

Gramatické formalismy pro ZPJ

Aleš Horák

E-mail: hales@fi.muni.cz

http://nlp.fi.muni.cz/poc_lingv/

Obsah:

- Gramatické formalismy
- Kategoriální gramatiky
- Závislostní gramatiky
- Stromové gramatiky TAG a LTAG

Gramatické formalismy

- existuje množství různých přístupů k formální specifikaci gramatik (přirozených jazyků), různé **gramatické formalismy**
- popíšeme několik nejrozšířenějších formalismů:
 - *kategoriální 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)

Gramatické formalismy

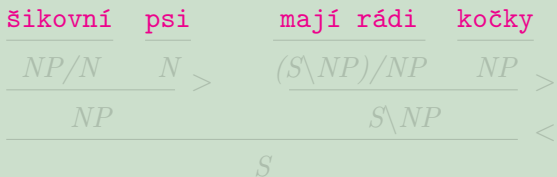
- existuje množství různých přístupů k formální specifikaci gramatik (přirozených jazyků), různé **gramatické formalismy**
- popíšeme několik nejrozšířenějších formalismů:
 - *kategoriální gramatiky* – categorial grammars, CG
 - *závislostní gramatiky* – dependency grammars
 - *stromové gramatiky* – (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)

Gramatické formalismy

- existuje množství různých přístupů k formální specifikaci gramatik (přirozených jazyků), různé **gramatické formalismy**
- popíšeme několik nejrozšířenějších formalismů:
 - *kategoriální 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)

Kategoriální gramatiky

- existuje několik různých variant notace

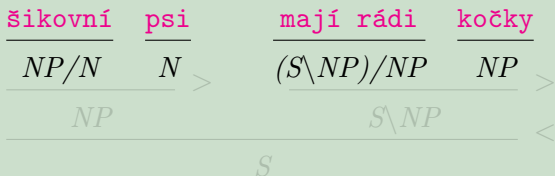


- jiný rozšířený zápis – *výsledek na vrcholku* (result on top) Lambek 1958

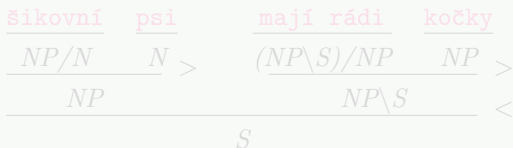


Kategoriální gramatiky

- existuje několik různých variant notace

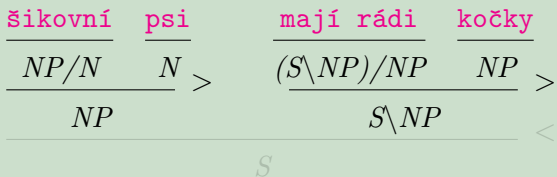


- jiný rozšířený zápis – **výsledek na vrcholku** (result on top) Lambek 1958

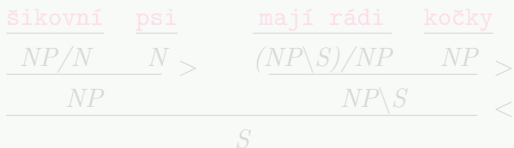


Kategoriální gramatiky

- existuje několik různých variant notace



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



Kategoriální gramatiky

- existuje několik různých variant notace

$$\begin{array}{c}
 \underline{\text{šikovní}} \quad \underline{\text{psi}} \quad \quad \underline{\text{mají rádi}} \quad \underline{\text{kočky}} \\
 \underline{NP/N} \quad N > \quad \underline{(S \setminus NP)/NP} \quad NP > \\
 \underline{NP} \quad \quad \quad \quad \quad \underline{S \setminus NP} < \\
 \hline
 S
 \end{array}$$

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

$$\begin{array}{c}
 \underline{\text{šikovní}} \quad \underline{\text{psi}} \quad \quad \underline{\text{mají rádi}} \quad \underline{\text{kočky}} \\
 \underline{NP/N} \quad N > \quad \underline{(NP \setminus S)/NP} \quad NP > \\
 \underline{NP} \quad \quad \quad \quad \quad \underline{NP \setminus S} < \\
 \hline
 S
 \end{array}$$

