

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Dokumentace projektu

pro předměty IFJ a IAL

Identifikace: Tým 55, varianta a/4/I

Vedoucí:	Vendula Poncová	xponco00	25%
Řešitelé:	Marek Salát	xsalat00	25%
	Tomáš Trkal	xtrkal00	25%
	Patrik Hronský	xhrons00	25%

Rozšíření: žádná

6. prosince 2011

Obsah

1	Úvod	1
2	Návrh řešení	1
3	Datové struktury	1
3.1	Abstraktní datové typy	1
3.2	Tabulka symbolů	1
3.3	Instrukce	1
4	Popis řešení	2
4.1	Lexikální analyzátor	2
4.2	Syntaktický analyzátor	2
4.3	Syntaktický analyzátor výrazů	2
4.4	Interpret	2
4.5	Algoritmy	2
4.5.1	Knuth-Moris-Prattův algoritmus	3
4.5.2	Merge sort	3
4.5.3	Binární vyhledávací strom	3
5	Vývoj a práce v týmu	3
5.1	Způsob práce v týmu	3
5.2	Rozdělení práce	3
6	Závěrem	3
A	Konečný automat	4
B	LL-gramatika	5
C	Metriky kódu	6

1 Úvod

Nějaké úvodní kecy.

2 Návrh řešení

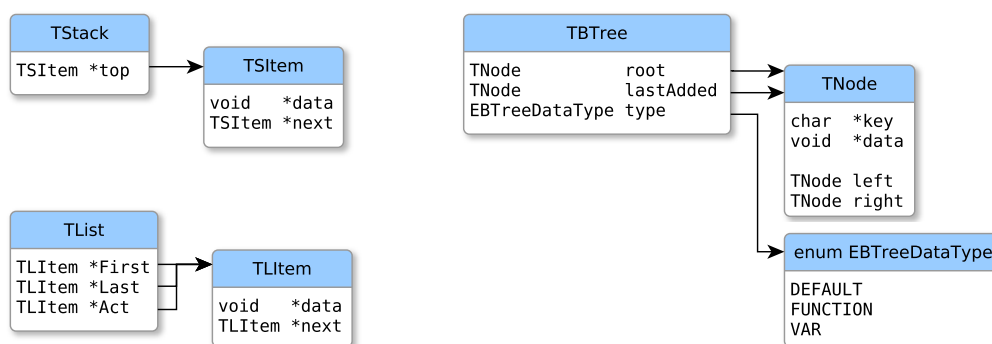
Obečné zamyšlení se nad problémy interpretu.

3 Datové struktury

Popis používaných datových struktur. Ilustrace na obrázcích.

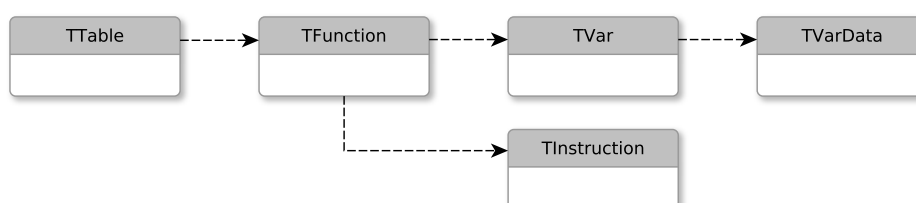
3.1 Abstraktní datové typy

Seznam, zásobník, binární strom - knihovna binary_tree.



Obrázek 1: Zásobník, seznam, binární vyhledávací strom.

3.2 Tabulka symbolů



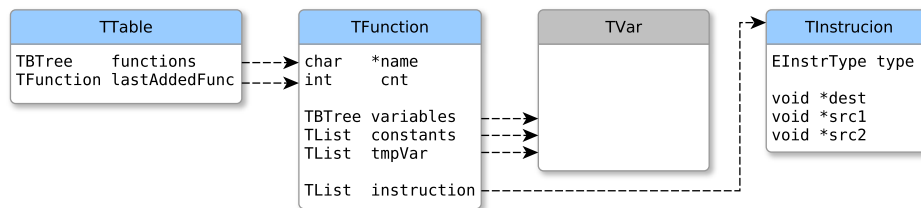
Obrázek 2: Model tabulky symbolů.

Tabulka symbolů Popis tabulky symbolů - knihovna table.

