# Strojový překlad přes tektogramatickou rovinu

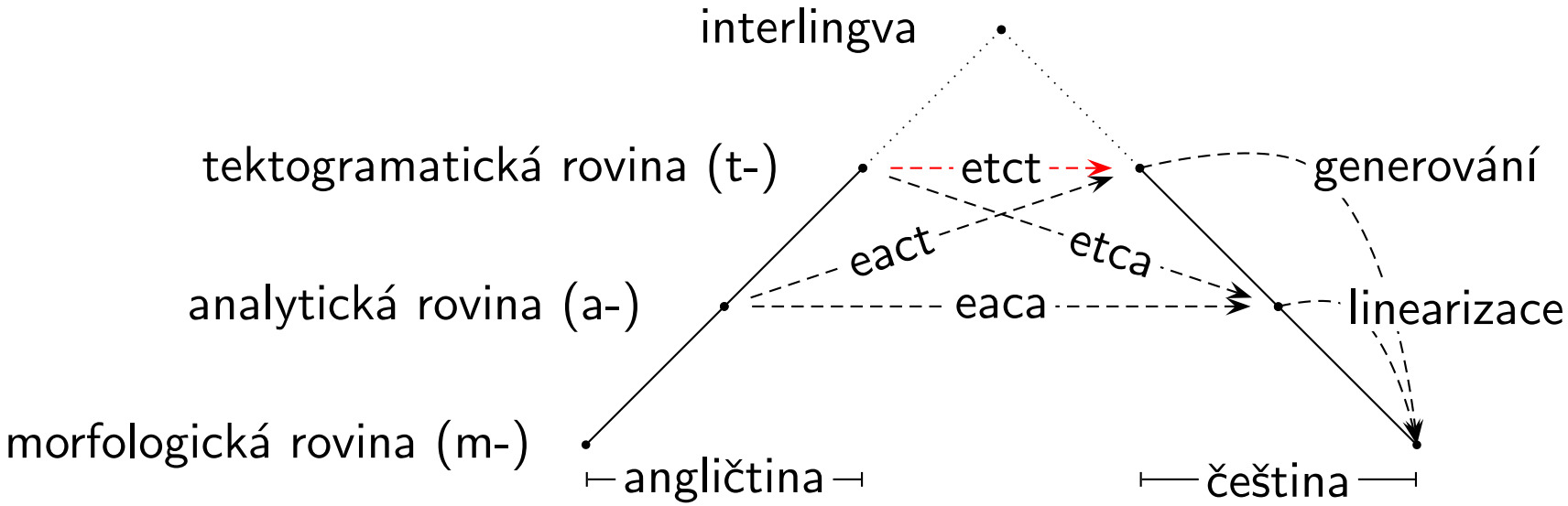
Ondřej Bojar
obo@cuni.cz

10. března 2008

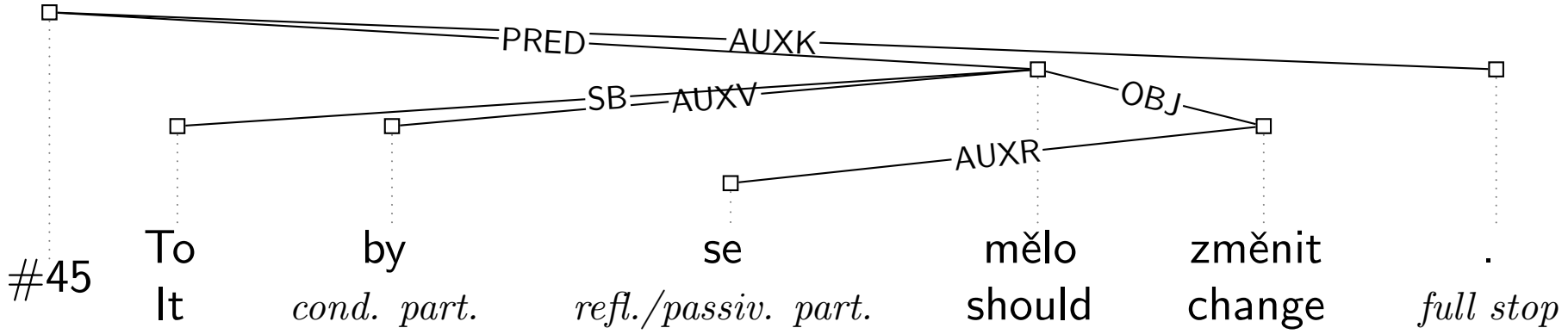
# Osnova

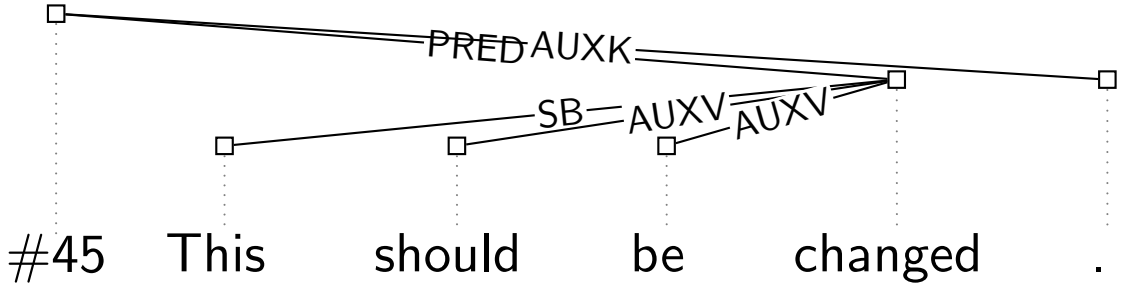
- Motivace pro překlad přes tektogramatickou rovinu.
- STSG – model pro převod stromu na strom.
- Problém ředění dat.
- Záchranné metody (konstrukce nových stromečků).
- Vyhodnocení pomocí BLEU.
- Diskuse.

# Varianty překladu se strukturním transferem

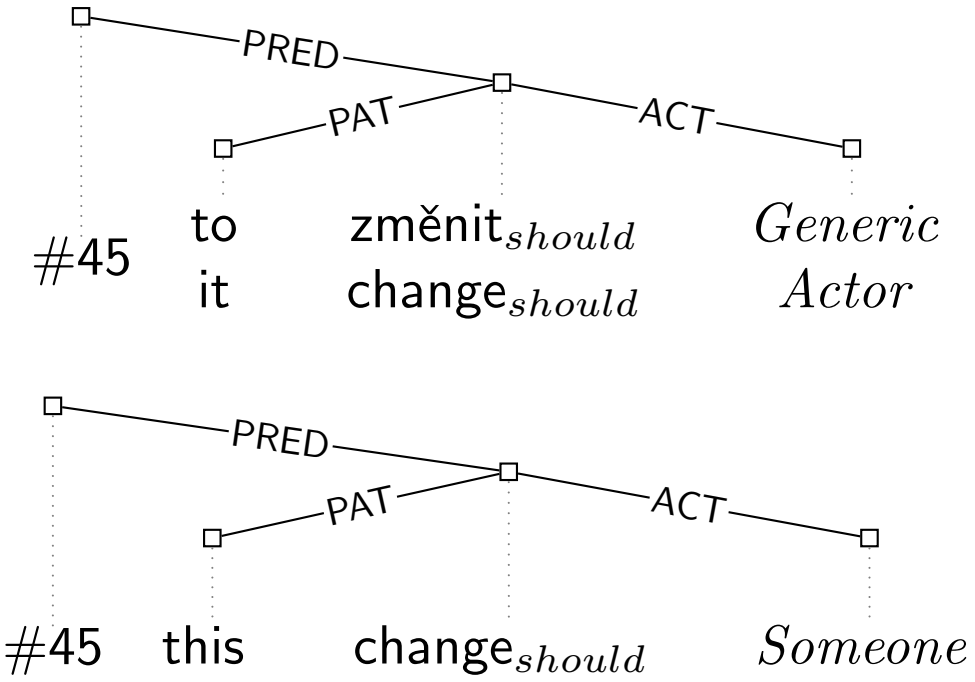


# Český a anglický analytický strom



# Český a anglický tektogramatický strom




Vyjadřuje predikáty a argumenty: change$_{\texttt{should}}$(ACT: someone, PAT: it)

