

Parsowanie składniowe LFG i jego zastosowania

Agnieszka Patejuk i Adam Przepiórkowski



INSTYTUT PODSTAW INFORMATYKI
POLSKIEJ AKADEMII NAUK
ul. Jana Kazimierza 5, 01-248 Warszawa



Warsztaty CLARIN-PL, Wrocław, 20 maja 2015



- wykład:
 - krótki wstęp do teorii LFG
 - implementacja: polskie LFG w XLE
 - INESS i bank struktur LFG
 - zastosowania
- warsztaty:
 - prezentacja banku struktur LFG
 - wspólne ręczne ujednoznacznianie struktur



Gramatyka leksykalno-funkcyjna:

- ang. *Lexical Functional Grammar* (LFG; Bresnan 1982, 2000, Dalrymple 2001),
- teoria **generatywna** (w szerokim sensie tego terminu),
- ale **nie transformacyjna**,
- silnie **zleksykalizowana**,
- **sformalizowana**,
- **implementowalna**,
- obejmuje **różne poziomy** wiedzy lingwistycznej,
- użyta do opisu **wielu** typologicznie różnorodnych języków,
- aktywnie **rozwijana** (liczne ośrodki, coroczna konferencja).

Poziomy reprezentacji



Dwa podstawowe poziomy reprezentacji:

- **c-struktura:**

- struktura składnikowa w formie drzewa,
- oparta na **kategoriach składniowych**,
- struktura **powierzchniowa**,
- **zależna od języka**;

- **f-struktura:**

- struktura funkcyjna w formie struktury atrybutów,
- oparta na **funkcjach gramatycznych**,
- struktura **głęboka**,
- **dość uniwersalna**,
- bliska **semantycy** (ale to nie semantyka).

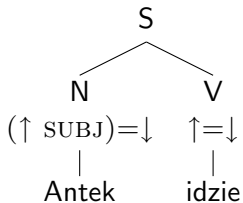
Pozostałe: s(ematic)-structure, a(rgument)-structure,
i(nformation)-structure. . .



Reguła (uproszczona) dla zdania *Antek idzie*:

- $S \rightarrow N \quad V$
(\uparrow SUBJ)= \downarrow \uparrow = \downarrow
- S (zdanie): lewa strona reguły (matka)
- N (rzeczownik) i V (czasownik): prawa strona reguły (dzieci)
- N ma anotację mówiącą, że jest to podmiot
- V ma anotację mówiącą, że przekazuje cechy matce

Powstałe struktury



PRED	'IŚĆ<[1]>'								
SUBJ	[1] <table border="1"> <tr> <td>PRED</td> <td>'ANTEK'</td> </tr> <tr> <td>CASE</td> <td>NOM</td> </tr> <tr> <td>GEND</td> <td>M1</td> </tr> <tr> <td>NUM</td> <td>SG</td> </tr> </table>	PRED	'ANTEK'	CASE	NOM	GEND	M1	NUM	SG
PRED	'ANTEK'								
CASE	NOM								
GEND	M1								
NUM	SG								
TENSE	PRES								

Funkcje gramatyczne



- **podstawowe pojęcie** w LFG,
- określa relację między **nadrzędnikiem a podrzędnikiem**,
- **niezależne od pozycji** w drzewie,
- **mogą być** w pewnym stopniu **skorelowane z semantyką**,
- funkcje są **uniwersalnie dostępne**,
- jednak nie każdy język korzysta z całego zestawu.

Podstawowy zestaw funkcji



- SUBJ: podmiot (*Antek idzie*),
- OBJ: dopełnienie bliższe (*Eryk lubi Antka*),
- OBJ_θ: dopełnienie dalsze (*Antek dał Erykowi płytę*),
- OBL: dopełnienie przyimkowe (*Eryk czeka na Antka*),
- COMP: dopełnienie zdaniowe (*Antek mówi, że idzie*),
- XCOMP: dopełnienie bezokolicznikowe (*Antek chce iść*),
- XCOMP-PRED: dopełnienie predykatywne (*Antek jest miły*).

Przykład: podmiot (SUBJ)



Antek idzie.

