

Gramatické formalismy pro ZPJ

Aleš Horák

E-mail: hales@fi.muni.cz
http://nlp.fi.muni.cz/poc_lingv/

Obsah:

- ▶ Gramatické formalismy
- ▶ Kategoriaální gramatiky
- ▶ Závislostní gramatiky
- ▶ Stromové gramatiky TAG a LTAG
- ▶ Lexikální funkční gramatiky LFG

Kategoriaální gramatiky

- ▶ **kategoriaální gramatika** (categorial grammar, CG) – skupina teorií syntaxe a sémantiky PJ s velkým důrazem na **lexikon**
 - ▶ neobsahuje *pravidla* pro kombinování slov → **lexikální kategorie** slov tvoří **funkce**, které určují, jak se dané kategorie kombinují s jinými výraz je výsledkem **aplikace podvýrazů na sebe**
- pěkný := NP/N ... funkce, která má argument N a vrací NP

- ▶ všechny verze CG se opírají o **princip kompozicionality**:
Význam složeného výrazu je jednoznačně určen významy částí tohoto výrazu a způsobem, jakým jsou tyto části složeny dohromady.
- ▶ **zakladatelé** generativních gramatik – Leśniewski (publ. 1929) a Ajdukiewicz (publ. 1935) ve vazbě na Husserlova a Russellova teorii kategorií a teorii typů
- ▶ první použití kategoriaálních gramatik pro **popis přirozeného jazyka** – Jehošua Bar-Hillel, 1953

Gramatické formalismy

- ▶ existuje množství různých přístupů k formální specifikaci gramatik (přirozených jazyků), různé **gramatické formalismy**
- ▶ popíšeme několik nejrozšířenějších formalismů:
 - *kategoriaální gramatiky* – categorial grammars, CG
 - *závislostní gramatiky* – dependency grammars
 - *stromové gramatiky* – (Lexicalized) Tree Adjoining Grammar, (L)TAG
 - *lexikální funkční gramatiky* – Lexical Functional Grammar, LFG
 - *gramatiky příznakových struktur* – Head Phrase Structure Grammar, HPSG
- ▶ soustředíme se na **zápis gramatiky** (notaci)

Notace kategoriaálních gramatik

- ▶ existuje několik různých variant notace

$$\begin{array}{ccc}
 \frac{\frac{\text{šikovní}}{NP/N} \quad \frac{\text{psi}}{N}}{NP} > & \frac{\frac{\text{mají rádi}}{(S \setminus NP)/NP} \quad \frac{\text{kočky}}{NP}}{S \setminus NP} > \\
 & S <
 \end{array}$$

- ▶ jiný rozšířený zápis – **výsledek na vrcholku** (result on top) Lambek 1958

$$\begin{array}{ccc}
 \frac{\frac{\text{šikovní}}{NP/N} \quad \frac{\text{psi}}{N}}{NP} > & \frac{\frac{\text{mají rádi}}{(NP \setminus S)/NP} \quad \frac{\text{kočky}}{NP}}{NP \setminus S} > \\
 & S
 \end{array}$$

Notace kategoriálních gramatik – pokrač.

kategoriální gramatika je šestice $\langle \Sigma, C_{base}, C, Lex, RS, C_{complete} \rangle$, kde

- Σ je konečná množina **slov**
- C_{base} je konečná množina **základních kategorií** (funkčních typů)
- C je množina **kategorií** definovaná induktivně takto:
 - $C_{base} \subseteq C$
 - pokud $X, Y \in C$, potom i $(X/Y) \in C$ a $(X \setminus Y) \in C$
 - C obsahuje pouze prvky dané výše uvedenými body a) a b)
- $Lex \subseteq \Sigma \times C$ je konečná množina – **lexikon** (zapisujeme v indexovém tvaru slovo_{kategorie})
- RS je množina následujících **schémat pravidel**:
 - $\alpha_{(X/Y)} \circ \beta_{(Y)} \rightarrow \alpha\beta_{(X)}$
 - $\beta_{(Y)} \circ \alpha_{(X \setminus Y)} \rightarrow \beta\alpha_{(X)}$,
 kde $\alpha, \beta \in \Sigma$ a $X, Y \in C$
- $C_{complete} \subseteq C$ je množina **dokončených (kompletních) kategorií**