# Motivace pro tektogramatický transfer

Transfer na t-rovině by běl být snazší než přímý překlad:

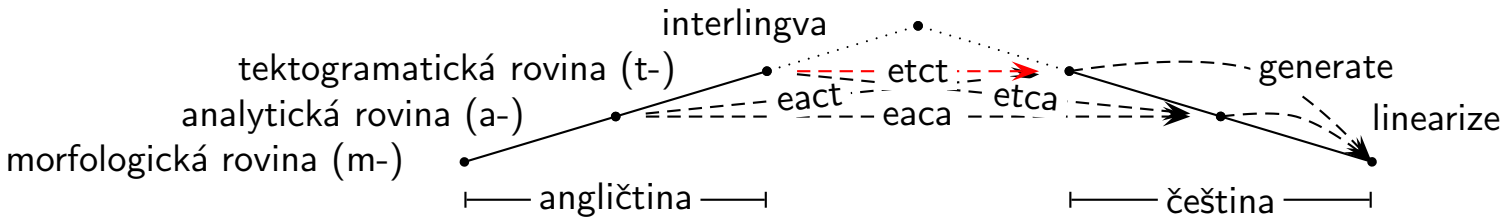
- Menší počet uzlů (pomocná slova skryta).
- Vhodnější (závislostní) kontext, nikoli kontext sousedství.
- Vyřešeny neprojektivní závislosti.
- Pořádek uzlů můžeme ponechat beze změn (vyjadřuje informační strukturu).
- Menší slovník (lemata místo forem).
- České a anglické t-stromy jsou si podobnější
  $\Rightarrow$ mohlo by stačit méně paralelních dat (ale více jednojazyčných).
- Možnost později zohlednit pokročilé rysy, např. koreferenci.

Drobné komplikace:

- 47 stran dokumentace formátu dat (PML)
- 1200 stran dokumentace české t-roviny

# Převod stromu na strom

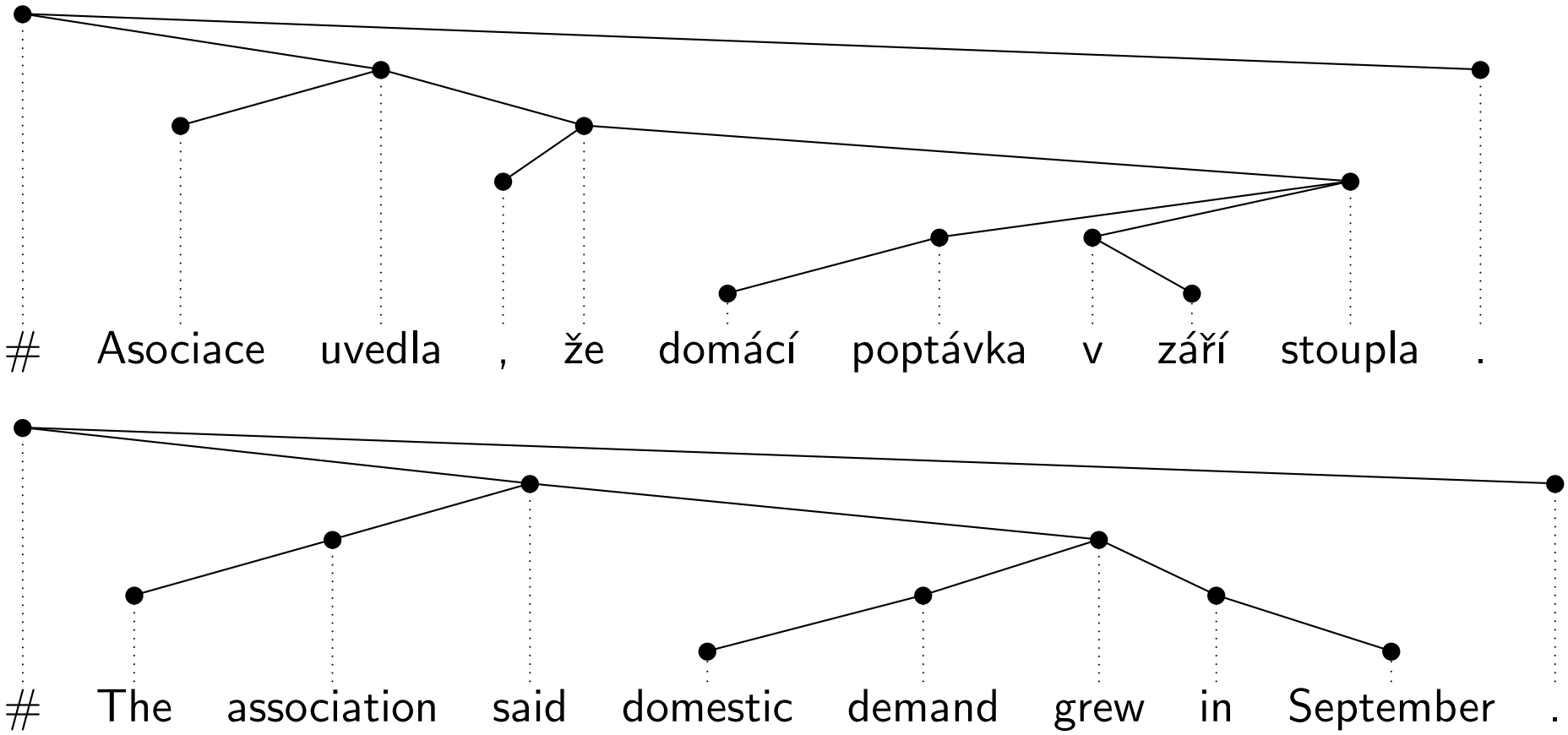
**Synchronous Tree Substitution Grammar** (Čmejrek, 2006).



