



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**SYNTAKTICKÁ ANALÝZA ZALOŽENÁ NA NĚKOLIKA
GRAMATIKÁCH**

PARSING BASED ON SEVERAL GRAMMARS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ONDŘEJ KOUMAR

VEDOUCÍ PRÁCE

SUPERVISOR

prof. RNDr. ALEXANDR MEDUNA, CSc.

BRNO 2024

Abstrakt

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v českém (slovenském) jazyce.

Abstract

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v anglickém jazyce.

Klíčová slova

Sem budou zapsána jednotlivá klíčová slova v českém (slovenském) jazyce, oddělená čárkami.

Keywords

Sem budou zapsána jednotlivá klíčová slova v anglickém jazyce, oddělená čárkami.

Citace

KOUMAR, Ondřej. *Syntaktická analýza založená na několika gramatikách*. Brno, 2024. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. RNDr. Alexandr Meduna, CSc.

Syntaktická analýza založená na několika gramatikách

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana X... Další informace mi poskytli... Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Ondřej Koumar

18. března 2024

Poděkování

V této sekci je možno uvést poděkování vedoucímu práce a těm, kteří poskytli odbornou pomoc (externí zadavatel, konzultant apod.).

Obsah

1	Úvod	4
2	Abstrakt	5
3	Základy teorie formálních jazyků	6
3.1	Abeceda, řetězec a jazyk	6
3.2	Chomského hierarchie	6
3.3	Bezkontextová gramatika	6
3.4	Konečný automat	6
3.5	[[doplnit více vecí?]]	6
4	Důležité pojmy syntaktické analýzy [[rozdělit na více kapitol? srazit některé pojmy do jedné sekce/rozdělit do více? jaké další pojmy doplnit?]]	7
4.1	Derivační krok	7
4.2	Množiny potřebné k sestrojení LL tabulky	7
4.3	LL tabulka	7
4.4	Zásobníkový automat	8
4.5	Prediktivní syntaktická analýza	8
4.6	Precedenční tabulka	8
4.7	Precedenční syntaktická analýza	8
4.8	Abstraktní syntaktický strom	8
5	Cooperating distributed gramatický systém	9
5.1	Derivační krok v CDGS	9
5.2	Jazyk generovaný CD gramatickým systémem	10
6	Představení jazyka Koubp [[opět si nejsem jistý názvem + co všechno vůbec napsat?]]	11
6.1	Gramatický systém definující syntax jazyka Koubp	12
6.2	Deklarace a definice funkcí	12
6.3	Příkazy	12
6.4	Výrazy	12
6.5	Vestavěné funkce	12
7	Implementace syntaktického analyzátoru pro jazyk Koubp [[Urcite rozdělit na více sekcí, co případně z projektu doplnit?]]	13
7.1	Hlavní myšlenky	13
7.2	Lexikální analýza a nástroj Flex	13

7.3	Syntaktická analýza	13
7.4	Abstraktní syntaktický strom	13
8	Závěr	14
	Literatura	15

Seznam obrázků

Kapitola 1

Úvod

Kapitola 2

Abstrakt

Kapitola 3

Základy teorie formálních jazyků

3.1 Abeceda, řetězec a jazyk

3.2 Chomského hierarchie

3.3 Bezkontextová gramatika

Definice 1. *Bezkontextová gramatika* je čtveřice $G = (N, T, P, S)$, kde:

- N je konečná množina neterminálních symbolů,
- T je konečná množina terminálních symbolů,
- P je konečná množina přepisovacích pravidel ve tvaru $A \rightarrow x$, $A \in N$ a $x \in (N \cup \Sigma)^*$,
- $S \in N$ je výchozí symbol gramatiky.

[[derivacni krok sem misto k syntakticke analyze?]]

3.4 Konečný automat

Definice 2.

3.5 [[doplnit vice veci?]]

Kapitola 4

Důležité pojmy syntaktické
analýzy [[rozdelit na vice kapitol?
srazit nektere pojmy do jedne
sekce/rozdelit do vice? jake dalsi
pojmy doplnit?]]

4.1 Derivační krok

Myšlenka *derivačního kroku* je aplikovat pravidlo z množiny pravidel bezkontextové gramatiky, čímž se část původního řetězce přepíše na novou.

Definice 3. Necht $G = (N, T, P, S)$ je BKG, $u, v \in (N \cup T)^*$ a $p = A \rightarrow x \in P$.

Potom uAv přímo derivuje uxv za použití p v G , zapsáno $uAv \Rightarrow uxv[p]$, zjednodušeně $uAv \Rightarrow uxv$.

