mathbf{MP1}\ \ \forall x\,\forall\mathcal{P}\,\Box\left[\delta(u,\mathcal{P})\ \leftrightarrow\ \mathcal{P}\{{}^{\frown}\lambda v\,[\delta_{\star}(u,v)]\}\right],\ \text{gdzie } \delta\in\mathcal{E}_{f(IV)}-\{\mathsf{be},\ \mathsf{seek},\ \mathsf{conceive},\ldots\}.$$

Zgodnie z poglądami Montague (a także prezentujących je Dowty'ego, Walla i Petersa) powyższe „usprawnienia” nie są zabiegiem czysto technicznym. Problem stanowią czasowniki przechodnie typu *seek, conceive* czy *dream about* (nie objęte postulatem MP1), które mogą tworzyć prawdziwe frazy nawet w przypadkach, w których obiekt poszukiwań bądź marzeń w ogóle nie istnieje! Na przykład w zdaniu *John seeks a beautiful woman.* żadna z istniejących *pięknych kobiet* nie musi być obiektem poszukiwań Johna. Ponadto konieczne jest rozróżnienie pwyższego wypowiedzenia od *John seeks every beautiful woman.* czy *John seeks two beautiful women.*. Na koniec autorzy uznają, że konieczne jest rozróżnienie odmiennych rzeczowników pospolitych (kategorii CN typu $\langle e,t\rangle$), którym w modelu odpowiadają zbiory obiektów (poprzez ich funkcję charakterystyczną), nie istniejących w świecie bieżącym (a więc reprezentowanym przez zbiór pusty), takich jak np. jednorożce i centaury.

Okazuje się, że w celu uwzględnienia wszystkich tych wymagań niezbędne są pojęcia o złożoności nie mniejszej niż własności własności obiektów, a dopełnienia czasowników przechodnich są właśnie tego typu!

Spójrzmy teraz, jak to wygląda w praktyce.

Przykład *2.15.* Weźmy zdanie *John seeks a unicorn.*. Również to zdanie posiada dwa podstawowe drzewa rozbioru.



Tak jak w poprzednim przykładzie, najpierw przeanalizujemy drugi rozbiór.

$$\begin{array}{ll}
\mathfrak{g}\ (\textit{John seeks a unicorn}) & \equiv \\
\mathfrak{g}\,(\text{a unicorn})\,({}^\frown\lambda x_3\,\mathfrak{g}(\textit{John seeks}\ \texttt{him}_3)) & \equiv \\
\lambda Q\,\exists x\,[\text{unicorn}'(x)\ \&\ Q\{x\}]\left({}^\frown\lambda x_3\,\lambda R\,[R\{j\}]\left({}^\frown\mathfrak{g}(\text{seek}\ \texttt{him}_3)\right)\right) & \equiv \\
\exists x\left[\text{unicorn}'(x)\ \&\ \lambda x_3\,\lambda R\,[R\{j\}]\left({}^\frown\text{seek}'({}^\frown\lambda P\,[P\{x_3\}])\right)(x)\right] & \equiv \\
\exists x\left[\text{unicorn}'(x)\ \&\ \lambda x_3\left(\text{seek}'({}^\frown\lambda P\,[P\{x_3\}])(j)\right)(x)\right] & \equiv \\
\exists x\left[\text{unicorn}'(x)\ \&\ \text{seek}'({}^\frown\lambda P\,[P\{x\}])(j)\right] & \equiv \\
\exists x\ [\text{unicorn}'(x)\ \&\ \text{seek}'_\star(j,x)] &
\end{array}$$

Tak jak w przypadku „zwykłych" czasowników przechodnich, ostatnie przekształcenie wykorzystuje definicję 2.9. Interpretacja ta (*de re*) oznacza istnienie konkretnego jednorożca, którego John szuka, i nie jest spełniona w światach, w których jednorożce nie istnieją. Przejdźmy teraz do pierwszego rozbioru.

$$\begin{array}{llll}
\mathfrak{g}\ (\textit{John seeks a unicorn}) & \equiv & \mathfrak{g}\,(\text{John})\left({}^\frown\mathfrak{g}(\text{seeks a unicorn})\right) & \equiv \\
\lambda P\,[P\{j\}]\left({}^\frown\text{seek}'\left({}^\frown\mathfrak{g}(\text{a unicorn})\right)\right) & \equiv & {}^\frown\text{seek}'\left({}^\frown\lambda Q\,\exists x\,(\text{unicorn}'(x)\ \&\ Q\{x\})\right)\{j\} & \equiv \\
\text{seek}'\left({}^\frown\lambda Q\,\exists x\,(\text{unicorn}'(x)\ \&\ Q\{x\})\right)(j) & \equiv & \text{seek}'\left(j, {}^\frown\lambda Q\,\exists x\,(\text{unicorn}'(x)\ \&\ Q\{x\})\right) &
\end{array}$$

Tym razem nie możemy stosować postulatu MP1. Ponieważ powyższa interpretacja nie jest bynajmniej jasna, przeanalizujemy jeszcze jej wartość semantyczną w pewnym modelu $\mathcal{M}$.
$\left[\!\left[\text{seek}'\left(j, {}^\frown\lambda Q\,\exists x\,(\text{unicorn}'(x)\ \&\ Q\{x\})\right)\right]\!\right]^{\mathcal{M},\mathtt{w},\mathtt{t},g} = F(\text{believe}')(\langle\mathtt{w},\mathtt{t}\rangle)\left(F(j)\,(\langle\mathtt{w},\mathtt{t}\rangle), h_1\right)$,
przy czym $h_1\colon\ W\times T\longrightarrow D_{\langle\langle s,\langle e,t\rangle\rangle,t\rangle}$ jest taką funkcją, że
$h_1(\langle\mathtt{w}',\mathtt{t}'\rangle) = [\![\lambda\mathtt{Q}\,\exists\mathtt{x}\,(\text{unicorn}'(\mathtt{x})\ \&\ \mathtt{Q}\{\mathtt{x}\}]\!]^{\mathcal{M},\mathtt{w}',\mathtt{t}',\mathtt{g}} = \mathtt{h}_2$, gdzie $h_2\colon\ D_{\langle s,\langle e,t\rangle\rangle}\longrightarrow D_t$ jest z kolei taką funkcją, że $h_2(P) = [\![\exists x\,(\text{unicorn}'(x)\ \&{}^\smile Q(x))]\!]^{\mathcal{M},\mathtt{w}',\mathtt{t}',g_Q^P} = 1$ wtw, gdy istnieje $g'\in G_x^{g_Q^P}$ takie, że $[\![\text{unicorn}'(x)]\!]^{\mathcal{M},\mathtt{w}',\mathtt{t}',g'} = 1$ oraz $[\![Q(x)]\!]^{\mathcal{M},\mathtt{w}',\mathtt{t}',g'}(\langle\mathtt{w}',\mathtt{t}'\rangle) = 1$ wtw, gdy istnieje $g'\in G_x^{g_Q^P}$ takie, że $F(\text{unicorn}')(\langle\mathtt{w}',\mathtt{t}'\rangle)(g'(x)) = 1$ oraz $P(g'(x))(\langle\mathtt{w}',\mathtt{t}'\rangle) = 1$.
Cóż to jednak oznacza? Po pierwsze, funkcja („tabelka") h_2 przyporządkowuje każdemu „obiektowi funkcyjnemu" (czyli znów pewnej „tabelce") $P\in D_{\langle s,\langle e,t\rangle\rangle}$ wartość 1, gdy w świecie $\langle\mathsf{w}',\mathsf{t}'\rangle$ istnieje jakiś obiekt $g'(x)$ będący jednorożcem, spełniający także warunek $P(\langle\mathsf{w}',\mathsf{t}'\rangle)$. Tak więc h_2 jest funkcją charakterystyczną dla takich P, które (w danym świecie) są spełnione przez $g'(x)$. Z kolei h_1 (będąca drugim argumentem (dopełnieniem) „tabelki" reprezentującej stała seek$'$) jest intensją h_2 (czyli ma tą samą wartość we wszystkich światach), tzn. $h_1(\langle\mathsf{w}',\mathsf{t}'\rangle) = \mathsf{h}_2^{\langle\mathsf{w}',\mathsf{t}'\rangle}$ (gdyż rzecz jasna h_2 definiowana jest dla każdego świata z osobna).

Ostatecznie możemy stwierdzić, że jednorożca (czy inny obiekt poszukiwań, marzeń itp.) identyfikuje (w sposób dość złożony) funkcja h_1, zależna od tego w których światach istnieją jednorożce. Ponadto

jeśli nawet jednorożce i centaury istnieją dokładnie w tych samych światach, to zbiory relacji P^j i P^c „wybierające” obiekty będące jednorżcami bądź centaurami będą różne (ich przecięcie stanowić będą relacje zachodzące przynajmniej dla jednego jednorożca i jednego centaura), a stąd funkcje h_1, h_2 będą także odmienne, co umożliwi odróżnienie poszukiwania jednorożca od poszukiwania centaura. Rzecz jasna w sytuacji, w której obiekt będący jednorożcem w pewnym świecie jest jednocześnie i centaurem w tymże świecie (np. w żadnym świecie nie istnieją ani jedne, ani drugie), to poszukiwanie jednego z nich jest równoznaczne z poszukiwaniem drugiego. Sprowadza się to do dość banalnego stwierdzenia, że żadna technika nie wykaże różnicy między stałymi czy też innymi wyrażeniami, jeśli brak takich różnic w modelu jako takim.
Oczywiście interpretacja predykatu seek' zależy także od pierwszego argumentu, czyli agenta (tak więc John może w danym świecie szukać jednorożców, a Bill nie). Z kolei to, że zdanie *John seeks every unicorn.* interpretowane jest w odrębny sposób wydaje się teraz ewidentne i wynika z faktu, że odpowiednie h_2 jest tutaj funkcją charakterystyczną dla takich P, które akceptują wszystkie jednorożce naraz.

Zdania zanegowane zawierające w podmiocie kwantyfikowaną frazę rzeczownikową mają zaskakującą interpretację w IL.

Przykład *2.16.* Weźmy zdanie *A man desn't walk.*. Ma ono dwa rozkłady w gramatyce PTQ wraz z odpowiadającymi im translacjami na IL.

a man doesn't walk, **12**
- a man, **2**
  - a
  - man
- walk

$$\mathfrak{g}\,(a\ man\ doesn't\ walk) \equiv \neg\mathfrak{g}\,(\mathsf{a\ woman})\left({}^\frown\mathsf{walk}'\right) \equiv \neg\left(\exists x\,[\mathsf{man}'(x)\ \&\ \mathsf{walk}'(x)]\right) \equiv \forall x\,[\mathsf{man}'(x) \to \mathsf{walk}'(x)]$$

a man doesn't walk, **10**, 12
- $\mathtt{he}_{12}$ *doesn't walk,* **12**
  - $\mathtt{he}_{12}$
  - walk
- talk

$$\mathfrak{g}\,(a\ man\ doesn't\ walk) \equiv \mathfrak{g}\,(\mathsf{a\ woman})\left({}^\frown\lambda x_n\,\mathfrak{g}(\mathtt{he}_{12}\ doesn't\ walk)\right) \equiv \lambda P\,\exists x\,[\mathsf{woman}'(x)\ \&\ P\{x\}]\left({}^\frown\lambda x_n\,(\neg\lambda Q\,[Q\{x_n\}])\,(\mathsf{walk}')\right) \equiv \exists x\,[\mathsf{woman}'(x)\ \&\ \lambda x_n\,{}^\frown\neg\mathsf{walk}'\,(x_n)\,\{x\}] \equiv \exists x\,[\mathsf{woman}'(x)\ \&\ \neg\mathsf{walk}'\,(x)]$$

Tylko druga z tych interpretacji wydaje się uzasadniona i zgodna z intuicją. Co więcej, dla zdania *Every man desn't walk.* otrzymujemy dokładnie takie same interpretacje w IL (choć uzyskane z „odwrotnych” rozbiorów). Autorzy sugerują, że także w tym przypadku wiarygodna jest druga interpretacja, co wydaje się wątpliwe, choć samo zdanie brzmi na tyle dziwnie, by trudno było wyrokować cokolwiek w tej kwestii.

Czasownik be

Wyróżnionym przypadkiem czasownika przechodniego jest czasownik be. Jego translacja na IL, która nie jest stałą jak w przypadku pozostałych czasowników, została podana w rozdz. 2.2.2. Poniżej prezentujemy przykłady interpretacji zdań zawierających be.

Przykład *2.17.* Analiza zdania *John is Bill.*[8] przedstawiona jest za pomocą poniższego drzewa.

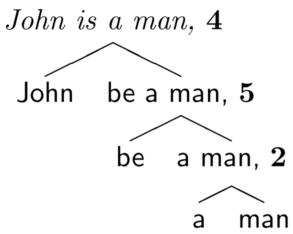
John is Bill, **4**
- John
- be Bill, **5**
  - be
  - Bill

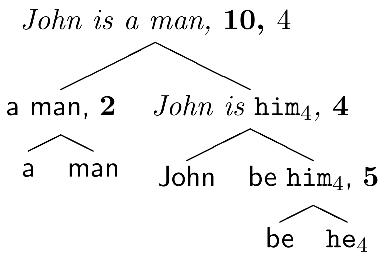
Zdanie to przekładane jest na język IL jako

$$\mathfrak{g}\,(John\ is\ Bill) \equiv \mathfrak{g}\,(\mathsf{John})\,({}^\frown\mathfrak{g}(\mathsf{be\ Bill})) \equiv \lambda P\,[P\{j\}]\left({}^\frown\mathfrak{g}(\mathsf{be})\left({}^\frown\mathfrak{g}(\mathsf{Bill})\right)\right) \equiv {}^\frown\lambda\mathcal{P}\,\lambda x\,\mathcal{P}\{{}^\frown\lambda y\,[x=y]\}\left({}^\frown\lambda Q\,[Q\{m\}]\right)\{j\} \equiv \left({}^\frown\lambda x\left({}^\frown\lambda Q\,[Q\{m\}]\right)\,\{{}^\frown\lambda y\,[x=y]\}\right)\{j\} \equiv {}^\frown\lambda x\left({}^\frown\lambda y\,[x=y]\,\{b\}\right)\{j\} \equiv j=b$$

[8] Choć zdanie to może wydawać się dziwaczne, to już zdanie *Samuel Clemens is Mark Twain.* jest całkiem sensowne.

Przykład *2.18.* Weźmy z kolei zdanie *John is a man.*. Podobnie jak inne zdania tej postaci, posiada ono dwa podstawowe drzewa rozbioru.





Zaczniemy od analizy pierwszego rozbioru.

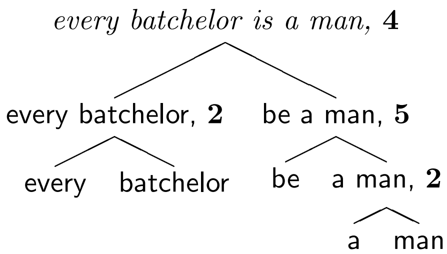
$$
\begin{aligned}
&\mathfrak{g}\ (\textit{John is a man}) \equiv \mathfrak{g}(\mathsf{John})\ ({}^\wedge\mathfrak{g}(\mathsf{be\ a\ man})) \equiv\\
&\lambda P\,[P\{j\}]\left({}^\wedge\mathfrak{g}(\mathsf{be})\left({}^\wedge\mathfrak{g}(\mathsf{a\ man})\right)\right) \equiv\\
&{}^\wedge\lambda\mathcal{P}\,\lambda x\,\mathcal{P}\{{}^\wedge\lambda y\,[x=y]\}\left({}^\wedge\lambda Q\,\exists z\,[\mathsf{man}'(z)\ \&\ Q\{z\}]\right)\{j\} \equiv\\
&{}^\wedge\lambda x\left({}^\wedge\lambda Q\,\exists z\,[\mathsf{man}'(z)\ \&\ Q\{z\}]\{{}^\wedge\lambda y\,[x=y]\}\right)\{j\} \equiv\\
&{}^\wedge\lambda x\,\exists z\left[\mathsf{man}'(z)\ \&\ {}^\wedge\lambda y\,[x=y]\{z\}\right]\{j\} \equiv {}^\wedge\lambda x\,\exists z\,[\mathsf{man}'(z)\ \&\ x=z]\ \{j\} \equiv\\
&\exists z\ [\mathsf{man}'(z)\ \&\ j=z] \equiv \mathsf{man}'(j)
\end{aligned}
$$

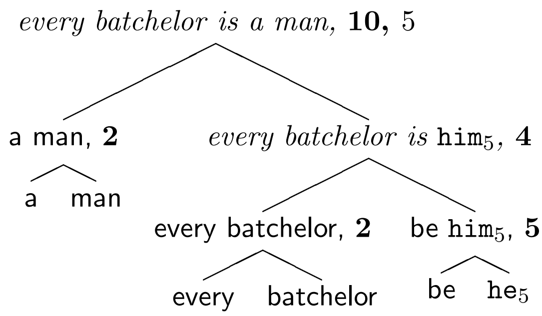
Przejdźmy teraz do drugiego rozbioru.

$$
\begin{aligned}
&\mathfrak{g}\ (\textit{John is a man}) \equiv\\
&\mathfrak{g}\,(\mathsf{a\ man})\,({}^\wedge\mathfrak{g}(\textit{John is } \mathtt{him}_4)) \equiv\\
&\lambda P\,\exists x\,[\mathsf{man}'(x)\ \&\ P\{x\}]\left({}^\wedge\lambda x_4\,\mathfrak{g}(\mathsf{John})\left({}^\wedge\mathfrak{g}(\mathsf{be}\ \mathtt{he}_4)\right)\right) \equiv\\
&\exists x\left[\mathsf{man}'(x)\ \&\ \lambda x_4\,\lambda Q\,[Q\{j\}]\left({}^\wedge\lambda\mathcal{P}\,\lambda z\,\mathcal{P}\{{}^\wedge\lambda y\,[z=y]\}\,({}^\wedge\lambda R\,[R\{x_4\}])\right)(x)\right] \equiv\\
&\exists x\left[\mathsf{man}'(x)\ \&\ \lambda x_4\left({}^\wedge\lambda z\,{}^\wedge\lambda R\,[R\{x_4\}]\,\{{}^\wedge\lambda y\,[z=y]\}\right)\{j\}(x)\right] \equiv\\
&\exists x\left[\mathsf{man}'(x)\ \&\ \lambda x_4\left({}^\wedge\lambda z\,({}^\wedge\lambda y\,[z=y]\{x_4\})\right)\{j\}(x)\right] \equiv\\
&\exists x\left[\mathsf{man}'(x)\ \&\ \lambda x_4\,[j=x_4](x)\right] \equiv\\
&\exists x\,[\mathsf{man}'(x)\ \&\ j=x] \equiv \mathsf{man}'(j)
\end{aligned}
$$

Tak więc translacja czasownika be zachowuje się w taki sposób, że oba powyższe rozbiory mają równoważną interpretację w języku IL. W obu przypadkach ostatnie przekształcenie wynika z zasad logiki I rzędu z równością (identycznością; por. [Grzegorczyk, 1969]).

Przykład *2.19.* Na koniec rozważmy najbardziej skomplikowane zdanie zawierające be, jakie można wyrazić w gramatyce PTQ, tzn. *Every batchelor is a man.*. Także i to zdanie posiada dwa podstawowe rozbiory.





Pierwszy rozbiór przekładamy na język IL w następujący sposób.

$$
\begin{aligned}
&\mathfrak{g}\ (\textit{every batchelor is a man}) && \equiv\\
&\mathfrak{g}\,(\mathsf{every\ batchelor})\left({}^\frown\mathfrak{g}(\mathsf{be\ a\ man})\right) && \equiv\\
&\lambda P\,\forall x\,[\mathsf{batchelor}'(x) \to P\{x\}]\left(\left({}^\frown\lambda\mathcal{P}\,\lambda u\,\mathcal{P}\{{}^\frown\lambda v\,[u=v]\}\right)\left({}^\frown\lambda Q\,\exists y\,[\mathsf{man}'(y)\ \&\ Q\{y\}]\right)\right) && \equiv\\
&\forall x\left[\mathsf{batchelor}'(x)\ \to\ {}^\frown\lambda u\left({}^\frown\lambda Q\,\exists y\,[\mathsf{man}'(y)\ \&\ Q\{y\}]\{{}^\frown\lambda v\,[u=v]\}\right)\{x\}\right] && \equiv\\
&\forall x\left[\mathsf{batchelor}'(x)\ \to\ {}^\frown\lambda u\left(\exists y\,[\mathsf{man}'(y)\ \&\ \lambda v\,[u=v]\,(y)]\right)\{x\}\right] && \equiv\\
&\forall x\left[\mathsf{batchelor}'(x)\ \to\ {}^\frown\lambda u\,(\exists y\,[\mathsf{man}'(y)\ \&\ u=y])\,\{x\}\right] && \equiv\\
&\forall x\left[\mathsf{batchelor}'(x) \to \exists y\,[\mathsf{man}'(y)\ \&\ x=y]\right] && \equiv\\
&\forall x\,[\mathsf{batchelor}'(x)\ \to\ \mathsf{man}'(x)]
\end{aligned}
$$

Następnie zinterpretujemy drugi rozbiór.

$$
\begin{aligned}
&\mathfrak{g}\ (\textit{every batchelor is a man}) && \equiv\\
&\mathfrak{g}\,(\mathsf{a\ man})\,({}^\frown\lambda x_5\,\mathfrak{g}(\textit{every batchelor is}\ \mathtt{him}_5)) && \equiv\\
&\lambda P\,\exists x\,[\mathsf{man}'(x)\ \&\ P\{x\}]\left(\lambda x_5\,{}^\frown\mathfrak{g}(\mathsf{every\ batchelor})\left({}^\frown\mathfrak{g}(\mathsf{be}\ \mathtt{him}_5)\right)\right) && \equiv\\
&\exists x\left[\mathsf{man}'(x)\ \&\left({}^\frown\lambda x_5\,\lambda Q\,\forall y\ [\mathsf{batchelor}'(y) \to Q\{y\}]\left({}^\frown\lambda u\,[u=x_5]\right)\right)\{x\}\right] && \equiv\\
&\exists x\left[\mathsf{man}'(x)\ \&\ \lambda x_5\,\forall y\,\left[\mathsf{batchelor}'(y) \to {}^\frown\lambda u\,[u=x_5]\{y\}\right](x)\right] && \equiv\\
&\exists x\left[\mathsf{man}'(x)\ \&\ \lambda x_5\,\forall y\ [\mathsf{batchelor}'(y) \to y=x_5]\,(x)\right] && \equiv\\
&\exists x\left[\mathsf{man}'(x)\ \&\ \forall y\ [\mathsf{batchelor}'(y) \to y=x]\right]
\end{aligned}
$$

Powyższe dwie interpretacje są ewidentnie różne, co jest zgodne z wszelkimi zasadami traktowania niejednoznaczości zakresu kwantyfikatorów. Nie mniej jednak, o ile pierwsza translacja jest w pełni zgodna z naszymi oczekiwaniami i intuicją, druga nie tylko wydaje się dziwaczna i nieintuicyjna, ona jest wręcz sprzeczna z naszą wiedzą o świecie i języku.

Jak zauważają autorzy, definicja funkcji $\mathfrak{g}$ umożliwia analogiczną interpretację także takich zdań jak *John is every man.*, *A man is John.*, *John is no man.* itp., które brzmią dość dziwnie (por. [Bach, 1968]).

Reguła „orzeczenie-dopełnienie bezokolicznikowe”

W przeciwieństwie do tradycyjnego transformacyjnego podejścia do dopełnienia bezokolicznikowego, zaproponowanego przez Rosenbauma (1967), w gramatyce PTQ nie konstruuje się zdań typu *John tries to walk.* ze zdania *John walks.* modyfikowanego czasownikiem try, tylko traktuje się frazę *try to walk* jako całość. Tak więc reguła ta łączy czasownik kategorii IV/IV ze „zwykłą” frazą czasownikową kategorii IV dając w rezultacie zmodyfikowaną frazę czasownikową.

S8 Jeśli $\delta \in P_{IV/IV}$ oraz $\beta \in P_{IV}$, to $F_{17}(\delta,\beta) = \delta\ \mathsf{to}\ \beta \in P_{IV}$.

T8 Jeśli $\delta \in P_{IV/IV}$ oraz $\beta \in P_{IV}$, to $\mathfrak{g}(F_{17}(\delta,\beta)) = \mathfrak{g}(\delta)({}^\frown\mathfrak{g}(\beta))$.

Przykład *2.20.* Zdanie *John tries to walk.* posiada prostą interpretację (podobną do interpretacji zdania *John walks slowly.*, co o tyle nie jest zaskakujące, że składające się na nie wyrażenia elementarne są analogicznych typów).

John tries to walk, **4**

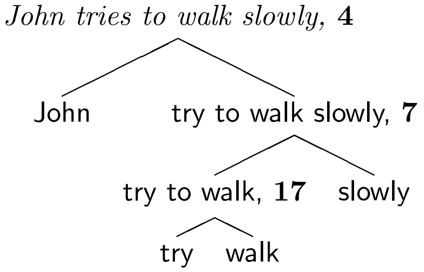
John   try to walk, **17**

try   walk

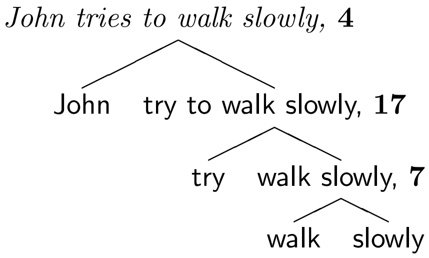
$$
\begin{aligned}
&\mathfrak{g}\ (\textit{John tries walks}) &&\equiv\ \mathfrak{g}\,(\mathsf{John})\,({}^\frown\mathfrak{g}(\mathsf{try\ to\ walk})) &&\equiv\\
&\lambda P\,[P\{j\}]\left({}^\frown\mathfrak{g}(\mathsf{try})\,\mathfrak{g}(\mathsf{walk})\right) &&\equiv\ {}^\frown\mathsf{try}'\,({}^\frown\mathsf{walk}')\,\{j\} &&\equiv\\
&\mathsf{try}'\,({}^\frown\mathsf{walk}')\,(j)
\end{aligned}
$$

Rzecz jasna, podobnie jak przysłowki, także czasowniki o dopełnieniu bezokolicznikowym mogą być łączone ze złożonymi frazami czasownikowymi. I tak zdanie *John tries to walk or talk.* będzie miało dwa rozbiory, które przekładane są na IL jako $\mathsf{try}'({}^\frown\mathsf{walk}')(j) \vee \mathsf{talk}'(j)$ lub $\mathsf{try}'({}^\frown[\mathsf{walk}'(j) \vee \mathsf{talk}'(j)])$ (por. przykład 2.6). Poniżej rozważymy jednak jeszcze jeden przykład.

<u>Przykład</u> *2.21.* Zdanie *John tries to walk slowly.* posiada dwa rozbiory przedstawione poniżej. W obu przypadkach translacja polega na prostym składaniu funkcji $\mathfrak{g}$.



$$\mathfrak{g}\ (\textit{John tries to walk slowly}) \equiv$$
$$\lambda P\,[P\{j\}]\left({}^\frown\mathfrak{g}(\mathsf{slowly})\left({}^\frown\mathfrak{g}(\mathsf{try})\left({}^\frown\mathfrak{g}(\mathsf{walk})\right)\right)\right) \equiv$$
$$\mathsf{slowly}\,({}^\frown\mathsf{try}'\,({}^\frown\mathsf{walk}'))\,(j)$$



$$\mathfrak{g}\ (\textit{John tries to walk slowly}) \equiv$$
$$\lambda P\,[P\{j\}]\left({}^\frown\mathfrak{g}(\mathsf{try})\left({}^\frown\mathfrak{g}(\mathsf{slowly})\left({}^\frown\mathfrak{g}(\mathsf{walk})\right)\right)\right) \equiv$$
$$\mathsf{try}\,({}^\frown\mathsf{slowly}'\,({}^\frown\mathsf{walk}'))\,(j)$$

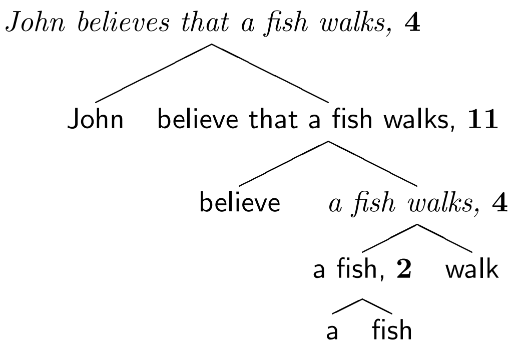
<u>Reguła „orzeczenie-dopełnienie zdaniowe”</u>
Reguła ta łączy czasowniki o dopełnieniu zdaniowym kategorii IV/t (jak *believe*) ze zdaniem (kategorii t) za pomocą spójnika that, tworząc frazę czasownikową (kategorii IV).

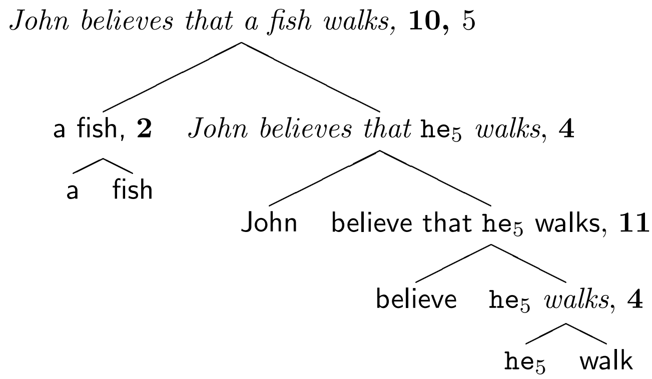
S7 Jeśli $\alpha \in P_{IV/t}$ oraz $\phi \in P_t$, to $F_{11}(\alpha, \phi) = \alpha\ \mathsf{that}\ \phi \in P_{IV}$.

Zauważmy, że wyraz that nie należy do zbioru wyrażeń podstawowych B, podobnie jak $\mathsf{such\ that}$ w regule S3 czy and i or w regułach S11, S12, S13. Z powyższą regułą związana jest następująca reguła translacji:

T7 Jeśli $\alpha \in P_{IV/t}$ oraz $\phi \in P_t$, to $\mathfrak{g}(F_{11}(\alpha, \phi)) = \mathfrak{g}(\alpha)({}^\frown\mathfrak{g}(\phi))$.

<u>Przykład</u> *2.22.* Zdanie *John believes that a fish walks.* posiada dwa drzewa analizy. Pierwsze z nich (po lewej) odpowiada odczytaniu tego zdania *de dicto* (tzn. John wierzy, że istnieje jakaś nieokreślona, chodząca ryba). Drugie drzewo analizy (po prawej) odpowiada odczytaniu rozważanego zdania *de re* (tzn. istnieje jakaś konkretna ryba, którą John uważa za chodzącą). Powstaje ono dzięki zastosowaniu w analizie reguły **S14**.





Pierwszej analizie powyższego zdania odpowiada następujący przekład na język IL:

$$\begin{array}{llll}
\mathfrak{g}\ (\textit{John believes that a fish walks}) & \equiv & \mathfrak{g}(\mathsf{John})\ ({}^\frown\mathfrak{g}(\mathsf{believe\ that\ a\ fish\ walks})) & \equiv \\
\lambda P\,[P\{j\}]\left({}^\frown\mathfrak{g}(\mathsf{believe})\left[{}^\frown\mathfrak{g}(\mathsf{a\ fish\ walks})\right]\right) & \equiv & {}^\frown\mathsf{believe}'\left({}^\frown\mathfrak{g}(\mathsf{a\ fish})\ \left({}^\frown\mathfrak{g}(\mathsf{walk})\right)\right)\{j\} & \equiv \\
\mathsf{believe}'\left({}^\frown\lambda Q\,\exists x\,[\mathsf{fish}'(x)\ \&\ Q\{x\}]\left({}^\frown\mathsf{walk}'\right)\right)(j) & \equiv & \mathsf{believe}'\left({}^\frown\exists x\ [\mathsf{fish}'(x)\ \&\ \mathsf{walk}'(x)]\right)(j) & \equiv \\
\mathsf{believe}'\left(j, {}^\frown\exists x\ [\mathsf{fish}'(x)\ \&\ \mathsf{walk}'(x)]\right) & & &
\end{array}$$

Drugiej analizie omawianego zdania odpowiada następujący przekład na język IL:

$$\begin{array}{ll}
\mathfrak{g}\ (\textit{John believes that a fish walks}) & \equiv \\
\mathfrak{g}\ (\mathsf{a\ fish})\ ({}^\frown\lambda x_5\,\mathfrak{g}(\mathsf{John\ believes\ that\ \mathtt{he}_5\ walks})) & \equiv \\
\lambda Q\,\exists x\,[\mathsf{fish}'(x)\ \&\ Q\{x\}]\left({}^\frown\lambda x_5\,\mathfrak{g}(\mathsf{John})\left[{}^\frown\mathfrak{g}(\mathsf{believe\ that\ \mathtt{he}_5\ walks})\right]\right) & \equiv \\
\exists x\left[\mathsf{fish}'(x)\ \&\ \lambda x_5\left({}^\frown\lambda P\,[P\{j\}]\left[{}^\frown\mathfrak{g}(\mathsf{believe})\left({}^\frown\mathfrak{g}(\mathtt{he}_5\ \mathsf{walks})\right)\right]\right)\{x\}\right] & \equiv \\
\exists x\left[\mathsf{fish}'(x)\ \&\ \lambda x_5\left({}^\frown\mathsf{believe}'\left({}^\frown\mathfrak{g}(\mathtt{he}_5)\ ({}^\frown\mathfrak{g}\mathsf{walk})\right)\{j\}\right)(x)\right] & \equiv \\
\exists x\left[\mathsf{fish}'(x)\ \&\ \lambda x_5\left(\mathsf{believe}'\ \left({}^\frown\lambda P\,[P\{x_5\}]({}^\frown\mathsf{walk}')\right)(j)\right)(x)\right] & \equiv \\
\exists x\left[\mathsf{fish}'(x)\ \&\ \lambda x_5\left(\mathsf{believe}'\ \left({}^\frown{}^\frown\mathsf{walk}'\{x_5\}\right)(j)\right)(x)\right] & \equiv \\
\exists x\left[\mathsf{fish}'(x)\ \&\ \mathsf{believe}'\left({}^\frown\mathsf{walk}'(x)\right)(j)\right] & \equiv \\
\exists x\left[\mathsf{fish}'(x)\ \&\ \mathsf{believe}'\ (j, {}^\frown\mathsf{walk}'(x))\right] &
\end{array}$$

W rozważanych do tej pory przykładach zdań zadowolaliśmy się ich translacją na język IL za pomocą funkcji $\mathfrak{g}$; nie było potrzeby interpretacji uzyskanych wyrażeń w modelu, gdyż były to standardowe formuły I rzędu. Tym razem jednak rzecz ma się inaczej. Dlatego pozwolimy sobie na przeanalizowanie interpretacji uzyskanych formuł. Dla zwiększenia czytelności, poniżej w tym przykładzie przyjmiemy konwencję notacyjną, że $[F(\mathsf{believe}')](\langle \mathtt{w},\mathtt{t}\rangle) = \mathfrak{Bel}^{\langle \mathtt{w},\mathtt{t}\rangle}$, $[F(j)](\langle \mathtt{w},\mathtt{t}\rangle) = \mathfrak{J}^{\langle \mathtt{w},\mathtt{t}\rangle}$, $[F(\mathsf{fish}')](\langle \mathtt{w},\mathtt{t}\rangle) = \mathfrak{f}^{\langle \mathtt{w},\mathtt{t}\rangle}$, $[F(\mathsf{walk}')](\langle \mathtt{w},\mathtt{t}\rangle) = \mathfrak{w}^{\langle \mathtt{w},\mathtt{t}\rangle}$.

$\left[\!\left[\mathsf{believe}'\left(j, {}^\frown\exists x\ [\mathsf{fish}'(x)\ \&\ \mathsf{walk}'(x)]\right)\right]\!\right]^{\mathcal{M},\mathtt{w},\mathtt{t},g} = [F(\mathsf{believe}')](\langle \mathtt{w},\mathtt{t}\rangle)\left([\mathtt{F(j)}](\langle \mathtt{w},\mathtt{t}\rangle), h_1\right) =$ $\mathfrak{Bel}^{\langle w,t\rangle}\left(\mathfrak{J}^{\langle w,t\rangle}, h_1\right)$, gdzie $h_1\colon W\times T \longrightarrow \{0,1\}$ jest taką funkcją, że $h_1(\langle \mathtt{w}',\mathtt{t}'\rangle) = [\![\exists \mathtt{x}\ (\mathsf{fish}'(\mathtt{x})\ \&\ \mathsf{walk}'(\mathtt{x}))]\!]^{\mathcal{M},\mathtt{w}',\mathtt{t}',\mathtt{g}} = 1$ wtw, gdy istnieje $g' \in \mathcal{G}^g_x$ takie, że $[F(\mathsf{fish}')](\langle \mathtt{w}',\mathtt{t}'\rangle)(\mathtt{g}'(\mathtt{x})) = 1$ oraz $[F(\mathsf{walk}')](\langle \mathtt{w}',\mathtt{t}'\rangle)(\mathtt{g}'(\mathtt{x})) = 1$ wtw, gdy istnieje $g' \in \mathcal{G}^g_x$ takie, że $\mathfrak{f}^{\langle w',t'\rangle}(g'(x)) = 1$ oraz $\mathfrak{w}^{\langle w',t'\rangle}(g'(x)) = 1$.

Oznacza to, że fakt, czy John wierzy w istnienie *de dicto* ryby umiejącej chodzić określa wartość pewnej zależności $\mathfrak{Bel}$ (o wartościach prawda/fałsz) zdefiniowanej w bieżącym świecie, której argumentami są obiekt reprezentujący Johna oraz intensja formuły $\exists x\ [\mathsf{fish}'(x)\ \&\ \mathsf{walk}'(x)]$ (stwierdzająca istnienie (niezależnie w każdym świecie) obiektu będącego rybą, która chodzi).

$\left[\!\left[\exists x\left(\mathsf{fish}'(x)\ \&\ \mathsf{believe}'\ [j, {}^\frown\mathsf{walk}'(x)]\right)\right]\!\right]^{\mathcal{M},\mathtt{w},\mathtt{t},g} = 1$ wtw, gdy istnieje $g' \in \mathcal{G}^g_x$, że $[F(\mathsf{fish}')](\langle \mathtt{w},\mathtt{t}\rangle)(\mathtt{g}'(\mathtt{x})) = 1$ oraz $[F(\mathsf{believe}')](\langle \mathtt{w},\mathtt{t}\rangle)\left([\mathtt{F(j)}](\langle \mathtt{w},\mathtt{t}\rangle), h_2\right) = 1$, istnieje $g' \in \mathcal{G}^g_x$ takie, że $\mathfrak{f}^{\langle w,t\rangle}(g'(x)) = 1$ oraz $\mathfrak{Bel}^{\langle w,t\rangle}\left(\mathfrak{J}^{\langle w,t\rangle}, h_2\right)$, gdzie $h_2\colon W\times T \longrightarrow \{0,1\}$ jest taką funkcją, że $h_1(\langle \mathtt{w}',\mathtt{t}'\rangle) = [\![\mathsf{walk}'(\mathtt{x})]\!]^{\mathcal{M},\mathtt{w}',\mathtt{t}',\mathtt{g}'} = [\mathtt{F}(\mathsf{walk}')](\langle \mathtt{w}',\mathtt{t}'\rangle)(\mathtt{g}'(\mathtt{x})) = \mathfrak{w}^{\langle \mathtt{w}',\mathtt{t}'\rangle}(\mathtt{g}'(\mathtt{x}))$.

Oznacza to istnienie (*de re*) pewnego obiektu będącego rybą w bieżącym świecie, co do której wiara Johna w jej umiejętność chodzenia określana jest przez wartość pewnej funkcji $\mathfrak{Bel}$ zdefiniowanej w bieżącym świecie, której argumentami są obiekt reprezentujący Johna oraz intensja prostszej formuły $\mathsf{walk}'(x)$ (stwierdzająca, czy obiekt będący rybą w bieżącym świecie chodzi (bądź nie) w dowolnym świecie). Zauważmy, że $g'(x)$ musi być rybą jedynie w bieżącym świecie, w pozostałych może być czymkolwiek, byle to coś chodziło.

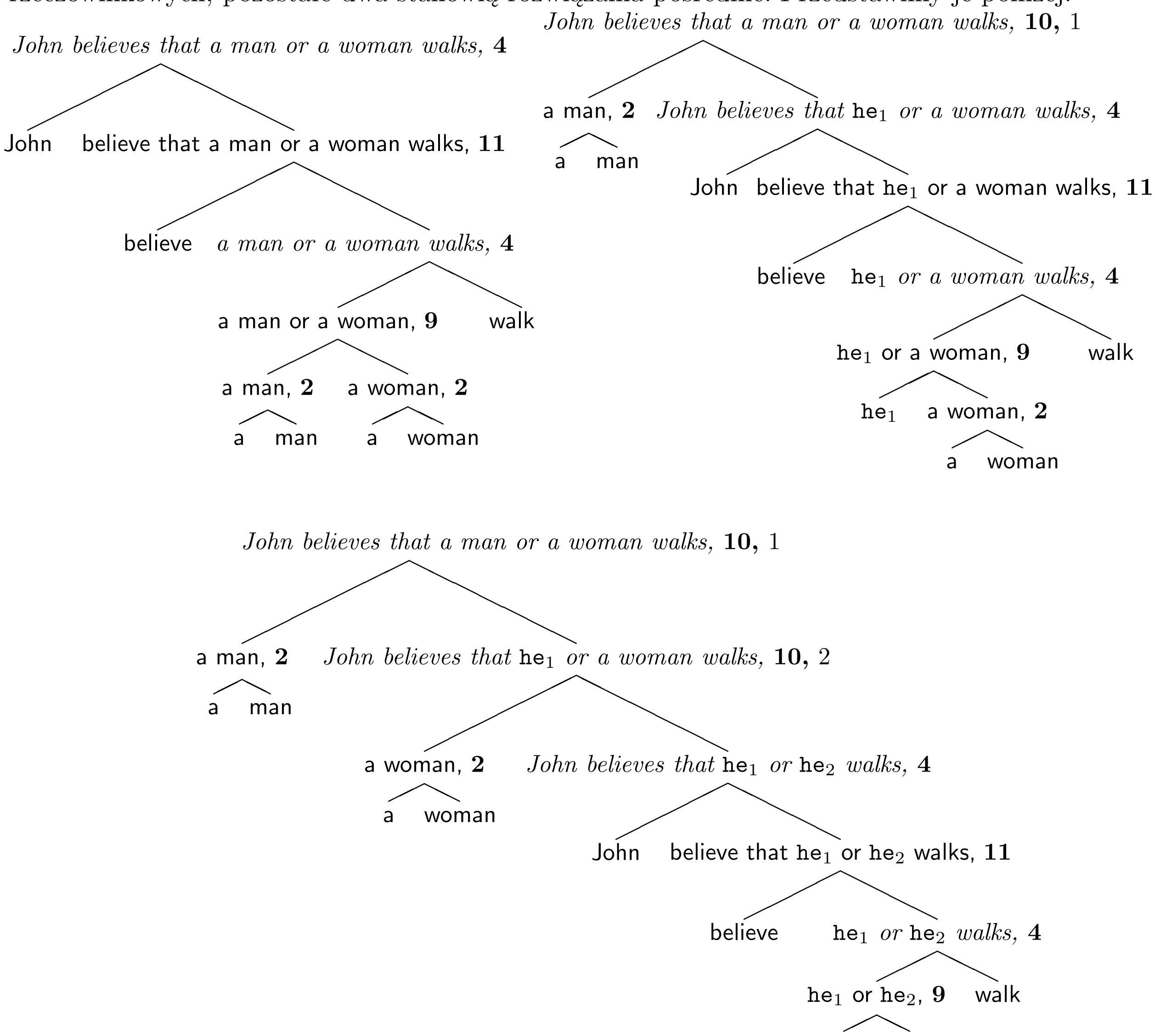
W przypadku bardziej skomplikowanych zdań liczba analiz może być znacząco większa. Na przykład zdanie *Every man believes that a fish walks* posiada siedem drzew analizy, którym odpowiadają następujące trzy translacje:

$\exists x \left[\mathsf{fish'}(x)\ \&\ \forall y\ [\mathsf{man}'(y) \to \mathsf{believe}'(y, {}^\frown(\mathsf{walk}'(x)))]\right]$,

$\forall y \left[\mathsf{man'}(y) \ \to\ \exists x\ [\mathsf{fish}'(x)\,\&\,\mathsf{believe}'(y, {}^\frown(\mathsf{walk}'(x)))]\right]$,

$\forall y \left[\mathsf{man'}(y) \ \to\ \mathsf{believe}'(y, {}^\frown(\exists x\ [\mathsf{fish}'(x)\,\&\,\mathsf{walk}'(x)]))\right]$. Rozważmy jeszcze jedno do'ść skomplikowane stwirdzenie.

Przykład *2.23.* Zdanie *John believes that a man or a woman walks.* posiada cztery drzewa rozbioru. Pierwsze z nich odpowiada odczytaniu tego zdania *de dicto*, a czwarte *de re*[9] względem obu fraz rzeczownikowych, pozostałe dwa stanowią rozwiązania pośrednie. Przedstawimy je poniżej.



[9]Tak naprawdę istnieją dwa równoważne rozbiory *de re* w zależności od kolejności rozpatrywania fraz rzeczownikowych.

Przedstawimy teraz translacje tych rozbiorów na język IL w tej samej kolejności.

$$
\begin{array}{ll}
\mathfrak{g}\ (\textit{John believes that a man or a woman walks}) & \equiv \\
\mathfrak{g}\,(\mathsf{John})\ ({}^\frown\mathfrak{g}\,(\mathsf{believe\ that\ a\ man\ or\ a\ woman\ walk})) & \equiv \\
\lambda P\,[P\{j\}]\ \Big({}^\frown\mathfrak{g}\,(\mathsf{believe})\ \big[{}^\frown\mathfrak{g}\,(\mathsf{a\ man\ or\ a\ woman\ walk})\big]\Big) & \equiv \\
{}^\frown\mathsf{believe}'\ \Big({}^\frown\mathfrak{g}\,(\mathsf{a\ man\ or\ a\ woman})\ \big({}^\frown\mathfrak{g}\,(\mathsf{walk})\big)\Big)\ \{j\} & \equiv \\
\mathsf{believe}'\ \big({}^\frown(\lambda R\,[\mathfrak{g}\,(\mathsf{a\ man})\,(R)] \vee \lambda R\,[\mathfrak{g}\,(\mathsf{a\ woman})\,(R)])\ (\mathsf{walk}')\big)\ (j) & \equiv \\
\mathsf{believe}'\ \Big(j, {}^\frown(\lambda P\,\exists x\,[\mathsf{man}'(x)\ \&\ P\{x\}]({}^\frown\mathsf{walk}') \vee \lambda Q\,\exists y\,[\mathsf{woman}'(y)\ \&\ Q\{y\}]\,({}^\frown\mathsf{walk}'))\Big) & \equiv \\
\mathsf{believe}'\ \big(j, {}^\frown\exists x\,[\mathsf{man}'(x)\ \&\ \mathsf{walk}'(x)] \vee \exists y\,[\mathsf{woman}'(y)\ \&\ \mathsf{walk}'(y)]\big) &
\end{array}
$$

$$
\begin{array}{ll}
\mathfrak{g}\ (\textit{John believes that a man or a woman walks}) & \equiv \\
\mathfrak{g}(\mathsf{a\ man})\ \big({}^\frown\lambda x_1\,\mathfrak{g}(\textit{John believes that } \mathtt{he}_1 \textit{ or a woman walks})\big) & \equiv \\
\lambda P\,\exists x\,[\mathsf{man}'(x)\ \&\ P\{x\}]\ \Big({}^\frown\lambda x_1\,\mathfrak{g}(\mathsf{John})\ \big({}^\frown\mathfrak{g}(\mathsf{believe\ that\ }\mathtt{he}_1\mathsf{\ or\ a\ woman\ walk})\big)\Big) & \equiv \\
\exists x\,\Big[\mathsf{man}'(x)\ \&\ \lambda x_1\,\lambda Q\,[Q\{j\}]\ \Big({}^\frown\mathsf{believe}'\ \big({}^\frown\mathfrak{g}(\mathtt{he}_1\mathsf{\ or\ a\ woman})\,({}^\frown\mathsf{walk}')\big)\Big)\ (x)\Big] & \equiv \\
\exists x\,\Big[\mathsf{man}'(x)\ \&\ \lambda x_1\ \Big(\mathsf{believe}'\ \big(j, {}^\frown(\lambda R\,[\mathfrak{g}(\mathtt{he}_1)\,(R)] \vee \lambda R\,[\mathfrak{g}(\mathsf{a\ woman})\,(R)])\,({}^\frown\mathsf{walk}')\big)\Big)\ (x)\Big] & \equiv \\
\exists x\,\Big[\mathsf{man}'(x)\ \&\ \lambda x_1\ \Big(\mathsf{believe}'\ \big(j, {}^\frown(\mathsf{walk}'\,(x_1) \vee \exists y\,[\mathsf{woman}'(y)\ \&\ \mathsf{walk}'(y)])\big)\Big)\ (x)\Big] & \equiv \\
\exists x\,\Big[\mathsf{man}'(x)\ \&\ \mathsf{believe}'\ \big(j, {}^\frown(\mathsf{walk}'\,(x_1) \vee \exists y\,[\mathsf{woman}'(y)\ \&\ \mathsf{walk}'(y)])\big)\Big] &
\end{array}
$$

$$
\begin{array}{ll}
\mathfrak{g}\ (\textit{John believes that a man or a woman walks}) & \equiv \\
\mathfrak{g}(\mathsf{a\ man})\ \big({}^\frown\lambda x_1\,\mathfrak{g}(\textit{John believes that } \mathtt{he}_1 \textit{ or a woman walks})\big) & \equiv \\
\lambda P\,\exists x\,[\mathsf{man}'(x)\ \&\ P\{x\}]\ \Big({}^\frown\lambda x_1\,\mathfrak{g}(\mathsf{a\ woman})\,{}^\frown\lambda x_2\,\mathfrak{g}(\textit{John believes that } \mathtt{he}_1 \textit{ or } \mathtt{he}_2 \textit{ walks})\Big) & \equiv \\
\exists x\ \Big[\mathsf{man}'(x)\ \&\ \lambda x_1\,\lambda Q\,\exists y\ [\mathsf{woman}'(y)\ \&\ Q\{y\}]\ {}^\frown\lambda x_2\,\mathsf{believe}'\ \big(j, {}^\frown(\mathsf{walk}'\,(x_1) \vee \mathsf{walk}'\,(x_2))\big)\ (x)\Big] & \equiv \\
\exists x\ \Big[\mathsf{man}'(x)\ \&\ \lambda x_1\,\exists y\ \Big[\mathsf{woman}'(y)\ \&\ \lambda x_2\,\mathsf{believe}'\ \big(j, {}^\frown(\mathsf{walk}'\,(x_1) \vee \mathsf{walk}'\,(x_2))\big)\ (y)\Big]\ (x)\Big] & \equiv \\
\exists x\ \Big[\mathsf{man}'(x)\ \&\ \exists y\ \Big[\mathsf{woman}'(y)\ \&\ \mathsf{believe}'\ \big(j, {}^\frown(\mathsf{walk}'\,(x) \vee \mathsf{walk}'\,(y))\big)\Big]\Big] & \equiv \\
\exists x\,\exists y\ \Big[\mathsf{man}'(x)\ \&\ \mathsf{woman}'(y)\ \&\ \mathsf{believe}'\ \big(j, {}^\frown(\mathsf{walk}'\,(x) \vee \mathsf{walk}'\,(y))\big)\Big] &
\end{array}
$$

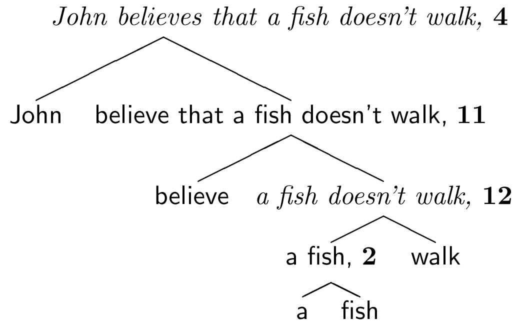
We wszystkich przypadkach czasownik believe jest interpretowany (w każdym świecie niezależnie) jako pewna funkcja $\mathfrak{Bel}^{\langle w,t\rangle}(\mathfrak{ob}, h)$, (a więc „tabelka") o wartościach pradwa/fałsz, gdzie $\mathfrak{ob}$ jest wybranym obiektem (agentem), który w coś wierzy, zaś funkcja $h\colon W \times T \longrightarrow \{0,1\}$ jest intensją przedmiotu jego wiary φ. Otóż pamiętajmy, że intensja dowolnej formuły jest niezmienna w modelu, czyli ma tą samą wartość w każdym świecie. Jej zastosowanie umożliwia jedynie korzystanie z tej wiedzy w różny sposób, w zależności od agenta i świata. Innymi słowy, funkcja $\mathfrak{Bel}^{\langle w,t\rangle}$ określa jedynie, w których światach „oceniana" formuła φ ma być spełniona, by $\mathfrak{ob}$ w nią uwierzył. Niestety, takiej „oceny" można dokonać w sposób dowolny. Dopuszczalne jest nawet, by agent wierzył w formułę, która jest fałszywa we wszystkich światach!

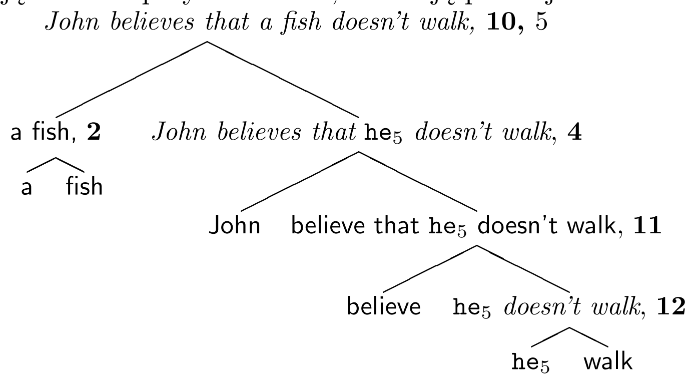
Ależ to normalne! — zareaguje czytelnik na takie banalne stwierdzenie. Do wykluczenia „absurdalnych" interpretacji służą aksjomaty! No tak, tylko język logiki IL daje dość ograniczone możliwości ich

formułowania. Możemy oczywiście wykluczyć wiarę w formuły „koniecznie fałszywe", których intesje składają się z samych fałszów, poprzez aksjomat: $\mathsf{Bel}(c, {}^\frown\varphi) \rightarrow \Diamond\varphi$.
Jednak już zażądanie, by formuła taka zachodziła na dowolnej osi w bieżącym momencie nie daje się w IL sformułować.[10] Jedynym innym dostępnym ograniczeniem jest aksjomat $\mathsf{Bel}(c, {}^\frown\varphi) \rightarrow \Box\varphi$ banalizujący zależność Bel, uniezależniając ją zarówno od podmiotu wierzącego, jak i od świata, w którym zależność ta ma zachodzić. Istnieje jednak podstawowa własność predykatów (operatorów) typu believe, którą daje się sformułować w omawianym języku. Mówi ona, że każdy wierzy, że wierzy w to, w co wierzy: $\mathsf{Bel}(c, {}^\frown\varphi) \rightarrow \mathsf{Bel}\left(c, {}^\frown\mathsf{Bel}(c, {}^\frown\varphi)\right)$.

Oczywiście, również orzeczenie z dopełnieniem zdaniowym można zanegować.

Przykład *2.24.* Reprezentacja zdania *John doesn't believe that a fish walks.* w PTQ i IL jest banalna. Rozpatrzmy odrobinę bardziej skomplikowane zdanie *John believes that a fish doesn't walk.*. Dwa drzewa analizy tego zdania, będące prostą modyfikacją drzew z przykładu 2.22, widnieją poniżej.





Omawiane zdanie posiada dwie następujące translacje na język IL (bel. jest skrótem od believe):

$$\begin{array}{llll}
\mathfrak{g}\ (\textit{John bel.s that a fish doesn't walk}) & \equiv & \mathfrak{g}\,(\mathsf{John})\ ({}^\frown\mathfrak{g}(\mathsf{bel.\ that\ a\ fish\ doesn't\ walk})) & \equiv \\
\lambda P\,[P\{j\}]\left({}^\frown\mathfrak{g}(\mathsf{bel.})\left[{}^\frown\mathfrak{g}(\mathsf{a\ fish\ doesn't\ walk})\right]\right) & \equiv & {}^\frown\mathsf{bel.}'\left({}^\frown\neg\mathfrak{g}(\mathsf{a\ fish})\left({}^\frown\mathfrak{g}(\mathsf{walk})\right)\right)\{j\} & \equiv \\
\mathsf{bel.}'\left({}^\frown\neg\lambda Q\,\exists x\ [\mathsf{fish}'(x)\ \&\ Q\{x\}]\left({}^\frown\mathsf{walk}'\right)\right)(j) & \equiv & \mathsf{bel.}'\left({}^\frown\neg\exists x\ [\mathsf{fish}'(x)\ \&\ \mathsf{walk}'(x)]\right)(j) & \equiv \\
\mathsf{bel.}'\left(j, {}^\frown\neg\exists x\ [\mathsf{fish}'(x)\ \&\ \mathsf{walk}'(x)]\right) & \equiv & \mathsf{bel.}'\left(j, {}^\frown\forall x\,[\mathsf{fish}'(x) \rightarrow \neg\mathsf{walk}'(x)]\right) &
\end{array}$$

$$\begin{array}{ll}
\mathfrak{g}\ (\textit{John believes that a fish doesn't walk}) & \equiv \\
\mathfrak{g}\,(\mathsf{a\ fish})\,({}^\frown\lambda x_5\,\mathfrak{g}(\textit{John believes that } \mathtt{he}_5 \textit{ doesn't walk})) & \equiv \\
\lambda Q\,\exists x\ [\mathsf{fish}'(x)\ \&\ Q\{x\}]\left({}^\frown\lambda x_5\,\mathfrak{g}(\mathsf{John})\left[{}^\frown\mathfrak{g}(\mathsf{believe\ that\ \mathtt{he}_5\ doesn't\ walk})\right]\right) & \equiv \\
\exists x\left[\mathsf{fish'}(x)\ \&\ \lambda x_5\left({}^\frown\lambda P\,[P\{j\}]\left[{}^\frown\mathfrak{g}(\mathsf{believe})\left({}^\frown\mathfrak{g}(\mathtt{he}_5\ \mathsf{doesn't\ walk})\right)\right]\right)\{x\}\right] & \equiv \\
\exists x\left[\mathsf{fish'}(x)\ \&\ \lambda x_5\left(\mathsf{believe'}\ \left({}^\frown\neg\lambda P\,[P\{x_5\}]({}^\frown\mathsf{walk}')\right)(j)\right)(x)\right] & \equiv \\
\exists x\left[\mathsf{fish'}(x)\ \&\ \lambda x_5\left(\mathsf{believe'}\ \left(j, {}^\frown\neg{}^\frown\mathsf{walk}'\{x_5\}\right)\right)(x)\right] & \equiv \\
\exists x\left[\mathsf{fish'}(x)\ \&\ \mathsf{believe}'\ (j, {}^\frown\neg\mathsf{walk}'(x))\right] &
\end{array}$$

Zaimki na pozycji dopełnienia

Reguła wiązania zaimków S14 przedstawiona na str. 17 operuje na zdaniach. Oznacza to, że umożliwia nam ona nie tylko reprezentację zdań z przykładów 2.11, 2.26, czy odczytanie *de re* zdań z przykładów

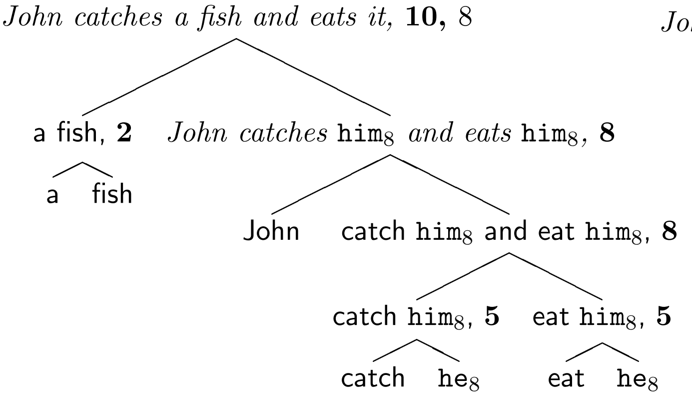
[10]Swoją drogą trudno pojąć, dlaczego operator $\Box$ nie został zdefiniowany w bardziej ograniczony sposób jako:
$[\![\Box\phi]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g} = 1$ wtw, gdy $[\![\phi]\!]^{\mathcal{M},\mathtt{w}',\mathtt{t},g} = 1$ dla dowolnych $\mathtt{w}' \in W$ oraz dowolnego $\phi \in \mathcal{E}_t$.
Wówczas „totalną konieczność" możnaby zdefiniować jako $\mathbf{H}\ \Box\varphi \vee \Box\varphi \vee \mathbf{G}\ \Box\varphi$, a i wspomniane powyżej ograniczenie dałoby się sformułować.