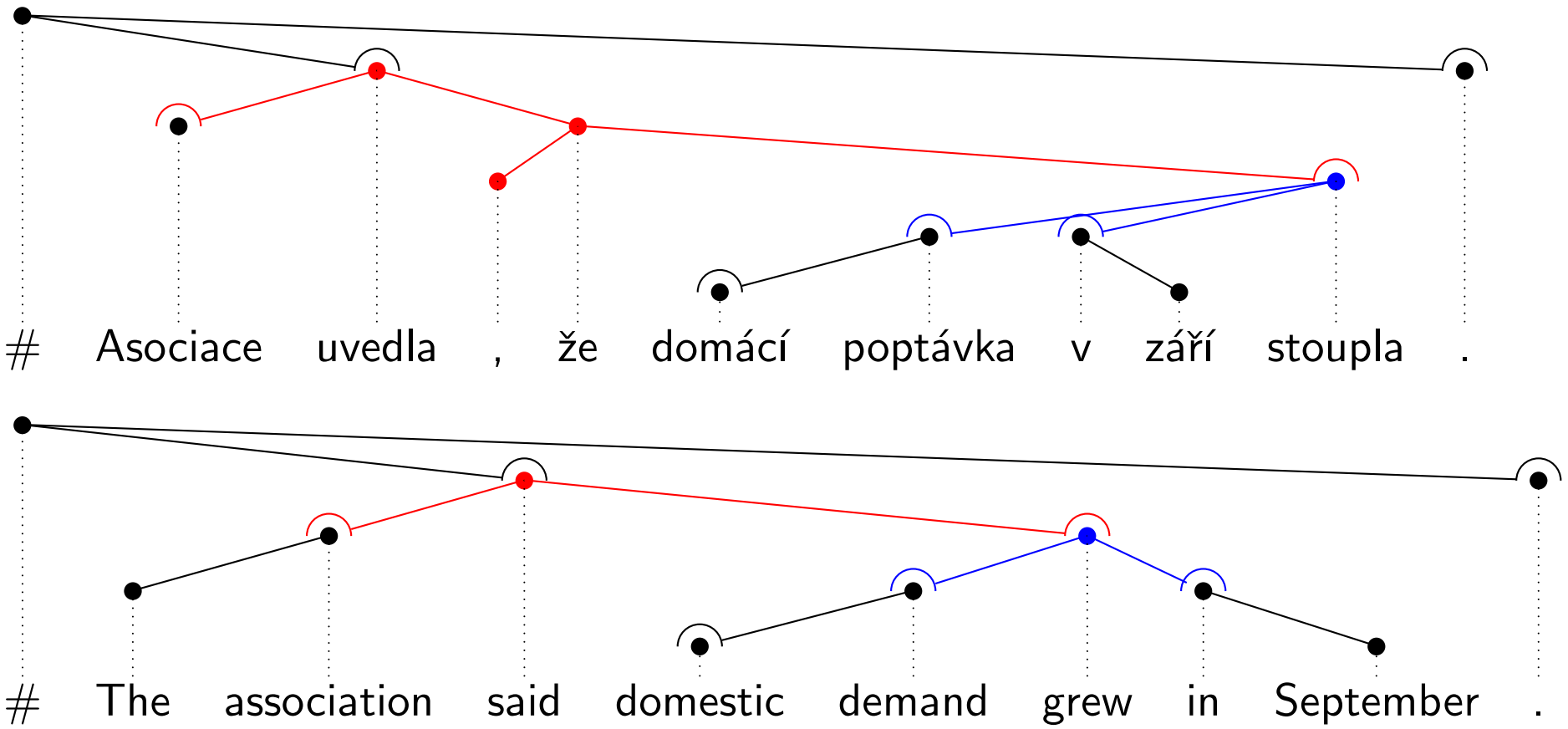

Vstupní závistlostní strom:

- rozlož na malé stromečky
- najdi překlady jednotlivých stromečků
- spoj přeložené stromečky dohromady
  - v případě a-stromů přečti uzly zleva doprava
  - v případě t-stromů použij generování (Ptáček, 2005)

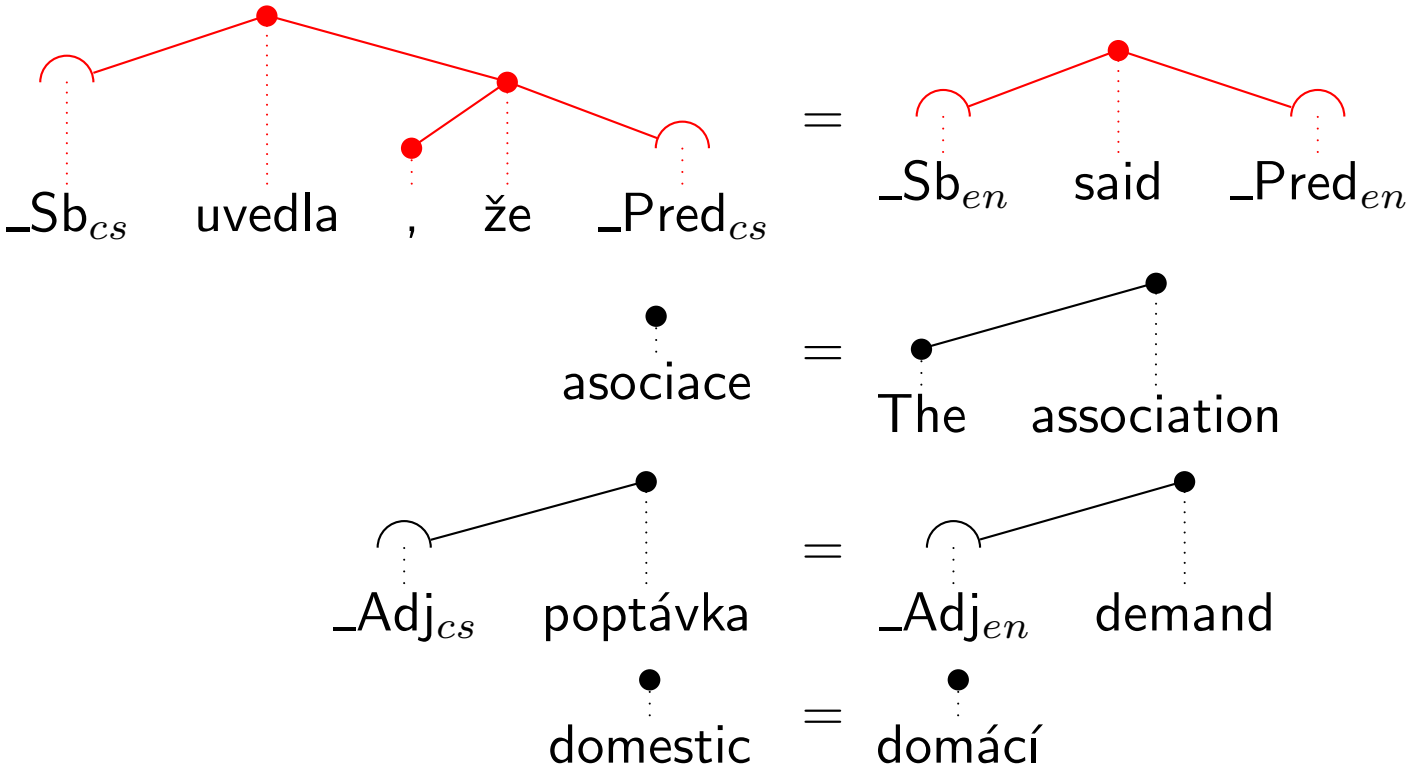
# Ilustrace: Dvojici analytických stromů. . .



# . . . rozložíme na stromečky. . .

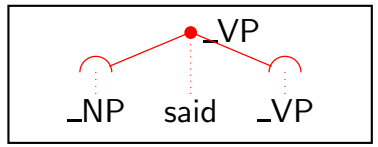


# . . . a sesbíráme překladový slovník stromečků.



# Stromečky formálně

Dána množina stavů $Q$ a množina slov (popisků uzlů) $L$. Definujeme:




STROMEČEK $t$ je šestice $(V, V^i, E, q, l, s)$, kde:

- $V$ je množina UZLŮ,
- $V^i \subseteq V$ je neprázdná množina VNITŘNÍCH UZLŮ (INTERNAL NODES). Zbývající uzly, $V^f = V \setminus V^i$ tvoří množinu SLOTŮ (FRONTIER NODES),
- $E \subseteq V^i \times V$ je množina orientovaných hran začínajících v interních uzlech a tvořící kořenový strom (souvislý orientovaný graf bez cyklů),
- $q \in Q$ je STAV KOŘENE,
- $l : V^i \rightarrow L$ je funkce přiřazující každému internímu uzlu slovo,
- $s : V^f \rightarrow Q$ je funkce přiřazující každému slotu stav.

