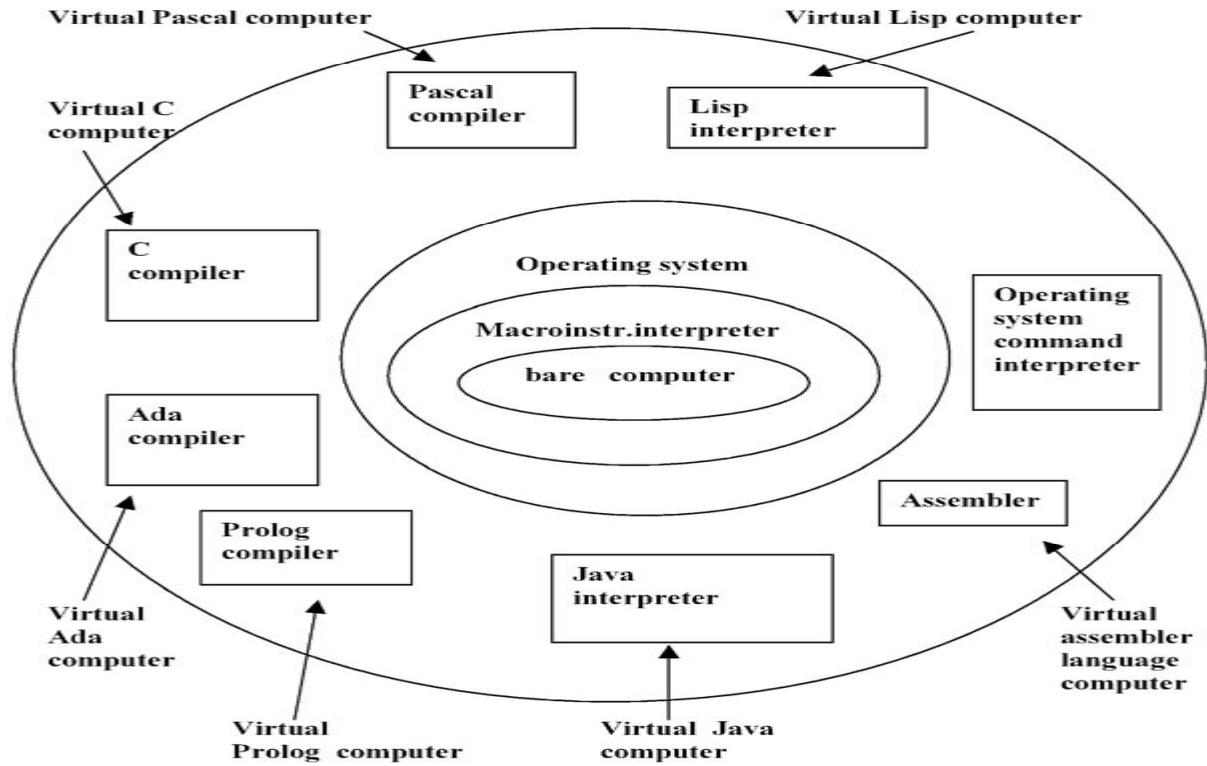


Virtuální počítač

- Uživatelský program
- Překladač programovacího jazyka
- Operační systém
- Interpret makroinstrukcí
- Procesor

