Regulární gramatiky, regulární výrazy a konečné automaty

Thursday, May 30, 2013 8:36 AM

Regulární gramatiky

Gramatika G je čtveřice (N,Σ,P,S), kde:

- N je konečná množina neterminálních symbolů (neterminálů).
- Σ je konečná množina terminálních symbolů tak, že žádný symbol nepatří do N a Σ zároveň (jsou disjunktní).
- P je konečná množina odvozovacích pravidel. Každé pravidlo je tvaru

$$(\Sigma \cup N)^* \longrightarrow (\Sigma \cup N)^*$$

"cokoli poskládaný ze všech možnejch symbolů na cokoli"; S je prvek z N nazývaný počáteční symbol.

- **VĚTNÁ FORMA**
  Def.: Řetězec α se nazývá větnou formou v gramatice G, s počátečním symbolem S, platí-li:

  S =>* α, kde α ∈ (N ∪ T)*

- **VĚTA**
  Def.: Řetězec α se nazývá větou v gramatice G, s počátečním symbolem S, platí-li:

  S =>* α, kde α ∈ T*

- **FRÁZE**
  Def.: Nechť λ = α β γ je větná forma v gramatice G. Podřetězec β se nazývá frází větné formy λ vzhledem k neterminálnímu symbolu A, platí-li

  S =>* α A γ a A =>* β

  Tzn.frázi tvoří listy podstromu derivačního stromu.

- **JEDNODUCHÁ FRÁZE** větné formy α A γ vzhledem k neterm. A je podřetězec β, platí-li

  S =>* α A γ a A => β

- **L-FRÁZE**

  je nejlevější jednoduchou frází

Lineární gramatika = bezkontextová gramatika, která má nanejvýš jeden neterminál na pravé straně. Regulární gramatika je speciálním případem lineární gramatiky, kdy všechny neterminály jsou na levém konci/straně (levá lineární = levá regulární) nebo ekvivalentně pro pravou stranu.

Regulární gramatika – je to gramatika typu 3 = lineární, navíc převedená do regulárního tvaru (podle Chomského hierarchie). Pravidla těchto lineárních gramatik jsou omezena na jeden neterminál na levé straně. Pravá strana se u pravé regulární gramatiky skládá z jednoho terminálu (u lineární i z více), který může být následován jedním neterminálem, tedy:

$$X \to wY$$
$$X \to w,$$

kde X, Y jsou neterminály a w je řetězcem terminálů. Regulární gramatiky se také nazývají **pravé lineární gramatiky**. Obdobně se definují i **levé regulární gramatiky**, které obsahují pravidla typu:

$$X \to Yw$$
$$X \to w$$

Pravé a levé gramatiky jsou ekvivalentní. Jazyky generované regulárními (=lineárními) gramatikami jsou právě jazyky rozpoznatelné konečným automatem.

Lineární gramatika = má na pravé straně právě jeden neterminál

Regulární gramatika = gramatika, která popisuje regulární jazyk, přesně definovaný tvar pravidel (B -> a, B -> aC, B -> e pro pravou regulární gramatiku)

Regulární gramatika je tedy buď jen levá lineární gramatika nebo jen pravá lineární gramatika. Čistě lineární (levo-pravá) gramatika je pak taková gramatika, která sestává z pravých i levých pravidel současně.

Regular languages are also characterized by special grammars called regular grammars whose productions take the following form, where w is a string of terminals.
A → wB or A → w.
*Example. A regular grammar for the language of a*b* is*
S → Λ | aS | T
T → b | bT.
http://www.postech.ac.kr/~seungjin/courses/automata/handouts/handout04_4pp.pdf

http://web.cecs.pdx.edu/~jhein/lectures/Section.11.4.pdf

Grammar	Languages	Automaton	Production rules (constraints)
Type-0	Recursively enumerable	Turing machine	$\alpha \rightarrow \beta$ (no restrictions)
Type-1	Context-sensitive	Linear-bounded non-deterministic Turing machine	$\alpha A \beta \rightarrow \alpha \gamma \beta$
Type-2	Context-free	Non-deterministic pushdown automaton	$A \rightarrow \gamma$
Type-3	Regular	Finite state automaton	$A \rightarrow a$ and $A \rightarrow aB$