PRED	‘IŚĆ<1>’												
SUBJ	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 10px; vertical-align: top;"> 1 </td> <td style="padding-left: 10px;"> <table style="border-collapse: collapse;"> <tr> <td style="padding-right: 10px;">PRED</td> <td>‘ANTEK’</td> </tr> <tr> <td style="padding-right: 10px;">CASE</td> <td>NOM</td> </tr> <tr> <td style="padding-right: 10px;">GEND</td> <td>M1</td> </tr> <tr> <td style="padding-right: 10px;">NUM</td> <td>SG</td> </tr> <tr> <td style="padding-right: 10px;">PERS</td> <td>3</td> </tr> </table> </td> </tr> </table>	1	<table style="border-collapse: collapse;"> <tr> <td style="padding-right: 10px;">PRED</td> <td>‘ANTEK’</td> </tr> <tr> <td style="padding-right: 10px;">CASE</td> <td>NOM</td> </tr> <tr> <td style="padding-right: 10px;">GEND</td> <td>M1</td> </tr> <tr> <td style="padding-right: 10px;">NUM</td> <td>SG</td> </tr> <tr> <td style="padding-right: 10px;">PERS</td> <td>3</td> </tr> </table>	PRED	‘ANTEK’	CASE	NOM	GEND	M1	NUM	SG	PERS	3
1	<table style="border-collapse: collapse;"> <tr> <td style="padding-right: 10px;">PRED</td> <td>‘ANTEK’</td> </tr> <tr> <td style="padding-right: 10px;">CASE</td> <td>NOM</td> </tr> <tr> <td style="padding-right: 10px;">GEND</td> <td>M1</td> </tr> <tr> <td style="padding-right: 10px;">NUM</td> <td>SG</td> </tr> <tr> <td style="padding-right: 10px;">PERS</td> <td>3</td> </tr> </table>	PRED	‘ANTEK’	CASE	NOM	GEND	M1	NUM	SG	PERS	3		
PRED	‘ANTEK’												
CASE	NOM												
GEND	M1												
NUM	SG												
PERS	3												
TENSE	PRES												

Przykład: dopełnienie bliższe (OBJ)



Eryk lubi *Antka*.

PRED	‘LUBIĆ	⟨	1,2	⟩’	
SUBJ	1	PRED	‘ERYK’		
		CASE	NOM		
OBJ	2	PRED	‘ANTEK’		
		CASE	ACC		
TENSE	PRES				

Przykład: dopełnienie dalsze (OBJ_θ)

Antek dał *Erykowi* płytę.

PRED	‘DAĆ$\langle 1,2,3 \rangle$’
SUBJ	$\left[\begin{array}{l} \boxed{1} \left[\begin{array}{l} \text{PRED} \quad \text{‘ANTEK’} \\ \text{CASE} \quad \text{NOM} \end{array} \right] \end{array} \right]$
OBJ	$\left[\begin{array}{l} \boxed{2} \left[\begin{array}{l} \text{PRED} \quad \text{‘PŁYTA’} \\ \text{CASE} \quad \text{ACC} \end{array} \right] \end{array} \right]$
OBJ _θ	$\left[\begin{array}{l} \boxed{3} \left[\begin{array}{l} \text{PRED} \quad \text{‘ERYK’} \\ \text{CASE} \quad \text{DAT} \end{array} \right] \end{array} \right]$
TENSE	PRES

Przykład: dopełnienie przyimkowe (OBL)



*Eryk czeka **na Antka**.*

PRED	‘CZEKAĆ<[1],[2]>’						
SUBJ	[1] <table style="border-collapse: collapse; display: inline-table; vertical-align: middle;"> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">PRED</td> <td style="padding-left: 5px;">‘ERYK’</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">CASE</td> <td style="padding-left: 5px;">NOM</td> </tr> </table>	PRED	‘ERYK’	CASE	NOM		
PRED	‘ERYK’						
CASE	NOM						
OBL	[2] <table style="border-collapse: collapse; display: inline-table; vertical-align: middle;"> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">PRED</td> <td style="padding-left: 5px;">‘ANTEK’</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">CASE</td> <td style="padding-left: 5px;">ACC</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">PFORM</td> <td style="padding-left: 5px;">NA</td> </tr> </table>	PRED	‘ANTEK’	CASE	ACC	PFORM	NA
PRED	‘ANTEK’						
CASE	ACC						
PFORM	NA						
TENSE	PRES						

Przykład: dopełnienie zdaniowe (COMP)



Antek mówi, że idzie.

