Relatório da Etapa 5 - Turma A (Recuperação)

Prof. Lucas Mello Schnorr

November 22, 2013

Contents

1 g1 ../../ga/compiladores-etapa5/

1.1 input/e00.iks

- Valgrind with input/e00.iks says (w,r,c,l)=(0,0,0,0)

```
/* global variable */
int:x;
int:main()
{
  x = 0;
}
```

- Generated ILOC

```
L2: loadI 0 => r2
L3: storeAI r2 => bss,0
```

1.2 input/e01.iks

- Valgrind with input/e01.iks says (w,r,c,l)=(0,0,0,0)

```
/* local variable */
int:main()
int:x;
{
  x = 0;
}
```

- Generated ILOC

```
L2: loadI 0 => r2
L3: storeAI r2 => rarp,4
```

1.3 input/e02.iks

- Valgrind with input/e02.iks says (w,r,c,l)=(0,0,0,0)

```
/* global variable with offset */
int:x;
int:y;
int:main()
{
  y = 0;
}
```

- Generated ILOC

```
L2: loadI 0 => r2
L3: storeAI r2 => bss,4
```

1.4 input/e03.iks

- Valgrind with input/e03.iks says (w,r,c,l)=(0,0,0,0)

```
/* local variable with offset */
int:main()
int:x;
int:y;
{
  y = 0;
}
```

- Generated ILOC

```
L2: loadI 0 => r2
L3: storeAI r2 => rarp,4
```

1.5 input/e04.iks

- Valgrind with input/e04.iks says (w,r,c,l)=(0,0,0,0)

```
/* global arranjo */
int:x;
int:y[10];
int:main()
{
  y[0] = 0;
  y[1] = 0;
}
```

- Generated ILOC

```
tip: 11
tip: 11
L3: loadI 0 => r3
L4: storeAI r3 => bss,4
L7: loadI 0 => r6
L8: storeAI r6 => bss,8
```

1.6 input/e05.iks

- Valgrind with input/e05.iks says (w,r,c,l)=(0,0,0,256)

```
/* global arranjo with expression */
int:x;
int:y[10];
int:main()
{
  y[x+1] = 0;
  y[x*x] = 0;
}
```

- Generated ILOC

```
L1: loadAI bss,0 => r1
L2: loadI 1 => r2
L3: add r1,r2 => r3
L8: add r3,4 => r7
L6: loadI 0 => r6
L7: storeAO r6 => bss,r7
L9: loadAI bss,0 => r8
L10: loadAI bss,0 => r9
L11: mul r8,r9 => r10
L16: add r10,4 => r14
L14: loadI 0 => r13
L15: storeAO r13 => bss,r14
```

1.7 input/e06.iks

- Valgrind with input/e06.iks says (w,r,c,l)=(0,0,0,0)

```
/* arithm. */
int:x;
int:main()
{
  x = 0*1+2*3;
}
```

- Generated ILOC

```
L2: loadI 0 => r2
L3: loadI 1 => r3
L4: loadI 2 => r4
L5: loadI 3 => r5
L6: mul r4,r5 => r6
L7: add r3,r6 => r7
L8: mul r2,r7 => r8
L9: storeAI r8 => bss,0
```

1.8 input/e07.iks

- Valgrind with input/e07.iks says (w,r,c,l)=(0,0,0,0)

```
/* arithm. */
int:x;
int:main()
{
  x = 0*(1+2)*3;
}
```

- Generated ILOC

```
L2: loadI 0 => r2
L3: loadI 1 => r3
L4: loadI 2 => r4
L5: add r3,r4 => r5
L6: loadI 3 => r6
L7: mul r5,r6 => r7
L8: mul r2,r7 => r8
L9: storeAI r8 => bss,0
```

1.9 input/e08.iks

- Valgrind with input/e08.iks says (w,r,c,l)=(0,0,0,0)

```
int:x;
int:main()
{
  x = 0;
}
```

- Generated ILOC

```
L2: loadI 0 => r2
L3: storeAI r2 => bss,0
```

1.10 input/e09.iks

- Valgrind with input/e09.iks says (w,r,c,l)=(0,0,0,0)

```
int:x;
int:main()
{
  if (x) then {
    x = 0;
  }
}
```

- Generated ILOC

```
L1: loadAI bss,0 => r1
L6: crb r1 => L3,L5
L3: loadI 0 => r3
L4: storeAI r3 => bss,0
L5: nop
```

1.11 input/e10.iks

- Valgrind with input/e10.iks says (w,r,c,l)=(0,0,0,0)

```
int:x;
int:main()
{
  if (x) then {
    x = 0;
  }else{
    x = 1;
  }
}
```

- Generated ILOC

```
L1: loadAI bss,0 => r1
L9: crb r1 => L3,L6
L6: loadI 1 => r5
L7: storeAI r5 => bss,0
L10: jumpI  => L8
L3: loadI 0 => r3
L4: storeAI r3 => bss,0
L8: nop
```

