Kontroly valence v PDT 2.0

Petr Pajas

ÚFAL MFF UK

28. listopadu 2005

Úvod

Budu mluvit

- o tom, co lze nalézt ve valenčním slovníku PDT-Vallexu
- o tom, jak souvisí údaje v PDT-Vallexu s daty v PDT 2.0
- o pravidlech, která jsem použil při kontrole anotace valence
- téměř výhradně o anotaci valence sloves

Nebudu mluvit

- o valenci jako takové ani o lingvistické motivaci za PDT-Vallexem
- o konkrétních jevech, na které jsme narazili
- o anotaci valence jmen
- o úsilí, vynaloženém při vytváření slovníku a anotaci dat
- o XML formátu valenčního slovníku
- o nástrojích na jeho zpracování

Základní data o PDT 2.0

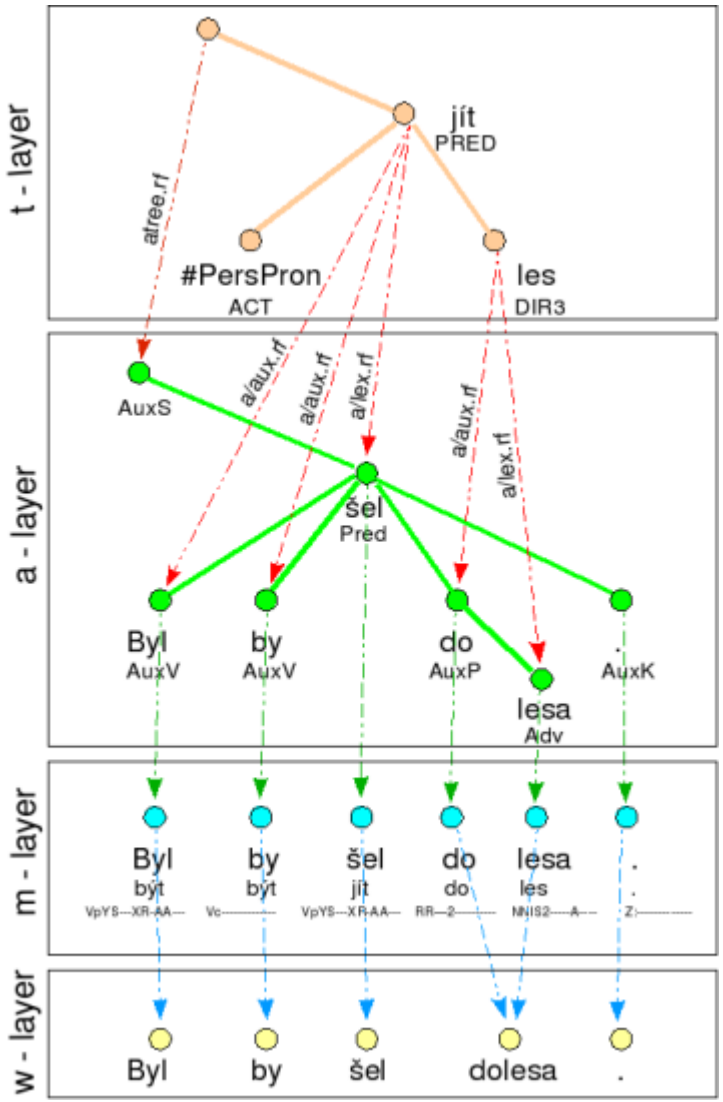
PDT 2.0 obsahuje:

- 116 tisíc vět anotovaných na morfologické rovině (m-rovina)
- z nich 88 tisíc vět anotovaných na analytické rovině (a-rovina)
- z nich 49 tisíc vět anotovaných na tektogramatické rovině (t-rovina)

Slovesa na t-rovině:

- 5 327 různých slovesných t-lemat
  - z nich 1 808 (tj. asi 33%) se v datech vyskytuje jen jednou
- celkem 88 037 výskytů (z celkem 730 435 uzlů, tj. asi 12%)
- nejčetnější slovesa:

13 652	být
3 078	mít
1 039	říci
896	jít



Jednotlivé roviny anotace jsou shora dolů provázány odkazy.

Uzly t-roviny jsou provázány s uzly a-roviny pomocí odkazů v atributech `a/lex.rf` a `a/aux.rf` t-uzlů, přičemž jde o vztah N:M.