PRED	‘MÓWIĆ<[1],[2]>’										
SUBJ	[1] <table style="border-left: 1px solid black; border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px; margin-left: 10px;"> <tr> <td style="padding: 2px 5px;">PRED</td> <td style="padding: 2px 5px;">‘ANTEK’</td> </tr> <tr> <td style="padding: 2px 5px;">CASE</td> <td style="padding: 2px 5px;">NOM</td> </tr> </table>	PRED	‘ANTEK’	CASE	NOM						
PRED	‘ANTEK’										
CASE	NOM										
COMP	[2] <table style="border-left: 1px solid black; border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px; margin-left: 10px;"> <tr> <td style="padding: 2px 5px;">PRED</td> <td style="padding: 2px 5px;">‘IŚĆ<[3]>’</td> </tr> <tr> <td style="padding: 2px 5px;">SUBJ</td> <td style="padding: 2px 5px;">[3] <table style="border-left: 1px solid black; border-right: 1px solid black; border-bottom: 1px solid black; padding: 2px 5px; margin-left: 5px;"> <tr> <td style="padding: 2px 5px;">PRED</td> <td style="padding: 2px 5px;">‘PRO’</td> </tr> </table> </td> </tr> <tr> <td style="padding: 2px 5px;">TENSE</td> <td style="padding: 2px 5px;">PRES</td> </tr> <tr> <td style="padding: 2px 5px;">COMP-FORM</td> <td style="padding: 2px 5px;">ŻE</td> </tr> </table>	PRED	‘IŚĆ<[3]>’	SUBJ	[3] <table style="border-left: 1px solid black; border-right: 1px solid black; border-bottom: 1px solid black; padding: 2px 5px; margin-left: 5px;"> <tr> <td style="padding: 2px 5px;">PRED</td> <td style="padding: 2px 5px;">‘PRO’</td> </tr> </table>	PRED	‘PRO’	TENSE	PRES	COMP-FORM	ŻE
PRED	‘IŚĆ<[3]>’										
SUBJ	[3] <table style="border-left: 1px solid black; border-right: 1px solid black; border-bottom: 1px solid black; padding: 2px 5px; margin-left: 5px;"> <tr> <td style="padding: 2px 5px;">PRED</td> <td style="padding: 2px 5px;">‘PRO’</td> </tr> </table>	PRED	‘PRO’								
PRED	‘PRO’										
TENSE	PRES										
COMP-FORM	ŻE										
TENSE	PRES										

Przykład: dopełnienie bezkolicznikowe (XCOMP)



Antek chce *iść*.

PRED		‘CHCIEĆ<1,2>’				
SUBJ	1	<table> <tr> <td>PRED</td> <td>‘ANTEK’</td> </tr> <tr> <td>CASE</td> <td>NOM</td> </tr> </table>	PRED	‘ANTEK’	CASE	NOM
PRED	‘ANTEK’					
CASE	NOM					
XCOMP	2	<table> <tr> <td>PRED</td> <td>‘IŚĆ<1>’</td> </tr> </table>	PRED	‘IŚĆ<1>’		
PRED	‘IŚĆ<1>’					
TENSE		PRES				

Antek kazał Erykowi *iść*.

PRED		‘KAZAĆ<1,2,3>’				
SUBJ	1	<table> <tr> <td>PRED</td> <td>‘ANTEK’</td> </tr> <tr> <td>CASE</td> <td>NOM</td> </tr> </table>	PRED	‘ANTEK’	CASE	NOM
PRED	‘ANTEK’					
CASE	NOM					
OBJ _θ	2	<table> <tr> <td>PRED</td> <td>‘ERYK’</td> </tr> <tr> <td>CASE</td> <td>DAT</td> </tr> </table>	PRED	‘ERYK’	CASE	DAT
PRED	‘ERYK’					
CASE	DAT					
XCOMP	3	<table> <tr> <td>PRED</td> <td>‘IŚĆ<2>’</td> </tr> </table>	PRED	‘IŚĆ<2>’		
PRED	‘IŚĆ<2>’					
TENSE		PRES				

Przykład: dopełnienie bezkolicznikowe (XCOMP)



Antek kazał Erykowi iść.