From <http://en.wikipedia.org/wiki/Chomsky_hierarchy>

Regulární výrazy

Regulární výrazy umožňují algebraické manipulace s regulárními množinami - umožňují **vyjádření regulárních množin**. Třída regulárních výrazů nad abecedou Σ je definována takto:

- *e* a $\varnothing$ jsou regulární výrazy
- každé písmeno (symbol - znak) $\sigma \in \Sigma$ je regulární výraz nad Σ
- jsou-li R_1 a R_2 regulární výrazy nad Σ, pak i $(R_1 + R_2)$, $(R_1 . R_2)$ a R_1^* jsou regulární výrazy nad Σ

Daná množina je regulární množina nad Σ, právě když může být popsána vhodným regulárním výrazem nad Σ. Každý regulární výraz **U** popisuje jistou množinu **Ũ** slov nad **Σ**: **Ũ** $\subseteq$ **Σ***

Regulární množiny se vhodně charakterizují přechodovými grafy. Přechodový graf T nad abecedou Σ je konečný orientovaný graf, jehož každá hrana je pojmenována jistým slovem $w \in \Sigma^*$;alespoň jeden uzel je počáteční.

Množinu všech slov akceptovaných konečným automatem **A** označíme **Ã**. Množina je regulární nad **Σ** právě když je akceptována vhodným automatem nad **Σ**

Regulární výraz je řetězec popisující celou množinu řetězců (slov), konkrétně regulární jazyk. Používají se nejčastěji v počítačových programech a skriptovacích jazycích pro vyhledávání a úpravu textu. V případě, že uživatel chce v textu vyhledat nějaký řetězec, který nezná přesně nebo který může mít více variant, může zadat regulární výraz, který postihne všechny chtěné varianty. Program tak nalezne všechny části textu, které danému výrazu odpovídají.

Každý z regulárních výrazů označuje jistý regulární jazyk.

Kleeneho teorém: Každý regulární výraz je převoditelný na konečný automat.

Konečné automaty

formálně je konečný automat definován jako **uspořádaná pětice (S,Σ,P,s,F)** , kde:

S je konečná množina stavů.

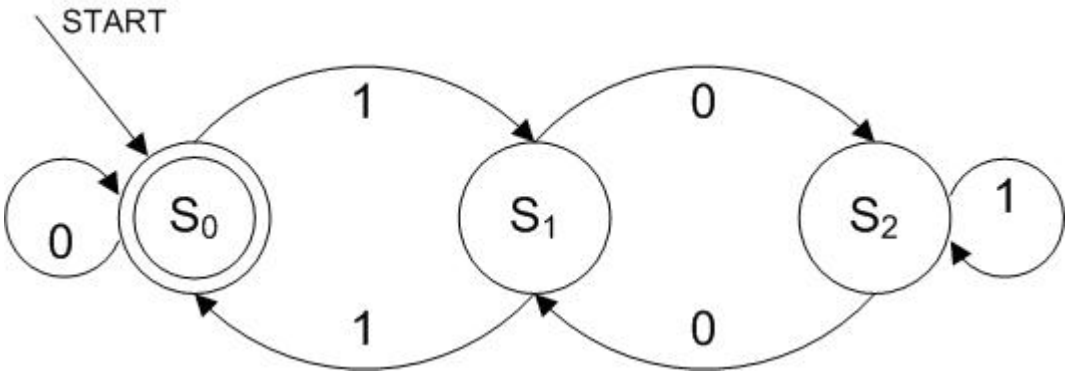
Σ (*velké sigma*) je konečná množina vstupních symbolů nazývaná abeceda.

P je tzv. přechodová funkce (též přechodová tabulka), formálně zobrazení $\delta: S \times \Sigma \rightarrow S$, popisující pravidla přechodů mezi stavy. Přechod je určen stavem ve kterém se automat nachází a symbolem, který přichází na vstup (nebo který je čten na vstupu)

s je počáteční stav (s naleží S)

F je množina koncových (přijímacích) stavů (F je podmnožinou S)

Popis činnosti automatu: Na počátku se automat nachází v definovaném počátečním stavu. Dále v každém kroku přečte jeden symbol ze vstupu a přejde do stavu, který je dán hodnotou, která v přechodové tabulce odpovídá aktuálnímu stavu a přečtenému symbolu. Poté pokračuje čtením dalšího symbolu ze vstupu, dalším přechodem podle přechodové tabulky, atd. Podle toho, zda automat skončí po přečtení vstupu ve stavu, který patří do množiny koncových stavů, platí, že automat buď daný vstup přijal nebo nepřijal. Množina všech řetězců, který daný automat přijme, tvoří regulární jazyk.