Kategoriální gramatiky

- existuje několik různých variant notace

$$\begin{array}{c}
 \underline{\text{šikovní}} \quad \underline{\text{psi}} \quad \quad \underline{\text{mají rádi}} \quad \underline{\text{kočky}} \\
 \underline{NP/N} \quad N \quad > \quad \underline{(S \setminus NP)/NP} \quad NP \quad > \\
 \underline{NP} \quad \quad \quad \quad \quad \quad \quad \underline{S \setminus NP} \quad < \\
 \hline
 S
 \end{array}$$

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

$$\begin{array}{c}
 \underline{\text{šikovní}} \quad \underline{\text{psi}} \quad \quad \underline{\text{mají rádi}} \quad \underline{\text{kočky}} \\
 \underline{NP/N} \quad N \quad > \quad \underline{(NP \setminus S)/NP} \quad NP \quad > \\
 \underline{NP} \quad \quad \quad \quad \quad \quad \quad \underline{NP \setminus S} \quad < \\
 \hline
 S
 \end{array}$$

Kategoriální gramatiky

- **kategoriální 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 kategoriálních gramatik pro **popis přirozeného jazyka** – Jehošua Bar-Hillel, 1953

Kategoriální gramatiky

- **kategoriální 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 kategoriálních gramatik pro **popis přirozeného jazyka** – Jehošua Bar-Hillel, 1953

Kategoriální gramatiky

- **kategoriální 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 kategoriálních gramatik pro **popis přirozeného jazyka** – Jehošua Bar-Hillel, 1953

Kategoriální gramatiky

- **kategoriální 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 kategoriálních gramatik pro **popis přirozeného jazyka** – Jehošua Bar-Hillel, 1953

Kategoriální gramatiky

- **kategoriální 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 kategoriálních gramatik pro **popis přirozeného jazyka** – Jehošua Bar-Hillel, 1953

Notace kategoriálních gramatik

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

Notace kategoriálních gramatik

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**

Notace kategoriálních gramatik

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

Notace kategoriálních gramatik

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**

Notace kategoriálních gramatik

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

Notace kategoriálních gramatik

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**

Notace kategoriálních gramatik – pokrač.

- daná schémata umožňují 2 způsoby kombinace:
 - argument **vpravo** ($/$) – $\alpha_{(X/Y)} \circ \beta_{(Y)} \rightarrow \alpha\beta_{(X)}$
 - argument **vlevo** (\backslash) – $\beta_{(Y)} \circ \alpha_{(X \backslash Y)} \rightarrow \beta\alpha_{(X)}$
- tento typ kategoriální gramatiky označoval Bar-Hillel jako **obousměrný** (bidirectional CG)
- Karel miluje Marii:
 - bázové kategorie = $\{NP, S\}$
 - kategorie z lexikonu: $Karel_{(NP)}$, $Marii_{(NP)}$, $miluje_{((S \backslash NP)/NP)}$
 - $C_{complete} = \{S\}$
- 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

Notace kategoriálních gramatik – pokrač.

- daná schémata umožňují 2 způsoby kombinace:
 - argument **vpravo** ($/$) – $\alpha_{(X/Y)} \circ \beta_{(Y)} \rightarrow \alpha\beta_{(X)}$
 - argument **vlevo** (\backslash) – $\beta_{(Y)} \circ \alpha_{(X \backslash Y)} \rightarrow \beta\alpha_{(X)}$
- tento typ kategoriální gramatiky označoval Bar-Hillel jako **obousměrný** (bidirectional CG)

- Karel miluje Marii:

- bázové kategorie = $\{NP, S\}$
- kategorie z lexikonu: $Karel_{(NP)}$, $Marii_{(NP)}$, $miluje_{((S \backslash NP)/NP)}$
- $C_{complete} = \{S\}$

- 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

Notace kategoriálních gramatik – pokrač.

