

dokumentace.pdf, 3-5 stran A4.

vývojový cyklus, způsob práce v týmu, speciální použité techniky a algoritmy a různé odchylky od přednášené látky či tradičních přístupů. Nezapomínejte také citovat literaturu a uvádět reference na čerpané zdroje včetně správné citace převzatých částí (obrázky, magické konstanty, vzorce). Nepopisujte záležitosti obecně známé či přednášené na naší fakultě.

- Rozdělení práce mezi členy týmu (uveďte kdo a jak se podílel na jednotlivých částech projektu; povinně zdůvodněte odchylky od rovnoměrného rozdělení bodů).
- LL-tabulka
- být založena pouze na výčtu a obecném popisu jednotlivých použitých metod (jdeo váš vlastní přístup k řešení; a proto dokumentujte postup, kterým jste se při řešení ubírali; překážkách, se kterými jste se při řešení setkali; problémech, které jste řešili a jak jste je řešili; atd.) V rámci dokumentace bude rovněž vzat v úvahu stav kódu jako jeho čitelnost, srozumitelnost a dostatečné, ale nikoli přehnané komentáře.



PROJEKT Z PREDMETOV IFJ A IAL
**IMPLEMENTÁCIA PREKLADAČA IMPERATÍVNEHO
JAZYKA IFJ19**

TÍM 36, VARIANTA 02

Vedúci: Žitňanský Adam, xzitna02 (%)

Otčenáš Matej, xotcen01 (%)

Dúdik Samuel, xdudik01 (%)

Žitňanská Lívia, xzitna03 (%)

Obsah:

Úvod	3
Návrh riešenia	3
Lexikálna analýza	3
Syntaktická analýza	3
Sémantická analýza.....	3
Implementácia	4
Dátové štruktúry	4
Tabuľka symbolov	4
Generovanie kódu	4
Práca v tíme	4
Komunikácia	?
Verzovací systém	?
Prílohy	?
Konečný automat lexikálneho analyzátoru.....	5
LL gramatika	5
LL tabuľka	5
Precedentná tabuľka	5

1.Úvod:

Úlohou projektu je načítať zdrojový kód Ifj2019 a generovať výsledný medzikód v IFJcode2019 aj s návratovou hodnotou, kde jazyk Ifj2019 je podmnožina jazyka Python 3².

2.Návrh riešenia:

Kapitola popisuje návrh a implementáciu riešenia, časti prekladača a predávanie informácií medzi nimi.

2.1. Lexikálna analýza:

Lexikálna analýza bola prvá časť, na ktorej sa začalo pracovať. Jej implementácia sa nachádza v súbore **scanner.s** s hlavičkovým súborom **scanner.h** - jeho reprezentácia konečným automatom sa nachádza nižšie v dokumentácii (viz obr.č.1). Scanner využíva zásobník a knižnicu **string_lib**, ktorá slúži na prácu s reťazcami. Hlavné telo Scanneru pozostáva zo switchu, ktorý získa zo vstupného súboru token a zisťuje počet odsadení, či token patrí medzi operátory, operandy, identifikátory, kľúčové slová alebo čísla. Na základe tohto vyhodnotenia vykonáva príslušné príkazy pre každý stav a posunie sa na ďalší token, pre ktorý celý proces zopakuje. Každý spracovaný token je odoslaný na syntaktickú analýzu do Parseru.

2.2. Syntaktická analýza:

Syntaktická analýza sa riadi LL gramatikou a je reprezentovaná súbormi **parser.s** a **expr_parser.c**. Po alokovaní pamäte a inicializovaní Parseru a tabulky symbolov, si Parser vezme token zo Scannera a pravidlami LL gramatiky (obrázok č.2) ho zredukuje a vyhodnotí. Pre prácu s tokenmi sa zaviedol typ zodpovedajúci symbolom, čo uľahčilo prevod tokenu na symbol. Niektoré si automaticky zodpovedali, iné bolo treba manuálne konvertovať, na čo sa použil podmienený príkaz „if“. Na vyhodnocovanie výrazov pre Parser slúži Expr_parser, ktorý je Parserom zavolaný v prípade, že sa za premennou nevyskytuje ľavá zátvorka. Parser obsahuje makrá slúžiace na sprehľadnenie kódu.