Antek chce iść.

PRED	‘CHCIEĆ<1,2>’
SUBJ	<div style="display: inline-block; border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px; margin-right: 5px;">1</div> <div style="display: inline-block; border-left: 1px solid black; border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px 10px;"> PRED ‘ANTEK’ CASE NOM </div>
XCOMP	<div style="display: inline-block; border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px; margin-right: 5px;">2</div> <div style="display: inline-block; border-left: 1px solid black; border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px 10px;"> PRED ‘IŚĆ<1>’ </div>
TENSE	PRES

PRED	‘KAZAĆ<1,2,3>’
SUBJ	<div style="display: inline-block; border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px; margin-right: 5px;">1</div> <div style="display: inline-block; border-left: 1px solid black; border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px 10px;"> PRED ‘ANTEK’ CASE NOM </div>
OBJ _θ	<div style="display: inline-block; border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px; margin-right: 5px;">2</div> <div style="display: inline-block; border-left: 1px solid black; border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px 10px;"> PRED ‘ERYK’ CASE DAT </div>
XCOMP	<div style="display: inline-block; border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px; margin-right: 5px;">3</div> <div style="display: inline-block; border-left: 1px solid black; border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px 10px;"> PRED ‘IŚĆ<2>’ </div>
TENSE	PRES

Przykład: dopełnienie predykatywne (XCOMP-PRED)



Antek uważa Eryka za miłego.

Antek jest miły.

[PRED		‘BYĆ<1,2>’]
	SUBJ	1	PRED ‘ANTEK’	
			CASE NOM	
	XC-P	2	PRED ‘MIŁY<1>’	
CASE NOM				
TENSE		PRES		

[PRED		‘UWAŻAĆ<1,2,3>’]
	SUBJ	1	PRED ‘ANTEK’	
			CASE NOM	
	OBJ	2	PRED ‘ERYK’	
CASE ACC				
XC-P	3	PRED ‘MIŁY<2>’		
		CASE ACC		
		PFORM	ZA	
TENSE		PRES		

Przykład: dopełnienie predykatywne (XCOMP-PRED)



Antek uważa Eryka za miłego.

Antek jest miły.

PRED	‘BYĆ<1,2>’				
SUBJ	1 <table style="border-left: 1px solid black; border-right: 1px solid black; border-collapse: collapse;"> <tr> <td style="padding: 5px;">PRED</td> <td style="padding: 5px;">‘ANTEK’</td> </tr> <tr> <td style="padding: 5px;">CASE</td> <td style="padding: 5px;">NOM</td> </tr> </table>	PRED	‘ANTEK’	CASE	NOM
PRED	‘ANTEK’				
CASE	NOM				
XC-P	2 <table style="border-left: 1px solid black; border-right: 1px solid black; border-collapse: collapse;"> <tr> <td style="padding: 5px;">PRED</td> <td style="padding: 5px;">‘MIŁY<1>’</td> </tr> <tr> <td style="padding: 5px;">CASE</td> <td style="padding: 5px;">NOM</td> </tr> </table>	PRED	‘MIŁY<1>’	CASE	NOM
PRED	‘MIŁY<1>’				
CASE	NOM				
TENSE	PRES				