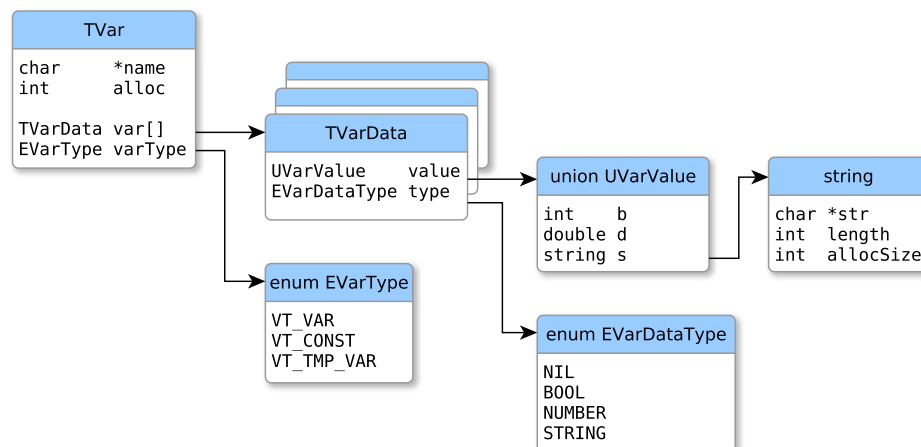
Proměnné Popis přístupu k datům. Zmínit datový typ string a knihovnu nad ním postavenou.

3.3 Instrukce

Jen popis datové struktury, bez výčtu instrukcí.



Obrázek 3: Datové struktury pro tabulku symbolů, funkce, instrukce.



Obrázek 4: Datové struktury pro proměnné a jejich data.

4 Popis řešení

Popis implementace.

4.1 Lexikální analyzátor

Tom!

Převážně se odkáže na konečný automat v příloze.

4.2 Syntaktický analyzátor

Marek!

LL gramatika v příloze.

4.3 Syntaktický analyzátor výrazů

Vendy!

4.4 Interpret

Všichni!

Součástí výčet typů instrukcí s popisem co dělají.

4.5 Algoritmy

Pro potřeby IAL.

4.5.1 Knuth-Moris-Prattův algoritmus

Vendy!

4.5.2 Merge sort

Patrik!

4.5.3 Binární vyhledávací strom

Marek!

Jen bych napsala, kde a proč se zvolila rekurze a kde ne, zanádalá na to, že AVL strom jsme implementovali a zas smazali. Proč je AVL strom lepší.