2.3. Sémantická analýza:

TODO

3. Implementácia

3.1. Dátové štruktúry

TODO

3.2. Tabuľka symbolov

Implementácia tabuľky symbolov sa nachádza v súbore **symtable.c**. Ako alokovanú veľkosť tabuľky sme zvolili prvočíslo, aby sme sa vyhli zoskupovaniu hodnôt a zabezpečili tak rovnomernejšie distribuovanú tabuľku. Hashovaciu funkciu sme zvolili pre jej rýchlosť a malé množstvo kolízií, čo zvyšuje jej efektivitu.

3.3. Generovanie kódu

TODO

4. Práca v tíme:

4.1. Komunikácia:

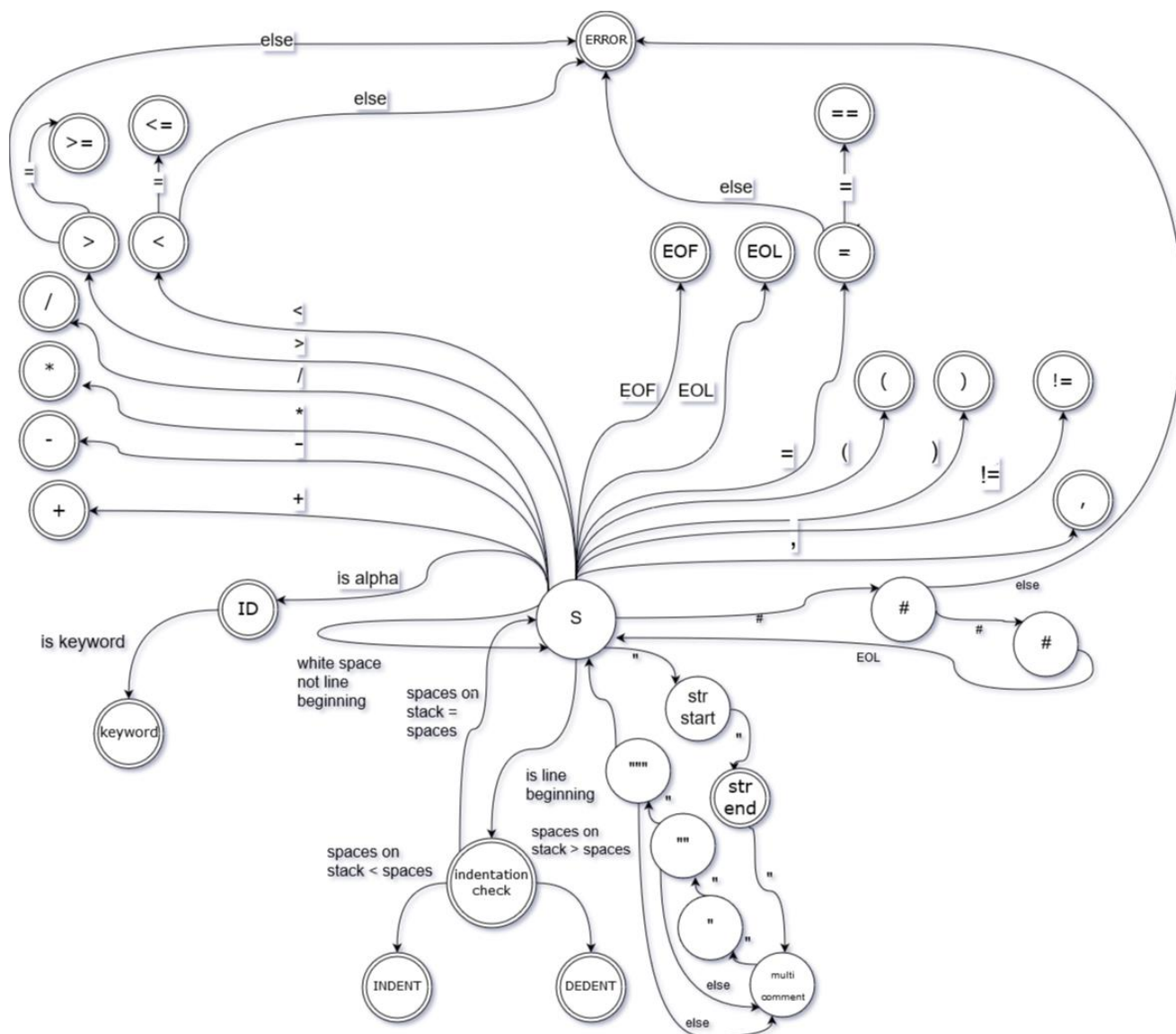
Práca na projekte začala v septembri. Ako komunikačný kanál sme zvolili Discord vzhľadom na skúsenosti s prácou v ňom. Preferovali sme osobné stretnutia a skupinovú prácu - všetci členovia tímu sa stretli a na mieste sa rozdelili úlohy.