Rozšíření kategoriálních gramatik

- ▶ klíčový problém – nespojitě větné části, tzv. **neprojektivity**
- ▶ řešení pomocí rozšíření CG – přídavné **kombinatorické operátory** založené na **typech**
- ▶ dva možné přístupy:
 - ▶ **pravidlově orientovaný** přidává pravidla odpovídající jednoduchým operacím nad kategoriemi, jako jsou:
 - **wrap** – komutace argumentů
 - **type-raising** – aplikace typů podobná aplikaci tradičních pádů na jmenné fráze
 - **comp** – kompozice funkcí
 k nejpracovnějším systémům tohoto typu patří **kombinatorické kategoriální gramatiky (CCG)**.
 - ▶ **deduktivní přístup** vychází z Lambekova syntaktického kalkulu
 - pohled na kategoriální lomítko (slash) jako formu **logické implikace**
 - axiomy a inferenční pravidla potom definují **teorii důkazu**
např. *aplikace funkce* \approx pravidlo *modus ponens* $P \wedge (P \Rightarrow Q) \Rightarrow Q$

OpenCCG library – <http://openccg.sourceforge.net/>

Notace kategoriálních gramatik – pokrač.

- ▶ daná schémata umožňují 2 způsoby kombinace:
 - argument **vpravo** (/) – $\alpha_{(X/Y)} \circ \beta_{(Y)} \rightarrow \alpha\beta_{(X)}$
 - argument **vlevo** (\) – $\beta_{(Y)} \circ \alpha_{(X \setminus Y)} \rightarrow \beta\alpha_{(X)}$
- ▶ tento typ kategoriální gramatiky označoval Bar-Hillel jako **obousměrný** (bidirectional CG)
- ▶ Karel miluje Marii:
 - bázové kategorie = $\{NP, S\}$
 - kategorie z lexikonu: Karel_(NP), Marii_(NP), miluje_{((S \setminus NP)/NP)}
 - $C_{complete} = \{S\}$
- ▶ v tomto tvaru je odvození **ekvivalentní derivačním stromům** CFG
- ▶ existují ale **rozšíření kategoriálních gramatik**, která vedou k systémům s vyšší vyjadřovací silou, než mají standardní CFG

Závislostní gramatiky

- ▶ blízko ke kategoriálním gramatikám – vztah **závislosti** mezi **řídícími** a **závislými** větnými členy
- ▶ vhodné pro popis jazyků s volným slovosledem
- ▶ používají výhradně **lexikalizovaných uzlů** (v závislostním stromu) – neexistují žádné neterminály
→ závislostní analýza se jeví *jednodušší*
- ▶ využívá **valence** či subkategorizace – vztah mezi jedním slovem a jeho argumenty
typicky vztah mezi slovesem a jeho možnými doplněními:

nosit
= koho|co
= komu & koho|co

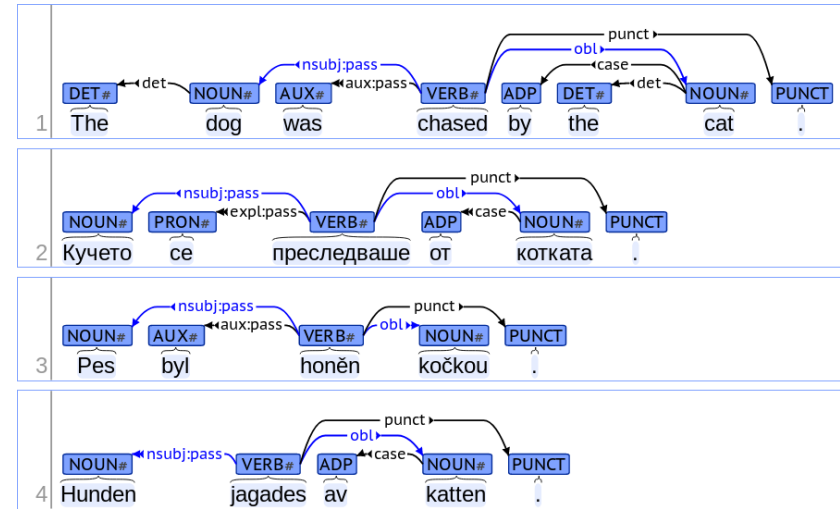
Závislostní gramatiky – pokrač.

hlavní přístupy:

- ▶ navazuje na evropskou lingvistickou tradici – až k antice
- ▶ nejstarší užití – Tesnière 1959
- ▶ **funkční generativní popis** (*Functional Generative Description*, FGD) – jeden z nejpracovanějších závislostních systémů, pražská lingvistická škola (Sgall, Hajičová, Panevová)
- ▶ UDG, *Unification Dependency Grammar* – Maxwell
- ▶ MTT, *Meaning-Text Theory* – Mel'čuk
- ▶ WG, *Word Grammar* – Hudson
- ▶ Lexicase – Starosta
- ▶ FG, *Functional Grammar* – Dik
- ▶ LG, *Link Grammar* – Temperley, Carnegie Mellon University <http://www.link.cs.cmu.edu/link/>
- ▶ DUG, *Dependency Unification Grammar* – Halliday

Universal Dependencies

- ▶ www.universaldependencies.org, **UD**
- ▶ sjednocení **závislostní anotace** pro různé jazyky
- ▶ více než **100 stromových bank** (*treebanks*) ve více než 60 jazycích



Google Universal Tagset

- ▶ **gramatiky** pro jednotlivé jazyky založené na **podobných principech**
- ▶ detaily **značkování** ale často **nejsou převoditelné 1:1**
- ▶ **sjednocení** – značkování v UD založené na minimalistické **Google Universal Tagset**

Open class words	Closed class words	Other
ADJ	ADP	PUNCT
ADV	AUX	SYM
INTJ	CCONJ	X
NOUN	DET	
PROPN	NUM	
VERB	PART	
	PRON	
	SCONJ	

Universal Features

- ▶ značky z **Universal Tagset** vymezují základní **třídy**
- ▶ lexikální a gramatické vztahy popisují **Universal Features**

Lexical features	Inflectional features	
	Nominal	Verbal
PronType	Gender	VerbForm
NumType	Animacy	Mood
Poss	Number	Tense
Reflex	Case	Aspect
Foreign	Definite	Voice
Abbr	Degree	Evident
		Polarity
		Person
		Polite

Jazykové instrukce pro Universal Dependencies

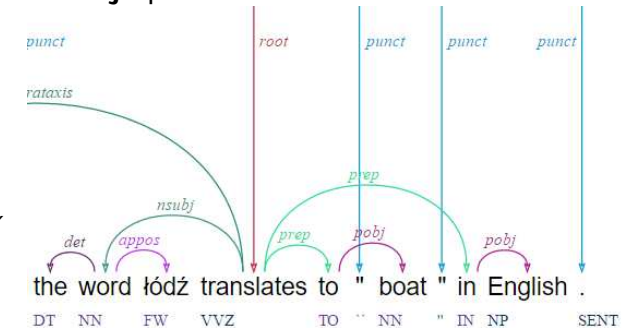
- ▶ každý jazyk má uvedené **instrukce** pro:
 - tokenizaci (hranice slov)
 - morfologické značky
 - syntax – základní a rozšířené závislosti
- ▶ např. pro **češtinu** – www.universaldependencies.org/cs/
- ▶ **cíl instrukcí** – **sjednocení** anotací napříč jazyky
- ▶ obsahuje i instrukce **netypické** pro daný jazyk – např. v češtině značkování některých zájmen jako determiner nebo expandování slov – *kdybych* = *když* + *bych*

Využití Universal Dependencies

- ▶ **srovnání** lingvistických fenomenů **napříč jazyky**
- ▶ **testování** syntaktické analýzy na různých jazycích
- ▶ **vícejazyčná syntaktická analýza** – paralelní dokumenty
- ▶ snadné **porozumění** rozdílům v anotacích

UD poskytuje **univerzální nástroje** pro:

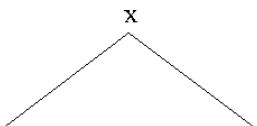
- ▶ **anotace** (editor, statistiky, validace)
- ▶ **vizualizace**
- ▶ **dotazování**
- ▶ **UDPipe** – trénování a automatické anotace



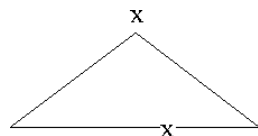
Stromové gramatiky TAG a LTAG

- ▶ Tree Adjoining Grammar – Joshi, Levy a Takahashi: *TAG Formalism*, 1975
- ▶ Lexicalized TAG – Joshi a Schabes: *Parsing with Lexicalized TAG*, 1991
- ▶ pracují přímo se **stromy** a ne s řetězcí slov
- ▶ množina **počátečních stromů** – základní stavební prvky
- ▶ složitější věty odvozovány s použitím **pomocných stromů**

počáteční (*initial*) strom:



pomocný (*auxiliary*) strom:



