

Modele de calcul

Masini de calcul abstracte idealizate, simplificate
organizarea masinii

- actiuni (instructiuni) + valorile cu care operam (operanzii)

Masina cu registri

- registri: valorile cu care operam
- instructiuni:
operanzii: registri de intrare , registrul de iesire

Masina cu stiva *calculator de tip stiva*

- operanzii – in stiva
- instructiuni: opereaza asupra varfului stivei

Masina cu registri

Exemplu:

Fie expresia aritmetica

A+B

Codul pentru masina cu registri echivalent expresiei aritmetice date este:

load A, r0

load B, r1

add r0, r1, r2

Pentru simplificare, noi vom folosi doar exemple de tipul:

Dandu-se o expresie aritmetica

cu operatorii: ... (add, sub, mul, div), stim sa determinam:

codul pentru masina cu registri

Operatorii pentru exemplele cu care vom lucra:

load, add, sub, mul, div, store

Masina cu stiva

Modelul se bazeaza pe faptul ca orice expresie poate fi scrisa in forma poloneza postfixata.

Exemplu:

Fie expresia aritmetica

B + C - D

Forma poloneza postfixata este

B C + D -

Codul pentru masina cu stiva echivalent expresiei aritmetice date este:

push B

push C

add

push D

sub

Pentru simplificare, noi vom folosi doar exemple de tipul:

Dandu-se o expresie aritmetica, stim sa determinam:

- formă poloneza postfixată
- codul pentru masina cu stiva

In exemplele cu care vom lucra vom considera o **masina cu stiva** si 6 operatii: push, pop, add, sub, mul, div

Masina cu stiva / masina cu registri

Exemplu: $2+3$

- Masina cu stiva

Push 2

Push 3

Add

- Masina cu registri

Load 2, r0

Load 3, r1

Add r0, r1, r2

Alte modele de calcul

+ combinatii

- Masina cu accumulator
 - accumulator : un registru special
 - operand implicit
 - rezultate stocate in accumulator
 - + ...
- Masina cu accumulator si stiva
 - stiva + accumulator (registru special)

... ...

Java bytecode : exemplu

i = j + k;	1	ILOAD j // i = j + k
if (i == 3)	2	ILOAD k
k = 0;	3	IADD
else	4	ISTORE i
j = j - 1;	5	ILOAD i // if (i == 3)
	6	BIPUSH 3
	7	IF_ICMPEQ L1
	8	ILOAD j // j = j - 1
	9	BIPUSH 1
	10	ISUB
	11	ISTORE j
	12	GOTO L2
	13 L1:	BIPUSH 0 // k = 0
	14	ISTORE k
	15 L2:	