PRED	‘UWAŻAĆ<1,2,3>’						
SUBJ	1 <table style="border-left: 1px solid black; border-right: 1px solid black; border-collapse: collapse;"> <tr> <td style="padding: 5px;">PRED</td> <td style="padding: 5px;">‘ANTEK’</td> </tr> <tr> <td style="padding: 5px;">CASE</td> <td style="padding: 5px;">NOM</td> </tr> </table>	PRED	‘ANTEK’	CASE	NOM		
PRED	‘ANTEK’						
CASE	NOM						
OBJ	2 <table style="border-left: 1px solid black; border-right: 1px solid black; border-collapse: collapse;"> <tr> <td style="padding: 5px;">PRED</td> <td style="padding: 5px;">‘ERYK’</td> </tr> <tr> <td style="padding: 5px;">CASE</td> <td style="padding: 5px;">ACC</td> </tr> </table>	PRED	‘ERYK’	CASE	ACC		
PRED	‘ERYK’						
CASE	ACC						
XC-P	3 <table style="border-left: 1px solid black; border-right: 1px solid black; border-collapse: collapse;"> <tr> <td style="padding: 5px;">PRED</td> <td style="padding: 5px;">‘MIŁY<2>’</td> </tr> <tr> <td style="padding: 5px;">CASE</td> <td style="padding: 5px;">ACC</td> </tr> <tr> <td style="padding: 5px;">PFORM</td> <td style="padding: 5px;">ZA</td> </tr> </table>	PRED	‘MIŁY<2>’	CASE	ACC	PFORM	ZA
PRED	‘MIŁY<2>’						
CASE	ACC						
PFORM	ZA						
TENSE	PRES						

Struktura gramatyki LFG



Dwie podstawowe części:

- **leksykon** (wpisy dla poszczególnych słów):
 - ograniczenia morfoskładniowe,
 - ograniczenia walencyjne,
 - ograniczenia zleksykalizowane;
- **reguły**:
 - umożliwiają łączenie słów we frazy,
 - mogą wprowadzać dodatkowe ograniczenia.

Poza tym:

- tokenizator,
- morfologia,
- makra,
- deklaracja cech.

O gramatyce POLFIE



Gramatyka POLFIE:

- podejście **LFG**,
- implementacja w **XLE** (platforma dedykowana LFG),
- stworzona w oparciu o **wcześniejsze implementowane gramatyki języka polskiego**:
 - GFJP (DCG): Świdziński 1992, Woliński 2004,
 - FOJP (HPSG): Przepiórkowski *et al.* 2002;
- **korzysta** z informacji:
 - walencyjnych z **Walentego**,
 - fleksyjnych z **Morfeusza**,
- **testowana** na:
 - banku struktur Składnica: Świdziński and Woliński 2010,
 - korpusie NKJP (szczególnie NKJP1M);
- intensywnie **rozwijana** (m.in. semantyka).

Analizator fleksyjny Morfeusz



Morfeusz (Woliński 2006, 2014):

- **najlepszy** analizator dla języka polskiego,
- **szybki, stabilny**, wygodny w użyciu,
- bardzo **duże pokrycie**,
- tagset podobny do NKJP,
- **publicznie dostępny** na licencji swobodnej (*open source*):
<http://sgjp.pl/morfeusz/>.

Źródło:

- informacji o charakterystyce fleksyjnej,
- informacje są konwertowane na wpisy leksykalne.

Analizator fleksyjny Morfeusz



Morfeusz (Woliński 2006, 2014):

- **najlepszy** analizator dla języka polskiego,
- **szybki, stabilny**, wygodny w użyciu,
- bardzo **duże pokrycie**,
- tagset podobny do NKJP,
- **publicznie dostępny** na licencji swobodnej (*open source*):
<http://sgjp.pl/morfeusz/>.

Źródło:

- informacji o charakterystyce fleksyjnej,
- informacje są konwertowane na wpisy leksykalne.

Przykład konwersji informacji z Morfeusza



Wybrana interpretacja z Morfeusza:

- kot, kot, subst:sg:nom:m2
- forma: kot
- lemat: kot
- znacznik:
 - część mowy: rzeczownik (subst),
 - liczba: pojedyncza (sg),
 - przypadek: mianownik (nom),
 - rodzaj: męsko-zwierzęcy (m2).

Po konwersji:

```
kot  N  *  (^ PRED)= 'kot'
      (^ NUM)= sg
      (^ CASE)= nom
      (^ GEND)= m2
```

Przykład konwersji informacji z Morfeusza



Wybrana interpretacja z Morfeusza:

- kot, kot, subst:sg:nom:m2
- forma: kot
- lemat: kot
- znacznik:
 - część mowy: rzeczownik (subst),
 - liczba: pojedyncza (sg),
 - przypadek: mianownik (nom),
 - rodzaj: męsko-zwierzęcy (m2).