Valenční slovník PDT-Vallex

Začal vznikat již během anotací.

Měl sloužit především:

- anotátorům k udržení vzájemné konzistence při přiřazování funktorů aktantům a dalším slovesným doplněním
- k následné kontrole anotace
- v budoucnu též potencionálně pro generování, automatické přiřazování funktorů a další aplikace

Struktura valenčního slovníku PDT-Vallex

Slovo obsahuje následující informace:

- jednoznačný identifikátor
- t-lema + slovní druh (V, N, A, D)
- seznam rámců

Rámec popisuje vztah t-roviny k nižším rovinám (lokálně, v okolí výskytu slova). Obsahuje:

- jednoznačný identifikátor
- podmínky na formu slovesa (jen výjimečně - např. negace „není nad pořádné pivo“, apod.)
- seznam aktantů a povinných doplnění
- u každého z nich funktor a nějaké „idealizované“ podmínky na jeho formu
- příklad (v něm bývají mj. uvedeny formy a funktory dalších obvyklých doplnění)
- poznámku (synonyma, upřesnění významu, atp.)

Odkazy do PDT-Vallexu v datech

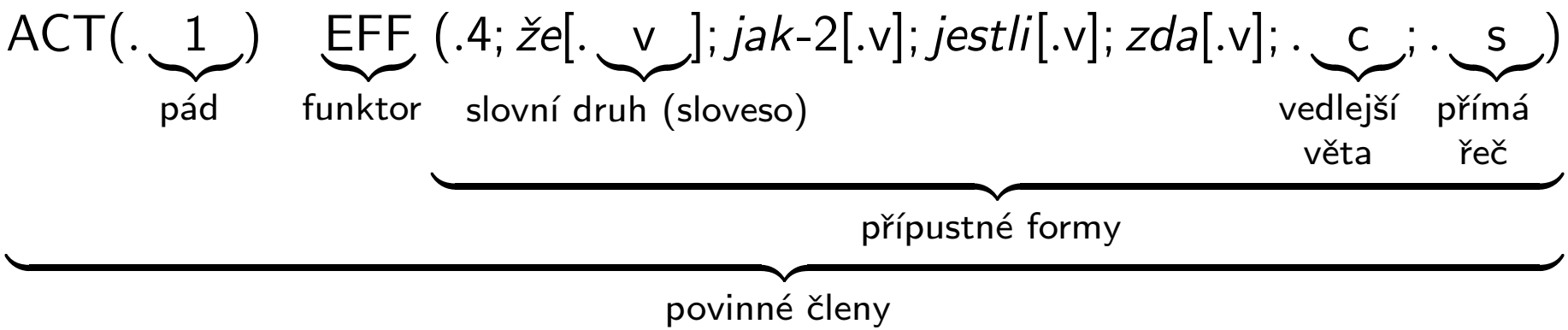
V PDT 2.0 je na t-rovině každý výskyt sémantického slovesa svázán s právě jedním rámcem daného slovesa (t-uzel obsahuje identifikátor rámce v atributu `val_frame.rf`).

