

Interpret jazyka IFJ16

Tím 013, b/3/I:

Adam Šuba (vedúci)

Jakub Paliesek

Peter Šuhaj

Adrián Tóth

Organizácia a spolupráca

- Stretávky po prednáškach
- Verzovací systém Git, hostený na GitHubu
- Facebook chat – skupinový, súkromný
- Rozšírenie BOOLOP

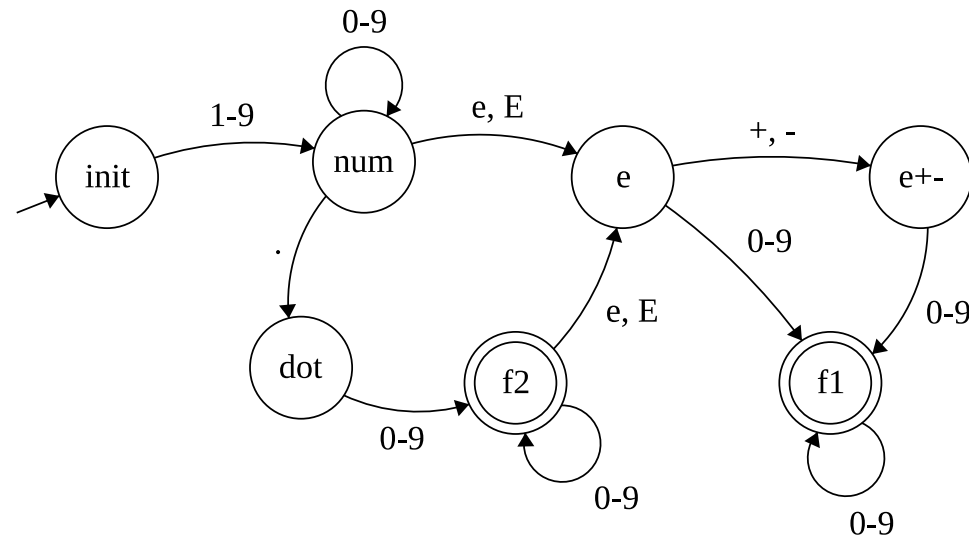
Implementácia

Lexikálny analyzátor

- Transformácia zdrojového súboru na reťazec tokenov
- Konečný automat
- Komunikácia so syntaktickou analýzou pomocou `get_token`
- Vracia:
 - Token (typ, atribút)
 - Chybový kód

type: `token_double`

attr: 41.99



Syntaktický analyzátor

- Rekurzívny zostup
- Implementácia LL gramatiky
- 2 priechody
- Fronta tokenov
- Sémantická akcia \rightarrow syntaktická kontrola
 - Prechod do precedenčnej syntaktickej analýzy
 - patrí identifikátor funkcii?
 - ifj16.print()

STAT \rightarrow if (*E*) STAT-COM else STAT-COM

STAT \rightarrow while (*E*) STAT-COM

AS-CA \rightarrow = ASSIGN

ASSIGN \rightarrow *E*

ASSIGN \rightarrow ID (FN-PLIST)

STAT \rightarrow ID AS-CA ;

AS-CA \rightarrow (FN-PLIST)



Ukážka LL gramatiky

Sémantická analýza

- 1. priechod
 - Vkladanie do tabuľky symbolov
 - Kontrola redefinície
- 2. priechod
 - Spracovanie tiel funkcií (aj vkladanie do TS)
 - **Generovanie trojadresného kódu**, pomocné premenné
 - Kontroly: prototypy funkcií, nedefinované symboly, typová kontrola

Precedenčná syntaktická analýza

- Spracovanie výrazov
- Precedenčná tabuľka
- Vlastná fronta tokenov
 - Zjednotenie literálov a premenných
 - Identifikátory → Odkazy do TS