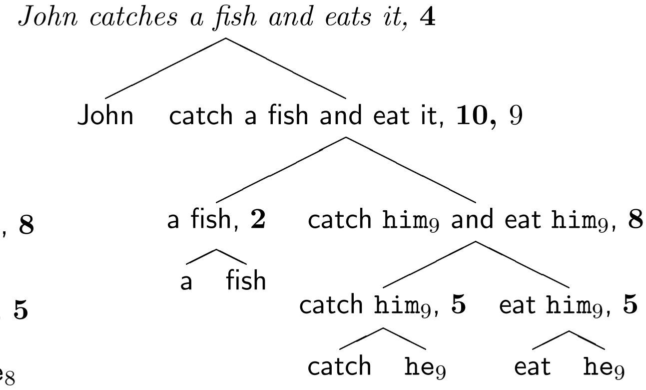
2.22, 2.14, 2.15, ale także i interpretację *de re* zdań o złożonym dopełnieniu, typu *John catches a fish and eats it.* determinującą, że istnieje konkretna ryba, którą John łowi i zjada. Jeśli chcielibyśmy jednak zinterpretować to zdanie w taki sposób, że John łapie dowolną, nieokreśloną rybę, a następnie zjada tę właśnie rybę (która to interpretacja wydaje wręcz bardziej naturalna), rozważana reguła nam tego nie umożliwia. Potrzebna jest reguła formująca frazę czasownikową (kategorii IV; *catch a fish and eat it*) z frazy rzeczownikowej (kategorii T; *a fish*) oraz frazy czaswonikowej (kategorii IV; *catch* $\texttt{him}_n$ *and eat* $\texttt{him}_n$).

S16 Jeśli $\alpha \in P_T$ oraz $\delta \in P_{IV}$, to $F_{10,n}(\alpha, \delta) \in P_{IV}$.

T16 Jeśli $\alpha \in P_T$ oraz $\delta \in P_{IV}$, to $\mathfrak{g}(F_{10,n}(\alpha, \phi)) = \lambda y\, \mathfrak{g}(\alpha) \left({}^\frown\lambda x_n\, \mathfrak{g}(\phi)] \right)$.

<u>Przykład</u> *2.25.* Zdanie *John catches a fish and eats it.* posiada dwa drzewa analizy widoczne poniżej.





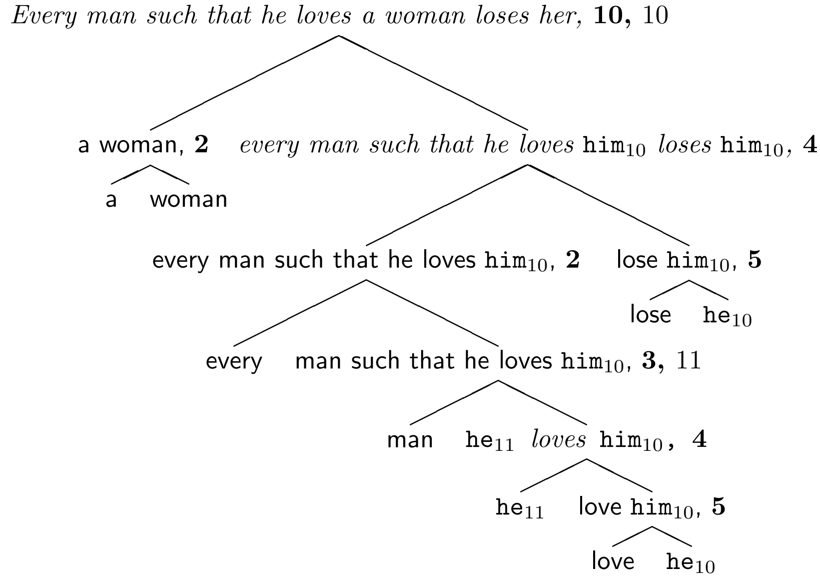
Odpowiadają im dwie równoważne translacje.

$$
\begin{aligned}
&\mathfrak{g}\ (\textit{John catches a fish and eats it}) && \equiv\\
&\mathfrak{g}\,(\mathsf{a\ fish}) \left({}^\frown\lambda x_8\, \mathfrak{g}(\textit{John catches } \texttt{him}_8 \textit{ and eats } \texttt{him}_8) \right) && \equiv\\
&\lambda Q\, \exists x\, [\mathsf{fish}'(x)\ \&\ Q\{x\}] \left({}^\frown\lambda x_8\, \mathfrak{g}\ (\mathsf{John})\ ({}^\frown\mathfrak{g}(\mathsf{catch}\ \texttt{him}_8\ \mathsf{and\ eat}\ \texttt{him}_8)) \right) && \equiv\\
&\exists x \left[\mathsf{fish}'(x)\ \&\ \lambda x_8\, \lambda P\, [P\{j\}] \left({}^\frown\lambda z\, [\mathfrak{g}(\mathsf{catch}\ \texttt{him}_8)(z)\ \&\ \mathfrak{g}(\mathsf{eat}\ \texttt{him}_8)(z)] \right) (x) \right] && \equiv\\
&\exists x \left[\mathsf{fish}'(x)\ \&\ \lambda x_8 \lambda z \left[\mathsf{catch}'\, ({}^\frown\lambda R\, [R\{x_8\}])(z)\ \&\ \mathsf{eat}'\, ({}^\frown\lambda R\, [R\{x_8\}](z)) \right] (x)\, (j) \right] && \equiv\\
&\exists x \left[\mathsf{fish}'(x)\ \&\ \lambda x_8 \lambda z\, [\mathsf{catch}'_\star\, (z, x_8)\ \&\ \mathsf{eat}'_\star\, (z, x_8)]\, (x)\, (j) \right] && \equiv\\
&\exists x \left[\mathsf{fish}'(x)\ \&\ \mathsf{catch}'_\star\, (j, x)\ \&\ \mathsf{eat}'_\star\, (j, x) \right]
\end{aligned}
$$

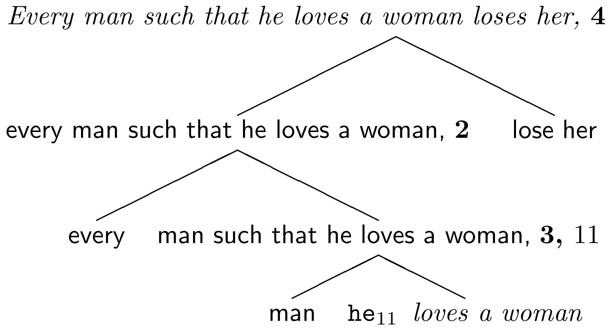
$$
\begin{aligned}
&\mathfrak{g}\ (\textit{John catches a fish and eats it}) && \equiv\\
&\mathfrak{g}\,(\mathsf{John})\ ({}^\frown\mathfrak{g}(\mathsf{catch\ a\ fish\ and\ eat\ it})) && \equiv\\
&\lambda P\, [P\{j\}] \left({}^\frown\lambda y\, \mathfrak{g}(\mathsf{a\ fish}) \left({}^\frown\lambda x_9\, [\mathfrak{g}(\mathsf{catch}\ \texttt{him}_9\ \mathsf{and\ eat}\ \texttt{him}_9)\, (y)] \right) \right) && \equiv\\
&\lambda y\, \lambda Q\, \exists x\, [\mathsf{fish}'(x)\ \&\ Q\{x\}] \left({}^\frown\lambda x_9\, [\lambda z\, [\mathfrak{g}(\mathsf{catch}\ \texttt{him}_9)(z)\ \&\ \mathfrak{g}(\mathsf{eat}\ \texttt{him}_9)(z)](y)] \right) (j) && \equiv\\
&\lambda y\, \exists x \left[\mathsf{fish}'(x)\ \&\ \lambda x_9 \left(\lambda z \left[\mathsf{catch}'\, ({}^\frown\lambda R\, [R\{x_9\}])(z)\ \&\ \mathsf{eat}'\, ({}^\frown\lambda R\, [R\{x_9\}](z)) \right] (y) \right) (x) \right] (j) && \equiv\\
&\lambda y\, \exists x \left[\mathsf{fish}'(x)\ \&\ \lambda x_9 \left[\mathsf{catch}'\, (y, {}^\frown\lambda R\, [R\{x_9\}])\ \&\ \mathsf{eat}'\, (y, {}^\frown\lambda R\, [R\{x_9\}]) \right] (x) \right] (j) && \equiv\\
&\lambda y\, \exists x \left[\mathsf{fish}'(x)\ \&\ [\mathsf{catch}'\, (y, x)\ \&\ \mathsf{eat}'\, (y, x)]\, (j) \right] && \equiv\\
&\exists x \left[\mathsf{fish}'(x)\ \&\ \mathsf{catch}'\, (j, x)\ \&\ \mathsf{eat}'\, (j, x) \right]
\end{aligned}
$$

Oczywiście reguła S16 może być zastosowana do zdania *John loves Mary.* z przykładu 2.13 w sposób równie banalny i nie wnoszący nic nowego jak reguła S14 do zdania *John walks.* z przykładu 2.2 w przykładzie 2.12. Jednak dla zdania *Every man catches a fish and eats it.* interpretacje analogiczne do powyższych nie byłyby już rzecz jasna równoważne.

Nawet jednak reguła S16 nie zapewnia traktowania zakresu działania kwantyfikatorów w sposób w pełni satysfakcjonujący. Weźmy jako przykład zdanie *Every man such that he loves a woman loses her.* powinien posiadać dwa rozbiory (a stąd i dwie translacje). Jeden rozbiór, który powoduje, że kwantyfikator uniwersalny pojawia się wewnątrz zakresu kwantyfikatora egzystencjalnego jest następujący:



Niestety rozbiór związany z przeciwnym usytuowaniem kwantyfiaktorów nie daje się skonstruować w sposób poprawny. Jak widać, nie daje się zanalizować frazy lose her, gdyż w takiej fazie rozbioru nie powinien już występować zaimek nieindeksowany. Z tego także względu nie ma sposobu na powiązanie zaimka her z frazą a woman.



<u>Reguła konstrukcji względnych</u>

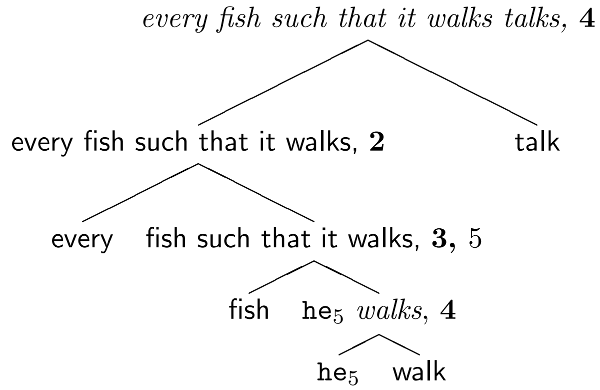
Syntaktyczna operacja polegająca na wiązaniu „indeksowanych” zaimków jest stosowana w przypadku konstrucji względnych. Konstrukcje takie są jednak dość skomplikowane. Montague ograniczył się do najprostszego przypadku *such that*, dość rzadkiego w „naturalnej” angielszczyźnie, w celu zachowania przejrzystości opisu. Jednak zaprezentowaną regułę można zmodyfikować w sposób umożliwiający uwzględnienie trudniejszych konstrukcji względnych.

Prezentowana reguła łączy rzeczownik pospolity (kategorii *CN*) ze zdaniem (kategrii t) w nową frazę kategorii *CN*. Zauważmy, że reguła ta <u>nie</u> dotyczy fraz rzeczownikowych kategorii T. Przyczyna ma charakter semantyczny i jest związana z zakresem rodzajnika (który musi zostać dołączony do frazy kategorii *CN*). Frazę *Every man who walks* rozumiemy jako każdego spacerującego mężczyznę (tzn. *Every (man who walks)*), bynajmniej nie mamy na myśli, że wszyscy mężczyźni spacerują (a więc *(Every man) who walks* jest bez sensu). Przy okazji blokuje to zastosowanie reguły S14 (patrz strona 17) bądź S16 (patrz strona 30), co groziłoby kontrintuicyjnym wiązaniem zaimków w zdaniach typu *Every student such that he talk admires every professor such that he talk.*.

S3 Jeśli $\xi \in P_{CN}$ oraz $\phi \in P_t$, to $F_{3,n}(\xi, \phi) = \xi$ such that $\phi' \in P_{CN}$, gdzie ϕ' powstaje z ϕ przez zastąpienie każdego wystąpienia $\texttt{he}_n/\texttt{him}_n$ przez he/him, she/her lub it w zależności od rodzaju pierwszego $\beta \in B_{CN}$ w ξ.

T3 Jeśli $\xi \in P_{CN}$ oraz $\phi \in P_t$, to $\mathfrak{g}(F_{3,n}(\xi, \phi)) = \lambda x_n\, [\mathfrak{g}(\xi)\,(x_n)\ \&\ \mathfrak{g}(\phi)]$.

Przykład *2.26.* Analiza zdania *Every fish such that it walks talks.* przedstawiona jest za pomocą poniższego drzewa.



Zdanie to przekładane jest na język IL jako

$$
\begin{array}{ll}
\mathfrak{g}\ (\textit{every fish such that it walks talks}) & \equiv \\
\mathfrak{g}\ (\text{every fish such that it walks})\,({}^\frown\mathfrak{g}(\text{talk})) & \equiv \\
\mathfrak{g}\ (\text{every})\left({}^\frown\mathfrak{g}(\text{fish such that it walks})\right)({}^\frown\text{talk}') & \equiv \\
\lambda P\left[\lambda Q\,\forall x\ [P\{x\} \rightarrow Q\{x\}]\right]\left({}^\frown(\lambda x_5\,[\mathfrak{g}(\text{fish})\,(x_5)\ \&\ \mathfrak{g}(\mathtt{he}_5\ \text{walks})])\right)({}^\frown\text{talk}') & \equiv \\
\lambda Q\left[\forall x\ \left[\left({}^\frown(\lambda x_5\,[\text{fish}'(x_5)\ \&\ \text{walk}'(x_5)])\right)\{x\}\ \rightarrow\ Q\{x\}\right]\right]({}^\frown\text{talk}') & \equiv \\
\forall x\ \left[\text{fish'}(x)\ \&\ \text{walk}'(x)\ \rightarrow\ \text{talk}'(x)\right] &
\end{array}
$$

Gramatyka PTQ umożliwia zagnieżdżanie konstrukcji względnych, np. *Every man such that he walks such that he talks loves Mary.*.

Niniejsze omówienie nie zawiera kompletnego opisu gramatyki PTQ, nawet względem pracy [Dowty i in., 1981], tzn. pominięte zostały frazy przyimkowe oraz kwestia czasu gramatycznego (ang. *tense*). Gramatyka ta była także rozszerzana i modyfikowana. Mimo to jednak wydaje się, że przedyskutowane zostały podstawowe własności tego podejścia, co daje wyobrażenie o samym formalizmie i jego potencjalnych możliwościach.

3 Teoria reprezentacji dyskursu

Teoria reprezentacji dyskursu (ang. *Discourse Representation Theory*, w skrócie DRT) zaproponowana przez Kampa i Reyle'go (1993) jest jedną z bardziej popularnych i współcześnie rozwijanych teorio-modelowych metod reprezentacji semantyki języka naturalnego. Jej zasadniczą własnością jest fakt, że interpretacja semantyczna przypisywana jest nie pojedynczym zdaniom, lecz dłuższym fragmentom tekstu czy dialogu (dyskursu), przy zachowaniu *spójności semantycznej* (ang. *semantic cohesiveness*) analizowanego tekstu. Analiza semantyczna prowadzona jest krok po kroku dla kolejnych zdań: reprezentacja semantyczna zanalizowanego fragmentu tekstu stanowi kontekst dla dalszego przetwarzania. Autorzy przyjęli przy tym upraszczające założenie, że pierwsze zdanie posiada pusty kontekst, a więc ignorują wszelką wiedzę o świecie, posiadaną w rzeczywistości przez każdego „użytkownika" języka naturalnego. Uproszczenie to nie ma jednak żadnego wpływu na samą teorię — nietrudno uzupełnić ją o „informację wstępną".

Poniżej prezentujemy jedynie podstawowy fragment formalizmu, tak w zakresie języka logicznego, jak i zestawu reguł analizy semantycznej dyskursu, których działanie nie wymaga bardziej złożonych pojęć. Fragment ten wystarcza do ukazania podstawowych własności DRT.

3.1 Syntaktyka

Podobnie jak autorzy, zacznijmy od przedstawienia syntaktyki pewnego ograniczonego fragmentu języka angielskiego, umożliwiającego analizę zjawisk będących tematem ich rozważań. Wybrany przez nich do tego celu formalizm syntaktyczny to GPSG (ang. *Generalized Phrased Structured Grammar*) zaproponowany przez Gazdara i in. (1985).

Podobnie jak w przypadku gramatyki PTQ Montague, zaczniemy od wymienienia zestawu kategorii gramatycznych używanych w omawianym formalizmie syntaktycznym. Ponieważ jednak nie są wśród nich wyróżnione kategorie proste i złożone, a przypisywanie wyrażeń podstawowych (słownikowych) ma bardziej złożony charakter, zrezygnujemy z ustawiania ich w tabeli, lecz wyliczymy je wprost. Są to: S — zdanie, VP′ — fraza czasownikowa złożona, VP — fraza czasownikowa prosta, V — czasownik, AUX — czasownik pomocniczy, NP — fraza rzeczownikowa, N — rzeczownik pospolity, PN — nazwa własna, PRO — zaimek osobowy, DET — rodzajnik, RC — fraza względna oraz RPRO — zaimek względny. Rozróżnienie dwóch kategorii fraz czasownikowych spowodowane jest faktem, że negacja w języku angielskim wymaga użycia czasownika pomocniczego `do`, `does` (por. reguły PS 4 i PS 5 w tabeli 3.3).

Jak w większości języków, także w angielskim wyrazy należące do poszczególnych kategorii posiadają pewne *cechy* (ang. *features*), które podlegają uzgodnieniom w ramach zdania. Tak więc poszczególne kategorie muszą być oznakowane (subkategoryzowane) przez takie cechy (*atrybuty* — ang. *atributes* tych kategorii). W poniższej tabeli (3.1) przedstawimy zestaw atrybutów niezbędnych do opisu wybranego podzbioru języka angielskiego.

Nazwa atrybutu	Opis atrybutu	Zbiór wartości	Kategorie znakowane daną cechą
Num	liczba	*sing, plur*	S, VP′, VP, V, AUX, NP, N, PN, PRO, DET
Case	przypadek	*+nom, –nom*	PRO
Gen	rodzaj	*male, fem, –hum*	NP, N, PN, PRO
Trans	przechodniość	+, −	V
Fin	finitywność	+, −	V
Gap	luka	−, *NP*	S, VP′, VP
Dis	rożróżnienie	+, −	S, VP′, NP
Refl	zwrotność	+, −	PRO

Tabela 3.1: Atrybuty poszczególnych kategorii składniowych

Atrybut *Dis* służy do konstruowania ciągów fraz typu $X, X, ...X$ **or** (**and**) X gwarantując, że spójnik **or** bądź **and** pojawi się wyłącznie na końcu ciągu. Domyślną wartością (pominiętą w pozostałych regułach) tego atrybutu jest $+$. Czasowniki (V) posiadają ponadto atrybuty *Tense* (czas) o wartościach *Past, Pres* oraz *Person* (osoba) o wartościach 1, 2, 3. Ponieważ jednak czasowniki w czasie przeszłym oraz w czasie teraźniejszym w liczbie charakteryzują się synkretyzmem form ze względu na osobę, w rezultacie uzyskujemy ograniczony zestaw atrybutów: *Past*, $\langle Pres, plur\rangle$, $\langle Pres, sing1^{st}\rangle$, $\langle Pres, sing2^{nd}\rangle$, $\langle Pres, sing3^{rd}\rangle$. Zamiast umieszczać tę morfologiczną informację bezpośrednio w indeksach kategorii, autorzy zaproponowali specjalnie zmodyfikowane reguły słownikowe (LI 19 i LI 20 w tabeli 3.2).

Kategoriom tym odpowiadają następujace reguły struktur frazowych (tabela 3.3) oraz słownikowe (tabela 3.2). Symbol / oznacza alternatywę, zaś symbol $\star$ — wszystkie możliwe wartości atrybutu. Pominięte zostały reguły LI 9 – LI 14 przypisujące rzeczownikom (nazwom własnym oraz pospolitym) oraz LI 17 – LI 18 czasownikom (przechodnim bądź nie) wartości słownikowe.

LI 1 $\mathrm{PRO}\begin{bmatrix} Num = sing \\ Gen = male \\ Case = +nom \\ Refl = - \end{bmatrix} \longrightarrow$ he

LI 2 $\mathrm{PRO}\begin{bmatrix} Num = sing \\ Gen = male \\ Case = -nom \\ Refl = - \end{bmatrix} \longrightarrow$ him

LI 3 $\mathrm{PRO}\begin{bmatrix} Num = sing \\ Gen = fem \\ Case = +nom \\ Refl = - \end{bmatrix} \longrightarrow$ she

LI 4 $\mathrm{PRO}\begin{bmatrix} Num = sing \\ Gen = fem \\ Case = -nom \\ Refl = - \end{bmatrix} \longrightarrow$ her

LI 5 $\mathrm{PRO}\begin{bmatrix} Num = sing \\ Gen = -hum \\ Case = \star \\ Refl = - \end{bmatrix} \longrightarrow$ it

LI 6 $\mathrm{PRO}\begin{bmatrix} Num = plur \\ Gen = \star \\ Case = +nom \\ Refl = - \end{bmatrix} \longrightarrow$ they

LI 7 $\mathrm{PRO}\begin{bmatrix} Num = plur \\ Gen = \star \\ Case = -nom \\ Refl = - \end{bmatrix} \longrightarrow$ them

LI 8 $\mathrm{PRO}\begin{bmatrix} Num = sing \\ Gen = -hum \\ Case = -nom \\ Refl = - \end{bmatrix} \longrightarrow$ itself

LI 9 $\mathrm{PRO}\begin{bmatrix} Num = sing \\ Gen = male \\ Case = -nom \\ Refl = + \end{bmatrix} \longrightarrow$ himself

LI 10 $\mathrm{PRO}\begin{bmatrix} Num = sing \\ Gen = fem \\ Case = -nom \\ Refl = + \end{bmatrix} \longrightarrow$ herself

LI 11 $\mathrm{PRO}\begin{bmatrix} Num = plur \\ Gen = \star \\ Case = -nom \\ Refl = + \end{bmatrix} \longrightarrow$ themselve

LI 12 $\mathrm{DET}_{Num\, =\, sing} \longrightarrow$ a, every, the, some

LI 13 $\mathrm{AUX}\begin{bmatrix} Num = sing \\ Gen = + \end{bmatrix} \longrightarrow$ do

LI 14 $\mathrm{AUX}\begin{bmatrix} Num = plur \\ Gen = + \end{bmatrix} \longrightarrow$ does

LI 15 $\mathrm{V}\begin{bmatrix} Num = sing \\ Fin = + \\ Trans = \beta \end{bmatrix} \longrightarrow \langle Pres, sing3^{rd}\rangle(\alpha)$
gdzie $\alpha \in \mathrm{V}\begin{bmatrix} Num = \star \\ Fin = - \\ Trans = \beta \end{bmatrix}$

LI 16 $\mathrm{V}\begin{bmatrix} Num = plur \\ Fin = + \\ Trans = \beta \end{bmatrix} \longrightarrow \langle Pres, plur\rangle(\alpha)$
gdzie $\alpha \in \mathrm{V}\begin{bmatrix} Num = \star \\ Fin = - \\ Trans = \beta \end{bmatrix}$

LI 17 $\mathrm{RPRO}\begin{bmatrix} Num = \star \\ Gen = m/f \end{bmatrix} \longrightarrow$ who

LI 18 $\mathrm{RPRO}\begin{bmatrix} Num = \star \\ Gen = -hum \end{bmatrix} \longrightarrow$ which

Tabela 3.2: Reguły słownikowe gramatyki Kampa i Reyle'a

Znaczenie poszczególnych reguły wydaje się intuicyjnie zrozumiałe. Jedyny kłopot mogą sprawić reguły PS 13 i PS 14 opisujące konstrukcje względne. Są one traktowane jako „rzeczownik złożony" składający się ze zwykłego rzeczownika oraz frazy względnej (reg. PS 13), która z kolei składa się z zaimka względnego poprzedzającego zdanie (reg. PS 14). Ponieważ w zdaniu tym któraś z fraz rzeczownikowych (podmiot bądź dopełnienie) jest luką (*Gap*), taka fraza realizowana jest jako pusty ciąg wyrazów (reg. PS 8). Zauważmy, że w „zwykłym” zdaniu atrybut *Gap* ma wartość $-$.

Zwróćmy ponadto uwagę na fakt, że reguły PS 16A, PS 16B, PS 17A i PS 17B są to tak naprawde wzorce reguł, a nie pojedyncze reguły. Dwie pierwsze formują wzorzec dla ciągu fraz stanowiących alternatywę, przy czym symbol X oznacza jedną z kategorii S, VP$'$, V, NP i N. Natomiast dwie kolejne tworzą wzorzec dla ciągu fraz stanowiących koniunkcję, przy czym symbol Y oznacza jedną z kategorii S, VP$'$ i V. Przez Ξ oznaczono zbiór atrybutów właściwych dla danej kategorii (prócz *Disj*) wraz z ich wartościami, które nie ulegają w wyniku stosowania tych reguł zmianie.

PS 1
$$\mathrm{S}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Gap} & = & - \end{bmatrix} \longrightarrow \mathrm{NP}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Gen} & = & \beta \\ \mathit{Case} & = & +\mathit{nom} \\ \mathit{Gap} & = & - \end{bmatrix} \mathrm{VP}'\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Fin} & = & + \\ \mathit{Gap} & = & - \end{bmatrix}$$

PS 2
$$\mathrm{S}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Gap} & = & \mathrm{NP}_{\mathit{Num}=\gamma} \end{bmatrix} \longrightarrow \mathrm{NP}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Gen} & = & \beta \\ \mathit{Case} & = & +\mathit{nom} \\ \mathit{Gap} & = & \mathrm{NP}_{\mathit{Num}=\gamma} \end{bmatrix} \mathrm{VP}'\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Fin} & = & + \\ \mathit{Gap} & = & - \end{bmatrix}$$

PS 3
$$\mathrm{S}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Gap} & = & \mathrm{NP}_{\mathit{Num}=\gamma} \end{bmatrix} \longrightarrow \mathrm{NP}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Gen} & = & \beta \\ \mathit{Case} & = & +\mathit{nom} \\ \mathit{Gap} & = & - \end{bmatrix} \mathrm{VP}'\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Fin} & = & + \\ \mathit{Gap} & = & \mathrm{NP}_{\mathit{Num}=\gamma} \end{bmatrix}$$

PS 4
$$\mathrm{VP}'\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Fin} & = & + \\ \mathit{Gap} & = & \gamma \end{bmatrix} \longrightarrow \mathrm{AUX}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Fin} & = & + \\ \mathit{Gap} & = & - \end{bmatrix} \textbf{not}\ \mathrm{VP}\begin{bmatrix} \mathit{Num} & = & \delta \\ \mathit{Fin} & = & - \\ \mathit{Gap} & = & \gamma \end{bmatrix}$$

PS 5
$$\mathrm{VP}'\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Fin} & = & + \\ \mathit{Gap} & = & \gamma \end{bmatrix} \longrightarrow \mathrm{VP}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Fin} & = & + \\ \mathit{Gap} & = & \gamma \end{bmatrix}$$

PS 6
$$\mathrm{VP}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Fin} & = & \beta \\ \mathit{Gap} & = & \gamma \end{bmatrix} \longrightarrow \mathrm{V}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Fin} & = & \beta \\ \mathit{Trans} & = & + \end{bmatrix} \mathrm{NP}\begin{bmatrix} \mathit{Num} & = & \gamma \\ \mathit{Gen} & = & \delta \\ \mathit{Case} & = & -\mathit{nom} \\ \mathit{Gap} & = & \gamma \end{bmatrix}$$

PS 7
$$\mathrm{VP}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Fin} & = & \beta \end{bmatrix} \longrightarrow \mathrm{V}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Fin} & = & \beta \\ \mathit{Trans} & = & - \end{bmatrix}$$

PS 8
$$\mathrm{NP}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Gen} & = & \beta \\ \mathit{Case} & = & \gamma \\ \mathit{Gap} & = & \mathrm{NP}_{\mathit{Num}=\alpha} \end{bmatrix} \longrightarrow \emptyset$$

PS 9
$$\mathrm{NP}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Gen} & = & \beta \\ \mathit{Case} & = & \gamma \end{bmatrix} \longrightarrow \mathrm{DET}_{\mathit{Num} = \alpha}\ \mathrm{N}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Gen} & = & \beta \end{bmatrix}$$

PS 10
$$\mathrm{NP}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Gen} & = & \beta \\ \mathit{Case} & = & \gamma \end{bmatrix} \longrightarrow \mathrm{PN}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Gen} & = & \beta \\ \mathit{Case} & = & \gamma \end{bmatrix}$$

PS 11
$$\mathrm{NP}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Gen} & = & \beta \\ \mathit{Case} & = & \gamma \end{bmatrix} \longrightarrow \mathrm{PRO}\begin{bmatrix} \mathit{Num} & = & \alpha \\ \mathit{Gen} & = & \beta \\ \mathit{Case} & = & \gamma \end{bmatrix}$$

PS 12
$$\mathrm{NP}\begin{bmatrix} \mathit{Num} & = & \mathit{plur} \\ \mathit{Gen} & = & \beta \\ \mathit{Case} & = & \gamma \end{bmatrix} \longrightarrow \mathrm{NP}\begin{bmatrix} \mathit{Num} & = & \delta \\ \mathit{Gen} & = & \epsilon \\ \mathit{Case} & = & \gamma \end{bmatrix} \textbf{and}\ \mathrm{NP}\begin{bmatrix} \mathit{Num} & = & \eta \\ \mathit{Gen} & = & \zeta \\ \mathit{Case} & = & \gamma \end{bmatrix}$$

$$\text{PS 13}\quad \mathrm{N}\begin{bmatrix} Num = \alpha \\ Gen = \beta \end{bmatrix} \longrightarrow \mathrm{N}\begin{bmatrix} Num = \alpha \\ Gen = \beta \end{bmatrix} \mathrm{RC}\begin{bmatrix} Num = \alpha \\ Gen = \beta \end{bmatrix}$$

$$\text{PS 14}\quad \mathrm{RC}\begin{bmatrix} Num = \alpha \\ Gen = \beta \end{bmatrix} \longrightarrow \mathrm{RPRO}\begin{bmatrix} Num = \alpha \\ Gen = \beta \end{bmatrix} \mathrm{S}\begin{bmatrix} Num = \alpha \\ Gap = \mathrm{NP}_{Num=\alpha} \end{bmatrix}$$

$$\text{PS 15}\quad \mathrm{S}\begin{bmatrix} Num = \alpha \\ Gap = - \end{bmatrix} \longrightarrow \textbf{if}\ \mathrm{S}\begin{bmatrix} Num = \beta \\ Gap = - \end{bmatrix} \textbf{then}\ \mathrm{S}\begin{bmatrix} Num = \gamma \\ Gap = - \end{bmatrix}$$

$$\text{PS 16A}\quad \mathrm{X}\begin{bmatrix} \Xi \\ Dis = + \end{bmatrix} \longrightarrow \mathrm{X}\begin{bmatrix} \Xi \\ Dis = - \end{bmatrix}, \mathrm{X}\begin{bmatrix} \Xi \\ Dis = + \end{bmatrix}$$

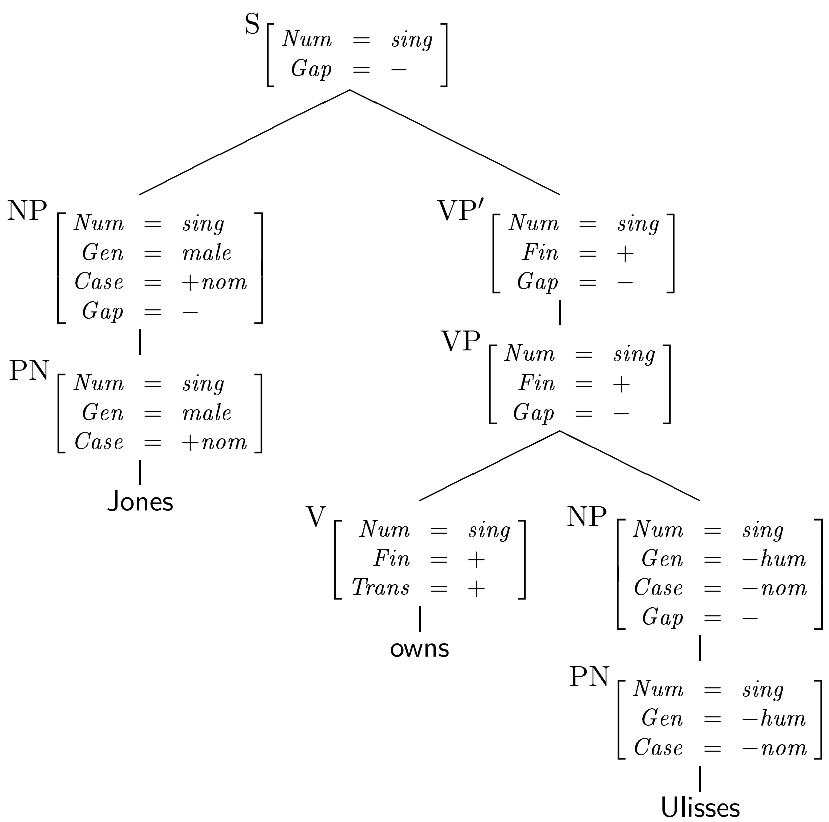
$$\text{PS 16B}\quad \mathrm{X}\begin{bmatrix} \Xi \\ Dis = + \end{bmatrix} \longrightarrow \mathrm{X}\begin{bmatrix} \Xi \\ Dis = - \end{bmatrix} \textbf{or}\ \mathrm{X}\begin{bmatrix} \Xi \\ Dis = - \end{bmatrix}$$

$$\text{PS 17A}\quad \mathrm{Y}\begin{bmatrix} \Xi \\ Dis = + \end{bmatrix} \longrightarrow \mathrm{Y}\begin{bmatrix} \Xi \\ Dis = - \end{bmatrix}, \mathrm{Y}\begin{bmatrix} \Xi \\ Dis = + \end{bmatrix}$$

$$\text{PS 17B}\quad \mathrm{Y}\begin{bmatrix} \Xi \\ Dis = + \end{bmatrix} \longrightarrow \mathrm{Y}\begin{bmatrix} \Xi \\ Dis = - \end{bmatrix} \textbf{and}\ \mathrm{Y}\begin{bmatrix} \Xi \\ Dis = - \end{bmatrix}$$

$$\text{PS 18}\quad \mathrm{DET}_{Num = \alpha} \longrightarrow \mathrm{NP}\begin{bmatrix} Num = \alpha \\ Gen = \beta \\ Case = +nom \\ Gap = - \end{bmatrix}$$

Tabela 3.3: Reguły struktur frazowych gramatyki Kampa i Reyle'a



Rozbiory zdań tworzone za pomocą tej gramatyki reprezentowane są za pomocą *drzew rozbioru.* Drzewo D o korzeniu r zapisujemy jako listę $\langle r, \langle D^1\rangle, ..., \langle D^n\rangle\rangle$, gdzie $D^1, ..., D^n$ są drzewami potomnymi (poddrzewami) dla węzła r; drzewo takie będzie także oznaczane $\mathfrak{d}(r)$. Ponadto węzeł $n \in D$ wtedy i tylko wtedy, gdy $n = r$ lub $n \in D^1$, ..., $n \in D^n$. Podobnie $\mathfrak{d} \in D$ wtedy i tylko wtedy, gdy $\mathfrak{d} = D^1$, ..., $\mathfrak{d} = D^n$ lub $\mathfrak{d} \in D^1, ..., \mathfrak{d} \in D^n$. Zbiór wszystkich drzew rozbioru będziemy oznaczać przez $\mathfrak{D}$.

Drzewo rozbioru zdania przedstawimy na prostym przykładzie, wraz z reprezentacją graficzną. W przykładzie tym uwzględniono wszystkie atrybuty; w dalszych rozważaniach zazwyczaj nierelewnantne atrybuty będą pomijane.

Przykład *3.1.* Weźmy proste zdanie *Jones owns Ulisses.*. Jego drzewo rozbioru $\langle \mathrm{S}, \langle \mathrm{NP}, \langle \mathrm{PN}, \langle \mathsf{Jones} \rangle\rangle\rangle$, $\langle \mathsf{VP'}, \langle \mathsf{VP}, \langle \mathsf{V}, \langle \mathsf{owns} \rangle\rangle, \langle \mathsf{NP}, \langle \mathsf{PN}, \langle \mathsf{Ulisses} \rangle\rangle\rangle\rangle\rangle\rangle$ przedstawione jest graficznie powyżej.

3.2 Formalny opis DRT

W niniejszym rozdziale przedstawimy formalny opis języka logicznego, będącego podstawą Teorii Reprezentacji Dyskursu. Zaczniemy jak zwykle od języka, w którym omawiany formalizm zostanie zapisany. Pragniemy przy tym zwrócić uwagę na fakt, że poniższy zapis różni się nieco od oryginału.

3.2.1 Syntaktyka

Alfabet tej logiki jest następujący:

(i) niepusty, przeliczalny zbiór znaczników dyskursu R (ang. *discourse referents*);

(ii) niepusty, przeliczalny zbiór „symboli” (autorzy używają angielskiego określenia *vocabulary*) V składający się z:

   (a) zbioru stałych $V_\pi \subseteq V$ odpowiadających nazwom własnym;

   (b) zestawu zbiorów $V_n \subseteq V$ n-argumentowych symboli predykatywnych, wśród których wyróżniamy:

   - dwa podzbiory jednoargumentowych symboli predykatywnych: $V_\eta \subseteq V_1$ odpowiadający rzeczownikom pospolitym oraz $V_\zeta \subseteq V_1$ odpowiadający czasownikom nieprzechodnim;
   - zbiór dwuargumentowych symboli predykatywnych $V_\xi \subseteq V_2$ odpowiadający czasownikom przechodnim;
   - wyróżniony dwuargumentowy symbol predykatywny $= \in V_2$.

Język prezentowanego formalizmu nie składa się z termów ani formuł, tylko ze *Struktur Reprezentacji Dyskursu*, (ang. *Discourse Representation Structures*, w skrócie DRS).

Definicja 3.1. Struktura DRS K względem (ang. *confined to*) V i R jest to para $\langle U_K, \mathrm{Con}_K \rangle$, gdzie $U_K \subseteq R$ jest zbiorem znaczników dyskursu, zaś Con_K jest zbiorem DRS-warunków względem V i R, które mogą przyjmować następujące postaci:

(a) $x = y$, gdzie $x, y \in R$;

(b) $\pi(x)$, gdzie $x \in R$ i $\pi \in V_\pi$;

(c) $\eta(x)$, gdzie $x \in R$ i $\eta \in V_\eta$;

(d) $x\zeta$, gdzie $x \in R$ i $\zeta \in V_\zeta$;

(e) $x\xi y$, gdzie $x, y \in R$, zaś $\xi \in V_\xi$;

(f) $\neg L$, gdzie L jest DRS-em względem V i R;

(g) $L_1 \Longrightarrow L_2$, gdzie L_1, L_2 są DRS-ami względem V i R,

(h) $L_1 \vee ... \vee L_n$, gdzie $L_1, ..., L_n$ są DRS-ami względem V i R.

Zbiór wszystkich struktur DRS będziemy oznaczać przez $\mathfrak{K}$. Zauważmy, że warunki (f)–(h) powodują, że mamy do czynienia z definicją rekurencyjną — warunki wewnątrz DRS-a mogą zawierać inne DRS-y. Warunki tego typu nazywamy *warunkami złożonymi*, zaś warunki takie jak (a)–(e) nazywamy *warunkami prostymi* bądź *atomowymi*. Aby odróżnić DRS skonstruowany dla pewnego zdania czy dyskursu od jego pod-DRS-ów, będziemy go nazwać *głównym*, a pod-DRS-y będą określane mianem podrzędnych. Natomiast struktura nie posiadająca struktur podrzędnych, a więc złożona z samych warunków prostych, będzie także nazywana strukturą *prostą*.

Definicja 3.2. Niech $K_1 = \langle U_{K_1}, \mathrm{Con}_{K_1}\rangle$, $K_2 = \langle U_{K_2}, \mathrm{Con}_{K_2}\rangle$ będą dowolnymi DRS-ami. Powiemy, że K_1 jest *bezpośrednio podrzędny* (ang. *immediate subordinate*) K_2 (piszemy $K_1 \lessdot K_2$) wtw, gdy $\neg K_1 \in \mathrm{Con}_{K_2}$ lub $K_1 \implies L \in \mathrm{Con}_{K_2}$ lub $L \implies K_1 \in \mathrm{Con}_{K_2}$ (dla pewnego DRS-u L) lub $L_1 \vee ... \vee L_n \in \mathrm{Con}_{K_2}$ i $K_1 = L_i$ (dla pewnych DRS-ów $L_1, ..., L_n$). Powiemy, że K_1 jest *podrzędny* (ang. *subordinate*) K_2 (piszemy $K_1 < K_2$) wtw, gdy $K_1 \lessdot K_2$ lub istnieje K_3 taka, że $K_1 \lessdot K_3$ i $K_3 < K_2$. Powiemy, że K_1 jest *słabo podrzędny* (ang. *weakly subordinate*) K_2 (piszemy $K_1 \leqslant K_2$) wtw, gdy $K_1 = K_2$ lub $K_1 < K_2$. Wówczas K_2 nazwiemy *(bezpośrednio, słabo) nadrzędnym*, odpowiednio. Powiemy, że DRS K_1 jest *główny* wtw, gdy $\neg(K_1 < K_2)$ dla dowolnego DRS K_2.

Pewne istotne definicje dotyczą także znaczników dyskursu.

Definicja 3.3. Niech K oraz $K' = \langle U_{K'}, \mathrm{Con}_{K'}\rangle$ będą DRS-ami, zaś $\mathtt{x} \in U_{K_1}$ będzie znacznikiem dyskursu. Powiemy, że $\mathtt{x}$ jest *dostępny* (ang. *accessible*) z K wtw, gdy:

(a) albo $K \leqslant K'$,

(b) albo istnieją K_1 i $K_2 = \langle U_{K_2}, \mathrm{Con}_{K_2}\rangle$ takie, że $K' \implies K_1 \in \mathrm{Con}_{K_2}$ oraz $K \leqslant K_1$.

Zbiór wszystkich znaczników dyskursu dostępnych z DRS-u K (czyli dla jego warunków) będziemy oznaczać jako $\overline{U_K}$.

Tak więc z danego „poziomu” widać nie tylko znaczniki dyskursu ze wszystkich struktur nadrzędnych, ale także ze wszystkich ich poprzedników implikacji. Natomiast składniki dyzjunkcji nie są dla siebie nawzajem widoczne, i to niezależnie od uporządkowania. Taka a nie inna postać powyższej definicji jest uwarunkowana czysto lingwistycznie zasadami wiązania zaimków osobowych.

Z drugiej strony, zbiór wszystkich znaczników dyskursu zawartych w zbiorach znaczników pewnej struktury K oraz wszystkich jej struktur podrzędnych będziemy oznaczać jako $U_K^{\downarrow}$. Rzecz jasna dla struktur prostych $\overline{U_K} = U_K^{\downarrow} = U_K$.

Ponieważ $=$ jest relacją równoważności[11] w każdej strukturze możemy wyznaczyć klasy abstrakcji równych sobie znaczników dyskursu.

Definicja 3.4. Niech $K = \langle U_K, \mathrm{Con}_K\rangle$, będzie dowolnym DRS-em, zaś $x \in \overline{U_K} \cup U_K^{\downarrow}$[12] będzie dowolnym znacznikiem dyskursu. Wówczas przez $[x]_K \subseteq \overline{U_K} \cup U_K^{\downarrow}$ będziemy oznaczać zbiór wszystkich znaczników dyskursu $y \in \overline{U_K} \cup U_K^{\downarrow}$ takich, że $x = y \in \mathrm{Con}_L$ lub istnieje $z \in \overline{U_K} \cup U_K^{\downarrow}$ taki, że $x = z \in \mathrm{Con}_{L_1}$ oraz $z = y \in \mathrm{Con}_{L_2}$, gdzie L, L_1, L_2 są dowolnymi DRS-ami takimi, że $L \leqslant K$, $L_1 \leqslant K$ oraz $L_2 \leqslant K$.

Kamp i Reyle nie narzucają ograniczeń na występowanie tych samych znaczników dyskursu w różnych strukturach podrzędnych tej samej struktury. W przypadku struktur powiązanych relacją dostępności znaczników ma to wielkie, trudne do przecenienia znaczenie, bowiem w przeciwieństwie do rachunku predykatów nie istnieje tu zjawisko *zasłaniania*, wręcz przeciwnie, jeśli dowolny znacznik x

[11] Co w zasadzie można stwierdzić jedynie na poziomie semantycznym (patrz poniżej), gdyż (prezentowana wersja) formalizmu DRT nie posiada aksjoamtyzacji.

[12] Czyli wszystkie znaczniki dyskursu dostępne z K oraz te, dla których x jest „współdostępna”.

jest dostępny w pewnej strukturze K to przynależność tegoż znacznika do U_K jest redundantna, tzn. nie ma wpływu na interpretację struktury.

Dlatego niejednokrotnie warto ograniczyć się do klasy struktur, w których zbiory znaczników dyskursu w dowolnych dwóch strukturach podrzędnych względem pewnej podstruktury są rozłączne. Klasę wszystkich takich struktur będziemy oznaczać przez $\mathfrak{K}^{\neq}$. Pamiętajmy jednak, że „urozłącznienie" zbiorów znaczników dyskursu poprzez zamianę wszystkich wystąpień wybranego znacznika we wszystkich strukturach podrzędnych jakiejkolwiek struktury jest niedopuszczalne, gdyż zmienia jej interpretację.

Przykład *3.2.* Niech $K = \langle\{x\}, \{\mathbf{Jones}\,(x)\}\rangle$, $L = \langle\{y\}, \{\mathbf{Smith}\,(y)\}\rangle$. Wówczas warunek $K \Longrightarrow \langle\emptyset, \{L \Longrightarrow K\}\rangle$ zapiszemy jako $\gamma_1 = \langle\{x\}, \{\mathbf{Jones}\,(x)\}\rangle \Longrightarrow \langle\emptyset, \{\langle\{y\}, \{\mathbf{Smith}\,(y)\}\rangle \Longrightarrow \langle\{x\}, \{\mathbf{Jones}\,(x)\}\rangle\}\rangle$, natomiast warunek $\gamma_2 = \langle\{x\}, \{\mathbf{Jones}\,(x)\}\rangle \Longrightarrow \langle\emptyset, \{\langle\{y\}, \{\mathbf{Smith}\,(y)\}\rangle \Longrightarrow \langle\{z\}, \{\mathbf{Jones}\,(z)\}\rangle\}\rangle$ różniący się od poprzedniego tym, że w drugim wystąpieniu K zastąpiliśmy znacznik x przez z jest odmiennym warunkiem.

Definicja 3.5. Znacznik dyskursu z jest *wolny* w strukturze K wtw, gdy $z \notin U_K$ oraz jest on wolny w pewnym warunku $\gamma \in \mathrm{Con}_K$, tzn.:

(a) γ ma postać $x = y$ oraz $z \approx x$ lub $z \approx y$;

(b) γ ma postać $\pi(x)$ oraz $z \approx x$;

(c) γ ma postać $\eta(x)$ oraz $z \approx x$;

(d) γ ma postać $x\zeta$ oraz $z \approx x$;

(e) γ ma postać $x\xi y$ oraz $z \approx x$ lub $z \approx y$;

(f) γ ma postać $\neg L$ oraz z jest wolna w L;

(g) γ ma postać $L_1 \Longrightarrow L_2$ oraz z jest wolna w L_1 i L_2;

(h) γ ma postać $L_1 \vee ... \vee L_n$ oraz z jest wolna w $L_i \in \{L_1 \vee ... \vee L_n\}$;

gdzie $z \approx x$ oznacza identyczność symboli.

Powyższa definicja oznacza, że znacznik z jest wolny, kiedy nie należy do zbioru U swojego DRS i żadnego z jego nadrzędników (czyli nie jest nigdzie „zaczepiony"). Nie jest to rzecz jasna sytuacja poprawna.

Definicja 3.6. Struktura DRS K jest poprawna wtedy i tylko wtedy, gdy nie ma w niej odwołań do żadnego wolnego znacznika dyskursu.

Poniżej przedstawimy przykłady struktur.

Przykład *3.3.* Niech $R = \{x, y, z, u, v, q, r\}$, $V_\pi = \{\mathbf{Jones}, \mathbf{Smith}, \mathbf{Ulysses}, \mathbf{Candide}\}$, $V_\eta = \{\mathbf{Porsche}\}$, $V_\xi = \{\mathbf{owns}, \mathbf{fascinates}\}$.
Strukturą DRS wzgl. V i R jest $K = \langle U_K, \mathrm{Con}_K\rangle$, gdzie $U_K = \{x, y, u, v\}$ oraz $\mathrm{Con}_K = \{\mathbf{Jones}\,(x), \mathbf{Ulysses}\,(y),\ x\ \mathbf{owns}\ y,\ u = y,\ v = x,\ u\ \mathbf{fascinates}\ v\}$.
Kolejną DRS względem V i R jest $K' = \langle U_{K'}, \mathrm{Con}_{K'}\rangle$, gdzie $U_{K'} = \{x, y, z\}$ oraz $\mathrm{Con}_{K'} = \{\mathbf{Jones}\,(x), \mathbf{Ulysses}\,(y),\ x\ \mathbf{owns}\ y,\ z = y,\ \neg\langle U = \{u\}, \mathrm{Con} = \{\mathbf{Porsche}\,(u),\ z\ \mathbf{owns}\ u\}\rangle\}$.
Graficzna reprezentacja tych struktur widnieje obok.

$$\boxed{\begin{array}{c} x, y, u, v \\ \mathbf{Jones}\,(x) \\ \mathbf{Ulysses}\,(y) \\ x\ \mathbf{owns}\ y \\ u = y \\ v = x \\ u\ \mathbf{fascinates}\ v \end{array}} \qquad \boxed{\begin{array}{c} x, y, z \\ \mathbf{Jones}\,(x) \\ \mathbf{Ulysses}\,(y) \\ x\ \mathbf{owns}\ y \\ z = x \\ \neg \boxed{\begin{array}{c} u \\ \mathbf{Porsche}\,(u) \\ z\ \mathbf{owns}\ u \end{array}} \end{array}}$$

3.2.2 Semantyka

Dla tak sformułowanego języka można już zaproponować odpowiednią definicję modelu i weryfikacji struktury DRS w tymże modelu. Zanim to uczynimy, przypomnimy pewne pomocnicze pojęcia związane z pojęciem funkcji.

Definicja 3.7. Niech X, Y, X_1, X_2 będą zbiorami, $X_1, X_2 \subseteq X$. Rozważmy funkcje $f\colon X_1 \longrightarrow Y$ oraz $g\colon X_2 \longrightarrow Y$. Obie funkcje f i g są funkcjami *częściowymi* ze zbioru X. Dziedziną funkcji f jest X_1 (piszemy $\mathrm{Dom}(f) = X_1$), zaś funkcji g — X_2. Powiemy, że funkcje f i g są *zgodne* (ang. *compatible*), jeśli $f(x) = g(x)$ dla $x \in X_1 \cap X_2$. Jeżeli ponadto $X_1 \subseteq X_2$, to funkcja g *rozszerza* funkcję f (piszemy $f \subseteq g$). Dla zgodnych funkcji f i g możemy w naturalny sposób zdefiniować ich sumę $f \cup g$, przecięcie $f \cap g$ oraz różnicę $f - g$ o dziedzinach $\mathrm{Dom}(f) \cup \mathrm{Dom}(g)$, $\mathrm{Dom}(f) \cap \mathrm{Dom}(g)$ oraz $\mathrm{Dom}(f) - \mathrm{Dom}(g)$, odpowiednio.

Stwierdzenie 3.1. Dla dowolnej funkcji f, jedyną rozszerzającą ją funkcją $g \supseteq f$ taką, że $\mathrm{Dom}(g) = \mathrm{Dom}(f)$, jest sama f.

Definicja 3.8. Modelem M dla zbioru symboli V jest trójka uporządkowana: $\langle U_M, R^n_M, \Phi_M\rangle$, gdzie:

(a) U_M jest zbiorem obiektów (*dziedziną, uniwersum*);

(b) $R^n_M \subseteq 2^{(U_M)^n}$ jest zbiorem relacji n-argumentowych;

(c) $\Phi_M = \langle \mathsf{Name}_M, \mathsf{Pred}_M\rangle$ jest interpretacją, czyli parą funkcji takich, że:

- $\mathsf{Name}_M\colon V_\pi \longrightarrow U_M$ jest funkcją przypisującą nazwom $\pi \in V_\pi$ obiekty z U_M;
- $\mathsf{Pred}^n_M\colon V_n \longrightarrow R^n_M$ jest funkcją przypisującą n-argumentowym symbolom predykatywnym z V_n n-argumentowe relacje z R^n_M.

Kamp i Reyle nie posługują się terminem *wartościowania* i *spełnialności*, tylko *osadzenia* (ang. *embedding*) zbioru znaczników dyskursu oraz *weryfikacji* (ang. *verification*) struktury K w modelu.

Definicja 3.9. Niech $K = \langle U_K, \mathrm{Con}_K\rangle$ będzie DRS-em względem V i R, zaś $M = \langle U_M, R^n_M, \Phi_M\rangle$ będzie modelem względem V. *Osadzeniem* (ang. *embedding*) nazywamy funkcję częściową $f\colon R \longrightarrow U_M$; zbiór wszystkich osadzeń w pewnym modelu będzemy oznaczać przez $\mathfrak{E}_M$. Powiemy, że f *weryfikuje* warunek $\gamma \in \mathrm{Con}_K$ w modelu M (piszemy $M, f \models \gamma$), gdy:

(a) $M, f \models x = y$ wtw, gdy $f(x) = f(y)$;

(b) $M, f \models \pi(x)$ wtw, gdy $f(x) = \mathsf{Name}_M(\pi)$;[13]

(c) $M, f \models \eta(x)$ wtw, gdy $f(x) \in \mathsf{Pred}^1_M(\eta)$;

(d) $M, f \models x\,\zeta$ wtw, gdy $f(x) \in \mathsf{Pred}^1_M(\zeta)$;

(e) $M, f \models x\,\xi\,y$ wtw, gdy $\langle f(x), f(y)\rangle \in \mathsf{Pred}^2_M(\xi)$;

(f) $M, f \models \neg L$ wtw, gdy nie istnieje osadzenie $g\colon R \longrightarrow U_M$ takie, że $f \subseteq g$ i $\mathrm{Dom}(g) = \mathrm{Dom}(f) \cup U_L$ i $M, g \models L$;

[13] W tym miejscu warto zwrócić uwagę na tzw. *identyfikację obiektów*. Zgodnie z prezentowaną tu (i oryginalnie przez Kampa i Reyle'go) definicją funkcja Name_M przypisuje każdej nazwie własnej pojedynczy obiekt, czyli każda nazwa *jednoznacznie* identyfikuje pewien obiekt. Możliwe jest jednak mniej rygorystyczne podejście, w którym ta sama nazwa może reprezentować kilka obiektów (wielu ludzi nosi nazwisko *Jones*); wówczas Name_M przyporządkowywaóby nazwom pewien zbiór obiektów i pisalibyśmy $f(x) \in \mathsf{Name}_M(\pi)$.

(g) $M, f \models L_1 \Longrightarrow L_2$ wtw, gdy dla każdego osadzenia $g\colon R \longrightarrow U_M$ takiego, że $f \subseteq g$ i $\mathrm{Dom}(g) = \mathrm{Dom}(f) \cup U_{L_1}$ i $M, g \models L_1$ istnieje $h\colon R \longrightarrow U_M$ takie, że $g \subseteq h$ i $\mathrm{Dom}(h) = \mathrm{Dom}(g) \cup U_{L_2}$ i $M, h \models L_2$;

(h) $M, f \models L_1 \vee ... \vee L_n$ wtw, gdy istnieje $L_i \in \{L_1, ..., L_n\}$ takie, że dla pewnego osadzenia $g\colon R \longrightarrow U_M$ takiego, że $f \subseteq g$ i $\mathrm{Dom}(g) = \mathrm{Dom}(f) \cup U_{L_i}$ zachodzi $M, g \models L_i$.

Powiemy, że f *weryfikuje* strukturę K w modelu M (piszemy $M, f \models K$) wtw, gdy $M, f \models \gamma$ dla każdego $\gamma \in \mathrm{Con}_K$.[14] Strukturę K nazywamy *prawdziwą* w modelu M (piszemy $M \models K$) wtedy i tylko wtedy, gdy istnieje osadzenie $f\colon R \longrightarrow U_M$ takie, że $\mathrm{Dom}(f) = U_K$ oraz $M, f \models K$. Strukturę K nazywamy *logicznie prawdziwą* względem V, jeśli jest jest prawdziwa we wszystkich modelach dla V. Ponadto powiemy, że struktura K jest *logiczną konsekwencją*[15] struktur $K_1, ..., K_n$ (piszymy $K_1, ..., K_n \vdash K$), jeżeli dla każdego modelu M dla V takiego, że $M \models K_i$ dla $1 \leqslant i \leqslant n$ zachodzi $M \models K$. Struktury K_1 i K_2 nazwiemy *równoważnymi* wtw, gdy $K_1 \vdash K_2$ oraz $K_2 \vdash K_1$.

Zauważmy, że jest to definicja słabsza niż w przypadku wartościowania zmiennych w rachunku predykatów I rzędu, gdyż wówczas formuły muszą być spełnione dla dowolnego wartościowania.

To, czy dane osadzenie weryfikuje pewną strukturę czy też nie zależy w równym stopniu od obiektów przypisywanych poszczególnym znacznikom, jak i od jego dziedziny. Dlatego tak ważne jest wymaganie w definicji prawdziwości struktury w modelu, by była weryfikowana przez osadzenie o dziedzinie równej jej zbiorowi znaczników dyskursu. Równie istotne jest, w której strukturze znaczniki są umieszczane.

Przykład *3.4.* Rozważmy model $M = \langle U_M, R^n_M, \Phi_M \rangle$, w którym $U_M = \{a, b, c\}$ oraz struktury $K = \langle \emptyset, \{\neg\langle \{x, y\}, \{x = y\}\rangle\}\rangle$, $K' = \langle \{x\}, \{\neg\langle \{y\}, \{x = y\}\rangle\}\rangle$ oraz $K'' = \langle \{x, y\}, \{\neg\langle \emptyset, \{x = y\}\rangle\}\rangle$. Jak łatwo sprawdzić, $M, f \models K$, $M, f \models K'$ i $M, f \models K''$ dla dowolnego osadzenia f takiego, że $\{x, y\} \subseteq \mathrm{Dom}(f)$ oraz $f(x) \neq f(y)$. Weźmy teraz dowolne g takie, że $\mathrm{Dom}(g) = U_{K'} = \{x\}$. Dla ustalenia uwagi przyjmijmy, że $g(x) = a$. Wówczas istnieje h takie, że $g \subseteq h$ oraz $\mathrm{Dom}(h) = \mathrm{Dom}(g) \cup \{y\} = \{x, y\}$ takie, że $g(x) = g(y) = a$, czyli $M, g \not\models K'$. Zupełnie identycznie rzecz się ma z K. Tak więc $M \not\models K$, $M \not\models K'$, jednak $M \models K''$.

Istnieje wiele ogólnych zależności pomiędzy strukturami wynikających z ich budowy, czyli tak naprawdę z semantyki łączących je operatorów. Zależności takie omówione zostały w [Hajnicz, 2003]. Poniższe twierdzenie zawiera kilka przykładowych zależności; jego dowód można z naleźć we wspomnianej pracy.

Twierdzenie 3.1. Niech L, L_1, L_2, L_3 będą dowolnym strukturami, zaś U dowolnym zbiorem. Wówczas dla następujących struktur zachodzi zależność równoważności:

(i) L oraz $\langle \emptyset, \{\neg\langle \emptyset, \{\neg L\}\rangle\}\rangle$,

(ii) $\langle U, \{\langle \emptyset, \{L_1 \vee L_2\}\rangle \vee L_3\}\rangle$ oraz $\langle U, \{L_1 \vee \langle \emptyset, \{L_2 \vee L_3\}\rangle\}\rangle$ oraz $\langle U, \{L_1 \vee L_2 \vee L_3\}\rangle$,

(iii) $\langle U, \{L_1 \Longrightarrow \langle \emptyset, \{L_2 \Longrightarrow L_3\}\rangle\}\rangle$ oraz $\langle U, \{L_2 \Longrightarrow \langle \emptyset, \{L_1 \Longrightarrow L_3\}\rangle\}\rangle$.

Modele zdefiniowane dla tej samej klasy symboli V tworzą pewne uporządkowane rodziny.