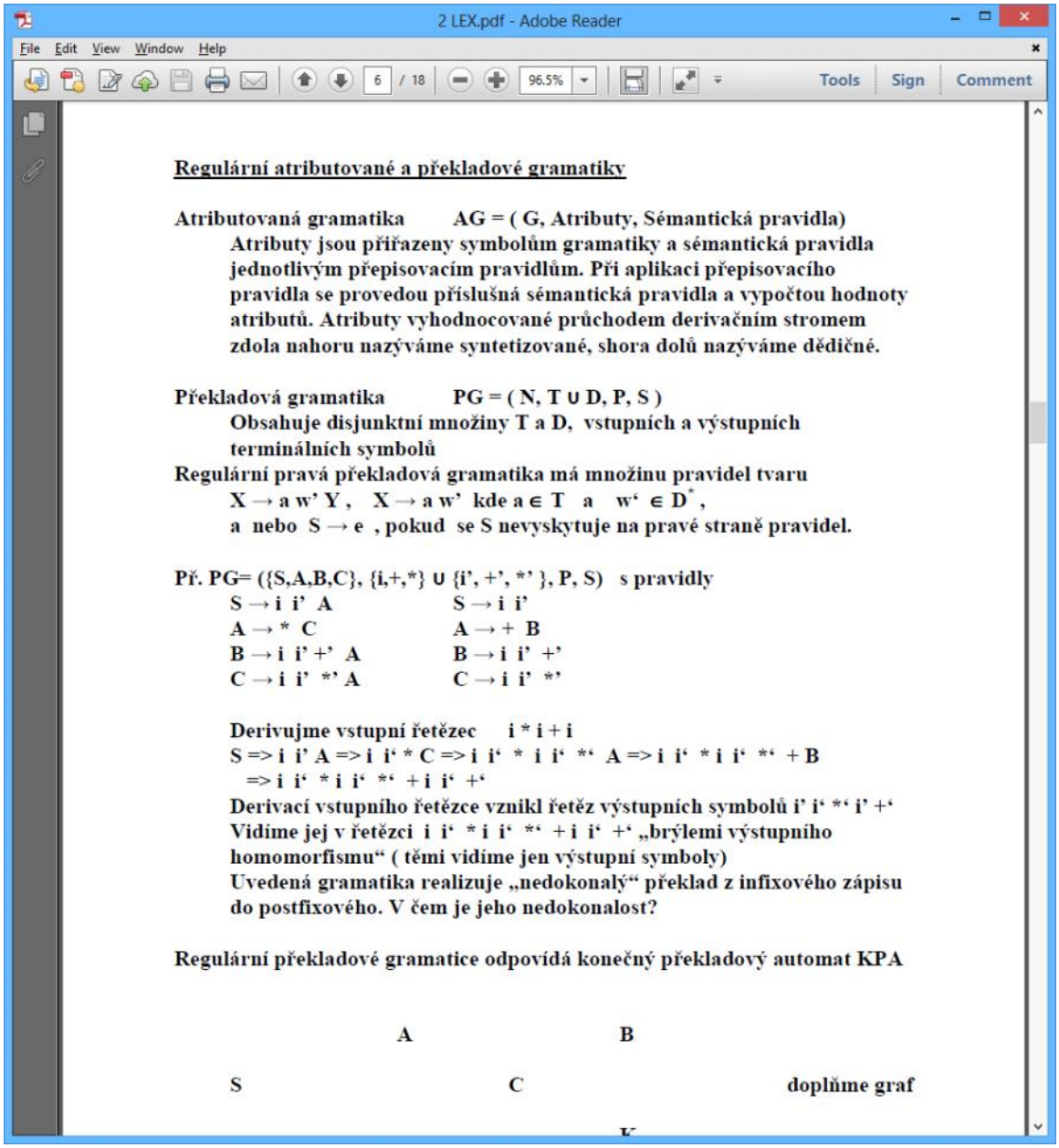
Determinismus

Konečné automaty se dělí dále na deterministické (DKA, DFA) a nedeterministické (NKA, NFA). U deterministických automatů, každý stav má právě jeden možný přechod pro každý možný vstup. U nedeterministického automatu jsou navíc povoleny e-hrany a více hran z jednoho stavu pro stejný vstup => v jeden okamžik se NKA může nacházet ve více stavech současně. Existuje však algoritmus, který umožňuje libovolný NKA převést na o něco složitější DKA (nejhůře v exponenciálním čase, prakticky však řádově rychleji).