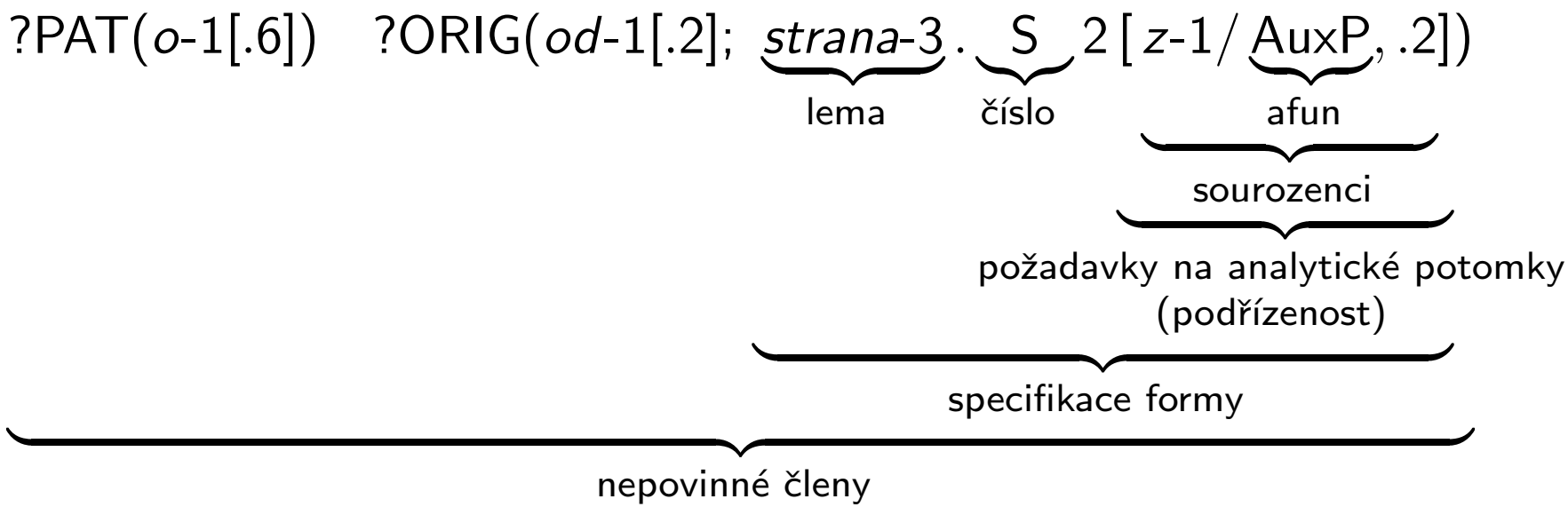
Stejným způsobem jsou s PDT-Vallexem svázána některá vybraná jména:

- všechny výskyty substantiv na -ní,-tí, která se aspoň jednou vyskytla s nějakým aktantem, případně s DPHR
- veškerá substantiva s funktorem CPHR
- veškerá jména, pod nimiž visí DPHR
- některá další jména, u nichž jsme ovšem správnost rámce a jeho přiřazení systematicky nekontrolovali