Je možné říci, že G provádí derivační krok z uAv do uxv .

Sekvence derivačních kroků

[[definici jsem nasek v prezentacich ifj, nicmene to nevypada nejformalneji; bude stacit nebo radsí z literatury?]]

4.2 Množiny potřebné k sestavení LL tabulky

[[tady by mohlo byt Empty(x), First(x), Follow(x), Predict(x) + algoritmy pro jejich sestavení?]]

4.3 LL tabulka

[[algoritmus pro sestavení tabulky?]]

4.4 Zásobníkový automat

Zásobníkový automat je rozšíření konečného automatu, popsaného v definici 2, o zásobník, matematicky přesněji o zásobníkovou abecedu a počáteční symbol na zásobníku.

Definice 4. *Zásobníkový automat* (ZA) je sedmice $M = (Q, \Sigma, \Gamma, R, s, S, F)$, kde:

- Q je konečná množina stavů,
- Σ je vstupní abeceda,
- Γ je zásobníková abeceda,
- R je konečná množina pravidel tvaru $Apa \rightarrow wq$, kde $A \in \Gamma$, $p, q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$,
- $s \in Q$ je počáteční stav,
- $S \in \Gamma$ je počáteční symbol na zásobníku,
- $F \subseteq Q$ je množina koncových stavů.

Rozšířený zásobníkový automat

Původní zásobníkový automat lze rozšířit o další chování. Například o možnost čtení více symbolů ze zásobníku než původního jednoho, tedy při přechodech mezi stavy měnit celé řetězce na vrcholu zásobníku.

Definice 5. *Rozšířený zásobníkový automat* (RZA) je sedmice $M = (Q, \Sigma, \Gamma, R, s, S, F)$, kde:

- $Q, \Sigma, \Gamma, s, S, F$ jsou definovány stejně jako u ZA,
- R je konečná množina pravidel ve tvaru $vpa \rightarrow wq$, kde $v, w \in \Gamma^*$, $p, q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$.

4.5 Prediktivní syntaktická analýza

4.6 Precedenční tabulka

[[algoritmus pro sestavení tabulky?]]

4.7 Precedenční syntaktická analýza

4.8 Abstraktní syntaktický strom

Kapitola 5

Cooperating distributed gramatický systém

[[nejsem si jistý překladem, zatím nechavam v anglictine, ale vypada to divne]]

Cooperating distributed gramatický systém (CDGS) stupně n je systém gramatik, které mezi sebou sdílejí množinu neterminálů i terminálů a startovací symbol.

Definice 6. CDGS je n -tice $\Gamma = (N, T, S, P_1, \dots, P_n)$ pro $1 \leq i \leq n$, kde:

- N , T , a S jsou definovány stejně jako v definici 1,
- P_i je konečná množina pravidel ve tvaru $A \rightarrow x$, kde A i x jsou definovány stejně jako v definici 1, nazývaná *komponentou* systému,
- i -tá gramatika systému se zapisuje jako $G_i = (N, T, S, P_i)$

Alternativní definice pro CDGS je $\Gamma = ((N, T, S, P_1), \dots, (N, T, S, P_n))$.

[[staci takovyto popis? mam se vyhnout vysvetlovani "vlastnimi slovy"nebo je naopak dobre, ze pred definici +- uvedu, o co se jedna?]]

5.1 Derivační krok v CDGS

Notace derivačního kroku v CDGS je

$$x_i \Rightarrow^f y,$$

což znamená, že řetězec $x \in (N \cup T)^*$ derivuje řetězec $y \in (N \cup T)^*$ v i -té komponentě za použití *derivačního režimu* f .

Derivační režimy

Prvním a nejpřirozenějším příkladem je režim $*$ **[[Jak tento režim nazvat?]]**. V tomto případě stačí, aby řetězec y byl derivovatelný z řetězce x v i -té komponentě, zapsáno $x \Rightarrow^* y$ v $G_i = (N, T, P_i, S)$.

Podobným příkladem je režim *ukončovací*, který spočívá v nutné derivaci řetězce v dané komponentě, dokud je to možné. Značí se písmenem t . Jsou dvě nutné podmínky, aby y bylo derivovatelné z x v komponentě G_i režimem t .

1. $x \Rightarrow^* y$ v $G_i = (N, T, P_i, S)$ – v dané komponentě lze posloupností derivačních kroků získat řetězec y z řetězce x ,
2. $y \nRightarrow z$ pro všechna $z \in (N \cup T)^*$ – není jiný další řetězec, který by z y šel odvodit.