Meze regulárních gramatik

Jak určit zdali je nějaký jazyk možné rozpoznat regulárním výrazem (konečným automatem, regulární gramatikou)? K tomuto lze použít tzv. Pumping teorém [wiki] nebo česky [abclinuxu].

Teorém vlastně říká, že v dostatečně dlouhém slově *w* daného regulárního jazyka můžeme nalézt tři části — *x*, *y* a *z*, přičemž nejdůležitější část *y* může zahrnovat i celé slovo. Aby byl tento jazyk regulární, musí platit, že část *y* můžeme ze slova vyjmout, nebo jí libovolně zopakovat, a přitom stále zůstáváme v rámci stejného jazyka.



Regulární atributované a překladové gramatiky

Atributovaná gramatika AG = (G, Atributy, Sémantická pravidla)

Atributy jsou přiřazeny symbolům gramatiky a sémantická pravidla jednotlivým přepisovacím pravidlům. Při aplikaci přepisovacího pravidla se provedou příslušná sémantická pravidla a vypočtou hodnoty atributů. Atributy vyhodnocované průchodem derivačním stromem zdola nahoru nazýváme syntetizované, shora dolů nazýváme dědičné.

Překladová gramatika PG = (N, T ∪ D, P, S)

Obsahuje disjunktní množiny T a D, vstupních a výstupních terminálních symbolů

Regulární pravá překladová gramatika má množinu pravidel tvaru

$X \rightarrow a\ w'\ Y$, $X \rightarrow a\ w'$ kde $a \in T$ a $w' \in D^*$,

a nebo $S \rightarrow e$, pokud se S nevyskytuje na pravé straně pravidel.

Př. PG= ({S,A,B,C}, {i,+,*} ∪ {i', +', *' }, P, S) s pravidly

$S \rightarrow i\ i'\ A$	$S \rightarrow i\ i'$
$A \rightarrow *\ C$	$A \rightarrow +\ B$
$B \rightarrow i\ i'\ +'\ A$	$B \rightarrow i\ i'\ +'$
$C \rightarrow i\ i'\ *'\ A$	$C \rightarrow i\ i'\ *'$

Derivujme vstupní řetězec i * i + i

S => i i' A => i i' * C => i i' * i i' *' A => i i' * i i' *' + B

=> i i' * i i' *' + i i' +'

Derivací vstupního řetězce vznikl řetěz výstupních symbolů i' i' *' i' +'

Vidíme jej v řetězci i i' * i i' *' + i i' +' „brýlemi výstupního homomorfismu" (těmi vidíme jen výstupní symboly)

