

## Generacja kodu pośredniego

- Elastyczność generacji kodu docelowego
- Wykorzystanie ogólnych algorytmów optymalizacji
  - Postać trójadresowa:

**$X := Y$  operator  $Z$**

- Reprezentacje:
  - Kod czwórkowy – operator, arg1, arg2, wynik
  - Triplety (trójki) – operator, arg1, arg2

### Zestaw operacji

1. Przypisanie wyniku operatora 2-argumentowego:  **$x := y \text{ op } z$**

2. Przypisanie wyniku operatora 1-argumentowego:  **$x := \text{op } y$**

## Zestaw operacji c.d.

3. Kopiowanie wartości

**x := y**

4. Skok bezwarunkowy

**goto L**

5. Skok warunkowy:

**if x relop y goto L**

6. Instrukcje procedur:

**param x**

**call p,n**

**return y**

7. Przypisanie tablicowe:

**x := y[i]**

**x[i] := y**

8. Przypisanie wskaźnika:

**x := &y**

(ustaw x na lokalizację y)

**x := \*y**

(ustaw x na wartość wskazywaną przez y)

**\*x := y**

(ustaw wartość wskazywaną przez x na y)

## Generacja kodu pośredniego - zagadnienia

- Deklaracje zmiennych (offset)
  - a) zmienne globalne
  - b) zmienne w blokach
- Instrukcja podstawienia
- Wyrażenia logiczne
- Warunki i pętle
- Instrukcja wyboru
- Wywołania procedur

### Akcje semantyczne

Atrybutami symboli niech będą **adres** oraz **kod**.  
Procedura *nowatymcz* zwraca w kolejnych  
wywołaniach nowe nazwy  $t_1, t_2, \dots$

$S ::= id := E$	$E ::= E_1 + E_2$
$E ::= E_1 * E_2$	$E ::= - E_1$
$E ::= (E_1)$	$E ::= id$

## Akcje semantyczne c.d.

Produkcja	Akcja
$S ::= id := E$	$S.kod = E.kod \wedge \text{gen}(id.adres, ":", E.adres)$
$E ::= E_1 + E_2$	$E.adres = \textit{nowatymcz}$ $E.kod = E_1.kod \wedge E_2.kod \wedge \text{gen}(E.adres, ":", E_1.adres, "+", E_2.adres)$
$E ::= (E_1)$	$E.adres = E_1.adres; E.kod = E_1.kod$
$E ::= id$	$E.adres = id.adres; E.kod = ""$

Przykład dla instrukcji pętli:      **$S ::= \text{while } E \text{ do } S_1$**

$S.początek = \textit{nowaEtykieta}; S.koniec = \textit{nowaEtykieta};$

$S.kod = \text{gen}( S.początek, ":" ) \wedge E.kod \wedge \text{gen}( "if ", E.adres, "=0 goto", S.koniec ) \wedge S1.kod \wedge$   
 $\text{gen}( "goto ", S.początek ) \wedge \text{gen}( S.koniec, ":" )$

## Oszczędne używanie zmiennych

- Rozpatrzmy instrukcję  $x := a*b+c*d-e*f$
- Przyjmijmy, że zmienne tymczasowe mają ten sam rozmiar
- Utrzymujemy licznik zmiennych tymczasowych **L**.
- Za każdym razem gdy użyjemy zmiennej tymczasowej jako argumentu zmniejszamy licznik o jeden ( $L := L-1$ )
- Gdy potrzebujemy nowej zmiennej użyjemy nazwy **\$L** oraz  $L:=L+1$  .

<i>Instrukcja</i>	<i>Wartość L</i>	
	0	
\$0:=a*b	1	
\$1:=c*d	2	
\$0:=\$0+\$1	1	<i>użyto dwu zmiennych jako argumentów</i>
\$1:=e*f	2	
\$0:=\$0-\$1	1	
x := \$0	0	

## Kod pośredni - realizacje

- C Intermediate Language (<http://manju.cs.berkeley.edu/cil/>)
- Java Bytecode
- Register Transfer Language ( kompilatory GNU )
- Low-Level Virtual Machine (<http://www.llvm.org>)
- GENERIC ( podzbiór reprezentacji GIMPLE dla kompilatorów GNU)