Definicja 3.10. Dla dowolnych modeli $M = \langle U_M, R^n_M, \Phi_M \rangle$ oraz $M' = \langle U_{M'}, R^n_{M'}, \Phi_{M'} \rangle$ powiemy, że M' *rozszerza* (ang. *extendes*) M (piszemy $M \sqsubseteq M'$) wtedy i tylko wtedy, gdy $U_M \subseteq U_{M'}$, $\mathsf{Name}_M = \mathsf{Name}_{M'}$ oraz dla dowolnego predykatu $Q \in V$ mamy $\mathsf{Pred}_M(Q) = \mathsf{Pred}_{M'}(Q)/_{U_M}$.

[14] Brakuje tu warunku, że $U_K \subseteq f$, który autorzy prawdopodobnie uznali za domyślny.

[15] W systemach formalnych termin *logicznej konsekwencji* używany jest zazwyczaj w kontekście dowodu formalnego, a nie prawdziwości semantycznej. Jedynie w przypadku systemów *pełnych* mamy do czynienia z ich równoważnością.

Powyższe definicje zilustrujemy za pomocą następującego przykładu.

Przykład *3.5.* Niech V i R będą takie, jak w przykładzie 3.3. Rozważmy dwa modele: $M_1 = \langle U_{M_1}, R^n_{M_1}, \Phi_{M_1}\rangle$, przy czym $U_{M_1} = \{a, b, c, d, e, f\}$, $R^1_{M_1} = \mathsf{Porsche}_{M_1}$ oraz $R^2_{M_1} = \{\mathsf{owns}_{M_1}, \mathsf{fascinates}_{M_1}\}$, gdzie $\mathsf{Porsche}_{M_1} = \{f\}$, $\mathsf{owns}_{M_1} = \{\langle a, d\rangle, \langle a, e\rangle, \langle b, e\rangle, \langle b, f\rangle, \langle c, f\rangle\}$, zaś $\mathsf{fascinates}_{M_1} = \{\langle c, a\rangle, \langle c, b\rangle, \langle d, a\rangle, \langle d, b\rangle, \langle e, a\rangle, \langle f, a\rangle\}$. Funkcja Name_{M_1} opisana jest za pomocą następującej tabelki:

Jones	**Smith**	**Ulysses**	**Candide**
a	b	d	e

$M_2 = \langle U_{M_2}, R^n_{M_2}, \Phi_{M_1}\rangle$, gdzie $U_{M_2} = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n\}$, $R^2_{M_2} = \mathsf{Porsche}_{M_2}$ oraz $R^2_{M_2} = \{\mathsf{owns}_{M_2}, \mathsf{fascinates}_{M_2}\}$. Ponadto $\mathsf{Porsche}_{M_2} = \{f, k\}$, $\mathsf{owns}_{M_2} = \{\langle a, d\rangle, \langle a, e\rangle, \langle a, g\rangle, \langle a, k\rangle, \langle b, e\rangle, \langle b, f\rangle, \langle c, f\rangle\}$, zaś $\mathsf{fascinates}_{M_2} = \{\langle a, b\rangle, \langle c, a\rangle, \langle c, b\rangle, \langle d, a\rangle, \langle d, b\rangle, \langle e, a\rangle, \langle f, a\rangle, \langle h, a\rangle, \langle k, a\rangle\}$. Natomiast $\mathsf{Name}_{M_3} = \mathsf{Name}_{M_1}$.

Jak łatwo stwierdzić, $M_1 \sqsubseteq M2$. Ponadto dla struktury K zdefiniowanej w przykładzie 3.3 oraz dowolnego osadzenia f takiego, że $f(x) = f(v) = a$ oraz $f(y) = f(u) = d$ zachodzi $M_1, f \models K$ oraz $M_2, f \models K$.

Weźmy teraz osadzenie f' takie, że $f'(x) = f'(z) = a$ oraz $f'(y) = d$. Ponieważ $\langle a, d\rangle \in \mathsf{owns}_{M_1}$ oraz $\langle a, d\rangle \in \mathsf{owns}_{M_2}$, pierwsze cztery warunki z $\mathrm{Con}_{K'}$ z przykładu 3.3 są spełnione w obu tych modelach. Rozważmy teraz dowolne osadzenie g rozszerzające f' takie, że $\mathrm{Dom}(g) = \mathrm{Dom}(f') \cup \{u\}$. Aby spełnić warunek **Porsche** (u) w modelu M_1 musimy przyjąć $g(u) = f$. Ponieważ $\langle a, f\rangle \notin \mathsf{owns}_{M_1}$, żadne takie g nie spełnia warunków podstruktury, a więc spełniony jest piąty warunek struktury K' jako całość. Tak więc $M_1, f' \models K'$. Natomiast w przypadku modelu M_2 również dla osadzenia g takiego, że $g(u) = k$ zachodzi **Porsche** (u). Jednak tym razem $\langle a, k\rangle \notin \mathsf{owns}_{M_2}$, więc takie g spełnia warunki podstruktury, co falsyfikuje piąty warunek struktury K' jako całość. Przeto $M_2, f' \models K'$.

Zależność między modelami znajdującymi się w relacji rozszerzania oraz zbiorem weryfikowanych przez nie formuł nie jest przypadkowa. Pokazuje to następująca definicja.

Definicja 3.11. Dowolną strukturę K taką, że jeżeli $M \models K$ dla pewnego modelu M, to $M' \models K$ dla dowolnego modelu $M' \sqsupseteq M$, nazywamy strukturą *trwałą* (ang. *persistent*).

Tak więc struktura K z przykładu 3.3 jest trwała, zaś struktura K' — nie.

3.3 Dyskurs i algorytm jego analizy

Zasadniczym celem pracy nie jest opracowanie formalnego opisu struktur DRS, lecz formalnej semantyki dla dyskursu, do czego DRS-y mają służyć jako narzędzie do reprezentacji. Zajmiemy się więc teraz podstawową częścią teorii, a mianowicie algorytmem przekształcania zbioru (a właściwie ciągu, skoro kolejność występowania zdań w dyskursie nie jest obojętna, wręcz przeciwnie, determinuje kolejność ich przetwarzania) zdań na odpowiadającą im strukturę DRS.

3.3.1 Dyskurs

Zanim przejdziemy do konkretnych sposobów interpretowania dyskursu, przeanalizujemy pewne jego ważne własności w powiązaniu z formalizmem wprowadzonym w poprzednim rozdziale. Sformułowania języka naturalnego nie są ze swej natury jednoznaczne i dotyczy to także tak prostego podzbioru języka, jak ten generowany przez gramatykę z rozdziału 3.1, choćby ze względu na obecność zaimków osobowych czy też alternatywy bądź koniunkcji kilku fraz. Rozważmy więc pewien ciąg zdań stanowiący spójny dyskurs, który w rezultacie także nie musi być jednoznaczny. Jako że wiele zdań może mieć więcej niż jeden rozbiór, dowolny ciąg wybranych drzew rozbioru poszczególnych zdań stanowi reprezentację syntaktyczną tego ciągu.

Niech więc $\mathfrak{Sen}$ oznacza zbiór wszystkich ciągów drzew rozbiorów zdań uzyskanych dla wszystkich możliwych dyskursów omawianej gramatyki. Ponadto niech $\mathfrak{S} = \langle S_1, S_2, ..., S_n \rangle \in \mathfrak{Sen}^{\mathfrak{Dys}}$ będzie pewnym ciągiem drzew rozbiorów zdań (w skrócie *rozbiorem*) dla dyskursu $\mathfrak{Dys}$ spośród wszystkich możliwych rozbiorów $\mathfrak{Sen}^{\mathfrak{Dys}}$.

Oczywiście algorytm (symbolizowany przez funkcję $\mathfrak{G}$ zdefiniowaną poniżej) nie działa bezpośrednio na dyskursie, tylko na pojedynczym reprezentującym go ciągu rozbiorów zdań $\mathfrak{S}$, czego rezultatem jest pewna struktura DRS.

Definicja 3.12. Niech $\mathfrak{Dys}$ będzie dyskursem, którego słownik odpowiada zbiorowi symboli V, $\mathfrak{S}$ pewnym jego rozbiorem, K będzie DRS-em, zaś M pewnym modelem dla V. Powiemy, że dyskurs $\mathfrak{Dys}$ *jest prawdziwy w modelu M względem interpretacji K* (piszemy $M, K \models \mathfrak{Dys}$) wtedy i tylko wtedy, gdy $K = \mathfrak{G}(\mathfrak{S})$ oraz $M \models K$.

Rzecz jasna wielość drzew rozbioru nie oznacza jeszcze, że dyskurs jest niejednoznaczny, gdyż mogą one posiadać taką samą interpretację semantyczną. Z drugiej strony, dla jednego $\mathfrak{S}$ może istnieć kilka różnych DRS-ów będących rezultatem działania algorytmu (czyli nie do końca deterministycznego doboru reguł konstrukcyjnych stanowiących realizację poszczególnych wywołań rekurencyjnej funkcji $\mathfrak{g}$ wykonujących pojedynczy krok algorytmu, co świadczy o tym, że $\mathfrak{G}$ nie jest tak naprawdę funkcją, chociaż ma funkcjopodobną definicję, tylko relacją). Dlatego jego prawdziwość w modelu może być rozpatrywana tylko względem określonej struktury DRS.

Definicja 3.13. Dyskurs $\mathfrak{Dys}$ nazwiemy *jednoznacznym* wtw, gdy dowolne struktury K_1, K_2 takie, że $K_1 = \mathfrak{G}(\mathfrak{S}_1)$ oraz $K_2 = \mathfrak{G}(\mathfrak{S}_2)$ (dla $\mathfrak{S}_1, \mathfrak{S}_2 \in \mathfrak{Sen}^{\mathfrak{Dys}}$) są izomorficzne.

Trudno stwierdzić, dlaczego autorzy zaproponowali tak mocną definicję jednoznaczności dyskursu. Wydaje się przecież, że *równoważność* (por. def. 3.9) wszystkich interpretacji dyskursu wystarczy, by uznać go za taki. Oczywiście dwie równoważne struktury mogą różnić się położeniem znaczników w podstrukturach w sposób determinujący dalsze przetwarzanie. Jednak po jego zakończeniu (a chodzi tu wszak o interpretacje dyskursu jako <u>całości</u>) kwestia ta już się liczyć nie powinna.

Dla jednoznacznych dyskursów można zdefiniować prawdziwość bezpośrednio w modelu.

Definicja 3.14. Niech $\mathfrak{Dys}$ będzie jednoznacznym dyskursem, zaś M modelem dla zbioru symboli V. Powiemy, że $\mathfrak{Dys}$ *jest prawdziwy w modelu M* (piszemy $M \models \mathfrak{Dys}$) wtedy i tylko wtedy, gdy $M, K \models \mathfrak{S}$ dla dowolnego rozbioru $\mathfrak{S} \in \mathfrak{Sen}^{\mathfrak{Dys}}$ oraz dowolnej struktury K takiej, że $K = \mathfrak{G}(\mathfrak{S})$. W przeciwnym razie $\mathfrak{S}$ *jest fałszywy w modelu M*.

Nie wiadomo, dlaczego powyższa definicja została sformułowana jedynie dla jednoznacznych dyskursów, chociaż dla pozostałych dyskursów wystarczy wybrać <u>pewną</u> jego interpretację K, prawdziwą w danym modelu, by móc to powiedzieć o całym dyskursie.

3.3.2 Podstawowe zasady algorytmu

Właściwy algorytm działa nie na dyskursie, tylko na jego pojedynczym rozbiorze $\mathfrak{S} = \langle S_1, S_2, ..., S_n \rangle$. Przez $\mathfrak{S}^i = \langle S_i, ..., S_n \rangle$ będziemy oznaczać ogon tej listy bez jej pierwszych $i-1$ elementów. Analiza ciągu $\mathfrak{S}$ polega na iteracyjnym przetwarzaniu pary $\langle K_{i-1}, S_i \rangle$ złożonej ze zdania i struktury w nową strukturę K_i wzbogaconą o znaczenie zdania S_i. Ponieważ początkowo kontekst jest pusty, odpowiada mu struktura pusta $K_0 = \langle \emptyset, \emptyset \rangle$.

Definicja 3.15. Algorytm jest to funkcja rekurencyjna $\mathfrak{G}\colon \mathfrak{K} \times \mathfrak{Sen} \longrightarrow \mathfrak{K} \times \mathfrak{Sen}$ taką, że:

- $\mathfrak{G}(K_{i-1}, \mathfrak{S}^i) = \mathfrak{G}(\hat{\mathfrak{g}}(K_{i-1}, S_i), \mathfrak{S}^{i+1})$,
- $\mathfrak{G}(K_n, \langle\rangle) = \langle K_n, \langle\rangle \rangle$.

Jak wynika z powyższej definicji, pojedynczy krok algorytmu realizowany jest przez funkcję $\hat{\mathfrak{g}}$, która z kolei polega na rekurencyjnym wywoływaniu kolejnej funkcji, $\mathfrak{g}$. Konieczność zdefiniowania kolejnej funkcji wynika z faktu, że obiektem jej działania nie są pary ⟨struktura, drzewo⟩ jak w przypadku $\hat{\mathfrak{g}}$, ani po prostu „zwykłe” struktury DRS zdefiniowane w rozdz. 3.2.

Po pierwsze, ponieważ konstruowanie struktury DRS polega na równoległym przetwarzaniu tejże struktury i analizowanego zdania, drzewo rozbioru aktualnego zdania dołączane jest do zbioru warunków struktury. Ponieważ syntaktyka DRS-ów takiej możliwości nie przewiduje, struktury zawierające drzewa rozbioru (pojedyncze drzewo rozbioru zdania może ulec w trakcie analizy podziałowi na kilka poddrzew) należą już do innej, szerszej klasy struktur. Po całkowitym przetworzeniu drzewo takie staje się drzewem pustym, i wówczas taka quasi-struktura staje się „zwykłym” DRS-em.

Kolejne komplikacje powoduje jedna z najważniejszych reguł systemu, odpowiedzialna za wiązanie zaimków osobowych, a mianowicie CR.PRO. Zgodnie z wszelkimi zasadami, zaimek taki musi być zgodny co do rodzaju i liczby z rzeczownikiem, z którym jest wiązany (nie dotyczy to przypadka, gdyż charakteryzuje on jedynie dane wystąpienie rzeczownika czy zaimka). Aby spełnić ten warunek, Kamp i Reyle wprowadzają funkcje *Gen* i *Num*, które przypisują znacznikom dyskursu wartości rodzaju i liczby, takie same co nadawane reprezentowanym przez nie rzeczownikom przez odpowiednie reguły słownikowe. Funkcje te nie pojawiają się w „zwykłych” DRS-ach (autorzy w ogóle nie uwzględniają symboli funkcyjnych) i nie są interpretowane w modelu (czyli kwestia zgodności rodzaju i liczby z obiektami przypisywanymi nazwom własnym i rzeczownikom pospolitym, faktycznie nieco problematyczna, jest w ogóle ignorowana).[16]

W następnym rozdziale prezentowane są reguły składające się w sumie na funkcję $\mathfrak{g}$. Żeby to jednak faktycznie była funkcja, należałoby ustalić kolejność realizowania poszczególnych reguł, która sama z siebie nie jest jednoznacznie zdeterminowana. Autorzy sugerują, by wywoływać je w kolejności „od góry i od lewej” w strukturze analizowanego drzewa (oczywiście zawsze spośród tych, które spełniają niezbędne warunki). Takie podejście umożliwia wprowadzanie do DRS-ów nowe obiekty w kolejności ich pojawiania się w dyskursie, co zabezpiecza przed niepoprawną interpretacją zdań typu *He loves a girl who hates Bill.* sugerującą, że to Bill kocha nienawidzącą go dziewczynę.

Innym rozwiązaniem jest zdefiniowanie hierarchii reguł mówiącej, w jakiej kolejności powinny być wywoływane. Na przykład reguła CR.PN mogłaby mieć najwyższy priorytet, co pozwoliłoby zrezygnować z odwoływania się w niej do struktury głownej (bo tylko w takich byłaby aplikowana). Mogłoby to nieco uprościć definicję funkcji $\mathfrak{g}$, gdyż nie trzeba by było ingerować w struktury nadrzędne.

Tego typu ograniczenia ważne są także z praktycznego punktu widzenia, gdyż zmniejszają złożoność algorytmu, czyli przyspieszają jego działanie. Pierwsze z nich wręcz narzuca kolejność aplikacji reguł w sposób jednoznaczny, czyli $\mathfrak{G}$ jest wówczas faktycznie funkcją, a cały algorytm jest jednoznaczny (dla danego ciągu drzew rozbioru $\mathfrak{S}$). Drugie jest znacznie mniej restrykcyjne, a zależy to w dużym stopniu od proponowanej hierarchii reguł. Jednak nawet liniowe uporządkowanie kolejności aplikacji reguł nie zapewnia takiej jednoznaczności jak pierwsze rozwiązanie, gdyż czasami jedną regułę można zastosować do więcej niż jednego poddrzewa danego drzewa (np. w zdaniu *Every student hates every professor who knows every book.*.

Powyższe sposoby determinują kolejność aplikowania reguł w pojedynczym drzewie. Niejednokrotnie jednak quasi-struktura zawiera ich więcej (np. po „rozmnożeniu” zdań zawierających koniunkcję lub alternatywę pewnych fraz). Czasami kolejność analizy tych drzew nie ma wpływu na efekt, gdyż uzyskane w jej wyniku struktury są równoważne. Czasami jednak otrzymane struktury różnią się między sobą. Nie byłoby to problemem, gdybyśmy w rezultacie uzyskali odmienne, lecz poprawne interpretacje. Czasami jednak nieodpowiednia kolejność analizy drzew prowadzi do błędnych, niedopuszczalnych interpretacji lub wręcz akceptowania dyskursów niepoprawnych semantycznie (por. rozdz. 3.4.5).

[16] Pamiętajmy, że mówimy tu o rodzaju morfosyntaktycznym, niejednokrotnie odmiennym w różnych językach, i mającym ograniczony związek ze znaczeniem wyrazów. Jest to b. wyraźne w j. polskim, gdzie np. *krzesło, fotel* i *kanapa* są różnych rodzajów. Przyjęcie, że konkretny obiekt-mebel będący materialną reprezentacją danego wyrazu nie ma rodzaju jest w pełni zasadne.

Jako remedium na ten problem autorzy proponują opatrzenie DRS-warunków indeksami, które regulowałyby kolejność ich przetwarzania. Ponieważ jednak dalszej analizie podlegają wyłącznie drzewa, wystarczy ograniczyć indeksowanie jedynie do tego konkretnego typu warunków.

Mając na względzie powyższe uwagi, możemy już zdefiniować pojęcie quasi-struktury (QDRS) będącej faktycznym obiektem działania algorytmu.

Definicja 3.16. Indeksem I nazywamy zbiór par $\langle D, \mathfrak{i} \rangle$, gdzie $D \in \mathfrak{D}$ jest pewnym drzewem rozbioru, zaś $\mathfrak{i}$ jest liczbą całkowitą. Zbiór wszystkich indeksów będziemy oznaczać przez $\mathfrak{I}$.

Definicja 3.17. Quasi-struktura QDRS K jest to para $\langle U_K, \mathrm{Con}_K \rangle$, gdzie U_K jest zbiorem znaczników dyskursu, zaś Con_K jest zbiorem uogólnionych (indeksowanych) QDRS-warunków, które mogą być „zwykłymi" DRS-warunkami zgodnie z def. 3.1 lub przyjmować jedną z następujących postaci:

(i) $Gen(x)$, gdzie $x \in R$,

(ii) $Num(x)$, gdzie $x \in R$,

(iii) $\langle D, I \rangle$, gdzie $D \in \mathfrak{D}$, $I \in \mathfrak{I}$.

Zbiór wszystkich quasi-struktur będziemy oznaczać przez $\mathfrak{Ki}$.

Mając zdefiniowane QDRS-y, możemy opisać, na czym polega pojedynczy krok algorytmu, mający na celu przetworzenia kolejnego zdania. Na wstępie musimy dołączyć do rozważanej struktury $K_i = \langle U_{K_i}, \mathrm{Con}_{K_i} \rangle$ rozbióru drzewa S_i, pamiętając o tym, że początkowym jego indeksem jest zbiór pusty.

Funkcja $\mathfrak{g}$ wywoływana jest rekurencyjnie dopóki qausi-struktura (oraz jej podstruktury) zawiera jakieś drzewo (a więc nie jest „zwykłym" DRS-em).

Definicja 3.18. Funkcja $\hat{\mathfrak{g}}\colon \mathfrak{Ki} \times \mathfrak{Sen} \longrightarrow \mathfrak{Ki}$ polega na rekurencyjnym wywoływaniu funkcji $\mathfrak{g}\colon \mathfrak{Ki} \longrightarrow \mathfrak{Ki}$, czyli na utworzeniu dla struktury $K_{i-1} = \langle U_{K_{i-1}}, \mathrm{Con}_{K_{i-1}} \rangle$ ciągu quasi-struktur $K_i^1, ..., K_i^m$ takiego, że:

(i) $K_i^1 = \langle U_{K_i^1}, \mathrm{Con}_{K_i^1} \rangle$ jest taką quasi-strukturą, że $U_{K_i^1} = U_{K_i}$ zaś $\mathrm{Con}_{K_i^1} = \mathrm{Con}_{K_{i-1}} \cup \langle \{S_i\}, \emptyset \rangle$;

(ii) $K_i^j = \mathfrak{g}(K_i^{j-1})$;

(iii) $K_i = K_i^m = \hat{\mathfrak{g}}(K_{i_1}, S_i)$ (ani jej struktury podrzędne) nie zawiera żadnych drzew.

Jednakże struktura QDRS $K_i = \hat{\mathfrak{g}}\,(K_{i-1}) = K_i^m$ bynajmniej nie jest „zwykłą" strukturą DRS, choć kolejne wywołania $\mathfrak{g}$ doprowadziły do usunięcia wszystkich poddrzew, gdyż zawiera jeszcze instancje funkcji *Gen, Num* dla wszystkich znaczników zdefiniowanych w K_i^m i jej strukturach podrzędnych. Mogą one jednak zostać usunięte <u>dopiero</u> po przeanalizowaniu całego $\mathfrak{S}$, czyli po zakończeniu algorytmu, jako że zaimki występujące w kolejnych zdaniach mogą odnosić się właśnie do nich.[17] Tak więc także funkcja $\hat{\mathfrak{g}}$ (a w rezultacie i cały algorytm) działa na quasi-strukturach QDRS, a nie zwykłych, poprawnych strukturach DRS z rozdziału 3.2.

Niedeterminizm funkcji $\mathfrak{g}$ ograniczany jest m.in. przez założenie, że jakakolwiek reguła przekształcająca dany warunek będący parą $\langle D, I \rangle \in \mathrm{Con}_K$ może zostać zastosowana do zawierającej go struktury K wtw, gdy dla dowolnego innego $\langle D', I' \rangle \in \mathrm{Con}_K$ nie istnieją takie $\langle D_k, \mathfrak{i}_k \rangle \in I$, $\langle D'_l, \mathfrak{i}'_l \rangle \in I'$, że $D_k = D'_l$ oraz $\mathfrak{i}'_l < \mathfrak{i}_k$.

Zauważmy, że w powyższym założeniu nie odwołujemy się w zaden sposób do „zawartości" drzewa/warunku; istotna jest jedynie identyczność napisów. Zamiast niego mógłby występować w tym miejscu

[17] Ponieważ jednak funkcje *Gen* i *Num* i tak nie są interpretowane w modelu, ich obecność nie ma wpływu na prawdziwość struktury, czyli od momentu zakończenia algorytmu tracą jakiekolwiek znaczenie i stają się „przezroczyste".

dowolny napis-identyfikator (np. kolejne litery a, b, c), byleby różny od innych identyfikatorów. Autorzy zapewne nie chcieli zawracać sobie głowy sposobem generowania takich identyfikatorów.

Kamp i Reyle dodali indeksy do definicji struktur (tzn. występujących w nich warunków) dopiero przy okazji omawiania koniunkcji. W niniejszym opracowaniu wszystkie wcześniejsze definicje zostały zmodyfikowane, jednak w towarzyszących im przykładach indeksy (zawsze puste) zostały zignorowane, by nie zaciemniać i tak nieraz nieprostego rozumowania.

Zanim przejdziemy do omówienia poszczególnych reguł, chcielibyśmy wspomnieć o jeszcze jednej kwestii, związanej z naturą quasi-struktur. Język typowego formalizmu składa się z napisów (zazwyczaj nazywanych formułami) utworzonymi zgodnie z pewnymi regułami syntaktycznymi. Struktury DRS nie są co prawda napisami, tylko zbiorami napisów, więc są bliskie temu ideałowi, gdyż można uznać je za napisy z dokładnoscia do kolejności zapisu elementów zbiorów. Także drzewa w notacji listowej są doskonałymi napisami, więc dodanie ich do quasi-struktur nieczego nie burzy.

Poważny problem stanowią natomiast funkcje *Gen* i *Num*. Chodzi o to, że nie mamy tu do czynienia z symbolem funkcyjnym czy predykatywnym (gdybyśmy zastąpili funkcję $Gen(x)$ relacją $Gen(x, \alpha)$), czyli napisem, ale właśnie funkcją czy relacją (czyli zależnością). Napis $x = y$ jest tylko napisem (i dopiero interpretacja wymusza, by przypisano obu znacznikom ten sam obiekt). Tymczasem $Gen(x) = Gen(y)$ oznacza, że wartość funkcji jest identyczna, co warunkuje (a nie jest trywialną konsekwencją) stwierdzenia $x = y$, gdyż pozwala powiązać rzeczownik *book* z zaimkiem *it* a rzeczownik *professor* z zaimkiem *he*, ale nie odwrotnie. Jest to kwestia fundamentalna i żadne kruczki notacyjne tego nie zmienią.

Podobnie rzecz się ma z indeksami. Występujące w nich QDRS-warunki są poprawnymi napisami, a na dodatek operacja ich porównywania jest poprawną operacją na napisach.[18] Jednakże drugi element pary, *liczba całkowita*, nie może już być uznana za napis, gdyż podlegają one porównaniu liczbowemu, co ewidentnie nie jest operacją na napisach.

Ostatecznie musimy pogodzić się z faktem, że struktury QDRS nie są „wyrazami" języka formalnego, a więc i DRT jako całość nie jest „czystym" systemem formalnym. Nie należy uznawać tego od razu za wadę; po prostu trzeba o tym pamiętać.

Ponieważ w dalszych rozważaniach będzimy mieli do czynienia wyłącznie z quasi-strukturami QDRS, nie będziemy tego zazwyczaj podkreślać, nazywając je po prostu strukturami.

3.4 Reguły konstrukcyjne

Do zdefiniowania pozostało nam przekształcenie $\mathfrak{g} : \mathfrak{Ki} \longrightarrow \mathfrak{Ki}$ działające na zbiorze quasi-struktur. Jego definicja rozłożona jest na definicje zestawu reguł determinujących sposób określania znaczenia poszczególnych fraz składających się na zdanie S_i (czy też odpowidających im poddrzew rozbioru).

Reguły konstrukcyjne prezentowane w następnych podrozdziałach sformułowane są zawsze dla bieżącej struktury. Jednak przekształcenie takiej struktury automatycznie modyfikuje wszystkie jej struktury nadrzędne (ich zbiory warunków). Ponieważ na dodatek zgodnie z decyzją Kampa i Reyle'go niektóre reguły (np. CR.PN) odwołują się bezpośrednio do zbiorów znaczników dyskursu swych struktur nadrzędnych (głównych), zdecydowaliśmy się na zdefiniowanie przekształcenia $\mathfrak{g}$ wyłącznie na strukturach głównych, choć nie zawsze odbywa się to w sposób jawny.

Niech $K = \langle U_K, \mathrm{Con}_K\rangle$ będzie pewną quasi-strukturą, zaś $\bar{K} = \langle U_{\bar{K}}, \mathrm{Con}_{\bar{K}}\rangle \geqslant K$ będzie jej główną strukturą nadrzędną. Wówczas istnieje ciąg struktur $K = {}^1K \lhd {}^2K \lhd \ldots \lhd {}^lK = \bar{K}$ oraz $\mathfrak{g}(K) = \langle U_{K'}, \mathrm{Con}_{K'}\rangle$ wtw, gdy

(i) $\mathfrak{g}({}^iK) = \langle U_{{}^iK'}, \mathrm{Con}_{{}^iK'}\rangle$, gdzie $U_{{}^iK'} = U_{{}^iK}$ dla $i \neq 1$ oraz $i \neq l$,

(ii) $\mathrm{Con}_{{}^iK'} = \mathrm{Con}_{{}^iK} - \{\Upsilon({}^{i-1}K)\} \cup \{\Upsilon({}^{i-1}K')\}$, gdzie $\Upsilon(L) = \neg L$ lub $L \Longrightarrow L'$ lub $L' \Longrightarrow L$ lub $L_1 \vee \ldots \vee L \vee \ldots \vee L_k$.

[18] Podobnie jak zastępowanie (wszystkich wystąpień) napisu innym napisem, pojawiające się w poniższych regułach. Nawet operacje dodania/usunięcia elementu ze zbioru można ostatecznie uznać za operację dodania/usunięcia napisu.

3.4.1 Frazy rzeczownikowe

Reguła dla nazw własnych

Zaczniemy od najprostszego przypadku, całkowicie niezależnego od kontekstu, a mianowicie analizy podrzewa $\breve{S}^k = \langle \mathrm{NP}, \langle \mathrm{PN}, \langle \mathtt{nam} \rangle\rangle\rangle$ reprezentującego nazwy własne. Odpowiada za to reguła oznaczana **CR.PN** (ang. *Construction Rule for Proper Names*). Reguła ta dodaje nowy znacznik, wiąże go z nazwą, przypisuje mu właściwe wartości funkcji *Num* i *Gen* oraz zastępuje nim odpowiednie poddrzewo.

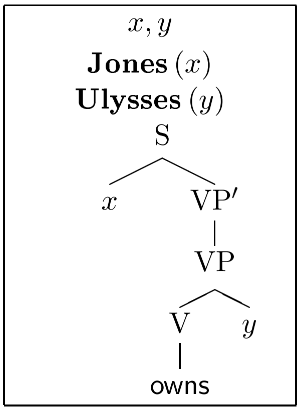
Aby podkreślić niezależność nazw własnych tak od kontekstu, jak i od struktury bieżącego zdania, realizujący ją znacznik dyskursu będzie zawsze umieszczany w strukturze głównej.

Definicja 3.19. (reguła **CR.PN**) Niech $\mathtt{ref} \in R$, zaś $K = \langle U_K, \mathrm{Con}_K \rangle$ będzie taką strukturą, że $\langle \breve{S}, I \rangle \in \mathrm{Con}_K$ oraz istnieje $\mathtt{nam} \in \mathrm{PN}$ takie, że $\breve{S}^k = \langle \mathrm{NP}, \langle \mathrm{PN}, \langle \mathtt{nam} \rangle\rangle\rangle \in \breve{\mathtt{S}}$. Niech $\bar{K} = \langle U_{\bar{K}}, \mathrm{Con}_{\bar{K}} \rangle \geqslant K$ będzie strukturą główna nadrzędną dla K. Wówczas $\mathfrak{g}(K) = \langle U_{K'}, \mathrm{Con}_{K'} \rangle$ jest strukturą spełniającą warunki:

1. $U_{\bar{K}'} = U_{\bar{K}} \cup \{\mathtt{ref}\}$,[19]
2. $\breve{S}' = \breve{S}[\langle \mathrm{NP}, \langle \mathrm{PN}, \langle \mathtt{nam} \rangle\rangle\rangle / \langle \mathtt{ref} \rangle]$,
3. $Num(\mathtt{ref}) = \mathrm{PN}.Num$,
4. $Gen(\mathtt{ref}) = \mathrm{PN}.Gen$,
5. $\mathrm{Con}_{K'} = \mathrm{Con}_K - \{\langle \breve{S}, I \rangle\} \cup \{\mathtt{nam}(\mathtt{ref}), Num(\mathtt{ref}), Gen(\mathtt{ref}), \langle \breve{S}', I \rangle\}$.

Regułę tę zilustrujemy następującym przykładem.

Przykład *3.6.* Zdanie z przykladu 3.1 po dwukrotnym zastosowaniu reguły CR.PN będzie reprezentowane jako struktura:
$K = \langle U_K, \mathrm{Con}_K \rangle$, gdzie $U_K = \{x, y\}$ oraz $\mathrm{Con}_K = \{$**Jones** (x), **Ulysses** (y), $Num(x)$, $Gen(x)$, $Num(y)$, $Gen(y), \breve{S}'\}$. Graficzna reprezentacja tej struktury widnieje obok. Zamiast drzewa rozbioru, niejednokrotnie w notacji graficznej będziemy umieszczać (w nawiasach kwadratowych) jego liniowy, uproszczony zapis, coś na kształt quasi-zdania. Informacja o liczbie i rodzaju jako niezmienna w formie i jednocześnie cały czas dostępna pośrednio ze słownikowego opisu rzeczownika bezpośrednio reprezentowanego w strukturze jest w reprezentacji graficznej pomijana.



x, y
Jones (x)
Ulysses (y)
$[x$ owns $y]$

Reguły dla zaimków osobowych

W omawianej pracy autorzy ograniczyli się do omawiania jedynie *anaforycznej*[20] funkcji zaimków, z pominięciem ich funkcji *deiktycznej* związanej z bezpośrednim wskazywaniem na obiekt jako nierelewantnej. Zazwyczaj opis anaforycznych zaimków osobowych polega na określaniu relacji z ich *poprzednikami anaforycznymi* (ang. *anaphoric antecedents*). Kamp i Reyle wybrali inne podejście: determinują relację między zaimkami osobowymi a znacznikami dyskursu już istniejącymi w konstruowanej reprezentacji semantycznej.

Podstawowym problemem w zastosowaniu tej reguły jest znalezienie w zbiorze znaczników dyskursu właściwego *poprzednika anaforycznego* analizowanego zaimka. Zasadniczym warunkiem jest zgodność rodzaju i liczby rozważanych obiektów. Jednak w wielu wypadkach jest to dalece niewystarczające. Autorzy powołują się na przykład *Billy hit Johnny with his baseball bat. He burst into tears.*. Zarówno

[19] Jest to legalne ze względu na definicję przekształcenia $\mathfrak{g}$.

[20] Termin ten używany jest w nieco innym, szerszym znaczeniu niż w pracy [Przepiórkowski i in., 2002].

Billy, jak i *Johnny* spełniają warunek na poprzednik zaimka *he*. Który tak naprawdę się rozpłakał, zależy od naszej wiedzy ogólnej o świecie (częściej płacze uderzony), o sytuacji (może Johnny rozzłościł i sprowokował Billy'ego, a nawet wcześniej go uderzył), lub o jej bohaterach (gdy Johnny to płaksa, a Billy twardziel). Żadna z tych informacji nie stanowi części wiedzy o języku, i autorzy nie podejmują się rozstrzygania tego typu zależności.[21]

Inny ciekawy przykład stanowi ciąg zdań *John likes Oskar. Tom likes him. Sam likes him. (Even) Patric likes him.*. Wydaje się oczywiste, że wszystkie wystąpienia zaimka *him* dotyczą Oskara. Tym razem wynika to jak najbardziej z wiedzy o języku: wszystkie te zdania mają dokładnie tę samą strukturę. Podobnie jest z ciągiem zdań: *John likes Oskar. He likes Tom. He likes Sam. He likes Patric(, too).*. Wynikałoby z tego, że preferowane (choć nie obowiązkowe), jest wiązanie podmiotu z podmiotem, zaś dopełnienia z dopełnieniem. Omawiana teoria także takich nie do końca monotonicznych zależności nie uwzględnia.

Kolejną kwestię stanowi typ zaimka znajdujący się w pozycji dopełnienia — czy jest to „zwykły" zaimek osobowy (*him, her*), czy też zaimek zwrotny (ang. *reflexive — himself, herself*). W pierwszym przypadku poprzednikiem tego zaimka nie może być znacznik dyskursu reprezentujący podmiot (ani żaden element jego klasy abstrakcji), w drugim przypadku, wręcz przeciwnie, musi. Rzecz jasna problematyka związana z wiązaniem zaimków jest znacznie bardziej złożona (weźmy dla przykładu zdanie *John compare Mary to herself.*). Jednak w ramach prostej gramatyki z rozdz. 3.1 takie podejście jest wystarczające.

Ostatecznie Kamp i Reyle proponują dwie reguły **CR.PRO** (ang. *Construction Rule for Pronouns*) przewidziane do analizy podrzewa $\breve{S}^k = \langle \mathrm{NP}, \langle \mathrm{PRO}, \langle \mathtt{pro} \rangle \rangle \rangle$ reprezentującego zaimki osobowe. Podobnie jak w regule **CR.PN**, dodają one nowy znacznik dyskursu, przypisują mu liczbę i rodzaj zaimka oraz zastępują nim odpowiednie poddrzewo. Żeby jednak nie komplikować zapisu, w niniejszym opracowaniu przypadek podmiotu potraktowany został osobno. Sformułowano dlań następującą definicję:

Definicja 3.20. (reguła **CR.PRO-SUB**) Niech $\mathtt{ref} \in R$, zaś $K = \langle U_K, \mathrm{Con}_K \rangle$ będzie taką strukturą, że $\langle \breve{S}, I \rangle \in \mathrm{Con}_K$, gdzie $\breve{S} = \langle S, \breve{S}^k, \mathfrak{d}(\mathrm{VP}') \rangle$ oraz istnieje $\mathtt{pro} \in \mathrm{PRO}$ takie, że $\breve{S}^k = \langle \mathrm{NP}, \langle \mathrm{PRO}, \langle \mathtt{pro} \rangle \rangle \rangle$. Wówczas $\mathfrak{g}(K) = \langle U_{K'}, \mathrm{Con}_{K'} \rangle$ spełnia warunki:

1. $U_{K'} = U_K \cup \{\mathtt{ref}\}$,
2. $\breve{S}' = \breve{S}[\langle \mathrm{NP}, \langle \mathrm{PRO}, \langle \mathtt{pro} \rangle \rangle \rangle / \langle \mathtt{ref} \rangle]$,
3. $\mathit{Num}(\mathtt{ref}) = \mathrm{PRO}.\mathit{Num}$,
4. $\mathit{Gen}(\mathtt{ref}) = \mathrm{PRO}.\mathit{Gen}$,
5. Istnieje $\mathtt{ref}' \in \overline{U_K}$ taka, że $\mathit{Num}(\mathtt{ref}) = \mathit{Num}(\mathtt{ref}')$ i $\mathit{Gen}(\mathtt{ref}) = \mathit{Gen}(\mathtt{ref}')$,
6. $\mathrm{Con}_{K'} = \mathrm{Con}_K - \{\langle \breve{S}, I \rangle\} \cup \{\mathtt{ref} = \mathtt{ref}', \mathit{Num}(\mathtt{ref}), \mathit{Gen}(\mathtt{ref}), \langle \breve{S}', I \rangle\}$.

Wystąpienie zaimka na pozycji dopełnienia analizowane jest w momencie, gdy podmiot został już rozpatrzony i w jego miejscu w drzewie znajduje się znacznik dyskursu. Najpierw przedstawimy przypadek niezwrotnych zaimków osobowych.

Definicja 3.21. (reguła **CR.PRO-NR**) Niech $\mathtt{ref1}, \mathtt{ref2} \in R$, zaś $K = \langle U_K, \mathrm{Con}_K \rangle$ będzie taką strukturą, że $\langle \breve{S}, I \rangle \in \mathrm{Con}_K$, gdzie $\breve{S} = \langle S, \langle \mathtt{ref1} \rangle, \mathfrak{d}(\mathtt{VP}') \rangle$ oraz istnieje $\mathtt{pro} \in \mathrm{PRO}$ takie, że $\mathfrak{d}(\mathrm{VP}') \ni \breve{S}^k = \langle \mathrm{NP}, \langle \mathrm{PRO}, \langle \mathtt{pro} \rangle \rangle \rangle$. Wówczas $\mathfrak{g}(K) = \langle U_{K'}, \mathrm{Con}_{K'} \rangle$ spełnia warunki:

1. $U_{K'} = U_K \cup \{\mathtt{ref2}\}$,
2. $\breve{S}' = \breve{S}[\langle \mathrm{NP}, \langle \mathrm{PRO}, \langle \mathtt{pro} \rangle \rangle \rangle / \langle \mathtt{ref2} \rangle]$,

[21] W języku polskim analogiczny przykład nie wydaje się rozstrzygający. W zdaniach *Janek uderzył Tomka. Rozpłakał się.* uznalibyśmy raczej, że to *Janek* się rozpłakał. Przeciwnie w zdanich *Janek uderzył Tomka. Ten rozpłakał się.*. Nie zmniejsza to jednak rangi problemu w ogólności.

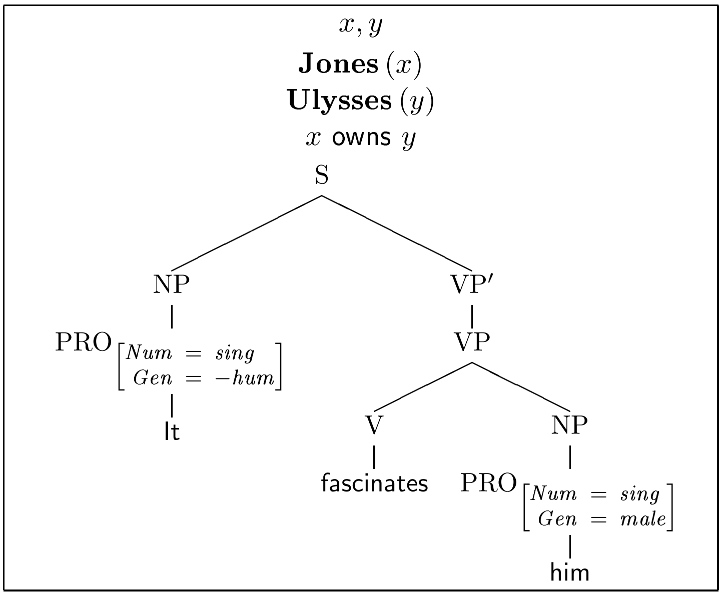
3. $Num(\texttt{ref2}) = \text{PRO}.Num$,

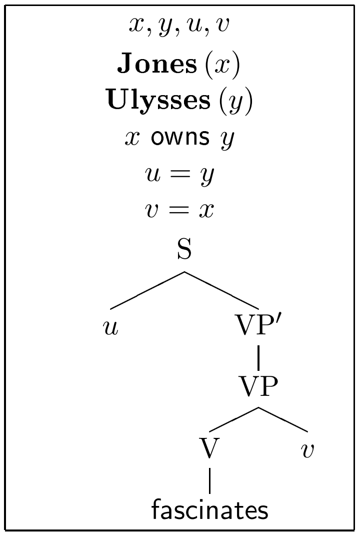
4. $Gen(\texttt{ref2}) = \text{PRO}.Gen$,

5. Istnieje $\texttt{ref3} \in \overline{U_K}$ taka, że $Num(\texttt{ref2}) = Num(\texttt{ref3})$ i $Gen(\texttt{ref2}) = Gen(\texttt{ref3})$ oraz $\texttt{ref3} \notin [\texttt{ref1}]_K$,

6. $\text{Con}_{K'} = \text{Con}_K - \{\langle \breve{S}, I\rangle\} \cup \{\texttt{ref2} = \texttt{ref3}, Num(\texttt{ref2}), Gen(\texttt{ref2}), \langle \breve{S}', I\rangle\}$.

Przykład *3.7.* Rozważmy ciąg dwóch zdań *Jones owns Ulisses. It fascinates him.*. Po zanalizowaniu pierwszego zdania reprezentować go będzie struktura $K = \langle U_K, \text{Con}_K\rangle$, gdzie $U_K = \{x, y\}$ oraz $\text{Con}_K = \{\textbf{Jones}\,(x), \textbf{Ulysses}\,(y), Num(x), Gen(x), Num(y), Gen(y), x\ \textsf{owns}\ y, S_2\}$ przedstawiona poniżej po lewej. Po zastosowaniu reguły CR.PRO-S uzyskamy strukturę $K' = \langle U_{K'}, \text{Con}_{K'}\rangle$, gdzie $U_{K'} = \{x, y, u\}$ oraz $\text{Con}_{K'} = \{\textbf{Jones}\,(x), \textbf{Ulysses}\,(y), Num(x), Gen(x), Num(y), Gen(y), x\ \textsf{owns}\ y, Num(u), Gen(u), u = y, [u\ \textsf{fascinates him}]\}$. Ponieważ $x \notin [u]_{K'}$, reguła CR.PRO-NR da nam strukturę $K''\langle U_{K''}, \text{Con}_{K''}\rangle$, przy czym $U_{K''} = \{x, y, u, v\}$ oraz $\text{Con}_{K''} = \{\textbf{Jones}\,(x), \textbf{Ulysses}\,(y), Num(x), Gen(x), Num(y), Gen(y), x\ \textsf{owns}\ y, Num(u), Gen(u), Num(v), Gen(v), u = y, v = x, S_2''\}$ widniejącą po prawej.





Zastanawiający jest fakt, dlaczego Kamp i Reyle zdecydowali się na wprowadzanie odrębnych znaczników (w powyższym przykładzie u i v) do reprezentacji zaimków osobowych, zamiast wiązać je bezpośrednio z ich poprzednikami (mielibyśmy wówczas $y\ \textsf{fascinates}\ x$). Takie podejście wydaje się jednak zbyt mocne. Dwa równe sobie znaczniki to nie jest całkiem to samo co jeden pojedynczy znacznik. Szczególnie staje się to zrozumiałe w sytuacji, gdy istnieje więcej niż jeden kandydat na poprzednik anaforyczny. Można sobie wówczas wyobrazić nawet dopuszczenie niemonotoniczności w ustalaniu poprzednika: gdyby informacje wynikające z dalszej analizy weryfikowały dotychczasowy wybór, mógłby on ulec modyfikacji.

Definicja 3.22. (reguła **CR.PRO-REFL**) Niech $\texttt{ref1}, \texttt{ref2} \in R$, zaś $K = \langle U_K, \text{Con}_K\rangle$ będzie taką strukturą, że $\langle \breve{S}, I\rangle \in \text{Con}_K$, gdzie $\breve{S} = \langle S, \langle \texttt{ref1}\rangle, \mathfrak{d}(\texttt{VP}')\rangle$ oraz istnieje $\texttt{pro} \in \text{PRO}$ takie, że $\mathfrak{d}(\text{VP}') \ni \breve{S}^k = \langle \text{NP}, \langle \text{PRO}, \langle \texttt{pro}\rangle\rangle\rangle$. Wówczas $\mathfrak{g}(K) = \langle U_{K'}, \text{Con}_{K'}\rangle$ spełnia warunki:

1. $U_{K'} = U_K \cup \{\texttt{ref2}\}$,

2. $\breve{S}' = \breve{S}[\langle \text{NP}, \langle \text{PRO}, \langle \texttt{pro}\rangle\rangle\rangle / \langle \texttt{ref2}\rangle]$,

3. $Num(\texttt{ref2}) = \text{PRO}.Num$,

4. $Gen(\texttt{ref2}) = \text{PRO}.Gen$,

5. Zachodzi $Num(\texttt{ref2}) = Num(\texttt{ref1})$ oraz $Gen(\texttt{ref2}) = Gen(\texttt{ref1})$,

6. $\text{Con}_{K'} = \text{Con}_K - \{\langle \breve{S}, I\rangle\} \cup \{\texttt{ref2} = \texttt{ref1}, Num(\texttt{ref2}), Gen(\texttt{ref2}), \langle \breve{S}', I\rangle\}$.

Jako że poprzednik anaforyczny dla zaimków zwrotnych jest ustalony jednoznacznie, wprowadzanie dodatkowego znacznika budzi wątpliwości poważniejsze niż w poprzednich przypadkach. Pamiętajmy jednak, że dotyczy to jedynie omawianej tutaj bardzo prostej gramtyki, w ogólnym przypadku wcale tak być nie musi.

Przykład 3.11 zawierający zaimek zwrotny prezentowany jest na stronie 52.

Reguła dla fraz nieokreślonych

Kolejny ważny typ fraz rzeczownikowych w języku angielskim to rzeczownik pospolity poprzedzony rodzajnikiem nieokreślonym *a(n)* reprezentowanych przez poddrzewo $\breve{S}^k = \langle \text{NP}, \langle \text{DET}, \langle \mathbf{a(n)}\rangle\rangle, \mathfrak{d}(\mathbf{N})\rangle$. Znaczenie takich fraz różni się od nazw własnych tym, że nazwa własna identyfikuje obiekt w sposób jednoznaczny, podczas gdy rzeczownik pospolity opisuje całą klasę obiektów, zaś rodzajnik nieokreślony oznacza wybranie z tej klasy pojedynczego choć dowolnego obiektu.[22]

Z praktycznego punktu widzenia istotne jest także, że frazy nieokreślone występują w dwóch rodzajach konstrukcji: samodzielnie (*Jones owns a book.*) — wówczas rzeczownik `noun` jest liściem, czyli poddrzewo $\mathfrak{d}(N) = \langle N, \texttt{noun}\rangle$, lub też wraz z *frazą/konstrukcją względną*, reprezentowaną przez poddrzewo $\mathfrak{d}(RC)$ w $\mathfrak{d}(N) = \langle N, \langle N, \texttt{noun}\rangle, \mathfrak{d}(RC)\rangle$ (*Jones owns a book which Smiths adores.*). Z tej przyczyny frazy tego typu analizowane są dwufazowo.

Pierwsza faza realizowana jest za pomocą reguły **CR.ID**: (ang. *Construction Rule for Indefinite Descriptions*). Podobnie jak poprzednie reguły dodaje ona nowy znacznik dyskursu oraz zastępuje nim odpowiednie poddrzewo. Jednak zamiast warunku wiążącego rzeczownik ze znacznikiem wprowadzane jest drugie drzewo.

Definicja 3.23. (reguła **CR.ID**) Rozważmy $\texttt{ref} \in R$, $\texttt{noun} \in \text{N}$ oraz $K = \langle U_K, \text{Con}_K\rangle$ takie, że $\langle \breve{S}, I\rangle \in \text{Con}_K$ oraz $\breve{S} \ni \breve{S}^k = \langle \text{NP}, \langle \text{DET}, \langle \mathbf{a(n)}\rangle\rangle, \mathfrak{d}(\text{N})\rangle$, gdzie $\mathfrak{d}(N) = \langle N, \langle \texttt{noun}\rangle\rangle$ lub $\mathfrak{d}(N) = \langle N, \langle N, \langle \texttt{noun}\rangle\rangle, \mathfrak{d}(RC)\rangle$. Wówczas $\mathfrak{g}(K) = \langle U_{K'}, \text{Con}_{K'}\rangle$ jest strukturą spełniającą warunki:

1. $U_{K'} = U_K \cup \{\texttt{ref}\}$,

2. $\breve{S}' = \breve{S}[\langle \text{NP}, \langle \text{DET}, \langle \mathbf{a(n)}\rangle\rangle, \mathfrak{d}(\text{N})\rangle / \langle \texttt{ref}\rangle]$,

3. $\hat{S} = \mathfrak{d}(\text{N})[\text{N}/\text{N}(\texttt{ref})]$,

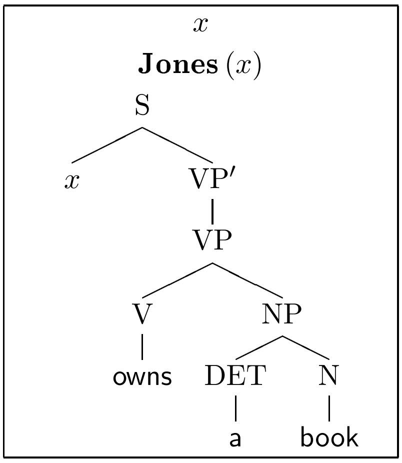
4. $Num(\texttt{ref}) = \text{N}.Num$,

5. $Gen(\texttt{ref}) = \text{N}.Gen$,

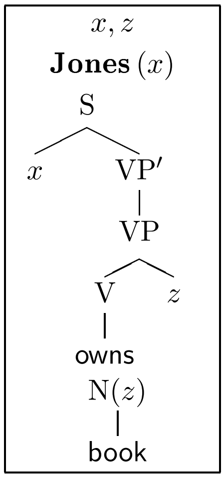
6. $\text{Con}_{K'} = \text{Con}_K - \{\langle \breve{S}, I\rangle\} \cup \{Num(\texttt{ref}), Gen(\texttt{ref}), \langle \breve{S}', I\rangle, \langle \hat{S}, I\rangle\}$.

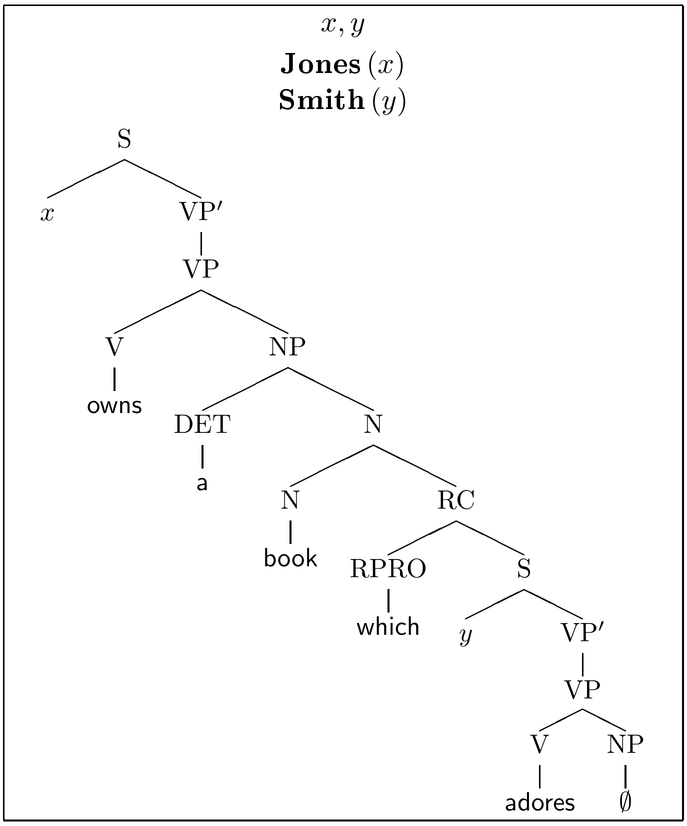
<u>Przykład</u> *3.8.* Rozważmy dwa zdania zawierające frazę nieokreśloną: *Jones owns a book.* oraz zawierające zdanie względne: *Jones owns a book which Smith adores.*.

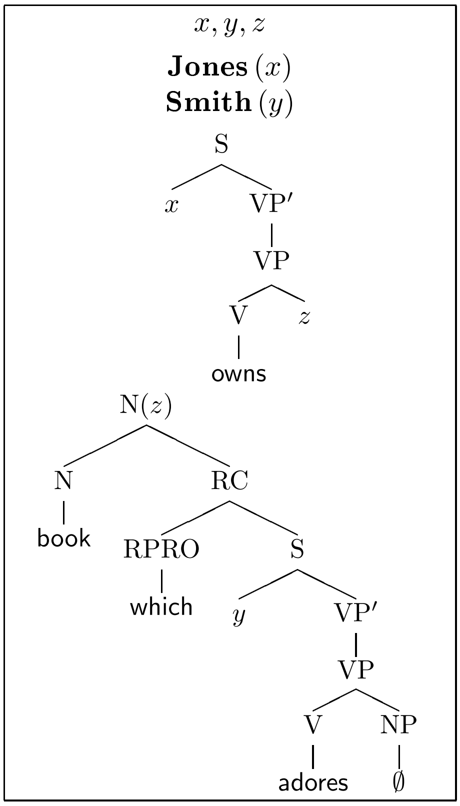
[22] To, że istnieje wielu mężczyzn o nazwisku *Jones* nie zmienia faktu, że pojawienie się kilkakrotnie w tym samym kontekście tego nazwiska zazwyczaj (czy też raczej domyślnie) oznacza, że chodzi nam o tego samego mężczyznę, podczas gdy kilkakrotne wystąpienie frazy *a man* bynajmniej do takiego wniosku nie prowadzi.



Po zastosowaniu reguły CR.PN będą one reprezentowane przez struktury K_1, K_2, gdzie $U_{K_1} = \{x\}$, $\mathrm{Con}_{K_1} = \{\mathbf{Jones}\,(x),\ Num(x),\ Gen\ (x), \check{S}^a\}$ oraz $U_{K_2} = \{x, y\}$, $\mathrm{Con}_{K_2} = \{\mathbf{Jones}\,(x),\ \mathbf{Smith}\,(y),$ $Num(x),\ Gen(x),\ Num(y),\ Gen(y),\ \check{S}^b\}$ przedstawione po lewej. Natomiast reguła CR.ID przekształci te struktury w K_1', K_2', gdzie $U_{K_1'} = \{x, z\}$, $\mathrm{Con}_{K_1'} = \{\mathbf{Jones}\,(x), Num(x), Gen(x), Num(z),$ $Gen(z), \check{S}^{a\prime}, \langle \mathrm{N}(z), \langle \mathsf{book} \rangle\rangle\}$ oraz $U_{K_2'} = \{x, y, z\}$, $\mathrm{Con}_{K_1'} = \{\mathbf{Jones}\,(x), \mathbf{Smith}\,(y), Num(x),$ $Gen(x), Num(y), Gen(y), Num(z), Gen(z),$ $\check{S}^{b\prime}, \hat{S}^b\}$ widniejące po prawej.







Reguła dla fraz względnych

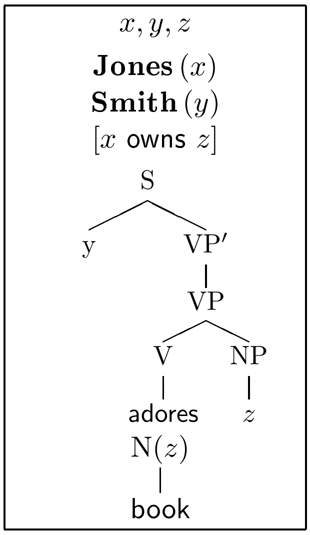
W przypadku konstrukcji względnych po zastosowaniu reguły CR.ID do przetworzenia zostaje złożone drzewo o korzeniu N($\mathtt{ref}$), a mianowicie $\mathfrak{d}(\mathrm{N}(\mathtt{ref})) = \langle \mathrm{N}, \langle \mathrm{N}, \langle \mathtt{noun} \rangle\rangle, \langle \mathrm{RC}, \langle \mathrm{RPRO}, \langle \mathtt{rpro} \rangle\rangle, \mathfrak{d}(\mathrm{S})\rangle\rangle$. Sposób przekształcania tego drzewa wydaje się oczywisty: należy powiązać znacznik $\mathtt{ref}$ zarówno z rzeczownikiem $\mathtt{noun}$, jak i odpowiadającą mu pustą frazą rzeczownikową $\langle \mathrm{N}, \langle \emptyset \rangle\rangle \in \mathfrak{d}(\mathrm{S})$. Dokonamy tego następująco:

Definicja 3.24. (reguła **CR.NRC**) Niech $K = \langle U_K, \mathrm{Con}_K \rangle$ będzie taką strukturą, że $\mathtt{ref} \in \overline{U_K}$ oraz istnieją $\mathtt{noun} \in \mathrm{N}$, $\mathtt{rpro} \in \mathrm{RPRO}$ takie, że $\mathfrak{d}(\mathrm{N}(\mathtt{ref})) = \langle \mathrm{N}, \langle \mathrm{N}, \langle \mathtt{noun} \rangle\rangle, \langle \mathrm{RC}, \langle \mathrm{RPRO}, \langle \mathtt{rpro} \rangle\rangle, \mathfrak{d}(\mathrm{S})\rangle\rangle \in$ Con_K. Wówczas $\mathfrak{g}(K) = \langle U_{K'}, \mathrm{Con}_{K'} \rangle$ spełnia warunki:

1. $U_{K'} = U_K$,
2. $\hat{S} = \mathfrak{d}(\mathrm{S})[\langle \mathrm{N}, \langle \emptyset \rangle\rangle / \mathtt{ref}]$,
3. $\mathrm{Con}_{K'} = \mathrm{Con}_K - \{\langle \mathfrak{d}(\mathrm{N}(\mathtt{ref})), I \rangle\} \cup \{\langle \mathrm{N}(\mathtt{ref}), \langle \mathtt{noun} \rangle\rangle, \langle \hat{S}, I \rangle\}$.

Przykład *3.9.* Rozważmy zdanie z poprzedniego przykładu zawierające frazę względną:

Jones owns a book which Smith adores.. Wówczas reguła CR.NRC przekształci strukturę K_2' uzyskaną uprzednio dla tego zdania za pomocą reguły CR.ID w taką strukturę K_2'', że $U_{K_2''} = \{x, y, z\}$ oraz $\mathrm{Con}_{K_1'} = \{\textbf{Jones}\,(x), \textbf{Smith}\,(y), Num(x), Gen(x), Num(y), Gen(y), Num(z), Gen(z), \check{S}^{b\prime}, \hat{\hat{S}}^{b}, \langle \mathrm{N}(z), \langle \mathsf{book} \rangle \rangle\}$. Reprezentacja graficzna struktury wynikowej prezentowana jest obok.



Reguła linearyzacji

Po zastosowniu reguły dla fraz nieokreślonych CR.ID oraz w przypadku fraz względnych także reguły CR.NRC struktura wynikowa zawiera drzewo postaci $\langle \mathrm{N}(\texttt{ref}), \langle \texttt{noun} \rangle \rangle$. Do zamiany takiego drzewa na odpowiadający mu warunek $\texttt{noun}(\texttt{ref})$ służy reguła linearyzacji **CR.LIN**.