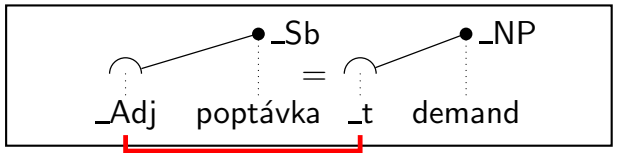
Navíc lze reprezentovat lokální (otec v řádce synů) nebo globální pořadí uzlů.

Na rozdíl od M. Čmejrka požaduji ve stromečku alespoň jeden vnitřní uzel.

# Dvojice stromečků, synchronní derivace

DVOJICE STROMEČKŮ $t_{1:2}$ je trojice $(t_1, t_2, m)$, kde:

- $t_1$ a $t_2$ jsou stromečky nad zdrojovou a cílovou množinou slov ($L_1$ a $L_2$) a stavů ($Q_1$ a $Q_2$),
- $m$ je PÁROVÁNÍ slotů v $t_1$ a $t_2$.




Na rozdíl od M. Čmejrka vyžaduji úplné párování, v důsledku čehož musí mít $t_1$ a $t_2$ stejný počet slotů.

Z počátečního SYNCHRONNÍHO STAVU $Start_{1:2} \in Q_1 \times Q_2$,
buduje SYNCHRONNÍ DERIVACE $\delta$ dvojici závilostních stromů takto:

- připojuje dvojice stromečků $t^0_{1:2}, \ldots, t^k_{1:2}$ do odpovídajících si dvojic slotů, a
- zajišťuje, aby stav kořene $q^0_{1:2}, \ldots, q^k_{1:2}$ dvojice stromečků $t^0_{1:2}, \ldots, t^k_{1:2}$ odpovídal stavu dvojice slotů, kam se stromečky připojují.

Definujeme pravděpodobnost derivace: $p(\delta) = p(t^0_{1:2}|Start_{1:2}) * \prod_{i=1}^{k} p(t^k_{1:2}|q^k_{1:2})$

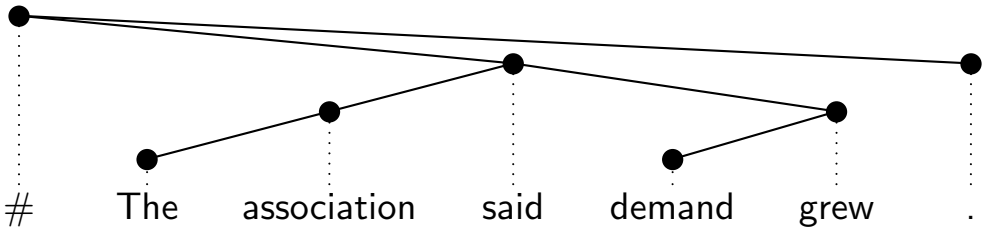
# Překlad (decoding) pomocí STSG

- Ke zdrojovému stromu hledáme dekompozici a cílový závislostní strom, aby jejich synchronní derivace $\delta$ měla maximální pravděpodobnost.
- Implementováno ve dvou krocích:

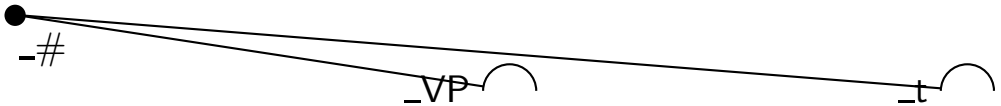
1. Příprava tabulky **možností překladu**:
   - Pro každý vstupní uzel studuji všechny stromečky, které zde mohou začínat.
   - Pokud ke zvolenému stromečku existuje cílový, našli jsme možnost překladu.
   - Uchováváme jen $\tau$ nejlepších možností překladu pro každý uzel.

2. Postupné **budování částečných hypotéz**:
   - Od kořene dolů zdrojový strom pokrýváme překladovými možnostmi.
   - Uchováváme jen $\sigma$ nejlepších částečných hypotéz dané velikosti (počet vstupních uzlů pokrytých vnitřními uzly)

# Ukázka možností překladu



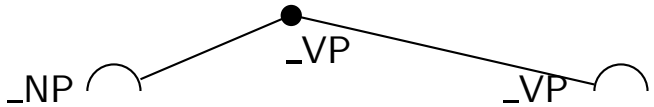

Možnosti překladu v kořeni:




⇒ # _Pred _AuxK


⇒ # _Pred .

Možnosti překladu v uzlu "said":

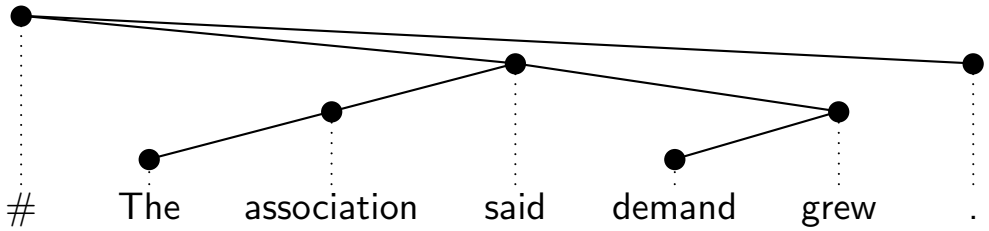



⇒ _Sb uvedla , že _Pred

Možnosti překladu v uzlu ".":

⇒ .

# Postupné budování hypotéz




Ukázková derivace:

Linearizovaný výstup:

$h_0$ _#

⇒ _#

$h_1$ _# _VP

⇒ # _Pred .

$h_2$ _# _NP _VP

⇒ # _Sb uvedla , že _Pred .

$h_3$ _# _NP _NP

⇒ # _Sb uvedla , že _Sb stoupla .

# Překladový slovník stromečků z paralelního treebanku

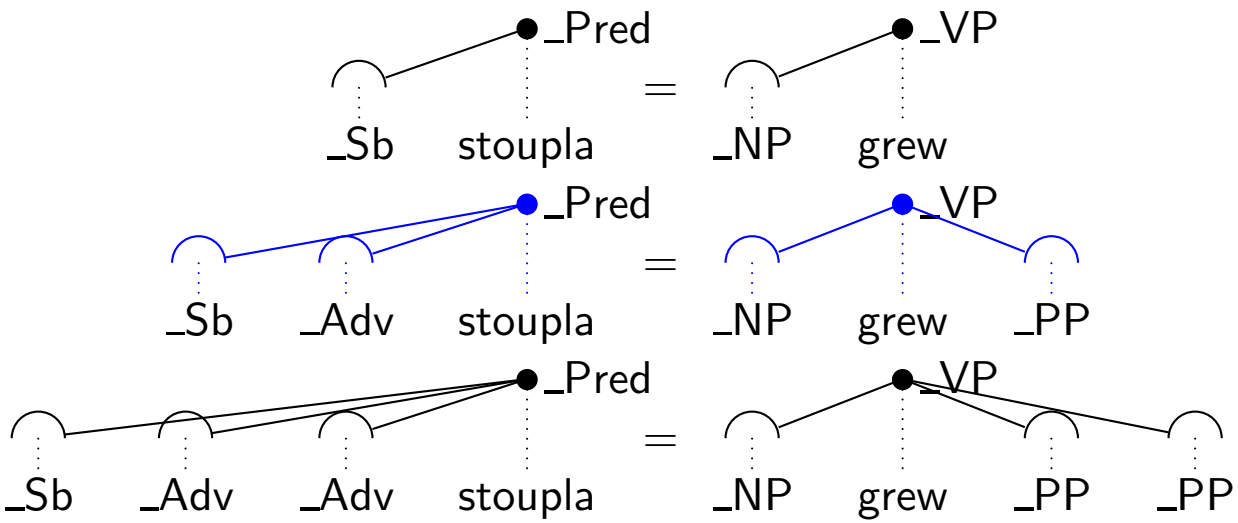
- Disertace Martina Čmejrka nabízí algoritmus zarovnávání stromu na strom.
- Já zatím používám jednoduchou heuristiku:

1. Získej **zarovnání uzel na uzel** (GIZA++ na linearizovaných stromech).
2. Extrahuj všechny dvojice stromečků splňujících všechny tyto podmínky:
   - ne více než $i$ vnitřních uzlů a $f$ slotů,
   - **kompatibilní se zarovnáním uzlu na uzel**,
     např. překlad žádného uzlu nesmí ležet mimo cílový stromeček a sloty si musí být překladem
   - stromečky splňují **podmínku STSG**:
     Všichni následníci vnitřního uzlu musí být rovněž součástí extrahovaného stromečku (ať už jako vnitřní uzly nebo sloty), tj. pro vybudování stromu nebyla třeba operace adjunkce.
3. Podílem frekvencí odhadni pravděpodobnosti, např. $p(t_1, t_2|\text{kořen}_1, \text{kořen}_2)$

# Rizika ředění dat (1)

Počet a stavy slotů pro volná doplnění:

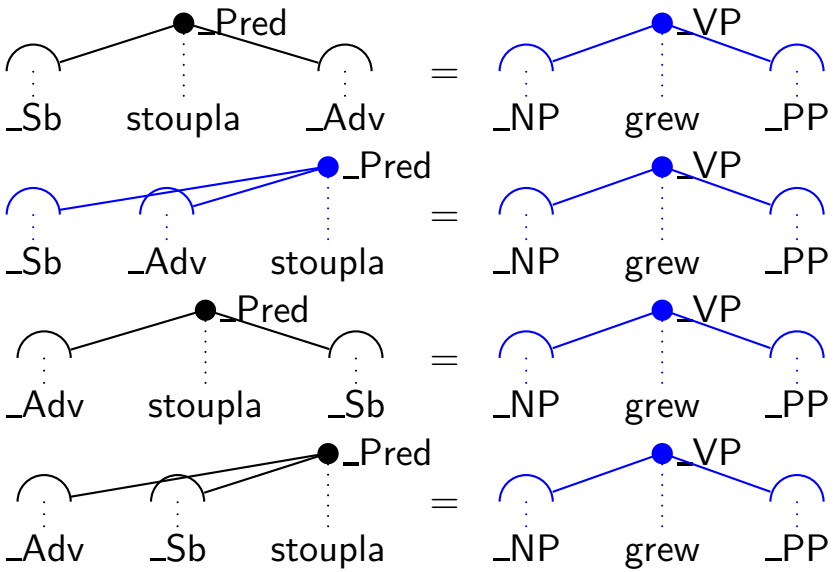
- Podmínka STSG: Jakmile je uzel použit jako vnitřní, musí být vyrobeni i všichni jeho následníci. (Neexistuje operace adjunkce, připojení dalších synů.)



# Rizika ředění dat (2)

Pořadí uzlů:

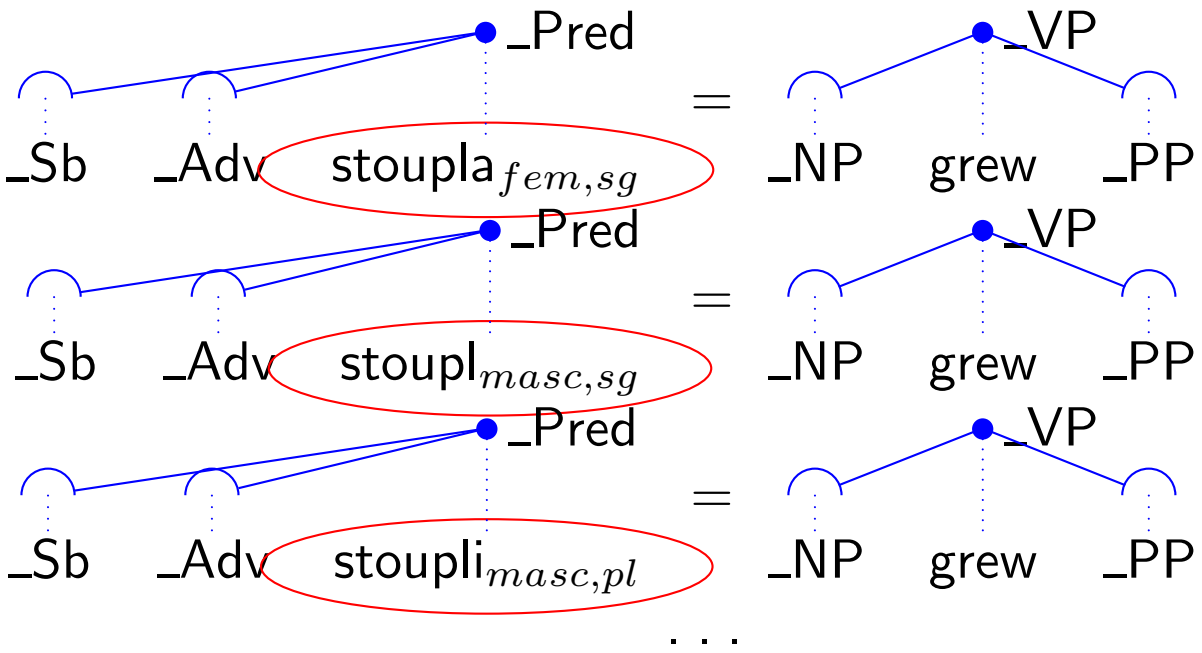
- Volný slovosled češtiny pro jednu anglickou variantu nabízí mnoho překladů.
- Na t-rovině problém můžeme odložit až do generátoru povrchového vyjádření.



# Rizika ředění dat (3)

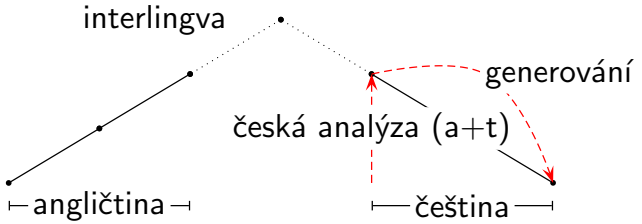
Morfologická bohatost:

- na t-rovině neměla být problémem, ale t-uzly mají množství atributů. . .



# Horní mez kvality překladu (BLEU)

- Analyzuj české věty až na t-rovinu.
- Případně ignoruj některé atributy uzlů.
- Generuj zpátky českou větu.
- Vyhodnoť BLEU proti původní české větě.

