Leksykon gramatyki kategorialnej dla języka polskiego

Paweł Marczewski
MIM UW

pm262952@students.mimuw.edu.pl

1 października 2012

Cel pracy

Celem jest konwersja polskiego banku drzew na format wywodów CCG.
Potrzebne będą:

- formalizm dostosowany do polskiego szyku swobodnego
- warstwa semantyczna (kodująca zależn ości składniowe)
- algorytm konwersji drzew Świgry
- ewaluacja (cross-validation na zależnościach)

Plan

Plan

Combinatory Categorial Grammar

- Leksykon: „typy proste” + informacje semantyczne
- Łączymy elementy zdania tworząc wywód
- Kilka prostych reguł, opartych na rachunku kombinatorów
- CCG pozwala na parsowanie bardziej „dowolnych” części tekstu niż CFG

Leksykon

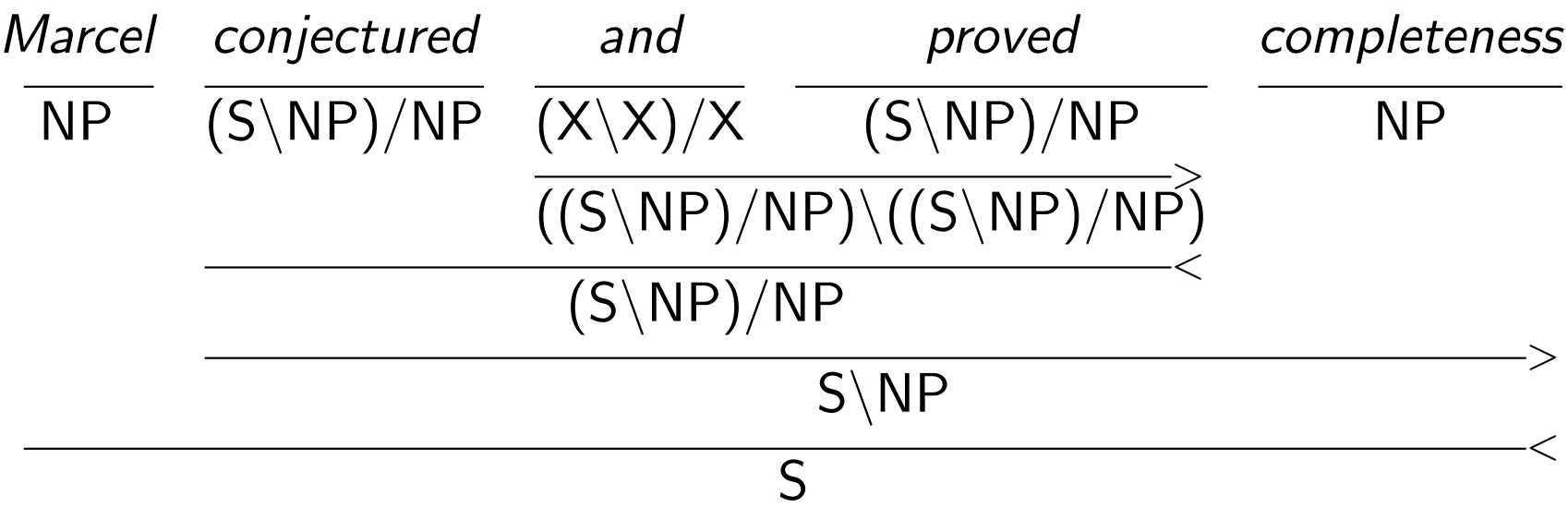
- Każde słowo ma przypisaną kategorię:
  - kategorie atomowe: S, N, PP...
  - kategorie złożone: X/Y, X\Y
    (prawa część to argument, lewa to wynik)
- Na przykład:
  - *dog* := NP
  - *big* := NP/NP
  - *very* := (NP/NP)/(NP/NP)
- Informacje o języku zawarte są w leksykonie, nie w regułach