Příklad rámce v PDT-Vallexu

Sloveso: *slyšet*



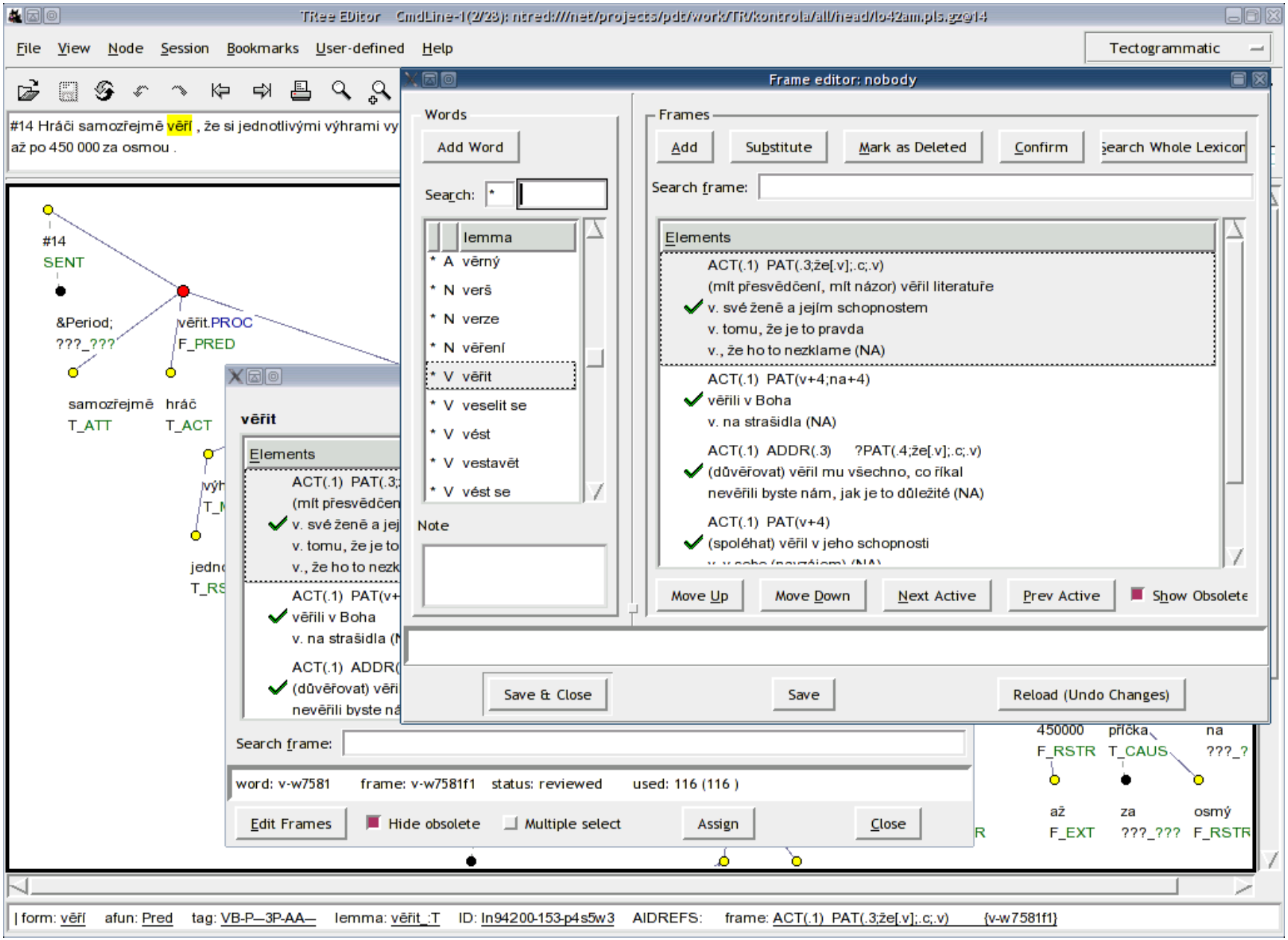


Vytváření PDT-Vallexu

- nejprve jsme do PDT-Vallexu převedli:
  - 997 rámců (332 sloves) z Valexu Markéty Lopatkové
  - 3231 rámců (1412 sloves) sebraných Zdenkou Urešovou v prvních fázích anotace
- další slova a rámce přidávali sami anotátoři dle potřeby
- off-line verze slovníků jednotlivých anotátorů jsem od nich průběžně vybíral, slučoval a vracel zpět. Konflikty verzí byly řešeny automaticky nástrojem na slévání.

Během této fáze se užíval podstatně jednodušší, v podstatě neformální zápis forem.

Uživatelské rozhraní slovníku v TrEdu



PDT-Vallex po skončení anotací

- Po skončení anotací obsahoval slovník
  - asi 8 tisíc platných rámců pro asi 5 tisíc sloves
  - asi 2 tisíce platných rámců pro 1,5 tisíce substantiv
- V datech měla většina sloves (vyjma *být* a *mít*) přiřazen jeden nebo i více rámců.
  - Slovesům *být* a *mít* anotátoři rámce nepřiřazovali.
  - Substantiva měla přiřazen rámec spíš výjimečně.
- Slovník bylo třeba vyčistit po formální stránce a specifikace forem pokud možno automaticky převést do formálního tvaru.
- Po obsahové stránce celý slovník prošly, sjednotily, vyčistily a doplnily Zdena Urešová (slovesa) a Alla Bémová (substantiva).

Kontrola anotace valence

- Kontroly a čištění dat probíhaly zhruba od poloviny roku 2003 do března 2005
- Jejich součástí byla iterativní kontrola shody PDT-Vallexu s anotací (validace), s cílem
  - využít maximum informace uložené ve Vallexu ke kontrole dat
  - propustit jen to, co je opravdu správně
  - poradit si přitom se systematickými odchylkami ve formě
  - v závěru využít prověřené validační procedury k automatickému přiřazení rámce u některých typů (*být*, *mít*, některá substantiva)
- Během jednotlivých iterací se opravovala jak data, tak slovník
- Současně se vylepšovala i validační pravidla.
- Vybrané typy konfliktů mezi daty a slovníkem ručně procházely Zdena Urešová (slovesa), Alla Bémová (substantiva) a Veronika Kolářová (slovesa s CPHR).