- daná schémata umožňují 2 způsoby kombinace:
 - argument **vpravo** ($/$) – $\alpha_{(X/Y)} \circ \beta_{(Y)} \rightarrow \alpha\beta_{(X)}$
 - argument **vlevo** (\backslash) – $\beta_{(Y)} \circ \alpha_{(X \backslash Y)} \rightarrow \beta\alpha_{(X)}$
- tento typ kategoriální gramatiky označoval Bar-Hillel jako **obousměrný** (bidirectional CG)
- Karel miluje Marii:
 - báze kategorie = $\{NP, S\}$
 - kategorie z lexikonu: $Karel_{(NP)}, Marii_{(NP)}, miluje_{((S \backslash NP)/NP)}$
 - $C_{complete} = \{S\}$
- 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

Notace kategoriálních gramatik – pokrač.

- daná schémata umožňují 2 způsoby kombinace:
 - argument **vpravo** ($/$) – $\alpha(x/y) \circ \beta(y) \rightarrow \alpha\beta(x)$
 - argument **vlevo** (\backslash) – $\beta(y) \circ \alpha(x \backslash y) \rightarrow \beta\alpha(x)$
- tento typ kategoriální gramatiky označoval Bar-Hillel jako **obousměrný** (bidirectional CG)
- Karel miluje Marii:
 - bázové kategorie = $\{NP, S\}$
 - kategorie z lexikonu: $Karel_{(NP)}$, $Marii_{(NP)}$, $miluje_{((S \backslash NP)/NP)}$
 - $C_{complete} = \{S\}$
- 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

Notace kategoriálních gramatik – pokrač.

- daná schémata umožňují 2 způsoby kombinace:
 - argument **vpravo** ($/$) – $\alpha_{(X/Y)} \circ \beta_{(Y)} \rightarrow \alpha\beta_{(X)}$
 - argument **vlevo** (\backslash) – $\beta_{(Y)} \circ \alpha_{(X \backslash Y)} \rightarrow \beta\alpha_{(X)}$
- tento typ kategoriální gramatiky označoval Bar-Hillel jako **obousměrný** (bidirectional CG)
- Karel miluje Marii:
 - báze kategorie = $\{NP, S\}$
 - kategorie z lexikonu: $Karel_{(NP)}$, $Marii_{(NP)}$, $miluje_{((S \backslash NP)/NP)}$
 - $C_{complete} = \{S\}$
- 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

Rozšíření kategoriálních gramatik

- klíčový problém – nespojité 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 nejpropracovanějším 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/>

Rozšíření kategoriálních gramatik

- klíčový problém – nespojité větné části, tzv. **neprojektivity**
- řešení pomocí rozšíření CG – přídavné **kombinatorické operátory** založené na **typech**
- dva možné přístupy:
 - pravidlově orientovaný přídává pravidla odpovídající jednoduchým operacím nad kategoriemi, jako jsou:
 - wrap – komutace argumentů
 - type-raising – aplikace typů podobná aplikaci tradičních pádů na jmenné fráze
 - comp – kompozice funkcík nejpropracovanějším 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/>

Rozšíření kategoriálních gramatik

- klíčový problém – nespojité 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 nejpropracovanějším 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/>

Rozšíření kategoriálních gramatik

- klíčový problém – nespojité 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 nejpropracovanějším 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/>

Obsah

- 1 Gramatické formalismy
- 2 Kategoriální gramatiky
 - Notace kategoriálních gramatik
 - Rozšíření kategoriálních gramatik
- 3 Závislostní gramatiky
 - Universal Dependencies
- 4 Stromové gramatiky TAG a LTAG
 - operace
 - lexikalizace
 - generované jazyky

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

- 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
typický vztah mezi slovesem a jeho možnými doplněními:

```
nosit  
= koho|co  
= komu & koho|co
```


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
typický vztah mezi slovesem a jeho možnými doplněními:

```
nosit  
= koho|co  
= komu & koho|co
```

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 nejpropracovanějších 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

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 nejpropracovanějších 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

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 nejpropracovanějších 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

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 nejpropracovanějších 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

