

Генериране на код. Тетради.
Генериране на команди за общи
програмни структури, преходи,
условен оператор, оператори за
цикъл

Тетради

- Тетрадите (quadruples) са междинна форма на представяне на генериран код
- Всички оператори могат да бъдат сведени до двуместни (бинарни) и едноместни (унарни)
- Тетрадите представят програмата като последователност от стъпки, в която резултатът от всяка стъпка се съхранява във временна променлива
- В тетрадата за всеки оператор има следните елементи:
 - операцията
 - операндите (един или два)
 - адрес на резултата

(оператор, операнд1, операнд2, резултат)

Примери за тетради

- Пример на тетрада за прост израз:

$B + C \Rightarrow (+, B, C, tmp)$

- По-сложни изрази се преобразуват в множество тетради.

Междинните резултати се съхраняват във временни променливи

$a * (9 + d)$

\Rightarrow

$(+, 9, d, tmp1)$

$(*, a, tmp1, tmp2)$

Представяне на общи програмни структури чрез тетради

- Математически и булеви изрази

$a + b \Rightarrow (+, a, b, tmp1)$

$a * (b + c) \Rightarrow (+, b, c, tmp1), (*, a, tmp1, tmp2)$

$a < b \Rightarrow (<, a, b, tmp)$

- Унарни оператори

$-a \Rightarrow (uminus, a, , tmp)$

- Присвояване

$a = a + b \Rightarrow (+, a, b, tmp), (:=, tmp, , a)$

- Декларации

$int\ a, b \Rightarrow -$

$int\ a = 5 \Rightarrow (:=, 5, , a)$

Достъп до елемент на масив и преобразуване на типове

- Достъп до елемент на масив

$a = x[i] \Rightarrow ([]=, x, i, tmp1), (:=, tmp1, , a)$

$x[i] = a \Rightarrow ([]=, x, i, tmp1), (:=, a, , tmp1)$

- Преобразуване на типове

(itof = 'int to float')

- $a = 1; b = a + 0.7$

\Rightarrow

$(:=, 1, , a)$

(itof, a, , tmp1)

(+, tmp1, 0.7, tmp2)

$(:=, tmp2, , b)$

Представяне на оператори за преход

- Безусловен преход

(jmp, <jump_address>, ,)

- Условен преход

(<, i, 5, t1) ; условие за преход

(jtrue, L10, t1,)

- Независимо дали преходът е условен или безусловен адресът на прехода е във втория аргумент
- Адресът на прехода представлява цяло число (номер)

Оператор for

• **for (i=0, i<5, i++) do <body> end**

=>

1 (:=, 0, , i) ; инициализация на i

2 (<, i, 5, t1) ; условие за край на цикъла

3 (jfalse, 7, t1, ,)

... <body of loop>

5 (+, i, 1, i) ; увеличаване на i

6 (jmp, 2, ,)

7

Оператор if-else

- **if (a < b) then c = 1 else c = 2**

=>

1 (<, a, b, t1)

2 (jfalse, 5, t1,)

3 (:=, 1, , c)

4 (jmp, 6, ,)

5 (:=, 2, , c)

6

Оператор while

- **while (a < b) do a = a +1**

=>

1 (<, a, b, tmp1)

2 (jfalse, 6, tmp1, ,)

3 (+, a, 1, tmp2)

4 (:=, tmp2, , a)

5 (jmp, 1, ,)

6

Упражнение

- Дадена е следната програма:

```
int a = 2, b = 8, c = 4, d;
```

```
for (j=0; j<=10; j++){
```

```
  a = a * (j* (b/c));
```

```
  d = a * (j* (b/c));
```

```
}
```

- Да се покаже генерираният междинен код във вид на теради