Aplikacja

$$(>) \quad X/Y, \; Y \vdash X$$

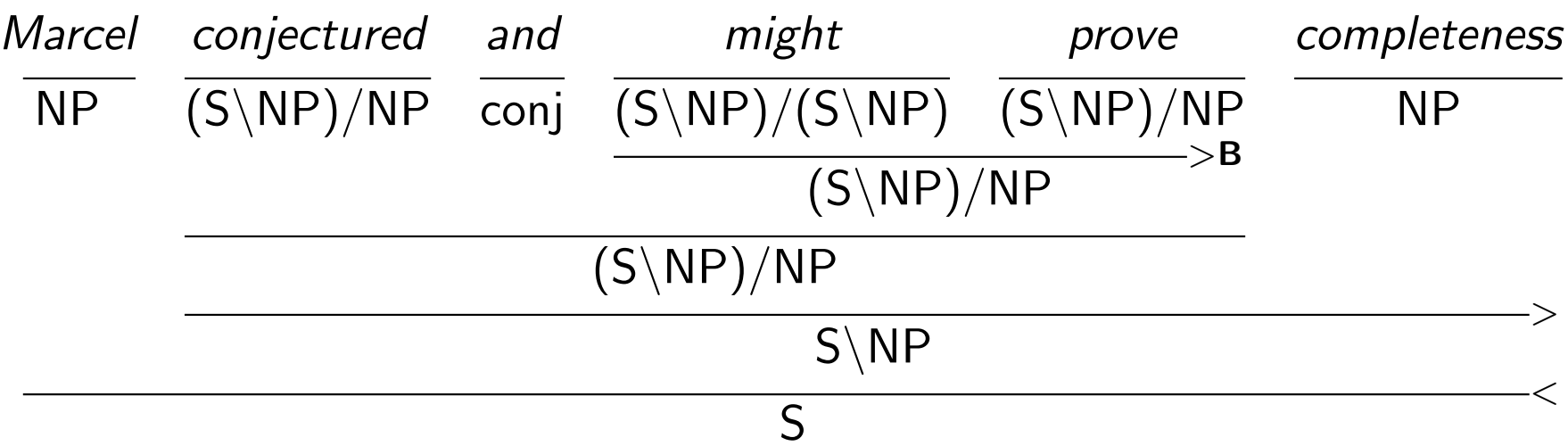
$$\dfrac{\dfrac{\textit{Marcel}}{\mathrm{NP}} \quad \dfrac{\dfrac{\textit{proved}}{(\mathrm{S}\backslash\mathrm{NP})/\mathrm{NP}} \quad \dfrac{\textit{completeness}}{\mathrm{NP}}}{\mathrm{S}\backslash\mathrm{NP}}>}{\mathrm{S}}<$$

Przykład koordynacji



Kompozycja

$$(> B) \quad X/Y,\ Y/Z \vdash X/Z$$



Podnoszenie typu

$$(>T) \quad X \vdash Y/(Y\backslash X)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\textit{Marcel}}{\mathrm{NP}}}{\mathrm{S}/(\mathrm{S}\backslash\mathrm{NP})}>\mathbf{T}}{\mathrm{S}/\mathrm{NP}}>\mathbf{B} \quad \dfrac{\textit{and}}{\mathrm{conj}} \quad \dfrac{\dfrac{\dfrac{\textit{I}}{\mathrm{NP}}}{\mathrm{S}/(\mathrm{S}\backslash\mathrm{NP})}>\mathbf{T}}{\mathrm{S}/\mathrm{NP}}>\mathbf{B}}{\mathrm{S}/\mathrm{NP}} \quad \dfrac{\dfrac{\textit{proved}}{(\mathrm{S}\backslash\mathrm{NP})/\mathrm{NP}} \quad \dfrac{\dfrac{\textit{disproved}}{(\mathrm{S}\backslash\mathrm{NP})/\mathrm{NP}} \quad \dfrac{\dfrac{\textit{completeness}}{\mathrm{NP}}}{}}{}}{}}{\mathrm{S}}>$$

Semantyka

Zazwyczaj termy rachunku lambda.