5 Vývoj a práce v týmu

To si napíšu sama :-P

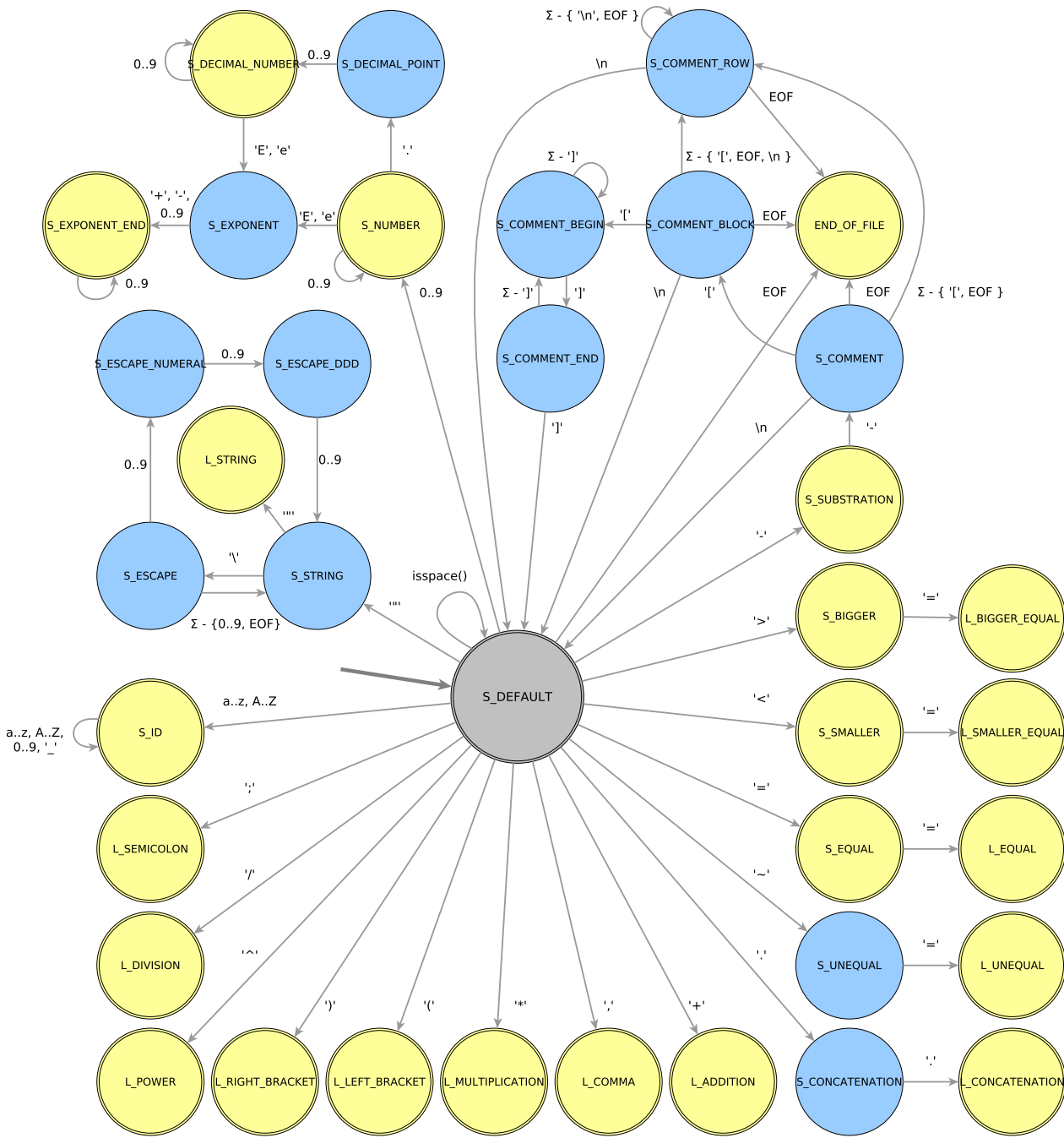
5.1 Způsob práce v týmu

5.2 Rozdělení práce

6 Závěrem

bla bla bla...

A Konečný automat



Obrázek 5: Graf konečného automatu pro lexikální analýzu.

B LL-gramatika

1	<program>	→	function <def_func>
2	<def_func>	→	main () <stat>end ; <EOF>
3	<def_func>	→	id (<params>) <stat>end <program>
4	<params>	→	eps
5	<params>	→	id <params_n>
6	<params_n>	→	eps
7	<params_n>	→	, id <params_n>
8	<stat>	→	<def_var><stat_list>
9	<def_var>	→	eps
10	<def_var>	→	local id <INIT>; <def_var>
11	<init>	→	eps
12	<init>	→	= <lit>
13	<lit>	→	literal
14	<stat_list>	→	eps
15	<stat_list>	→	<command>; <stat_list>
16	<command>	→	if expression then <stat_list>else <stat_list>end
17	<command>	→	while expression do <stat_list>end
18	<command>	→	return expression
19	<command>	→	write (expression <expression_n>)
20	<expression_n>	→	eps
21	<expression_n>	→	, expression <expression_n>
22	<command>	→	id = <assign>
23	<assign>	→	expression
24	<assign>	→	read (<lit>)
25	<assign>	→	id (<params>)
26	<var_params>	→	eps
27	<var_params>	→	<var><var_n>
28	<var>	→	<lit>
29	<var>	→	id
30	<var_n>	→	eps
31	<var_n>	→	, <var><var_n>

Tabulka 1: Pravidla LL gramatiky.

	function	main	()	end	;	id	,	local	=	literal	if	exp	then	else	while	return	write	read
<program>	1																		
<def_func>		2					3												
<stat>					8		8		8			8				8	8	8	
<params>				4			5												
<params_n>				6			7												
<def_var>					9		9		10			9				9	9	9	
<stat_list>					14		15					15			14	15	15	15	
<init>							11				12								
<lit>											13								
<command>							22					16				17	18	19	
<expression_n>				20			21												
<assign>							25						23						24
<var_params>				26			27				27								
<var>							19				28								
<var_n>				30			31												

Tabulka 2: Tabulka LL gramatiky.

C Metriky kódu

Počet souborů: ??? soubor

Počet řádků zdrojového textu: ??? řádků

Velikost statických dat: ???B

Velikost spustitelného souboru: ???B (systém Linux, 32 bitová architektura, při překladu bez ladicích informací)