Základní kroky validace

1. transformace rámce přiřazeného v `val_frame.rf`
   - Z rámce na vstupu se vygeneruje na základě informací o kontrolovaném uzlu a dané sady transformačních pravidel (jiná pro slovesa a substantiva) jeden nebo více rámců, vůči nimž je následně uplatněna validace.
2. vlastní validace dat vůči rámcům získaným transformací

Transformační pravidla

Každé pravidlo má dvě hlavní části:

1. podmínku na kontrolovaný uzel. Není-li splněna, pravidlo selhává a nesmí být aplikováno.
2. sadu přepisů

Každý přepis má tři části:

1. typ (náhrada, alternativa)
2. předpoklad o tvaru rámce
3. specifikace změn v rámci

Transformační procedura

prochází jednotlivá pravidla v daném pořadí a testuje, zda kontrolovaný uzel vyhovuje podmínce pravidla.

Pokud ano, prochází přepisy a testuje předpoklady o tvaru rámce.

Jsou-li splněny, aplikuje změny a transformace končí; žádné další přepisy ani pravidla se již neaplikují.

Typy přepisů:

- Náhrada — nejčastější typ. Transformace vrátí jen rámec vzniklý přepisem.
- Alternativa — Transformace vrátí oba rámce: vstupní i transformovaný. Užívá se v případě, kdy podmínka nemůže dokonale rozlišit, zda k transformaci má nebo nemá dojít.

Příklad transformačního pravidla

Věta: *Tonda má cestu povolenu od vedení.*

Slovníkový rámec

povolit ACT(.1) PAT(.4;.f;*aby*[.v];*ať*[.v];*že*[.v]) ADDR(.3)

Podmínka transformačního pravidla

Detekuje typ „má uděláno od někoho“ a odliší ho od ostatních užití pasíva s *mít*

Jeden z přepisů

alternativa $\underbrace{\text{ACT(.1)} \quad \text{ADDR(.3)}}_{\text{podmínka pravidla (toto musí původní rámec obsahovat)}} \Longrightarrow$

$\underbrace{\text{-ACT(.1)} \quad \text{-ADDR(.3;}\textit{pro-1}\text{[.4])}}_{\text{odstranění forem}} \quad \underbrace{\text{+ACT(.7;}\textit{od-1}\text{[.2])} \quad \text{+ADDR(.1)}}_{\text{přidání forem}}$

Výsledný rámec

ACT(.7;*od-1*[.2]) PAT(.4;.f;*aby*[.v];*ať*[.v];*že*[.v]) ADDR(.1)

Pravidlo obsahuje ještě podobný přepis pro rámce bez ADDR.

Další pravidla týkající se sloves shrneme už jen stručně:

- transformace pro slovesa v pasivním tvaru (tag /^Vs/); zahrnuje případy *„problém je vyřešen někým“* i *„problém má být vyřešen“*, ale již ne případ *„někdo má problém vyřešen“*, pokrytý předchozím pravidlem.
  - Celkem 10 různých možných přepisů. Nejčastější přepis:
    ACT(.1) PAT(.4) $\Longrightarrow$
    -ACT(.1) -PAT(.4) +ACT(.7;*od-1*[.2]) +PAT(.1)
  - Místo PAT může ovšem být i ADDR, CPHR, dokonce i DPHR
  - Místo PAT(.4) může být ADDR(.2) (*tázat se*).
  - Za pozornost stojí též přepis řešící případ:
    „podnik zaměstnal studenty jako brigádníky“ $\Longrightarrow$
    „studenti byli od podniku zaměstnání jako brigádníci“
    forma doplňkového EFF se zde přepisuje shodně s PAT
- podobně pro reflexivní pasívum (*se*/AuxR) (7 přepisů)
  (validace odhalila mnoho chyb, kdy `afun` reflexivního *„se“* nebyl `AuxR`)

- typ *„mřížku nejde udělat“*, *„mřížku je vidět“*; PAT(.4) $\Longrightarrow$ +PAT(.1)
- dispoziční modalita
  „Tonda bruslí“ $\Longrightarrow$ „Tondovi se krásně bruslilo“

  ACT(.1) PAT(.4) $\Longrightarrow$
  -ACT(.1) -PAT(.4) +(.[*se*]) +ACT(.3) +PAT(.1) +MANN(*)

  plus totéž bez PAT
- typ „chce se mu riskovat“

  $ACT(.1) \Longrightarrow -ACT(.1) \; + ACT(.3)$
- typ „nechat si/dát si udělat něco od někoho/někým“