Definicja 3.25. (Reguła **CR.LIN**) Niech $K = \langle U_K, \mathrm{Con}_K \rangle$ będzie taką strukturą, że $\texttt{ref} \in U_K$ oraz $\langle \langle \mathrm{N}(\texttt{ref}), \langle \texttt{noun} \rangle \rangle, I \rangle \in \mathrm{Con}_K$. Wówczas $\mathfrak{g}(K) = \langle U_{K'}, \mathrm{Con}_{K'} \rangle$ jest strukturą spełniającą warunki:

1. $U_{K'} = U_K$,
2. $\mathrm{Con}_{K'} = \mathrm{Con}_K - \{\langle \langle \mathrm{N}(\texttt{ref}), \langle \texttt{noun} \rangle \rangle, I \rangle\} \cup \{\texttt{noun}(\texttt{ref})\}$.

Przykład *3.10.* Rozważmy struktury:

x, z
Jones (x)
book (z)
$[x$ owns $z]$

K_1' uzyskaną w przyk. 3.8 za pomocą reguły CR.ID dla zdania *Jones owns a book.* oraz K_2'' uzyskaną w przyk. 3.9 za pomocą reguły CR.NRC dla zdania *Jones owns a book which Smith adores.*. Reguła CR.LIN przekształci je w K_1'', K_2''', gdzie $U_{K_1''} = \{x, z\}$, $\mathrm{Con}_{K_1''} = \{Num(x), Gen(x), \textbf{Jones}\,(x), Num(z), Gen(z), \textbf{book}\,(z), \check{S}^{a\prime}\}$ (po lewej) oraz $U_{K_2'''} = \{x, y, z\}$, $\mathrm{Con}_{K_2'''} = \{\textbf{Jones}\,(x), \textbf{Smith}\,(y), Num(x), Gen(x), Num(y), Gen(y), Num(z), Gen(z), \textbf{book}\,(z), \check{S}^{b\prime}, \hat{\hat{S}}^{b}\}$ (po prawej).

x, y, z
Jones (x)
Smith (y)
book (z)
$[x$ owns $z]$
[y adores z]

Przedstawimy jeszcze przykład zawierający zarówno frazę względną, jak i zaimek zwrotny.

Przykład *3.11.* Weźmy zdanie *Mary employs a man who admires himself.*. Po zastosowaniu reguł CR.PN, CR.ID, CR.LIN uzyskamy strukturę $K = \langle U_K, \mathrm{Con}_K \rangle$, gdzie $U_K = \{x, y\}$ oraz $\mathrm{Con}_K = \{Num(x), Gen(x), \textbf{Mary}\,(x), [x \text{ employs } y], \textbf{man}\,(y), Num(y), Gen(y), [y \text{ admires himself}]\}$ (po lewej).

x, y
Mary (x)
$[x$ employs $y]$
man (y)
$[y$ admires himself$]$

Ponieważ zachodzi zgodność rodzaju i liczby między zaimkiem zwrotnym *himself* oraz reprezentującym nazwę własną znacznikiem x, aplikacja reguły CR.PRO-REFL daje strukturę $K' = \langle U_{K'}, \mathrm{Con}_{K'} \rangle$, gdzie $U_{K'} = \{x, y, u\}$ oraz $\mathrm{Con}_{K'} = \{\textbf{Mary}\,(x), Num(x), Gen(x), [x \text{ employs } y], \textbf{man}\,(y), Num(y), Gen(y), Num(u), Gen(u), u = y, [y \text{ admires } u]\}$ (po prawej). Natomiast zdanie *Mary employs a man who admires herself.* zostanie odrzucone z powodu niezgodności rodzaju, gdyż podmiotem zdania w momencie stosowania reguły CR.PRO-REFL jest znacznik y mający przypisany rodzaj męski.

x, y, u
Jones (x)
$[x$ employs $y]$
woman (y)
$u = y$
$[y$ admires $u]$

Reguła dla fraz określonych

Podobnie jak frazy nieokreślone w języku angielskim stanowi rzeczownik pospolity poprzedzony rodzajnikiem nieokreślonym *a(n)*, *frazy określone* formowane są przez rzeczownik pospolity poprzedzony rodzajnikiem określonym *the* i reprezentowane przez poddrzewo $\breve{S}^k = \langle \mathrm{NP}, \langle \mathrm{DET}, \langle \mathbf{the} \rangle\rangle, \mathfrak{d}(\mathbf{N})\rangle$. Jednak przeciwnie do fraz nieokreślnych i w pewnym stopniu podobnie do nazw własnych fraza taka jednoznacznie identyfikuje obiekt. Wszelako również w tym wypadku jednoznaczność takiej identyfikacji to ewidentne uproszczenie. Niejednokrotnie wynika ona z wiedzy o świecie, np. w zdaniu *Jones lives in the capital of France.* użycie rodzajnika **the** uzasadnione jest faktem, że każde państwo (przynajmniej w danym czasie) ma jedną stolicę. Z kolei w zdaniu *Mary went to the bank.* zakładamy, że chodzi o najbliższy bank lub też taki, z którego usług Mary korzysta najczęściej. Często jednak identyfikacja taka, podobnie jak dla zaimków, jest czysto kontekstowa, np. w ciągu zdań *A man and a boy were walking in a park. The man was wearing a brown overcoat.* chodzi o tego samego mężczyznę.[23] Zauważmy przy tym, że nie wystarczy tu zgodność rodzaju i liczby, więc użycie zaimka *he* w miejsce frazy *the man* nie byłoby zadowalające. Często również w takich sformułowaniach niezbędna jest wiedza o świecie (np. interpretacja zdań *Mary ate strawberries and peanuts. The fruits were very tasty.* wymaga wiedzy, że truskawki są owocami).

Nie decydując się na rozstrzygnięcie tych kwestii, Kamp i Reyle proponują następującą wstępną wersję reguły obsługującej frazy określone CR.DD.

Definicja 3.26. (reguła **CR.ID**) Rozważmy $\mathtt{ref} \in R$, $\mathtt{noun} \in \mathrm{N}$ oraz $K = \langle U_K, \mathrm{Con}_K \rangle$ takie, że $\langle \breve{S}, I\rangle \in \mathrm{Con}_K$ oraz $\breve{S} \ni \breve{S}^k = \langle \mathrm{NP}, \langle \mathrm{DET}, \langle \mathbf{the}\rangle\rangle, \langle \mathrm{N}, \langle \mathtt{noun}\rangle\rangle\rangle$. Wówczas $\mathfrak{g}(K) = \langle U_{K'}, \mathrm{Con}_{K'}\rangle$ jest strukturą spełniającą warunki:

1. $U_{K'} = U_K \cup \{\mathtt{ref}\}$,
2. $\breve{S}' = \breve{S}[\breve{S}^k / \langle \mathtt{ref}\rangle]$,
3. $\mathit{Num}(\mathtt{ref}) = \mathrm{N}.\mathit{Num}$,
4. $\mathit{Gen}(\mathtt{ref}) = \mathrm{N}.\mathit{Gen}$,
5. $\mathrm{Con}_{K'} = \mathrm{Con}_K - \{\langle \breve{S}, I\rangle\} \cup \{\mathbf{the}\ \mathtt{noun}(\mathtt{ref}), \mathit{Num}(\mathtt{ref}), \mathit{Gen}(\mathtt{ref}), \langle \breve{S}', I\rangle\}$.

Reguła dla fraz dzierżawczych

Istotne związki pomiędzy frazami rzeczownikowymi tworzą zależności *dzierżawcze*. W języku angielskim są one formowane przez „dzierżawcze” **’s**. Zależność taka może być interpretowana jako zależność posiadania czegoś przez kogoś (*Bill’s donkey*), choć nawet w tym konkretnym wypadku może chodzić o ulubionego osiołka Billa, nawet gdy ten nie jest jego właścicielem. Inne typowe wystąpienie takiej zależności to opis relacji wiążącej dwoje ludzi (*Bill’s father*). Kamp i Reyle nie decydują się na wnikanie w rzeczywisty charakter zależności dzierżawczej. Wprowadzają jedynie nowy znacznik reprezentujący obiekt opisywany przez całą frazę i zastępują nim całe odpowiadające jej poddrzewo, które z kolei umieszczane jest w strukturze osobno (podobnie jak w regule dla fraz nieokreślonych CR.ID).

Definicja 3.27. (reguła **CR.NP’sN**) Rozważmy $\mathtt{ref} \in R$, oraz $K = \langle U_K, \mathrm{Con}_K \rangle$ takie, że $\langle \breve{S}, I\rangle \in \mathrm{Con}_K$ oraz $\breve{S} \ni \breve{S}^k = \langle \mathrm{NP}, \langle \mathrm{DET}, \mathfrak{d}(\mathrm{NP}), \langle \mathbf{'s}\rangle\rangle, \mathfrak{d}(\mathrm{N})\rangle$. Wówczas $\mathfrak{g}(K) = \langle U_{K'}, \mathrm{Con}_{K'}\rangle$ jest strukturą spełniającą warunki:

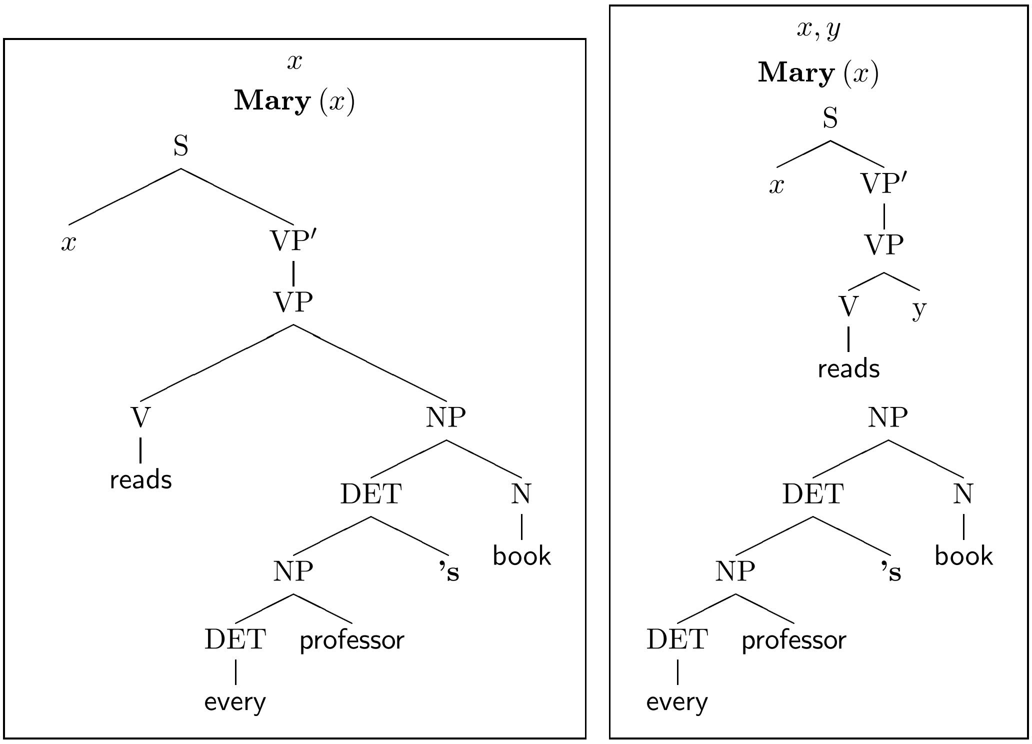
[23] W języku polskim nie ma rodzajników. Jednak w podobnych sytuacjach rozróżnienie między *określonym* a *nieokreślonym* wystąpieniem rzeczownika można uzyskać za pomocą szyku. Dla przykładu porównaj *Piotr i Ewa wysiedli z samochodu. Mężczyzna wszedł do sklepu.* z podobnym *Piotr i Ewa wysiedli z samochodu. Do sklepu wszedł mężczyzna.*. W pierwszym wypadku mężczyzną jest Piotr, w drugim ktoś inny. Zjawisko to nie dotyczy nazw własnych. Zmianę szyku w zdaniu *Do sklepu wszedł Piotr.* uznalibyśmy jedynie za podkreślenie, że Ewa pozostała na zewnątrz.

1. $U_{K'} = U_K \cup \{\mathtt{ref}\}$,
2. $\breve{S}' = \breve{S}[\breve{S}^k/\langle\mathtt{ref}\rangle]$,
3. $\hat{S} = \breve{S}^k[\mathrm{N}/\mathrm{N}(\mathtt{ref})]$,
4. $Num(\mathtt{ref}) = \mathrm{N}.Num$,
5. $Gen(\mathtt{ref}) = \mathrm{N}.Gen$,
6. $\mathrm{Con}_{K'} = \mathrm{Con}_K - \{\langle\breve{S}, I\rangle\} \cup \{Num(\mathtt{ref}), Gen(\mathtt{ref}), \langle\breve{S}', I\rangle, \langle\hat{S}, I\rangle\}$.

Do drzewa $\mathfrak{d}(NP) \in \hat{S}$ stosuje się następnie reguły omówione powyżej, jak dla każdej innej frazy rzeczownikowej.

Przykład 3.25 ze strony 61 zawiera użycie omawianej reguły.

Przykład *3.12.* Weźmy zdanie *Mary reads every professor's book.*. Po zastosowaniu reguły CR.PN będzie ono reprezentowane przez strukturę K taką, że $U_K = \{x\}$, $\mathrm{Con}_K = \{\textbf{Mary}\,(x), \breve{S}\}$. Wówczas reguła CR.NP'sNP da nam strukturę K' taką, że $U_{K'} = \{x, y\}$, $\mathrm{Con}_K = \{\textbf{Mary}\,(x), \breve{S}', \hat{S}\}$.



3.4.2 Negacja

Rozważane dotychczas reguły dotyczyły reprezentacji zdań oznajmujących, zawierających informację, że pewne zależności między obiektami mają miejsce (są prawdziwe). W komunikacji międzyludzkiej ważną rolę pełnią także zdania przeczące, niosące ze sobą informację, że, wręcz przeciwnie, pewne zależności między obiektami nie zachodzą (są fałszywe). Języki naturalne (w tym angielski) dysponują wieloma takimi negatywnymi sformułowaniami. Podstawowe z nich to zaprzeczenie orzeczenia, w naszym wypadku poprzedzenie frazy czasownikowej VP partykułą **not** w połączeniu z czasownikiem pomocniczym **do**. Zdania takie traktowane są jako pewne stwierdzenie φ będące negacją innego stwierdzenia ψ, a więc zawartość φ jest funkcją ψ i zależność ta jest wyrażana przez symbol negacji $\neg$.

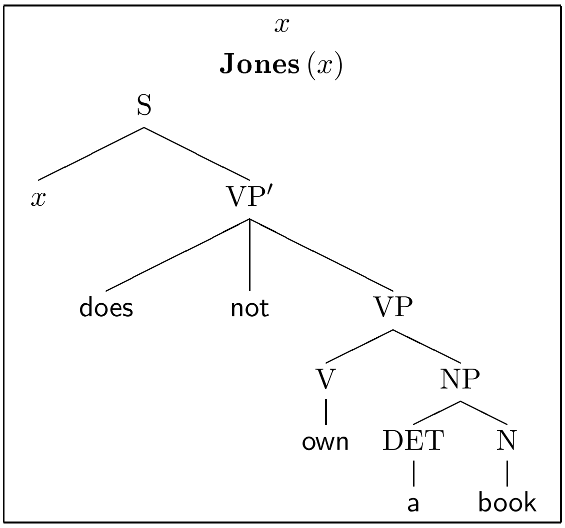
Reguła negacji

W przeciwieństwie do reguł przetwarzających frazy rzeczownikowe omówionych powyżej, niniejsza reguła **CR.NEG** zajmuje się całym zdaniem (choć nie musi to być zdanie główne), tzn. drzewem o wierzchołku S. Z drzewa tego usuwana jest informacja o negacji, a tak przekształcone drzewo wstawiane jest do tworzonej w tym celu zanegowanej podstruktury, co umożliwia dalszą jego obróbkę. Ponadto reguła wymaga, by fraza rzeczownikowa stanowiąca podmiot zdania była już całkowicie przetworzona. Innymi słowy, negacja nie obejmuje podmiotu. Uzasadnieniem tej decyzji jest ewidentnie niesprzeczny ciąg zdań: *Somebody owns a Porsche. Somebody does not own a Porsche.* — ktoś ma, ktoś inny nie.

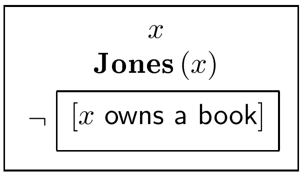
Definicja 3.28. (reguła **CR.NEG**) Niech $K = \langle U_K, \mathrm{Con}_K\rangle$ będzie taką strukturą, że $\mathtt{ref} \in U_K$ oraz $\langle \breve{S}, I\rangle \in \mathrm{Con}_K$ i $\breve{S} = \langle \mathrm{S}, \langle \mathtt{ref}\rangle, \langle \mathrm{VP}', \langle \mathrm{AUX}, \langle \mathbf{does}\rangle\rangle, \langle \mathbf{not}\rangle, \mathfrak{d}(\mathrm{VP})\rangle\rangle$. Wówczas $\mathfrak{g}(K) = \langle U_{K'}, \mathrm{Con}_{K'}\rangle$ spełnia warunki:

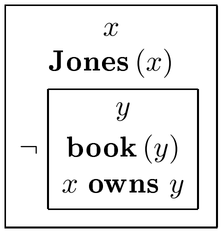
1. $U_{K'} = U_K$,
2. $L = \langle U_L, \mathrm{Con}_L\rangle$ jest taką strukturą, że $U_L = \emptyset$, zaś $\mathrm{Con}_L = \{\langle\langle \mathrm{S}, \langle \mathtt{ref}\rangle, \langle \mathrm{VP}', \mathfrak{d}(\mathrm{VP})\rangle\rangle, \emptyset\rangle\}$,
3. $\mathrm{Con}_{K'} = \mathrm{Con}_K - \{\langle \breve{S}, \emptyset\rangle\} \cup \{\neg L\}$.

Przykład *3.13.* Rozważmy zdanie przeczące *Jones does not own a book.*. Po zastosowaniu reguły CR.PN będzie ono reprezentowane przez strukturę K, gdzie $U_K = \{x\}$, $\mathrm{Con}_K = \{\mathbf{Jones}\,(x), \breve{S}\}$[24]



przedstawioną graficznie po lewej. Wówczas reguła CR.NEG przekształci tę strukturę w K' taką, że $U_{K'} = \{x\}$, $\mathrm{Con}_{K'} = \{\mathbf{Jones}\,(x), \neg\langle\emptyset, \{\breve{S}'\}\rangle\}$, której reprezentacja graficzna widnieje poniżej po lewej. Drzewo $\breve{S}'$ wyrysowane jest w przykładzie 3.8 u góry po lewej. Po całkowitym przetworzeniu uzyskujemy strukturę K'' taką, że $U_{K''} = \{x\}$, $\mathrm{Con}_{K''} = \{\mathbf{Jones}\,(x), \neg\langle\{y\}, \{\mathbf{book}\,(y), x\ \mathbf{owns}\ y\}\rangle\}$ (u dołu po prawej).





Istotny w zrozumieniu tego przykładu (i działania reguły CR.NEG w ogólności) jest sposób weryfikacji warunku $\neg L$ w modelu (por. def. 3.9). Ponieważ dowolny obiekt będący książką weryfikuje podstrukturę L i w rezultacie falsyfikuje warunek $\neg L$, zdanie uznane zostanie za prawdziwe, gdy **Jones** nie posiada żadnej książki w ogóle.

Zauważmy przy tym, że zastosowanie reguł CR.ID (wraz z CR.LIN) przed regułą CR.NEG dałoby w rezultacie całkiem odmienną interpretację tego zdania (nie miałoby to znaczenia, gdyby obie frazy rzeczownikowe zawierały nazwy własne, np. *John does not love Mary.*). Jest to pierwszy przykład sytuacji, w której kolejność stosowania reguł ma wpływ na reprezentację i interpretację znaczenia dyskursu.

Zjawisko negacji posiada ciekawe własności w połączeniu z innymi omawianymi już zjawiskami lingwistycznymi, na przykład zaimkami osobowymi.

[24] W strukturze tej pominęliśmy „termy" $Num(x)$, $Gen(x)$ jako nieistotne. Będą one pomijane w większości przykładów, chyba że pełnią w nich jakąś istotną rolę.

Przykład *3.14.* Rozważmy ciąg dwóch zdań, z których pierwsze jest oznajmujące, a drugie przeczące: *Jones owns a book. He does not like it.*. Po pełnym przeanalizowaniu pierwszego zdania oraz zastosowaniu reguły CR.PRO do podmiotu drugiego z nich uzyskujemy strukturę K, gdzie $U_K = \{x, y, u\}$, $\mathrm{Con}_K = \{\textbf{Jones}\,(x), \mathit{Num}(x), \mathit{Gen}(x), \mathit{Num}(y), \mathit{Gen}(y), \textbf{book}\,(y), x\ \textbf{own}\ y, \mathit{Num}(u), \mathit{Gen}(u), u = x, \breve{S}\}$ zaprezentowaną po lewej. Reguła CR.NEG przekształci ją w strukturę K' taką, że $U_{K'} = \{x, y, u\}$, $\mathrm{Con}_{K'} = \{\textbf{Jones}\,(x), \mathit{Num}(x), \mathit{Gen}(x), \mathit{Num}(y), \mathit{Gen}(y), \textbf{book}\,(y), x\ \textbf{own}\ y, \mathit{Num}(u), \mathit{Gen}(u), u = x, \neg\langle\emptyset, \{\breve{S}'\}\rangle\}$, której reprezentacja graficzna widnieje pośrodku. Po kolejnym użyciu reguły CR.PRO i dalszych przekształceniach uzyskamy docelową strukturę K'' zawierającą $U_{K''} = \{x, y, u\}$ oraz $\mathrm{Con}_{K''} = \{\textbf{Jones}\,(x), \mathit{Num}(x), \mathit{Gen}(x), \mathit{Num}(y), \mathit{Gen}(y), \textbf{book}\,(y), x\ \textbf{own}\ y, \mathit{Num}(u), \mathit{Gen}(u), u = x, \neg\langle\{v\}, \{v = y, u\ \mathsf{like}\ v\}\rangle\}$ ukazaną po prawej.

$$\boxed{\begin{array}{c} x, y, u \\ \textbf{Jones}\,(x) \\ \textbf{book}\,(y) \\ x\ \mathsf{owns}\ y \\ u = x \\ [u\ \mathsf{does\ not\ like\ it}] \end{array}} \qquad \boxed{\begin{array}{c} x, y, u \\ \textbf{Jones}\,(x) \\ \textbf{book}\,(y) \\ x\ \mathsf{owns}\ y \\ u = x \\ \neg\ \boxed{[u\ \mathsf{likes\ it}]} \end{array}} \qquad \boxed{\begin{array}{c} x, y, u \\ \textbf{Jones}\,(x) \\ \textbf{book}\,(y) \\ x\ \mathsf{owns}\ y \\ u = x \\ \neg\ \boxed{\begin{array}{c} v \\ v = y \\ u\ \textbf{likes}\ v \end{array}} \end{array}}$$

Przykład *3.15.* Rozważmy podobny do poprzedniego ciąg dwóch zdań, z których jednak pierwsze jest przeczące, a drugie oznajmujące: *Jones does not own a book. He likes it.*.
Znowu, po przeanalizowaniu pierwszego zdania (jak w przykładzie 3.13) oraz zastosowaniu reguły CR.PRO do podmiotu drugiego z nich uzyskujemy strukturę K, gdzie $U_K = \{x, u\}$, $\mathrm{Con}_K = \{\textbf{Jones}\,(x), \mathit{Num}(x), \mathit{Gen}(x), \neg\langle\{y\}, \{\textbf{book}\,(y), x\ \textbf{owns}\ y\}\rangle, \mathit{Num}(u), \mathit{Gen}(u), u = x, \breve{S}\}$ zaprezentowaną po lewej. Jednak nie możemy zastosować reguły CR.PRO do zaimka *it* będącego dopełnieniem w drzewie $\breve{S}$, gdyż jedyny „pasujący” znacznik dyskursu v nie jest widoczny na poziomie głównej struktury K. Tak więc nie istnieje reguła umożliwiająca redukcję drzewa $\breve{S}$ ze struktury K, a więc nie ma sposobu na przekształcenie jej na „zwykłą” strukturę, co oznacza brak interpretacji dla tego ciągu zdań.

$$\boxed{\begin{array}{c} x, u \\ \textbf{Jones}\,(x) \\ \neg\ \boxed{\begin{array}{c} y \\ \textbf{book}\,(y) \\ x\ \textbf{owns}\ y \end{array}} \\ u = x \\ [u\ \mathsf{does\ not\ like\ it}] \end{array}}$$

Przykład *3.16.* Rozważmy jeszcze jeden ciąg zdań tego typu *Jones does not own Ulisses. He likes it.*.

$$\boxed{\begin{array}{c} x, y \\ \textbf{Jones}\,(x) \\ \neg\ \boxed{\begin{array}{c} \textbf{Ulisses}\,(y) \\ x\ \textbf{owns}\ y \end{array}} \end{array}}$$

Struktura K uzyskana po przeanalizowaniu pierwszego zdania jest dość podobna do poprzednich, mamy $U_K = \{x, y\}$, $\mathrm{Con}_K = \{\textbf{Jones}\,(x), \neg\langle\{y\}, \{\textbf{Ulisses}\,(y), x\ \textbf{owns}\ y\}\rangle\}$ (po lewej). Ponieważ jednak zgodnie z regułą CR.PN oba x i y znajdują się w U_K, nie ma problemu w znalezieniu poprzedników zarówno dla zaimka *he*, jak i *it* przy użyciu reguły CR.PRO, co daje strukturę K' taką, że $U_{K'} = \{x, y, u, v\}$, $\mathrm{Con}_{K'} = \{\textbf{Jones}\,(x), \neg\langle\{y\}, \{\textbf{Ulisses}\,(y), x\ \textbf{owns}\ y\}\rangle, u = x, v = y, u\ \textbf{likes}\ v\}$ (po prawej).

$$\boxed{\begin{array}{c} x, y, u, v \\ \textbf{Jones}\,(x) \\ \neg\ \boxed{\begin{array}{c} \textbf{Ulisses}\,(y) \\ x\ \textbf{owns}\ y \end{array}} \\ u = x \\ v = y \\ x\ \textbf{borrows}\ y \end{array}}$$

3.4.3 Implikacja

Zdania rozważane dotychchczas, zarówno oznajmujące, jak i przeczące, miały charakter kumulatywny, czy też z logicznego punktu widzenia monotoniczny. Po dodaniu każdego kolejnego zdania S_i struktura K_i zawierała więcej informacji, a więc była prawdziwa w węższej klasie modeli. Jednak jeśli dwa zdania zawierają jedno warunek, a drugie konkluzję zależności warunkowej (jak w zdaniach *Suppose Jones owns a book on semantics. Then he uses it.*), taka własność nie zachodzi. Chociaż bowiem mamy tu do czynienia z ciągiem złożonym z dwóch zdań, zawiera on jednak pojedyncze, spójne stwierdzenie. Dodajmy, że zarówno warunek, jak i konkluzja mogą składać się z więcej niż jednego zdania; co więcej, mogą być rozdzielone innymi, „zwykłymi”, oznajmującymi zdaniami. Wówczas czasami trudno stwierdzić, które zdania należą do konkluzji stwierdzenia warunkowego, a które są niezależnymi stwierdzeniami, gdyż często nie zależy to bezpośrednio od struktury zdań, lecz od znaczenia poszczególnych wyrazów i związków między nimi. Problematyka ta jest wciąż daleka od rozwiązania, i nie wchodzi ona w zakres omawianej pracy.

Reguła dla zdań warunkowych

Ostatecznie Kamp i Reyle ograniczają się do zwrotów warunkowych funkcjonujących w ramach pojedynczego zdania reprezentowanych przez regułę PS 15 $S \longrightarrow$ **if** S **then** S. Tak więc autorzy pomijają także konstrukcje typu S_1 if S_2. Wynika z tego, że zdanie warunkowe nie może pojawić się wewnątrz frazy względnej, gdyż sformułowanie:

Jones owns a book which he reads if he has time.

jest całkowicie poprawne, gdy tymczasem zdanie:

Jones owns a book which if Smith has time he reads.

nie jest w j. angielskim akceptowane.

Zauważmy, że w związku z powyższymi ograniczeniami powracamy do monotonicznego charakteru tworzonej reprezentacji.

Zdania warunkowe będą reprezentowane za pomocą struktur zawierających implikację. Jak łatwo się domyślić, warunek będzie reprezentowany przez poprzednik, zaś konkluzja przez następnik implikacji.

Definicja 3.29. (reguła **CR.COND**) Niech $K = \langle U_K, \mathrm{Con}_K\rangle$ będzie taką strukturą, że $\langle\langle S, \langle \mathbf{if}\rangle, \mathfrak{d}(S)_1, \langle \mathbf{then}\rangle, \mathfrak{d}(S)_2\rangle, I\rangle \in \mathrm{Con}_K$. Wówczas $\mathfrak{g}(K) = \langle U_{K'}, \mathrm{Con}_{K'}\rangle$ jest strukturą spełniającą warunki:

1. $U_{K'} = U_K$,
2. $L_1 = \langle\emptyset, \{\langle\mathfrak{d}(S)_1, I\rangle\}\rangle$,
3. $L_2 = \langle\emptyset, \{\langle\mathfrak{d}(S)_2, I\rangle\}\rangle$,
4. $\mathrm{Con}_{K'} = \mathrm{Con}_K - \{\langle\langle S, \langle \mathbf{if}\rangle, \mathfrak{d}(S)_1, \langle \mathbf{then}\rangle, \mathfrak{d}(S)_2\rangle, I\rangle\} \cup \{L_1 \Longrightarrow L_2\}$.

Przykład *3.17.* Rozważmy proste zdanie warunkowe: *If Jones owns a book, he reads it.*. Po zastosowaniu reguły CR.COND uzyskujemy strukturę K, gdzie $U_K = \emptyset$ oraz $\mathrm{Con}_K = \{L_1 \Longrightarrow L_2\}$, przy czym $L_1 = \langle\emptyset, \{[\text{Jones owns a book}]\}\rangle$, zaś $L_2 = \langle\emptyset, \{[\text{he reads it}]\}\rangle$ przedstawioną obok.

$$\boxed{\boxed{[\text{Jones owns a book}]} \Longrightarrow \boxed{[\text{he reads it}]}}$$

Po przeanalizowaniu struktury L_1 (reguły CR.PN, CR.ID, CR.LIN) uzyskujemy strukturę K', w której $U_{K'} = \{x\}$, zaś strukturę L_1 zastąpiono strukturą $L_1' = \langle\{y\}, \{\mathbf{Jones}\,(x), \mathbf{book}\,(y), x\ \mathbf{owns}\ y\}\rangle$ ukazaną po lewej. Ponieważ zgodnie z regułą CR.PRO (i zasadą dostępności znaczników dyskursu z def. 3.3) zarówno x, jak i y widoczne są z L_2, po dwukrotnym zastosowaniu tej reguły uzyskujemy strukturę K'', w której struktura L_2 została zastąpiona strukturą $L_2'' = \langle\{u, v\}, \{u = x, v = y, u\ \mathbf{reads}\ v\}\rangle$ widoczną po prawej.

$$\boxed{\begin{array}{c} x \\ \boxed{\begin{array}{c} y \\ \mathbf{Jones}\,(x) \\ \mathbf{book}\,(y) \\ x\ \mathbf{owns}\ y \end{array}} \Longrightarrow \boxed{[\text{he reads it}]} \end{array}} \qquad \boxed{\begin{array}{c} x \\ \boxed{\begin{array}{c} y \\ \mathbf{Jones}\,(x) \\ \mathbf{book}\,(y) \\ x\ \mathbf{owns}\ y \end{array}} \Longrightarrow \boxed{\begin{array}{c} u, v \\ u = x \\ v = y \\ u\ \mathbf{reads}\ v \end{array}} \end{array}}$$

Przykład *3.18.* Rozważmy inne, pozornie podobne zdanie warunkowe: *If he owns it, Jones reads Ulisses.*. Po zastosowaniu reguły CR.COND uzyskujemy analogiczną strukturę K, gdzie $U_K = \emptyset$ oraz $\mathrm{Con}_K = \{L_1 \Longrightarrow L_2\}$, przy czym $L_1 = \langle\emptyset, \{[\text{he owns it}]\}\rangle$, zaś $L_2 = \langle\emptyset, \{[\text{Jones reads Ulisses}]\}\rangle$. Ponieważ jednak nie możemy zastosować reguły CR.PRO nie dysponując żadnymi dostępnymi znacznikami dyskursu, jedyną regułą, jaką można teraz zastosować jest CR.PN. Po jej dwukrotnym wykorzystaniu uzyskujemy strukturę K'', w której $U_{K'} = \{x, y\}$, zaś L_2 zastąpiono strukturą $L_2' = \{\emptyset, \{\mathbf{Jones}\,(x), \mathbf{book}\,(y), x\ \mathbf{reads}\ y\}\}$ zaprezentowaną po lewej. W tym momencie oba znaczniki x, y są dostępne, więc dwukrotne zastosowanie reguły CR.PRO prowadzi do uzyskania struktury K'', w której struktura L_1 została zastąpiona strukturą $L_1'' = \{\{u, v\}, \{u = x, v = y, u\ \mathbf{owns}\ v\}\}$ widoczną po prawej.

$$\boxed{\begin{array}{c} x, y \\ \boxed{\text{[he owns it]}} \quad \Longrightarrow \quad \boxed{\begin{array}{c}\mathbf{Jones}\,(x)\\ \mathbf{Ulisses}\,(y)\\ x\ \mathbf{reads}\ y\end{array}} \end{array}} \qquad \boxed{\begin{array}{c} x, y \\ \boxed{\begin{array}{c} u, v \\ u = x \\ v = y \\ u\ \mathbf{owns}\ v \end{array}} \quad \Longrightarrow \quad \boxed{\begin{array}{c}\mathbf{Jones}\,(x)\\ \mathbf{Ulisses}\,(y)\\ x\ \mathbf{reads}\ y\end{array}} \end{array}}$$

Podobna sytuacja wystąpi w przypadku zdań *If Jones owns it, he reads Ulisses.* oraz *If he owns a book, Jones reads it.*. Natomiast zdanie *If Jones owns it, he reads a book.* nie zostanie zanalizowane, ponieważ znacznik dyskursu y będzie umieszczony wraz z warunkiem $\mathbf{book}\,(y)$ w strukturze L_2 zgodnie z regułą CR.ID, więc stosownie do def. 3.3 nie będzie widoczny z poziomu struktury L_1, gdyż znaczniki z poprzednika implikacji są widoczne z następnika (i jego podrzędników), ale nie odwrotnie.

Definicja 3.3 udostępnia następnikowi implikacji znaczniki występujące także w strukturach podrzędnych jego poprzednika. Uzasadnia to następujący przykład:

Przykład *3.19.* Rozważmy proste zdanie warunkowe: *If Jones does not own a book, he borrows it.*. Po zastosowaniu reguły CR.COND uzyskujemy podobną do poprzednich strukturę K, przy czym $U_K = \emptyset$ oraz $\mathrm{Con}_K = \{L_1 \Longrightarrow L_2\}$, gdzie $L_1 = \langle \emptyset, \{[\text{Jones does not own a book}]\}\rangle$, zaś $L_2 = \langle \emptyset, \{[\text{he borrows it}]\}\rangle$. Po przeanalizowaniu struktury L_1 (reguły CR.PN, CR.NEG CR.ID, CR.LIN) uzyskujemy strukturę K', w której $U_{K'} = \{x\}$, zaś strukturę L_1 zastąpiono strukturą $L_1' = \langle \emptyset, \{\mathbf{Jones}\,(x), \neg\langle\{y\}, \{\mathbf{book}\,(y), x\ \mathbf{owns}\ y\}\}\rangle\rangle$ zaprezentowaną po lewej. Ponieważ dzięki def. 3.3 zarówno x, jak i y widoczne są z L_2 po dwukrotnym zastosowaniu reguły CR.PRO uzyskujemy strukturę K'', w której struktura L_2 została zastąpiona strukturą $L_2'' = \langle\{u, v\}, \{u = x, v = y, u\ \mathbf{borrows}\ v\}\rangle$ widoczną po prawej.

$$\boxed{\begin{array}{c} x \\ \boxed{\begin{array}{c}\mathbf{Jones}\,(x)\\ \neg \boxed{\begin{array}{c} y \\ \mathbf{book}\,(y) \\ x\ \mathbf{owns}\ y\end{array}}\end{array}} \quad \Longrightarrow \quad \boxed{\text{[he borrows it]}} \end{array}} \qquad \boxed{\begin{array}{c} x \\ \boxed{\begin{array}{c}\mathbf{Jones}\,(x)\\ \neg \boxed{\begin{array}{c} y \\ \mathbf{book}\,(y) \\ x\ \mathbf{owns}\ y\end{array}}\end{array}} \quad \Longrightarrow \quad \boxed{\begin{array}{c} u, v \\ u = x \\ v = y \\ u\ \mathbf{borrows}\ v\end{array}} \end{array}}$$

We wszystkich poprzednich przykładach podmiot (agent) był określany przez nazwę własną. Możliwe są jednak zdania, w których amy wyłącznie do czynienia z frazami nieokreślonymi.

Przykład *3.20.* Rozważmy kolejne zdanie warunkowe: *If a farmer owns a donkey, then he beats it.*. Po zastosowaniu reguły CR.COND i przeanalizowaniu pierwszego zdania uzyskamy strukturę K, podobną do K' z przykładu 3.17, z tym że zgodnie z regułą CR.ID znacznik x znajduje się w podstrukturze: $U_K = \emptyset$ oraz $\mathrm{Con}_K = \{L_1 \Longrightarrow L_2\}$, przy czym $L_1 = \langle\{x, y\}, \{\mathbf{farmer}\,(x), \mathbf{donkey}\,(y), x\ \mathbf{owns}\ y\}\rangle$ zaś $L_2 = \langle\emptyset, \{[\text{he beats it}]\}\rangle$. Jednak dostępność znaczników dyskursu wedle def. 3.3 jest analogiczna, i po dwukrotnym zastosowaniu reguły CR.PRO uzyskujemy strukturę K', w której struktura L_2 została zastąpiona strukturą $L_2' = \{\{u, v\}, \{u = x, v = y, u\ \mathbf{beats}\ v\}\}$ widoczną obok.

$$\boxed{\boxed{\begin{array}{c} x, y \\ \mathbf{farmer}\,(x) \\ \mathbf{donkey}\,(y) \\ x\ \mathbf{owns}\ y\end{array}} \quad \Longrightarrow \quad \boxed{\begin{array}{c} u, v \\ u = x \\ v = y \\ u\ \mathbf{beats}\ v\end{array}}}$$

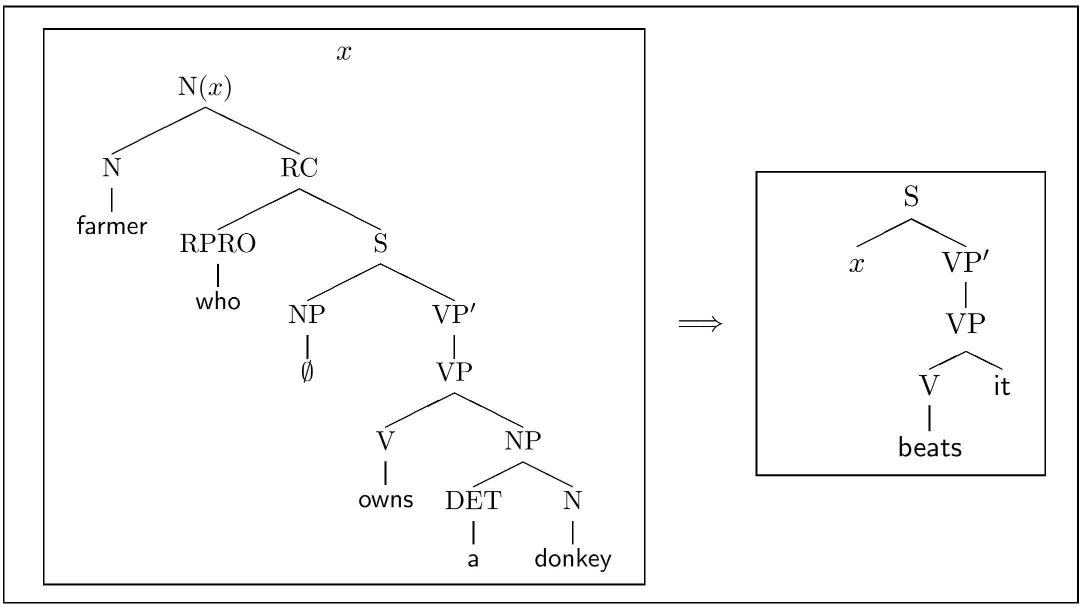
Reguła kwantyfikacji ogólnej

W rachunku predykatów formuły kwantyfikowane ogólnie mają zazwyczaj (choć nie jest to w żaden sposób narzucone przez formalizm jako taki) postać implikacji: $\forall x\ (P(x) \rightarrow Q(x))$. Nie jest to bynajmniej przypadek. Rozważmy bowiem dwa zdania: *If a farmer owns a donkey, then he beats it.* oraz *Every framer who owns a donkey beats it.*. Nawet jeśli znaczenie tych zdań nie jest identyczne, to rozumiemy je w sposób bardzo podobny: Każdy farmer, jeśli tylko posiada osła, bije go. Dlatego kwantyfikatora ogólny **every** jest traktowany w sposób bardzo podobny do zdań warunkowych. Z drugiej strony, ponieważ **every** $\in$ DET jest rodzajnikiem podobnie jak **a** i mogą po nim występować dokładnie takie same frazy rzeczownikowe jak we frazie nieokreślonej, istnieją także spore podobieństwa do reguły CR.ID.

Definicja 3.30. (reguła **CR.EVR**) Rozważmy $\mathtt{ref} \in R$, $\mathtt{noun} \in \mathrm{N}$ oraz $K = \langle U_K, \mathrm{Con}_K\rangle$ takie, że $\langle \breve{S}, I\rangle \in \mathrm{Con}_K$ oraz $\breve{S} \ni \breve{S}^k = \langle \mathrm{NP}, \langle \mathrm{DET}, \langle \mathbf{every}\rangle\rangle, \mathfrak{d}(\mathrm{N})\rangle$, gdzie $\mathfrak{d}(\mathrm{N}) = \langle \mathrm{N}, \langle \mathrm{N}, \langle \mathtt{noun}\rangle\rangle, \mathfrak{d}(\mathrm{RC})\rangle$ lub $\mathfrak{d}(\mathrm{N}) = \langle N, \langle \mathtt{noun}\rangle\rangle$. Wówczas $\mathfrak{g}(K) = \langle U_{K'}, \mathrm{Con}_{K'}\rangle$ jest strukturą spełniającą warunki:

1. $U_{K'} = U_K$,
2. $\breve{S}' = \breve{S}[\langle \mathrm{NP}, \langle \mathrm{DET}, \langle \mathbf{every}\rangle\rangle, \mathfrak{d}(\mathrm{N})\rangle / \langle \mathtt{ref}\rangle]$,
3. $\hat{S} = \mathfrak{d}(\mathrm{N})[\mathrm{N}/\mathrm{N}(\mathtt{ref})]$,
4. $\mathit{Num}(\mathtt{ref}) = \mathrm{N}.\mathit{Num}$,
5. $\mathit{Gen}(\mathtt{ref}) = \mathrm{N}.\mathit{Gen}$,
6. $L_1 = \langle \{\mathtt{ref}\}, \{\mathit{Num}(\mathtt{ref}), \mathit{Gen}(\mathtt{ref}), \langle \hat{S}, \emptyset\rangle\}\rangle$,
7. $L_2 = \langle \emptyset, \{\langle \breve{S}', \emptyset\rangle\}\rangle$,
8. $\mathrm{Con}_{K'} = \mathrm{Con}_K - \{\langle \breve{S}, I\rangle\} \cup \{L_1 \Longrightarrow L_2\}$.

<u>Przykład</u> *3.21.* Weźmy wspomniane powyżej zdanie *Every farmer who owns a donkey beats it.*. Po użyciu reguły CR.EVR uzyskamy strukturę K, w której $U_K = \emptyset$, $\mathrm{Con}_K = \{L_1 \Longrightarrow L_2\}$, przy czym $L_1 = \langle \{x\}, \{\mathit{Num}(x), \mathit{Gen}(x), \langle \mathrm{N}(x), \langle \mathrm{N}, \langle \mathbf{farmer}\rangle\rangle, \mathfrak{d}(\mathrm{RC})\rangle\}\rangle$, zaś $L_2 = \langle \emptyset, \{[x \text{ beats it}]\}\rangle$ (poniżej).



Po pełnej analizie struktury L_1 uzyskamy strukturę K', w której L_1 zastąpiona zostanie $L_1' = \langle \{x, y\}, \{\mathbf{farmer}\,(x), \mathit{Num}(x), \mathit{Gen}(x), \mathbf{donkey}\,(y), \mathit{Num}(y), \mathit{Gen}(y), x\ \mathbf{owns}\ y\}\rangle$ (poniżej po lewej). Ponieważ znaczniki z L_1 są dostępne w L_2, możemy przeprowadzić jej analizę otrzymując strukturę K'', w której L_2 zastąpiona zostanie $L_2'' = \langle \{u\}, \{u = y, x\ \mathbf{owns}\ u\}\rangle$ (po prawej).

$$\boxed{\boxed{\begin{array}{c} x,y \\ \textbf{farmer}\,(x) \\ \textbf{donkey}\,(y) \\ x\ \textbf{owns}\ y \end{array}} \quad \Longrightarrow \quad \boxed{[x\ \textsf{beats it}]}} \qquad \boxed{\boxed{\begin{array}{c} x,y \\ \textbf{farmer}\,(x) \\ \textbf{donkey}\,(y) \\ x\ \textbf{owns}\ y \end{array}} \quad \Longrightarrow \quad \boxed{\begin{array}{c} u \\ u = y \\ x\ \textbf{beats}\ u \end{array}}}$$

Zauważmy, że struktura K'' uzyskana w powyższym przykładzie jest niezmiernie podobna do struktury K' z przykładu 3.20: jedyną różnicę stanowi wystąpienie w tej drugiej znacznika v. Ponieważ jednak zachodzi $v = y$, interpretacja tych zdań jest taka sama przynajmniej w tym sensie, że prawdziwe są w dokładnie tej samej klasie modeli (a więc zgodnie z definicją 3.9 rozważane struktury są równoważne). Rzecz jasna znaczenie tych zdań nie jest w pełni identyczne dlatego, że w pewnych okolicznościach v mogłoby zostać powiązane z innym znacznikiem niż y (choćby w zdaniach: *Jones hates donkeys. If a farmer owns a donkey, then he beats it.*).

Rodzajnik *every* może oczywiście występować w prostszych zdaniach; może także występować na pozycji dopełnienia.

Przykład *3.22.* Rozważmy zdania *Every farmer beats a donkey.* oraz *A farmer beats every donkey.*. Akceptując zasadę „podmiot przed dopełnieniem" do pierwszego z nich aplikujemy regułę CR.EVR, a do drugiego — CR.ID, co daje struktury K_1 i K_2, gdzie $U_{K_1} = \emptyset$, $\mathrm{Con}_{K_1} = \{L_{1,1} \Longrightarrow L_{1,2}\}$, gdzie $L_{1,1} = \langle\{x\}, \{Num(x), Gen(x), \langle \mathrm{N}(x), \langle \textsf{farmer}\rangle\rangle\}\rangle$, $L_{1,2} = \langle\emptyset, \{[x\ \textsf{beats a donkey}]\}\rangle$,

$$\boxed{\boxed{\begin{array}{c} x \\ \mathrm{N}(x) \\ | \\ \textsf{farmer} \end{array}} \quad \Longrightarrow \quad \boxed{[x\ \textsf{beats a donkey}]}} \qquad \boxed{\begin{array}{c} x \\ \mathrm{N}(x) \\ | \\ \textsf{farmer} \\ [x\ \textsf{beats every donkey}] \end{array}}$$

zaś $L_{2,1} = \langle\{y\}, \{Num(y), Gen(y), \langle \mathrm{N}(y), \langle \textsf{donkey}\rangle\rangle\}\rangle$, $L_{2,2} = \langle\emptyset, \{[\textsf{a farmer beats}\ y]\}\rangle$ (po lewej). Natomiast $U_{K_2} = \{x\}$ oraz $\mathrm{Con}_{K_2} = \{Num(x), Gen(x), \langle \mathrm{N}(x), \langle \textsf{farmer}\rangle\rangle, [x\ \textsf{beats every donkey}]\}$ (po prawej). Stosując z kolei regułę CR.ID do struktury K_1, a CR.EVR — do K_2 oraz CR.LIN (po dwa razy), uzyskujemy struktury K_1', K_2', przy czym $L_{1,1}$, $L_{1,2}$ zostają zastąpione przez $L'_{1,1} = \langle\{x\}, \{Num(x), Gen(x), \textbf{farmer}\,(x)\}\rangle$, $L'_{1,2} = \langle\{y\}, \{\textbf{donkey}\,(y), x\ \textit{beats}\ y\}\rangle$, zaś $U_{K_2'} = \{Num(x), Gen(x), \textbf{farmer}\,(x), L_{2,1} \Longrightarrow L_{2,2}\}$, gdzie $L_{2,1} = \langle\{y\}, \{Num(y), Gen(y), \textbf{donkey}\,(y)\}\rangle$, a $L'_{2,2} = \langle\emptyset, \{x\ \textit{beats}\ y\}\rangle$.

$$\boxed{\boxed{\begin{array}{c} x \\ \textbf{farmer}\,(x) \end{array}} \quad \Longrightarrow \quad \boxed{\begin{array}{c} y \\ \textbf{donkey}\,(y) \\ x\ \textbf{beats}\ y \end{array}}} \qquad \boxed{\begin{array}{c} x \\ \textbf{farmer}\,(x) \\ \boxed{\begin{array}{c} y \\ \textbf{donkey}\,(y) \end{array}} \ \Longrightarrow\ \boxed{x\ \textbf{beats}\ y} \end{array}}$$

Dla zdania *A farmer beats every donkey which he owns.* analiza przebiegałaby w analogiczny sposób.

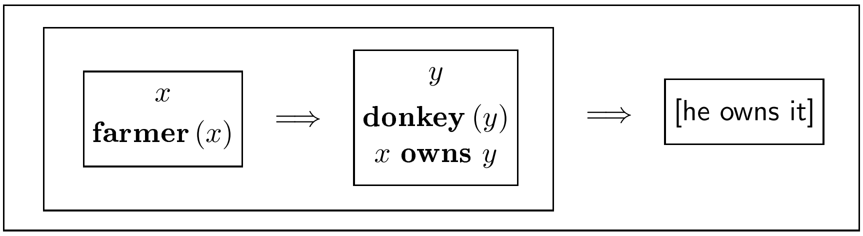
Ponieważ frazy zawierające rodzajnik *every* traktowane są podobnie jak zdania warunkowe, jest oczywiste, że także narzucają ograniczenia na powiązania anaforyczne. Pokażemy to na następującym przykładzie.

Przykład *3.23.* Weźmy ciąg dwóch zdań *Every profesor owns Ulisses. He likes it.*. Po przetworzeniu pierwszego z nich i „dołożeniu" drugiego otrzymamy strukturę K taką, że $U_K = \{y\}$, $\mathrm{Con}_K = \{L_1 \Longrightarrow L_2, [\textsf{He likes it}]\}$, gdzie $L_1 = \langle\{x\}, \{\textbf{professor}\,(x)\}\rangle$, zaś $L_2 = \langle\emptyset, \{\textbf{Ulisses}\,(y), x\ \textbf{owns}\ y\}\rangle$ (obok). Jesteśmy co prawda w stanie powiązać zaimek *it* ze znacznikiem y, ale znacznik x nie jest widoczny z poziomu głównej struktury, więc ciągu tego nie da się zinterpretować.

$$\boxed{\begin{array}{c} y \\ \boxed{\begin{array}{c} x \\ \textbf{professor}\,(x) \end{array}} \quad \Longrightarrow \quad \boxed{\begin{array}{c} \textbf{Ulisses}\,(y) \\ x\ \textbf{owns}\ y \end{array}} \end{array}}$$

Podobna sytuacja ma miejsce w przypadku ciągu zdań *Every farmer owns a donkey. He beats it.*, tyle że tym razem żaden zaimek nie zostanie powiązany. Problem ten dotyczy także wypadku, w którym frazę z rodzajnikiem *every* jest zagnieżdżona w zdaniu warunkowym.

Przykład *3.24.* Rozważmy zdanie *If every farmer owns a donkey, he beats it.*. Po zastosowaniu reguły CR.COND i przeanalizowaniu zdania *every farmer owns a donkey* otrzymujemy strukturę K, w której $U_K = \emptyset$, $\mathrm{Con}_K = \{L_1 \Longrightarrow L_2\}$, gdzie $L_1 = \langle \emptyset, \{M_1 \Longrightarrow M_2\}\rangle$, a $M_1 = \langle\{x\}, \{\mathbf{farmer}\,(x)\}\rangle$, $M_2 = \langle\{y\}, \{\mathbf{donkey}\,(y)$, x *owns* $y\}\rangle$, zaś $L_2 = \langle\emptyset$, $\{$[he beats it]$\}\rangle$ (patrz obok).

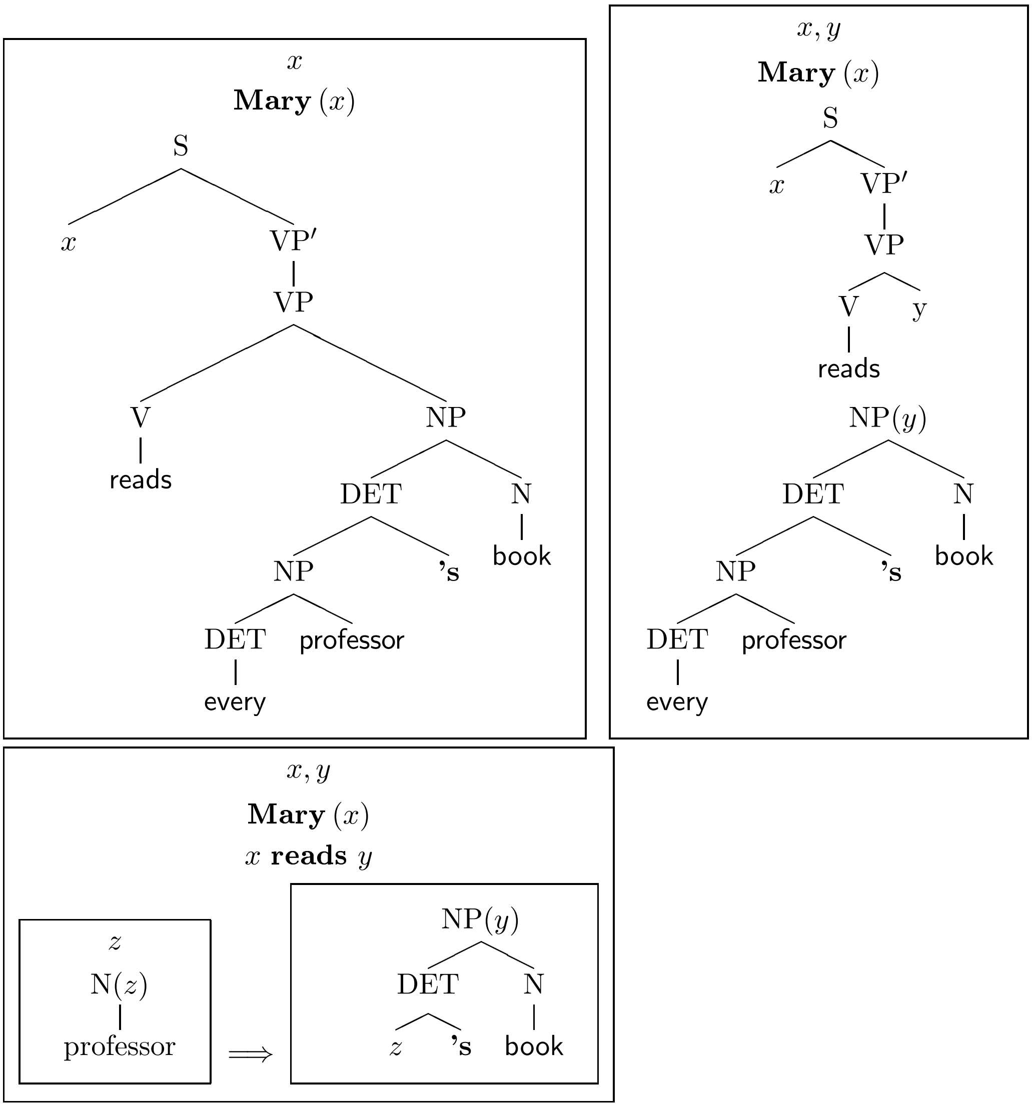


Ponieważ jednak co prawda znaczniki dyskursu zdefiniowane w poprzedniku implikacji są dostępne ze wszystkich struktur podrzędnych jej następnika, ale te zdefiniowane w strukturach podrzędnych poprzednika nie są bynajmniej dostępne z następnika, znaczniki x i y nie są widoczne z L_2, a więc brak poprzedników dla zaimków he i it.

Przedstawimy jeszcze przykład zdania zawierającego frazę dzierżawczą.

Przykład *3.25.* Weźmy zdanie *Mary reads every professor's book.*. Po zastosowaniu reguły CR.PN będzie ono reprezentowane przez strukturę K taką, że $U_K = \{x\}$, $\mathrm{Con}_K = \{\mathbf{Mary}\,(x), \breve{S}\}$. Wówczas reguła CR.NP'sNP da nam strukturę K' taką, że $U_{K'} = \{x, y\}$, $\mathrm{Con}_K = \{\mathbf{Mary}\,(x), [x \text{ reads } y], \hat{S}\}$. Następnie użycie reguły CR.EVR tworzy strukturę K' taką, że $U_{K'} = \{x, y\}$, $\mathrm{Con}_K = \{\mathbf{Mary}\,(x), [x \text{ reads } y], L_1 \Longrightarrow L_2\}$, gdzie $L_1 = \langle\{z\}, \{\langle \mathrm{N(z)}, \langle \text{professor}\rangle\rangle\}\rangle$, zaś $L_2 = \langle\emptyset, \{\hat{S}'\}\rangle$.

Zauważmy, że omawiane zdanie ma w gramatyce z rozdz. 3.1 tylko jedno drzewo rozbioru (takie jak poniżej), zaś zdanie *Mary reads every Bill's book.* w ogóle nie jest przez nią akceptowane.



3.4.4 Alternatywa

W przeciwieństwie do zależności warunkowych, które mogą wiązać wyłącznie określone typy fraz, wyraz **or** reprezentujący alternatywę może łączyć frazy dowolnego typu. Kamp i Reyle nie narzucają żadnych ograniczeń w tym względzie. Sami jednak stwierdzają, że dyzjunkcja nazw własnych (PN) prowadzi do generowania alernatywncyh rozbiorów dla tej samej klasy zdań, co dla nadrzędnej względem niej klasy fraz rzeczownikowych (NP), warto więc zrezygnować z dyzjunkcji klasy PN dla uniknięcia redundancji. To samo dotyczy klasy zaimków osobowych (PRO). W przypadku fraz czasownikowych sytuacja jest jeszcze bardziej wyrazista, bo dyzjunkcja dla VP nie tylko powoduje redundancję (wraz z VPP), ale dopuszcza też błędny rozkład (i interpretację semantyczną) zdań typu *Smiths do not (like Mary or hate John).*.[25]

Autorzy przyjmują, że znaczniki dyskursu należące do pewnego składnika dyzjunkcji nie są dostępne z poziomu pozostałych jej składników niezależnie od kolejności ich występowania (patrz def. 3.3). Wynika to z konstatacji, że powiązanie zaimka *it* z rzeczownikiem *book* jest niewłaściwe nie tylko w zdaniu *Jones owns it or Smith owns a book.*, lecz także w zdaniu *Jones owns a book or Smith owns it.*. Nie ulega jednak kwestii, że istnieją poprawne zdania tego typu. Wówczas w grę wchodzą pewne dodatkowe okoliczności semantyczne. I tak w zdaniu *Jones owns a green Porsche which often passes our house or Smith owns it.* uznajemy frazę *a green Porsche which often passes our house* za jednoznaczną identyfikację pojedynczego obiektu, a więc coś funkcjonującego jak nazwa własną. Podobnie w zdaniu *Jones has borrowed a bicycle or he has rented it.* czytelnik ma wrażenie, że chodzi o jedno i to samo zdarzenie. Kamp i Reyle sugerują, że w takich przypadkach znacznik (reprezentujący *a green Porsche* lub *a bicycle*, odpowiednio) powinien zostać umieszczony w strukturze głównej, tak jak to się dzieje w przypadku nazw własnych. Nie precyzują jednak, w jaki sposób identyfikować takie sytuacje, gdyż wykracza to poza zakres pracy.

Powstaje wrażenie, że pewien wpływ na ten fakt ma pozycja rzeczownika w zdaniu. Porównajmy zdania *Jones owns a book or Smith owns it.* oraz *A book belongs to Jones or it belongs to Smith.* — drugie zdecydowanie brzmi lepiej i wydaje się poprawne. Prawdopodobnie wynika to z faktu, że podmiot silniej identyfikuje obiekt niż dopełnienie.