$$\dfrac{\dfrac{\textit{Marcel}}{\textbf{marcel} : \mathrm{NP}} \quad \dfrac{\dfrac{\textit{proved}}{\lambda x.\lambda y.\textbf{prove}(x, y) : (\mathrm{S}\backslash\mathrm{NP})/\mathrm{NP}} \quad \dfrac{\textit{completeness}}{\textbf{completeness} : \mathrm{NP}}}{\lambda y.\textbf{prove}(\textbf{completeness}, y) : \mathrm{S}\backslash\mathrm{NP}}>}{\textbf{prove}(\textbf{completeness}, \textbf{marcel}) : \mathrm{S}}<$$

Plan

Multizbiory

- Części zdania mogą się pojawić w dowolnej kolejności: *Marysia lubi Janka*, *lubi Marysia Janka*, *Janka lubi Marysia*, itd.
- Chcemy, żeby *lubi* miało zawsze ten sam typ
- Rozwiązanie: multizbiory
  - *Marysia* : NP, *Janka* : NP
  - *lubi* : $S\{|NP_{nom}, |NP_{acc}\}$
  - *biegnie* : $S\{|NP_{nom}\}$

Okoliczniki w zdaniu

- Modyfikatory zdania – wyrażenia przyimkowe, wtrącenia, przysłówki
- Chcemy, żeby ich typ był niezależny od modyfikowanego czasownika i pozycji
- Najlepiej: zawsze S|S
- Rozwiązanie: wprowadzamy kompozycję

Kompozycja

- Dopuszczamy kompozycję dla specjalnych kategorii: $X /\!\!/ Y$, $X \backslash\!\!\backslash Y$, $X \| Y$
- Reguła:

$$(> B) \quad X \| Z, \; Z\{...\} \vdash X\{...\}$$

$$\dfrac{\dfrac{\mathit{Janek}}{\mathrm{NP}_{\mathrm{nom}}} \quad \dfrac{\dfrac{\dfrac{\mathit{je}}{\mathrm{S}\{|\mathrm{NP}_{\mathrm{nom}}, |\mathrm{NP}_{\mathrm{acc}}\}} \quad \dfrac{\mathit{szybko}}{\mathrm{S}\|\mathrm{S}}}{\mathrm{S}\{|\mathrm{NP}_{\mathrm{nom}}, |\mathrm{NP}_{\mathrm{acc}}\}}<\mathbf{B}}{\mathrm{S}|\mathrm{NP}_{\mathrm{nom}}}> \quad \dfrac{\mathit{obiad}}{\mathrm{NP}_{\mathrm{acc}}}}{\mathrm{S}}<$$

Modyfikatory jako argumenty

- Ta sama fraza (np. przysłówek) może być wymagana lub nie *wyglądam dobrze*, *dobrze piszę*
- Obecne rozwiązanie – przydzielić różne typy
  - *dobrze* : S‖S, AdvP
  - *piszę* : S
  - *wyglądam* : S|AdvP
- Inne możliwości
  - wszystko jest modyfikatorem (zawsze S‖S)
  - wszystko jest argumentem (zawsze AdvP)
  - argumenty opcjonalne (*piszę* : S{|AdvP$^?$})
  - specjalne reguły (AdvP $\rightarrow$ S‖S)

Reguły PL-CCG

$$(>) \quad X\{/Y, ...\}, \quad Y \vdash X\{...\}$$

$$(> B) \quad X/\!/Z, \quad Z\{...\} \vdash X\{...\}$$

$$(/\!/) \quad X/\!/Y \vdash X/Y$$

(oraz analogiczne reguły dla \ i |)

