

Metody kompilacji

Laboratorium 6

Piotr Błaszyński

4 kwietnia 2017

Zadania (wyjaśnienie w dalszej części dokumentu):

- dodać kompilację instrukcji do wprowadzania i wypisywania wartości,
 - reguły gramatyki,
 - generowanie kodu wynikowego,
- dodać kompilację instrukcji warunkowych,
 - reguły gramatyki,
 - generowanie kodu wynikowego,

Wypisywanie na ekran i pobieranie danych z konsoli w MIPS realizuje się przy pomocy wywołań systemowych o określonych identyfikatorach.

Przykładowy kod wyświetlający liczbę całkowitą na ekran:

```
li $v0, 1
li $a0, 42
syscall
```

Przykładowy kod pobierający liczbę całkowitą do zmiennej:

```
.data
x: .word 0
.text
li $v0, 5
syscall
sw $v0, x
```

Przykładowy kod wypisujący łańcuch znaków:

```
.data
str: .asciiz "Tekst do wypisania."
.text
    li $v0, 4
    la $a0, str
    syscall
```

W rejestrze \$v0 wpisujemy identyfikator wywołania systemowego. Lista przydatnych wywołań wraz z rejestrami, w których jest wynik lub należy przekazać wartość (identyfikator wywołania, nazwa, parametr lub wartość zwracana):

id	nazwa	parametr lub wartość zwracana
1	print integer	\$a0 - liczba całkowita do wypisania
2	print float	\$f12 - liczba zmiennoprzecinkowa do wypisania
4	print string	\$a0 - adres łańcucha znaków zakończonego zerem (asciiz) do wypisania
5	read integer	\$v0 - odczytana z klawiatury liczba całkowita
6	read float	\$f0 - odczytana z klawiatury liczba zmiennoprzecinkowa

Instrukcja warunkowa (w prostszej formie) składa się z dwóch elementów: części z wyrażeniem warunkowym i bloku kodu wykonywanego przy spełnionym warunku. Reguły dla wersji z else należy opracować samodzielnie.

Wyjściowe reguły gramatyki dla instrukcji warunkowej:

```
if_expr
    : if_begin code_block {gen_etykiety_koncowej();}
if_begin
    : IF '(' cond_expr ')' {gen_warunku_i_skoku();}
```

Etykiety w kodzie MIPS oznacza się przy pomocy nazwy i symbolu dwukropka, skok do etykiety odbywa się przez użyciu instrukcji skoku i nazwy etykiety:

```
    b   LBL42
    li  $v0, 6
LBL42:
    li  $v0, 5
LBL43:    syscall
    sw  $v0, x
```

Każda nowa instrukcja warunkowa (również else, później również instrukcje pętli) powoduje wygenerowanie nowej etykiety (z kolejnym numerem). Etykiety należy wstawić na stos (oddzielny stos etykiet), w momencie dopasowania (akcja semantyczna dla f_{expr}) całej konstrukcji warunkowej należy zdjąć etykietę ze sto-

su i wstawić ja do kodu (do kodu wstawiać z dwukropkiem, przechowywać bez). Zestaw przykładowych instrukcji skoków (można użyć np. instrukcji obliczających wyrażenie (seq itd.)):

```
beq $t0,$t1,label #skok gdy rowne
bne $t0,$t1,label #skok gdy nie rowne
bge $t0,$t1,label #skok gdy $t0 wiekszy, rowny
bgt $t0,$t1,label #skok gdy $t0 wiekszy
ble $t0,$t1,label #skok gdy $t0 mniejszy, rowny
blt $t0,$t1,label #skok gdy $t0 mniejszy
```

Dla kodu:

```
z=3;
if (x < 5+4 )
{
    y = 2+4;
    z = 3*3;
}
z = z*3;
```

Należy wygenerować następujący kod (symbolicznie):

```
z=3;
if (!(x < 5+4) )
goto LBL5:
{
    y = 2+4;
    z = 3*3;
}
LBL5:z = z*3;
```

Należy wygenerować następujący kod (mnemoniki, symbolicznie):

```
li $t0,3
sw $t0,z
obliczenie 5+4 (4 linie)
lw $t2, x
lw $t3, result23 #wynik(5+4)
bge $t2,$t3,LBL5:
#6 linii dla y=2+4
#6 linii dla z = 3*3;
LBL5: lw $t0,z
#5 pozostałych linii dla z = z*3;
```

Uwaga, to jest jedna możliwa metoda generowania kodu dla instrukcji warunkowych, można np. dodać dodatkową etykietę, dzięki której będzie można skorzystać z naturalnych (odwrotnych do wyżej użytych) instrukcji skoku.