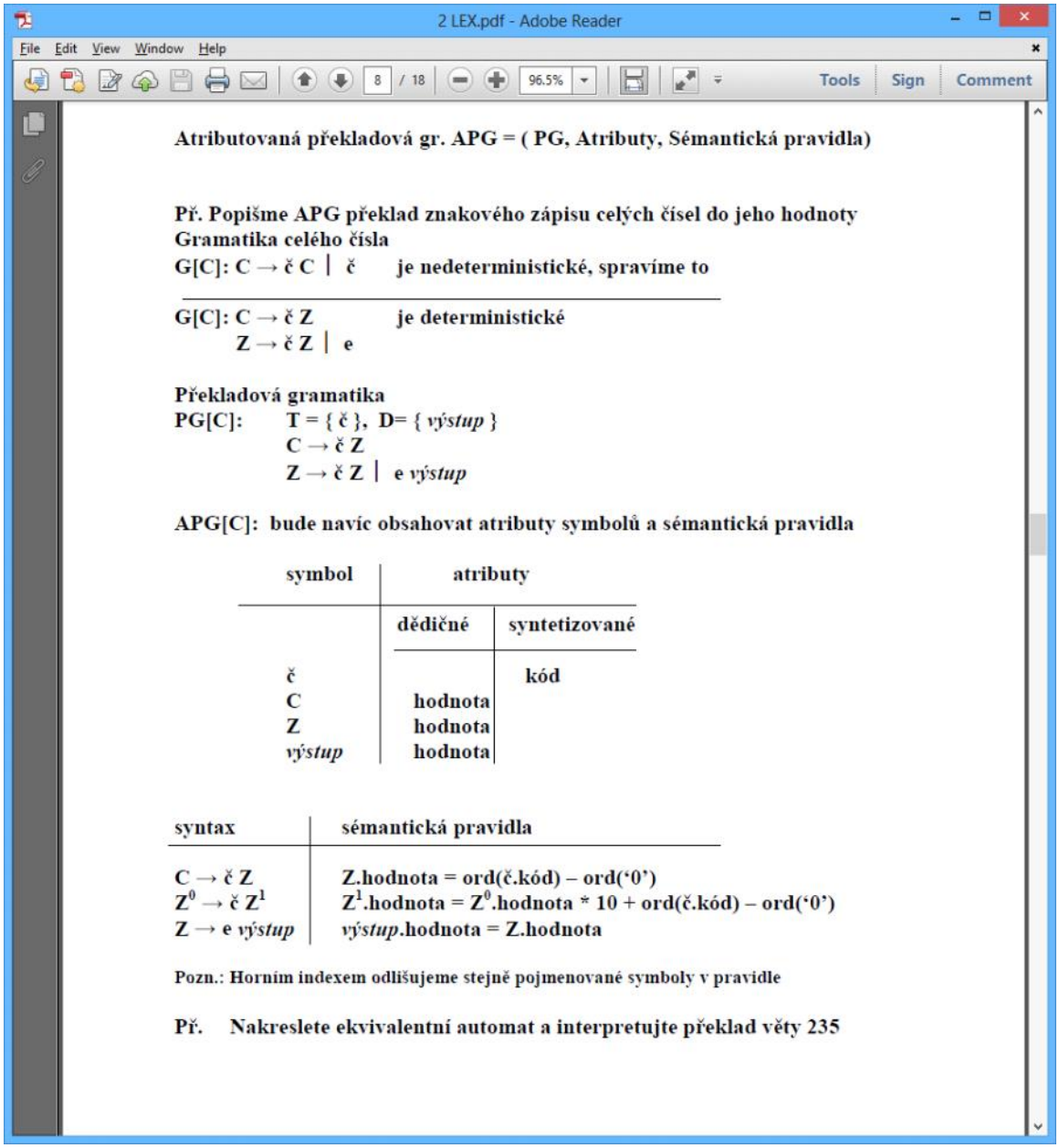
Uvedená gramatika realizuje „nedokonalý" překlad z infixového zápisu do postfixového. V čem je jeho nedokonalost?

Regulární překladové gramatice odpovídá konečný překladový automat KPA

A B

S C doplňme graf

K



Atributovaná překladová gr. APG = (PG, Atributy, Sémantická pravidla)

Př. Popišme APG překlad znakového zápisu celých čísel do jeho hodnoty
Gramatika celého čísla
G[C]: C → č C | č **je nedeterministické, spravíme to**

G[C]: C → č Z **je deterministické**
Z → č Z | e

Překladová gramatika
PG[C]: **T = { č }, D= { *výstup* }**
C → č Z
Z → č Z | e *výstup*

APG[C]: bude navíc obsahovat atributy symbolů a sémantická pravidla

symbol	atributy	
	dědičné	syntetizované
č		kód
C	hodnota	
Z	hodnota	
výstup	hodnota	

syntax	sémantická pravidla
$C \rightarrow č\ Z$	Z.hodnota = ord(č.kód) – ord('0')
$Z^0 \rightarrow č\ Z^1$	Z^1.hodnota = Z^0.hodnota * 10 + ord(č.kód) – ord('0')
Z → e *výstup*	*výstup*.hodnota = Z.hodnota

Pozn.: Horním indexem odlišujeme stejně pojmenované symboly v pravidle

Př. Nakreslete ekvivalentní automat a interpretujte překlad věty 235

From <https://d.docs.live.net/e3534876709763a3/Dokumenty/ZCU/Statnice/Statnice.docx>