Plan

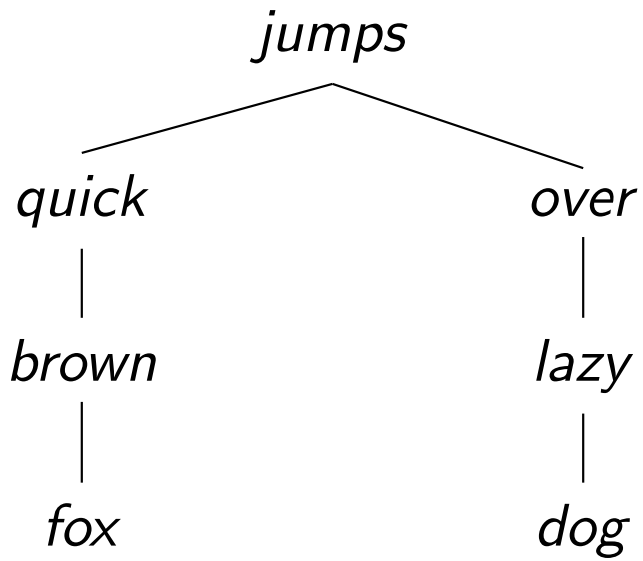
Semantyka

- „Wieloargumentowy” rachunek lambda
  - Marysia = **marysia**
  - Janka = **janek**
  - lubi = $\lambda\langle x, y\rangle.\mathbf{lubi}(x, y)$
- Służy do odzyskania zależności składniowych między słowami:

$$\mathbf{lubi(marysia,\ janek)} \rightarrow [\mathbf{lubi,\ marysia}], [\mathbf{lubi,\ janek}]$$

Zależności składniowe

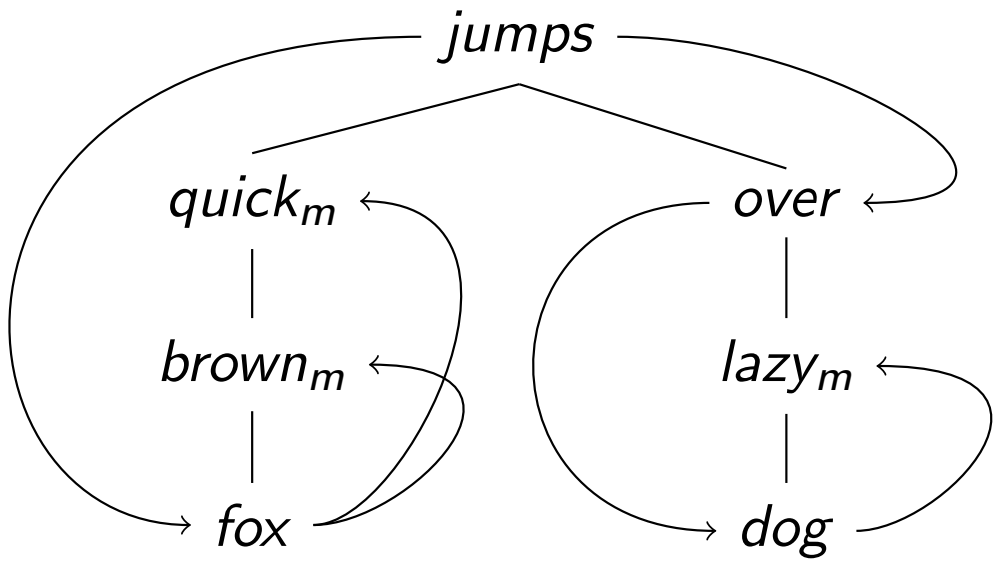
Wynikowy term nie zawsze odzwierciedla zależności!
Quick brown fox jumps over lazy dog



Widać, że modyfikatury (tu przymiotniki) stwarzają problem.

Język reprezentacji zależności

Rozwiązanie – oznaczyć węzły jako modyfikatory, zbierać zależności z ich uwzględnieniem.