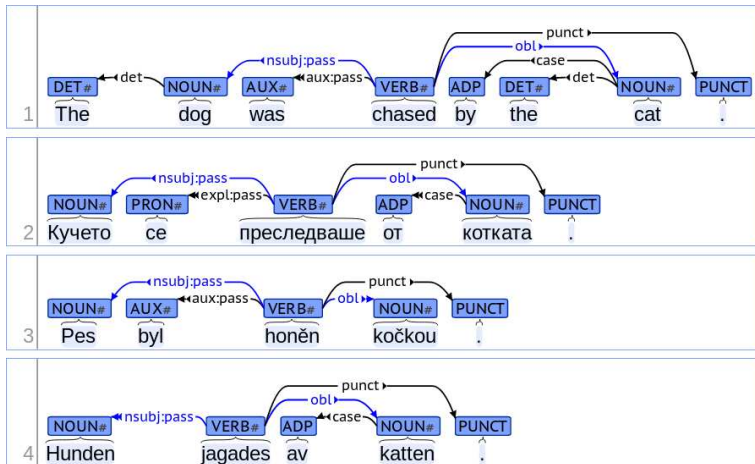
- www.universaldependencies.org, UD
- sjednocení závislostní anotace pro různé jazyky
- cca 200 stromových bank (*treebanks*) ve více než 100 jazycích

Universal Dependencies

- www.universaldependencies.org, UD
- sjednocení závislostní anotace pro různé jazyky
- cca 200 stromových bank (*treebanks*) ve více než 100 jazycích

Universal Dependencies

- www.universaldependencies.org, UD
- sjednocení závislostní anotace pro různé jazyky
- cca 200 stromových bank (*treebanks*) ve více než 100 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	

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	

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	

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

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

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

Universal Dependencies

1	Správkyňě	Správkyňě	NOUN	Case=Nom Gender=Fem Number=Sing Polarity=Pos
2	dědictví	dědictví	NOUN	Case=Gen Gender=Neut Number=Sing Polarity=Pos
3	Nováková	Nováková	PROPN	Case=Nom Gender=Fem NameType=Sur Number=Sing Polarity=Pos
4	označila	označit	VERB	Aspect=Perf Gender=Fem,Neut Number=Plur,Sing Polarity=Pos Tense=Past VerbForm=Part Voice=Act
5	pondělní	pondělní	ADJ	Case=Acc Degree=Pos Gender=Neut Number=Sing Polarity=Pos
6	rozhodnutí	rozhodnutí	NOUN	Case=Acc Gender=Neut Number=Sing Polarity=Pos
7	za	za	ADP	AdpType=Prep Case=Acc
8	potěšující	potěšující	ADJ	Aspect=Imp Case=Acc Gender=Neut Number=Sing Polarity=Pos Tense=Pres VerbForm=Part Voice=Act
9	.	.	PUNCT	-

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**

SLiDo

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**

SLiDo

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**

SLiDo

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**

SLiDo

Využití Universal Dependencies

- **rovnání** 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

Využití Universal Dependencies

- **rovnání** 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

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

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:

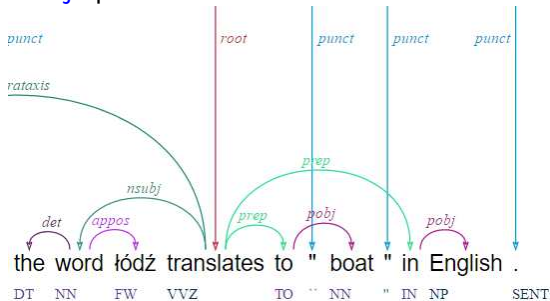
- anotace (editor, statistiky, validace)
- vizualizace
- dotazování
- UDPipe – trénování a automatické anotace