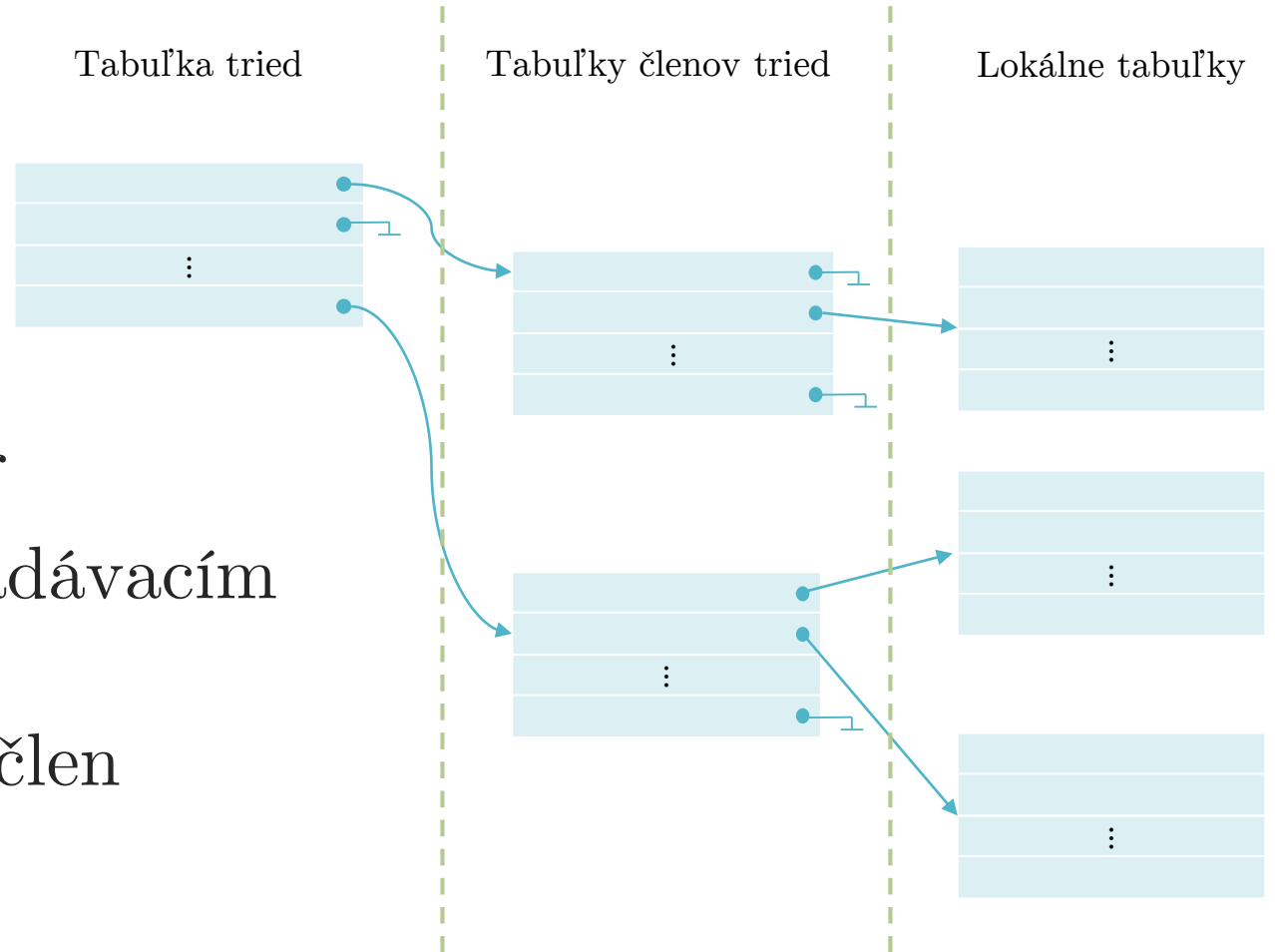


	+	-	*	/	<	>	<=	>=	==	!=	()	&&		!	id	\$
+	>	>	<	<	>	>	>	>	>	>	<	>	>	>	<	<	>
-	>	>	<	<	>	>	>	>	>	>	<	>	>	>	<	<	>
*	>	>	>	>	>	>	>	>	>	>	<	>	>	>	<	<	>
/	>	>	>	>	>	>	>	>	>	>	<	>	>	>	<	<	>
<	<	<	<	<					>	>	<	>	>	>	<	<	>
>	<	<	<	<					>	>	<	>	>	>	<	<	>
<=	<	<	<	<					>	>	<	>	>	>	<	<	>
>=	<	<	<	<					>	>	<	>	>	>	<	<	>
==	<	<	<	<	<	<	<	<	>	>	<	>	>	>	<	<	>
!=	<	<	<	<	<	<	<	<	>	>	<	>	>	>	<	<	>
(<	<	<	<	<	<	<	<	<	<	<	=	<	<	<	<	
)	>	>	>	>	>	>	>	>	>	>		>	>	>	>		>
&&	<	<	<	<	<	<	<	<	<	<	<	>	>	>	<	<	>
	<	<	<	<	<	<	<	<	<	<	<	>	>	>	>	<	>
!	>	>	>	>	>	>	>	>	>	>	<	>	>	>	<	<	>
id	>	>	>	>	>	>	>	>	>	>		>	>	>	>		>
\$	<	<	<	<	<	<	<	<	<	<	<		<	<	<	<	

