

Formální jazyky a překladače tým 132, varianta II.

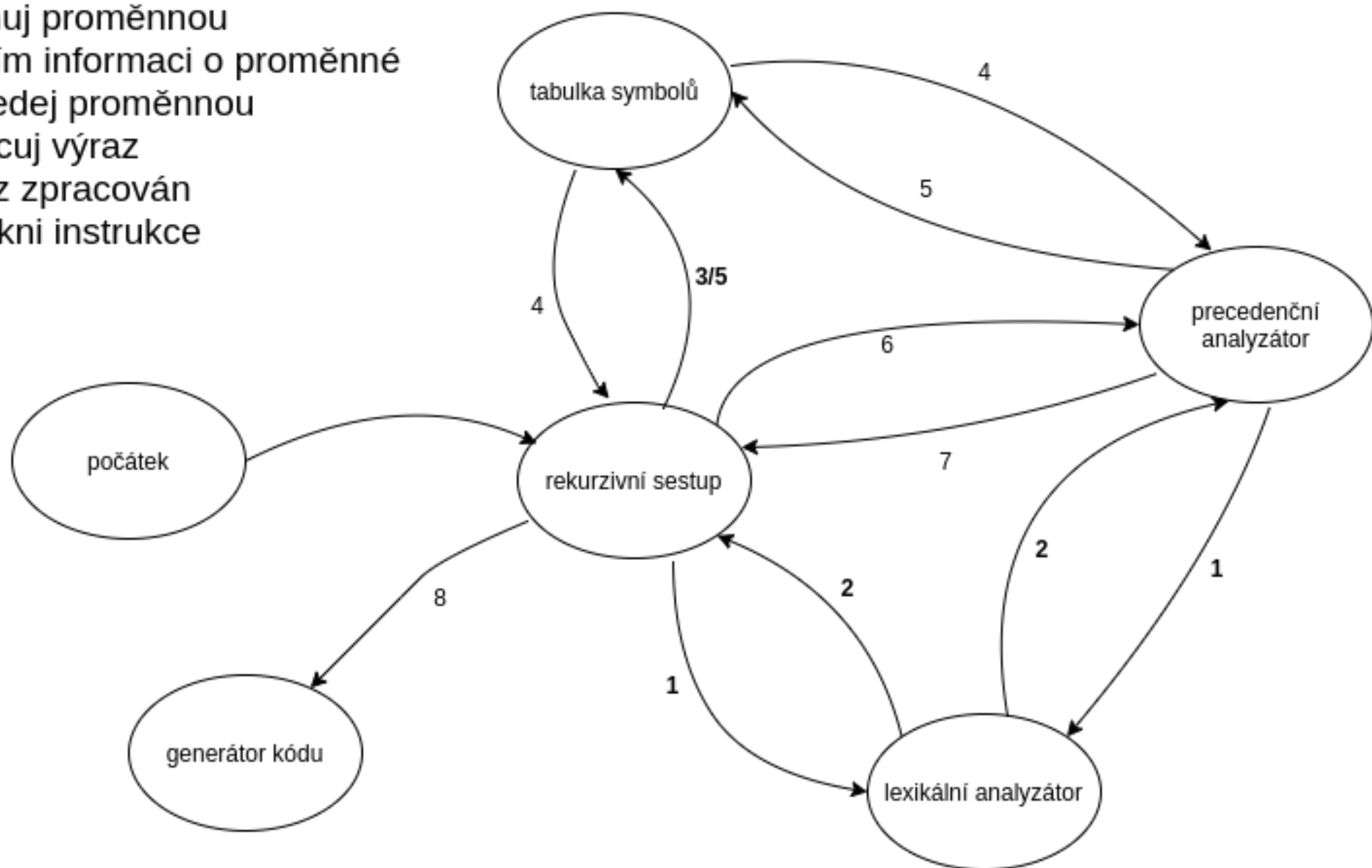
Ondřej Keprt, **xkeprt03**

Oleg Trofimov, xtrofi00

Maxim Gerasimov, xgeras00

Ondřej Sláma, xslama32

- 1) přečti lexikální token
- 2) vraćí lexikální token
- 3) definuj proměnnou
- 4) vraćím informaci o proměnné
- 5) vyhledej proměnnou
- 6) zpracuj výraz
- 7) výraz zpracován
- 8) vytiskni instrukce



Lexikální analyzátor

Lexikální analyzátor

- Načítá jednotlivé znaky ze standardního vstupu
- Zpracování pomocí konečného automatu
- V závislosti na konečném stavu, automat vytváří lexikální token
- Volán syntaktickou analýzou pomocí funkce `read_token`

Syntaktická analýza

Rekurzivní sestup

- Řídí celý překlad, volá ostatní části překladače
- Vykonává navrhnutou LL - gramatiku
- Rozhodování na základě LL – tabulky
- Zpracovává veškeré konstrukce jazyka, kromě výrazů
- Syntaktická analýza
- Sémantické kontroly
- Vytváření instrukcí, uložení do paměti překladače

Precedenční analýza

- Volána z rekurzivního sestupu
- Rozhodování na základě precedenční tabulky
- Komunikace s tabulkou symbolů
- Implementace pomocí rozšířeného zásobníkového automatu
- Vytváření instrukcí, uložení do paměti překladače

