

Překladač z jazyka daného specifikací do tříadresového kódu

Semestrální projekt z předmětu Programovací jazyky a překladače, LS 2016/17

Navrhněte, vytvořte, otestujte a zdokumentujte překladač programovacího jazyka X, jehož gramatika odpovídá následující slovní specifikaci:

- Program může být buď prázdný, nebo obsahuje neomezený počet nevnořených funkcí.
- Každá funkce musí obsahovat typ, identifikátor, argumenty a definici bloku.
- Typ je integer, ovšem gramatika musí být navržena tak, aby ji bylo možné rozšířit i o další typy.
- Identifikátor je programátorem zvolené jméno.
- Funkční argumenty jsou vždy uzavřeny normálními závorkami, tj. '(' a ')'. Uvnitř závorek je jejich seznam, který může být prázdný, případně obsahovat jeden nebo více prvků oddělených čárkou.
- Každý prvek seznamu argumentů obsahuje typ a svůj identifikátor, tj. programátorem zvolené jméno.
- Definice bloku je vždy uzavřena složenými závorkami, tj. '{' a '}'.
- Blok může být prázdný, případně obsahovat jeden nebo více řídicích příkazů.
- Řídicím příkazem může být:
 - Definice bloku
 - Podmíněný příkaz
 - Příkaz while
 - Přiřazovací příkaz ukončený středníkem
 - Návrátový příkaz ukončený středníkem
- Podmíněný příkaz
 - Obsahuje definici bloku nebo
 - Začíná klíčovým slovem IF a otevírací závorkou '(', následovanou podmínkou, které je ukončena zavírací závorkou ')', následovanou definicí bloku a alternativou podmínky
 - Alternativa podmínky může být buď prázdná, nebo začínat klíčovým slovem ELSE, následovaným definicí bloku.
- Příkaz while
 - Začíná klíčovým slovem WHILE a otevírací závorkou '(', následovanou podmínkou, které je ukončena zavírací závorkou ')', následovanou definicí bloku.
- Podmínka je ve tvaru:
 podmínka relace výraz
nebo pouze
 výraz

Relacemi jsou oba typy ostrých a neostrých nerovností (<, <=, >, >=) a absolutní rovnost a nerovnost (==, !=)

Zdrojový kód a tříadresový výstup programu *vzor.x*:

Zdrojový kód	Výstupní tříadresový kód
<pre>// Komentar /* I toto je komentar */ int prvni(){} int druha(int a, int b){} int tretí(int a, int b){int c = a+b;} int ctvrta(int a, int b){int c = a+b-c*d;} int pata(){int c = a+(b-c)*d;} int sesta(){ if(3*5 <= x){ if(a < 10){ int b = 20; } else{ int b = 40; } } else{ int b = x; while(a < 10){ int a = 100; } return b; } }</pre>	<pre>T1 = a + b ; c = T1 ; T1 = a + b ; T2 = c * d ; T1 = T1 - T2 ; c = T1 ; T1 = b - c ; T1 = T1 * d ; T1 = a + T1 ; c = T1 ; T1 = 3 * 5 ; T1 = T1 ; T1 = T1 <= x ; if not T1 goto L1 ; T1 = a ; T1 = T1 < 10 ; if not T1 goto L2 ; b = 20 ; goto L3 ; L2 : b = 40 ; L3 : goto L4 ; L1 : b = x ; T1 = a ; T1 = T1 < 10 ; L5 : if not T1 goto L6 ; a = 100 ; goto L5 ; L6 : L4 :</pre>

Pro akceptování semestrálního projektu:

- Utilitou *make* musí být možno zkompilevat překladač do spustitelné podoby v prostředí cygwin,
- Překladač musí zpracovat příklad *vzor.x* přinejmenším na úrovni syntaktické analýzy.

Termín odevzdání: pátek, 26. 5. 2017, 12:00.