Niezależnie od tego, jakich fraz dotyczy dyzjunkcja, reprezentacja semantyczna zawiera dyzjunkcję całych zdań. Dlatego odpowiednie reguły semantyczne usuwają drzewa o korzeniu S ze struktury, a następnie powielają je w podstrukturach powiązanych spójnikiem $\vee$. Ponieważ każde poddrzewo o korzeniu S na pewnym etapie przetwarzania (jako jeden z efektów reguły CR.NRC) staje się samodzielnym warunkiem w quasi-strukturze, bez utraty ogólności reguły CR.OR mogą być stosowane na takim właśnie etapie przetwarzania.

<u>Reguła dyzjunkcji dla zdań</u>

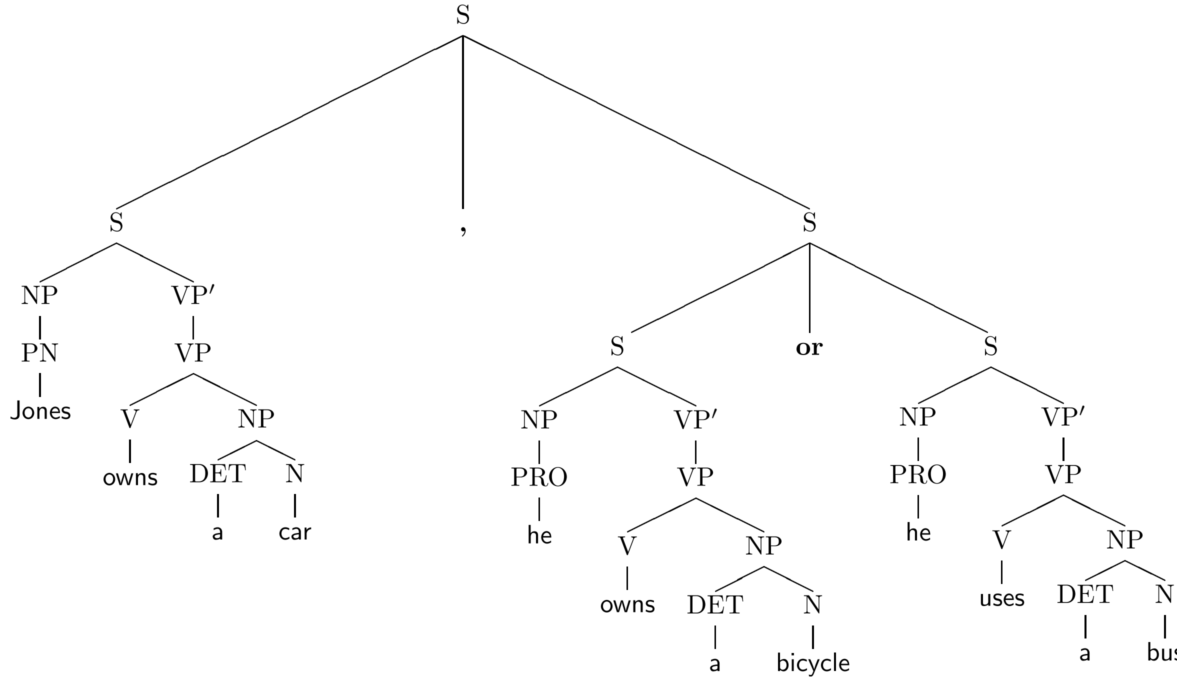
Zaczniemy od najprostszego przypadku, czyli sytuacji, w której alternatywie poddawane są całe zdania. Przypadek ten musiał być wyróżniony spośród pozostałych, gdyż drzewo o korzeniu S nie jest własnym poddrzewem.

Definicja 3.31. (reguła **CR.OR-S**) Weźmy strukturę $K = \langle U_K, \mathrm{Con}_K \rangle$ taką, że $\langle \breve{S}, I \rangle \in \mathrm{Con}_K$ oraz $\breve{S} = \langle S, \mathfrak{d}(S)_1, \langle , \rangle, \langle \mathbf{S}, \mathfrak{d}(\mathbf{S})_\mathbf{2}, \langle , \rangle, ..., \langle \mathbf{S}, \mathfrak{d}(\mathbf{S})_{\mathbf{n-1}}, \langle \mathbf{or} \rangle, \mathfrak{d}(\mathbf{S})_\mathbf{n} \rangle ... \rangle \rangle$. Wówczas $\mathfrak{g}(K) = \langle U_{K'}, \mathrm{Con}_{K'} \rangle$ spełnia warunki:

1. $U_{K'} = U_K$,
2. $L_i = \langle \emptyset, \{ \langle \mathfrak{d}(S)_i, \emptyset \rangle \} \rangle$, dla $i = 1, ..., n$,
3. $\mathrm{Con}_{K'} = \mathrm{Con}_K - \{ \langle \breve{S}, I \rangle \} \cup \{ L_1 \vee ... \vee L_n \}$.

[25] W liczbie pojedynczej zjawisko to nie zachodzi ze względu na przemieszczenie *(e)s*.

<u>Przykład</u> *3.26.* Zdanie *Jones owns a car, he owns a bicycle or he uses a bus.*[26] ma następujące drzewo rozbioru:



Dzięki regule CR.OR-S uzyskujemy strukturę K taką, że $U_K = \emptyset$, $\mathrm{Con}_K = \{L_1 \vee L_2 \vee L_3\}$, gdzie $L_1 = \langle \emptyset, \{[\text{Jones owns a car}]\}\rangle$, $L_2 = \langle \emptyset, \{[\text{he owns a bicycle}]\}\rangle$, zaś $L_3 = \langle \emptyset, \{[\text{he uses a bus}]\}\rangle$ zaprezentowaną graficznie poniżej.

$$\boxed{\boxed{[\text{Jones owns a car}]} \quad \vee \quad \boxed{[\text{he owns a bicycle}]} \quad \vee \quad \boxed{[\text{he uses a bus}]}}$$

Ponieważ znacznik odnoszący się do nazwy własnej *Jones* zostanie umieszczony w strukturze głównej, oba wystąpienia zaimka he zostaną zinterpretowane prawidłowo.

<u>Reguła dyzjunkcji dla innych fraz</u>

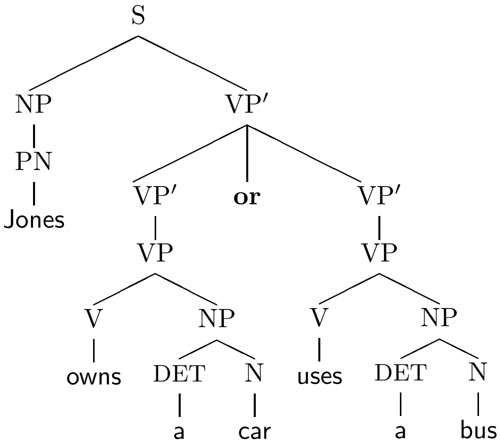
Dla pozostałych przypadków dyzjunkcja może być opisana za pomocą pojedynczej reguły. Jest to o tyle trudniejsze, że tym razem nie możemy po prostu rozdystrybuować fraz pomiędzy alternatywne podstruktury; musimy natomiast powielić zdanie, za każdym razem wybierając odpowiedni „egzemplarz” frazy będącej obiektem dyzjunkcji.

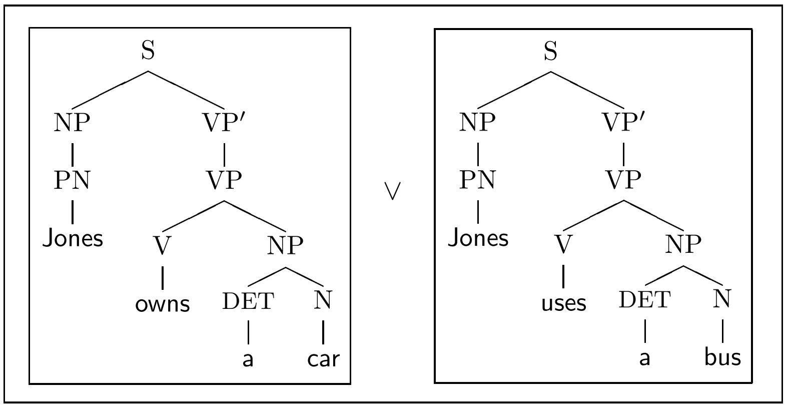
Definicja 3.32. (reguła **CR.OR-OTH**) Weźmy strukturę $K = \langle U_K, \mathrm{Con}_K \rangle$ taką, że $\langle \check{S}, I\rangle \in \mathrm{Con}_K$ oraz $\check{S} \ni \check{S}^k = \langle X, \mathfrak{d}(X)_1, \langle , \rangle, \langle \mathbf{X}, \mathfrak{d}(\mathbf{X})_\mathbf{2}, \langle , \rangle, ..., \langle \mathbf{X}, \mathfrak{d}(\mathbf{X})_{\mathbf{n-1}}, \langle \mathbf{or} \rangle, \mathfrak{d}(\mathbf{X})_\mathbf{n} \rangle ... \rangle\rangle$, gdzie X jest pewnym symbolem terminalnym spośród VP′, V, NP i N. Wówczas $\mathfrak{g}(K) = \langle U_{K'}, \mathrm{Con}_{K'} \rangle$ jest strukturą spełniającą warunki:

1. $U_{K'} = U_K$,
2. $L_i = \langle \emptyset, \{\langle \check{S}[\check{S}^k / \mathfrak{d}(X)_i], \emptyset \rangle\}\rangle$, dla $i = 1, ..., n$,
3. $\mathrm{Con}_{K'} = \mathrm{Con}_K - \{\langle \check{S}, I\rangle\} \cup \{L_1 \vee ... \vee L_n\}$.

<u>Przykład</u> *3.27.* Zdanie *Jones owns a car or uses a bus.* posiada drzewo rozbioru przedstawione po lewej. Dzięki regule CR.OR-OTH (dla X = VP′) uzyskujemy strukturę K_1 taką, że $U_{K_1} = \emptyset$, $\mathrm{Con}_{K_1} = \{L_1 \vee L_2\}$, gdzie $L_1 = \langle \emptyset, \{[\text{Jones owns a car}]\}\rangle$, zaś $L_2 = \langle \emptyset, \{[\text{Jones uses a bus}]\}\rangle$ widoczną po prawej.

[26] Zauważmy, że w przypadku zdań czasami wolimy oddzielać wszystkie składniki dyzjunkcji wyrazem *or*, którego to przypadku autorzy nie uwzględnili.



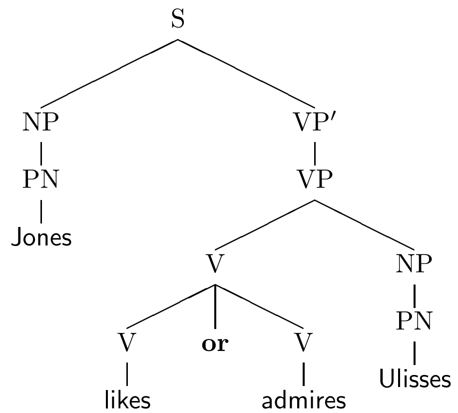


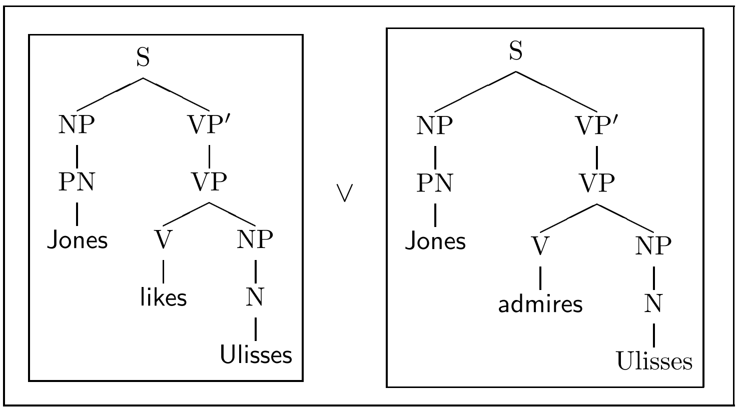
Reguła ta dla zdania *Jones does not own Ulisses or reads it.* daje w rezultacie strukturę K_2 taką, że $U_{K_2} = \emptyset$, $\mathrm{Con}_{K_1} = \{L_1 \vee L_2\}$, gdzie $L_1 = \langle\emptyset, \{[\textsf{Jones does not own Ulisses}]\}\rangle$, zaś $L_2 = \langle\emptyset, \{[\textsf{Jones reads it}]\}\rangle$ (obok). Tak jak

[Jones does not own Ulisses] ∨ [Jones reads it]

w przykładzie 3.26, zaimek *it* zostanie powiązany ze znacznikiem umieszczonym w strukturze głównej.

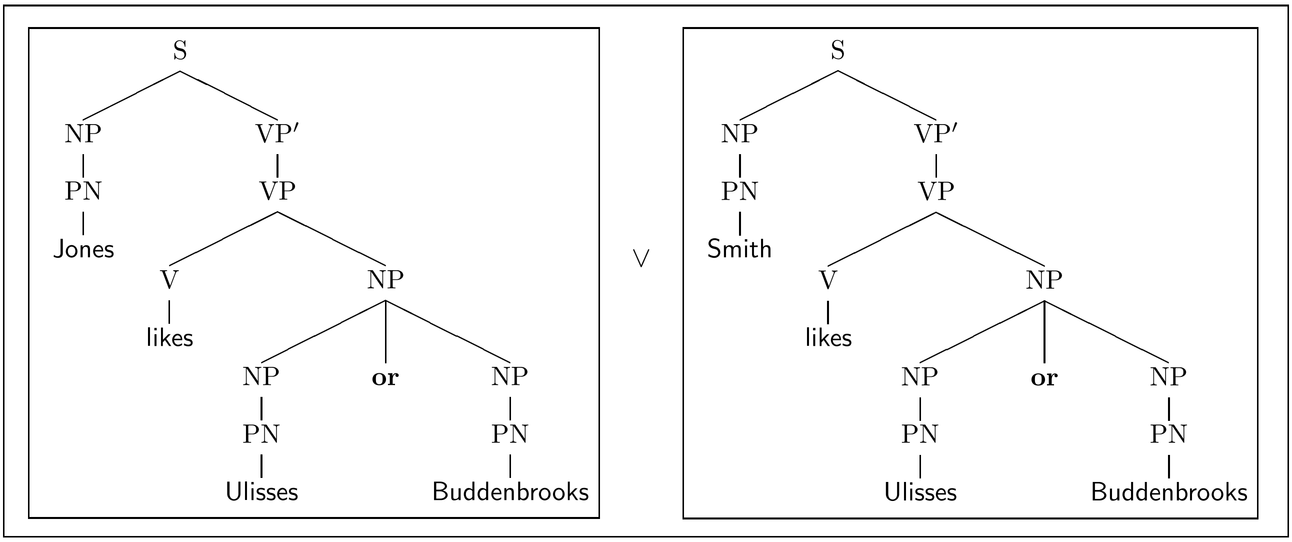
Przykład *3.28.* Rozbiór kolejnego zdania *Jones likes or (even) admires Ulisses.* przedstawiony jest po lewej. Zastosowanie reguły CR.OR-OTH (dla X = V) daje strukturę K taką, że $U_K = \emptyset$, $\mathrm{Con}_K = \{L_1 \vee L_2\}$, gdzie $L_1 = \langle\emptyset, \{[\textsf{Jones likes Ulisses}]\}\rangle$, zaś $L_2 = \langle\emptyset, \{[\textsf{Jones admires Ulisses}]\}\rangle$ (po prawej).



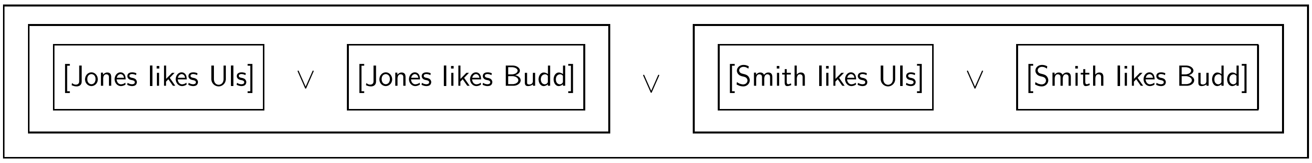


Gramatyka zaproponowana przez Kampa i Reyle'go nie akceptuje zdań typu *Jones does not own or reads Ulisses.*, gdyż negacja powiązana jest z frazą czasownikową jako całością.

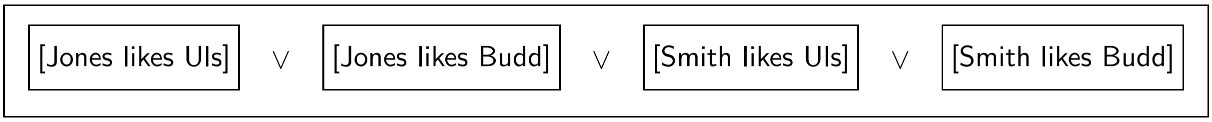
Przykład *3.29.* Przeanalizujmy zdanie z dyzjunkcją fraz rzeczownikowych zarówno w miejscu podmiotu, jak i dopełnienia *Jones or Smith likes Ulisses or Buddenbrooks.*. Zgodnie z przyjętą przez autorów zasadą aplikowania reguł „od lewej" najpierw zastosujemy regułę CR.OR-OTH (dla X = NP) dla podmiotu, co da w rezultacie strukturę K taką, że $U_K = \emptyset$, $\mathrm{Con}_{K_1} = \{L_1 \vee L_2\}$, gdzie $L_1 = \langle\emptyset, \{[\textsf{Jones likes Ulisses or Buddenbrooks}]\}\rangle$, zaś $L_2 = \langle\emptyset, \{[\textsf{Smith likes Ulisses or Buddenbrooks}]\}\rangle$. Po zastosowaniu tejże reguły do dopełnienia uzyskamy strukturę K' taką, że $U_{K'} = \emptyset$, $\mathrm{Con}_{K'} = \{L'_1 \vee L'_2\}$, gdzie $L'_1 = \langle\emptyset, \{M_{1,1} \vee M_{1,2}\}\rangle$, $M_{1,1} = \langle\emptyset, \{[\textsf{Jones likes Ulisses}]\}\rangle$, $M_{1,2} = \langle\emptyset, \{[\textsf{Jones likes Budd}]\}\rangle$, zaś $L'_2 = \langle\emptyset, \{M_{2,1} \vee M_{2,2}\}\rangle$, $M_{2,1} = \langle\emptyset, \{[\textsf{Smith likes Ulisses}]\}\rangle$, $M_{1,2} = \langle\emptyset, \{[\textsf{Smith likes Budd}]\}\rangle$.



Po zastosowaniu tejże reguły do dopełnienia uzyskamy strukturę K' taką, że $U_{K'} = \emptyset$, $\mathrm{Con}_{K'} = \{L'_1 \vee L'_2\}$, gdzie $L'_1 = \langle \emptyset, \{M_{1,1} \vee M_{1,2}\}\rangle$, $M_{1,1} = \langle \emptyset, \{[\textsf{Jones likes Ulisses}]\}\rangle$, $M_{1,2} = \langle \emptyset, \{[\textsf{Jones likes Budd}]\}\rangle$, zaś $L'_2 = \langle \emptyset, \{M_{2,1} \vee M_{2,2}\}\rangle$, $M_{2,1} = \langle \emptyset, \{[\textsf{Smith likes Ulisses}]\}\rangle$, $M_{1,2} = \langle \emptyset, \{[\textsf{Smith likes Budd}]\}\rangle$.



Gdyby kolejność aplikowania reguł zależała od ich hierarchii, a nie od pozycji w drzewie rozbioru, w powyższym zdaniu moglibyśmy najpierw zastosować regułę CR.OR-OTH najpierw do dopełnienia, a następnie do podmiotu, co dałoby w rezultacie nieco inną strukturę wynikową. Jednak twierdz. 3.1 (ii) umożliwia równoważną reprezentację za pomocą struktury K'' takiej, że $U_{K'} = \emptyset$, $\mathrm{Con}_{K'} = \{\langle \emptyset, \{[\textsf{Jones likes Ulisses}]\}\rangle \vee \langle \emptyset, \{[\textsf{Jones likes Budd}]\}\rangle \vee \langle \emptyset, \{[\textsf{Smith likes Ulisses}]\}\rangle \vee \langle \emptyset, \{[\textsf{Smith likes Budd}]\}\rangle\}$.



Reguła dyzjunkcji dla rzeczowników pospolitych

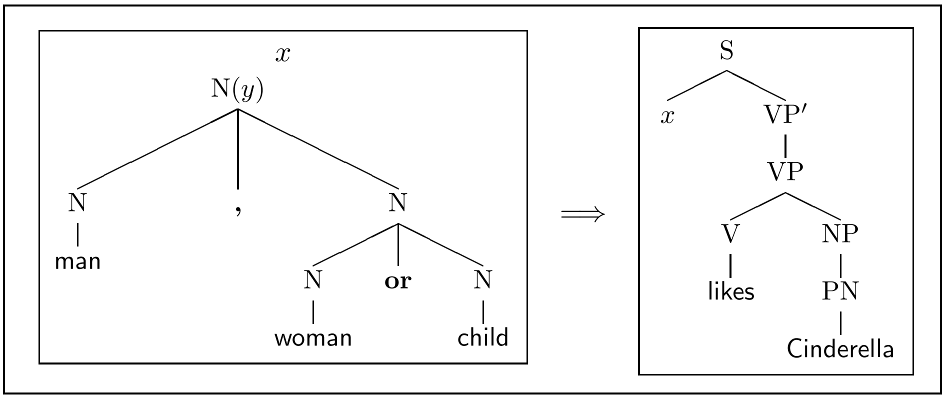
We wszystkich rozważanych dotychczas przypadkach dyzjunkcji powieleniu ulegało drzewo o korzeniu S reprezentujące zdanie. Rzecz ma się inaczej w przypadku rzeczowników pospolitych. Fraza „**every** $(\mathrm{N}_1, \mathrm{N}_2, ..., \mathrm{N}_\mathrm{n})$” po zastosowaniu reguły CR.EVR tworzy bowiem drzewo o korzeniu $\mathrm{N}(y)$ (analogicznie jest w przypadku akceptowanej przez gramatykę z rozdz. 3.1, choć niepoprawnej frazy „(**a**)**n** $(\mathrm{N}_1, \mathrm{N}_2, ..., \mathrm{N}_\mathrm{n})$”, którą zignorujemy). Jest to więc prosty przypadek: poszczególne rzeczowniki muszą zostać rozdystrybuowne pomiędzy alternatywne podstruktury.

Różnice notacyjne w stosunku do oryginalnej pracy Kampa i Reyle'go tłumaczą, dlaczego podział reguły dyzjunkcji na podprzypadki przebiega w niniejszym opracowaniu nieco inaczej niż u nich. Nie mniej jednak brak odrębnej reguły dla rzeczowników pospolitych (który to przypadek szczegółowo omawiają) jest zaiste zaskakujący.

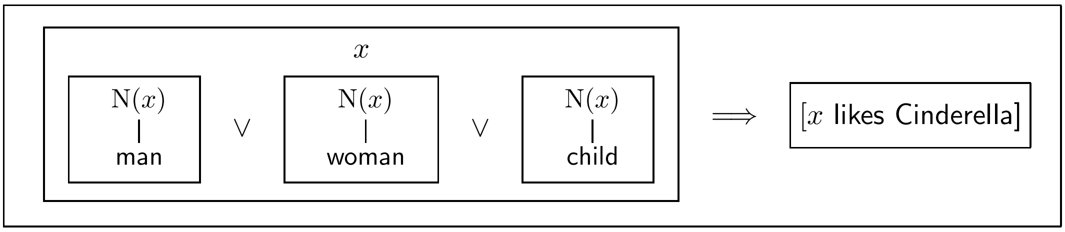
Definicja 3.33. (reguła **CR.OR-N**) Niech $\mathtt{ref} \in R$, $\mathtt{noun}_1, ..., \mathtt{noun}_1 \in \mathrm{N}$ oraz K będzie strukturą $\langle U_K, \mathrm{Con}_K\rangle$ taką, że $\langle \breve{S}, I\rangle \in \mathrm{Con}_K$ oraz $\breve{S} = \langle N(\mathtt{ref}), \langle N, \langle \mathtt{noun}_1\rangle\rangle, \langle , \rangle, \langle N, \langle N, \langle \mathtt{noun}_2\rangle\rangle, \langle , \rangle, ..., \langle N, \langle \mathtt{noun}_{n-1}\rangle\rangle, \langle \mathbf{or}\rangle, \langle N, \langle \mathtt{noun}_n\rangle\rangle ...\rangle\rangle$. Wówczas $\mathfrak{g}(K) = \langle U_{K'}, \mathrm{Con}_{K'}\rangle$ spełnia warunki:

1. $U_{K'} = U_K$,
2. $L_i = \langle \emptyset, \{\langle\langle N(\mathtt{ref}), \langle \mathtt{noun}_i\rangle\rangle, I\rangle\}\rangle$, dla $i = 1, ..., n$,
3. $\mathrm{Con}_{K'} = \mathrm{Con}_K - \{\langle \breve{S}, I\rangle\} \cup \{L_1 \vee ... \vee L_n\}$.

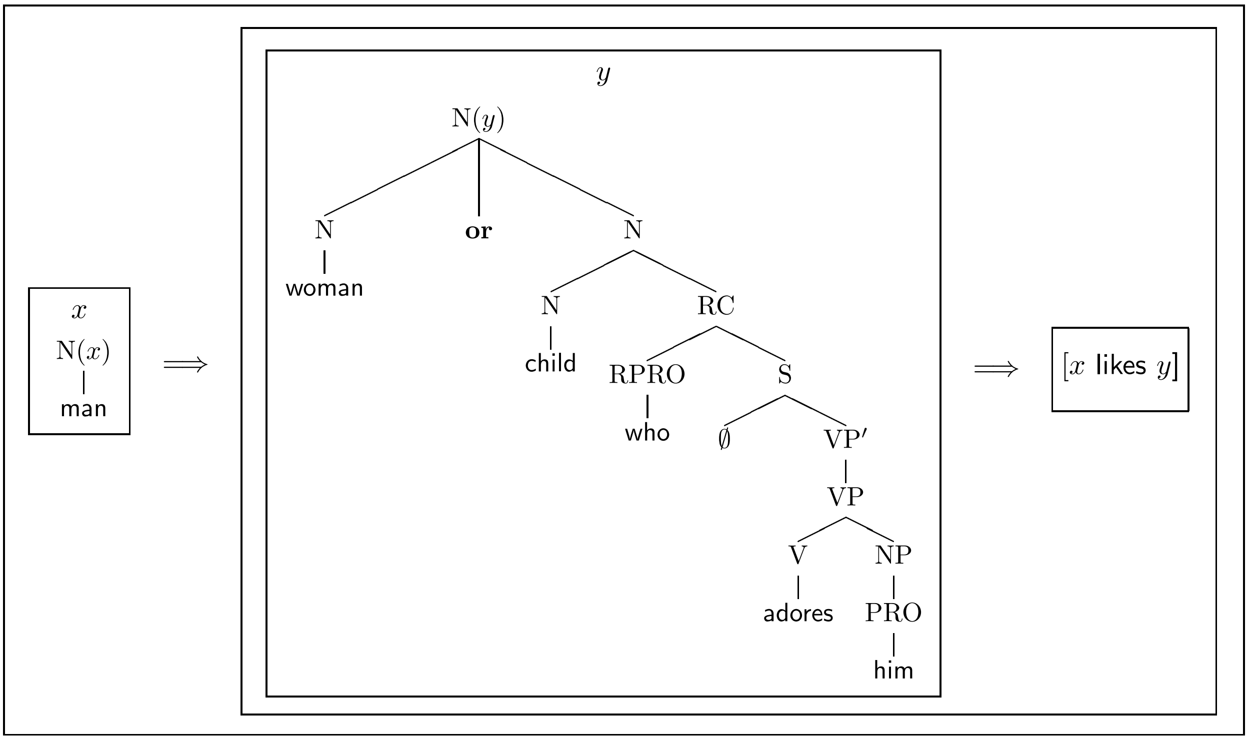
<u>Przykład</u> *3.30.* Weźmy najprostsze zdanie *Every man, woman or child likes Cinderella.*. Po zastosowaniu reguły CR.EVR uzyskamy strukturę K, w której $U_K = \emptyset$, $\mathrm{Con}_K = \{L_1 \Longrightarrow L_2\}$, przy czym $L_1 = \langle\{x\}, \{\langle \mathrm{N}(x), \langle \mathrm{N}, \langle \textsf{man}\rangle\rangle, \langle , \rangle, \langle \mathrm{N}, \langle \mathrm{N}, \langle \textsf{woman}\rangle\rangle, \langle \textbf{or}\rangle, \langle \mathrm{N}, \langle \textsf{child}\rangle\rangle\rangle\rangle\}\rangle$, zaś $L_2 = \langle\emptyset, \{[x \text{ likes Cinderella}]\}\rangle$.



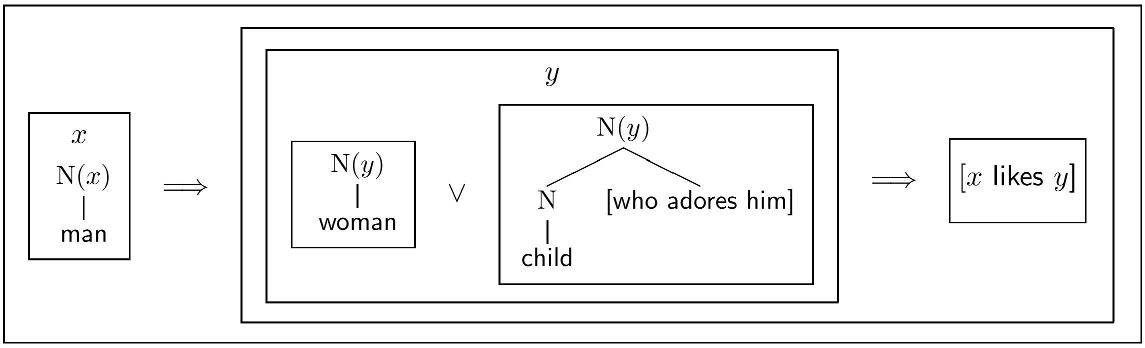
Następnie, dzięki regule CR.OR-N powstanie struktura K', w której L_1 zostanie zastąpiona przez $L_1' = \langle\{x\}, \langle\emptyset, \{\langle \mathrm{N}(x), \langle \textsf{man}\rangle\rangle\}\rangle \vee \langle\emptyset, \{\langle \mathrm{N}(x), \langle \textsf{woman}\rangle\rangle\}\rangle \vee \langle\emptyset, \{\langle \mathrm{N}(x), \langle \textsf{child}\rangle\rangle\}\rangle\rangle$.



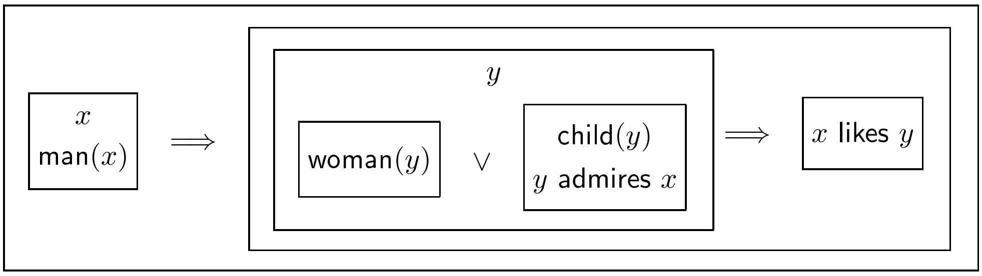
<u>Przykład</u> *3.31.* Reguła ta daje się zastosować do całkiem skomplikowanych zdań, np. *Every man likes every woman or child who adores him.*. Po dwukrotnym zastosowaniu reguły CR.EVR uzyskamy strukturę K, w której $U_K = \emptyset$, $\mathrm{Con}_K = \{L_1 \Longrightarrow L_2\}$, przy czym $L_1 = \langle\{x\}, \{\langle \mathrm{N}(x), \langle \textsf{man}\rangle\rangle\}\rangle$, zaś $L_2 = \langle\emptyset, \{M_1 \Longrightarrow M_2\}\rangle$, gdzie $M_1 = \langle\{y\}, \{\langle \mathrm{N}(y), \langle \mathrm{N}, \langle \textsf{woman}\rangle\rangle, \langle \textbf{or}\rangle, \mathfrak{d}(\mathrm{N})\rangle\}\rangle$, $M_2 = \langle\emptyset, \{[x \text{ likes } y]\}\rangle$.



Reguła CR.OR-N przekształci ją na strukturę K', w której M_1 zostanie zastąpiona przez $M_1' = \langle\{y\}, \langle\emptyset, \{\langle \mathrm{N}(y), \langle \textsf{woman}\rangle\rangle\}\rangle \vee \langle\emptyset, \{\langle \mathrm{N}(y), \langle \mathrm{N}, \langle \textsf{child}\rangle\rangle, \langle[\text{who adores him}]\rangle\rangle\}\rangle\rangle$.



Ostatecznie otrzymujemy strukturę K'' prezentowaną graficznie poniżej.



3.4.5 Koniunkcja

Koniunkcja traktowana jest zazwyczaj jako operacja „symetryczna” do alternatywy, takie też są jej własności logiczne.[27]

Jednak język nie kieruje się takimi formalnymi zasadami. Po pierwsze, kierując się materiałem lingwistycznym, Kamp i Reyle uznali, że zbiory znaczników dyskursu ze wszystkich struktur będących elementami alternatywy są dla siebie nawzajem niewidoczne. Na przykład zdanie *Jones owns a book and he reads it.* zdaje się perfekcyjnie poprawne, a jego interpretacja semantyczna analogiczna do interpretacji ciągu zdań *Jones owns a book. He reads it.*.[28] Kierując się tym podobieństwem można uznać, że koniunkcja powinna być reprezentowana dokładnie w ten sam sposób co ciąg zdań. Nie powinno to dziwić, jako że poszczególne warunki występujące w danej strukturze powiązane są domyślnym operatorem koniunkcji (tak jak klauzule w Prologu lub ogólniej w dyzjunktywnej postaci normalnej).

Niestety, istnieje pewna zasadnicza różnica. Zdania dołączane są do struktury reprezentującej interpretację dyskursu pojedynczo, ale w całości, co dotyczy także koniunkcji zdań podrzędnych. Aby uniknąć akceptacji ewidentnie błędnych zdań typu *He owns a book and Jones reads it.* czy *Jones owns it and he reads a book.*, należy narzucić kolejność przetwarzania zdań składowych. Do tego właśnie służą indeksy wprowadzone w rozdziale 3.3.2.

Pewne wątpliwości budzą także zdania, w których następuje zmiana podmiotu, na przykład zdanie *Jones reads a book and Smith reads it.* nie wydaje się bardziej poprawne od swojego „dyzjunktowego” odpowiednika *Jones reads a book or Smith reads it.*. Dotyczy to także (może nawet wyraźniej) sekwencji zdań. *Jones reads a book. Smith reads it.*. Jeszcze bardziej oczywiste jest to dla zdania *Jones owns a book and Smith owns it.*, ale tu dochodzi wiedza o świecie, że ludzie zazwyczaj nie posiadają wspólnie przedmiotów. Jeśli chcemy zaznaczyć, że jest inaczej, podkreślamy to za pomocą specjalnych zwrotów (np. *Jones and Smith own a common book.*).

W zasadzie reprezentujący koniunkcję wyraz **and** może łączyć dokładnie te same frazy co reprezentujący alternatywę wyraz **or**. Jednak ponieważ w przypadku fraz rzeczownikowych wpływa to na atrybut *Num* czasownika, frazy takie omawiane są przy okazji rozważania problematyli liczby mnogiej w ogólności.

[27] Podobnie jak w przypadku będących ich odpowiednikami operatorów na zbiorach $\cup$ i $\cap$, wzajemna zamiana wszystkich wystąpień tych operatorów powoduje powstanie równoważnego systemu, oczywiście zamieniając także prawdę na fałsz i odwrotnie. Innymi słowy, jest to odwrócenie kraty boolowskiej „do góry nogami”.

[28] Pomijamy tu subtelną różnicę polegającą na tym, że umieszczenie całej informacji w jednym zdaniu ma podkreślać ich wzajemny związek. Takie subtelności są przez współczesne teorie semantyczne całkowicie ignorowane.

Reguła koniunkcji dla zdań

Podobnie jak w przypadku alternatywy zaczniemy od najprostszego przypadku, czyli koniunkcji całych zdań. Przyczyna wyróżnienia tego przypadku też jest ta sama: drzewo o korzeniu S nie może być własnym poddrzewem.

Definicja 3.34. (reguła **CR.AND-S**) Weźmy strukturę $K = \langle U_K, \mathrm{Con}_K\rangle$ taką, że $\langle \breve{S}, I\rangle \in \mathrm{Con}_K$ oraz $\breve{S} = \langle S, \mathfrak{d}(S)_1, \langle,\rangle, \langle \mathbf{S}, \mathfrak{d}(\mathbf{S})_\mathbf{2}, \langle,\rangle, ..., \langle \mathbf{S}, \mathfrak{d}(\mathbf{S})_{\mathbf{n-1}}, \langle \mathbf{and}\rangle, \mathfrak{d}(\mathbf{S})_\mathbf{n}\rangle ...\rangle\rangle$. Wówczas $\mathfrak{g}(K) = \langle U_{K'}, \mathrm{Con}_{K'}\rangle$ spełnia warunki:

1. $U_{K'} = U_K$,
2. $\mathrm{Con}_{K'} = \mathrm{Con}_K - \{\langle \breve{S}, I\rangle\} \cup \bigcup_{i=1}^{n} \{\langle \mathfrak{d}(S)_i, I \cup \{\langle \breve{S}, i\rangle\}\rangle\}$.

Przykład *3.32.* Jako przykład rozważymy całkiem złożone zdanie *If Jones likes her and he owns a Porsche, Mary borrows it.*. Po zastosowaniu (zgodnie z zasadą *od góry*), najpierw reguły CR.COND a następnie reguły CR.AND-S uzyskujemy strukturę K taką, że $U_K = \emptyset$, $\mathrm{Con}_K = \{L_1 \Longrightarrow L_2\}$, gdzie $L_1 = \langle \emptyset, \{\langle \text{[Jones likes her]}, \{\langle\delta, 1\rangle\}\rangle, \langle \text{[he owns a Porsche]}, \{\langle \delta, 2\rangle\}\rangle\}\rangle$, zaś $L_2 = \langle \emptyset, \{\langle \text{[Mary borrows it]}, \emptyset\rangle\}\rangle$, przy czym $\delta = $ [Jones likes her and he owns a Porsche].

$$\boxed{\boxed{\begin{array}{c}\langle\text{[Jones likes her]},\ \{\langle\delta,1\rangle\}\rangle\\ \langle\text{[he owns a Porsche]},\ \{\langle\delta,2\rangle\}\rangle\end{array}} \Longrightarrow \boxed{\langle\text{[Mary borrows it]},\ \emptyset\rangle}}$$

Zauważmy, że w chwili obecnej możemy analizować albo pierwszy warunek z poprzednika implikacji L_1, albo następnik L_2 (L_2 mogło zostać rozpatrzone przed zastosowaniem reguły CR.AND-S; efekt byłby taki sam). Natomiast drugi warunek z L_1 jest zablokowany. Oznacza to, że możemy zastosować regułę CR.PN do nazw własnych *Jones* oraz *Mary* (w dowolnej kolejności). Zaimek *him* może zostać zinterpetowany (za pomocą reguły CR.PRO) dopiero po nazwie własnej *Jones*, jednak przed bądź po *Mary*. Natomiast zaimek *it* z L_2 jest zależny od frazy rzeczownikowej *a Porsche* z zablokowanego warunku, więc na razie nie może być przetwarzany. W rezultacie uzyskujemy strukturę K' taką, że $U_{K'} = \{x, y\}$, $\mathrm{Con}_{K'} = \{L'_1 \Longrightarrow L'_2\}$, gdzie $L'_1 = \langle\{u\}, \{\mathbf{Jones}\ (x), u = y, \langle \text{[x likes u]}, \{\langle\delta, 1\rangle\}\rangle, \langle\text{[he owns a Porsche]}, \{\langle\delta, 2\rangle\}\rangle\}\rangle$, zaś $L'_2 = \langle\emptyset, \{\mathbf{Mary}\ (y), \langle\ [y \text{ borrows it}], \emptyset\rangle\}\rangle$.

$$\boxed{\begin{array}{c} x, y \\ \boxed{\begin{array}{c} u \\ \mathbf{Jones}\ (x) \\ u = y \\ \langle [x \text{ likes } u], \{\langle\delta,1\rangle\}\rangle \\ \langle\text{[he owns a Porsche]},\ \{\langle\delta,2\rangle\}\rangle \end{array}} \Longrightarrow \boxed{\begin{array}{c} \mathbf{Mary}\ (y) \\ \langle [y \text{ borrows it}],\ \emptyset\rangle \end{array}} \end{array}}$$

Po całkowitym opracowaniu pierwszego warunku z L_1 odblokowany zostaje drugi warunek. Możemy teraz zastosować regułę CR.PRO do zaimka *him* oraz reguły CR.ID i CR.LIN dla frazy *a Porsche* (w tej kolejności zgodnie z zasadą „od lewej”, choć to bez znaczenia). To z kolei umożliwia użycie reguły CR.PRO do zaimka *it* ze struktury L_2. W efekcie otrzymamy strukturę K'' taką, że $U_{K''} = \{x, y\}$, $\mathrm{Con}_{K''} = \{L''_1 \Longrightarrow L''_2\}$, gdzie $L''_1 = \langle\{u, v, w\}, \{\mathbf{Jones}\ (x), u = y, \langle [x \text{ likes } u]\rangle, \{\langle\delta, 1\rangle\}, w = x, \mathbf{Porsche}\ (v), \langle [w \text{ owns } v], \{\langle\delta, 2\rangle\}\rangle\}\rangle$, zaś $L''_2 = \langle\{z\}, \{\mathbf{Mary}\ (y), z = v, \langle \text{[y borrows v]}, \emptyset\rangle\}\rangle$.

$$\boxed{\begin{array}{c} x, y \\ \boxed{\begin{array}{c} u, v, w \\ \mathbf{Jones}\ (x) \\ u = y \\ \langle \text{[x likes u]}, \{\langle\delta,1\rangle\}\rangle \\ w = x \\ \mathbf{Porsche}\ (v) \\ \langle [w \text{ owns } v], \{\langle\delta,2\rangle\}\rangle \end{array}} \Longrightarrow \boxed{\begin{array}{c} z \\ \mathbf{Mary}\ (y) \\ z = v \\ \langle [y \text{ borrows } z], \emptyset\rangle \end{array}} \end{array}}$$

Reguła koniunkcji dla pozostałych fraz

Podobnie jak w przypadku dyzjunkcji, nie możemy po prostu rozdystrybuować fraz pomiędzy warunki, tylko powielić całe zdanie z „podmienioną” frazą. Zauważmy, że fraza czasownikowa VP budzi podobne wątpliwości jak wówczas (vide zdanie *Smiths do not (like Mary and hate John).*).

Definicja 3.35. (reguła **CR.AND-OTH**) Weźmy strukturę $K = \langle U_K, \mathrm{Con}_K\rangle$ taką, że $\langle \breve{S}, I\rangle \in \mathrm{Con}_K$ oraz $\breve{S} \ni \breve{S}^k = \langle X, \mathfrak{d}(X)_1, \langle,\rangle, \langle \mathbf{X}, \mathfrak{d}(\mathbf{X})_\mathbf{2}, \langle,\rangle, ..., \langle \mathbf{X}, \mathfrak{d}(\mathbf{X})_{\mathbf{n-1}}, \langle \mathbf{and}\rangle, \mathfrak{d}(\mathbf{X})_\mathbf{n}\rangle ...\rangle\rangle$, gdzie X jest symbolem terminalnym VP′ lub V. Wówczas $\mathfrak{g}(K) = \langle U_{K'}, \mathrm{Con}_{K'}\rangle$ jest strukturą spełniającą warunki:

1. $U_{K'} = U_K$,
2. $\breve{S}_i = \breve{S}[\breve{S}^k/\mathfrak{d}(X)_i]$,
3. $\mathrm{Con}_{K'} = \mathrm{Con}_K - \{\langle \breve{S}, I\rangle\} \cup \bigcup_{i=1}^{n} \{\langle \breve{S}_i, I \cup \{\langle \breve{S}, i\rangle\}\rangle\}$.

<u>Przykład</u> *3.33.* Weźmy zdanie *Jones owns a car and uses it and he owns a bicycle and does not use it.* zawierające wszystkie interesujące na zajawiska. Po aplikacji reguły CR.AND-S uzyskujemy strukturę K taką, że $U_K = \emptyset$, $\mathrm{Con}_K = \{\langle[\textsf{Jones owns a car and uses it}], \{\langle\delta, 1\rangle\}\rangle, \langle[\textsf{he owns a bicycle and does not use it}], \{\langle\delta, 2\rangle\}\rangle\}$, gdzie $\delta =$ [Jones owns a car and uses it and he owns a bicycle and does not use it].

$$\begin{array}{|c|}\hline \langle[\textsf{Jones owns a car and uses it}],\ \{\langle\delta,1\rangle\}\rangle \\ \langle[\textsf{he owns a bicycle and does not use it}],\ \{\langle\delta,2\rangle\}\rangle \\ \hline \end{array} \qquad \begin{array}{|c|}\hline \langle[\textsf{Jones owns a car}],\ \{\langle\delta,1\rangle, \langle\delta',1\rangle\}\rangle \\ \langle[\textsf{Jones uses it}],\ \{\langle\delta,1\rangle, \langle\delta',2\rangle\}\rangle \\ \langle[\textsf{he owns a bicycle and does not use it}],\ \{\langle\delta,2\rangle\}\rangle \\ \hline \end{array}$$

Zgodnie z zasadą indeksów możemy przetwarzać wyłącznie pierwszy warunek. Zastosowanie reguły CR.AND-OTH daje strukturę K' taką, że $U_{K'} = \emptyset$, $\mathrm{Con}_{K'} = \{\langle[\textsf{Jones owns a car}], \{\langle\delta, 1\rangle, \langle\delta', 1\rangle\}\rangle, \langle[\textsf{Jones uses it}], \{\langle\delta, 1\rangle, \langle\delta', 2\rangle\}\rangle, \langle[\textsf{he owns a bicycle and does not use it}], \{\langle\delta, 2\rangle\}\rangle\}$ (powyżej po prawej).
Obecnie nie tylko ostatni warunek jest zablokowany, ale także drugi. Zauważmy, że i tak nie możemy go przetwarzać, gdyż brak znacznika dyskursu do którego mógłby się odnieść zaimek *it*, ale gdyby zaimek był w pierwszym zdaniu, indeksy uniemożliwiłyby interpretację niepoprawnego zdania. Po standardowej analizie najpierw pierwszego, a potem drugiego warunku uzyskujemy strukturę K'' taką, że $U_{K''} = \{x, y, u\}$, $\mathrm{Con}_{K''} = \{\mathbf{Jones}\ (x), \mathbf{car}\ (y), \langle[x\ \textsf{owns}\ y], \{\langle\delta, 1\rangle, \langle\delta', 1\rangle\}\rangle, u = y, \langle[x\ \textsf{uses}\ u], \{\langle\delta, 1\rangle, \langle\delta', 2\rangle\}\rangle,$ $\langle[\textsf{he owns a bicycle and does notuse it}], \{\langle\delta, 2\rangle\}\rangle\}$, gdzie $\delta' =$ Jones owns a car and uses it. (poniżej po lewej).

$$\begin{array}{|c|}\hline x, y, u \\ \mathbf{Jones}\ (x) \\ \mathbf{car}\ (y) \\ \langle[x\ \textsf{owns}\ y], \{\langle\delta,1\rangle, \langle\delta',1\rangle\}\rangle \\ u = y \\ \langle[x\ \textsf{uses}\ u], \{\langle\delta,1\rangle, \langle\delta',2\rangle\}\rangle \\ \langle[\textsf{he owns a bicycle and does not use it}],\ \{\langle\delta,2\rangle\}\rangle \\ \hline \end{array} \qquad \begin{array}{|c|}\hline x, y, u \\ \mathbf{Jones}\ (x) \\ \mathbf{car}\ (y) \\ \langle[x\ \textsf{owns}\ y], \{\langle\delta,1\rangle, \langle\delta',1\rangle\}\rangle \\ u = y \\ \langle[x\ \textsf{uses}\ u], \{\langle\delta,1\rangle, \langle\delta',2\rangle\}\rangle \\ \langle[\textsf{he owns a bicycle}],\ \{\langle\delta,2\rangle, \langle\delta'',1\rangle\}\rangle \\ \langle[\textsf{he does not use it}],\ \{\langle\delta,2\rangle, \langle\delta'',2\rangle\}\rangle \\ \hline \end{array}$$

Wreszcie pojawia się możliwość analizy ostatniego warunku (reguła CR.AND-OTH), co daje strukturę K''' taką, że $U_{K'''} = \{x, y, u\}$, $\mathrm{Con}_{K'''} = \{\mathbf{Jones}\ (x), \mathbf{car}\ (y), \langle[x\ \textsf{owns}\ y], \{\langle\delta, 1\rangle, \langle\delta', 1\rangle\}\rangle, u = y, \langle[x\ \textsf{uses}\ u], \{\langle\delta, 1\rangle, \langle\delta', 2\rangle\}\rangle, \langle[\textsf{he owns a bicycle}], \{\langle\delta, 2\rangle, \langle\delta'', 1\rangle\}\rangle, \langle[\textsf{he does not use it}], \{\langle\delta, 2\rangle, \langle\delta'', 2\rangle\}\rangle\}$, gdzie $\delta' =$ he owns a bicycle and does not use it. (powyżej po prawej). Przetwarzając dwa ostatnie warunki będące rezultatem poprzedniej operacji w odpowiedniej kolejności uzyskamy strukturę finalną K'''' taką, że $U_{K''''} = \{x, y, z, u, v\}$, $\mathrm{Con}_{K''''} = \{\mathbf{Jones}\ (x), \mathbf{car}\ (y), \langle[x\ \textsf{owns}\ y], \{\langle\delta, 1\rangle, \langle\delta', 1\rangle\}\rangle, u = y, \langle[x\ \textsf{uses}\ u], \{\langle\delta, 1\rangle, \langle\delta', 2\rangle\}\rangle, \mathbf{bicycle}\ (z), v = x, \langle[v\ \textsf{owns}\ z], \{\langle\delta, 2\rangle, \langle\delta'', 1\rangle\}\rangle, \neg L\}$, gdzie $L = \langle\{v', w\}, \{v' = x, w = z, \langle[v'\ \textsf{uses}\ w], \{\langle\delta, 2\rangle, \langle\delta'', 2\rangle\}\rangle\}\rangle$ (graficzna postać tej struktury widoczna jest obok).

$$\begin{array}{|c|}\hline x, y, z, u, v \\ \mathbf{Jones}\ (x) \\ \mathbf{car}\ (y) \\ \langle[x\ \textsf{owns}\ y], \{\langle\delta,1\rangle, \langle\delta',1\rangle\}\rangle \\ u = y \\ \langle[x\ \textsf{uses}\ u], \{\langle\delta,1\rangle, \langle\delta',2\rangle\}\rangle \\ \mathbf{bicycle}\ (z) \\ v = x \\ \langle[v\ \textsf{owns}\ z], \{\langle\delta,2\rangle, \langle\delta'',1\rangle\}\rangle \\ \neg \begin{array}{|c|}\hline v', w \\ v' = x \\ w = z \\ \langle[v'\ \textsf{uses}\ w], \{\langle\delta,2\rangle, \langle\delta'',2\rangle\}\rangle \\ \hline \end{array} \\ \hline \end{array}$$

3.4.6 Niejednoznaczność zakresu

W językach formalnych, takich jak FOL i DRT, zakres działania poszczególnych operatorów i kwantyfikatorów jest jednoznaczny i prosty do określenia. Zakres działania spójników logicznych stanowią łączone przez nie formuły (w DRT — struktury). Kwestia staje się znacznie bardziej istotna w przypadku kwantyfikatorów, gdyż wiążą one wyłącznie zmienne znajdujące się w zakresie ich działania. Ponieważ w DRT kwantyfikatory występują jedynie implicite — poprzez umieszczenie określonego znacznika dyskursu w zbiorze znaczników wybranej struktury, Kamp i Reyle nie mówią o zakresie działalności kwantyfikatorów, tylko (równoważnie) o zakresie widzialności znaczników, stanowiony przez zbiory warunków wszystkich struktur podrzędnych DRS-u, w którym dany znacznik jest umiejscowiony.

W przeciwieństwie do systemów formalnych, sformułowania języka naturalnego nie są jednoznaczne pod względem zakresu działania występujących w nich fraz, jednak w wyniku zastosowania formalnych technik analizy ulegają nieuniknionemu ujednoznacznieniu. Część tych niejednoznaczności jest wykrywana już na poziomie rozbioru syntaktycznego — zdanie posiada kilka drzew rozbioru.[29]

Często jednak niejednoznaczność zdania ujawniana jest dopiero na poziomie analizy semantycznej — jedno drzewo rozbioru przekształcane jest na kilka nieizomorficznych struktur. W rozważanym w niniejszym omówieniu podstawowym fragmencie DRT dotyczy to wyłącznie fraz rzeczownikowych, gdyż jedynie one identyfikowane są przez znaczniki dyskursu. Ponieważ w DRT analizie podlega nie pojedyncze zdanie, lecz cały dyskurs, zakres widoczności znaczników, a więc i fraz rzeczownikowych, wykracza poza pojedyncze zdanie. W szczególności, ponieważ znaczniki reprezentujące nazwy własne umieszczane są w strukturze głównej, ich zakres widoczności obejmuje zawsze cały dyskurs, a więc wszystkie występujące w nim zależności znajdują się wewnątrz tego zakresu (a stąd i pozostałe frazy rzeczownikowe). Natomiast źródłem rzeczywistej niejednoznaczności są frazy tworzone przez rzeczownik poprzedzony rodzajnikiem.

Przy założeniu sugerowanej przez autorów kolejności analizy „od lewej do prawej” (innymi słowy „podmiot przed dopełnieniem”), w celu uwzględnienia niejednoznaczności niezbędne jest uzupełnienie reguł CR.EVR, CR.ID oraz CR.DD o ich odpowiedniki umożliwiające przesunięcie adekwatnych znaczników i warunków do struktury nadrzędnej (CR.EVR) bądź podrzędnej (CR.ID), dzięki czemu uzyskuje się zmianę zakresu widoczności znaczników (i działania „ukrytych” kwantyfikatorów).

Innym, prostszym rozwiązaniem jest poluzowanie (czy też wręcz rezygnacja z) narzuconej kolejności aplikacji reguł zgodnie z położeniem fraz w drzewie. Konieczność „odfiltrowania” błędnej interpretacji zdań typu *He loves a girl who hates Bill.* będąca skutkiem takiej decyzji pojawia się także w całkiem odmiennych konstrukcjach zdaniowych, jak *Bill employs a man who he thinks Mary does not like.*, gdzie miepoprawność powiązania zaimka *he* ze znacznikiem reprezentującym frazę *a man* nie wynika ani z położenia w zdaniu, ani z ograniczeń na zaimki niezwrotne. Tak więc mechanizmy zabezpieczające przed błędnym wiązaniem zaimków muszą tak czy inaczej zostać dołączone do zestawu reguł.

Dzięki zmianie kolejności aplikacji reguł, niektóre zdania z przykładów omówionych w poprzednich rozdziałach uzyskują dodatkowe interpretacje.

Przykład *3.34.* Zmiana kolejności użycia reguł CR.EVR i CR.ID w zdaniach *Every farmer beats a donkey.* oraz *A farmer beats every donkey.* z przykładu 3.22 (str. 60) daje początkowo struktury K_1^a i K_2^a,

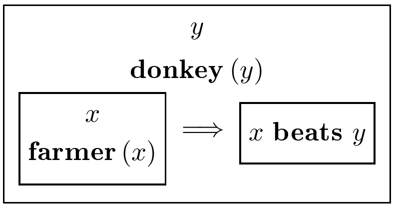
$$\boxed{\begin{array}{c} y \\ \mathrm{N}(y) \\ | \\ \textsf{donkey} \\ [\textsf{every farmer beats } y] \end{array}}$$

gdzie $U_{K_1^a} = \{y\}$, $\mathrm{Con}_{K_1^a} = \{\langle \mathrm{N}(y), \langle\textsf{donkey}\rangle\rangle,\ [\textsf{every farmer beats } y]\}$ (po lewej), oraz $U_{K_2^a} = \emptyset$, $\mathrm{Con}_{K_2^a} = \{L_{2,1}^a \Longrightarrow L_{2,2}^a\}$, mając $L_{2,1}^a = \langle\{y\}, \{\langle \mathrm{N}(y), \langle\textsf{donkey}\rangle\rangle\}\rangle$, $L_{2,2}^a = \langle\emptyset, \{[\textsf{a farmer beats } y]\}\rangle$ (po prawej).

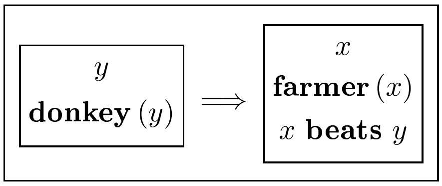
$$\boxed{\boxed{\begin{array}{c} y \\ \mathrm{N}(y) \\ | \\ \textsf{donkey} \end{array}} \Longrightarrow \boxed{[\textsf{a farmer beats } y]}}$$

Po dalszych przekształceniach uzyskujemy ostatecznie struktury $K_1^{a\prime}$, $K_2^{a\prime}$, gdzie $\mathrm{Con}_{K_1^{a\prime}} = \{Num(y), Gen(y), \textsf{donkey}(y), L_{1,1}^a \Longrightarrow L_{1,2}^a\}$, przy czym $L_{1,1}^a = \langle\{x\}, \{Num(x), Gen(x), \mathbf{farmer}\,(x)\}\rangle$, a $L_{1,2}^a =$

[29] Niejednokrotnie także zdania jednoznaczne mają kilka drzew rozbioru, przetwarzanych następnie na izomorficzne DRS-y (por. def. 3.13 w rozdz. 3.3.1), np. ciąg fraz połączonych spójnikiem *lub* (patrz przykł. 3.26 na str. 63).

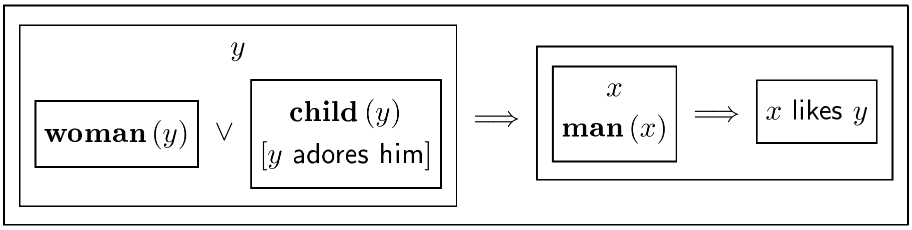


$\langle\emptyset, \{x\ beats\ y\}\rangle$ (po lewej). Natomiast $L^a_{2,1}$, $L^a_{2,2}$ zostają zastąpione przez $L^{a\prime}_{2,1} = \langle\{y\}, \{Num(y), Gen(y), \mathbf{donkey}(y)\}\rangle$, $L^{a\prime}_{2,2} = \langle\{x\}, \{Num(x), Gen(x), \mathbf{farmer}\,(x), x\ beats\ y\}\rangle$ (po prawej).

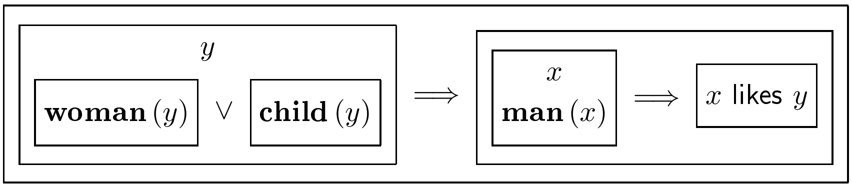


Czasem inne zjawiska lingwistyczne uniemożliwiają takie przestawienie.

Przykład *3.35.* W przypadku zdania *Every man likes every woman or child who adores him.* z przykładu 3.31 (str. 66), po zastosowaniu reguły CR.EVR w odwrotnej kolejności uzyskamy strukturę K^a, w której $U_K = \emptyset$, $\mathrm{Con}_K = \{L^a_1 \Longrightarrow L^a_2\}$, przy czym $L^a_1 = \langle\{y\}, \{\langle\emptyset, \{\mathbf{woman}\,(y)\}\rangle \vee \langle\emptyset, \{\mathbf{child}\,(y), [y\ \textsf{adores him}]\}\rangle\}\rangle$, a $L^a_2 = \langle\emptyset, \{\langle\{x\}, \{\mathbf{man}\,(x)\}\rangle \Longrightarrow \langle\emptyset, \{[x\ \textsf{likes}\ y]\}\rangle\}\rangle$.