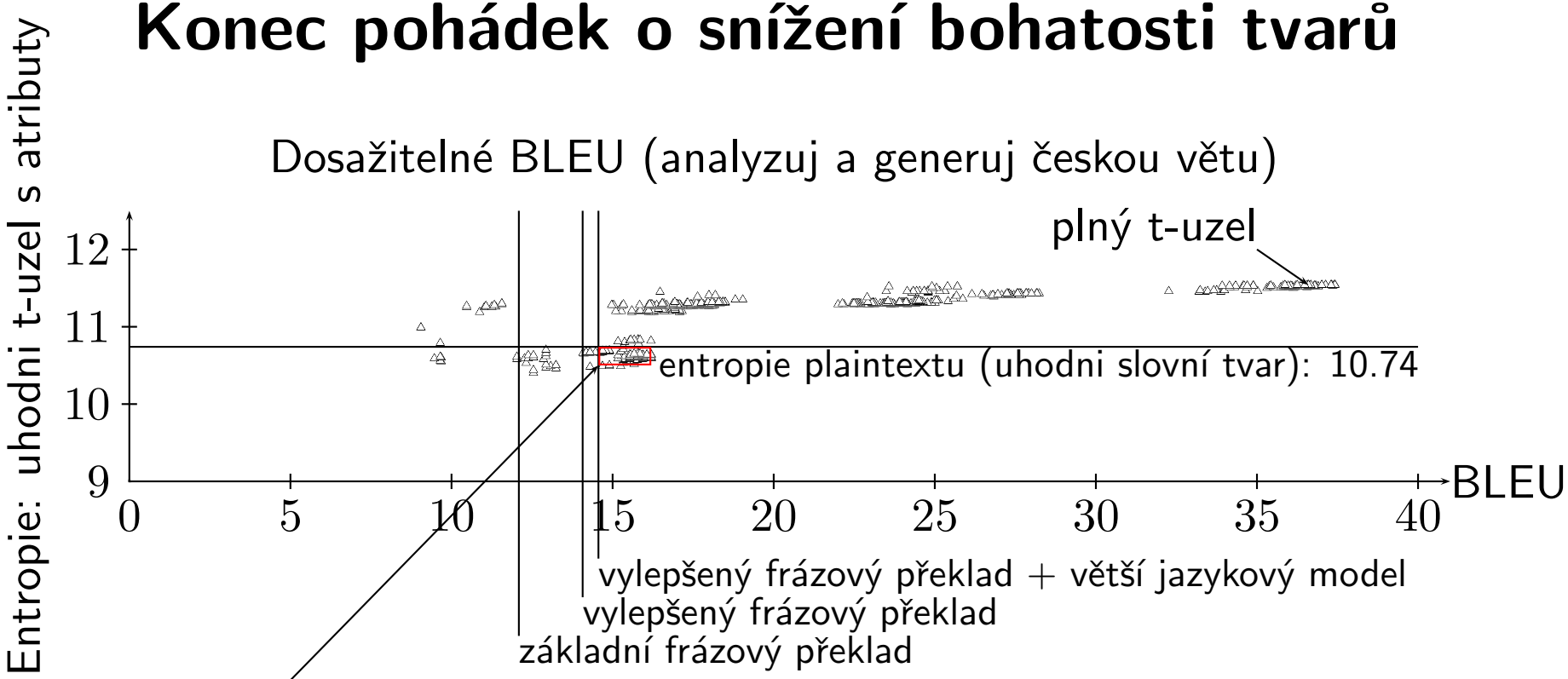



| | Horní mez BLEU |
|---|---|
| Plná automatická t-rovina, žádné atributy neignorovány | 36.6±1.2 |
| Ignoruj typ věty (všechny pokládej za oznamovací) | 36.6±1.2 |
| Ignoruj podrobné gramatémy sloves (resultativita, . . . ) | 36.6±1.2 |
| Ignoruj slovesný čas, rod, . . . | 24.9±1.1 |
| Ignoruj všechny gramatémy | 5.3±0.5 |

⇒ Atributy t-uzlů jsou zásadní pro správné generování.

⇒ Lze najít vhodnou rovnováhu mezi bohatostí slovníku a dosažitelným BLEU?

# Konec pohádek o snížení bohatosti tvarů




Prostor pro zlepšení za předpokladu:

- t-uzly jsou atomické (a omezujeme množinu atributů)
- chceme zůstat pod entropii plaintextu

$\Rightarrow$ I s bezchybným transferem je prostor pro zlepšení zanedbatelný.

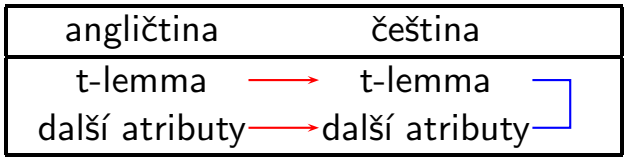
# Důsledek: potřebujeme víc faktorů

- t-rovina sama o sobe zvyšuje složitost výběru popisku uzlu

⇒ nemůžeme brát popisky uzlů jako nedělitelné jednotky: `go.V.past.third.sg...`

Naprosté minimum:

- dva faktory, pro překlad lematu a gramatických atributů odděleně
- kontrola slučitelnosti lematu a gramatémů (příp. s více jednojazyčnými daty)



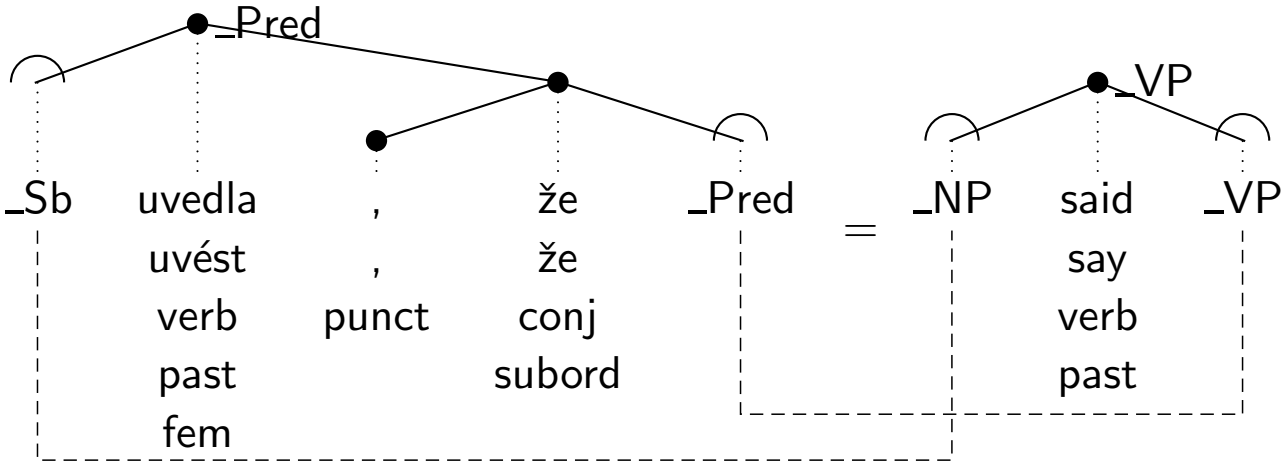

Faktorový překlad je v konfliktu s klíčovou silnou stránkou STSG:

- STSG umožňuje měnit tvar (a velikost) stromu,
- při faktorovém překladu musím vědět, který uzel odpovídá kterému.

⇒ otevřená otázka: jak vhodně spojit stromečky a více faktorů?