TAG – počáteční a pomocné stromy

- ▶ **počáteční stromy** – neobsahují rekurzi → popisují složkovou strukturu jednoduchých vět, jmenných skupin, předložkových skupin, ...
 1. všechny **nelistové uzly** odpovídají *neterminálům*
 2. všechny **listové uzly** odpovídají *terminálům* nebo *neterminálním* uzlům určeným k *substituci*

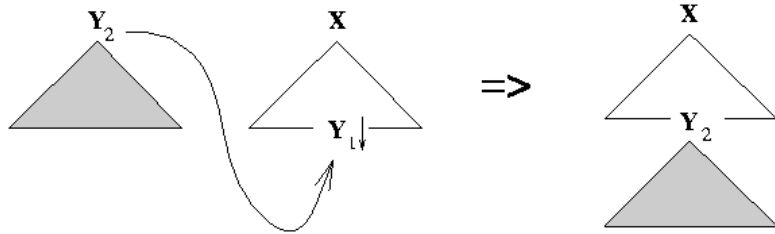
počáteční strom typu X = jeho kořen je označen termem X

- ▶ **pomocné stromy** – reprezentují *rekurzivní struktury* popisují větné členy, které se **připojují** k základním strukturám (např. příslovečné určení)
 - ▶ charakterizace:
 1. všechny **nelistové uzly** odpovídají *neterminálům*
 2. všechny **listové uzly** odpovídají *terminálům* nebo *neterminálním* uzlům určeným k *substituci* kromě právě jednoho neterminálního uzlu (**patový uzel**, *foot node*)
 3. **patový uzel** má stejné označení jako kořenový uzel
- patový uzel – slouží k připojení stromu k jinému uzlu

TAG – operace

dvě operace – **substituce** a **připojení** (*adjunction*)

operace **substituce** – nahrazuje označený neterminál v listech nějakého stromu stromem, jehož kořen nese stejné označení



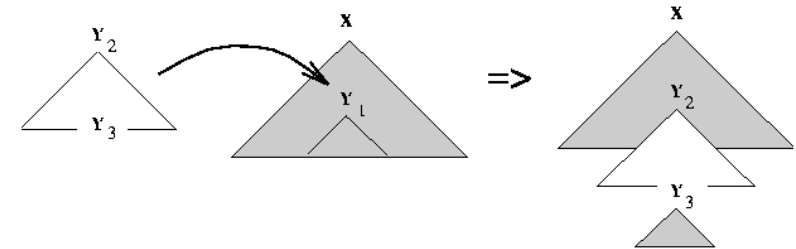
$Y_1 \downarrow$ – označený pro substituci

Definice TAG

- ▶ **TAG** $G = (I, A, S)$ je:
 - množina I konečných počátečních stromů
 - množina A pomocných stromů
 - typ stromu S – neterminál označující větu
- ▶ **množina stromů** $\mathcal{T}(G)$ TA gramatiky $G =$ množina všech stromů odvoditelných z počátečních stromů typu S z I , jejichž spodní okraj sestává čistě z terminálních uzlů (všechny substituční uzly byly doplněny)
- ▶ **jazyk řetězců** $\mathcal{L}(G)$ generovaných TA gramatikou $G =$ množina všech terminálních řetězců na spodním okraji stromů v $\mathcal{T}(G)$.

TAG – operace připojení

operace **připojení** – vložení pomocného stromu, popisujícího rekursi neterminálu X , se stromem, který obsahuje uzel označený rovněž X