Precedenčná tabuľka

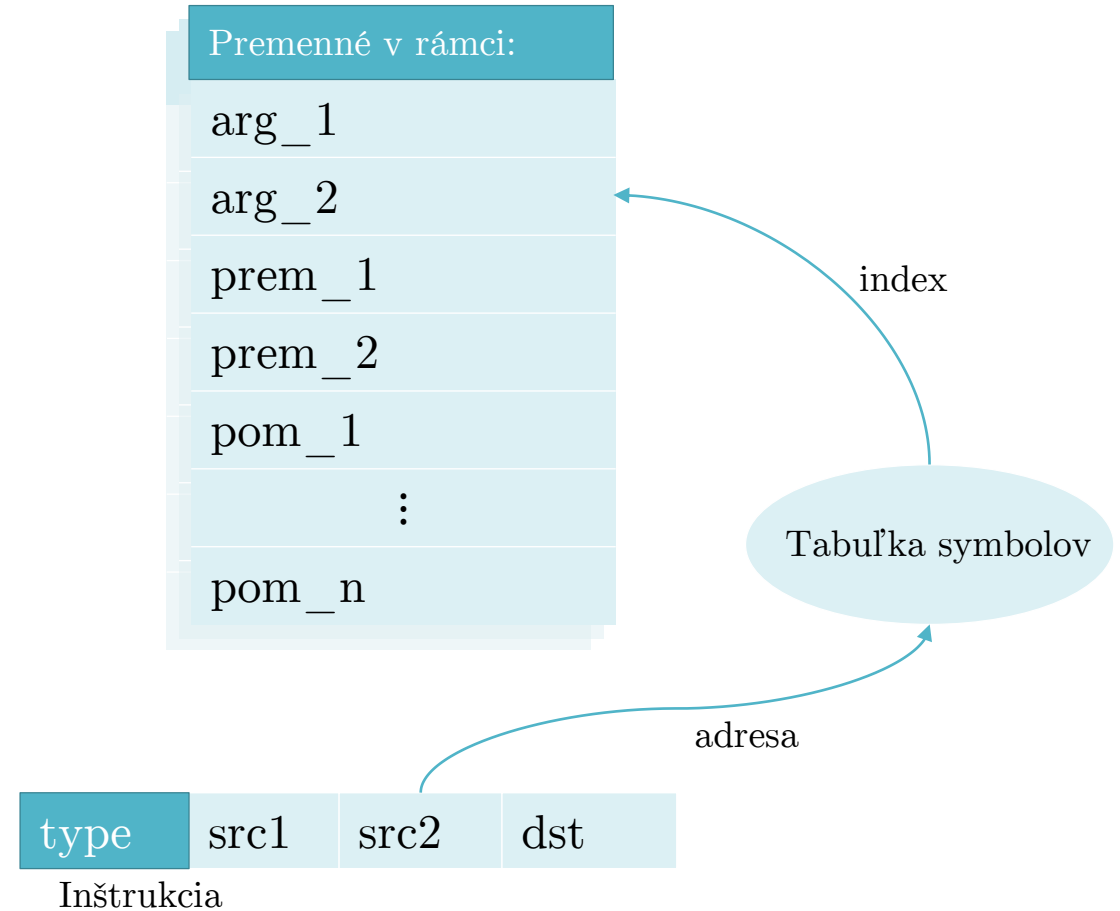
Tabuľka symbolov

- Vyhľadávací kľúč: identifikátor
- Implementácia binárnym vyhľadávacím stromom
- 2. a 3. úroveň – spoločný prvý člen štruktúry (op_t)



Interpret

- Vykonávanie inštrukcií
 - Globálna inštrukčná páska
 - Inštrukčné pásy funkcií
- Kontroly behových chýb
- Zásobník rámcov lokálnych premenných
- Indexovanie v rámcoch



Algoritmy

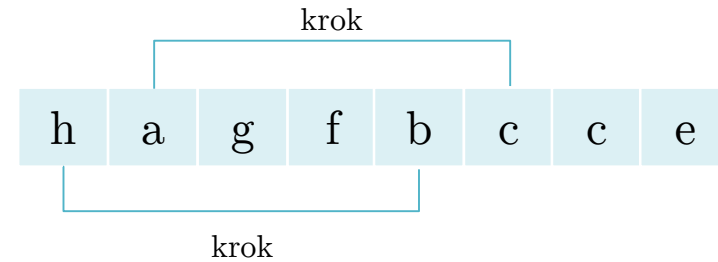
- Boyer-Mooreov algoritmus
- Shell sort
- Binárny vyhľadávací strom

Boyer-Moore algoritmus

- `ifj16.find()`