Po konwersji:

kot N * (^ PRED)='kot'
 (^ NUM)= sg
 (^ CASE)= nom
 (^ GEND)= m2

Słownik walencyjny Walenty



Walenty (Przepiórkowski *et al.* 2014b,c,a):

- **elektroniczny** słownik walencyjny (dla ludzi i maszyn),
- **duży i szczegółowy**,
- obecnie głównie **powierzchniowoskładniowy**,
- niebawem będzie dodawany poziom **semantyczny**,
- główne źródło empiryczne: NKJP (Przepiórkowski *et al.* 2012),
- **publicznie dostępny** na licencji swobodnej (*open source*):
<http://walenty.ipipan.waw.pl/>.

Źródło:

- informacji o wymaganiach walencyjnych predykatów,
- informacje są konwertowane na wpisy leksykalne.

Słownik walencyjny Walenty



Walenty (Przepiórkowski *et al.* 2014b,c,a):

- **elektroniczny** słownik walencyjny (dla ludzi i maszyn),
- **duży i szczegółowy**,
- obecnie głównie **powierzchniowoskładniowy**,
- niebawem będzie dodawany poziom **semantyczny**,
- główne źródło empiryczne: NKJP (Przepiórkowski *et al.* 2012),
- **publicznie dostępny** na licencji swobodnej (*open source*):
<http://walenty.ipipan.waw.pl/>.

Źródło:

- informacji o wymaganiach walencyjnych predykatów,
- informacje są konwertowane na wpisy leksykalne.

Przykład konwersji informacji z Walentego



Wybrany schemat z Walentego:

- chrapać: pewny: -: : imperf: subj{np(str)}
- predykat: chrapać
- ocena schematu: pewny
- aspekt: niedokonany (imperf)
- argumenty: podmiot nominalny (subj{np(str)})

Po konwersji (wersja uproszczona):

(^ PRED)='chrapać<(^ SUBJ)>'

(^ SUBJ CASE)=c nom

Przykład konwersji informacji z Walentego



Wybrany schemat z Walentego:

- chrapać: pewny: _: : imperf: subj{np(str)}
- predykat: chrapać
- ocena schematu: pewny
- aspekt: niedokonany (imperf)
- argumenty: podmiot nominalny (subj{np(str)})

Po konwersji (wersja uproszczona):

(^ PRED)= 'chrapać<(^ SUBJ)>'

(^ SUBJ CASE)=c nom

Jak działa POLFIE?



- tekst jest dzielony na zdania
- dla każdego słowa w zdaniu tworzony jest wpis leksykalny:
 - informacja fleksyjna z Morfeusza
 - informacja walencyjna z Walentego
- parser XLE generuje struktury (c- i f-) zgodne z gramatyką
- struktury można wyeksportować do pliku (Prolog, XML)
- niejednoznaczne struktury trzeba ujednoznaczyć:
 - w XLE: od razu, ale mało wygodnie
 - w INESS: dedykowane zaawansowane narzędzie

Krótkie wprowadzenie



Bank struktur to:

- zbiór **wzorcowych** rozkładów (c- i f-struktur)
- tworzony **ręcznie lub automatycznie**

Potencjalne zastosowania:

- **prezentacja** wyników
- możliwość **przeszukiwania**
- pomoc przy **rozwoju zasobów** (szczególnie gramatyki)
- **trenowanie** parsera stochastycznego
- na podstawie struktur można tworzyć **nowe zasoby**



- stworzony i aktywnie **rozwijany** na uniwersytecie w **Bergen** (Rosén *et al.* 2007)
- dostępny **przez sieć**, działa w różnych przeglądarkach:
 - instalacja norweska: <http://iness.uib.no/>
 - instalacja lokalna ZIL:
<http://iness.mozart.ipipan.waw.pl/>
- bardzo **elastyczny, wygodny i szybki** (można wybrać poprawny rozbiór spośród 5000 w 8 krokach)
- daje możliwość tworzenia **różnych** banków struktur, m.in.:
 - LFG
 - zależnościowe
- zintegrowany z XLE-Web: możliwość **parsowania online**

Jak działa INESS?