Virtual computer



Překladač

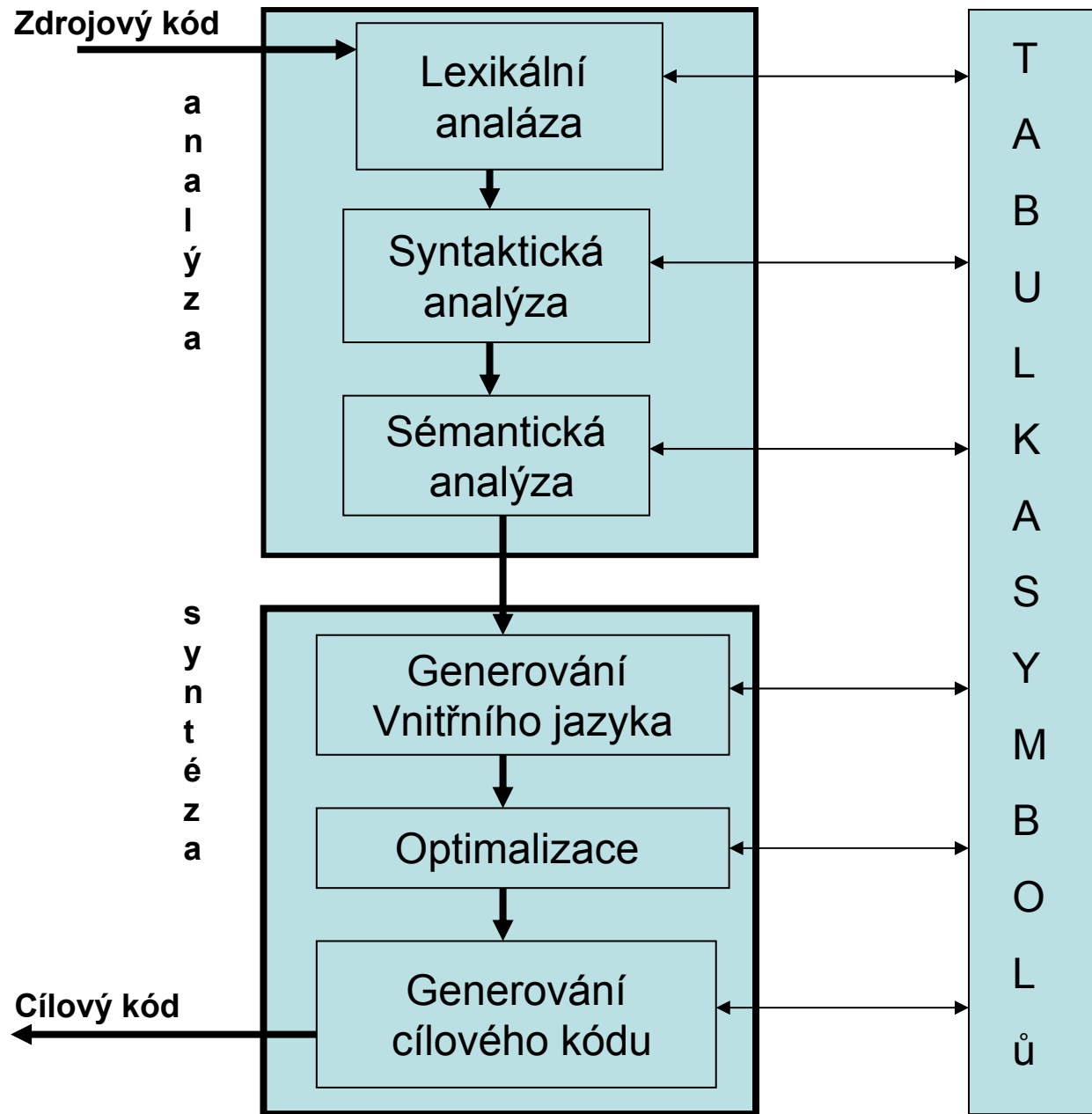
Překladač : Zdrojový jazyk → Cílový jazyk

Analytická část:

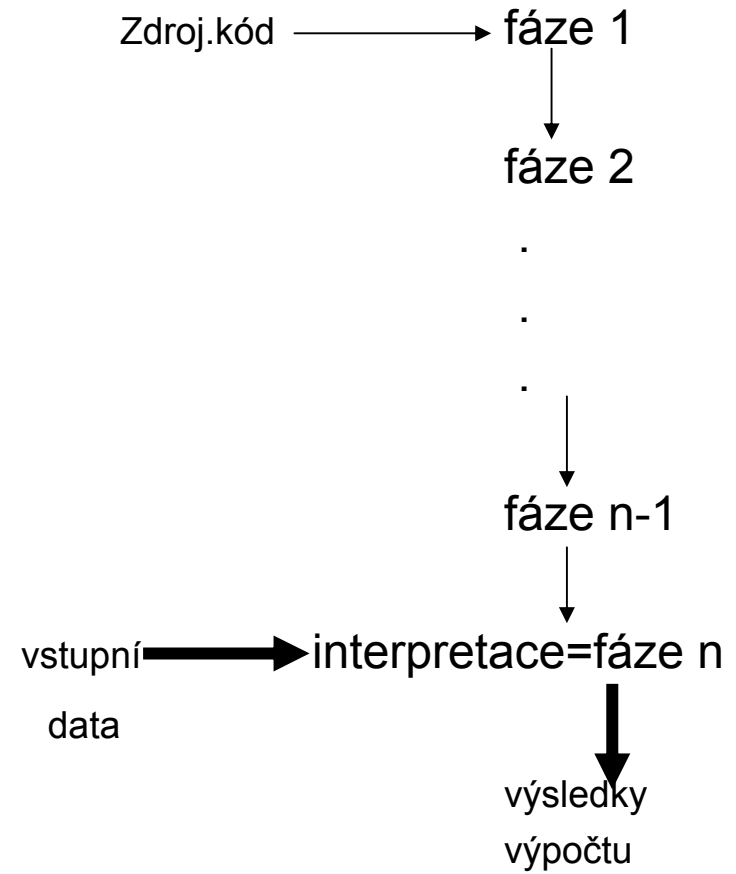
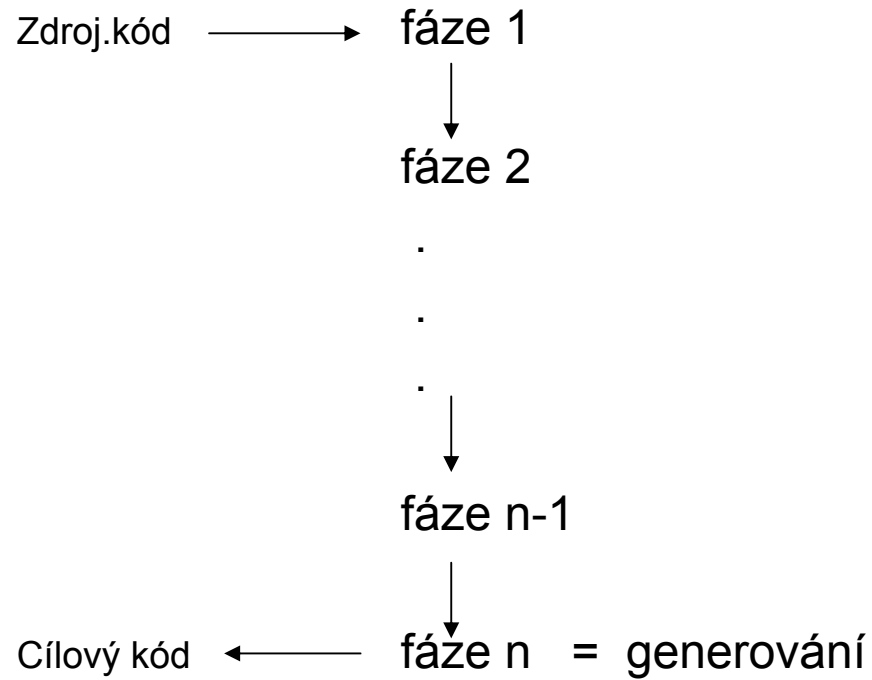
- Lexikální analýza
- Syntaktická a sémantická analýza

Syntetická část:

- Sémantické zpracování
- Generování cílového kódu



Kompilátor / Interpret



Interpret

V čistě jen interpretační podobě se nepoužívá (neefektivní)

Analytická část:

- Lexikální analýza
- Syntaktická a sémantická analýza

Syntetická část:

- Sémantické zpracování
- Interpretace

Lexikální analýza

1. Zakódování lexikálních elementů (do číselné podoby)
2. Vynechávání komentářů

Výhody: pevná délka symbolů pro další fáze zpracování

Např. { a = b * 3 + 5 ; /* poznámka */
c = a + 1 ;
}

při kódování:	ident	konst	+	*	=	{	}	;
na čísla	1	2	3	4	5	6	7	8

převede na: 1 adr a, 5, 1 adr b, 4, 2 3, 3, 2 5, 8, 1 adr c, 5, 1 adr a, 8, 7

Tj. vnitřní jazyk lexikálního analyzátoru (mezijazyk)

Lexikální analýza

Tvary lexikálních symbolů jsou popsatelné regulárními gramatikami:

$\langle \text{symbol} \rangle \rightarrow \langle \text{identifikátor} \rangle \mid$