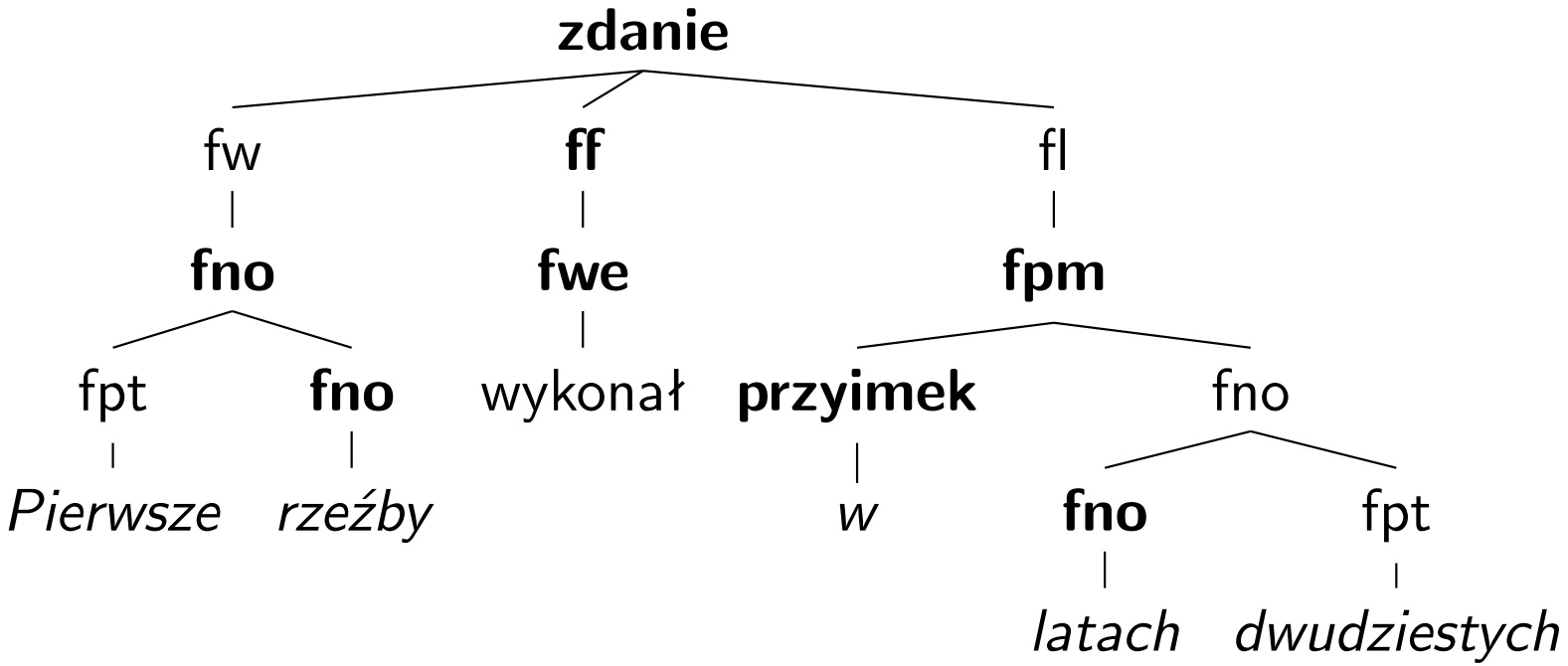
Dodatkowe komplikacje:

- modyfikatory wyższego poziomu (np. przysłówek)
- dodatkowe argumenty (np. rzeczownik we frazie przyimkowej)

Plan

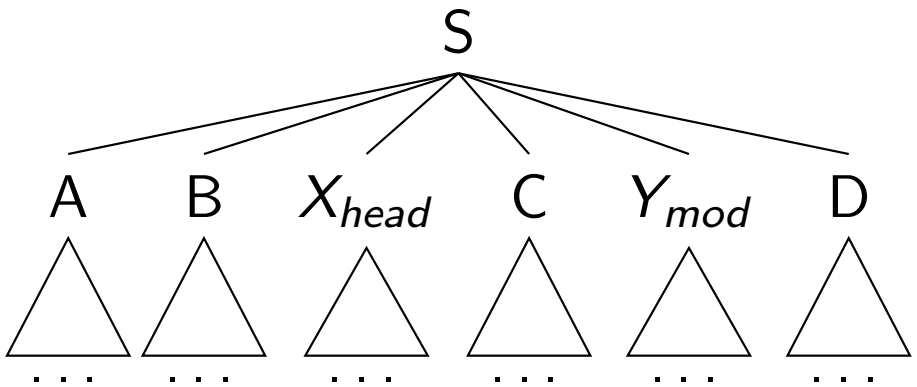
Składnica

- Składnica – polski bank ok. 8000 drzew
- Zdania sparsowane przez Świgrę, poprawne drzewa wybrane przez ludzi