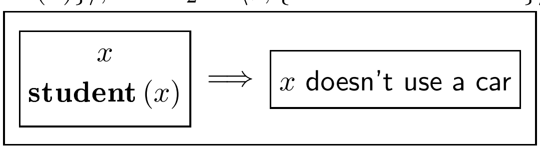
Niestety, znacznik x z którym można by powiązać zaimek *him* nie jest widoczny z poziomu struktury, w której zaimek ten się znajduje, więc cała interpretacja nie zostanie uznana. Jednak już tylko odrobinę prostsze zdanie *Every man likes every woman or child.* lub nawet zdanie, w którym zamienionoby zaimek *him* nazwą własną, posiada odpowiednią interpretację alternatywną przedstawioną graficznie powyżej.



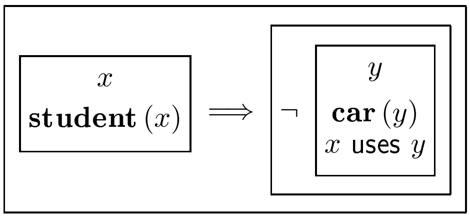
Należy tu jeszcze zauważyć, że w rachunku predykatów formuły $\forall x\ (A(x) \rightarrow (\forall y\ (B(y) \rightarrow C(x,y))$ i $\forall y\ (B(y) \rightarrow (\forall x\ (A(x) \rightarrow C(x,y))$ są równoważne, i podobnie jest w przypadku DRT (por. twierdz. 3.1 (iii). Oznacza to, że przestawianie kolejności kilkakrotnego stosowania reguły CR.EVR jest zbędne (pomijamy tu kwestię widoczności znaczników dyskursu). Ponieważ reguła CR.ID nie tworzy podstruktur, w jej przypadku zbędność przestawiania kolejności jej aplikacji jest banalnie oczywista.

Niejednoznaczność zakresu pojawia się rzecz jasna także w zdaniach zawierających negację.

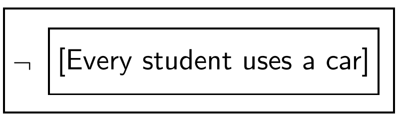
Przykład *3.36.* Rozważmy proste zanegowane zdanie *Every student doesn't use a car.*. Zaczynając od reguły CR.EVR uzyskujemy wpierw strukturę K taką, że $U_K = \emptyset$, $\mathrm{Con}_K = \{L_1 \Longrightarrow L_2\}$, gdzie $L_1 = \langle\{x\}, \{\mathbf{student}\,(x)\}\rangle$, zaś $L_2 = \langle\emptyset, \{x\ \textsf{doesn't use a car}\}\rangle$ (poniżej),

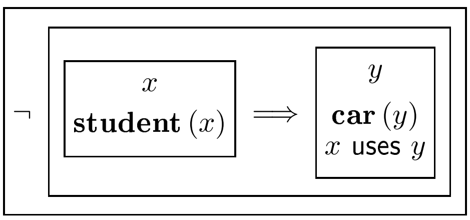


by ostatecznie otrzymać strukturę K'_a, przy czym L_2 zostaje zastąpione przez $L'_{2a} = \langle\emptyset, \{\neg\langle\{y\}, \{\mathbf{car}\,(y), x\ \textsf{uses}\ y\}\rangle\}\rangle$ (obok).

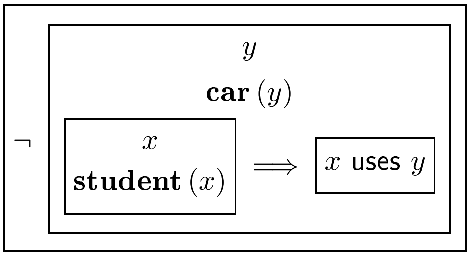


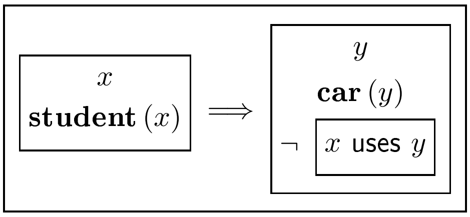
Aby zanalizować frazę czasownikową przed podmiotem zaczynamy od reguły CR.NEG uzyskując strukturę K^a taką, że $U_{K^a} = \emptyset$ a $\mathrm{Con}_{K^a} = \{\neg\langle\emptyset, \{\textsf{Every student uses a car}\}\rangle\}$ (poniżej). Ponieważ zanegowana struktura





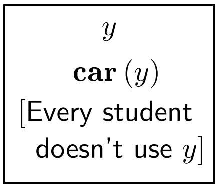
zawiera „zwykłe" zdanie oznajmujące, znowu mamy dwie możliwości. Startując od reguły CR.EVR otrzymamy strukturę K_a^a z $\text{Con}_{K_a^a} = \{\neg\langle\emptyset, \{\langle\{x\}, \{\textbf{student}\,(x)\}\rangle \Longrightarrow \langle\{y\}, \{\textbf{car}\,(y), x \text{ uses } y\}\rangle\}\rangle\}$ (po prawej u góry), natomiast rozpoczęcie od reguły CR.DD da w rezultacie strukturę K_b^a z $\text{Con}_{K_b^a} = \{\neg\langle\{y\}, \{\textbf{car}\,(y), \langle\{x\}, \{\textbf{student}\,(x)\}\rangle \Longrightarrow \langle\emptyset, \{x \text{ uses } y\}\rangle\}\rangle\}$ (po prawej u dołu).

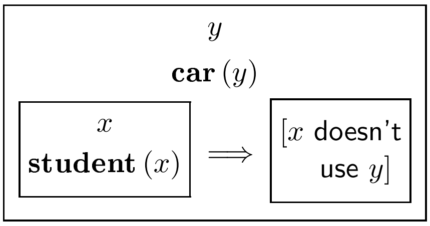


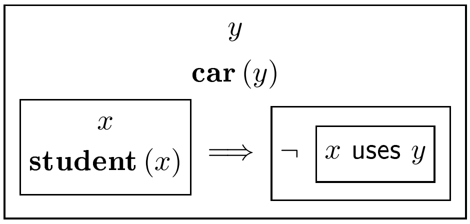


Pozostaje jeszcze kwestia analizowania dopełnienia przed orzeczeniem. Aplikacja reguły CR.ID bezpośrednio po regule CR.EVR oznacza przekształcenie struktury K w K_b', przy czym L_2 zostaje zastąpione przez $L_{2b}' = \langle\{y\}, \{\textbf{car}\,(y), \neg\langle\emptyset, \{x \text{ uses } y\}\rangle\}\rangle$ (obok po lewej).[30]

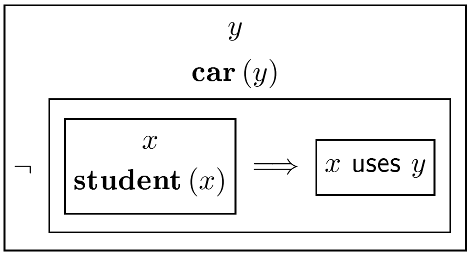
Natomiast rozpoczęcie od reguły CR.DD da strukturę K^b taką, że $U_{K^b} = \{y\}$ a $\text{Con}_{K^b} = \{\textbf{car}\,(y), \text{Every student doesn't use } y\}$ (po lewej). Z kolei reguła CR.EVR prowadzi do struktury K_a^b, gdzie $\text{Con}_{K_a^b} = \{\textbf{car}\,(y),\ L_{1a}^b \Longrightarrow L_{2a}^b\}$,







przy czym $L_{1a}^b = \langle\{x\}, \{\textbf{student}\,(x)\}\rangle$, zaś $L_{2a}^b = \langle\emptyset, \{x \text{ doesn't use } y\}\rangle$. Ostatecznie dzięki regule CR.NEG uzyskujemy strukturę $K_a^{b\prime}$, w której L_{2a}^b zastąpiono przez $L_{2a}^{b\prime} = \langle\emptyset, \{\neg\langle\emptyset, \{x \text{ uses } y\}\rangle\}\rangle$ (powyżej po prawej). Zato zastosowanie reguły CR.NEG przed CR.EVR przekształca strukturę K^b w K_b^b taką, że $\text{Con}_{K_b^b} = \{\textbf{car}(y), \neg\langle\emptyset, \{L_{1b}^b \Longrightarrow L_{2b}^b\}\rangle\}$, gdzie $L_{1b}^b = \langle\{x\}, \{\textbf{student}\,(x)\}\rangle$, zaś $L_{2b}^b = \langle\emptyset, \{x \text{ uses } y\}\rangle$ (obok).



Wydaje się oczywiste, że niektóre spośród sześciu możliwych interpretacji rozważanego zdania nie są logicznie niezależne. Kamp i Reyle rozważali inny, podobny przykład zanegowanego zdania *Every student doesn't know the answer.*. Uznając, że fraza *the answer* jednoznacznie identyfikuje obiekt, akceptowali jedynie struktury, w których fraza ta znajduje się w strukturze głównej (a więc odpowiedniki struktur $K_a^{b\prime}$ i K_b^b), stwierdzając jednocześnie, że K_b^b jest interpretacją preferowaną. Ponieważ jednak reguła CR.DD takiego podejścia nie uzasadnia, powyżej zaprezentowaliśmy mniej kontrowersyjne zdanie, nie decydując się oceniać, która z powyższych interpretacji jest preferowana, a i tak istnieje obawa, że nie wszystkie powyższe interpretacje są lingwistycznie uzasadnione.

Niejednoznaczność zakresu dotyczy także zdań złożonych (np. *If John likes Mary, then Susan doesn't like her and Fred doesn't like John.*) oraz wykraczających poza akceptowaną składnię zdań z (okolicznikami), jak słynne *John saw a man in a park with a telescope.*. W obu wypadkach jednak niejednoznaczność jest odwzorowywana już za pomocą różnych drzew rozbioru.

W niniejszym opracowaniu mnogość zjawisk lingwistycznych interpretowanych w teorii DRT została ledwie zasygnalizowana. W szczególności, nie zostały omówione ani kwestie związane z liczbą mnogą, ani z czasem i aspektem. Jednak mamy wrażenie, że nawet to wstępne studium pokazuje siłę i elastyczność tego formalizmu, nie pozbawionego rzecz jasna pewnych ograniczeń.

[30] Niniejsze omówienie nie obejmuje kwestii czasu gram. (ang. *tense*) i aspektu, a więc w przypadku zdań oznajmujących analiza „orzeczenia" jako takiego jest pomijana. Ponieważ jednak rozpatrywanie dopełnienia przed podmiotem *de facto* oznacza przekształcanie go i przed orzeczeniem, więc powinno to także dotyczyć zdań przeczących.

4 Semantyka oparta na zasobach leksykalnych

Ostatnim formalizmem, jaki zamierzamy omówić w niniejszym opracowaniu jest *Semantyka oparta na zasobach leksykalnych* (ang. *Lexical Resource Semantics* (LRS). Prace prowadzące do takiego podejścia do problematyki semantycznej rozpoczęte zostały pod koniec lat 90-tych, czego efektem był cały ciąg rozwiązań. Jednak aktualny format LRS sformułowany został przez Manfreda Sailera i Franka Richtera dopiero w 2001. Jest to więc podejście zupełnie nowe i wciąż intensywnie rozwijane. W przeciwieństwie do rozwiązań omawianych w poprzednich rozdziałach (przede wszystkim DRT), formalizm ten stworzony został w celu reprezentowania semantyki j. naturalnego, którego składnia zapisana została w języku RSRL (por. rozdz. 4.3) stanowiącego formalną podstawę HPSG i w tej postaci stanowi on integralną część tej gramatyki, a więc *ex definitione* nie nadaje się do opisu semantyki języka opisanego za pomocą innego formalizmu gramatycznego.

4.1 Logika Ty2 — język opisu semantyki

Richter i Sailer zdecydowali, że formalizmem logicznym w którym zapisywana będzie informacja semantyczna wydobyta ze sformułowań j. naturalnego opisynych w gramatyce HPSG będzie logika Ty2, zaproponowana przez Gallina (1975). Podejście to należy do szerokiego nurtu formalizmów semantycznych wywodzących się bezpośrednio z wprowadzonego przez Montague języka IL omówionego w rozdz. 2.1. Nie znaczy to bynajmniej, że koncepcje leżące u podstaw LRS nie dają się zapisać w innych formalizmach o podobnej mocy.

Niniejsze omówienie oparte jest na wersji Ty2 zaprezentowanej w [Sailer, 2000]. Jednak dla uzyskania jednolitości opracowania pominiemy niektóre wprowadzone tam czysto notacyjne zmiany, zachowując notację z rozdz. 2.

4.1.1 Syntaktyka

Definicja 4.1. Zbiór typów $\mathcal{T}$ jest to najmniejszy zbiór taki, że:

- $t, e, s \in \mathcal{T}$ (typy proste);
- jeżeli $\tau, \tau' \in \mathcal{T}$, to $\langle \tau, \tau' \rangle \in \mathcal{T}$.

Jedyna różnica w stosunku do definicji 2.1 dla IL Montague polega na tym, że s jako takie także zostaje uznane za typ prosty.

Zamiast wyrażeń znaczących rozważanych przez Montague (def. 2.2) autorzy używają bardziej tradycyjnego pojęcia *termu*.

Definicja 4.2. Niech $\tau, \tau' \in \mathcal{T}$ będą dowolnymi typami. Zbiór *termów* $\mathfrak{T}$ jest to suma (najmniejszych) takich zbiorów $\mathfrak{T}_\tau$, dla dowolnego $\tau \in \mathcal{T}$, że:

- niepusty, przeliczalny zbiór zmiennych $V_\tau \subseteq \mathfrak{T}_\tau$;
- niepusty, skończony zbiór stałych $C_\tau \subseteq \mathfrak{T}_\tau$;
- wyróżniona zmienna $\tilde{v}_s \in V_s$ (oznaczana także symbolem @);
- jeżeli $\varphi \in \mathfrak{T}_{\langle \tau, \tau' \rangle}$ oraz $\psi \in \mathfrak{T}_\tau$, to $\varphi(\psi) \in \mathfrak{T}_{\tau'}$;
- jeżeli $\varphi \in \mathfrak{T}_\tau$ oraz $v \in V_{\tau'}$, to $\lambda v \varphi \in \mathfrak{T}_{\langle \tau', \tau \rangle}$;
- jeżeli $\varphi, \psi \in \mathfrak{T}_\tau$, to $\varphi = \psi \in \mathfrak{T}_t$.

Zbiór wszystkich zmiennych będziemy oznaczać przez $V = \bigcup_{\tau \in \mathcal{T}} V_\tau$, zaś zbiór wszystkich stałych przez $C = \bigcup_{\tau \in \mathcal{T}} C_\tau$.

Zbiór stałych jest skończony zgodnie z wymaganiami stawianymi przez metodę kodowania Ty2 w logice HPSG: dopuszcza ona jedynie skończony zbiór funkcji, a stałe to przecież właśnie funkcje zeroargumentowe.

Za [Gallen, 1975; Sailer, 2000] prezentujemy poniżej definicje wszystkich spójników logicznych oraz kwantyfikatorów. Ich zgodność ze standardową intepretacją pokazana jest w tych pracach.
Niech $\varphi, \psi \in \mathfrak{T}_t$, $f \in \mathfrak{T}_{\langle t,t\rangle}$ oraz $x_\tau \in \mathfrak{T}_\tau$ dla $\tau \in \mathcal{T}$. Wówczas:

$$
\begin{array}{llllll}
\top & \equiv_{def} & \lambda x_t\ [x_t] = \lambda x_t\ [x_t] & \varphi \ \rightarrow \ \psi & \equiv_{def} & [\varphi \,\&\, \psi] = \varphi \\
\bot & \equiv_{def} & \lambda x_t\ [x_t] = \lambda x_t\ [\top] & \varphi \ \vee \ \psi & \equiv_{def} & \neg\varphi \ \rightarrow \ \psi, \\
\neg\, \varphi & \equiv_{def} & \varphi = \bot & \forall x_\tau\, \varphi & \equiv_{def} & \lambda x_\tau[\varphi] = \lambda x_\tau[\top] \\
\varphi \,\&\, \psi & \equiv_{def} & \lambda f\ [f(\varphi) = \psi] = \lambda f\ [f(\top)] & \exists x_\tau\ \varphi & \equiv_{def} & \neg\forall x_\tau\ \neg\varphi
\end{array}
$$

4.1.2 Semantyka

Podstawą teoriomodelowej interpretacji semantycznej jest zazwyczaj pewna algebra. Sailer (2000) używa terminu *rama* (ang. *frame*) i definiuje ją w następujący sposób.

Definicja 4.3. Niech A będzie dowolnym niepustym zbiorem indywiduów (obiektów), zaś W — niepustym zbiorem możliwych światów. Wówczas *ramą* nazywamy zbiór $\mathcal{F} = \bigcup_{\tau\in\mathcal{T}} D_{A,W,\tau}$ taki, że:

- $D_{A,W,t} = \{0, 1\}$,
- $D_{A,W,e} = A$,
- $D_{A,W,s} = W$,
- $D_{A,W,\langle\tau,\tau'\rangle} = D_{A,W,\tau'}^{D_{A,W,\tau}}$ dla dowolnych $\tau, \tau' \in \mathcal{T}$.

Modelem M nazywamy parę $\langle\mathcal{F}, I\rangle$, gdzie $\mathcal{F}$ jest ramą dla pewnych A, W, zaś I jest interpretacją stałych, tzn.

$$I(c) \in D_{A,W,\tau} \quad \text{dla dowolnego } c \in C_\tau.$$

Oznacza to, że I przyporządkowuje stałym typu e obiekty z A, stałym typu s światy z W, stałym typu t 0 bądź 1, a stałym przynależnym do typów złożonych (symbolom predykatywnym) relacje odpowiedniego typu.

Definicja 4.4. Niech $M = \langle\mathcal{F}, I\rangle$ będzie dowolnym modelem. *Wartościowaniem* nazywamy pewną funkcję $\mathfrak{v} = \bigcup_{\tau\in\mathcal{T}} \mathfrak{v}_\tau : V_\tau \longrightarrow D_{A,W,\tau}$. Zbiór wszystkich wartościowań to $\mathfrak{V}(M) \subseteq \mathcal{F}^V$. Ponadto przez $\mathfrak{v}_x^d$ będziemy oznaczać wartościowanie uzyskane z $\mathfrak{v}_x^d$ przez zastąpienie wartości zmiennej x obiektem $d \in D_{A,W,\tau}$.

Definicja 4.5. Niech $\tau, \tau' \in \mathcal{T}$ będą dowolnymi typami. *Wartością semantyczną*[31] wyrażenia $\varphi_\tau \in \mathfrak{T}_\tau$ w modelu M przy wartościowaniu $\mathfrak{v}$ (piszemy $[\![\varphi_\tau]\!]^{M,\mathfrak{v}}$) jest:

$$
\begin{array}{lcll}
[\![c_\tau]\!]^{M,\mathfrak{v}} & = & I(c_\tau) & \text{dla dowolnej stałej } c_\tau \in C_\tau; \\
[\![v_\tau]\!]^{M,\mathfrak{v}} & = & \mathfrak{v}_\tau(v_\tau) & \text{dla dowolnej zmiennej } v_\tau \in V_\tau; \\
\left[\!\left[\varphi_{\langle\tau,\tau'\rangle}(\psi_\tau)\right]\!\right]^{M,\mathfrak{v}} & = & \left[\!\left[\varphi_{\langle\tau,\tau'\rangle}\right]\!\right]^{M,\mathfrak{v}} ([\![(\psi_\tau)]\!]^{M,\mathfrak{v}}) & \text{dla dowolnych } \varphi_{\langle\tau,\tau'\rangle} \in \mathfrak{T}_{\langle\tau,\tau'\rangle},\ \psi_\tau \in \mathfrak{T}_\tau; \\
[\![\lambda v_\tau \varphi_{\tau'}]\!]^{M,\mathfrak{v}} & = & f\colon D_{A,W,\tau} \longrightarrow D_{A,W,\tau'} & \text{gdzie } f(d) = [\![\varphi_{\tau'}]\!]^{M,\mathfrak{v}_{v_\tau}^d} \text{ dla dowolnych} \\
 & & & \varphi_{\tau'} \in \mathfrak{T}_{\tau'} \text{ oraz } v_\tau \in V_\tau; \\
[\![\varphi_\tau = \psi_\tau]\!]^{M,\mathfrak{v}} & = & 1 \quad \text{wtw, gdy} \quad [\![\varphi_\tau]\!]^{M,\mathfrak{v}} & \text{jest równe } [\![\psi_\tau]\!]^{M,\mathfrak{v}} \text{ dla dowolnych } \varphi_\tau, \psi_\tau \in \mathfrak{T}_\tau.
\end{array}
$$

Sailer (2000) prezentuje także translację logiki Montague IL na logikę Gallina Ty2. Poniżej przedstawimy tę translację, co pozwoli nam uwydatnić podobieństwa i różnice pomiędzy tymi dwoma formalizmami. Na wstępie należy zauważyć, że translacja taka jest możliwa tylko w tą jedną stronę, gdyż zbiór typów w IL jest węższy niż w przypadku Ty2, jako że typy postaci $\langle\tau, s\rangle$ w IL nie występują (także samo s nie jest w IL typem prostym). Z tym zastrzeżeniem możemy traktować zbiór typów w IL jako podzbiór zbioru typów Ty2 bez konieczności dokonywania translacji. Podobne założenie można przyjąć

[31] Sailer (2000) używa terminu *extension* (rozszerzenie). Ponieważ termin ten używany jest w rozdz. 3 w zupełnie innym znaczeniu, postanowiliśmy posłużyć się tym samym określeniem co w omówieniu semantyki Montague w rozdz. 2.

dla zmiennych i stałych, z tym że musimy ograniczyć się do wersji IL o skończonej liczbie stałych (co jest ograniczeniem czysto technicznym). Z drugiej strony, ponieważ możliwe światy nie posiadają składnika temporalnego (nie są to uporządkowane *historie* złożone z punktów czy przedziałów, nie jest im też w żaden sposób przypisywany „czas wystąpienia" (poprzez rzutowanie na oś czasu), itp.), musimy ograniczyć się do okrojonej wersji IL, gdyż nie ma możliwości translacji operatorów $\mathbf{F}$, $\mathbf{P}$. Przy tych założeniach, dla dowolnego $\tau \in \mathcal{T}$, *translacja* z IL na Ty2 jest to przekształcenie $\Theta\colon \mathcal{E} \longrightarrow \mathfrak{T}$ przyporządkowujące każdemu wyrażeniu znaczącemu z $\mathcal{E}$ (por. def. 2.2) term z $\mathfrak{T}$ (por. def. 4.2) w następujący sposób:

- $\Theta(v_\tau) \ = \ v_\tau$;
- $\Theta(c_e) \ = \ c_e$;
- $\Theta(c_\tau) \ = \ c_{\langle s,\tau\rangle}(\tilde{v}_s)$ dla $\tau \neq e$;
- $\Theta(\varphi_{\langle\tau,\tau'\rangle}(\psi_\tau)) \ = \ \Theta(\varphi_{\langle\tau,\tau'\rangle})\,(\Theta(\psi_\tau))$;
- $\Theta(\lambda v_\tau \varphi'_\tau) \ = \ \lambda\Theta(v_\tau)\Theta(\varphi'_\tau)$;
- $\Theta(\varphi_\tau =_{\mathrm{IL}} \psi_\tau) \ = \ \Theta(\varphi_\tau) =_{\mathrm{Ty2}} \Theta(\psi_\tau)$;
- $\Theta({}^\wedge\varphi_\tau) \ = \ \lambda\tilde{v}_s[\Theta(\varphi_\tau)]$;
- $\Theta({}^\vee\varphi_{\langle s,\tau\rangle}) \ = \ \Theta(\varphi_{\langle s,\tau\rangle})(\tilde{v}_s)$;
- $\Theta(\Box\varphi_\tau) \ = \ \forall\tilde{v}_s\ \varphi_\tau$.

Analizując powyższą translację, musimy zwrócić uwagę na kilka istotnych jej cech. Po pierwsze, translacjami stałych „nieindywiduowych" nie są stałe tego samego typu τ (choć takowe w języku istnieją), lecz wyrażenia złożone ze stałej typu $\langle s,\tau\rangle$ i zmiennej $\tilde{v}_s$, które w całości też oczywiście są typu τ. Istotne jest to, że interpretacja takich dwóch wyrażeń w modelu M nie jest identyczna, gdyż $[\![c'_\tau]\!]^{M,\mathfrak{v}}$ jest ustalone w modelu, gdy tymczasem $\left[\!\left[c_{\langle s,\tau\rangle}(\tilde{v}_s)\right]\!\right]^{M,\mathfrak{v}}$ może ulegać zmianie w zależności od świata reprezentowanego przez $\tilde{v}_s$ (czyli „bieżącego"). Ponadto, zgodnie z intuicją, translacją ${}^\vee\varphi_{\langle s,\tau\rangle}$ jest wartość przyjmowana przez $\varphi_{\langle s,\tau\rangle}$ w świecie bieżącym. Natomiast $\Theta({}^\wedge\varphi_\tau)$ interpretowana jest zgodnie z oczekiwaniem wyłącznie wtedy, gdy φ_τ zawiera zmienną odwołującą się do bieżącego świata $\tilde{v}_s$ (tylko wówczas zmiana wartości tej zmiennej będzie miała wpływ na zmianę wartości całego wyrażenia, zgodnie z definicją interpretacji dla λ-wyrażenia; w przeciwnym razie wartość tej funkcji będzie stała — będzie nią $[\![\varphi_\tau]\!]^{M,\mathfrak{v}}$).

Sailer (2000) nie przytoczył translacji dla operatora $\Box$. Została ona sformułowana na potrzeby tego opracowania na podobieństwo translacji dla ${}^\wedge$, jest jednak jeszcze bardziej ułomna, gdyż $\tilde{v}_s$ nie tylko musi występować w φ_τ, lecz musi być w tej formule zmienną wolną.

IL Montague jest ewidentnie logiką modalną.[32] Marczewski (1987, s. 311) przytacza definicję funktora (operatora) modalnego: **F** *jest funktorem modalnym działającym na zdanie* φ *wtedy i tylko wtedy, gdy wartość logiczna zdania* **F** φ *nie jest określona (a przynajmniej nie w sposób całkowity) przez wartość logiczną zdania* φ. Operatory $\Box$, $\mathbf{F}$, $\mathbf{P}$ oraz ${}^\wedge$ spełniają tę definicję (ale ${}^\vee$ rzecz jasna już nie). Tymczasem operator λ nie ma charakteru modalnego, a więc w logice Ty2 mamy do czynienia jedynie z „klasycznymi" operatorami.

Oczywiście translacji logiki modalnej na logikę klasyczną dokonywano nieraz (choć niektóre formuły modalnej logiki zdań dają się przetłumaczyć dopiero na rachunek predykatów II rzędu, np. dedekindowska ciągłość porządku da się wyrazić w zwykłej zdaniowej *tense logic* (por. [van Benthem, 1983a])). Znajdowaniem klasycznych odpowiedników formuł modalnych zajmuje się *teoria korespondencji* [Szałas, 1992; van Benthem, 1983b; 1984]. Standardowa translacja aletycznego operatora modalnego $\Box p$ jest następująca: $\forall P\ \forall x, y\ (R(x,y) \rightarrow P(x))$, gdzie R jest pewną wybraną relacją. Jest to rzecz jasna formuła II rzędu, jednak istnieją metody eliminacji zmiennych predykatywnych [Szałas, 1992].

[32] Określenia ≫modalny≪ i ≫intensjonalny≪ są niejednokrotnie uważana za równoznaczne, por. [Marczewski, 1987, dział G].

Z drugiej strony semantyka możliwych światów Kripke'go dla logik modalnych pozwala nam spojrzeć na tę kwestię z innej strony. Otóż zbiór możliwych światów służy do interpretacji operatorów modalnych. W języku logiki nie ma bezpośrednich odwołań do światów za pomocą wybranych zmiennych czy stałych. Tak też jest i w IL. s nie jest typem, nie ma więc zmiennych ani stałych tego typu, za to interpretacja semantyczna dokonywana jest względem nich ($[\![\alpha]\!]^{\mathcal{M},\mathtt{w},\mathtt{t},g}$). Natomiast w Ty2 rzecz się ma wręcz przeciwnie. Są stałe i zmienne typu s występujące bezpośrednie w formułach, brak natomiast operatorów modalnych...

Podejście takie nie jest niczym zaskakującym. Odwoływanie się bezpośrednie do obiektów o specjalnym znaczeniu, na przykład do momentów czy przedziałów czasu w logikach czasu[33] występuje w wielu rozwiązaniach. Formalizmy takie mogą zawierać specjalne operatory prawdziwościowe (tzw. *logiki zreifikowane*, ang. *reified logics*), których argumentami jest formuła logiczna oraz świat (moment/przedział czasu), w którym formuła ta ma zachodzić [Allen, 1984; McDermott, 1982; Shoham, 1987]. Możliwe jest także wprowadzenie specjalnych, wyróżnionych argumentów bezpośrednio do symboli predykatywnych [Haugh, 1987, Bacchus i in., 1989].

W obu przypadkach teoriomodelowa semantyka takich formalizmów oparta jest na algebrze dwu- lub więcej rodzajowej (ang. *two-sorted, many-sorted*); podobnie zresztą jak w przypadku logik modalnych. Niektóre z pośród tych rodzajów mają przypisany specjalny status, tzn. obiekty tego rodzaju wiązane są przez z góry określone, nieprzypadkowe relacje (np. porządek dla punktów czasu).

Logika Ty2 jest bliższa drugiemu z tych podejść — operatory prawdziwościowe ewidentnie w niej nie występują. Dwa typy proste s i e (poza prawdziwościowym typem t) przekładają się na dwurodzajowość modelu (dziedziny „proste" D i W). Jednak typ s nie ma w żaden sposób przyznanego specjalnego statusu. Może występować w dowolnych miejscach w typach złożonych, a więc i nie ma ograniczeń na występowanie wyrażeń tego typu w wyrażeniach złożonych. Innymi słowy, wyrażenia typu s i e traktowane są w sposób symetryczny.

Ma to oczywiście swoje wady i zalety. Możliwość traktowania wyrażeń typu s w sposób dowolny rodzi zagrożenie uznania elementów zbioru W za zwykłe obiekty (jak w standardowej algebrze dwurodzajowej, gdzie np. elementy D to ludzie, zaś elementy W to przedmioty). Jednak taka elastyczność daje nam zarazem możliwość zastosowania języka będącego w naszym posiadaniu do konkretnych potrzeb. Na przykład wyrażenia typu $\langle s, s\rangle$ mogą definiować funkcje (przejść), zaś typu $\langle s, \langle s, t\rangle\rangle$ relacje (dostępności) wiążące pary światów. Bez trudu możemy wówczas narzucić, by każda taka relacja była relacją równoważności.

4.2 Head-driven Phrased Structured Grammar

Tak jak w poprzednich rozdziałach, w celu opisania danego formalizmu semantycznego niezbędne jest przedstawienie gramatyki, w której zamierzamy reprezentować syntaktyczne rozbiory zdań będące podstawą analizy semantycznej. Ponieważ jednak LRS jest podejściem zaprojektowanym specjalnie dla HPSG i jest analogiem jej części semantycznej (tzn. funkcjonuje w taki sam sposób i w tym znaczeniu stanowi jego integralną część), musimy omówić tę gramatykę nieco dokładniej.

HPSG (*Head-driven Phrased Structured Grammar*) [Pollard i Sag, 1987; 1994] jest *generatywną teorią lingwistyczną o podstawach formalnych należącą do rodziny tzw. formalizmów opartych na ograniczeniach (ang. constraint-based formalisms)*[34], podobnie jak na przykład LFG. Według tych teorii *wypowiedzenie* (a właściwie pewna odpowiadająca mu struktura) spełnia jednocześnie wszystkie ograniczenia (reguły, zasady) nakładane na nie przez gramatykę. Formalna teoria HPSG składa się z *sygnatury* i *teorii właściwej*. Właściwa teoria HPSG to w przybliżeniu zbiór ograniczeń, czyli formuł opisujących własności modelowanych obiektów. Sygnatura natomiast opisuje, jakie obiekty mogą wchodzić w skład modelu, to jest o jakich obiektach lingwistycznych mówi teoria (czasownik, rzeczowniki itp., przypadki, rodzaje, itp. itd.), oraz jakie podstawowe cechy mają te obiekty (np. rzeczowniki odmieniają się przez przypadki, ale nie przez osoby, itp.).

[33] Termin *logika temporalna* kojarzy się z formalizmem z modalnymi operatorami temporalnymi; określenie *logika czasu* zdaje się mniej nacechowane.

[34] Niniejszy rozdział opracowany został w oparciu pracę [Przepiórkowski i in., 2002].

4.2.1 Sygnatura

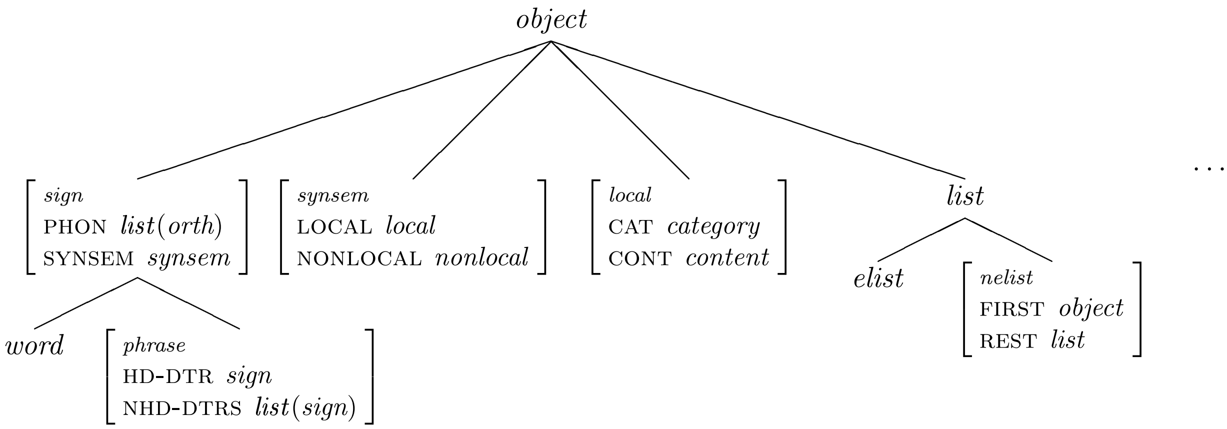
Zaczniemy od opisu sygnatury. Formalnie rzecz biorąc, jest to częściowy porządek na zbiorze *rodzajów* (ang. *sorts*),[35] określającym jakie rodzaje obiektów są akceptowane w rozważanej teorii. Jednocześnie definiuje ona *atrybuty* charakterystyczne dla tych obiektów.

I tak, w HPSG przyjmuje się, że wszystkie wyrażenia lingwistyczne (wyrazy, frazy, zdania) są bilateralnymi *znakami* (ang. *signs*) rozumianymi w sensie de Saussure'a (1959). Zgodnie z tym, jednym z rodzajów definiowanych w sygnaturze jest *sign*. Ponieważ wyrażenia lingwistyczne posiadają cechy fonologiczne i składniowo-semantyczne, przyjmuje się, że obiekty rodzaju *sign* mają (co najmniej) dwa atrybuty, PHONOLOGY (w skrócie: PHON) i SYNTAX-SEMANTICS (w skrócie: SYNSEM), których wartości odzwierciedlają te cechy. Graficzny zapis tej informacji prezentowany jest obok.

$$\begin{bmatrix} \textit{sign} \\ \text{PHON} \\ \text{SYNSEM} \end{bmatrix}$$

Obiekty rodzaju *sign* reprezentujące wyrazy różnią się od obiektów rodzaju *sign* reprezentujących frazy i zdania: te ostatnie posiadają wewnętrzną strukturę składniową, której nie posiadają wyrazy. W szczególności, wszystkie frazy posiadają *element główny* (ang. *head*) zawierający podstawowe cechy morfoskładniowe tej frazy (np. przypadek rzeczownika odzwierciedla przypadek całej frazy rzeczownikowej, której elementem głównym jest dany rzeczownik, itp.). Z kolei *list* jest rodzajem list obiektów (dla atrybutu NHD-DTRS (skrót od NON-HEAD-DAUGHTERS), obiektów rodzaju *sign*). Przyjmuje się, że rodzaj *list* posiada dwa podrodzaje, *elist* i *nelist*, odpowiadające listom pustym i listom niepustym.

Wstępna wersja sygnatury przedstawiona jest na rys. 4.1.[36]



Rysunek 4.1: Podstawowy fragment hierarchii typów gramatyki HPSG przyjętej w niniejszym rozdziale

W notacji przyjętej w HPSG, jeżeli rodzaj a jest położony poniżej rodzaju b i połączony z nim linią, to jest on *podrodzajem* rodzaju b. Tak więc rodzaje *word* i *phrase* są jedynymi podrodzajami rodzaju *sign*. Oznacza to, że każdy obiekt rodzaju *sign* należy albo do rodzaju *word*, albo do rodzaju *phrase* (ale nie do obu na raz). Ponadto atrybuty zdefiniowane dla danego rodzaju są „*dziedziczone*" (ang. *inherited*) przez każdy jego podrodzaj. Wynika z tego, że każdy obiekt rodzaju *word* i każdy obiekt rodzaju *phrase* posiada atrybuty PHON i SYNSEM. Ponadto obiekty rodzaju *phrase* posiadają także atrybuty HD-DTR (od HEAD-DAUGHTER) i NHD-DTRS określające składniki bezpośrednie danej frazy.

Fragment sygnatury z rys.4.1 opisuje tylko niektóre z podrodzajów rodzaju *object*; brak tutaj rodzajów *nonlocal*, *category* itp. Niemniej jednak już ten niewielki fragment sygnatury pokazuje, że obiekty lingwistyczne w rozumieniu HPSG mogą posiadać skomplikowaną strukturę wewnętrzną. Na przykład obiekty rodzaju *sign* posiadają atrybut SYNSEM, którego wartościami są obiekty rodzaju *synsem*.

$$\begin{bmatrix} \textit{sign} & \\ \text{PHON} & \textit{list}(\textit{orth}) \\ \text{SYNSEM} & \begin{bmatrix} \textit{synsem} & \\ \text{LOCAL} & \begin{bmatrix} \textit{local} & \\ \text{CAT} & \textit{category} \\ \text{CONT} & \textit{content} \end{bmatrix} \\ \text{NONLOCAL} & \textit{nonlocal} \end{bmatrix} \end{bmatrix}$$

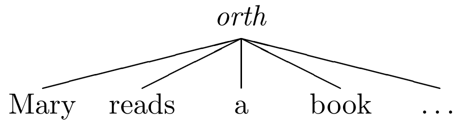
[35]Przepiórkowski i in. (2002) używają terminu *typ*. Ponieważ jednak termin ten używany jest na oznaczenie ang. terminu *type*, który bynajmniej nie jest tożsamy z określeniem *sort*, w niniejszym opracowaniu będziemy używać często spotykanego, choć może nieco kontrowersyjnego, terminu *rodzaj*.

[36]W HPSG stosunkowo często nadaje się taką samą nazwę atrybutowi i rodzajowi wartości tego atrybutu, np. SYNSEM i *synsem* oraz LOCAL i *local* na rys. 4.1. Nie powinno to być mylące, jako że atrybuty i rodzaje różnią się typograficznie: atrybuty wyróżnione są KAPITALIKAMI, zaś rodzaje — *kursywą*.

Te obiekty mają z kolei atrybuty LOCAL i NONLOCAL, których wartościami są odpowiednio obiekty rodzaju *local* i *nonlocal*, przy czym obiekty rodzaju *local* mają kolejne dwa atrybuty, CAT (skrót od CATEGORY) i CONT (CONTENT). Z powyższego opisu wynika, że każdy obiekt rodzaju *sign* posiada przynajmniej strukturę zaprezentowaną powyżej po prawej. W przypadku bardziej skomplikowanych struktur nazwy te podlegają czasem dalszemu skracaniu.

W rzeczywistości ta struktura jest znacznie bogatsza, gdyż również obiekty każdego z rodzajów *nonlocal*, *category* i *content* posiadają wewnętrzną strukturę.

Każdy atrybut posiada pewną wartość będącą również obiektem pewnego rodzaju. Na przykład wartościami atrybutu PHON są obiekty rodzaju *list*(*orth*), a wartościami atrybutu HD-DTR są obiekty rodzaju *sign*. Wszystkie rodzaje pojawiające się jako wartości atrybutów, a zatem także *synsem* i *content*, powinny być jawnie zdefiniowane w sygnaturze.



- PHON
  jest atrybutem rodzaju *sign*, a więc wszystkich obiektów reprezentujących wyrazy, frazy i zdania. Jako że nie zajmujemy się tu fonologią, przyjmujemy, że wartością atrybutu PHON jest lista postaci ortograficznych poszczególnych wyrazów wchodzących w skład danego wypowiedzenia. Tak więc sygnatura z rys. 4.1 musi zostać rozbudowana o nowy fragment prezentowany po prawej. W rezultacie wartościami atrybutu PHON obiektów rodzaju *sign* mogą być ⟨Mary⟩, ⟨reads a book⟩ itp.

- HD-DTR i NHD-DTRS
  to atrybuty właściwe dla fraz, reprezentują składniki bezpośrednie fraz. Pamiętajmy jednak, że wartość atrybutu HD-DTR jest rodzaju *sign*, czyli w szczególności może ona być rodzaju *phrase*. W takim wypadku, wartość ta wprowadza nowe, bardziej zagnieżdżone atrybuty HD-DTR i NHD-DTRS. Podobnie wartością atrybutu NHD-DTRS jest lista obiektów rodzaju *sign*. W rezultacie uzyskujemy bardzo złożone struktury.

- SYNSEM
  Wartości tego atrybutu odpowiadają informacji o cechach morfoskładniowych (z wyłączeniem struktury składników) i semantycznych danego wyrażenia. Wartości tego atrybutu należą do rodzaju *synsem*, który wprowadza dwa nowe atrybuty, a mianowicie LOCAL i NONLOCAL.

- LOCAL
  jest atrybutem o wartościach rodzaju *local* zawierającym właściwą informację morfoskładniową i semantyczną o danym wyrażeniu. Rodzaj ten wprowadza dwa atrybuty: CAT i CONT, których wartości odpowiadają informacji morfoskładniowej i semantycznej, odpowiednio.

- CAT
  Wartości tego atrybutu (rodzaju *category*) zawierają informację o cechach morfoskładniowych i składniowych danego wyrażenia. Jego struktura zaprezentowana jest obok po prawej.

$$\begin{bmatrix} \textit{category} \\ \textsc{head}\ \textit{head} \\ \textsc{subcat}\ \textit{list}(\textit{synsem}) \end{bmatrix}$$

- HEAD
  Wartości tego atrybutu określają kategorię morfoskładniową danego wyrażenia. Rodzaj *head* posiada co najmniej następujące podtypy: *verb* (czasownik), *noun* (rzeczownik), *adj* (*adjective*; przymiotnik), *adv* (*adverb*; przysłówek), *prep* (*preposition*; przyimek), *det* (*determiner*; rodzajnik), *conj* (*conjunction*; spójnik). Poszczególne podrodzaje wprowadzają różne atrybuty.

- SUBCAT
  Wartości tego atrybutu zawierają informację o wymaganiach składniowych tego wyrazu (lub frazy), tj. o argumentach, z którymi dany wyraz (fraza) musi się połączyć, aby powstała „pełna fraza". Na przykład czasownik *reads* musi się połączyć z dwoma argumentami: *Mary* oraz *a book* aby powstało zdanie *Mary reads a book*..

- CONT
  Atrybut ten zawiera informację semantyczną opisywanego wyrażenia. Dlatego scharakteryzowany zostanie dokładniej w rozdz. 4.5.

4.2.2 Ograniczenia

Natomiast ograniczenia są to formuły logiczne opisujące obiekty lingwistyczne. Jak już wspominaliśmy, wszystkie takie obiekty muszą spełniać wszystkie ograniczenia danej teorii, lub, innymi słowy, każde ograniczenie musi opisywać wszystkie obiekty z modelu. Najsłynniejszym ograniczeniem HPSG, częściowo odpowiedzialnym za nazwę tej teorii, jest przedstawiona poniżej Zasada elementu głównego (ang. *Head Feature Principle*, HFP).

Zasada 4.1. Zasada elementu głównego opisywana jest przez następującą zależność:

$$\textit{phrase} \implies \begin{bmatrix} \text{SYNSEM|LOCAL|CAT|HEAD } \boxed{1} \\ \text{HD-DTR|SYNSEM|LOCAL|CAT|HEAD } \boxed{1} \end{bmatrix}$$

Zasada elementu głównego ma postać implikacji logicznej; oznacza to, że każdy obiekt opisywany przez poprzednik implikacji musi być opisywany przez następnik implikacji, zupełnie jak w logice klasycznej. Lewa strona implikacji opisuje wszystkie obiekty rodzaju *phrase*. Prawa strona implikacji jest trochę bardziej skomplikowana — jest ona skrótowym zapisem struktury widocznej obok.

$$\begin{bmatrix} \textit{phrase} & \\ \text{SYNSEM} & \begin{bmatrix} \textit{synsem} & \\ \text{LOCAL} & \begin{bmatrix} \textit{local} & \\ \text{CAT} & \begin{bmatrix} \textit{category} & \\ \text{HEAD} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{HD-DTR} & \begin{bmatrix} \textit{sign} & \\ \text{SYNSEM} & \begin{bmatrix} \textit{synsem} & \\ \text{LOCAL} & \begin{bmatrix} \textit{local} & \\ \text{CAT} & \begin{bmatrix} \textit{category} & \\ \text{HEAD} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Jedynym nowym elementem w następniku implikacji jest (dwukrotne) użycie zmiennej $\boxed{1}$ jako wartości dwóch atrybutów HEAD, a ściślej jako wartości *ścieżek* SYNSEM|LOCAL|CAT|HEAD i HD-DTR|SYNSEM|LOCAL|CAT|HEAD. Oznacza to, że wartościami tych atrybutów musi być ten sam obiekt; innymi słowy, wartości tych atrybutów są *dzielone* (ang. *structure shared*). Ograniczenie z zasady 4.1 wymaga zatem, by wartość atrybutu HEAD dowolnej frazy była ta sama, co wartość atrybutu HEAD elementu głównego tej frazy („syna" HD-DTR tej frazy w drzewie składników bezpośrednich). Ponieważ wartości atrybutu HEAD odzwierciedlają kategorię morfoskładniową (czasownik, rzeczownik itp.) danego wyrażenia, fraza, której elementem głównym jest czasownik, musi być frazą czasownikową (jej wartość HEAD też jest „czasownikowa").

4.3 Relational Speciate Re-entrant Language

Relational Speciate Re-entrant Language (RSRL) jest to język formalny stworzony w celu formalnego zapisu gramatyki HPSG.

4.3.1 Syntaktyka

Jako język służacy do formalnego zapisu HPSG, RSRL składa się z *sygnatury* i *teorii*. Sygnatura pełni podobną rolę jak w *algebrach wielorodzajowych* (na co wskazuje choćby podobieństwo terminologii).

Definicja 4.6. Sygnaturą Σ nazywamy uporządkowaną szóstkę $\langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R} \rangle$, gdzie:

- $\langle \mathcal{G}, \sqsubseteq \rangle$ jest skończonym częściowo uporządkowanym zbiorem *rodzajów* — hierarchia rodzajów;
- $\mathcal{S}$ jest zbiorem elementów minimalnych w $\langle \mathcal{G}, \sqsubseteq \rangle$, tj. $\mathcal{S} = \{\sigma \in \mathcal{G} \colon \text{ dla każdego } \sigma' \in \mathcal{G}, \text{ jeżeli } \sigma' \sqsubseteq \sigma \text{ to } \sigma = \sigma'\}$;
- $\mathcal{A}$ jest zbiorem atrybutów;
- $\mathcal{F} : \mathcal{G} \times \mathcal{A} \longrightarrow \mathcal{G}$ jest taką funkcją częściową, że dla dowolnych $\sigma_1, \sigma_2 \in \mathcal{G}$ i każdego $\alpha \in \mathcal{A}$, jeżeli $\mathcal{F}(\langle \sigma_1, \alpha \rangle)$ jest zdefiniowane i $\sigma_2 \sqsubseteq \sigma_1$, to $\mathcal{F}(\langle \sigma_2, \alpha \rangle)$ jest zdefiniowane oraz $\mathcal{F}(\langle \sigma_2, \alpha \rangle) \sqsubseteq \mathcal{F}(\langle \sigma_1, \alpha \rangle)$;[37]
- $\mathcal{R} = \bigcup_{n \in \mathbb{N}^+} \mathcal{R}^n$ jest skończonym zbiorem (n-argumentowych) symboli predykatywnych.

[37] Innymi słowy, $\mathcal{F}$ zachowuje $\sqsubseteq$.

Tak więc $\mathcal{F}$ jest funkcją przyporządkowującą atrybutowi danego rodzaju odpowiadający mu rodzaj (oczywiście jedynie wówczas, gdy para ⟨atrybut, rodzaj⟩ istnieje), i to taką, która zapewnia, że wszystkie podrodzaje określonego rodzaju dziedziczą jego atrybuty.

Przykład *4.1.* Rozważmy fragment sygnatury przedstawiony na rys. 4.1.[38] Wówczas:

- $\mathcal{G} = \{$*object, sign, synsem, local, list, word, phrase, elist, nelist*$\}$;
- $\sqsubseteq\ = \{\langle \textit{object, object}\rangle, \langle \textit{sign, object}\rangle, \langle \textit{synsem, object}\rangle, \langle \textit{local, object}\rangle, \langle \textit{list, object}\rangle,$
  $\langle \textit{word, object}\rangle, \langle \textit{phrase, object}\rangle, \langle \textit{elist, object}\rangle, \langle \textit{nelist, object}\rangle, \langle \textit{sign, sign}\rangle, \langle \textit{word, sign}\rangle,$
  $\langle \textit{phrase, sign}\rangle, \langle \textit{synsem, synsem}\rangle, \langle \textit{local, local}\rangle, \langle \textit{list, list}\rangle, \langle \textit{elist, list}\rangle, \langle \textit{nelist, list}\rangle,$
  $\langle \textit{elist, elist}\rangle, \langle \textit{nelist, nelist}\rangle\}$;
- $\mathcal{S} = \{$*synsem, local, word, phrase, elist, nelist*$\}$;
- $\mathcal{A} = \{$PHON, SYNSEM, LOCAL, NONLOCAL, CAT, CONT, HD-DTR, NHD-DTRS$\}$;
- $\mathcal{F} = \{\langle\langle \textit{sign}, \textsc{phon}\rangle, \textit{list}\rangle, \langle\langle \textit{sign}, \textsc{synsem}\rangle, \textit{synsem}\rangle, \langle\langle \textit{synsem}, \textsc{local}\rangle, \textit{local}\rangle,$
  $\langle\langle \textit{synsem}, \textsc{nonlocal}\rangle, \textit{nonlocal}\rangle, \langle\langle \textit{local}, \textsc{cat}\rangle, \textit{category}\rangle, \langle\langle \textit{local}, \textsc{cont}\rangle, \textit{content}\rangle,$
  $\langle\langle \textit{phrase}, \textsc{hd-dtr}\rangle, \textit{sign}\rangle, \langle\langle \textit{phrase}, \textsc{nhd-dtrs}\rangle, \textit{list}\rangle\}$;
- $\mathcal{R} = \{$`append, member`$\}$.

W celu opisu *łańcuchów* obiektów (które wyglądają podobnie jak listy, i mają podobne własności, lecz służą do opisu ciągów obiektów znajdujących się na poszczególnych ścieżkach atrybutów), dla każdej sygnatury Σ Richter i Sailer definiują *rozszerzoną sygnaturę* $\widehat{\Sigma}$.

Definicja 4.7. Niech $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}\rangle$ będzie dowolną sygnaturą. Wówczas *rozszerzoną sygnaturą* $\widehat{\Sigma}$ nazywamy uporządkowaną szóstkę $\langle \widehat{\mathcal{G}}, \widehat{\sqsubseteq}, \widehat{\mathcal{S}}, \widehat{\mathcal{A}}, \mathcal{F}, \mathcal{R}\rangle$, gdzie:

- $\widehat{\mathcal{G}} = \mathcal{G} \cup \{\textit{chain}, \textit{echain}, \textit{nechain}, \textit{metatop}\}$;
- $\widehat{\sqsubseteq} = \sqsubseteq \cup \{\langle \textit{chain, chain}\rangle, \langle \textit{echain, chain}\rangle, \langle \textit{nechain, chain}\rangle, \langle \textit{echain, echain}\rangle, \langle \textit{nechain, nechain}\rangle\}$
  $\cup \left\{\langle \textit{metatop}, \sigma\rangle \colon \sigma \in \widehat{\mathcal{G}}\right\}$;
- $\widehat{\mathcal{S}} = \mathcal{S} \cup \{\textit{echain}, \textit{nechain}\}$;
- $\widehat{\mathcal{A}} = \mathcal{A} \cup \{\dagger, \triangleright\}$.

W formalizmie RSRL także można zdefiniować zbiór *termów* oraz zbiór *deskrypcji* (ang. *description*), odpowiadających w przybliżeniu formułom logicznym.

Definicja 4.8. Niech $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}\rangle$ będzie dowolną sygnaturą. Zbiór *termów* $\mathfrak{T}^{\Sigma}$ względem sygnatury Σ jest to najmniejszy taki zbiór, że:

- $: \in \mathfrak{T}^{\Sigma}$;
- niepusty, przeliczalny zbiór zmiennych $\mathcal{V} \subseteq \mathfrak{T}^{\Sigma}$;
- $\mathfrak{t}\alpha \in \mathfrak{T}^{\Sigma}$ dla dowolnych $\mathfrak{t} \in \mathfrak{T}^{\Sigma}$, $\alpha \in \widehat{\mathcal{A}}$.

Definicja 4.9. Niech $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}\rangle$ będzie dowolną sygnaturą. Zbiór *deskrypcji* $\mathfrak{D}^{\Sigma}$ względem sygnatury Σ jest to najmniejszy taki zbiór, że:

- $\mathfrak{t} \sim \sigma \in \mathfrak{D}^{\Sigma}$ dla dowolnych $\mathfrak{t} \in \mathfrak{T}^{\Sigma}$ i $\sigma \in \mathcal{G}$;
- $\mathfrak{t}_1 \approx \mathfrak{t}_2 \in \mathfrak{D}^{\Sigma}$ dla dowolnych $\mathfrak{t}_1, \mathfrak{t}_2 \in \mathfrak{T}^{\Sigma}$;
- $\rho(x_1, \ldots, x_n) \in \mathfrak{D}^{\Sigma}$ dla dowolnych $\rho \in \mathcal{R}^n, x_1 \in \mathcal{V}, \ldots, x_n \in \mathcal{V}$;
- $\mathbf{not}\,\delta \in \mathfrak{D}^{\Sigma}$ dla każdego $\delta \in \mathfrak{D}^{\Sigma}$;
- $\delta_1 \rightarrow \delta_2 \in \mathfrak{D}^{\Sigma}$ dla dowolnych $\delta_1, \delta_2 \in \mathfrak{D}^{\Sigma}$;
- $\mathbf{A}\,x\,\delta \in \mathfrak{D}^{\Sigma}$ dla dowolnych $x \in \mathcal{V}, \delta \in \mathfrak{D}^{\Sigma}$.

[38] A więc bez rodzajów ukazanych jedynie jako rodzaj wartości atrybutu.

Deskrypcją zamkniętą nazywamy dowolną deskrypcję $\delta \in \mathfrak{D}^{\Sigma}$ taką, że δ nie zawiera zmiennych wolnych. Zbiór wszystkich deskrypcji zamkniętych oznaczany będzie przez $\mathfrak{D}^{\Sigma}_{\bullet}$.

A, not są to standardowe symbole logiczne, odpowiadające symbolom $\forall, \neg$. Zmiana notacji ma na celu uniknięcie konfuzji w stosunku do języka Ty2 (por. rozdz. 4.1). Z tego samego względu w niniejszym opracowaniu termy oznakowane zostały przez $\mathfrak{t}$, zamiast używanego w [Sailer, 2000] τ.

Mając zdefiniowany zbiór deskrypcji RSRL, możemy bez trudu zdefiniować teorię RSRL a także gramatykę RSRL.

Definicja 4.10. Dla każdej sygnatury Σ, *teorią RSRL* względem Σ nazywamy dowolny zbiór deskrypcji zamkniętych $\theta \subseteq \mathfrak{D}^{\Sigma}_{\bullet}$.

Definicja 4.11. *Gramatyka RSRL* Γ jest to para $\langle \Sigma, \theta \rangle$ taka, że Σ jest pewną sygnaturą, zaś θ jest teorią względem tej sygnatury.

Gramatyka składa się zatem z sygnatury i z teorii, czyli zbioru deskrypcji RSRL.

4.3.2 Semantyka

Ponieważ RSRL jest językiem logicznym, niezbędne jest opisanie interpretacji semantycznej wyrażeń wchodzących w jego skład. Poniżej przedstawimy definicje pojęć niezbędnych do tego celu.

Dla dowlnego zbioru S, przez S^n będziemy oznaczać zbiór wszystkich n-tek zdefiniowanych w tym zbiorze, przez $S^{\star}$ — zbiór „krotek dowolnej długości, przez S^{+} — zbiór krotek niezerowej długości, zaś $\overline{S} = S \cup S^{\star}$.

Definicja 4.12. Niech $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R} \rangle$ będzie dowolną sygnaturą. *Interpretacją* I dla sygnatury Σ jest czwórka uporządkowana $\langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ taka, że:

- U jest pewnym niepustym zbiorem (uniwersum);
- $\mathsf{S} \colon \mathsf{U} \longrightarrow \mathcal{S}$ jest funkcją przypisującą dowolnemu obiektowi z U odpowiadający mu (minimalny) rodzaj z $\mathcal{S}$;
- $\mathsf{A} \colon \mathcal{A} \longrightarrow \mathfrak{U}$, gdzie $\mathfrak{U}$ jest zbiorem wszystkich funkcji częściowych $\Psi \colon \mathsf{U} \longrightarrow \mathsf{U}$, jest dowolną funkcją taką, że dla każdych $\alpha \in \mathcal{A}, u \in \mathsf{U}$, jeżeli $\mathsf{A}(\alpha)(u)$ jest zdefiniowane, to $\mathcal{F}(\langle \mathsf{S}(u), \alpha \rangle)$ jest zdefiniowane i $\mathsf{S}(\mathsf{A}(\alpha)(u)) \sqsubseteq \mathcal{F}(\langle \mathsf{S}(u), \alpha \rangle)$, oraz jeżeli $\mathcal{F}(\langle \mathsf{S}(u), \alpha \rangle)$ jest zdefiniowane, to $\mathsf{A}(\alpha)(u)$ jest zdefiniowane;
- $\mathsf{R} = \bigcup_{n \in \mathbb{N}^{+}} \mathsf{R}^n$ jest zbiorem funkcji $\mathsf{R}^n : \mathcal{R}^n \longrightarrow \overline{\mathsf{U}}^n$ przypisujących n-argumentowym symbolom predykatywnym n-argumentowe relacje nad $\overline{\mathsf{U}}$.[39]

Tak więc A jest funkcją w specyficzny sposób interpretującą atrybuty poszczególnych rodzajów, a mianowicie przypisującą atrybutowi α funkcje częściowe z U w U wiążące obiekty posiadające rodzaj α z obiektem rodzaju będącego wartością α. Tak więc *atrybuty* nie występują w interpretacji bezpośrednio, a jedynie jako funkcje łączące obiekty określonych rodzajów.

Tak jak dowolną sygnaturę Σ można rozszerzyć na $\widehat{\Sigma}$, tak na podstawie interpretacji I można uzyskać jej odpowiednie rozszerzenie.

Definicja 4.13. Niech $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ będzie dowolną interpretacją sygnatury $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R} \rangle$. Wówczas:

- $\widehat{\mathsf{S}} \colon \overline{\mathsf{U}} \longrightarrow \widehat{\mathcal{S}}$ jest taką funkcją, że:
  - dla dowolnego $u \in \mathsf{U}$ zachodzi $\widehat{\mathsf{S}}(u) = \mathsf{S}(u)$,
  - dla dowolnych $u_1, ..., u_n \in \mathsf{U}$ zachodzi $\widehat{\mathsf{S}}(\langle u_1, ..., u_n \rangle) = \begin{cases} \textit{echain} & \text{dla } n = 0, \\ \textit{nechain} & \text{dla } n > 0; \end{cases}$

[39] A więc argumentami tych relacji są nie tylko obiekty, ale i łańcuchy (krotki) obiektów.

- $\widehat{\mathsf{A}}\colon \widehat{\mathcal{A}} \longrightarrow \overline{\mathfrak{U}}$, gdzie $\overline{\mathfrak{U}}$ jest zbiorem wszystkich funkcji częściowych $\overline{\Psi}\colon \overline{\mathsf{U}} \longrightarrow \overline{\mathsf{U}}$, jest dowolną funkcją taką, że:
  - dla dowolnych $\alpha \in \mathcal{A}$ zachodzi $\widehat{\mathsf{A}}(\alpha) = \mathsf{A}(\alpha)$;
  - $\widehat{\mathsf{A}}(\dagger)\colon \mathsf{U}^{+} \longrightarrow \mathsf{U}$ jest taką funkcją, że $\widehat{\mathsf{A}}(\dagger)(\langle u_1, ..., u_n\rangle) = u_1$ dla dowolnych $u_1, ...u_n \in \mathsf{U}$;
  - $\widehat{\mathsf{A}}(\triangleright)\colon \mathsf{U}^{+} \longrightarrow \mathsf{U}^{\star}$ jest taką funkcją, że $\widehat{\mathsf{A}}(\triangleright)(\langle u_1, ..., u_n\rangle) = \langle u_2, ..., u_n\rangle$ dla każdych $u_1, ...u_n \in \mathsf{U}$;

Tak więc $\widehat{\mathsf{S}}$, $\widehat{\mathsf{A}}$ są to takie dziwne funkcje które działają zarówno na obiektach, jak i na łańcuchach obiektów. Rzecz jasna, zgodnie z definicją, ani $\widehat{\mathsf{A}}(\dagger)$, ani $\widehat{\mathsf{A}}(\triangleright)$ nie jest zdefiniowane dla listy pustej $\langle\rangle$.