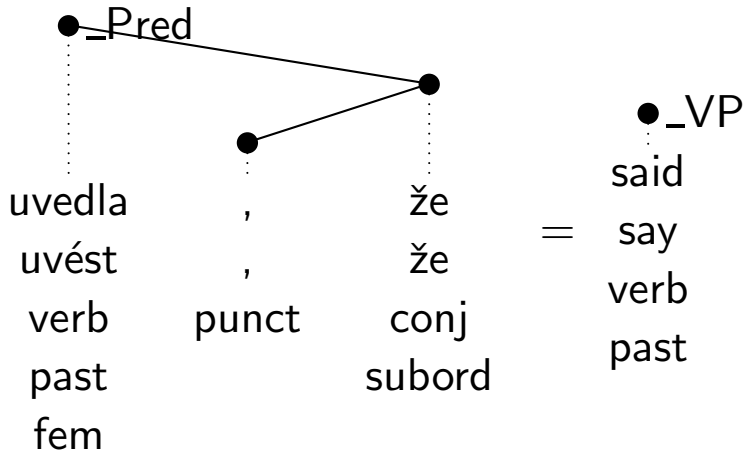
# Konstrukce stromečků 1: věrný převod

- Základní metoda STSG, zachovává:
  - tvary stromků, pořadí uzlů
  - všechny faktory vnitřních uzlů i stavy slotů



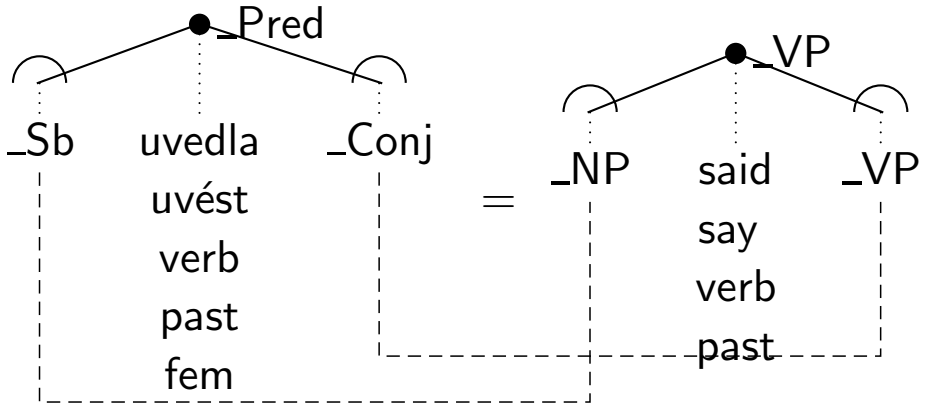
# Konstrukce 2: Přelož vnitřní uzly, dogeneruj sloty

- Zachováváme: tvar stromků, všechny faktory vnitřních uzlů.
- Zdrojové sloty odstraníme, přeložíme a umístíme každý zvlášť.
  - Pro jednoduchost jen v případě okamžité linearizace (netřeba rekonstruovat strukturu).
  - Umisťujeme sloty jen před nebo za vnitřní uzly, nikoli mezi ně.



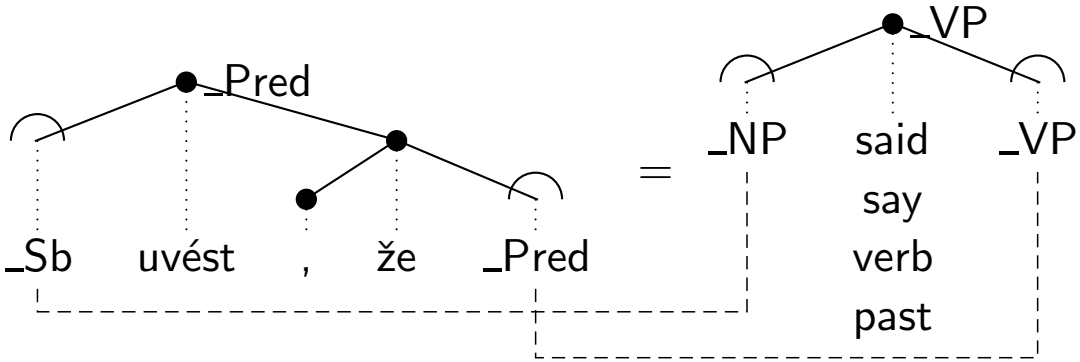
# Konstrukce 3: Překlad uzel po uzlu

- Stromky omezeny na jeden vnitřní uzel, všechny faktory zachovány.
- Sloty přeložíme každý zvlášť, pro jednoduchost zachováváme pořadí.
- Lze užít i pro generování struktury (otec slotů je evidentní).
- Pokud použijeme jen tuto metodu, nelze měnit počet uzlů.
  ⇒nevhodné pro překlad na analytické rovině



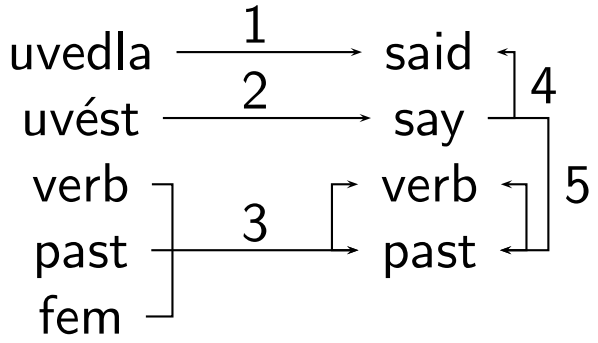
# Konstrukce 4: Ignoruj některé vstupní faktory

- Odhlédneme od některých vstupních atributů ⇒ omezíme bohatost vstupních struktur.
- Výstupní faktory generujeme všechny (tj. hádáme hodnoty dle kontextu).



# Konstrukce 5: Vícefaktorový překlad (uzel po uzlu)

- Analogie frázového překladu o více faktorech (Koehn and Hoang, 2007).
- Konfigurací dána posloupnost kroků:
  - **Překladové kroky** převádějí vstupní faktory na výstupní.
  - **Generující kroky** svazují výstupní faktory mezi sebou.
  - Pořadí důležité s ohledem na omezený zásobník částečných hypotéz.
- Zatím uplatňujeme jen v překladu uzel po uzlu, nevzniká konflikt struktur.



# Kombinace modelů

- Pravděpodobností model STSG rozšířen do log-lineární kombinace modelů:

$$\text{nejlepší derivace } \hat{\delta} = \underset{\delta \in \Delta(T_1)}{\operatorname{argmax}} \exp\Big( \sum_{m=1}^{M} \lambda_m h_m(\delta) \Big) \tag{1}$$

$$\text{místo } \hat{\delta} = \underset{\delta \in \Delta(T_1)}{\operatorname{argmax}} p(t^0_{1:2} | Start_{1:2}) * \prod_{i=1}^{k} p(t^k_{1:2} | q^k_{1:2}) \tag{2}$$

- Konfigurace určuje, které modely v jakém pořadí zapojit.
  - Např. přednostně věrný překlad stromků, nelze-li překládej uzel od uzlu.
- Váhy $\lambda_m$ pro souběžně užité komponenty volíme pro nejlepší skóre (MERT).
  - Aktuálně jen zkouším několik málo bodů.
  - Připraveno hledání optima dvěma metodami: (Och, 2003) a (Smith and Eisner, 2006)

# Poznámky k implementaci

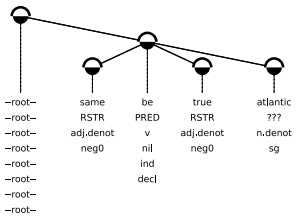
- Externí hašování překladových slovníků stromečků (pomocí Tinycdb, rychlejší implementace GDBM).
- Nově i externí sběr frekvencí (mergesort) $\Rightarrow$ neomezená velikost slovníku.
- Struktura cílového stromu může být rovnou linearizována.
  - Lze užít n-gramový jazykový model už při budování hypotéz (užívám IrstLM).
  - Při generování struktury užívám hranový jazykový model:
    $\Rightarrow$ preferuje pravděpodobnější kombinace otec-syn (konfigurovatelné faktory).