  ACT(.1) PAT(.7) $\Longrightarrow$ -ACT(.1) +ACT(*od-1*[.2];.7) +PAT(.4)

  plus totéž bez PAT
- typy:
  - „má málo/pramálo/něco/co (společného)“
  - „nemá nic/pranic/co (společného)“
  - „má 100/mnoho/hodně/... vlastností (společných)“
  - „má (společné/společného), že“

  PAT(.4) EFF(.4) $\Longrightarrow$ +EFF(.a2)

Postup validace uzlu vůči danému rámci

Na vstupu je

- rámec (slovníkový nebo vzniklý transformací)
- uzel t-stromu (s odkazy do a-roviny)
- sada příznaků (ovlivňují některé kroky validace zpřísněním či oslabením ověřovaných podmínek)

1. Kontrola vstupního rámce:
   - v rámci není zopakován žádný funktor, atp.
2. Kontrola formy vstupního uzlu předepsané rámcem
3. Pořízení seznamu efektivních potomků uzlu
   - Je-li kontrolovaným uzlem podstatné jméno, jsou do seznamu zahrnuty i efektivní potomci všech vztažných zájmen z téhož stromu, od nichž vede k danému uzlu gramatická koreference

4. Promazání „falešných“ potomků:
   - konstrukce s *což* (z apozice se nechá jen 1. člen v povrchovém pořadí)
   - pravá (povrchově) větev apozice přes *jako*, *#Colon* či *#Hyphen*
   - pravá (povrchově) větev koordinace přes *aniž*
     Např. „... mrzelo ho, že sáhli ke stávce, aniž by jednali s vedením... “
   - operace v pravé větvi apozice („...přišli různí hosté, od Havla, přes Slonkovou, až po Plíhala...“)
   - uzly typu *atd*, *a podobně*, ...
5. Kontrola přítomnosti aktantů a povinných doplnění
   - pro každý obligatorní člen rámce musí seznam potomků obsahovat uzel s předepsaným funktorem
   - z aktantů jsou mezi potomky jen ty uvedené v rámci, přičemž každý jen jednou (až na koordinaci a apozici)

6. Kontrola forem
   - u aktantů, DPHR a CPHR musejí předepsané formě vyhovovat všechny uzly v seznamu s příslušným funktorem
   - u ostatních funktorů stačí, když předepsané formě vyhovuje jeden uzel (u koordinace pak celá koordinovaná skupina)
7. Kontrola dogenerovaných uzlů
   pro každý takový uzel v seznamu musí platit aspoň jedna z podmínek:
   - zastupuje nějaký obligatorní člen rámce
   - vede od něj koreferenční šipka
   - zastupuje a-uzel (tj. odkazuje do a-stromu)
   - má potomky

Kontrola formy t-uzlu

Liší se dle typu specifikace formy:

'typical' (*) dle funktoru se vygenerují a ověří strukturní omezení na formu jako v následujícím bodu. Neimplementováno: každý t-uzel vyhovuje (seznamy typických forem jsem neměl k dispozici).

'state' (=) t-uzel musí mít `is_state=1`, dál jako `typical`.

'elided' (!) vyhovuje každý generovaný t-uzel, neodkazující do a-roviny. V PDT-Vallexu zůstal jediný případ takového rámce:
pozbýt ACT(.1) PAT(!)
„obce pozbyly hodně ze svých pravomocí“

strukturní nejčastější typ; stanovuje podmínku na odpovídající analytické podstromy. Způsob ověřování popíšeme dále.

Strukturní omezení formy

Testuje se množina uzlů analytického stromu získaná takto:

1. vyjde se z množiny a-uzlů odkazovaných z kontrolovaného t-uzlu
   - vynechají se přímí efektivní potomci AuxC/AuxP
   - u *#Rcp* se vynechá odkazované analytické *se*.
2. pro doplněné t-uzly neodkazující do a-roviny se množina naplní „falešným“ a-uzlem, vytvořeným na základě funktoru, t-lematu, apod.
   - Týká se *#Emp*, *#Neg*, *#ObligFM*, *#PersPron*, *#Total*, *#Equal*, *#AsMuch*, *#Some*

Z výsledné množiny a-uzlů musí aspoň jeden splňovat zadaná strukturní omezení.