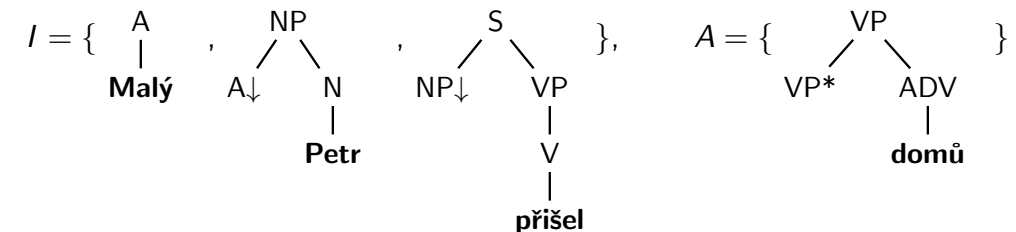


LTAG – lexikalizace

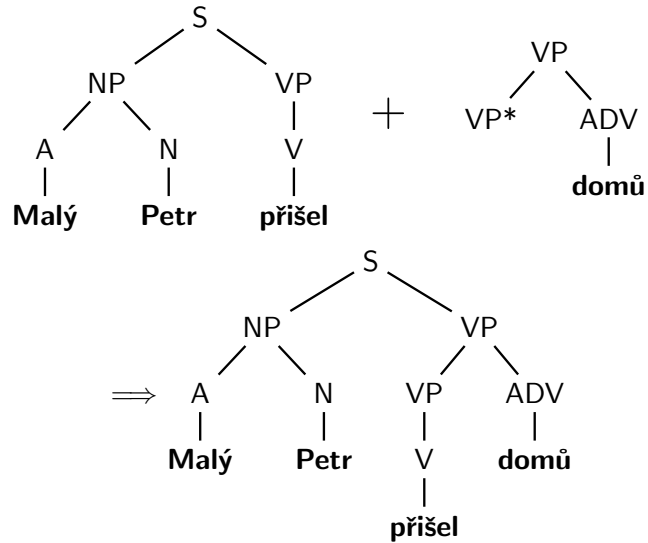
LTAG je **lexikalizovanou variantou** formalismu TAG

→ počáteční i pomocné stromy obsahují v listech jednu nebo více tzv. **lexikálních kotev** – uzly, které jsou přiřazeny (ukotveny) k určitým slovům lexikonu

lexikalizované stromy (*substituční uzly* – \downarrow , *patové uzly* – $*$):



LTAG – lexikalizované připojení



Lexikální funkční gramatiky LFG

- ▶ LFG, *Lexical Functional Grammar* – Kaplan a Bresnan, 1982
- ▶ dva typy syntaktických struktur
 - **vnější, c-struktura** – viditelná hierarchická organizace slov do frází
 - **vnitřní, f-struktura** – abstraktnější struktura gramatických funkcí, které tvoří hierarchii komplexních funkčních struktur
- důvod:
 - různé přirozené jazyky se významným způsobem odlišují v **organizaci fráze**, v pořadí a způsobech realizace gramatických funkcí
 - abstraktnější, **funkcionální** organizace jazyků se odlišuje mnohem méně v mnoha jazycích se např. objevují gramatické funkce *podmětu*, *předmětu* atd.

TAG a LTAG – generované jazyky

- ▶ díky použití operace připojení mají TAG a LTAG **větší generativní sílu** než bezkontextové gramatiky ($CFG \subset MCSL$) → generují **mírně kontextové jazyky** (*mildly context-sensitive languages*)
- MCSL:
 - vlastnost **konstantního růstu** – pokud uspořádáme řetězce jazyka vzestupně podle délky, potom rozdíl dvou po sobě jdoucích délek nemůže být libovolný (každá délka je lineární kombinací konečného počtu pevných délek).
 - analyzovatelnost v **polynomiálním čase** $O(n^6)$ vzhledem k délce vstupu
- ▶ i jiné formalismy umí MCSL (jsou ekvivalentní s (L)TAG):
 - LIG, *Linear Indexed Grammars* – Gazdar, 1985
 - HG, *Head Grammars* – Pollard, 1984
 - CCG, kombinatorické kategoriální gramatiky

The XTAG Project – <http://www.cis.upenn.edu/~xtag/>

Lexikální funkční gramatiky LFG – pokrač.

- ▶ **L** = vztahy mezi jazykovými formami, např. mezi aktivními a pasivními formami slovesa, jsou zobecněním struktury **lexikonu**, ne transformačními operacemi, derivujícími jednu formu z druhé
- ▶ **F = funkcionální teorie** – gramatické vztahy, jako je podmět, předmět atd., jsou základními konstrukty, a nejsou definovány pomocí konfigurace frázové struktury, nebo sémantických pojmů typu Agent a Patient
- ▶ v **LFG** – pro reprezentaci **funkcionální syntaktické informace** je vhodné definovat hierarchickou strukturu jazykových jednotek, avšak *vynucená linearizace* pořádku těchto struktur *není vhodná*