- struktury (c- i f-) są analizowane przez **INESS**:
 - znajdowane są miejsca, gdzie struktury się różnią,
 - **wyróżniki** służą do wybierania rozbioru na podstawie różnic,
 - wyróżniki dotyczą różnych poziomów struktury (c- i f-);
- przykładowe wyróżniki:
 - **liczba argumentów** wymaganych przez predykat,
 - miejsce **przyłączenia** (relacja nadrzędnik–podrzędnik);
- wybieranie wyróżników:
 - **pozytywne**: wybiera rozkłady spełniające warunek,
 - **negatywne**: wybiera rozkłady, które nie spełniają warunku,
 - swoboda: **dowolna** kolejność wyboru wyróżników,
- można **przenosić** wyróżniki między wersjami:

INESS: struktury przed ujednoznacznieniem



Discriminants

Selected solutions: 2 of 2 | gold no good finished

spurious amb. bad source

Order by: type/anchor frequency disc. power

Jak wygląda przepiórka ?

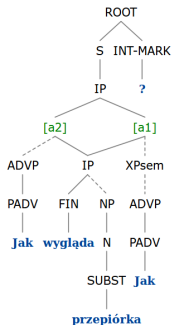
F-structure discriminants | show all

0:5	_TOP 'wyglądać<[],[]>'	1	compl (1)
0:5	_TOP 'wyglądać<[]>'	1	compl (1)
5:1	'wyglądać<[],[]>' OBL-MOD 'jak'	1	compl (1)
5:14	'wyglądać<[],[]>' SUBJ 'przepiórka'	1	compl (1)
5:1	'wyglądać<[]>' ADJUNCT \$ 'jak'	1	compl (1)
5:14	'wyglądać<[]>' SUBJ 'przepiórka'	1	compl (1)

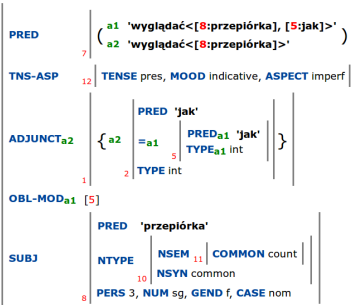
C-structure discriminants

1	Jak wygląda przepiórka		
	IP -> XPsem IP	1	compl (1)
	IP -> ADVP IP	1	compl (1)

C-structure



F-structure



INESS: ujednoznacznione struktury



Discriminants

Selected solutions: 1 of 2 | gold no good finished spurious amb.

bad source

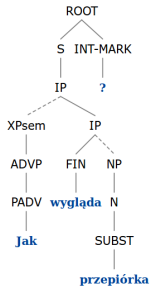
Order by: type/anchor frequency disc. power

Jak wygląda przepiórka ?

F-structure discriminants | show all

5:1 'wyglądać<[],[]>' OBL-MOD 'jak'

C-structure



F-structure

PRED	'wyglądać<[8:przepiórka], [5:jak]>'
TNS-ASP	TENSE pres, MOOD indicative, ASPECT imperf
OBL-MOD	PRED 'jak' TYPE int
SUBJ	PRED 'przepiórka' NTYPE NSEM COMMON count NSYN common PERS 3, NUM sg, GEND f, CASE nom

Strategia ujednoznaczniania



Strategia przyjęta przy tworzeniu polskiego banku struktur LFG:

- preferować wyróżniki dotyczące **f-struktury**
- wybierać tylko **niekontrowersyjne** wyróżniki
- **częściowe** ujednoznacznienie jest dopuszczalne (w sytuacji prawdziwej niejednoznaczności konieczne)
- dodawać **komentarze** do problematycznych zdań



- f-struktura jako **przybliżenie semantyki**
- **ekstrakcja informacji** z f-struktur
- rozszerzenie o **poziom semantyczny** (warsztaty CLARIN w Krakowie):
 - lepsza **ekstrakcja informacji**
 - rozpoznawanie relacji **wynikania** między tekstami
 - **wyszukiwanie semantyczne**

Podsumowanie



Przedstawiliśmy:

- krótki wstęp do teorii LFG
- polskie LFG i jego strukturę
- polski bank struktur LFG i INESS
- zastosowania parsowania LFG

POLFIE

<http://zil.ipipan.waw.pl/LFG/>

INESS@ZIL

<http://iness.mozart.ipipan.waw.pl/>

Pytania?



"Dziękujemy za uwagę."