4.2. Verzovací systém:

Vzhľadom na predošlé skúsenosti sme sa rozhodli pre prácu s Githubom, kvôli jeho prehľadnosti. Repozitár bol založený začiatkom septembra a každý člen tímu postupne pridával svoju prácu a prípadné úpravy do vetiev, ktoré sa po istej dobe alebo určitom počte úprav spájali v hlavnú.

5. Prílohy:

5.1. Konečný automat lexikálneho analyzátoru



4.2. LL Gramatika

1. <statement> -> EOF
2. <statement> -> EOL <statement>
3. <statement> -> DEF ID (<params>): EOL INDENT <statement_inside> <end> DEDENT <statement>
4. <statement> -> IF <expression_start>: EOL INDENT <statement_inside> EOL DEDENT ELSE: EOL INDENT <statement_inside> <end> DEDENT <statement>
5. <statement> -> WHILE <expression_start>: EOL INDENT <statement_inside> <end> DEDENT <statement>
6. <statement> -> ID = <expression_start> <end> <statement>
7. <statement> -> PASS <end> <statement>
8. <statement> -> PRINT (<arg>) <end> <statement>
9. <statement> -> <value> <end> <statement> // TODO nezahrna to ID, vstavane funkcie (tie sa musia priradit)
9. <statement_inside> -> IF <expression>: EOL INDENT <statement_inside> EOL DEDENT ELSE: EOL INDENT <statement_inside> EOL DEDENT <statement>
10. <statement_inside> -> WHILE <expression_start>: EOL INDENT <statement_inside> EOL DEDENT <statement>
11. <statement_inside> -> ID = <expression_start> EOL <statement_inside>
12. <statement_inside> -> RETURN <expression_start> EOL <statement_inside>
13. <statement_inside> -> PASS EOL <statement_inside>
14. <statement_inside> -> PRINT (<arg>) EOL
15. <end> -> EOF
16. <end> -> EOL
17. <params> -> ID <next_params>
18. <params> -> eps
19. <next_params> -> , ID <next_params>
20. <next_params> -> eps
21. <expression_start> -> <value> <expression_next>
22. <expression_next> eps
23. <expression_next> OPERATOR <expression_start>
24. <value> ID

- 25.** <value> INT
- 26.** <value> DOUBLE
- 27.** <value> STRING
- 28.** <value> NONE
- 29.** <value> -> ID (<arg>)
- 30.** <value> -> INPUTS()
- 31.** <value> -> INPTUF()
- 32.** <value> -> INPUTI()
- 33.** <value> -> LEN (<value>)
- 34.** <value> -> SUBSTR (<value>, <value>, <value>)
- 35.** <value> -> ORD (<value>, <value>)
- 36.** <value> -> CHR (<value>)
- 37.** <arg> -> eps
- 38.** <arg> -> <value> <arg>

4.3. LL tabuľka

TODO

4.4. Precedentná tabuľka

(PRACUJE SA NA NEJ EŠTE)

+	-	*	/	>	==	!=	<=	>=	<	()	o	\$
R	R	S	S	R	R	R	R	R	R	S	R	S	R
R	R	S	S	R	R	R	R	R	R	S	R	S	R
R	R	R	R	R	R	R	R	R	R	S	R	S	R
R	R	R	R	R	R	R	R	R	R	S	R	S	R
S	S	S	S	F	F	F	F	F	F	S	R	S	R
S	S	S	S	F	F	F	F	F	F	S	R	S	R
S	S	S	S	F	F	F	F	F	F	S	R	S	R
S	S	S	S	F	F	F	F	F	F	S	R	S	R
S	S	S	S	F	F	F	F	F	F	S	R	S	R
S	S	S	S	S	S	S	S	S	S	S	S	S	F
R	R	R	R	R	R	R	R	R	R	F	R	F	R
R	R	R	R	R	R	R	R	R	R	F	R	F	R
S	S	S	S	S	S	S	S	S	S	S	F	S	A