Syntaktické úrovně LFG

- ▶ dvě syntaktické úrovně:
 - **složková struktura** (*c-structure, constituent structure*) – zachycuje frázovou dominanci a prioritu a je reprezentována jako **strom** frázové struktury (CFG strom)
 - **funkcionální struktura** (*f-structure*) – zachycuje syntaktickou strukturu typu predikát-argumenty a je reprezentována *maticí dvojic atribut-hodnota* nabízí jednotnou reprezentaci syntaktické informace abstrahující od detailů struktury fráze a lineárního pořádku
- f-struktura obsahuje soubor atributů:
 - **příznaky** – čas, rod, číslo, ...
 - **funkce** – PRED, SUBJ, OBJ, jejichž hodnoty mohou být jiné f-struktury
- ▶ vztah mezi c-strukturami (stromy) a odpovídajícími f-strukturami:

projekce $\phi : \{\text{uzly stromu c-struktury}\} \rightarrow \{\text{f-struktury}\}$

vyjádřená **funkčními schémata**

LFG – pravidla

příklady:

$$S \rightarrow \text{NP} \quad \text{VP}$$

$$(\uparrow \text{SUBJ}) = \downarrow \quad \uparrow = \downarrow$$

$$VP \rightarrow \text{V} \quad (\text{NP})$$

$$\uparrow = \downarrow \quad (\uparrow \text{OBJ}) = \downarrow$$

$$NP \rightarrow (\text{DET}) \quad \text{N}$$

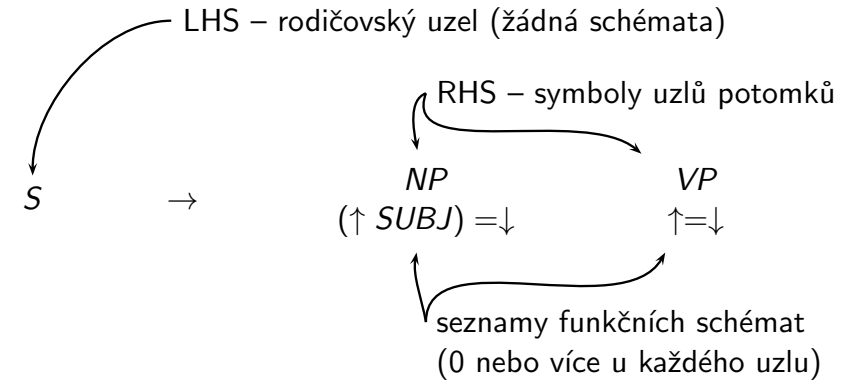
$$\uparrow = \downarrow \quad \uparrow = \downarrow$$

výrazy $(\uparrow \text{SUBJ}) = \downarrow$, $\uparrow = \downarrow$ a $(\uparrow \text{OBJ}) = \downarrow$ jsou *funkční schémata*

LFG – c-struktura

LFG pravidla:

- ▶ klasická CF pravidla
- ▶ plus **funkční schémata** – výrazy pracující se symboly na pravé straně pravidel (za \rightarrow , RHS)

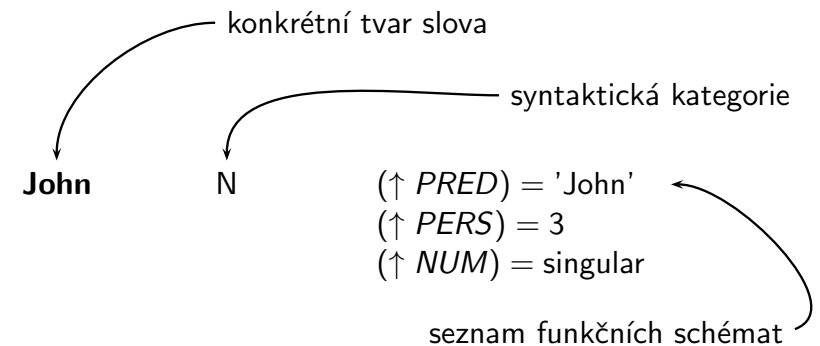


LFG – lexikon

lexikon také obsahuje **funkční schémata**

položka lexikonu:

1. konkrétní tvar slova
2. syntaktickou kategorii
3. seznam funkčních schémat



LFG – lexikon – pokrač.