$\langle \text{číslo} \rangle \mid$

$\text{class} \mid \text{if} \mid \text{while} \mid \dots \mid$

$+ \mid - \mid * \mid / \mid \dots \mid$

$(\mid) \mid \{ \mid \} \mid \dots \mid$

$=+ \mid ++ \mid == \mid \dots$

$\langle \text{identifikátor} \rangle \rightarrow \langle \text{identifikátor} \rangle a \mid \langle \text{identifikátor} \rangle b \mid \dots$

$\langle \text{identifikátor} \rangle 0 \mid \dots \langle \text{identifikátor} \rangle 9 \mid$

$a \mid b \mid \dots$

Lexikální analyzátor je konečným automatem

Syntaktický analyzátor

- Zjišťuje strukturu překládaného textu (syntaktický strom)
- Je založen na bezkontextových gramatikách (dovolují popisovat vnořované závorkové struktury)
- Vytváří derivační strom

$\langle \text{složený příkaz} \rangle \rightarrow \{ \langle \text{seznam příkazů} \rangle \}$ (1)

$\langle \text{seznam příkazů} \rangle \rightarrow \langle \text{příkaz} \rangle ; \langle \text{seznam příkazů} \rangle |$ (2)

$\langle \text{příkaz} \rangle ;$ (3)

$\langle \text{příkaz} \rangle \rightarrow \langle \text{proměnná} \rangle = \langle \text{výraz} \rangle$ (4)

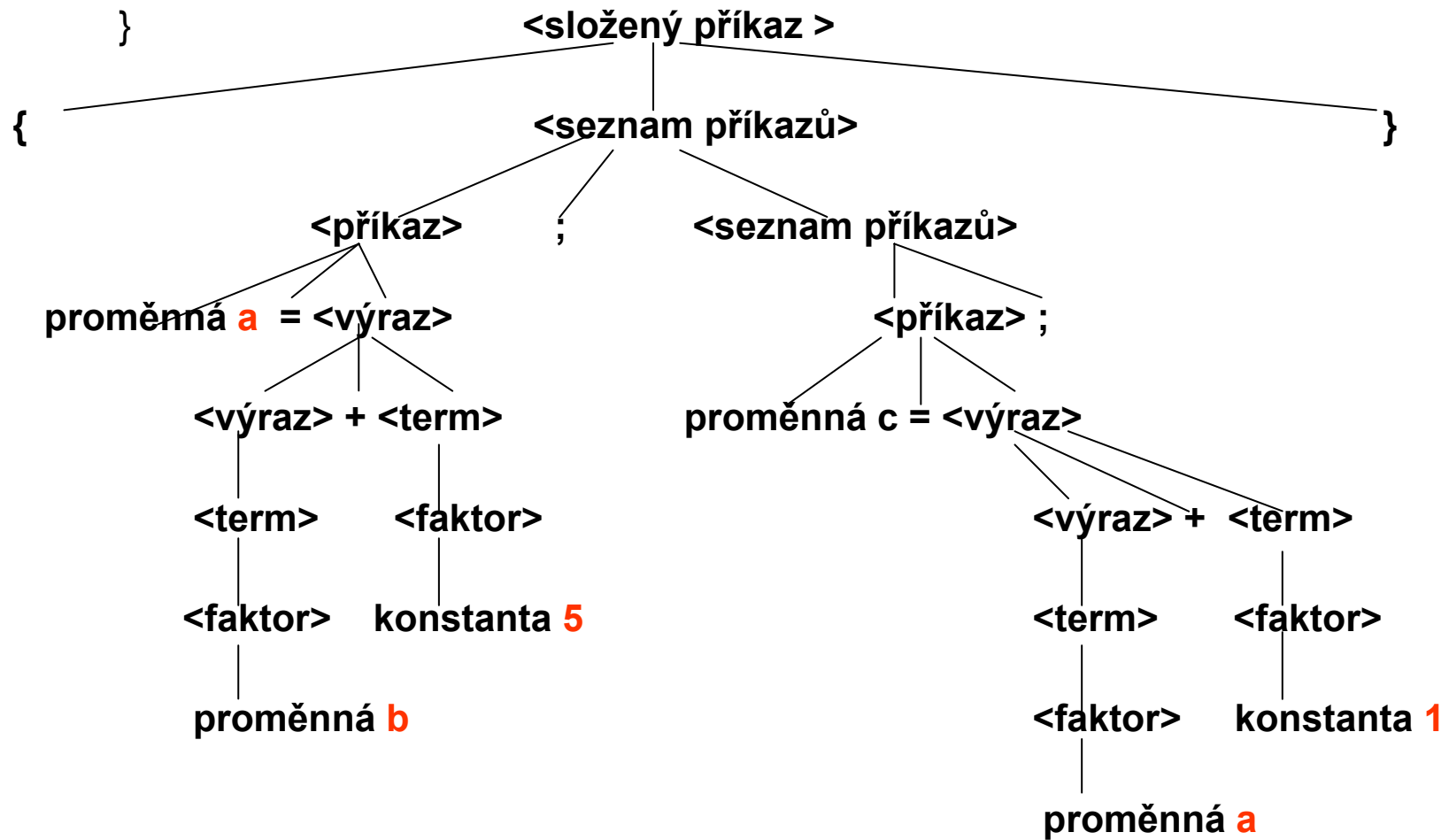
$\langle \text{výraz} \rangle \rightarrow \langle \text{výraz} \rangle + \langle \text{term} \rangle | \langle \text{výraz} \rangle - \langle \text{term} \rangle | \langle \text{term} \rangle$ (5)(6)(7)