Tabulka symbolů

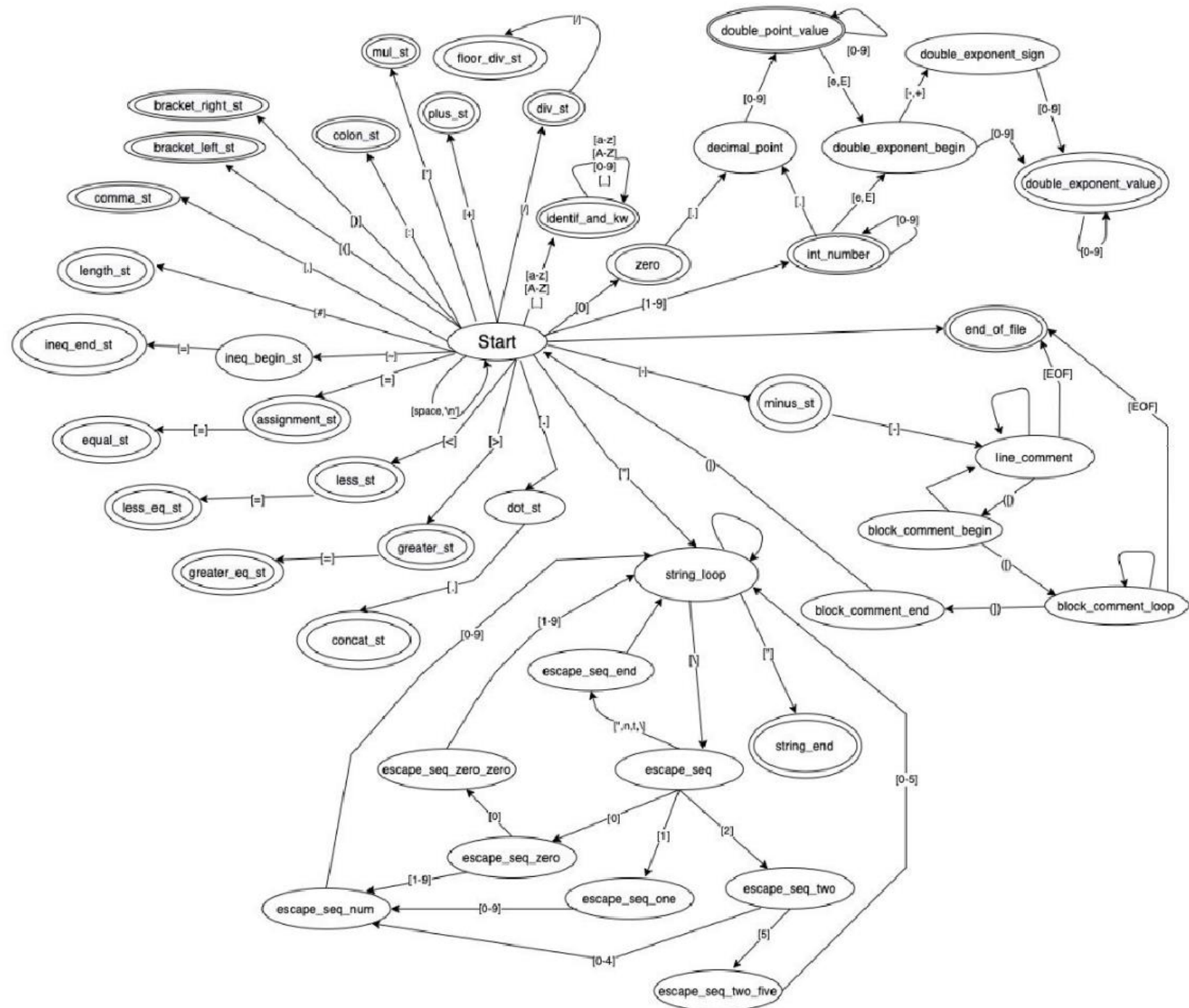
- Implementace pomocí tabulky s rozptýlenými položkami
- Synonyma uložena ve formě seznamu
- Stejná struktura pro funkci i proměnnou
- Lokální tabulky ve formě zásobníku

Generátor kódu

Generátor kódu

- Prochází vytvořené instrukce a generuje je na standardní výstup
- Generuje instrukce pro vestavěné funkce

Děkujeme za pozornost



LL - tabulka

	do	else	end	function	global	if	integer	local	nil	number	require	return	string	then	while	()	:	,	=	integer value	number value	string value	EOF	identif	#
<intro>											1															
<prog>				3	2																			5	4	
<prolog>											6															
<fce decl>					7																					
<type list>							9			9			9				8									
<type list2>			11	11	11	11		11				11			11		11		10					11	11	
<ret list>			13	13	13	13		13				13			13			12						13	13	
<fce def>				14																						
<param def list>																	15								16	
<param def list2>																	18		17							
<param>																									19	
<st list>		21	21			20		20				40			20										20	
<statement>						38		22							39										27	
<init>		23	23			23		23				23			23					24				23		
<init2>									26							26					26	26	26		25/26	26
<identif list>																		30	31							
<expression list>		35	35						34							34					34	34	34		34	34
<expression list2>		37	37			37		37				37			37				36						37	
<call fce>															41											
<value list>									43								42				43	43	43		43	
<value list2>																	45	44								
<value>									50												46	47	48		49	
<type>							51			52			53													
<assignment>									32							32					32	32	32		32/33	32
<expression>																										
<after id>																28			29	29						

Precedenční tabulka

[illegible]