- 1. heuristika: pre všetky znaky predpočítané posuny, o koľko sa hľadaná vzorka posunie od aktuálneho priloženia
- Vzorka sa postupne prikladá zľava doprava
- Porovnávanie jednotlivých znakov opačne
- Zložitosť $O(mn)$

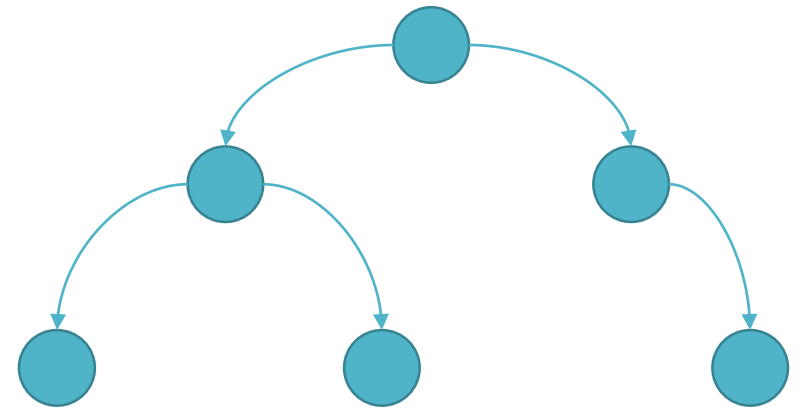
Shell sort

- `ifj16.sort()`
- „In situ“
- Nestabilný
- Usporiadáva prvky oddelené určitým krokom, ktorý sa postupne znižuje
- Naša implementácia: $\frac{N}{2}$, $\frac{N}{4}$, $\frac{N}{8}$, ...
- Zložitosť $O(n^2)$



Binárny vyhľadávací strom

- Priemerná dĺžka vyhľadania: $\log_2 n$
- Operácie
 - Vloženie prvku
 - Nájdenie prvku
 - Priebeh PostOrder – zrušenie tabuľky



Priestor na dotazy