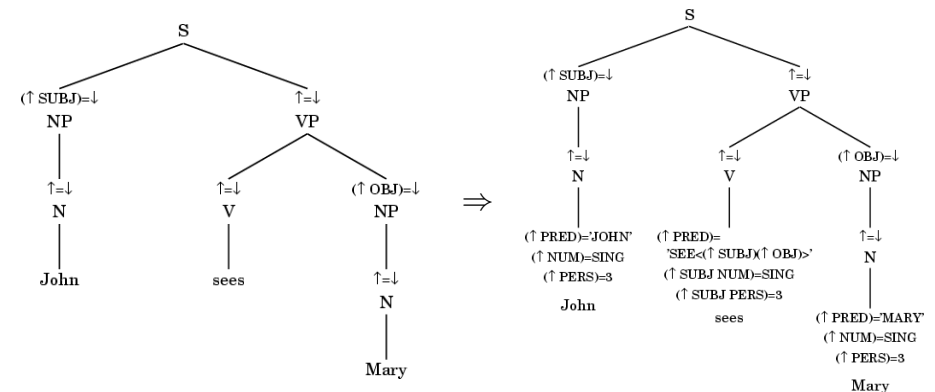
příklady:

John	N	(↑ PRED)	=	'JOHN'
		(↑ NUM)	=	SING
		(↑ PERS)	=	3
sees	V	(↑ PRED)	=	'SEE<(↑SUBJ)(↑OBJ)>'
		(↑ SUBJ NUM)	=	SING
		(↑ SUBJ PERS)	=	3
Mary	N	(↑ PRED)	=	'MARY'
		(↑ NUM)	=	SING
		(↑ PERS)	=	3

LFG – konstrukce c-struktury

informace v c-struktuře:

- ▶ hierarchická struktura větných členů
- ▶ funkční anotace (funkční schémata převedená do stromu) – po jejich interpretaci získáme výslednou f-strukturu



LFG – f-struktura

$$f_n \left[\begin{array}{c} A \\ F \\ H \end{array} \right] \quad f_m \left[\begin{array}{cc} B & C \\ D & E \\ G \\ I \end{array} \right]$$

grafický zápis:

matice atribut-hodnota (*attribute-value matrix*, AVM) – levé sloupce jsou atributy, pravé sloupce hodnoty (symbols, podřazené f-struktury nebo sémantické formy)

funkční rovnice a f-struktury:

$$(f_p \text{ ATT}) = \text{VAL}$$

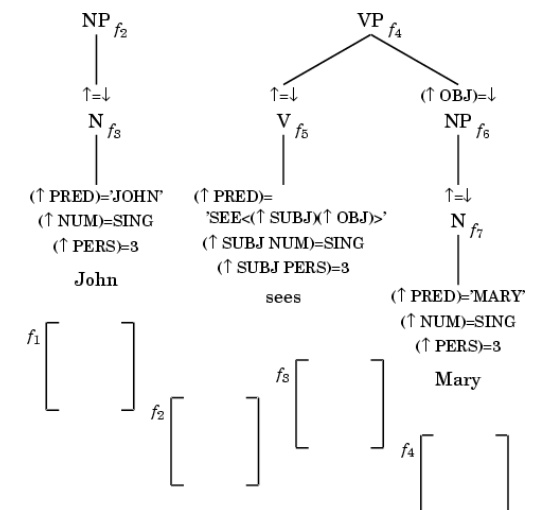
v f-struktuře f_p je řádek, kdeatribut je **ATT**a jeho hodnota je **VAL**funkční rovnice mohou být **splněny** nebo **nesplněny** (*true/false*)

LFG – instancie hodnot

Instancie hodnot

1. doplňuje hodnoty metaproměnných \uparrow a \downarrow
2. transformuje schémata na **funkční rovnice** – výrazy získané z f-struktur

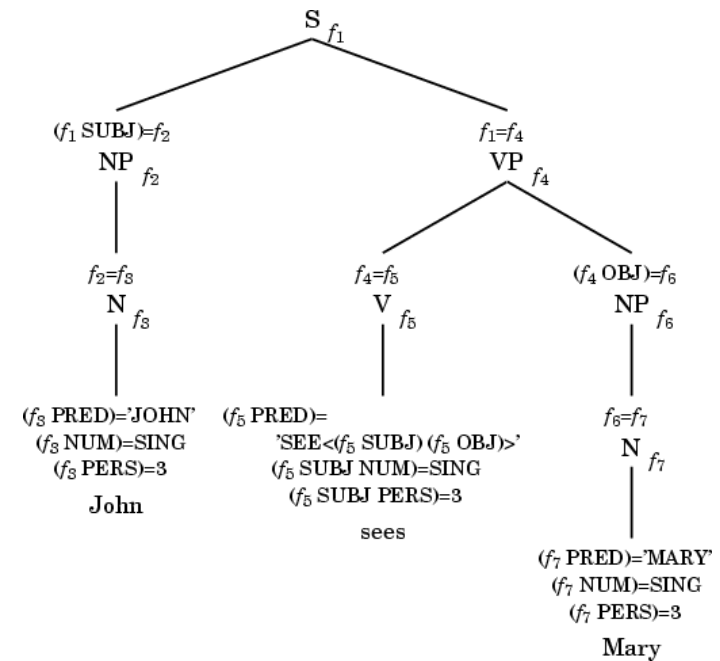
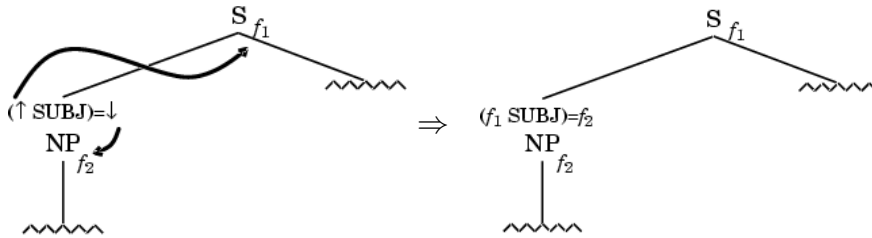
grafický zápis – f-struktura v hranatých závorkách []