[PRED	'dziękować<[6-SUBJ:pro], [10:uwaga]>']
[SUBJ	[]
		PRED 'pro'	
		CASE nom, NUM pl, PERS 1	
		PRED 'uwaga'	
		CHECK [_CAT subst]	
[OBL	[]
		NTYPE [NSEM [COMMON count]	
		[NSYN common	
		10 [CASE acc, GEND f, NUM sg, PERS 3, PFORM za, PTYPE nose]	
		CHECK [_CAT fin]	
6	TNS-ASP	[ASPECT imperf, MOOD indicative, TENSE pres]]

Dziękujemy za uwagę.

Podsumowanie



Przedstawiliśmy:

- krótki wstęp do teorii LFG
- polskie LFG i jego strukturę
- polski bank struktur LFG i INESS
- zastosowania parsowania LFG

POLFIE

<http://zil.ipipan.waw.pl/LFG/>

INESS@ZIL

<http://iness.mozart.ipipan.waw.pl/>

- Bresnan, J., editor (1982). *The Mental Representation of Grammatical Relations*. MIT Press Series on Cognitive Theory and Mental Representation. The MIT Press, Cambridge, MA.
- Bresnan, J. (2000). *Lexical-Functional Syntax*. Blackwell Textbooks in Linguistics. Blackwell.
- Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors (2014). *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014*, Reykjavík, Iceland. ELRA.
- Dalrymple, M. (2001). *Lexical-Functional Grammar*. Academic Press.
- Przepiórkowski, A., Kupść, A., Marciniak, M., and Mykowiecka, A. (2002). *Formalny opis języka polskiego: Teoria i implementacja*. Akademicka Oficyna Wydawnicza EXIT, Warsaw.
- Przepiórkowski, A., Bańko, M., Górski, R. L., and Lewandowska-Tomaszczyk, B., editors (2012). *Narodowy Korpus Języka Polskiego [Eng.: National Corpus of Polish]*. Wydawnictwo Naukowe PWN, Warsaw.
- Przepiórkowski, A., Hajnicz, E., Patejuk, A., and Woliński, M. (2014a). Extended phraseological information in a valence dictionary for NLP applications. In *Proceedings of the Workshop on Lexical and*


Grammatical Resources for Language Processing (LG-LP 2014), pages 83–91, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Przepiórkowski, A., Skwarski, F., Hajnicz, E., Patejuk, A., Świdziński, M., and Woliński, M. (2014b). Modelowanie własności składniowych czasowników w nowym słowniku walencyjnym języka polskiego. *Polonica*, **XXXIII**, 159–178.

Przepiórkowski, A., Hajnicz, E., Patejuk, A., Woliński, M., Skwarski, F., and Świdziński, M. (2014c). Walenty: Towards a comprehensive valence dictionary of Polish. In Calzolari *et al.* (2014), pages 2785–2792.

Rosén, V., Meurer, P., and Smedt, K. D. (2007). Designing and implementing discriminants for LFG grammars. In M. Butt and T. H. King, editors, *The Proceedings of the LFG'07 Conference*, pages 397–417, University of Stanford, California, USA. CSLI Publications.

Woliński, M. (2004). *Komputerowa weryfikacja gramatyki Świdzińskiego*. Ph.D. dissertation, Institute of Computer Science, Polish Academy of Sciences, Warsaw.

Woliński, M. (2006). Morfeusz — a Practical Tool for the Morphological Analysis of Polish. In M. Kłopotek, S. T. Wierzchoń, and  33/33

K. Trojanowski, editors, *Intelligent information processing and web mining*, pages 503–512. Springer-Verlag.

Woliński, M. (2014). Morfeusz reloaded. In Calzolari *et al.* (2014), pages 1106–1111.

Świdziński, M. (1992). *Gramatyka formalna języka polskiego*, volume 349 of *Rozprawy Uniwersytetu Warszawskiego*. Wydawnictwa Uniwersytetu Warszawskiego, Warsaw.

Świdziński, M. and Woliński, M. (2010). Towards a bank of constituent parse trees for Polish. In *Text, Speech and Dialogue: 13th International Conference, TSD 2010, Brno, Czech Republic*, Lecture Notes in Artificial Intelligence, pages 197–204, Berlin. Springer-Verlag.