$\langle \text{term} \rangle \rightarrow \langle \text{term} \rangle * \langle \text{faktor} \rangle | \langle \text{term} \rangle / \langle \text{faktor} \rangle | \langle \text{faktor} \rangle$ (8)(9)(10)

$\langle \text{faktor} \rangle \rightarrow (\langle \text{výraz} \rangle) | \text{proměnná} | \text{konstanta}$ (11)(12)(13)

Syntaktický analyzátor

Např. { a = b + 5 ; /* poznámka */
c = a + 1 ;
}



Sémantické zpracování

Souběžně či následně s rozpoznáváním syntaktické struktury jsou vyvolávány sémantické akce (sdružené s každým z gramatických pravidel), které převádí program do vnitřního jazyka překladače.

Formy vnitřních jazyků:

- a. **Postfixová notace (operátory následují za operandy)**
- b. **Prefixová notace (operá**
- c. **Víceadresové instrukce**

Např. $a = (b + c) * (a + c) ;$

LOA a	ST	PLUS a b pom1
LOV b	LOA a	PLUS a c pom2
LOV c	MUL	MUL pom1 pom2 pom3
PLUS	PLUS	ST a pom3
LOV a	LOV b	
LOV c	LOV c	
PLUS	PLUS	
MUL	LOV a	
ST	LOV c	

Optimalizace

```
PLUS a    b    pom1
PLUS a    c    pom2
MUL pom1 pom2 pom3
ST  a    pom3
```

Redukce počtu pomocných proměnných, eliminace nadbytečných přesunů mezi registry, optimalizace cyklů

```
PLUS a          b    pom1
PLUS a    c    pom2
MUL pom1 pom2 pom1
ST  a    pom1
```

Interpretace

LOA a dej adresu operandu a na vrchol zásobníku
LOV b dej obsah operandu b na vrchol zásobníku
LOV c dej obsah operandu c na vrchol zásobníku
PLUS sečti vrchol a podvrchol, výsledek vlož do zásobníku
LOV a dej obsah operandu a na vrchol zásobníku
LOV c dej obsah operandu c na vrchol zásobníku
PLUS sečti vrchol a podvrchol, výsledek vlož do zásobníku
MUL vynásob vrchol a podvrchol, výsledek vlož do zásobníku
ST ulož hodnotu z vrcholu zásobníku na adresu uloženou pod vrcholem

		c		c			
	b	b	b + c	a	a + c		
adr a	adr a	adr a	adr a	b + c	b + c	(a+b)*(c+d)	
1	2	3	4	5	6	7	8
				adr a	adr a	adr a	ulož na adr a

Generování

- Logicky jednoduché (expanze makroinstrukcí vnitřního jazyka)
- Prakticky komplikované (respektování instrukčních možností procesoru)

PLUS a b pom1

PLUS a c pom2

MUL pom1 pom2 pom1

ST a pom1

MOV a, R0

ADD b, R0

STO R0, p1

MOV a, R0

ADD c, R0

MUL p1, R0

STO R0, a