každý uzel c-struktury má k sobě připojenou matici f-struktury, které se označují indexy f_i 

LFG – doplnění hodnot metaproměnných

↑ a ↓ (**metaproměnné**) se odkazují na f-struktury
je potřeba najít správné proměnné f_i na místa šipek

- ▶ ↓ – metaproměnná **EGO** nebo **SELF** – odkazuje na f-strukturu uzlu nad schématem
- ▶ ↑ – metaproměnná **MOTHER** – odkazuje na f-strukturu rodičovského uzlu vzhledem k uzlu nad schématem



LFG – funkční popis

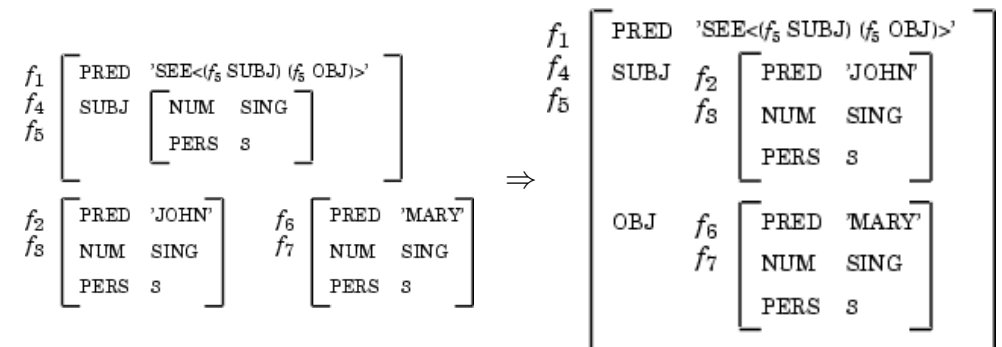
funkční popis = množina všech instanciovaných funkčních rovnic stromu vlastní konstrukce f-struktury pracuje pouze s tímto funkčním popisem
funkční popis předchází větě:

- | | |
|---|---|
| a. $(f_1 \text{ SUBJ}) = f_2$ | i. $(f_5 \text{ SUBJ NUM}) = \text{SING}$ |
| b. $f_3 = f_2$ | j. $(f_5 \text{ SUBJ PERS}) = f_3$ |
| c. $(f_3 \text{ PRED}) = \text{'JOHN'}$ | k. $(f_4 \text{ OBJ}) = f_6$ |
| d. $(f_3 \text{ NUM}) = \text{SING}$ | l. $f_6 = f_7$ |
| e. $(f_3 \text{ PERS}) = f_3$ | m. $(f_7 \text{ PRED}) = \text{'MARY'}$ |
| f. $f_1 = f_4$ | n. $(f_7 \text{ NUM}) = \text{SING}$ |
| g. $f_4 = f_5$ | o. $(f_7 \text{ PERS}) = f_3$ |
| h. $(f_5 \text{ PRED}) = \text{'SEE<(f_5 SUBJ)(f_5 OBJ)>'}$ | |

LFG – konstrukce f-struktury

f-struktura se tvoří z **funkčního popisu** tak, aby všechny funkční rovnice byly **splněny**

výsledná f-struktura musí být **minimální** taková f-struktura



XLE web interface – <http://pargram.b.uib.no/tools/>,
<http://www.xlfg.org/>