1.12 input/e11.iks

- Valgrind with input/e11.iks says (w,r,c,l)=(0,0,0,0)

```
int:x;
int:main()
{
  do {
    x = 0;
  }while (x);
}
```

- Generated ILOC

```
L2: loadI 0 => r2
L3: storeAI r2 => bss,0
L4: loadAI bss,0 => r3
L6: crb r3 => L2,L5
L5: nop
```

1.13 input/e12.iks

- Valgrind with input/e12.iks says (w,r,c,l)=(0,0,0,0)

```
int:x;
int:main()
{
  while (x) do {
    x = 0;
  }
}
```

- Generated ILOC

```
L1: loadAI bss,0 => r1
L6: crb r1 => L3,L7
L3: loadI 0 => r3
L4: storeAI r3 => bss,0
L5: jumpI  => L1
L7: nop
```

1.14 input/e13.iks

- Valgrind with input/e13.iks says (w,r,c,l)=(0,0,0,0)

```
int:x;
int:main()
{
  if (x || x && x) then {
    x = 0;
  }
}
```

- Generated ILOC

```
L1: loadAI bss,0 => r1
L5: cbr r1 => L7,L2
L2: loadAI bss,0 => r2
L4: cbr r2 => L3,L9
L3: loadAI bss,0 => r3
L10: crb r3 => L7,L9
L7: loadI 0 => r7
L8: storeAI r7 => bss,0
L9: nop
```

1.15 input/e14.iks

- Valgrind with input/e14.iks says (w,r,c,l)=(0,0,0,0)

```
/* global variable */
int:x[10][10];
int:y;
int:main()
{
  y = 0;
}
```

- Generated ILOC

```
L2: loadI 0 => r2
L3: storeAI r2 => bss,400
```

2 g2 ../../gb/etapa5-gb/

2.1 input/e00.iks

- Valgrind with input/e00.iks says (w,r,c,l)=(0,0,0,0)

```
/* global variable */
int:x;
int:main()
{
  x = 0;
}
```

- Generated ILOC

```
loadI   0       =>      r1
loadI   0       =>      r0
addI    r0, bss =>      r0
store   r1      =>      r0
```

2.2 input/e01.iks

- Valgrind with input/e01.iks says (w,r,c,l)=(0,0,0,0)

```
/* local variable */
int:main()
int:x;
{
  x = 0;
}
```

- Generated ILOC

```
loadI   0       =>      r1
loadI   0       =>      r0
addI    r0, rarp        =>      r0
store   r1      =>      r0
```

2.3 input/e02.iks

- Valgrind with input/e02.iks says (w,r,c,l)=(0,0,0,0)

```
/* global variable with offset */
int:x;
int:y;
int:main()
{
  y = 0;
}
```

- Generated ILOC

```
loadI   0       =>      r1
loadI   4       =>      r0
addI    r0, bss =>      r0
store   r1      =>      r0
```

2.4 input/e03.iks

- Valgrind with input/e03.iks says (w,r,c,l)=(0,0,0,0)

```
/* local variable with offset */
int:main()
int:x;
int:y;
{
  y = 0;
}
```

- Generated ILOC

```
loadI   0       =>      r1
loadI   0       =>      r0
addI    r0, rarp        =>      r0
store   r1      =>      r0
```

2.5 input/e04.iks

- Valgrind with input/e04.iks says (w,r,c,l)=(0,0,0,0)

```
/* global arranjo */
int:x;
int:y[10];
int:main()
{
  y[0] = 0;
  y[1] = 0;
}
```

- Generated ILOC

```
loadI   0       =>      r4
loadI   0       =>      r0
i2i     r0      =>      r2
multI   r2, 4   =>      r3
addI    r3, 4   =>      r1
addI    r1, bss =>      r1
store   r4      =>      r1
loadI   0       =>      r9
loadI   1       =>      r5
i2i     r5      =>      r7
multI   r7, 4   =>      r8
addI    r8, 4   =>      r6
addI    r6, bss =>      r6
store   r9      =>      r6
```

2.6 input/e05.iks

- Valgrind with input/e05.iks says (w,r,c,l)=(0,0,0,0)

```
/* global arranjo with expression */
int:x;
int:y[10];
int:main()
{
  y[x+1] = 0;
  y[x*x] = 0;
}
```

- Generated ILOC

```
loadI   0       =>      r7
loadI   0       =>      r0
addI    r0, bss =>      r0
load    r0      =>      r1
loadI   1       =>      r2
add     r1, r2  =>      r3
i2i     r3      =>      r5
multI   r5, 4   =>      r6
addI    r6, 4   =>      r4
addI    r4, bss =>      r4
store   r7      =>      r4
loadI   0       =>      r16
loadI   0       =>      r8
addI    r8, bss =>      r8
load    r8      =>      r9
loadI   0       =>      r10
addI    r10, bss        =>      r10
load    r10     =>      r11
mult    r9, r11 =>      r12
i2i     r12     =>      r14
multI   r14, 4  =>      r15
addI    r15, 4  =>      r13
addI    r13, bss        =>      r13
store   r16     =>      r13
```