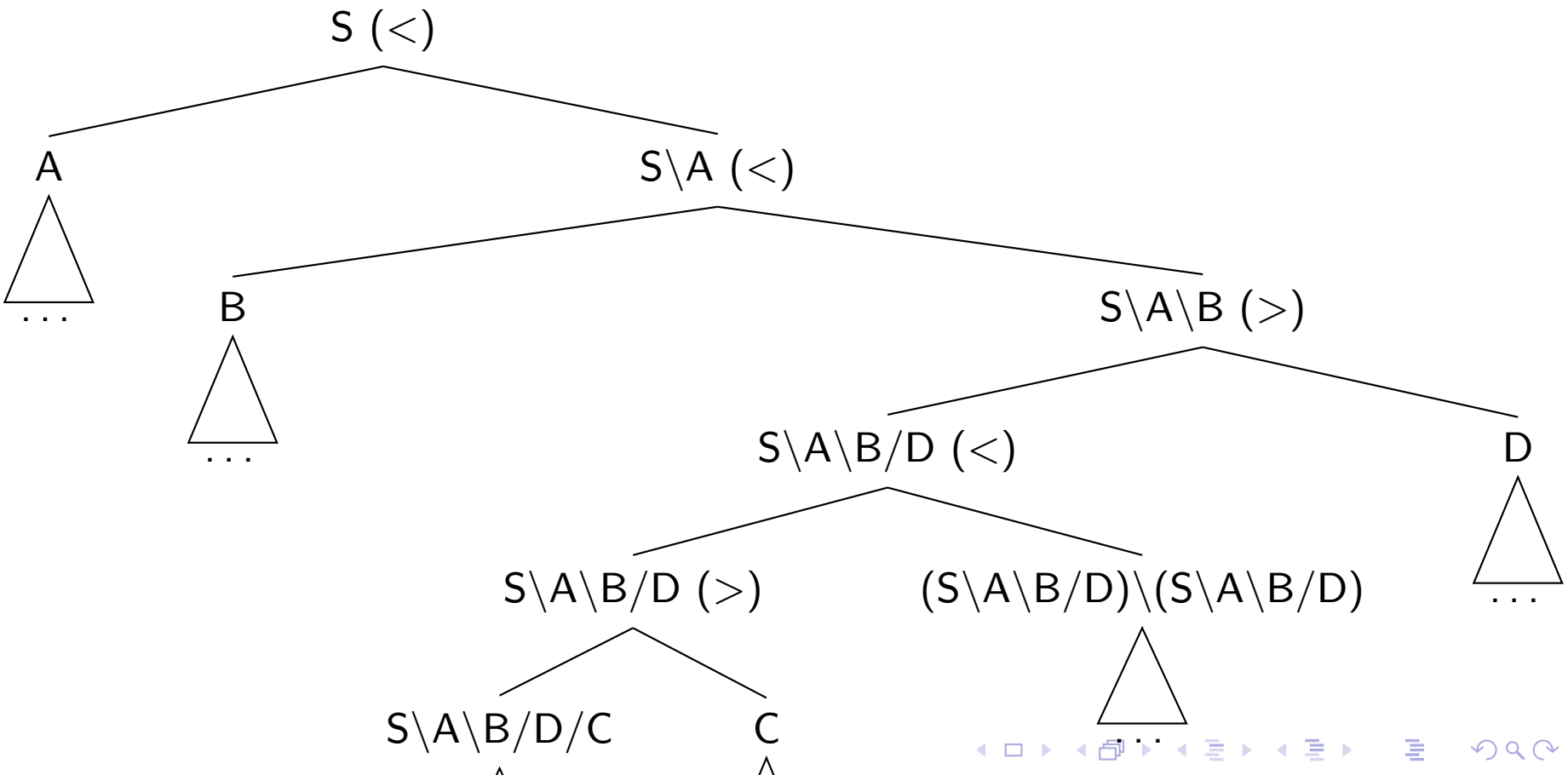
Algorytm konwersji

- Dzielimy węzły na głowę, argumenty i modyfikatory
- Argumentom przyznajemy kategorie