Využití Universal Dependencies

- **rovnání** 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

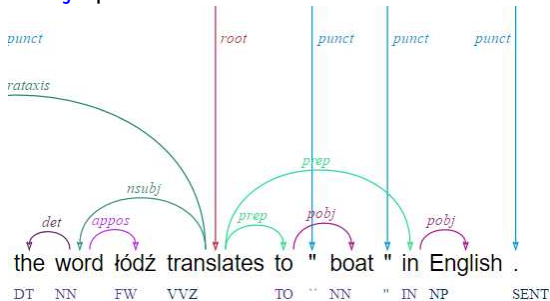


Využití Universal Dependencies

- **rovnání** 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

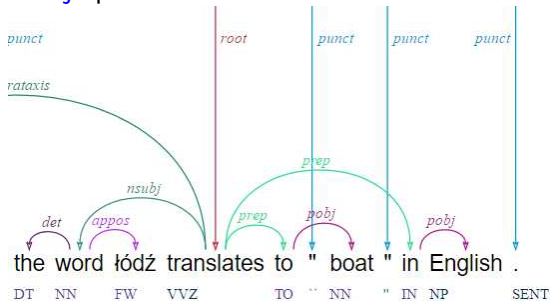


Využití Universal Dependencies

- **rovnání** 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

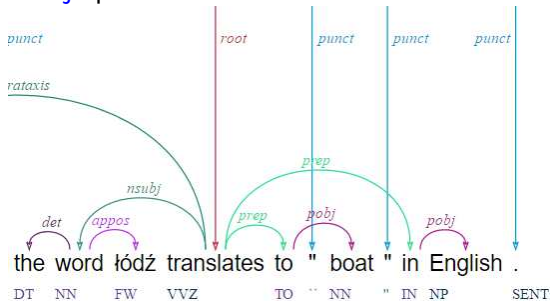


Využití Universal Dependencies

- **rovnání** 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



Obsah

- 1 Gramatické formalismy
- 2 Kategoriaální gramatiky
 - Notace kategoriaálních gramatik
 - Rozšíření kategoriaálních gramatik
- 3 Závislostní gramatiky
 - Universal Dependencies
- 4 Stromové gramatiky TAG a LTAG
 - operace
 - lexikalizace
 - generované jazyky

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:

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**

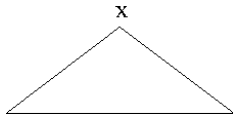
počáteční (*initial*) strom:

pomocný (*auxiliary*) strom:

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:

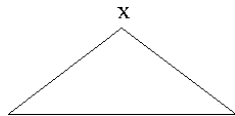


pomocný (*auxiliary*) strom:

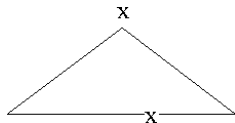
Stromové gramatiky TAG a LTAG

- 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:

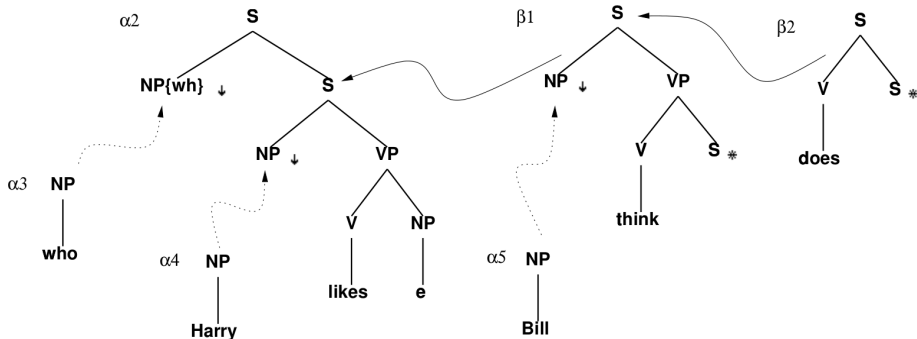


pomocný (*auxiliary*) strom:



XTAG Project

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



TuLiPA-frames

The image displays two windows from the TuLiPA software. The top window, titled "Derivation Trees + Derived Trees", shows a derivation tree for the sentence "Timm liebt Feigen". The root node is TVnp2-26-74--liebt--1. It branches into two children: TVnp2-26-75--liebt--1 (left) and TVnp2-26-73--liebt--1 (right). The left child branches into TNoun-0-6--Timm1--0, and the right child branches into TNoun-0-6--Feigen3--6. The bottom window, titled "TULIPA", is a configuration panel with the following settings:

- Mode: TT-MCTAG/TAG, RCG
- TT-MCTAG: Max LPA length: []
- TT-MCTAG/TAG: Show derivation steps in GUI, Dependency output
- Misc: Verbose mode, XML output, no GUI
- Grammar: /local/toy.xml [Browse]
- Lemmas: /local/lemma.xml [Browse]
- Morphological entries: /local/morph.xml [Browse]
- Output file: [Browse]
- Axiom: []
- Sentence: Timm liebt Feigen [Clear]
- [garse] [Quit]

At the bottom of the TuLiPA window, the following performance metrics are listed:

- Grammar conversion time: 0.572578 sec.
- Sentence "Timm liebt Feigen" parsed.
- Parsing time: 0.199693 sec.
- Forest extraction time: 0.010071 sec.
- Derivation trees extraction time: 0.129275 sec.
- Total parsing time for sentence "Timm liebt Feigen": 0.947014 sec.

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 – 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 – 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*

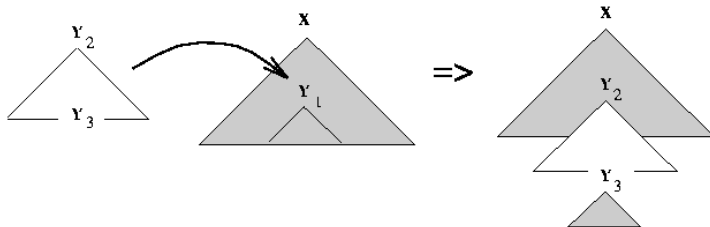
počáteční strom typu X = jeho kořen je označen termem X

- **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 připojení

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



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 $\mathcal{T}(G)$.

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 $\mathcal{T}(G)$.

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 $\mathcal{T}(G)$.

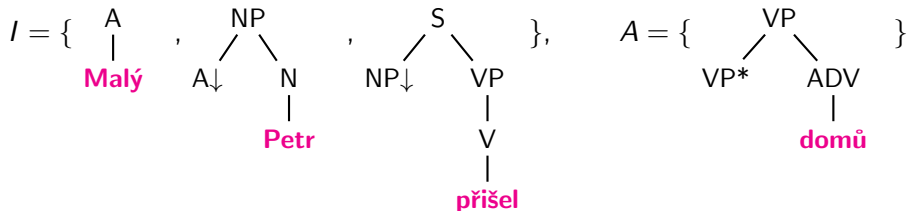
LTAG – lexikalizace

LTAG je **lexikalizovanou variantou** formalismu TAG

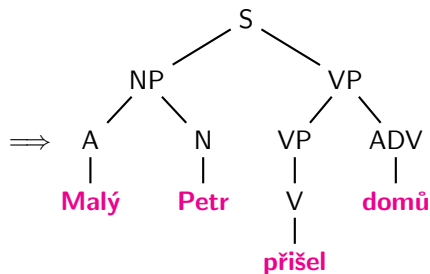
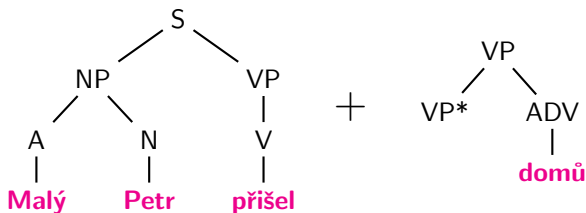
→ počáteční i pomocné stromy obsahují v listech jednu nebo více tzv.

lexikálních kotev – uzly, které jsou přiřazeny (ukotveny) k určitým slovům lexikonu

lexikalizované stromy (*substituční uzly* – ↓, *patové uzly* – *):



LTAG – lexikalizované připojení



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):
 - HG, *Head Grammars* – Pollard, 1984
 - CCG, kombinatorické kategoriální gramatiky

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

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

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