- Implementováno v Mercury (Somogyi, Henderson, and Conway, 1995).
- Paralelní sběr četností i překlad na Sun Grid Engine.

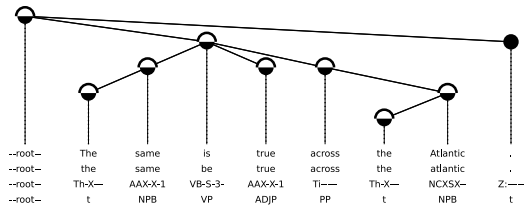
# Dosavadní výsledky (BLEU)

| Roviny \ Jazykové modely | žádný | $n$-gram/*binode* |
|---|---|---|
| epcp bez faktorů | 8.65±0.55 | 10.90±0.63 |
| eaca bez faktorů | 6.59±0.52 | 8.75±0.61 |
| etca bez faktorů | - | 6.30±0.57 |
| etct s faktory, zachovává strukturu | 5.31±0.53 | *5.61±0.50* |
| eact, faktory jen na vstupu, výstup atomický | - | *3.03±0.32* |
| etct, základní STSG (bez faktorů), všechny atributy | 1.61±0.33 | *2.56±0.35* |
| etct, základní STSG (bez faktorů), jen t-lemata | 0.67±0.19 | - |

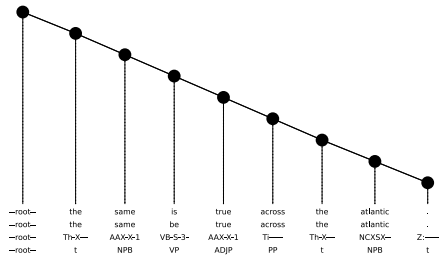
t:




a:




p:



# Diskuse: proč t-rovina škodí

- kumulace chyb každého kroku analýzy:
  - např. $93\% * 85\% * 93\% * 92\% = 67\%$
  - uvedé výsledky vycházejí ze starších nástrojů zejm. pro angličtinu: (Ratnaparkhi, 1996), (Collins, 1996); loňská Zdeňkova aj t-analýza (ignorovala slovesný čas)

- výrazná ztráta dat kvůli neparalelním strukturám:
  - stačí jedna chyba v českém či anglickém parsingu nebo slovním zarovnání
  - stačí "včerejší jednání" místo "meeting yesterday", abych nemohl užít věrný překlad stromku

- neužívá $n$-gramový jazykový model (generování Honzy Ptáčka deterministické) $\Rightarrow$ BLEU nás znevýhodňuje.

- generování počítá s ručními stromy, přeložené automatické mají spoustu chyb

# Srovnání s frázovým překladem

| Metoda | Jazykový model | BLEU |
|---|---|---|
| vícefaktorový Moses, víc dat | 4-gramy slov + 7-gramy značek, SYN 2006 | 15.3±0.9 |
| vícefaktorový Moses, víc dat | 3-gramy slov + 7-gramy značek | 14.2±0.7 |
| základní Moses (frázový překlad) | 3-gramový | 12.9±0.6 |
| epcp bez faktorů | 3-gramový | 10.90±0.63 |
| epcp bez faktorů | žádný | 8.65±0.55 |
| nejlepší etct | binodový | 5.61±0.50 |

- Moses lepší než "epcp":
  - "epcp" neumožňuje přehazovat fráze.
  - Moses má řádně implementován MERT.
- $n$-gramový LM očividně pomáhá, ale i "epcp" bez LM $>$ "etct".

# Aktuální vývoj

- Pochvala Zdeňkovu prostředí TectoMT:
  - Jednotný datový formát (.tmt) pro všechny roviny.
  - Snadné zapojení nových dílčích komponent (přispívají všichni).
  - Při dodržení limitu 50 vět v souboru unese skutečně velká data:
    SYN2006: 22.4 mil. vět; CzEng0.7: 1 mil. vět
  - Denní testy na třech počítačích.

Za týden překlad pro soutěž WMT 2008:

- Doufám, že stačím rozhodnout mezi:
  - Zdeňkovy aj+čj t-analýzy + Zdeňkovo generování.
  - Čj t-analýza Vaška Klimeše + generování Honzy Ptáčka.
  - Čj t-analýza Vaška Klimeše + Zdeňkovy formémy v transferu + generování Honzy Ptáčka.
- Frázový překlad pro srovnání připraven.

# Závěr

- Tektogramatický transfer skýtá naději gramaticky koherentního výstupu.
- Složitější scénář má ale výrazně více volných parametrů:
  - Přesné definice t-rovin (důraz na Zdeňkovy formémy vs. funktory a dogenerované uzly).
  - Konkrétní nástroje (vícero taggerů, vícero parserů) a jejich konfigurace.

  $\Rightarrow$ (Příliš) široký prostor pro experimentování.
- Kumulace chyb jednotlivých kroků.
- Potíže základního modelu STSG:
  - Silné předpoklady (kompatibilní struktura, atomické labely) ředí data.

  $\Rightarrow$ Nutno dobudovat záchranné metody, vyladit vícefaktorový překlad.

! Užíváme "inteligenci" tam, kde v praxi stačí doslovný opis.

$\Rightarrow$ Zatím vítězí frázový překlad.

(Optimisté ještě nevymřeli.)

# Literatura

Čmejrek, Martin. 2006. Using Dependency Tree Structure for Czech-English Machine Translation. Ph.D. thesis, ÚFAL, MFF UK, Prague, Czech Republic.

Collins, Michael. 1996. A New Statistical Parser Based on Bigram Lexical Dependencies. In Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, pages 184–191.

Koehn, Philipp and Hieu Hoang. 2007. Factored Translation Models. In Proc. of EMNLP.

Och, Franz Josef. 2003. Minimum Error Rate Training in Statistical Machine Translation. In Proc. of the Association for Computational Linguistics, Sapporo, Japan, July 6-7.

Ptáček, Jan. 2005. Generování vět z tektogramatických stromů Pražského závislostního korpusu. Master's thesis, MFF, Charles University, Prague.

Ratnaparkhi, Adwait. 1996. A Maximum Entropy Part-Of-Speech Tagger. In Proceedings of the Empirical Methods in Natural Language Processing Conference, University of Pennsylvania, May.

Smith, David A. and Jason Eisner. 2006. Minimum-Risk Annealing for Training Log-Linear Models. In Proceedings of the International Conference on Computational Linguistics and the Association for Computational Linguistics (COLING-ACL), Companion Volume, pages 787–794, Sydney, July.

Somogyi, Zoltan, Fergus Henderson, and Thomas Conway. 1995. Mercury: An Efficient Purely Declarative Logic Programming Language. In Proceedings of the Australian Computer Science Conference, pages 499–512, Glenelg, Australia, February.