Definicja 4.14. Niech $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R}\rangle$ będzie interpretacją dla sygnatury $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}\rangle$. *Wartościowaniem* nazywamy dowolną funkcję $\mathrm{V}\colon \mathcal{V} \longrightarrow \overline{\mathsf{U}}$. Zbiór wszystkich wartościowań to $\Upsilon(\mathsf{I}) = \overline{\mathsf{U}}^{\mathcal{V}}$. Ponadto przez V^{u}_{x} oznaczane będzie wartościowanie uzyskane z V przez przypisanie zmiennej $x \in \mathcal{V}$ wartości $u \in \overline{\mathsf{U}}$.

Konsekwentnie, wartościowanie V przypisuje zmiennej z $\mathcal{V}$ albo obiekt z U, albo łańcuch obiektów z $\mathsf{U}^{\star}$.

Powyższe pojęcia wystarczają już do zdefiniowania interpretacji semantycznej dla termów z $\mathfrak{T}^{\Sigma}$ oraz deskrypcji z $\mathfrak{D}^{\Sigma}$.

Definicja 4.15. Niech $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R}\rangle$ będzie interpretacją dla sygnatury $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}\rangle$, zaś $\mathrm{V} \in \Upsilon(\mathsf{I})$ będzie wartościowaniem. *Interpretacją termów* względem I i V nazywamy funkcję $T^{\mathrm{V}}_{\mathsf{I}}\colon \mathfrak{T}^{\Sigma} \longrightarrow \overline{\Psi}$ (gdzie $\overline{\Psi}\colon \overline{\mathsf{U}} \longrightarrow \overline{\mathsf{U}}$) taką, że dla dowolnego $u \in \overline{\mathsf{U}}$:

- $T^{\mathrm{V}}_{\mathsf{I}}(:)(u) = u$;
- $T^{\mathrm{V}}_{\mathsf{I}}(v)(u) = \mathrm{V}(v)$ dla każdego $v \in \mathcal{V}$;
- dla dowolnych $\mathsf{t} \in \mathfrak{T}^{\Sigma}, \alpha \in \widehat{\mathcal{A}}$ zachodzi $T^{\mathrm{V}}_{\mathsf{I}}(\mathsf{t}\alpha)(u) = \widehat{\mathsf{A}}(\alpha)(T^{\mathrm{V}}_{\mathsf{I}}(\mathsf{t})(u))$, jeżeli $T^{\mathrm{V}}_{\mathsf{I}}(\mathsf{t})(u)$ oraz $\widehat{\mathsf{A}}(\alpha)(T^{\mathrm{V}}_{\mathsf{I}}(\mathsf{t})(u))$ są zdefiniowane, w przeciwnym wypadku $T^{\mathrm{V}}_{\mathsf{I}}(\mathsf{t}\alpha)(u)$ nie jest zdefiniowane.

Definicja 4.16. Niech $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R}\rangle$ będzie interpretacją dla sygnatury $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}\rangle$, $\pi \in \mathcal{A}^{\star}$ będzie ścieżką atrybutów, zaś $\mathrm{V} \in \Upsilon(\mathsf{I})$ będzie wartościowaniem. Zbiorem składników (ang. *components*) danego obiektu $u \in \mathsf{U}$ nazywamy $\mathsf{Co_I}^{u} = \{u' \in \mathsf{U}\colon T^{\mathrm{V}}_{\mathsf{I}}(:\pi)(u)$ jest zdefiniowane i $u' \in T^{\mathrm{V}}_{\mathsf{I}}(:\pi)(u)\}$.

Przeto składnik pewnego obiektu u jest to obiekt będący wartością atrybutu znajdujący się gdzieś na ścieżce atrybutów $:\pi$ obiektu u.

Definicja 4.17. Niech $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R}\rangle$ będzie interpretacją dla sygnatury $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}\rangle$, zaś $\mathrm{V} \in \Upsilon(\mathsf{I})$ będzie wartościowaniem. *Interpretacją deskrypcji* względem I i V nazywamy funkcję $T^{\mathrm{V}}_{\mathsf{I}}\colon \mathfrak{T}^{\Sigma} \longrightarrow 2^{\mathsf{U}}$ taką, że:

- $D^{\mathrm{V}}_{\mathsf{I}}(\mathsf{t} \sim \sigma) \;=\; \left\{u \in \mathsf{U}\colon T^{\mathrm{V}}_{\mathsf{I}}(\mathsf{t})(u) \text{ jest zdefiniowane i } \mathsf{S}(T^{\mathrm{V}}_{\mathsf{I}}(\mathsf{t})(u))\widehat{\sqsubseteq}\sigma\right\}$ dla dowolnych $\mathsf{t} \in \mathfrak{T}^{\Sigma},\ \sigma \in \widehat{\mathcal{G}}$;
- $D^{\mathrm{V}}_{\mathsf{I}}(\mathsf{t}_1 \approx \mathsf{t}_2) \;=\; \{u \in \mathsf{U}\colon T^{\mathrm{V}}_{\mathsf{I}}(\mathsf{t}_1)(u), T^{\mathrm{V}}_{\mathsf{I}}(\mathsf{t}_2)(u) \text{ sa zdefiniowane i } T^{\mathrm{V}}_{\mathsf{I}}(\mathsf{t}_1)(u) \;=\; T^{\mathrm{V}}_{\mathsf{I}}(\mathsf{t}_2)(u)\}$ dla dowolnych $\mathsf{t}_1, \mathsf{t}_2 \in \mathfrak{T}^{\Sigma}$;
- $D^{\mathrm{V}}_{\mathsf{I}}(\rho(x_1, \ldots, x_n)) \;=\; \{u \in \mathsf{U}\colon \langle \mathrm{V}(x_1), ..., \mathrm{V}(x_2)\rangle \in \mathsf{R}(\rho)\}$ dla dowolnych $x_1, ..., x_n \in \mathcal{V}$ oraz dowolnego $\rho \in \mathcal{R}^{n}$;
- $D^{\mathrm{V}}_{\mathsf{I}}(\mathbf{not}\,\delta) \;=\; \mathsf{U} - D^{\mathrm{V}}_{\mathsf{I}}(\delta)$ dla każdego $\delta \in \mathfrak{D}^{\Sigma}$;
- $D^{\mathrm{V}}_{\mathsf{I}}(\delta_1 \rightarrow \delta_2) \;=\; (\mathsf{U} - D^{\mathrm{V}}_{\mathsf{I}}(\delta_1)) \cup D^{\mathrm{V}}_{\mathsf{I}}(\delta_1)$ dla dowolnych $\delta_1, \delta_2 \in \mathfrak{D}^{\Sigma}$;
- $D^{\mathrm{V}}_{\mathsf{I}}(\mathbf{A}\,x\,\delta) \;=\; \left\{u \in \mathsf{U}\colon u \in D^{\mathrm{V}^{u'}_{x}}_{\mathsf{I}} \text{ dla dowolnego } u' \in \overline{\mathsf{Co_I}^{u}}\right\}$ dla dowolnych $x \in \mathcal{V}, \delta \in \mathfrak{D}^{\Sigma}$.

Interpretacją deskrypcji względem I jest funkcja $D_{\mathsf{I}}\colon \mathfrak{D}^{\Sigma}_{\bullet} \longrightarrow 2^{\mathsf{U}}$ taka, że
$D_{\mathsf{I}}(\delta) = \{u \in \mathsf{U}\colon\ u \in D^{\mathrm{V}}_{\mathsf{I}}$ dla dowolnego $\mathrm{V} \in \Upsilon(\mathsf{I})\}$ dla każdego $\delta \in \mathfrak{D}^{\Sigma}$.
Interpretacją teorii względem I jest funkcja $\Theta_{\mathsf{I}}\colon 2^{\mathfrak{D}^{\Sigma}_{\bullet}} \longrightarrow 2^{\mathsf{U}}$ taka, że
$\Theta_{\mathsf{I}}(\theta) = \{u \in \mathsf{U}\colon\ u \in D_{\mathsf{I}}(\delta)$ dla dowolnego $\delta \in \theta\}$ dla każdego $\theta \subseteq \mathfrak{D}^{\Sigma}$.

Zauważmy, że argumentami predykatu ρ mogą być jedynie zmienne, a nie dowolne termy.

Powyższe pojęcia możemy bez trudu rozszerzyć na gramatyki.

Definicja 4.18. Niech $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R}\rangle$ będzie interpretacją dla sygnatury $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}\rangle$, zaś $\Gamma = \langle \Sigma, \theta\rangle$ będzie pewną gramatyką względem tej sygnatury. Powiemy, że I jest *modelem* gramatyki $\langle \Sigma, \theta\rangle$ wtedy i tylko wtedy, gdy $\Theta_{\mathsf{I}}(\theta) = \mathsf{U}$. Powiemy, że I jest *dogłębnym modelem* (ang. *exhaustive model*[40]) gramatyki Γ wtedy i tylko wtedy, gdy I jest modelem tej gramatyki oraz dla dowolnej teorii $\theta \subseteq \mathfrak{D}^{\Sigma}_{\bullet}$ i dowolnej interpretacji $\mathsf{I}' = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R}\rangle$ względem Σ, jeżeli I' jest modelem dla Γ i $\Theta_{\mathsf{I}'}(\theta') \neq \emptyset$, to $\Theta_{\mathsf{I}}(\theta') \neq \emptyset$.

Twierdzenie 4.1. Dla każdej gramatyki $\Gamma = \langle \Sigma, \theta\rangle$ istnieje dogłębny model tej gramatyki.

4.3.3 Notacja

RSRL jest formalizmem, w którym można zapisać dowolną gramatykę HPSG. Sposób zapisu sygnatury HPSG w formacie RSRL jest jasny i zrozumiały, a odpowiedniość wszystkich pojęć w oczywisty sposób jednoznaczna, co zostało zilustrowane w przykładzie 4.1. Ta sama zgodność dotyczy również teorii. Jednak zapisanie ograniczeń HPSG jako deskrypcji RSRL jest mało czytelne Zazwyczaj, wyrażenia HPSG zapisywane są w tzw. notacji AVM (ang. *Atribute-Value Matrix*). Szczegółowe zasady translacji z jednej notacji na drugą (a więc i ich logiczna odpowiedniość) przedstawione zostały w pracy [Richter, 1999], nie będziemy ich tu jednak prezentować.

Poniżej przedstawimy definicje relacji `append` i `member` z przykładu 4.1.

NOTACJA: $\langle \boxed{1} \,|\, \boxed{2} \rangle \quad \equiv_{def} \quad \begin{bmatrix} \textit{nelist} \\ \text{FIRST } \boxed{1} \\ \text{REST } \boxed{2} \end{bmatrix}.$

Definicja 4.19. Relacja `append` definiowana jest w sposób następujący:

- $\mathtt{append}(\langle\rangle, \boxed{1}, \boxed{1})$,
- $\mathtt{append}(\langle \boxed{1} \,|\, \boxed{2} \rangle, \boxed{3}, \langle \boxed{1} \,|\, \boxed{4} \rangle) \quad \stackrel{\forall}{\Longleftarrow} \quad \mathtt{append}(\boxed{2}, \boxed{3}, \boxed{4})$;

zaś relacja `memember` jako:

- $\mathtt{member}(\boxed{1}, \langle \boxed{1} \,|\, \boxed{2} \rangle)$,
- $\mathtt{member}(\boxed{0}, \langle \boxed{1} \,|\, \boxed{2} \rangle) \quad \stackrel{\forall}{\Longleftarrow} \quad \mathtt{member}(\boxed{0}, \boxed{2})$.

Według tej definicji, trzy listy a, b i c znajdują się w relacji `append` (czyli lista c jest konkatenacją list a i b) wtedy i tylko wtedy, gdy albo lista a jest listą pustą, a listy b i c stanowią tę samą listę (pierwsza

[40] Przepiórkowski i in. (2003) używają terminu *pełny*. Ponieważ w logice określenie to ma zupełnie inne znaczenie, wprowadziliśmy inne, chyba bardziej właściwe określenie.

klauzula), albo listy a i c mają ten sam pierwszy element ($\boxed{1}$), zaś reszta listy c jest konkatenacją reszty listy a i całej listy b (druga klauzula). Natomiast listy a i b znajduje się w relacji `member` (czyli lista a jest elementem listy b, jeśli lista a jest albo pierwszym elementem listy b, albo jest elementem reszty listy b).

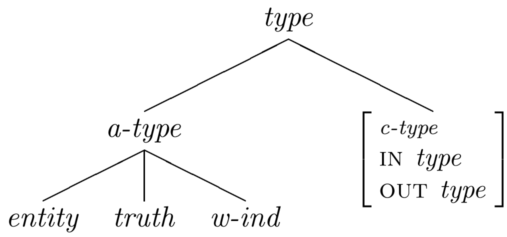
Wśród istotnych założeń autorzy wymieniają konwencję, by domykać deskrypcje, wiążąc wszystkie zmienne wolne za pomocą kwantyfikatora egzystencjalnego, w zakresie których znajdzie się cała domyjana deskrypcja. Zauważmy, że również znaczniki dyskursu struktur DRS z rozdz. 3 są kwantyfikowane egzystencjalnie (przynajmniej na poziomie głównym).

4.4 Zapis języka Ty2 w formalizmie RSRL jako gramatyki $\mathcal{TY}2$

Powyżej przedstawione zostały dwa formalizmy: klasyczny język logiczny Ty2 oraz logika deskrypcyjna RSRL, służąca do zapisu gramatyk HPSG. Richter i Sailer przyjęli założenie, że (zgodnie z ideą twórców HPSG Pollarda i Saga) semantyka będzie nierozłączną częścią samej gramatyki. W tym celu musimy zapisać język Ty2 będący docelowym językiem interpretacji semantycznej w notacji HPSG, lub równoważnie w języku RSRL. Aby tego dokonać, wystarczy potraktować język Ty2 w ten sam sposób, co każdy inny język (np. naturalny, jak angielski czy polski), i stworzyć jego gramatykę HPSG. W ten sposób uzyskujemy translację z Ty2 na RSRL, którą będziemy oznaczać przez $\mathfrak{y}$.

4.4.1 Typy

Rekurencyjną definicję zbioru typów $\mathcal{T}$ z def. 4.1 można zapisać jako hierarchię rodzajów przedstawioną na rys. 4.2. Oczywiście notacja ta wystarcza do zapisu dowolnego typu z $\mathcal{T}$. Zauważmy, że *a-type* jest skrótem od *atomic-type*, *c-type* od *complex-type* oraz *w-ind* od *w-index*.



Rysunek 4.2: Hierarchia rodzajów RSRL odpowiadająca rekurencyjnej definicji typów Ty2

<u>Przykład</u> *4.2.* Przykładowe typy Ty2 tłumaczone są na RSRL w sposób następujący:

$$\mathfrak{y}(t) = \left[\,\textit{truth}\,\right] \qquad \mathfrak{y}(e) = \left[\,\textit{entity}\,\right] \qquad \mathfrak{y}(s) = \left[\,\textit{w-ind}\,\right]$$

$$\mathfrak{y}(\langle e,t\rangle) = \begin{bmatrix} \textit{c-type} \\ \text{IN } \textit{entity} \\ \text{OUT } \textit{truth} \end{bmatrix} \quad \mathfrak{y}(\langle s,e\rangle) = \begin{bmatrix} \textit{c-type} \\ \text{IN } \textit{w-ind} \\ \text{OUT } \textit{entity} \end{bmatrix} \quad \mathfrak{y}(\langle\langle s,e\rangle,t\rangle) = \begin{bmatrix} \textit{c-type} \\ \text{IN } \begin{bmatrix} \textit{c-type} \\ \text{IN } \textit{w-ind} \\ \text{OUT } \textit{entity} \end{bmatrix} \\ \text{OUT } \textit{truth} \end{bmatrix}$$

Dla tak zdefiniowanego fragmentu sygnatury (rodzaj *type*) niezbędne są ograniczenia zapewniające rzeczywistą odpowiedniość względem Ty2.

Zasada 4.2. Rodzaj *type* musi spełniać następujące zasady:

- Zasadę identyczności typów (ang. *Type Identity Principe*, TyIP) opisywaną przez poniższą zależność:

$$\textit{type} \implies \mathbf{A}\,\boxed{1}\ \mathbf{A}\,\boxed{2}\ (\texttt{same-type}(\boxed{1},\boxed{2}) \implies \boxed{1} \approx \boxed{2});$$

- Zasadę acykliczności typów (ang. *Type Non-Cyclity Principe*, TyNP):

$$\textit{type} \implies \mathbf{A}\,\boxed{1}\ (\left(\left[\text{IN } \boxed{1}\right] \mathbf{or} \left[\text{OUT } \boxed{1}\right]\right) \implies \mathbf{not}\ \texttt{component}(:,\boxed{1}));$$

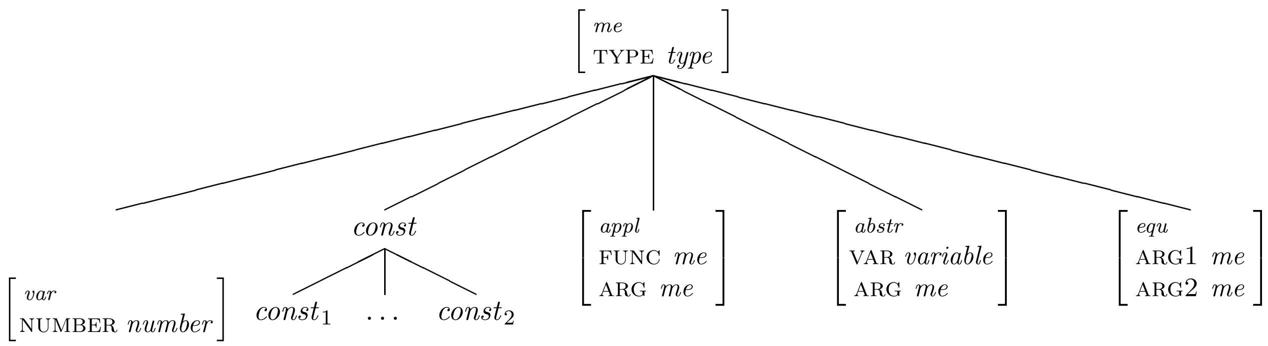
- Zasadę skończoności typów (ang. *Type Finiteness Principe*, TyFP):

$$\textit{type} \implies \mathbf{E}\,a\ {}^{a}\!\left[\,\textit{chain}\,\right] \mathbf{and}\ \ \mathbf{A}\,\boxed{1}\ (\texttt{component}(\boxed{1},:) \implies \texttt{member}(\boxed{1},a)).$$

Zasada TyIP wymusza, by dla każdego typu złożonego, wszystkie jego podtypy wyglądające identycznie faktycznie zostały uznane za identyczne. Zasada TyNP, jak sama nazwa wskazuje, zabezpiecza przez cyklicznością typów, wykuczając by nadtyp danego typu był jednocześnie jego podtypem. Na koniec zasada TyFP wyklucza typy nieskończone.

4.4.2 Termy

Na koniec zapiszemy zbiór termów $\mathfrak{T}$ jako kolejny fragment sygnatury RSRL. W tym celu wprowadzamy rodzaj *me wyrażeń znaczących* (ang. *meaningful expressions*).[41] Jest on korzeniem hierarchii rodzajów przedstawionej na rys. 4.3. Zauważmy, że *var* jest skrótem od *variable*, *const* od *constant*, *appl* od *application*, *abstr* od *abstraction* i *equ* od *equation*. Ponadto FUNC jest skrótem od FUNCTOR, ARG od ARGUMENT oraz VAR od VARIABLE. Rodzaj *appl* odpowiada za aplikację funktora będącego wartością atrybutu FUNC do pewnego termu będącego wartością atrybutu ARG, czyli wyrażeniu $\varphi(\psi)$ z def. 4.2. Rodzaj *abstr* odpowiada λ-wyrażeniom, zaś rodzaj *equ* równości dwóch termów. Typ każdego obiektu rodzaju *me* jest ustalony jako wartość atrybutu TYPE.



Rysunek 4.3: Hierarchia rodzajów RSRL odpowiadająca definicji termów Ty2

Rzecz jasna, tworzenie termów złożonych z termów składowych wymaga odpowiednich zależności pomiędzy typami tych termów. Aby zgodność ta była zachowana, definiowane są następujące ograniczenia (w $const_n$ n oznacza dowolną acz określoną liczbę naturalną):

Zasada 4.3. Spełnione muszą być następujące warunki:

- $appl \Longrightarrow \begin{bmatrix} \text{TYPE } \boxed{2} \\ \text{FUNC}|\text{TYPE} \begin{bmatrix} \textit{c-type} \\ \text{IN } \boxed{1} \\ \text{OUT } \boxed{2} \end{bmatrix} \\ \text{ARG}|\text{TYPE } \boxed{1} \end{bmatrix}$;
- $abstr \Longrightarrow \begin{bmatrix} \text{TYPE} \begin{bmatrix} \textit{c-type} \\ \text{IN } \boxed{1} \\ \text{OUT } \boxed{2} \end{bmatrix} \\ \text{VAR}|\text{TYPE } \boxed{1} \\ \text{ARG}|\text{TYPE } \boxed{2} \end{bmatrix}$;
- $const_n \Longrightarrow \begin{bmatrix} \text{TYPE} \begin{bmatrix} \textit{c-type} \\ \text{IN } \textit{w-ind} \\ \text{OUT} \begin{bmatrix} \textit{c-type} \\ \text{IN } \textit{entity} \\ \text{OUT } \textit{truth} \end{bmatrix} \end{bmatrix} \end{bmatrix}$;
- $equ \Longrightarrow \begin{bmatrix} \text{TYPE } \textit{truth} \\ \text{ARG1}|\text{TYPE } \boxed{1} \\ \text{ARG2}|\text{TYPE } \boxed{1} \end{bmatrix}$.

Termy muszą spełniać także zasady podobne do tych narzuconych na typy.

Zasada 4.4. Rodzaj *me* musi spełniać następujące zasady:

- Zasadę identyczności termów (ang. *Term Identity Principe*, TIP) opisywaną przez zależność:

$$term \Longrightarrow \mathbf{A}\boxed{1}\ \mathbf{A}\boxed{2}\ (\texttt{same-term}(\boxed{1},\boxed{2}) \Longrightarrow \boxed{1} \approx \boxed{2});$$

[41] Zauważmy, że jest to dokładnie to samo określenie, którego używał Montague (por. rozdz. 2.1) i to przede wszystkim z tego względu, że nie rozróżniał termów i formuł (deskrypcji).

- Zasadę acykliczności termów (ang. *Term Non-Cyclity Principe*, TNP):

$$\textit{term} \implies \mathbf{A}\boxed{1}\ \big(\big(\big[\,\text{FUNC}\ \boxed{1}\,\big]\mathbf{or}\big[\,\text{ARG}\ \boxed{1}\,\big]\mathbf{or}\big[\,\text{ARG1}\ \boxed{1}\,\big]\mathbf{or}\big[\,\text{ARG2}\ \boxed{1}\,\big]\mathbf{or}\big[\,\text{VAR}\ \boxed{1}\,\big]\big) \implies \\ \mathbf{not}\ \texttt{component}\,(:,\boxed{1})\big);$$

- Zasadę skończoności termów (ang. *Term Finiteness Principe*, TFP):

$$\textit{term} \implies \mathbf{E}\,a\ {}^{q}\big[\,\textit{chain}\,\big]\mathbf{and}\ \ \mathbf{A}\boxed{1}\ \big(\texttt{component}\,(\boxed{1},:) \implies \texttt{member}(\boxed{1},a)\big).$$

Aby te (i wszystkie inne omawiane w niniejszym rozdziale) zasady były spełnione, potrzebne nam są definicje trzech nowych relacji: `same-type`, `same-term` i `component`.

Definicja 4.20. Relacja `same-type` definiowana jest następująco:

- $\texttt{same-type}\,(\boxed{1},\boxed{2}) \ \overset{\forall}{\Longleftarrow}$
$$\big(\boxed{1}\big[\,\textit{entity}\,\big]\mathbf{and}\ \boxed{1}\big[\,\textit{entity}\,\big]\big)\ \mathbf{or}\ \big(\boxed{1}\big[\,\textit{truth}\,\big]\mathbf{and}\ \boxed{1}\big[\,\textit{truht}\,\big]\big)\ \mathbf{or}\ \big(\boxed{1}\big[\,\textit{w-index}\,\big]\mathbf{and}\ \boxed{1}\big[\,\textit{w-index}\,\big]\big),$$

- $\texttt{same-type}\,(\boxed{1},\boxed{2}) \ \overset{\forall}{\Longleftarrow}$
$$\boxed{1}\begin{bmatrix}\textit{c-type}\\ \text{IN}\ \boxed{3}\\ \text{OUT}\ \boxed{4}\end{bmatrix}\ \mathbf{and}\ \ \boxed{2}\begin{bmatrix}\textit{c-type}\\ \text{IN}\ \boxed{5}\\ \text{OUT}\ \boxed{6}\end{bmatrix}\ \mathbf{and}\ \ \texttt{same-type}\,(\boxed{3},\boxed{5})\ \ \mathbf{and}\ \ \texttt{same-type}\,(\boxed{4},\boxed{6});$$

relacja `same-term` definiowana jest w sposób następujący:

- $\texttt{same-term}\,(\boxed{1},\boxed{2}) \ \overset{\forall}{\Longleftarrow}$
$$\boxed{1}\begin{bmatrix}\textit{var}\\ \text{TYPE}\ \boxed{3}\\ \text{NUMBER}\ \boxed{4}\end{bmatrix}\ \mathbf{and}\ \ \boxed{2}\begin{bmatrix}\textit{var}\\ \text{TYPE}\ \boxed{5}\\ \text{NUMBER}\ \boxed{6}\end{bmatrix}\ \begin{array}{ll}\mathbf{and} & \texttt{same-type}\ (\boxed{3},\ \boxed{5})\\ \mathbf{and} & \texttt{same-number}\ (\boxed{4},\ \boxed{6}),\end{array}$$

- $\texttt{same-term}\,(\boxed{1},\boxed{2}) \ \overset{\forall}{\Longleftarrow}\ \ \big(\boxed{1}\big[\,\textit{const}_i\,\big]\mathbf{and}\ \boxed{1}\big[\,\textit{const}_i\,\big]\big)$ dla dowolnej $\textit{const}_i \sqsubseteq \textit{const}$,

- $\texttt{same-term}\,(\boxed{1},\boxed{2}) \ \overset{\forall}{\Longleftarrow}$
$$\boxed{1}\begin{bmatrix}\textit{appl}\\ \text{TYPE}\ \boxed{3}\\ \text{FUNC}\ \boxed{4}\\ \text{ARG}\ \boxed{5}\end{bmatrix}\ \mathbf{and}\ \ \boxed{2}\begin{bmatrix}\textit{appl}\\ \text{TYPE}\ \boxed{6}\\ \text{FUNC}\ \boxed{7}\\ \text{ARG}\ \boxed{8}\end{bmatrix}\ \begin{array}{ll}\mathbf{and} & \texttt{same-type}\ (\boxed{3},\ \boxed{6})\\ \mathbf{and} & \texttt{same-term}\ (\boxed{4},\ \boxed{7})\\ \mathbf{and} & \texttt{same-term}\ (\boxed{5},\ \boxed{8}),\end{array}$$

- $\texttt{same-term}\,(\boxed{1},\boxed{2}) \ \overset{\forall}{\Longleftarrow}$
$$\boxed{1}\begin{bmatrix}\textit{abstr}\\ \text{TYPE}\ \boxed{3}\\ \text{VAR}\ \boxed{4}\\ \text{ARG}\ \boxed{5}\end{bmatrix}\ \mathbf{and}\ \ \boxed{2}\begin{bmatrix}\textit{abstr}\\ \text{TYPE}\ \boxed{6}\\ \text{VAR}\ \boxed{7}\\ \text{ARG}\ \boxed{8}\end{bmatrix}\ \begin{array}{ll}\mathbf{and} & \texttt{same-type}\ (\boxed{3},\ \boxed{6})\\ \mathbf{and} & \texttt{same-term}\ (\boxed{4},\ \boxed{7})\\ \mathbf{and} & \texttt{same-term}\ (\boxed{5},\ \boxed{8}),\end{array}$$

- $\texttt{same-term}\,(\boxed{1},\boxed{2}) \ \overset{\forall}{\Longleftarrow}$
$$\boxed{1}\begin{bmatrix}\textit{equ}\\ \text{TYPE}\ \boxed{3}\\ \text{ARG1}\ \boxed{4}\\ \text{ARG2}\ \boxed{5}\end{bmatrix}\ \mathbf{and}\ \ \boxed{2}\begin{bmatrix}\textit{equ}\\ \text{TYPE}\ \boxed{6}\\ \text{ARG1}\ \boxed{7}\\ \text{ARG2}\ \boxed{8}\end{bmatrix}\ \begin{array}{ll}\mathbf{and} & \texttt{same-type}\ (\boxed{3},\ \boxed{6})\\ \mathbf{and} & \texttt{same-term}\ (\boxed{4},\ \boxed{7})\\ \mathbf{and} & \texttt{same-term}\ (\boxed{5},\ \boxed{8}),\end{array}$$

zaś relacja `component` w sposób następujący:

- $\texttt{component}\,(\boxed{1},\boxed{2}) \ \overset{\forall}{\Longleftarrow}\ \ \boxed{1} \approx \boxed{2},$

- $\texttt{component}\,(\boxed{1},\boxed{2}) \ \overset{\forall}{\Longleftarrow}$
$$\Big(\boxed{1}\big[\text{IN}\ \boxed{3}\big]\ \mathbf{or}\ \boxed{1}\big[\text{OUT}\ \boxed{3}\big]\ \mathbf{or}\ \boxed{1}\big[\text{NUMBER}\ \boxed{3}\big]\ \mathbf{or}\ \boxed{1}\big[\text{ARG}\ \boxed{3}\big]\ \mathbf{or}\ \boxed{1}\big[\text{ARG1}\ \boxed{3}\big]\ \mathbf{or}\ \boxed{1}\big[\text{ARG2}\ \boxed{3}\big]\ \mathbf{or} \\ \boxed{1}\big[\text{FUNC}\ \boxed{3}\big]\ \mathbf{or}\ \boxed{1}\big[\text{VAR}\ \boxed{3}\big]\Big)\ \ \mathbf{and}\ \ \texttt{component}\,(\boxed{1},\boxed{3})$$

$$\begin{bmatrix} \textit{equ} & \\ \textsc{type} & \textit{truth} \\ \textsc{arg1} & \begin{bmatrix} \textit{abstr} & \\ \textsc{type} & \boxed{1} \begin{bmatrix} \textit{c-type} \\ \textsc{in}\ \boxed{2}\textit{type} \\ \textsc{out}\ \textit{truth} \end{bmatrix} \\ \textsc{var} & \boxed{3} \begin{bmatrix} \textsc{type}\ \boxed{2} \\ \textsc{number}\ \textit{2} \end{bmatrix} \\ \textsc{arg} & \boxed{4} \begin{bmatrix} \textit{me} \\ \textsc{type}\ \textit{truth} \end{bmatrix} \end{bmatrix} \\ \textsc{arg2} & \begin{bmatrix} \textit{abstr} & \\ \textsc{type} & \boxed{1} \\ \textsc{var} & \boxed{3} \\ \textsc{arg} & \begin{bmatrix} \textit{equ} & \\ \textsc{type} & \textit{truth} \\ \textsc{arg1} & \boxed{5} \begin{bmatrix} \textit{abstr} & \\ \textsc{type} & \begin{bmatrix} \textit{c-type} \\ \textsc{in}\ \textit{truth} \\ \textsc{out}\ \textit{truth} \end{bmatrix} \\ \textsc{var} & \boxed{6} \begin{bmatrix} \textsc{type}\ \textit{truth} \\ \textsc{number}\ \textit{6} \end{bmatrix} \\ \textsc{arg} & \boxed{6} \end{bmatrix} \\ \textsc{arg2} & \boxed{5} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Zapis języka Ty2 jako gramatyki $\mathcal{TY}2$ tylko pozornie został zakończony. Nierozpatrzone pozostały bowiem definicje operatorów i kwantyfikatorów logicznych z rozdz. 4.1.1. Niestety, ich zapis w RSRL czy w postaci AVM jest bardzo złożony. Na przykład zapis formuły $\forall x_\tau\, \varphi$ przedstawiony jest obok, przy czym zmienna x oznaczona jest przez $\boxed{3}$, a jej typ τ przez $\boxed{2}$, zaś φ odpowiada ARG $\boxed{4}$. Rzecz jasna nie sposób posługiwać się takimi strukturami, więc autorzy (niejawnie) wprowadzili skróty notacyjne. I tak kwantyfikatory mają postać następujących prostych struktur:
$\begin{bmatrix} \textit{universal} \\ \textsc{var}\ \textit{var} \\ \textsc{scope}\ \textit{me} \end{bmatrix}$ oraz $\begin{bmatrix} \textit{existential} \\ \textsc{var}\ \textit{var} \\ \textsc{scope}\ \textit{me} \end{bmatrix}$. Definiowany jest także $\textit{quantifier} =_{def} \textit{universal} \lor \textit{existential}$.
Podobnie $\textit{l-const} =_{def} \textit{conjunction} \lor \textit{disjunction} \lor \textit{implication} \lor \textit{equivalence}$, przy czym mamy także następujące definicje: $\begin{bmatrix} \textit{l-const} \\ \textsc{arg1}\ \textit{me} \\ \textsc{arg1}\ \textit{me} \end{bmatrix}$ oraz $\begin{bmatrix} \textit{negation} \\ \textsc{arg1}\ \textit{me} \end{bmatrix}$, przy czym wszystkie *me* w powyższych wyrażeniach są typu *truth*.

W praktyce, dla dodatkowego zwiększenia czytelności, Richter i Sailer w celu zapisu obiektów rodzaju *me* (formuł $\mathcal{TY}2$) posługują się standardowymi symbolami logicznymi. Ponadto używają skrótu $Q\,[\rho \circ \nu]$ dla zapisania struktury widniejącej po prawej. Cały czas należy jednak pamiętać, że za tymi skrótami notacyjnymi kryją się złożone obiekty $\mathcal{TY}2$.

$$\begin{bmatrix} \textit{quantifier} & \\ \textsc{var} & \textit{var} \\ \textsc{scope} & \begin{bmatrix} \textit{l-const} \\ \textsc{arg1}\ \rho \\ \textsc{arg2}\ \nu \end{bmatrix} \end{bmatrix}.$$

4.5 LRS — nieciągła reprezentacja semantyczna

Okazuje się, że także i powyższe narzędzia umożliwiają reprezentację semantyczną na kilka sposobów. Różnice między nimi polegają przede wszystkim na metodzie reprezentowania wyrażeń niejednoznacznych semantycznie (np. kwestia zakresu kwantyfikatorów). Richter i Sailer (2003) porównują ze sobą kilka takich reprezentacji. Są to Ty2U [Richter i Sailer, 1999a], LF-Ty2 [Richter i Sailer, 1999b; 1999c] oraz LRS [Richter i Sailer, 2001a; 2001b]. Nie zamierzamy przytaczać tu tego porównania; wystarczy powiedzieć, że LRS jest ostatnim z zaproponowanych przez nich formalizmów, opracowywanych i przerabianych przez kilka lat, i że sami twórcy uznają go za najlepiej przystający do wymagań stawianych przez reprezentację semantyczną wyrażeń języka naturalnego.

Tak jak zwykle w HPSG czy RSRL, zaczniemy od zdefiniowania rodzaju *lrs*. Reprezentacja tego rodzaju widnieje obok, przy czym EX-CONT jest skrótem od EXTERNAL-CONTENT, zaś INCONT od INTERNAL-CONTENT. Wartość atrybutu INCONT wyraża wkład semantyczny słowa, zaś atrybutu EXCONT — całościową postać logiczną frazy. W końcu w atrybucie PARTS wyliczone są poszczególne podwyrażenia wnoszone przez daną frazę (co może wykraczać poza bieżącą frazę, a więc antycypuje obecność elementów znajdujących się poza nią).

$$\begin{bmatrix} \textit{lrs} & \\ \textsc{excont} & \textit{me} \\ \textsc{incont} & \textit{me} \\ \textsc{parts} & \textit{list}(\textit{me}) \end{bmatrix}$$

Naturalne wydawałoby się, by obiekty rodzaju *lrs* były wartościami atrybutu CONT wprowadzonego w rozdz. 4.2.1. Jednak atrybut ten pełni ważną rolę w wielu gramatykach napisanych w formalizmie HPSG. Ostatecznie autorzy, mając na względzie także inne własności HPSG, postanowili rozbudować sygnaturę tak, by obiekty rodzaju *lrs* były wartościami nowego atrybutu rodzaju *sign* oznaczanego przez LF (skrót od LOGICAL FORM). Zmodyfikowana postać rodzaju *sign* widnieje powyżej po prawej. Z kolei rodzaj *content* posiada strukturę przedstawioną obok, z uwzględnieniem struktury rodzaju *ext-index* (skrót od *extended-index*).

$$\begin{bmatrix} \textit{sign} & \\ \textsc{phon} & \textit{list}(\textit{orth}) \\ \textsc{synsem} & \textit{synsem} \\ \textsc{lf} & \textit{lrs} \end{bmatrix}$$

$$\begin{bmatrix} \textit{conent} & \\ \textsc{index} & \textit{ext-index} \\ \textsc{main} & \textit{term} \end{bmatrix}$$

$$\begin{bmatrix} \textit{ext-index} & \\ \textsc{phi} & \textit{index} \\ \textsc{var} & \textit{term} \end{bmatrix}$$

Ponieważ wyrażenia rodzaju *lrs* jako wartości obiektu LF są integralną częścią struktur HPSG, niemożliwością jest stworzenie jakiejkolwiek translacji struktur będących rozbiorami zdań czy składających się na nie fraz na Ty2: nie można przecież przekładać jakiejkolwiek całości na jej część (jedynym możliwym w takiej sytuacji przekształceniem jest rzutowanie). Możliwe jest co prawda sformułowanie translacji jednych fragmentów struktury (reprezentujących informację syntaktyczną) na inne (reprezentujące informację semantyczną). Ale działanie HPSG (a więc i LRS) nie na tym polega. Podobnie możemy wyodrębnić proces interepretacji zwrotów j. naturalnego za pomocą LRS z całego procesu wyodrębniania poprawnych struktur tylko w taki sposób, że najpierw wybierzemy struktury spełniające „warunki syntaktyczne", a dopiero spośród nich te spełniające także „warunki semantyczne". Pamiętać jednak należy, że kolejność aplikowania poszczególnych ograniczeń nie ma wpływu na ostateczny wynik, tzn. na zbiór struktur spełniających je wszystkie. Pamiętając o tym, w dalszych rozważaniach zamierzamy się skupić na analizie semantycznej, zwyczajnie ignorując syntaktyczną część całego procesu.

W tym miejscu muszą pojawić się zasady zapewniające poprawność wprowadzonych pojęć.

Zasada 4.5. Dla rodzaju *lrs* muszą być spełnione następujące zasady:

- Zasada zawartości wewnętrznej (ang. *The Incont Value Principle*, (IContP)):

$$\begin{bmatrix} \textit{lrs} \\ \textsc{excont}\ \boxed{1} \\ \textsc{incont}\ \boxed{2} \\ \textsc{parts}\ \boxed{3} \end{bmatrix} \Longrightarrow \texttt{member}\,(\boxed{2}, \boxed{3}) \quad \textbf{and} \quad \texttt{component}\,(\boxed{2}, \boxed{1});$$

- Zasada zawartości zewnętrznej (ang. *The Excont Value Principle*, (EContP)):

$$1.\ \begin{bmatrix} \textit{phrase} \\ \textsc{nhd-dtrs}\ \boxed{1} \end{bmatrix} \Longrightarrow$$

$$\mathbf{A}\,\boxed{2}\left(\texttt{member}(\boxed{2},\boxed{1})\ \textbf{and}\ \boxed{2}\begin{bmatrix}\textsc{lf}\begin{bmatrix}\textsc{excont}\ \boxed{3} \\ \textsc{parts}\ \boxed{4}\end{bmatrix}\end{bmatrix} \Longrightarrow \texttt{member}(\boxed{3},\boxed{4})\right);$$

$$2.\ \boxed{3}\begin{bmatrix} \textit{sign} \\ \textsc{lf}\begin{bmatrix}\textsc{excont}\ \boxed{1} \\ \textsc{parts}\ \boxed{2}\end{bmatrix}\end{bmatrix}\ \textbf{and}\ \begin{bmatrix} \textit{phrase} \\ \textsc{hd-dtr}\ \boxed{4} \\ \textsc{nhd-dtrs}\ \boxed{5} \end{bmatrix}\ \textbf{and}\ \ \textbf{not}\,(\boxed{3} \approx \boxed{4})\ \ \textbf{and}$$

$$\mathbf{A}\,\boxed{6}\left(\texttt{member}(\boxed{6},\boxed{5}) \Longrightarrow \textbf{not}\ (\boxed{3} \approx \boxed{6})\right) \ \Longrightarrow\ \mathbf{A}\,\boxed{6}\left(\texttt{member}(\boxed{6},\boxed{2}) \Longleftrightarrow \texttt{component}(\boxed{6},\boxed{1})\right);$$

- Zasada Semantyczna (ang. *Semantics Principle*, (SP)):

$$1.\ \textit{phrase} \Longrightarrow \begin{bmatrix} \textsc{lf}\begin{bmatrix}\textsc{incont}\ \boxed{1} \\ \textsc{excont}\ \boxed{2}\end{bmatrix} \\ \textsc{hd-dtr}|\textsc{lf}\begin{bmatrix}\textsc{incont}\ \boxed{1} \\ \textsc{excont}\ \boxed{2}\end{bmatrix}\end{bmatrix};$$

$$2.\ \begin{bmatrix} \textit{phrase} \\ \textsc{lflf}|\textsc{parts}\ \boxed{1} \\ \textsc{hd-dtr}|\textsc{lf}|\textsc{parts}\ \boxed{2} \\ \textsc{nhd-dtrs}\ \boxed{3} \end{bmatrix} \Longrightarrow$$

$$\Bigg(\texttt{member}(\boxed{4},\boxed{1}) \quad \Longleftrightarrow$$

$$\texttt{member}(\boxed{4},\boxed{2})\ \textbf{or}\ \mathbf{E}\,\boxed{5}\left(\texttt{member}(\boxed{5},\boxed{3})\ \textbf{and}\ \boxed{5}\big[\ \textsc{lf}|\textsc{parts}\ \boxed{6}\big]\ \textbf{and}\ \texttt{member}(\boxed{4},\boxed{6})\right)\Bigg).$$

Tak więc dla dowolnego *lrs*, zasada IContP mówi, że wartość atrybutu INCONT jest składnikiem listy PART oraz podwyrażeniem wartości atrybutu EXCONT. Następnie zasada EContP stwierdza, że, po pierwsze, dla atrybutu NHD-DTRS dowolnej frazy wartością jego atrybutu EXCONT jest dokładnie jeden ze składników jej listy PART, a po drugie dla dowolnego skompletowanego wypowiedzenia (tzn. takiego, które nie jest niczyim składnikiem), każde podwyrażenie jego atrybutu EXCONT jest składnikiem jego listy PART i *vice versa*. Na koniec dla dowolnej frazy z elementem głównym zasada SP wymaga, by, po pierwsze, zarówno wartość atrybutu INCONT, jak i wartość atrybutu EXCONT, były identyczna dla rozważanej struktury i jej elementu głównego, a po drugie by wartości atrybutu PART frazy stanowiły wyłącznie wszystkie składniki wartości tegoż atrybutu jej wszystkich synów.

Zasada SP posiada jeszcze dalsze podzasady. Ponieważ jednak wiążą się one z tworzeniem interpretacji semantycznej dla konkretnych fraz, zdecydowaliśmy się na ich wyodrębnienie i omówienie po kolei, wraz z konkretnymi przykładami.

4.5.1 Słownik

Bardzo ważną częścią gramatyki HPSG jest słownik. Zgodnie z konstrukcją HPSG musi on być zdefiniowany jako część sygnatury i/lub jako zbiór ograniczeń. Rozwiązaniem często przyjmowanym w literaturze HPSG jest zdefiniowanie słownika jako hierarchii typów poniżej typu *word*.

Informacja zawarta w hasłach słownikowych ma kluczowe znaczenie dla analizy w HPSG bardziej złożonych zwrotów j. naturalnego — to ona wszak determinuje (po spełnieniu narzuconych ograniczeń), jakie zwroty zostaną w HPSG zaakceptowane i jakie struktury zostaną im przypisane.

Tak więc podobnie jak w poprzednich rozdziałach, poniżej przedstawimy zestaw podstawowych wyrazów, niezbędnych do ukazania własności interpretacji semantycznej złożonych z nich zwrotów. Informacje syntaktyczne potraktujemy jednak bardzo skrótowo i za [Richter i Sailer, 2003] skupimy się głównie na atrybucie LF o wartości rodzaju *lrs*, uwzględniając ponadto inne wybrane dane, które mogą okazać się przydatne. W szczególności zignorujemy atrybur PHI zawierający informację morfologiczną, rozważany w [Richter i Sailer, 2003] jedynie w zmodyfikowanej wersji LRS.

read:

$$\begin{bmatrix} \textit{word} \\ \text{PHON } \langle \textit{read} \rangle \\ \text{SYNS|LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD } \textit{verb} \\ \text{SUBCAT} \left\langle \begin{matrix} \begin{bmatrix} \text{LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD } \textit{noun} \\ \text{SUBCAT } \langle\rangle \end{bmatrix} \\ \text{CONT|INDEX|VAR } x \end{bmatrix} \end{bmatrix}, \\ \begin{bmatrix} \text{LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD } \textit{noun} \\ \text{SUBCAT } \langle\rangle \end{bmatrix} \\ \text{CONT|INDEX|VAR } x \end{bmatrix} \end{bmatrix} \end{matrix} \right\rangle \end{bmatrix} \\ \text{CONT|MAIN } \mathsf{read}' \end{bmatrix} \\ \text{LF} \begin{bmatrix} \textit{lrs} \\ \text{EXCONT } me \\ \text{INCONT } \boxed{1}\mathsf{read}'(y)(x) \\ \text{PARTS } \langle x, y, \boxed{1}, \boxed{1a}\mathsf{read}'(y), \boxed{1b}\mathsf{read}' \rangle \end{bmatrix} \end{bmatrix}$$

John:

$$\begin{bmatrix} \textit{word} \\ \text{PHON } \langle \textit{John} \rangle \\ \text{SYNS|LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD } \textit{noun} \\ \text{SUBCAT } \langle\rangle \end{bmatrix} \\ \text{CONT} \begin{bmatrix} \textit{content} \\ \text{INDEX|VAR } \boxed{3}j \\ \text{MAIN } \boxed{3} \end{bmatrix} \end{bmatrix} \\ \text{LF} \begin{bmatrix} \textit{lrs} \\ \text{EXCONT } me \\ \text{INCONT } \boxed{3} \\ \text{PARTS } \langle \boxed{3} \rangle \end{bmatrix} \end{bmatrix}$$

walk:

$$\begin{bmatrix} \textit{word} \\ \text{PHON } \langle \textit{walk} \rangle \\ \text{SYNS|LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD } \textit{verb} \\ \text{SUBCAT} \left\langle \begin{bmatrix} \text{LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD } \textit{noun} \\ \text{SUBCAT } \langle\rangle \end{bmatrix} \\ \text{CONT|INDEX|VAR } x \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{CONT|MAIN } \mathsf{walk}' \end{bmatrix} \\ \text{LF} \begin{bmatrix} \textit{lrs} \\ \text{EXCONT } me \\ \text{INCONT } \boxed{2}\mathsf{walk}'(x) \\ \text{PARTS } \langle x, \boxed{2}, \boxed{2a}\mathsf{walk}' \rangle \end{bmatrix} \end{bmatrix}$$

Ulisses:

$$\begin{bmatrix} \textit{word} \\ \text{PHON } \langle \textit{Ulisses} \rangle \\ \text{SYNS|LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD } \textit{noun} \\ \text{SUBCAT } \langle\rangle \end{bmatrix} \\ \text{CONT} \begin{bmatrix} \textit{content} \\ \text{INDEX|VAR } \boxed{3}u \\ \text{MAIN } \boxed{3} \end{bmatrix} \end{bmatrix} \\ \text{LF} \begin{bmatrix} \textit{lrs} \\ \text{EXCONT } me \\ \text{INCONT } \boxed{4} \\ \text{PARTS } \langle \boxed{4} \rangle \end{bmatrix} \end{bmatrix}$$

book:

$$\begin{bmatrix} \textit{word} \\ \text{PHON } \langle \textit{book} \rangle \\ \text{SS|L} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD } \textit{noun} \\ \text{SUBCAT} \left\langle \begin{bmatrix} \text{LOC} \begin{bmatrix} \text{CAT|HEAD } \textit{det} \\ \text{CONT|INDEX|VAR } x \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{CONT|MAIN } \mathsf{book}' \end{bmatrix} \\ \text{LF} \begin{bmatrix} \textit{lrs} \\ \text{EXCONT} \begin{bmatrix} \textit{quantifier} \\ \text{VAR } y \end{bmatrix} \\ \text{INCONT } \boxed{5}\mathsf{book}'(y) \\ \text{PARTS } \langle y, \boxed{5}, \boxed{5a}\mathsf{book}' \rangle \end{bmatrix} \end{bmatrix}$$

student:

$$\begin{bmatrix} \textit{word} \\ \text{PHON } \langle \textit{student} \rangle \\ \text{SS|L} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD } \textit{noun} \\ \text{SUBCAT} \left\langle \begin{bmatrix} \text{LOC} \begin{bmatrix} \text{CAT|HEAD } \textit{det} \\ \text{CONT|INDEX|VAR } x \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{CONT|MAIN } \mathsf{student}' \end{bmatrix} \\ \text{LF} \begin{bmatrix} \textit{lrs} \\ \text{EXCONT} \begin{bmatrix} \textit{quantifier} \\ \text{VAR } x \end{bmatrix} \\ \text{INCONT } \boxed{6}\mathsf{student}'(x) \\ \text{PARTS } \langle x, \boxed{6}, \boxed{6a}\mathsf{student}' \rangle \end{bmatrix} \end{bmatrix}$$

a:

$$\left[\begin{array}{l} \textit{word} \\ \text{PHON } \langle a \rangle \\ \text{SS|L|CAT} \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \textit{det} \\ \text{SPEC} \left[\begin{array}{l} \textit{synsem} \\ \text{L|C} \left[\begin{array}{l} \text{HEAD } \textit{noun} \\ \text{SBC } \langle [\textit{synsem}] \rangle \end{array}\right] \end{array}\right] \end{array}\right] \\ \text{SUBCAT } \langle\rangle \end{array}\right] \\ \text{LF} \left[\begin{array}{l} \textit{lrs} \\ \text{EXCONT } me \\ \text{INCONT } \boxed{7} \exists y\,[\alpha \,\&\, \beta] \\ \text{PARTS } \langle y, \boxed{7}, \boxed{7a}[\alpha \,\&\, \beta] \rangle \end{array}\right] \end{array}\right]$$

every:

$$\left[\begin{array}{l} \textit{word} \\ \text{PHON } \langle \textit{every} \rangle \\ \text{SS|L|CAT} \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \textit{det} \\ \text{SPEC} \left[\begin{array}{l} \textit{synsem} \\ \text{L|C} \left[\begin{array}{l} \text{HEAD } \textit{noun} \\ \text{SBC } \langle [\textit{synsem}] \rangle \end{array}\right] \end{array}\right] \end{array}\right] \\ \text{SUBCAT } \langle\rangle \end{array}\right] \\ \text{LF} \left[\begin{array}{l} \textit{lrs} \\ \text{EXCONT } me \\ \text{INCONT } \boxed{8} \forall x\,[\gamma \rightarrow \delta] \\ \text{PARTS } \langle x, \boxed{8}, \boxed{8a}[\gamma \rightarrow \delta] \rangle \end{array}\right] \end{array}\right]$$

doesn't:

$$\left[\begin{array}{l} \textit{word} \\ \text{PHON } \langle \textit{doesn't} \rangle \\ \text{SYNS|LOC} \left[\begin{array}{l} \text{CAT} \left[\begin{array}{l} \text{HEAD } \textit{part} \\ \text{SUBCAT } \langle [\text{LOC|CAT|HEAD } \textit{det}] \rangle \end{array}\right] \\ \text{CONT|MAIN } \neg \end{array}\right] \\ \text{LF} \left[\begin{array}{l} \textit{lrs} \\ \text{EXCONT } me \\ \text{INCONT } \boxed{10} \neg \xi \\ \text{PARTS } \langle \boxed{10} \rangle \end{array}\right] \end{array}\right]$$

Tabela 4.1: Skrótowa reprezentacja wybranych haseł słownikowych

Richter i Sailer stwierdzają, że w większości haseł słownikowych (poza rodzajnikami) wartości atrybutu INCONT jednoznacznie identyfikują pewne wyrażenia Ty2 »z dokładnością nazw zmiennych«. Należy to rozumieć, że zmienne x, y posiadają jedynie etykiety, i mogą następnie podlegać unifikacji ze zmiennymi oznakowanymi innymi symbolami. Należy jednak pamiętać, że nie dotyczy to zmiennych j, u.

W haśle słownikowym *read* wartość atrybutu EXCONT pozostaje niewyspecyfikowana. Wartość atrybutu INCONT to wkład semantyczny wyrazu jako takiego (jego „wartość semantyczna" odpowiadająca translacji $\mathfrak{g}$ zgodnie z podejściem w poprzednich rozdziałach), wraz z wymaganymi przezeń argumentami. Atrybut PART zawiera dokładnie wszystkie podwyrażenia wyrażenia będącego wartość atrybutu INCONT. W wersji zmodyfikowanej LRS czasowniki uzyskują dodatkowy argument e oznaczający zdarzenie, w niniejszym omówieniu ignorowany.

W hasłach słownikowych rzeczowników pospolitych *student* oraz *book* reprezentacja atrybutów INCONT i PART jest analogiczna jak w przypadku czasowników. Natomiast wartość atrybutu EXCONT narzuca istnienie pewnego kwantyfikatora wiążącego zmienną pojawiającą się także w pozostałych atrybutach.

W przypadku haseł kwantyfikatorów *every* oraz *a* wartość INCONT nie identyfikuje jednoznacznie wyrażenia Ty2, a jedynie określa jego główne elementy. I tak w przypadku *every* jest to wyrażenie rozpoczynające się kwantyfikatorem uniwersalnym kwantyfikującym pewną implikację wiążącą dwa w żaden sposób nie sprecyzowane wyrażenia. Jest to uproszczona notacja bardziej skomplikowanego wyrażenia; ponieważ jednak składające się nań obiekty i ich atrybuty nie są (w dostępnych mi pracach) opisane, szczegóły techniczne zostaną tu pominięte.

Wypadałoby jeszcze wspomnieć, że autorzy opisują wyraz *some* w miejsce *a*. Ponieważ jednak zarówno Montague, jak i Kamp i Reyle, opisywali zdania z rodzajnikiem nieokreślonym *a*, w niniejszym opracowaniu dostosowujemy się do tej konwencji.

4.5.2 Semantyczne reguły dla poszczególnych fraz

Po omówieniu podstaw gramatyki HPSG, formalizmu Ty2 mającego służyć do interpretacji semantycznej sformułowań j. naturalnego wraz ze sposobem jego kodowania w gramatyce oraz zaprezentowania minimalnego słowniczka zawierającego w szczególności zakodowaną informację semantyczną o umieszczonych w nim wyrazach przyszedł wreszcie czas na pokazanie, jak działa LRS, czyli w jaki sposób określana jest interpretacja semantyczna fraz na podstawie interpretacji semantycznej ich podwyrażeń.

Zasada „rodzajnik-rzeczownik pospolity"

Zaczniemy od utworzenia frazy rzeczownikowej poprzez połączenie rodzajnika z rzeczownikiem pospolitym. Nie wnikając w syntaktyczne szczegóły tego procesu, zauważmy jedynie, że elementem głównym powstałej frazy jest rzeczownik. Natomiast sposób uzyskania poprawnej wartości atrybutu CONT warunkuje następująca zasada.

Zasada 4.6. Zasada dla kwantfikowanej frazy rzeczownikowej (ang. *Quantified Noun Phrase Principle*, (QNP)) opisywana jest przez następującą zależność:

$$\begin{bmatrix} \textit{phrase} \\ \textsc{nhd-dtrs}\ \left\langle \left[\ \textsc{synsem|local|cat|head}\ \textit{det}\right] \right\rangle \end{bmatrix} \Longrightarrow$$

$$\begin{bmatrix} \textsc{hd-dtr} \begin{bmatrix} \textsc{synsem|local|cat|head}\ \textit{noun} \\ \textsc{lf} \begin{bmatrix} \textsc{excont}\ \boxed{1} \\ \textsc{incont}\ \boxed{2} \end{bmatrix} \end{bmatrix} \\ \textsc{nhd-dtrs}\ \left\langle \left[\ \textsc{lf|excont}\ \boxed{1} Q\,[\rho \circ \nu]\right] \right\rangle \end{bmatrix} \quad \textbf{and} \quad \texttt{component}\ (\boxed{2}, \rho).$$

Zasada ta mówi, że jeśli element poboczny (NHD-DTRS) jest rodzajnikiem, to wartość jego atrybutu EXCONT jest postaci $Q\,[\rho \circ \nu]$ i jest równa wartości tegoż atrybutu elementu głównego, którego z kolei wartość atrybutu INCONT jest podwyrażeniem ρ. Autorzy nie zdecydowali się narzucić mocniejszego warunku równości pomiędzy tymi formułami (co można zrobić zapisując po prostu $\boxed{2}\rho$ i rezygnując z `component` ($\boxed{2}$, ρ)), gdyż wpłynęłoby to na sposób traktowania zakresu kwantyfikatorów.

Warunki postaci `component` (α, β) nie dają się wstawić bezpośrednio do struktury. Nalży więc pamiętać, że prawdziwość struktur ze wszystkich poniższych przykładów jest weryfikowana wyłącznie w koniunkcji z odpowiednimi pomocniczymi wyrażeniami tej postaci.

Przykład *4.3.* Rozważmy rodzajnik *a* oraz rzeczownik pospolity *book*. Część semantyczna struktury reprezentującej frazę *a book* spełniająca wszystkie zasady wygląda następująco:

$$\begin{bmatrix} \textsc{lf} \begin{bmatrix} \textsc{excont}\ \boxed{7} \\ \textsc{incont}\ \boxed{5} \\ \textsc{parts}\ \langle y, \boxed{5}, \boxed{5a}, \boxed{7}, \boxed{7a} \rangle \end{bmatrix} \\ \textsc{hd-dtr|lf} \begin{bmatrix} \textsc{excont}\ \boxed{7} \\ \textsc{incont}\ \boxed{5}\mathsf{book}'(y) \\ \textsc{parts}\ \langle y, \boxed{5}, \boxed{5a}\mathsf{book}' \rangle \end{bmatrix} \\ \textsc{nhd-dtrs} \left\langle \left[\textsc{lf} \begin{bmatrix} \textsc{excont}\ \boxed{7} \\ \textsc{incont}\ \boxed{7}\exists y\,[\alpha\,\&\,\beta] \\ \textsc{parts}\ \langle y, \boxed{7}, \boxed{7a}[\alpha\,\&\,\beta] \rangle \end{bmatrix} \right] \right\rangle \end{bmatrix}$$

Zasady QNP i SP.1 narzucają, by wartości wszystkich atrybutów EXCONT były równe, zaś zasada EContP i SP.2 wymuszają, by wartością tą było $\boxed{7}$. Ponieważ zasada QNP wymaga także, by `component` ($\boxed{5}, \alpha$), uzyskujemy formułę $\exists y\,[...\mathsf{book}'(y)...\,\&\,\beta]$. Rzecz jasna zasada EContP.2 jest spełniona w sposób trywialny.

Podobną strukturę uzyskamy dla frazy *every student*.

Zasada „kwantyfikowana fraza rzeczownikowa-czasownik"

Następna zasada polega na przyłączeniu kwantyfikowanej frazy rzeczownikowej do jej nadrzędnika (czyli frazy czasownikowej). Ciekawe jest to, że stosuje się ją zarówno do podmiotu, jak i dopełnień. Rzecz jasna elementem głównym całej frazy jest czasownik. Poniżej przedstawimy zasadę zabezpieczającą poprawność wartości atrybutu LF.