Kontrola strukturních omezení na a-uzlu

Strukturní omezení může stanovit určité podmínky na a-uzel a jeho přímé efektivní potomky; zpravidla se jedná o omezení na lema, některé význačné pozice morfologické značky, někdy i na analytickou funkci, případně určité speciální podmínky (shoda s efektivním rodičem, uzel je kořen přímé řeči, uzel je kořen „přímé“ vedlejší věty).

Tyto podmínky jsou zjednodušené (různé konvence) a **idealizované**, ověřování je tudíž částečně heuristické.

Validace musí být maximálně přísná, ale současně musí být rozumná (ve skutečnosti nechceme ve slovníku uvádět všechny možné formy explicitně)

Ukážu jen ty nejzajímavější případy, které nejsou nasnadě, případně ty kde se „nakukuje“ zpět do t-roviny.

Podmínka na `lema` či `form`:
U zájmen (`P5`, `PP`, `PJ`, případně `P4` s lematy *který*, *jaký*, *co-4*) se místo hodnot atributů `lemma` a `form` berou hodnoty z a-uzlu, odpovídajícího t-uzlu k němuž zájmeno (tranzitivně) koreferuje.

Podmínka „přímá vedlejší věta“ (`.c`):
Současně musí platit:

1. a-uzel má tag `V`
2. v t-podstromu odpovídajícího t-uzlu se nachází uzel s lematem z množiny
   co-1, co-4, jak-2, jak-3, jaký, kdo, kudy, kolik, proč, kde, kdeže, kdy, kam, který, nakolik, odkud, čí
3. mezi uzlem z této množiny a kořenem t-podstromu není hranice klauze, tj. není tam žádné sloveso, až na výjimky typu:
   - neurčitek pod *dokázat, lze, schopný, ...*
   - krátký tvar adjektiva pod *být, bývat, mít*

Pád

Nejvíc heuristik je při kontrole pádu. Kromě zjevných případů se řeší typy:

- *10%, trochu, plno, hodně, nemálo, málo, dost*
- typ „do/kolem/okolo milionu lidí“
- typ „přes/na milion (lidí)“
- a-uzel je v druhém pádě a je přímo efektivně řízen „číslovkovou konstrukcí“ v předepsaném pádu, úplně bez pádu, nebo vyhovující jednomu z předchozích dvou typů
- a-uzel je v 2. pádu a pod ním analyticky visí číslovková konstrukce v předepsaném pádu nebo bez pádu
- typ „za 100 korun benzínu“
- typ „mezi 10 a 20 kusy“
- typ „po jablíčku“
- typ „snědl, co viděl“ (pád se pak bude kontrolovat vůči „co“)
- „vyslovil své ano/ne/pro/proti“

Číslovkové konstrukce

Při kontrole pádu a na některých dalších místech je třeba brát ohled na číslovkové konstrukce (u nich je valenční člen často obsazen počítaným substantivem v 2. pádu).

Detekce číslovkové konstrukce označí daný a-uzel jako kořen *číslovkové konstrukce* právě tehdy, začínal-li tag na `C` (číslovka dle morfologie) nebo měl lema z následujícího seznamu:
dost, málo-3, kolik, tolik-1, trochu, plno, hodně, nesčetně, spousta, půl-1, půl-2, sto-1, sto-2, tisíc-1, tisíc-2, milión, půldruhý, miliarda, stovka, desítka, pár-1, pár-2, příliš

Závěr

- Začínali jsme s 15 tisíci výskyty sloves (a asi 6 tisíci substantiv zmíněných skupin) s nevalidním přiřazením rámce.
- Dalších 15 tisíc nemělo přiřazen žádný rámec (zejména mít, být).
- Po kontrolách má všech 88 037 výskytů sloves (vedle mnoha dalších substantiv) přiřazen právě jeden rámec, a ten je validní.
- ... až na 87 (tedy 0,1%) případů, kde validace selhává, ale data jsou správně (false negative)
- Většina z nich jsou vyšinuté větné konstrukce, nebo konstrukce, jež bychom mohli rozpoznat a vyřešit nějakou heuristickou, ale vzhledem k četnosti jejich výskytů se to už nevyplatilo (bylo rychlejší je projít a zkontrolovat ručně).
- Validační pravidla umožnila rozhodnout rámec u *být*, *mít* z velké části zcela automaticky.