Další derivační režimy:

- alespoň k derivací, tedy $x_i \Rightarrow^{\geq k} y$,
- nejvíce k derivací, tedy $x_i \Rightarrow^{\leq k} y$,
- právě k derivací, tedy $x_i \Rightarrow^{=k} y$,

kde $k \in \mathbb{N} \cup \{0\}$ a i symbolizuje i -tou komponentu gramatického systému.

Derivační režimy mohou být reprezentovány jako množina, což pomůže definovat další pojmy v následující podkapitole o generovaných jazycích.

Definice 7. Necht $k \in \mathbb{N}$ a $*$, t představují derivační režimy.

Potom množina $D = \{*, t\} \cup \{\leq k, \geq k, = k\}$ reprezentuje derivační režimy použitelné v CD gramatických systémech.

5.2 Jazyk generovaný CD gramatickým systémem

Než bude definován samotný jazyk, je vhodné definovat pomocnou množinu, která reprezentuje *možné derivace* z řetězců.

Definice 8. Necht $\Gamma = (N, T, S, P_1, \dots, P_n)$.

Potom $F(G_j, u, f) = \{v : u_j \Rightarrow^f v\}$, $1 \leq j \leq n$, $f \in D$, $u \in (N \cup T)^*$ je množina všech řetězců v derivovatelných z u v j -té komponentě za použití derivačního režimu f .

Definice 9. Necht $\Gamma = (N, T, S, P_1, \dots, P_n)$.

Jazyk generovaný systémem Γ za derivačního režimu f , $L_f(\Gamma) = \{w \in T^* : \text{existují } v_0, v_1, \dots, v_m \text{ takové, že } v_k \in F(G_{j_k}, v_{k-1}, f), 1 \leq k \leq m, 1 \leq j_k \leq n, v_0 = S, v_m = w \text{ pro } m \geq 1\}$.

[[jsou definice v tomto formátu v pořadku?]]

Výsledný řetězec w , který vznikl postupnou derivací startovacího symbolu v_0 . Měl několik mezikroků, které jsou reprezentovány řetězci v_1, \dots, v_{m-1} . Každý řetězec v_k , kde $1 \leq k \leq m$ byl zderivován z řetězce v_{k-1} v komponentě G_{j_k} , kde $1 \leq j_k \leq n$ za derivačního režimu f .

Kapitola 6

Představení jazyka Koubp **[[opet si nejsem jisty nazvem + co vsechno vubec napsat?]]**

Jazyk *Koubp* je založený na jazyce IFJ22, který je podmnožinou jazyka PHP 8, jenž byl specifikován v rámci zadání projektu do předmětu Formální jazyky a překladače v akademickém roce 2022/2023. **[[ideálně citovat zadání projektu?]]**

Některé aspekty jazyků jsou společné. Oba dva jazyky jsou strukturované, podporují definici proměnných a funkcí. Hlavní tělo programu se skládá z prolínání sekvence příkazů a definic funkcí, které se mohou vzájemně rekurzivně volat. Neexistuje funkce `main()`, jak lze nalézt například u jazyka C [1]. V uživatelem definovaných funkcích může být větvení, iterace a další běžné konstrukce. Veškeré proměnné jsou lokální, i v rámci hlavního těla programu. Soubory se zdrojovým kódem nelze slučovat a vytvářet tak jediný modul, který by bylo možné zkompileovat. **[[doplnit věci, které nejsou společné (nezabíhat do detailu, vše bude specifikováno v podkapitolách)]]**

6.1 Gramatický systém definující syntax jazyka Koubp

Indexace neterminálů a význam pro implementaci

Deadlock mezi neterminály statement a codeBlock

6.2 Deklarace a definice funkcí

6.3 Příkazy

Přiřazení

Větvení

Cyklus while

Cyklus for

6.4 Výrazy

Operátory

Priorita operátorů

Volání funkcí

6.5 Vestavěné funkce

Vstupně-výstupní funkce

Funkce pro typovou konverzi

Funkce pro práci s řetězcí

Kapitola 7

Implementace syntaktického analyzátoru pro jazyk Koubp

[[Urcite rozdelit na vice sekci, co pripadne z projektu doplnit?]]

- 7.1 Hlavní myšlenky
- 7.2 Lexikální analýza a nástroj Flex
- 7.3 Syntaktická analýza
- 7.4 Abstraktní syntaktický strom

Kapitola 8

Závěr

Literatura

- [1] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO). *ISO/IEC 9899:2018, Programming languages - C*. International Organization for Standardization (ISO), 2018. Dostupné z: <https://www.open-std.org/jtc1/sc22/wg14/www/docs/n2310.pdf>.