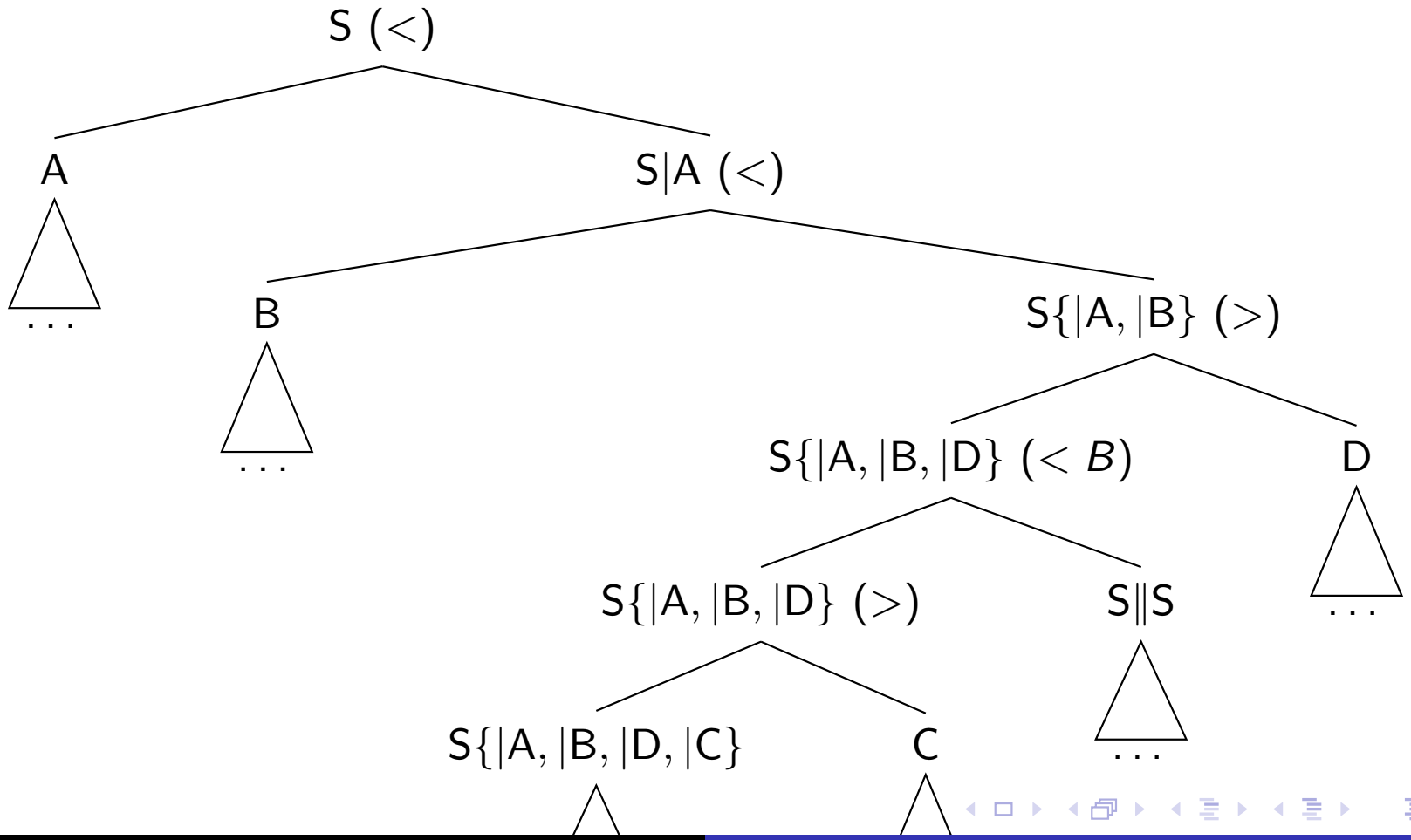
Algorytm konwersji

Aplikujemy głowę do argumentów, dodajemy modyfikator:



Algorytm konwersji

W przypadku zdań kategorie są multisetowe:



Użyte kategorie atomowe

Kategorie atomowe dla argumentów zależą od typu węzła w Świgrze:

Fraza	**Typ**	**Kategoria**	**Cechy**
nominalna	fno	NO	przyp, rodzaj, liczba, osoba
przymiotnikowa	fpt	PT	przyp, rodzaj, liczba
przyim.-nom.	fpm	PM	przyim, przyp
przyim.-przym.	fpmpt	PMPT	przyim, przyp
przysłówkowa	fps	PS	
werbalna	fwe	WE	
zdaniowa	fzd	SS	typ
zdanie	zdanie, wypowiedzenie	S	
luźna	fl	L	

Plan

Metoda ewaluacji

- Cztery wersje algorytmu: normal, no-mods, no-sets, nm+ns
- Leave-one-out cross-validation
- Parser zwraca wszystkie możliwe wyniki dla danego zdania
- Porównuję uzyskane relacje zależnościowe z wzorcowymi
- W wyniku: LP, LR, UP, UR (labeled/unlabeled, precision/recall)

Wyniki parsowania

Wersja	Sparsowane	LP	LR	UP	UR
nm+ns	215/8227 (2.61%)	97.08%	98.09%	98.45%	98.82%
no-mods	220/8227 (2.67%)	96.46%	97.87%	98.12%	98.48%
no-sets	539/8227 (6.55%)	86.78%	96.56%	91.36%	98.46%
normal	588/8227 (7.15%)	85.13%	96.32%	90.11%	98.36%

- Udaje się sparsować najwyżej 7% zdań
- Można użyć triku: zastąpić rzadkie słowa tagami...
- ...ale wtedy potrzebny jest lepszy parser

Leksykon – ilość kategorii

Wersja	**Kategorii**	**Kategorii (uogólnionych)**
nm+ns	13231	3821
no-mods	12662	3202
no-sets	9735	1546
normal	8055	1207

Uogólnione kategorie – nie bierzemy pod uwagę atrybutów (liczby, przypadku itd.)

Najczęstsze uogólnione kategorie

Oznaczenia: *fraza* o danej kategorii, *[argument]*.

	Kategoria	**Przykład**
1.	NO	*etap konkursu*
2.	S\S	kropka na końcu zdania
3.	NO|NO	*[służbę] wojskową*
4.	NO/NO	*etap [konkursu]*
5.	','	przecinek
6.	S‖S	*jeszcze raz to potwierdza*
7.	PM/NO	*Z [domu] wyniosłam zamiłowanie do pracy.*
8.	S‖S/NO	*Nie znali się za [życia].*
9.	NO|NO/NO	*zmiany w [rządzie]*
10.	S|NO	*odbywa [służbę wojskową]*

Różne problemy z leksykonem:

- modyfikatory – np. ten sam przyimek może mieć wiele typów (PM/NO, S‖S/NO, NO|NO/NO...)
- *który* jest traktowany jak zwykły rzeczownik
- wiele kategorii dla zdań (S, SS, WE)
- inne szczególne przypadki
  - być może potrzebna jest jakaś integracja ze Świgrą?

Plan

Podsumowanie

Praca dostępna jest pod adresem
`http://students.mimuw.edu.pl/~pm262952/`.

- Wyniki pracy:
  - gramatyka (PL-CCG)
  - semantyka i reprezentacja zależności
  - algorytm konwersji
  - leksykon i jego ewaluacja
- Możliwa kontynuacja:
  - lepszy parser (statystyczny)
  - szczególne przypadki w konwersji
  - inne traktowanie modyfikatorów
  - wersja bez multisetów
  - integracja ze Świgrą?

Pytania?

Bibliografia


Mark Steedman and Jason Baldridge, *Combinatory Categorial Grammar*, draft (2011), to appear in: R. Borsley and K. Borjars (eds.), *Non-Transformational Syntax*, Blackwell, 181–224.


Julia Hockenmaier, *Data and Models for Statistical Parsing with Combinatory Categorial Grammar*, PhD thesis, University of Edinburgh, 2003.


Beryl Hoffman, *The Computational Analysis of the Syntax and Interpretation of "Free" Word Order in Turkish*, PhD thesis, University of Pennsylvania, 1995.