Zasada 4.7. Zasada przyłączenia kwantfikowanej frazy rzeczownikowej do frazy czasownikowej (ang. *Quantified Noun Phrase–Verb Phrase Principle*, (QNVP)) opisywana jest przez następującą zależność:

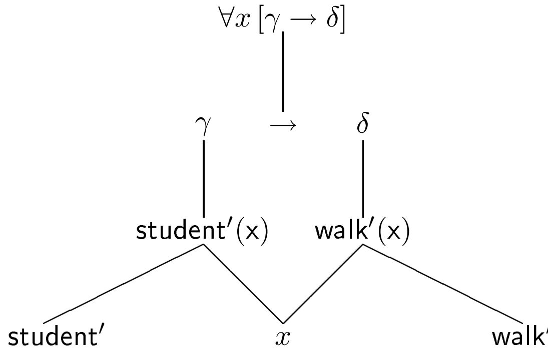
$$\begin{bmatrix} \textit{phrase} \\ \textsc{hd-dtr|lf|incont}\ \boxed{1} \\ \textsc{nhd-dtrs}\ \boxed{2} \end{bmatrix} \ \textbf{and}\ \ \textbf{E}\boxed{3} \left(\texttt{member}(\boxed{3},\boxed{2})\ \textbf{and}\ \boxed{3} \begin{bmatrix} \textit{phrase} \\ \textsc{syns|loc|cat|head}\ \textit{noun} \\ \textsc{lf|excont}\ Q\,[\rho \circ \nu] \end{bmatrix} \right)$$

$$\Longrightarrow \quad \texttt{component}\ (\boxed{1}, \nu).$$

Zasada ta mówi, że jeżeli element poboczny jest kwantyfikowaną frazą rzeczownikową mającą wartość atrybutu EXCONT postaci $Q\,[\rho \circ \nu]$, to wartość atrybutu INCONT elementu głównego (czyli czasownik) jest podwyrażeniem ν.

Przykład *4.4.* Weźmy zdanie *Every student walks.*. Część semantyczna reprezentującej go struktury (bez wchodzenia wgłąb elementu pobocznego) spełniająca wszystkie zasady jest widoczna po lewej:

$$\left[\begin{array}{ll} \text{LF} & \left[\begin{array}{ll} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{2} \\ \text{PARTS} & \langle x, \boxed{2}, \boxed{2a}, \boxed{6}, \boxed{6a}, \boxed{8}, \boxed{8a} \rangle \end{array}\right] \\ \text{HD-DTR}|\text{LF} & \left[\begin{array}{ll} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{2}\mathsf{walk}'(x) \\ \text{PARTS} & \langle x, \boxed{2}, \boxed{2a}\mathsf{walk}' \rangle \end{array}\right] \\ \text{NHD-DTRS} & \left\langle \left[\begin{array}{ll} \text{LF} & \left[\begin{array}{ll} \text{EXCONT} & \boxed{8} \\ \text{INCONT} & \boxed{6}\mathsf{student}'(x) \\ \text{PARTS} & \begin{array}{l} \langle\, x, \boxed{6}, \boxed{6a}\mathsf{student}', \\ \boxed{8}\forall x\,[\gamma \to \delta], \boxed{8a}\,[\gamma \to \delta]\,\rangle \end{array} \end{array}\right] \end{array}\right] \right\rangle \end{array}\right]$$



Ponieważ ponadto zachodzi `component` ($\boxed{6}$, γ) oraz `component` ($\boxed{2}$, δ), wstępnie uzyskujemy formułę $\forall x\,[...\mathsf{student}'(x)... \to ...\mathsf{walk}'(x)...]$. Dalsze rozumowanie przebiega jak następuje: Na liście PART frazy właściwej mamy $x, \mathsf{student}'(x), \mathsf{student}', \forall x\,[\gamma \to \delta], [\gamma \to \delta], \mathsf{walk}'(x), \mathsf{walk}'$. I zgodnie z regułą SP.2 nic więcej tam się znaleźć nie może. Ponadto reguła EContP.2 narzuca, by ostateczny kształt formuły obejmował wyłącznie wyrażenia z listy PART. Ponieważ ani γ, ani δ na tej liście nie występują, muszą być więc równe któremuś z elementów listy. Relacja podwyrażeń `component` elementów listy PART (symbolicznie oznaczana przez $\lhd$) tworzy hierarchię widoczną po prawej. Jest oczywiste, że jedynie formuła $\forall x\,[\mathsf{student}'(x) \to \mathsf{walk}'(x)]$ spełnia wszystkie te warunki równocześnie.

Zauważmy, że gdyby γ i δ znajdowały się na liście PART, mogłyby zawierać bardziej skomplikowane podformuły. Na przykład $\gamma = \neg\mathsf{student}'(x)$ oraz $\delta = \mathsf{student}'(x)\,\&\,\mathsf{walk}'(x)$ dawałyby w rezultacie formułę $\forall x\,[\neg\mathsf{student}'(x) \to \mathsf{student}'(x)\,\&\,\mathsf{walk}'(x)]$, która nie tylko nie jest interpretacją zdania *Every student walks.*, lecz w ogóle jest formułą sprzeczną. Należy więc zwrócić uwagę na fakt, jak ważne jest optymalne dobranie składu listy PART wyrazów: warunkuje to *de facto* poprawne zinterpretowanie zdania.

Przykład *4.5.* Ponieważ reguła ta działa także dla dopełnienia, rozważmy frazę *reads a book.* Część semantyczna reprezentującej go struktury (bez wchodzenia wgłąb elementu pobocznego) spełniająca wszystkie zasady jest następująca:

$$\left[\begin{array}{ll} \text{LF} & \left[\begin{array}{ll} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{1} \\ \text{PARTS} & \langle x, y, \boxed{1}, \boxed{1a}, \boxed{1b}, \boxed{5}, \boxed{5a}, \boxed{7}, \boxed{7a} \rangle \end{array}\right] \\ \text{HD-DTR}|\text{LF} & \left[\begin{array}{ll} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{1}\mathsf{read}'(y)(x) \\ \text{PARTS} & \langle x, y, \boxed{1}, \boxed{1a}\mathsf{read}'(y), \boxed{1b}\mathsf{read}' \rangle \end{array}\right] \\ \text{NHD-DTRS} & \left\langle \Phi, \left[\begin{array}{ll} \text{LF} & \left[\begin{array}{ll} \text{EXCONT} & \boxed{7} \\ \text{INCONT} & \boxed{5}\mathsf{book}'(y) \\ \text{PARTS} & \begin{array}{l} \langle\, y, \boxed{5}, \boxed{5a}\mathsf{book}', \boxed{7}\exists y\,[\alpha\,\&\,\beta], \\ \boxed{7a}\,[\alpha\,\&\,\beta]\,\rangle \end{array} \end{array}\right] \end{array}\right] \right\rangle \end{array}\right]$$

Ponieważ ponadto zachodzi `component` ($\boxed{5}$, α) oraz `component` ($\boxed{1}$, β), uzyskujemy formułę:

$$\exists y\,[...\mathsf{book}(y)...\,\&\,...\mathsf{read}'(x)(y)...].$$

Jako że nie jest to wyrażenie skompletowane, reguła EContP.2 nie ma tu zastosowania i jest to ostateczna postać $\boxed{9}$. Jak należało się spodziewać, zmienna x pozostaje wolna.

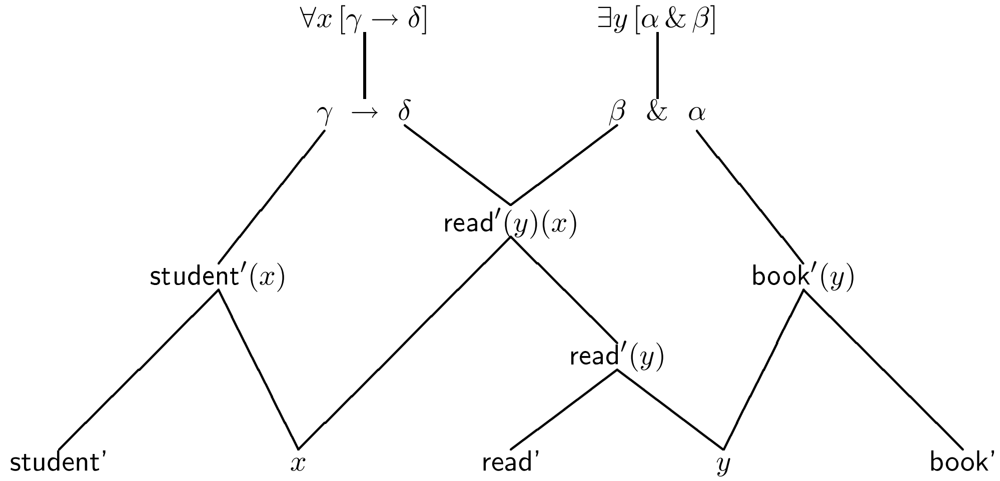
Zasada QNVSP zabezpiecza także poprawną interpretację zdań zawierających dwie (lub więcej, gdybyśmy dysponowali czasownikami z dopełnieniem bliższym i dalszym) kwantyfikowane frazy rzeczownikowe. Co więcej, jest ona tak sformułowana, by zapewnić swobodną interpretację zakresu kwantyfikatorów.

Przykład *4.6.* Weźmy zdanie z dwiema kwant. frazami rzeczownikowymi *Every student reads a book.*.

$$\begin{bmatrix} \text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{1} \\ \text{PARTS} & \langle x, y, \boxed{1}, \boxed{1a}, \boxed{1b}, \boxed{5}, \boxed{5a}, \boxed{6}, \boxed{6a}, \boxed{7}, \boxed{7a}, \boxed{8}, \boxed{8a} \rangle \end{bmatrix} \\ \text{HD-DTR}|\text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{1}\mathsf{read}'(y)(x) \\ \text{PARTS} & \begin{array}{l} \langle\, x, y, \boxed{1}, \boxed{1a}\mathsf{read}'(y), \boxed{1b}\mathsf{read}', \boxed{5}\mathsf{book}'(y), \\ \boxed{5a}\mathsf{book}', \boxed{7}\exists y\,[\alpha\,\&\,\beta], \boxed{7a}\,[\alpha\,\&\,\beta]\,\rangle \end{array} \end{bmatrix} \\ \text{NHD-DTRS} & \left\langle \begin{bmatrix} \text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{8} \\ \text{INCONT} & \boxed{6}\mathsf{student}'(x) \\ \text{PARTS} & \begin{array}{l} \langle\, x, \boxed{6}, \boxed{6a}\mathsf{student}', \\ \boxed{8}\forall x\,[\gamma \to \delta], \boxed{8a}\,[\gamma \to \delta]\,\rangle \end{array} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

Jako informacją wejściową można posłużyć się interpretacją frazy *every student* (por. przykł. 4.3 i 4.4) oraz *reads a book* z przykładu 4.5. I tak część semantyczna reprezentującej je struktury spełniająca wszystkie zasady jest ukazana po lewej. Na liście PART skompletowanej frazy mamy x, y, $\mathsf{read}'(y)(x)$, $\mathsf{read}'(y)$, read', $\mathsf{student}'(x)$, $\mathsf{student}'$, $\mathsf{book}'(y)$, book', $\exists y\,[\alpha\,\&\,\beta], [\alpha\,\&\,\beta], \forall x\,[\gamma \to \delta], [\gamma \to \delta]$.

Tworzą one następującą hierarchię podwyrażeń:



Niech $\beta = \mathsf{read}'(y)(x)$. Wówczas $\beta \lhd \delta$. Ponieważ ani β, ani δ nie należą do PART, *de facto* dysponujemy zależnością $[\alpha\,\&\,\beta] \lhd [\gamma \to \delta]$. Jako że także kwantyfikatory są związane z tymi formułami „na sztywno” (tzn. nie występują w innych konfiguracjach), uzyskujemy $\beta \lhd [\alpha\,\&\,\beta] \lhd \exists y\,[\alpha\,\&\,\beta] \lhd \delta \lhd [\gamma \to \delta] \lhd \forall x\,[\gamma \to \delta]$. Powstaje pytanie, czy δ może zawierać jakieś wyrażenie poza $\exists y\,[\alpha\,\&\,\beta]$. Ponieważ wyrażenie to musiałoby być związane z poprzednim jakimś spójnikiem logicznym, w grę wchodzą jedynie obie kwantyfikowane formuły z listy PART, co prowadziłoby do absurdalnej sytuacji, w której pewne wyrażenie byłoby własnym podwyrażeniem. Tak więc $\delta = \exists y\,[\alpha\,\&\,\beta]$. Podobnie można pokazać, że $\alpha = \mathsf{book}'(y)$, a $\gamma = \mathsf{read}'(y)(x)$. Wówczas uzyskujemy etykietowaną przez $\boxed{9}$ formułę $\forall x\,(\mathsf{student}'(x) \to \exists y\,[\mathsf{book}'(y)\,\&\,\mathsf{read}'(y)(x)])$.

Jeśli z kolei $\delta = \mathsf{read}'(y)(x)$, to $\delta \lhd \beta$, skąd $\delta \lhd [\gamma \to \delta] \lhd \forall x\,[\gamma \to \delta] \lhd \beta \lhd [\alpha\,\&\,\beta] \lhd \exists y\,[\alpha\,\&\,\beta]$. Przeto podobnie jak poprzednio można wykazać, że jedyną formuła etykietowaną przez $\boxed{9}$ spełniającą ten warunek jest $\exists y\,[\mathsf{book}'(y)\,\&\,\forall x\,(\mathsf{student}'(x) \to \mathsf{read}'(y)(x)])$. Z powyższego wynika także, że nie jest możliwe $\beta = \delta = \mathsf{read}'(y)(x)$. Także przypuszczenie, że każda z nich zawiera jakieś dodatkowe podywrażenie, prowadzi podobnie jak poprzednio do sprzeczności.

Ostatecznie dwie wyszczególnione formuły są jedynymi możliwymi interpretacjami zdania *Every student reads a book.*.

Zasada „nazwa własna-czasownik”

Ta zasada pełni podobną rolę do poprzedniej ((QNVP)), tylko tym razem do nadrzędnika (frazy czasownikowej) przyłączana jest nazwa własna.

Zasada 4.8. Zasada przyłączenia nazwy własnej do frazy czasownikowej (ang. *Proper Name–Verb Phrase Principle*, (NPVP)) opisywana jest przez następującą zależność:

$$\mathbf{E}\,\boxed{3}\left(\left(\begin{bmatrix} \textit{phrase} \\ \text{HD-DTR}|\text{LF}|\text{PARTS}\ \boxed{1} \\ \text{NHD-DTRS}\ \boxed{2} \end{bmatrix} \textbf{ and } \texttt{member}(\boxed{3},\boxed{2}) \textbf{ and } \boxed{3}\begin{bmatrix} \textit{phrase} \\ \text{SYNS}|\text{LOC}|\text{CAT}|\text{HEAD}\ \textit{noun} \\ \text{LF}\ \begin{bmatrix} \text{EXCONT}\ \boxed{4} \\ \text{INCONT}\ \boxed{4} \end{bmatrix} \end{bmatrix}\right) \Longrightarrow \texttt{member}(\boxed{4},\boxed{1})\right)$$

Zasada ta wymaga po pierwsze, by wartości atrybutów EXCONT i INCONT elemetu pobocznego były równe. Ponieważ kwantyfikowane frazy rzeczownikowe nie spełniają tego wymagania, zasada ta ewidentnie ich nie dotyczy. Przede wszystkim jednak narzuca ona, by zmienna reprezentująca nazwę własną znajdowała się na liście PART elementu głównego.

Przykład *4.7.* Weźmy zdanie *John walks.*. Struktury reprezentująca jego semantykę jest następująca.

$$\begin{bmatrix} \text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{2} \\ \text{PARTS} & \boxed{10} \end{bmatrix} \\ \text{HD-DTR}|\text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{2}\mathsf{walk}'(j) \\ \text{PARTS} & \boxed{10}\,\langle j, \boxed{2}, \boxed{2a}\mathsf{walk}' \rangle \end{bmatrix} \\ \text{NHD-DTRS} & \left\langle \begin{bmatrix} \text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{3} \\ \text{INCONT} & \boxed{3}j \\ \text{PARTS} & \langle \boxed{3} \rangle \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

Po pierwsze, zasada EContP.1 narzuca, by wartość atrybutu EXCONT elementu pobocznego pochodziła z jego listy PARTS, a ta jest w tym wypadku jednolelementowa. A przede wszystkim, zasada PMVP wymaga, by wartość parametru EXCONT elemetnu pobocznego znajdowała się na liście PART elementu głównego, co w rezultacie wymusza równość zmiennych x i j (gdyż oczywiście równe obiekty muszą być tego samego typu i rodzaju). Ewidentnym efektem tych wymagań jest formuła $\mathsf{walk}'(j)$.

Podobnie jak w przypadku zasady QNVP, także powyższą zasadę można stosować dwukrotnie. Efektem tego będzie interpretacja zdania *John reads Ulisses.* za pomocą formuły $\mathsf{reads}'(u)(j)$. Do pewnych zdań obie te zasady mogą mieć zastosowanie. Zdanie *Every student reads Ulisses.* nie powinno sprawiać trudności. Rozważymy odrobinę bardziej skomplikowany przypadek.

Przykład *4.8.* Weźmy zdanie wykorzystujące obie reguły QNVP oraz PNVP *John reads a book.*.

$$\begin{bmatrix} \text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{2} \\ \text{PARTS} & \boxed{10} \end{bmatrix} \\ \text{HD-DTR}|\text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{2}\mathsf{read}'(j)(x) \\ \text{PARTS} & \boxed{10} \begin{matrix} \langle\, j, y, \boxed{1}\mathsf{read}'(y)(j), \boxed{1a}\mathsf{read}'(y), \\ \boxed{1b}\mathsf{read}', \boxed{5}\mathsf{book}'(y), \boxed{5a}\mathsf{book}', \\ \boxed{7}\exists y\,[\alpha \,\&\, \beta], \boxed{7a}\,[\alpha \,\&\, \beta]\,\rangle \end{matrix} \end{bmatrix} \\ \text{NHD-DTRS} & \left\langle \begin{bmatrix} \text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{3} \\ \text{INCONT} & \boxed{3}j \\ \text{PARTS} & \langle \boxed{3} \rangle \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

Będziemy je reprezentować za pomocą następującej struktury. Podobnie jak poprzednio, reguła PNVP prowadzi do unifikacji zmiennych x i j, dysponujemy więc formułą $\exists y\,[...\mathsf{book}(y)...\,\&\,...\mathsf{read}'(j)(y)...]$ (zgodnie z interpretacją frazy *reads a book* z przykładu 4.5). Ponieważ tym razem jest to interpretacja całego zdania, reguła SP.2 wymusza ostateczną postać formuły:

$$\exists y\,[\mathsf{book}(y) \,\&\, \mathsf{read}'(j)(y)].$$

Zasada negacji

Richter i Sailer nie omawiają prostych przypadków negacji, przynajmniej w dostępnych nam pracach, omawiają jedynie złożone reguły negacjiw j. francuskim [Richter i Sailer, 1999b] i polskim [Richter i Sailer, 1999c, 2001, 2003]. Na tej podstawie pokusiliśmy się w niniejszy opracowaniu sformułować najprostszą wersję negacji dla j. angielskiego.

Zasada 4.9. Zasada negacji (ang. *Negation Principle*, (NP)) opisywana jest przez następującą zależność:

$$\begin{bmatrix} \textit{phrase} & \\ \text{HD-DTR} & \begin{bmatrix} \text{SYNSEM}|\text{LOCAL}|\text{CAT}|\text{HEAD} & \textit{verb} \\ \text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{1} \\ \text{INCONT} & \boxed{2} \end{bmatrix} \end{bmatrix} \\ \text{NHD-DTRS} & \left\langle \begin{bmatrix} \text{SYNSEM}|\text{LOCAL}|\text{CONT}|\text{MAIN} & \neg \\ \text{LF}|\text{INCONT} & \boxed{3}\ \neg\xi \end{bmatrix} \right\rangle \end{bmatrix} \Longrightarrow \texttt{component}(\boxed{2}, \xi)\ \textbf{and}\ \texttt{component}(\boxed{3}, \boxed{1})$$

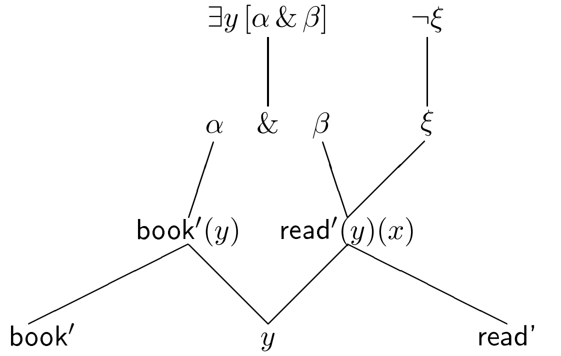
Przykład *4.9.* Rozważmy proste zanegowane zdanie *John doesn't walk.*. Musimy zacząć od rozpatrzenia frazy *doesn't walk*, dla której formowana jest struktura widoczna poniżej po lewej. Ponieważ ponadto $\texttt{component}(\boxed{2}, \xi)$ oraz $\texttt{component}(\boxed{10}, \boxed{9})$, uzyskujemy formułę postaci $...\neg[...\mathsf{walk}'(x)...]...$

$$\begin{bmatrix} \text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{2} \\ \text{PARTS} & \langle x, \boxed{2}, \boxed{2a}, \boxed{10} \rangle \end{bmatrix} \\ \text{HD-DTR|LF} & \begin{bmatrix} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{2}\mathsf{walk}'(x) \\ \text{PARTS} & \langle x, \boxed{2}, \boxed{2a}\mathsf{walk}' \rangle \end{bmatrix} \\ \text{NHD-DTRS} & \left\langle \begin{bmatrix} \text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{10} \\ \text{INCONT} & \boxed{10}\neg\xi \\ \text{PARTS} & \langle \boxed{10}\neg\xi \rangle \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \qquad \begin{bmatrix} \text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{2} \\ \text{PARTS} & \langle j, \boxed{2}, \boxed{2a}, \boxed{10} \rangle \end{bmatrix} \\ \text{HD-DTR|LF} & \begin{bmatrix} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{2}\mathsf{walk}'(j) \\ \text{PARTS} & \langle j, \boxed{2}, \boxed{2a}\mathsf{walk}', \boxed{10}\neg\xi \rangle \end{bmatrix} \\ \text{NHD-DTRS} & \left\langle \begin{bmatrix} \text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{3} \\ \text{INCONT} & \boxed{3}j \\ \text{PARTS} & \langle \boxed{3} \rangle \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

Po dołączeniu do powyższej frazy nazwy własnej *John* uzyskujemy strukturę widoczną powyżej po prawej, dla której rozważona już struktura stanowi element główny. Tak jak i poprzednio, dodanie nazwy własnej powoduje wyłącznie unifikację zmiennych. Na koniec ograniczenie EContP dla skompletowanego zdania „usuwa" nieistniejące fragmenty, co daje nam ostateczną postać interpretacji semantycznej zdania $\neg[\mathsf{walk}'(j)]$.

Przykład *4.10.* Weźmy kolejne zanegowane zdanie *John doesn't read a book.*. Zaczniemy od interpretacji frazy *doesn't read a book*, posługując się obiektami rodzaju *lrs* uzyskanymi dla fraz *a book* (por. przykł. 4.3 i 4.5 oraz *doesn't read* (por. przykł. 4.9), czego wynikiem jest struktura widoczna poniżej po lewej.

$$\begin{bmatrix} \text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{1} \\ \text{PARTS} & \langle x, y, \boxed{1}, \boxed{1a}, \boxed{1b}, \boxed{5}, \boxed{5a}, \boxed{7}, \boxed{7a}, \boxed{10} \rangle \end{bmatrix} \\ \text{HD-DTR|LF} & \begin{bmatrix} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{1}\mathsf{read}'(y)(x) \\ \text{PARTS} & \langle x, y, \boxed{1}, \boxed{1a}\mathsf{read}'(y), \boxed{1b}\mathsf{read}', \boxed{10}\neg\xi \rangle \end{bmatrix} \\ \text{NHD-DTRS} & \left\langle \begin{bmatrix} \text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{7} \\ \text{INCONT} & \boxed{5}\mathsf{book}'(y) \\ \text{PARTS} & \begin{matrix} \langle y, \boxed{5}, \boxed{5a}\mathsf{book}', \\ \boxed{7}\exists y\,[\alpha \,\&\, \beta], \boxed{7a}\,[\alpha \,\&\, \beta] \rangle \end{matrix} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$



Mamy ponadto zależności $\texttt{component}(\boxed{5}, \alpha)$, $\texttt{component}(\boxed{1}, \beta)$, $\texttt{component}(\boxed{1}, \xi)$ oraz $\texttt{component}(\boxed{10}, \boxed{9})$, przedstawionymi w sposób graficzny powyżej po prawej. Dołączenie podmiotu *John* daje w rezultacie widniejącą obok strukturę; jedynym tego efektem jest unikfikacja x z j. Teraz jednak dysponujemy już skompletowanym zdaniem, co oznacza, że posiadamy »wszystkie klocki do układanki« jaką jest finalna postać formuły. Jak widać na rysunku, nieustalona jest jedynie zależność między β a ξ. Przyjęcie $\xi \lhd \beta$ prowadzi do $\xi \lhd \neg\xi \lhd \beta \lhd \exists y\,[\alpha \,\&\, \beta]$, co daje formułę $\exists y\,[\mathsf{book}'(y) \,\&\, \neg\mathsf{read}'(y)(j)]$. Z kolei wybór $\beta \lhd \xi$ daje porządek $\beta \lhd \exists y\,[\alpha \,\&\, \beta] \lhd \xi \lhd \neg\xi$, skąd uzyskujemy formułę $\neg\exists y\,[\mathsf{book}'(y) \,\&\, \mathsf{read}'(y)(j)] \;\equiv\; \forall y\,[\mathsf{book}'(y) \;\rightarrow\; \neg\mathsf{read}'(y)(j)]$. Druga z tych formuł pasuje raczej jako interpretacja zdania *John doesn't read any book.*, ale brak interpretacji wyrazu *any* nie pozwala nam na żadne porównania.

$$\begin{bmatrix} \text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{2} \\ \text{PARTS} & \boxed{11} \end{bmatrix} \\ \text{HD-DTR|LF} & \begin{bmatrix} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{1}\mathsf{read}'(y)(x) \\ \text{PARTS} & \boxed{11}\; \begin{matrix} \langle x, y, \boxed{1}, \boxed{1a}\mathsf{read}'(y), \boxed{1b}\mathsf{read}', \\ \boxed{5}\mathsf{book}'(y), \boxed{5a}\mathsf{book}', \boxed{10}\neg\xi, \\ \boxed{7}\exists y\,[\alpha \,\&\, \beta], \boxed{7a}\,[\alpha \,\&\, \beta] \rangle \end{matrix} \end{bmatrix} \\ \text{NHD-DTRS} & \left\langle \begin{bmatrix} \text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{3} \\ \text{INCONT} & \boxed{3}j \\ \text{PARTS} & \langle \boxed{3} \rangle \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

Przykład *4.11.* Pozważmy podobne zdanie *A student doesn't walk.*. W celu jego interpretacji posłużymy się strukturami uzyskanymi dla fraz *a student* (por. przykł. 4.3 i 4.5) oraz *doesn't walk* z przykładu 4.9.

$$\left[\begin{array}{ll} \text{LF} & \left[\begin{array}{ll} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{2} \\ \text{PARTS} & \langle x, \boxed{2}, \boxed{2a}, \boxed{6}, \boxed{6a}, \boxed{7}, \boxed{7a}, \boxed{10} \rangle \end{array}\right] \\ \text{HD-DTR}|\text{LF} & \left[\begin{array}{ll} \text{EXCONT} & \boxed{9} \\ \text{INCONT} & \boxed{2}\mathsf{walk}'(x) \\ \text{PARTS} & \langle x, \boxed{2}, \boxed{2a}\mathsf{walk}', \boxed{10}\neg\xi \rangle \end{array}\right] \\ \text{NHD-DTRS} & \left\langle \left[\begin{array}{ll} \text{LF} & \left[\begin{array}{ll} \text{EXCONT} & \boxed{7} \\ \text{INCONT} & \boxed{6}\mathsf{student}'(y) \\ \text{PARTS} & \begin{array}{l} \langle y, \boxed{6}, \boxed{6a}\mathsf{student}', \\ \boxed{7}\exists y\,[\alpha \,\&\, \beta], \boxed{7a}\,[\alpha \,\&\, \beta] \,\rangle \end{array} \end{array}\right] \end{array}\right] \right\rangle \end{array}\right]$$

Posiadamy też zależności $\mathtt{component}(\boxed{6}, \alpha)$, $\mathtt{component}(\boxed{2}, \beta)$, $\mathtt{component}(\boxed{2}, \xi)$ oraz $\mathtt{component}(\boxed{10}, \boxed{9})$, co daje nam podobną hierarchię jak w poprzednim przykładzie. W analogiczny więc sposób uzyskujemy także i ostateczną postać formuł:

$$\exists y\,[\mathsf{student}'(y) \,\&\, \neg\mathsf{walk}'(y)] \quad \text{oraz}$$
$$\neg\exists y\,[\mathsf{student}'(y) \,\&\, \mathsf{walk}'(y)] \;\equiv\; \forall y\,[\mathsf{student}'(y) \rightarrow \neg\mathsf{walk}'(y)].$$

Są to dokładnie te same interpretacje co dla analogicznego zdania z przykładu 2.16 ze strony 22 w gramatyce PTQ Montague.

Pojawia się więc ten sam dylemat: zdania *A student doesn't walk.* oraz *Every student doesn't walk.* posiadają dokładnie te same interpretacje!

Podobnie zdanie *Every student doesn't read a book.* analogiczne do zdania przetwarzanego w ramach DRT w przykładzie 3.36 ze strony 71 będzie miało następujące interpretacje:

1. $\forall x\,(\mathsf{student}'(x) \;\rightarrow\; \exists y\,[\mathsf{book}'(y) \,\&\, \neg\mathsf{read}'(y)(x)]),$
2. $\forall x\,\forall y\,[\mathsf{student}'(x) \,\&\, \mathsf{book}'(y) \;\rightarrow\; \neg\mathsf{read}'(y)(x)],$
3. $\exists x\,(\mathsf{student}'(x) \,\&\, \forall y\,[\mathsf{book}'(y) \rightarrow \neg\mathsf{read}'(y)(x)]),$
4. $\exists y\,[\mathsf{book}'(y) \,\&\, \forall x\,(\mathsf{student}'(x) \rightarrow \neg\mathsf{read}'(y)(x)]),$
5. $\exists x\,\exists y\,[\mathsf{student}'(x) \,\&\, \mathsf{book}'(y) \,\&\, \neg\mathsf{read}'(y)(x)]$

Jedynie formuły 1 oraz 4 wydają się uprawnionymi interpretacjami tego zdania. Formuła 2 zdaje się pasować do zdania *Every student doesn't read any book.*, formuła 3 — do zdania *A student doesn't read any book.*, zaś formuła 5 — do zdania *A student doesn't read a book.*.

W przypadku LRS jednak można to bez trudu naprawić, wzmacniając zasadę NS tak, by ściśle wiązała negację z czasownikiem. Zauważmy, że w poniższej wersji reguły wymagamy, by negacja wiązała się z wyrazem, a nie frazą czasownikową, by uniknąć błędnej interpretacji zdań typu *John doesn't walk slowly.*, w których nie zamierzamy zaprzeczać samemu aktowi chodzenia.

Zasada 4.10. Poprawiona Zasada negacji (ang. *Corrected Negation Principle*, (CNP)) opisywana jest przez następującą zależność:

$$\left[\begin{array}{ll} \textit{phrase} & \\ \text{HD-DTR} & \left[\begin{array}{l} \textit{word} \\ \text{SYNSEM}|\text{LOCAL}|\text{CAT}|\text{HEAD}\ \textit{verb} \end{array}\right] \\ \text{NHD-DTRS} & \left\langle \left[\text{SYNSEM}|\text{LOCAL}|\text{CONT}|\text{MAIN}\ \neg\right] \right\rangle \end{array}\right] \Longrightarrow \left[\begin{array}{ll} \text{HD-DTR}|\text{LF} & \left[\begin{array}{ll} \text{EXCONT} & \boxed{1} \\ \text{INCONT} & \boxed{2}\xi \end{array}\right] \\ \text{NHD-DTRS} & \left\langle \left[\text{LF}|\text{INCONT}\ \boxed{3}\neg\xi\right] \right\rangle \end{array}\right] \textbf{ and } \mathtt{component}(\boxed{3}, \boxed{1}).$$

Jedyną różnicą w stosunku do poprzedniej wersji tej zasady jest „zrównanie” ξ z czasownikiem ($\mathsf{walk}'(x)$ lub $\mathsf{read}'(x)$ w rozważanych przykładach). Ogranicza to możliwe interpretacje zdania *John doesn't read a book.* z przykładu 4.10 do $\exists y\,[\mathsf{book}'(y) \,\&\, \neg\mathsf{read}'(y)(j)]$ oraz zdania *A student doesn't walk.* z przykładu 4.11 do $\exists y\,[\mathsf{student}'(y) \,\&\, \neg\mathsf{walk}'(y)]$.

5 Podsumowanie

W niniejszej pracy omówiliśmy trzy teoriomodelowe formalizmy semantyczne: gramatykę PTQ Montague, teorię reprezentacji dyskursu DRT Kampa i Reyle'go oraz semantykę opartą na zasobach leksykalny LRS Richtera i Sailera. Jednakże porównanie tych koncepcji jest niezmiernie trudne ze względu na krańcowo różne podejścia do tematu zaproponowane przez poszczególnych autorów.

Po pierwsze, całkowicie różnią się już same formalizmy, którymi posługują się autorzy w celu zapisu interpretacji sformułowań j. naturalnego. Język IL zaproponowany przez Montague jest to niezwykle wyrafinowany λ-rachunek z operatorami intensji i ekstensji oraz innymi operatorami modalnymi. Nawet sam autor jednak nie wykorzystuje do końca siły modelu tej logiki: istnieją operatory temporalne umożliwiające „przemieszczanie" się między światami wzdłuż osi czasu `w` (przy zmiennym `t`), jednak w żaden sposób nie można się przemieszczać się pomiędzy różnymi wariantami światów w danym momencie `t` (przy zmiennym `w`). Przez to przekonania agenta (believe) oraz poszukiwanie jednorożców nie jest ograniczone temporalnie w żadnym zakresie.

Z kolei wykorzystywany przez Richtera i Sailera język Ty2 wprowadzony przez Gallina jest klasycznym dwurodzajowym λ-rachunkiem będącym odpowiednikiem IL (prezentowana jest nawet translacja IL na Ty2), z wybraną zmienną $\tilde{v}_s$ oznaczającą świat bieżący. Formalizm ten zakodowany został w formacie RSRL służącym do formalnego zapisu gramatyki HPSG. Wyrażenia Ty2 zapisywane są jako obiekty rodzaju *lrs*, i format ten ma za zadanie ułatwiać interpretację j. naturalnego.

Na koniec DRT Kampa i Reyle'go jest całowicie niestandardowym formalizmem o mocy klasycznego języka pierwszego rzędu, z nietypowym sposobem definiowania (i zapisu) operatorów logicznych (działających na strukturach) oraz zastąpieniem zmiennych znacznikami dyskursu, których dostępność z różnych poziomów zagnieżdżenia struktur jest ważną częścią proponowanej metody.

Trzy rozwiązania, trzy odmienne formalizmy...

Po drugie, istotnie różnią się i gramatyki, którymi posługują się poszczególni autorzy jako podstawą do dalszego przetwarzania. Montague używa *gramatyki kategorialnej*, w której wyrażenie kategorii A/B złożone z wyrażeniem kategorii B daje wyrażenie kategorii A. Jest to podejście niezmiernie rygorystyczne. Uzyskane drzewa rozbioru są wyłącznie binarne, przy czym węzły etykietowane są nazwą funkcji realizującej dane przekształcenie, a nie typem umieszczonych w nich fraz. Wynika to z faktu, że tak naprawdę mamy do czynienia z wyrażeniem matematycznym złożonym z rekurencyjnie składanych funkcji. Takie ograniczenie powoduje, że co poniektóre frazy traktowane są nietypowo, a wyrazy pomocnicze w ogóle nie występują w słowniku.

Z kolei DRT jako takie nie narzuca ograniczeń na używaną gramatykę, choć rzecz jasna każda konkretna realizacja wiąże się z pewną gramatyką. Kamp i Reyle przedstawili prostą gramatykę transformacyjną w formacie GPSG. Jednak istnieją także przypadki użycia tego formalizmu wraz z gramatyką HPSG.

Richter i Sailer w końcu posługują się gramatyką HPSG. Co więcej, formalizm przez nich zaproponowany jest immanentną częścią HPSG i nie może zostać użyty wraz z jakąkolwiek inną gramatyką.

Trzy rozwiązania, trzy odmienne gramatyki...

Po trzecie wreszcie, także podejście do interpretacji semantycznej sformułowań j. naturalnego w omawianych rozwiązaniach jest istotnie różne. W PTQ Montague każdej regule syntaktycznej odpowiada reguła semantyczna służąca do rekurencyjnej konkatenacji interpretacji wyrażeń składowych, przy czym każdemu wyrazowi (wyrażeniu podstawowemu) odpowiada jednoznacznie określona translacja na IL. Oznacza to, że także każdemu rozbiorowi (niekoniecznie zdaniu) odpowiada dokładnie jedna translacja $\mathfrak{g}$ na IL. Ponadto proces analizy syntaktycznej poprzedza proces interpretacji semantycznej: funkcja translacji $\mathfrak{g}$ aplikowana jest do wyrażenia będącego efektem analizy syntaktycznej.

Kamp i Reyle interpretują nie pojedyncze zdanie, ale cały dyskurs. Do przekształceń semantycznych potrzebują (początkowo pustej) struktury DRT oraz ciągu rozbiorów kolejnych zdań. Analiza semantyczna polega na modyfikowaniu i usuwaniu (fragmentów) poszczególnych drzew i odpowiednim modyfikowaniu i rozszerzaniu struktury DRT. Żadne specjalne ograniczenia (poza syntaktyką DRT, a i to nie do końca, gdyż w strukturach umieszczane są przejściowo wyrażenia nie będące poprawnymi DRT-warunkami; w szczególności drzewa) nie są na ten proces nakładane. Pojedyncze drzewo może być przekształcane w różny sposób (w zależności od „aplikowalności” sterujących tym reguł), a to oznacza, że dla jednego rozbioru może istnieć więcej interpretacji (niejednoznaczność semantyczna). Wynika z tego, że translacja $\mathfrak{G}$ nie jest funkcją, tylko przekształceniem niedeterministycznym.

W przypadku LRS interpretacja semantyczna jest immanentną częścią procesu znajdowania struktur spełniających wszystkie ograniczenia (tak syntaktyczne jak i semantyczne) narzucone przez teorię. Ma to raczej charakter dowodu matematycznego niż translacji. Istotną cechą tego rozwiązania jest zawarcie w pojedynczym obiekcie rodzaju *lrs* wszystkich interpretacji semantycznych zdania dla danej reprezentacji syntaktycznej.

Trzy rozwiązania, trzy odmienne gramatyki... Jakie to szczęście, że chociaż język jest ten sam! To znaczy j. angielski, a i to nie do końca, bo chociaż wszystkie podejścia oparte zostały na bardzo prostym podzbiorze angielszczyzny, to jednak w żadnym wypadku nie są to równe podzbiory. Jednak jest to jedyny poziom który daje szansę jakichkolwiek porównań. Wspomniane różnice biorą się z odmienności celów, jakie stawiali sobie autorzy. Montague chciał pokazać, że w ogóle można reprezentować semantykę j. naturalnego w elegancki, systematyczny i rygorystyczny sposób. Kampowi i Reyle'mu zależało na reprezentacji wykraczającej poza pojedyncze zdanie, zaś Richter i Sailer pragnęli w gładki sposób włączyć semantykę bezpośrednio do sygnatury i teorii HPSG, uzyskując przy tym upakowaną reprezentację wypowiedzeń niejednoznacznych.

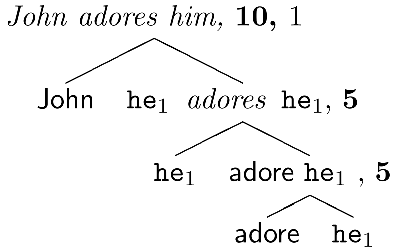
Proste zdania typu *John loves Mary.*, *John reads a book.* czy *John doesn't hate Bill.* są bez problemu interpretowane we wszystkich rozważanych podejściach, dając w rezultacie równoważne wyrażenia, niezależnie od różnic w metodzie użytej do tego celu. Montague reprezentuje nazwy własne jako zbiory własności. Niezależnie od wszelkich motywacji dla takiego rozwiązania kierujących samym autorem, podejście takie ma swoich zwolenników. Na przykład Landman (1986a) opisuje obiekty, dla których wzrost naszaj wiedzy na ich temat nie przeszkadza w ich dalszej identyfikacji, chyba że coś nam tę identyfikację zakłóci (jak w przypadku Wenus czy braci-bliźniaków). Do reprezentacji takich częściowych obiektów tworzy *kołki* (ang. *pegs*), będące zbiorami kolejno „zawieszanych” na nich własności. Jednak u Landmana kołki nie są obiektami, natomiast Montague wiąże własności z konkretnymi stałymi (np. stała j reprezentująca Johna), a ostateczna wersja formuły zawiera wyłącznie konkretną własność opisywaną w zdaniu.

Z kolei w DRT nazwy własne nie mają charakteru stałych indywiduowych lecz predykatów jednoargumentowych, których argumentami, tak jak i pozostałych predykatów (np. rzeczowników pospolitych), są znaczniki dyskursu. W LRS sprawa zdaje się wyglądać podobnie, lecz jest to jedynie pozór, gdyż zmienne x i y mogą podlegać unikfikacji z innymi zmiennymi (w szczególności ze sobą nawzajem), a więc reprezentujący je atrybut VAR posiada jedynie etykietę. Wręcz przeciwnie, zmienne j i u są wartościami na sztywno przypisanymi temu atrybutowi, i nie mogą się ze sobą unifikować, a unifikacja ze zmienną x czy y polega na zastąpieniu tychże wartością j bądź u.

Kamp i Reyle szczególny nacisk kładą na wiązanie zaimków z odpowiadającymi im rzeczownikami, znajdującymi się w bieżącym zdaniu lub w którymś z poprzednich. Cała konstrukcja struktur DRS podporządkowana jest celowi, by proces ten przebiegał poprawnie. Tworzenie struktur ma charakter wielostopniowy i zarazem kumulacyjny. Pozornie nic nie stoi na przeszkodzie, by formuły klasyczne podlegały podobnym przekształceniom, lecz od strony technicznej byłoby to znacznie trudniejsze.

Montague dokonuje wiązania zaimków w specyficzny sposób. Mianowicie „wydobywa” ze zdania frazę rzeczownikową (podmiot bądź dopełnienie) i przemianowuje ją na „sztuczny” zaimek $\mathtt{he}_n$ wraz ze wszystkimi występującymi w zdaniu zaimkami, czego rezultatem jest znowu zdanie (a właściwie odbywa się to odwrotnie, gdyż proces analizy jest wstępujący). Reguła ta służy nie tylko do wiązania zaimków, ale i „przesuwania” poszczególnych fraz rzeczownikowych w zdaniu, co wykorzystywane jest także przy analizie zdań o niejednoznacznym zakresie.

Uniwersalność tej reguły ma jednak także swoje wady. Mianowicie Montague nie brał pod uwagę zdań, w których występują zaimki wiążące się z frazami rzeczownikowymi znajdującymi się poza zdaniem. Nie znaczy to jednak bynajmniej, że zaimki takie zostaną odrzucone. Na przykład zdanie *John adores him.* będzie potraktowane następująco:



$$\mathfrak{g}\ (\textit{John adores him}) \equiv$$
$$\lambda P\,[P\{j\}]\left({}^\frown\lambda x_n\,\mathfrak{g}[\mathtt{he}_1\ \textit{adores}\ \mathtt{he}_1]\right) \equiv$$
$$\lambda x_n\left[\lambda Q\,[Q\{x_n\}]\,{}^\frown(\mathsf{adore}'\,({}^\frown(\lambda Q\,[Q\{x_n\}])))\right](j) \equiv$$
$$\lambda x_n\,[\mathsf{adores}'_\star(x_n,x_n)] \equiv \mathsf{adores}'_\star(j,j)$$

co jest raczej interpretacją dla zdania *John adores himself.*. Co dziwniejsze, niepoprawnie traktowane są zdania z konstrukcją względną w których zaimki odnoszą się zarówno do podmiotu, jak i dopełnienia, gdyż każde zastosowanie dowolnej z reguł wiążących zaimki »pożera« je wszystkie od razu. I tak zdanie *John knows a man such taht he hates him.* posiada dwie interpretacje: $\exists x\,[\mathsf{man}'(x)\,\&\,\mathsf{know}'(x)(j)\,\&\,\mathsf{hates}'(j,j)]$ oraz $\exists x\,[\mathsf{man}'(x)\,\&\,\mathsf{know}'(x)(j)\,\&\,\mathsf{hates}'(x,x)]$, i żadna z nich nie jest akceptowalna.

W dostępnych pracach na temat LRS nie wspomina się o zaimkach. Jednak teoria wiązania zaimków (w ramach pojedynczego zdania) jest ważnym nurtem prac w dziedzinie HPSG; istnieją opracowania na ten temat dotyczące j. polskiego [Marciniak, 1999; 2001; Przepiórkowski i in., 2003]. Rozbudowując LRS o tę dziedzinę należy jedynie w sposób umiejętny włączyć do atrybutu LF informację znajdującą się już w strukturze.

W pracach dotyczących LRS stanowiących źródło dla niniejszego raportu nie wspomniano też nic ani o frazach skoordynowanych za pomocą zaimków, ani o konstrukcjach względnych. Można jednak sobie wyobrazić, że efekt będzie przypominał bardziej PTQ niż DRT (z dokładnością do sposobu traktowania zaimków) ze względu na rodzaj użytego formalizmu.

Różnica pomiędzy traktowaniem konstrukcji względnych w PTQ i w DRT ma charakter raczej syntaktyczny: Montague omawia konstrukcję *such that* której składnikiem jest zwykłe zdanie, zaś Kamp i Reyle prezentują konstrukcje typu *who* i *which* zawierające zdania z luką. Jednak interpretacja tych konstrukcji opiera się na interpretacji zdań zależnych jako takich (w sposób charakterystyczny dla danego formalizmu). Warto zauważyć, że Rodman (1986) rozszerzył PTQ o konstrukcje względne zawierające pozostałe zaimki względne.

Interpretacja fraz rzeczownikowych zawierających rodzajnik jest w PTQ i LRS w zasadzie identyczna, nietypowość DRT w tej kwestii kryje się w samej definicji języka logicznego. Otóż kwantyfikatory istnieją tam jedynie implicite: wskaźniki dyskursu głównej struktury kwantyfkowane są egzystencjalnie, negacja (standardowo) „odwraca" kwantyfikator, zaś w implikacji uniwersalnie kwantyfikowany jest następnik. Tak więc, w przeciwieństwie do pozostałych dwóch formalizmów, interpretacja rodzajników *a* i *every* nie jest symetryczna.

Na koniec przejdziemy do kwestii, do której autorzy rozmaitych prac na temat semantyki j. naturalnego przywiązują chyba największą uwagę, a mianowicie kwestii zakresu działania poszczególnych operatorów; głównie chodzi tu o kwantyfikatory i negację. Wszyscy autorzy zakładają akceptację pełnej dowolności porządku operatorów dla danego zdania (z tym zastrzeżeniem, że, przynajmniej w logice klasycznej, kolejność kwantyfikatorów uniwersalnych czy kwantyfikatorów egzystencjalnych nie ma znaczenia). Realizują ją jednak w całkowicie różny sposób.

W PTQ każdy rozbiór zdania ma dokładnie jedną interpretację, więc dla uzyskania kilku translacji niezbędne jest stworzenie kilku rozbiorów. Służą do tego wspomniane już reguły wiązania zaimków S14 i S16. W przypadku DRT nie są tworzone żadne „sztuczne", pomocnicze rozbiory; różnorodność interpretacji (dla pojedynczego drzewa rozbioru) uzyskiwana jest przez odmienną kolejność stosowania (aplikowalnych w danym momencie) reguł. Natomiast W LRS sam proces interpretacji semantycznej nie prowadzi do rozmnożenia struktur (choć analiza syntaktyczna może taki skutek spowodować); pojedynczy atrybut LRS zawiera w sobie niedospecyfikowany zestaw podformuł, które w połączeniu z dodatkowymi narzuconymi na nie warunkami (postaci `component` ($\boxed{a}$, $\boxed{b}$)) daje się uformować wyłącznie w formuły pożądanej postaci.

Efekty tych operacji są bardzo zbliżone. Także ich ograniczenia są zaskakująco podobne. Na przykład zdanie *A student doesn't walk.* ma w PTQ (por. przykł. 2.16) oraz w LRS (przykł. 4.11) dokładnie taką samą interpretację w postaci formuł $\exists y\, [\mathsf{student}'(y) \,\&\, \neg\mathsf{walk}'(y)]$ oraz $\forall y\, [\mathsf{student}'(y) \rightarrow \neg\mathsf{walk}'(y)]$, które co więcej są także interpretacją zdań *Every student doesn't walk.*!

W LRS bez trudu dało się ten zaskakujący efekt usunąć, związując negację bezpośrednio z czasownikiem. W PTQ nie ma mechanizmu umożliwiającego taką modyfikację.

W DRT zdanie *A student doesn't walk.* reprezentowane jest za pomocą dwóch DRT-struktur przedstawionych poniżej po lewej, zaś zdanie *Every student doesn't walk.* za pomocą dwóch struktur przedstawionych poniżej po prawej.

$$\boxed{\begin{array}{c} x \\ \textbf{student}\,(x) \\ \neg\ \boxed{\textbf{walk}\,(x)} \end{array}} \qquad \boxed{\boxed{\begin{array}{c} x \\ \textbf{student}\,(x) \end{array}} \Longrightarrow \neg\ \boxed{\textbf{walk}\,(x)}}$$

$$\boxed{\neg\ \boxed{\begin{array}{c} x \\ \textbf{student}\,(x) \\ \textbf{walk}\,(x) \end{array}}} \qquad \boxed{\neg\ \boxed{\boxed{\begin{array}{c} x \\ \textbf{student}\,(x) \end{array}} \Longrightarrow \boxed{\textbf{walk}\,(x)}}}$$

Niezależnie od różnic notacyjnych można bez trudu sprawdzić, że interpretacja rozważanych zdań w DRT jest równoważna dwum pozostałym. Na szczęście także w tym przypadku zablokowanie przekształcania poddrzewa zawierającego negację przed interpretacją drzew zawierających kwantyfikatory nie powinno być trudne.

W niniejszym raporcie dokonane zostało porównanie trzech formalizmów reprezentacji interpretacji semantycznej sformułówań języka naturalnego. Opracowanie to dalekie jest jednak od wyczerpania tej tematyki. Literatura na temat wyłącznie tych wybranych podejść obejmuje wiele pominiętych tu zjawisk lingwistycznych oraz opisów konkretnych zastosowań (zwłaszcza w przypadku DRT będącego obecnie chyba najbardziej popularnym rozwiązaniem). PTQ Montague była, a pozostałe dwa formalizmy wciąż są wielokrotnie rozszerzane i modyfikowane.

Nie wyczerpują też one bynajmniej listy logik stosowanych w zakresie semantyki języka naturalnego. Wymienić należy tu przede wszystkim *Logikę Dynamiczną* (ang. *Dynamic Logic*) [Groenendijk i Stokhof, 1991; Heim, 1982; Landman, 1986b; Veltman, 1991] oraz *Semantykę Sytuacyjną* (ang. *Situation Semantics*) [Barwise, 1981; Barwise i Perry, 1983; Cooper, 1993; Gawron i Peters, 1990]. Jednak duża różnorodność omówionych podejść pozwala mieć nadzieję, że niniejsze opracowanie daje choć wyobrażenie na czym omawiana dziedzina polega, jakie stawia przed sobą cele i jakich środków używa do ich realizacji.

Bibliografia

Ajdukiewicz, K. (1935) Die syntakische Konnexität, *Studia Philosophica* **1**, s. 1–27. Przetłumaczone jako: Syntactic connexion, w S. McCall (red.) (1967) *Polish Logic*, Clarendon Press, Oxford, s. 207–231

Allen, J.F. (1984) Towards a general theory of action and time, *Artificial Intelligence* **23***(2)*, s. 123–154

Artale, A. & Franconi, E. (1994) A computational approach for a description logic of time and action, *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning*, Bonn, Germany, Morgan Kaufmann Publishers, s. 3–14

Bach, E. (1968) Nouns and noun phrases w E. Bach i R.T. Harms (red.) *Universals of linguistic theory*, s. 91–122, Holt, Rinehart & Winston, New York

Barwise, J. (1981) Scenes and other situations, *The Journal of Philosophy* **78**, s. 369–397

Barwise, J. i Perry, J. (1983) *Sittuations and Attitudes*, MIT Press, Cambrridge, MA

Cooper, R. (1993) Generalized quantifiers and resource situations, w P. Aczel, D. Israele, Y. Katagiri i S. Peters (red.) *Situation theory and its applications*, vol. 3, CSLI Publications, Stanford, CA, s. 191–212

de Saussure, F. (1959) *Corse in general linguistics*, McGrow-Hill, New York

Dowty, D.R., Wall, R.E. & Peters, St. (1981) *Introduction to Montague semantics*, Kluwer Academic Publishers, Dordrecht

Gallin, D. (1975) *Intensional and higher-order modal logics*, North-Holland, Amsterdam

Galton, A. (1990) A critical examination of Allen's theory of action and time, *Artificial Intelligence* **42** (2-3), s. 159–188

Gawron, M. i Peters, S. (1990) Anaphora and quantification in Sitaution Semantics, CSLI Lecture Notes **19**, Stanford, CA,

Groenendijk, J. i Stokhof, M. (1991) Dynamic predicate logic, *Linguistics and Philosophy* **14**, s. 39–100

Grzegorczyk, A. *Zarys logiki matematycznej*, PWN, Warszawa

Heim, I. (1982) *The semantics of definite and indefinite noun phrases*, praca doktorska, University of Massachusests, Amherst

Kamp, H. (1981) *A theory of truth and semantic representation*, Formal methods in the study of language, red. J. Groenendijk i in., Mathematisch Centrum, Amsterdam

Kamp, H. & Reyle, U. (1991) *A calculus for first order discourse representation structures*, Arbeitspapiere des Sonderforschungsbereichs 340, nr. 16, IMS Stuttgart

Kamp, H. & Reyle, U. (1993) *From discoruse to logic*, Kluwer, Dordrecht

Kripke, S. (1959) A completeness theorem in modal logic, *Journal of Symbolic Logic* **24**, pp. 1–14

Kripke, S. (1963) Semantical analysis of modal logic I: normal modal propositional calculi, *Zeitschrift für Math. Logic und Grundlagen der Math.* **9**, pp. 67–96

Kripke, S. (1965) Semantical analysis of modal logic II: non normal modal propositional calculi, w: *The theory of models*, North-Holland Puplishing, Amsterdam, pp. 206–220

Landman, F. (1986a) Pegs and alecs, w J.Y. Halpern (red.) *Proceedings of the Fifth Conference on Theoretical Aspects of Reasoning about Knowledge*, Monterey, CA, Morgan Kaufmann Publishers, s. 45–61

Landman, F. (1986b) *Towards a theory of information*, praca doktorska, University of Amstrdam

Marciniak, M. (1999) Toward a binding theory in Polish, w R.D. Borsley i A. Przepiórkowski (red.) *Slavic in HPSG*, CSLI Publications, Stanford, CA, s. 125–147

Marciniak, M. (2001) *Algorytmy implementacyjne syntaktycznych reguł koreferncji zaimków dla języka polskiego w terminach HPSG*, praca doktorska, Instytut Podstaw Informatyki PAN

Marciszewski, W. (1987) (red.) *Logika formalna. Zarys encyklopedyczny z zastosowaniem do informatyki i lingwistyki*, Państwowe Wydawnictwo Naukowe, Warszawa

McDermott, D. (1982) A temporal logic for reasoning about processes and plans, *Cognitive Science* **6**, s. 101–155

Montague, R. (1970a) English as a formal language, w *Linguaggi nella società e nella tecnica*, red. B. Visentini i in., Edizioni di Communità, Milan; przedruk w Thomason, R. (red.) (1974) *Formal philosophy, selected papers of Richard Montague*, Yale University Press, New Haven

Montague, R. (1970b) Universal grammar, *Theoria* **36**; przedruk w Thomason, R. (red.) (1974) *Formal philosophy, selected papers of Richard Montague*, Yale University Press, New Haven

Montague, R. (1973) The proper treatment of quantification in ordinary English, w *Approaches to natural language: proceedings of the 1970 Stanford workshop of grammar and semantics*, red. J. Hintikka, J. Moravcsik & p. Suppes, Reidel, Dordrecht, str. 221–242; przedruk w Thomason, R. (red.) (1974) *Formal philosophy, selected papers of Richard Montague*, Yale University Press, New Haven

Pollard, C. i Sag, I.A. (1987) *Information-based syntax and semantics*, vol. I: *Fundamentals*, CSLI Publications, Stanford, CA

Pollard, C. i Sag, I.A. (1994) *Head-driven Phrased Structured Grammar*, Chicago University Press / CSLI Publications, Chicago, IL

Przepiórkowski, A., Kupść, A., Marciniak, M. i Mykowiecka, A. (2002) *Formalny opis języka polskiego. Teoria i implementacja.* Akademicka Oficyna Wydawnicza Exit, Warszawa

Richter, F. (1999) „RSRL for HPSG", w V. Kordoni (red.) *Tübingen Studies in Head–Driven Phrase Structure Grammar*, Arbaitspapiere des SFB 340, Nr. 132, Vol. 1, Universität Tübingen, s. 74–115

Richter, F. (2000) *A mathematical formalism for linguistic theories with an application in Head-driven Phrase Structure Grammar*, praca doktorska, Universität Tübingen

Richter, F. i Sailer, M. (1999a) Underspecified semantics in HPSG w H. Bunt i R. Muskens (red.) *Computing Meaning*, Studies in Linguistics an Philosophy, Kluver Academic Publishers, s. 95–112

Richter, F. i Sailer, M. (1999b) A lexicalist collocation analysis of sentential negation and negative concord in French, w V. Kordoni (red.) *Tübingen Studies in Head–Driven Phrase Structure Grammar*, Arbaitspapiere des SFB 340, Nr. 132, Volume 1, Universität Tübingen, s. 231–300

Richter, F. i Sailer, M. (1999c) LF conditions on expressions of Ty2: an HPSG analysis of negative concord in Polish, w R.D. Borsley i A. Przepiórkowski (red.) *Slavic in HPSG*, CSLI Publications, Stanford, CA, s. 247–282

Richter, F., Sailer, M. (2001a) On the left periphery of german finite sentences, w W.D. Meuerers i T. Kiss (red.) *Constraint-based approaches to german syntax*, CSLI Publications, Stanford, CA, s. 257–300

Richter, F., Sailer, M. (2001b) Polish negation and Lexical Resource Semantics, w G.-J. Kruijff, L.S. Moss i R.T. Oehrle (red.), *Proceedings FG-MOL 2001* nr. 53 w Electronic Notes in Theorietical Computer Science (w druku)

Richter, F., Sailer, M. (2003) Basic concepts of Lexical Resource Semantics (w przygotowaniu)

Richter, F., Sailer, M. i Penn, G. (1999) A formal interpretation of relations and quantification in HPSG, w G. Bouma i in. (red.) *Constraints and resources in natural language syntax and semantics*, CSLI Publications, Stanford, CA, s. 281–298

Rosenbaum, P. (1967) *The grammar of english predicate compliment constructions*, MIT Press, Cambridge, MA

Sailer, M. (2000) *Combinatorial semantics and idiomatic expressions in Head-Driven Phrase Structure Grammar*, praca doktorska, Eberhard-Karls-Universität Tübingen, Seminar für Sprachwissenschaft

Shoham, Y. (1987) *Reasoning about change: time and causation from the standpoint of artificial intelligence*, praca doktorska, Yale University

Szałas, A. (1992) *Zarys dedukcyjnych metod automatycznego wnioskowania*, Akademicka Oficyna Wydawnicza RM, Warszawa

van Benthem, J.F.A.K. (1983a) *The logic of time*, D. Reidel Publishing Company, Dordrecht, Holland

van Benthem, J.F.A.K. (1983b) *Modal logic and classical logic*, Bibliopolis

van Benthem, J.F.A.K. (1984) Correspondence theory, w D. Gabbay, F. Guenthner (red.) *Handbook of philosophical logic*, vol. 2, s. 167–247

van Eijck, J. & Kamp, H. (1997) Representing discourse in context, w *Handbook of logic and language*, red. J. van Benthem & A. ter Meulen, Elsevier Science

Veltman, F. (1991) *Defaults in update semantics*, Technical report, Department of Philosophy, University of Amsterdam

Pracę zgłosił: Leonard Bolc

Adres autorki: Elżbieta Hajnicz
Instytut Podstaw Informatyki Polskiej Akademii Nauk
Ordona 21 01-237 Warszawa
e-mail: Elzbieta.Hajnicz@ipipan.waw.pl

Symbole klasyfikacji rzeczowej: CR: I.2.7

Nakład 140 egzemplarzy. Papier kserograficzny klasy III. Oddano do druku w grudniu 2003. Wydawnictwo IPI PAN.
ISSN 0138-06-48.