2.7 input/e06.iks

- Valgrind with input/e06.iks says (w,r,c,l)=(0,0,0,0)

```
/* arithm. */
int:x;
int:main()
{
  x = 0*1+2*3;
}
```

- Generated ILOC

```
loadI   0       =>      r1
loadI   1       =>      r2
mult    r1, r2  =>      r3
loadI   2       =>      r4
loadI   3       =>      r5
mult    r4, r5  =>      r6
add     r3, r6  =>      r7
loadI   0       =>      r0
addI    r0, bss =>      r0
store   r7      =>      r0
```

2.8 input/e07.iks

- Valgrind with input/e07.iks says (w,r,c,l)=(0,0,0,0)

```
/* arithm. */
int:x;
int:main()
{
  x = 0*(1+2)*3;
}
```

- Generated ILOC

```
loadI   0       =>      r1
loadI   1       =>      r2
loadI   2       =>      r3
add     r2, r3  =>      r4
mult    r1, r4  =>      r5
loadI   3       =>      r6
mult    r5, r6  =>      r7
loadI   0       =>      r0
addI    r0, bss =>      r0
store   r7      =>      r0
```

2.9 input/e08.iks

- Valgrind with input/e08.iks says (w,r,c,l)=(0,0,0,0)

```
int:x;
int:main()
{
  x = 0;
}
```

- Generated ILOC

```
loadI   0       =>      r1
loadI   0       =>      r0
addI    r0, bss =>      r0
store   r1      =>      r0
```

2.10 input/e09.iks

- Valgrind with input/e09.iks says (w,r,c,l)=(0,0,0,0)

```
int:x;
int:main()
{
  if (x) then {
    x = 0;
  }
}
```

- Generated ILOC

```
        loadI   0       =>      r0
        addI    r0, bss =>      r0
        load    r0      =>      r1
        loadI   1       =>      r5
        cmp_GE  r1, r5  ->      r4
        cbr     r4      ->      L0, L1
L0:     nop
        loadI   0       =>      r3
        loadI   0       =>      r2
        addI    r2, bss =>      r2
        store   r3      =>      r2
L1:     nop
```

2.11 input/e10.iks

- Valgrind with input/e10.iks says (w,r,c,l)=(0,0,0,0)

```
int:x;
int:main()
{
  if (x) then {
    x = 0;
  }else{
    x = 1;
  }
}
```

- Generated ILOC

```
        loadI   0       =>      r0
        addI    r0, bss =>      r0
        load    r0      =>      r1
        loadI   1       =>      r7
        cmp_GE  r1, r7  ->      r6
        cbr     r6      ->      L0, L1
L0:     nop
        loadI   0       =>      r3
        loadI   0       =>      r2
        addI    r2, bss =>      r2
        store   r3      =>      r2
        jumpI   ->      L2
L1:     nop
        loadI   1       =>      r5
        loadI   0       =>      r4
        addI    r4, bss =>      r4
        store   r5      =>      r4
        jumpI   ->      L2
L2:     nop
```

2.12 input/e11.iks

- Valgrind with input/e11.iks says (w,r,c,l)=(0,0,0,0)

```
int:x;
int:main()
{
  do {
    x = 0;
  }while (x);
}
```

- Generated ILOC

```
L0:     nop
        loadI   0       =>      r1
        loadI   0       =>      r0
        addI    r0, bss =>      r0
        store   r1      =>      r0
        loadI   0       =>      r2
        addI    r2, bss =>      r2
        load    r2      =>      r3
        loadI   1       =>      r5
        cmp_GE  r3, r5  ->      r4
        cbr     r4      ->      L1, L2
L1:     nop
        jumpI   ->      L0
L2:     nop
```

2.13 input/e12.iks

- Valgrind with input/e12.iks says (w,r,c,l)=(0,0,0,0)

```
int:x;
int:main()
{
  while (x) do {
    x = 0;
  }
}
```

- Generated ILOC

```
L0:     nop
        loadI   0       =>      r0
        addI    r0, bss =>      r0
        load    r0      =>      r1
        loadI   1       =>      r5
        cmp_GE  r1, r5  ->      r4
        cbr     r4      ->      L1, L2
L1:     nop
        loadI   0       =>      r3
        loadI   0       =>      r2
        addI    r2, bss =>      r2
        store   r3      =>      r2
        jumpI   ->      L0
L2:     nop
```

2.14 input/e13.iks

- Valgrind with input/e13.iks says (w,r,c,l)=(0,0,0,0)

```
int:x;
int:main()
{
  if (x || x && x) then {
    x = 0;
  }
}
```

- Generated ILOC

```
        loadI   1       =>      r10
        loadI   0       =>      r0
        addI    r0, bss =>      r0
        load    r0      =>      r1
        cmp_GE  r1, r10 ->      r11
        cbr     r11     ->      L6, L7
L6:     loadI   1       =>      r9
        jumpI   ->      L5
L7:     nop
        loadI   1       =>      r7
        loadI   0       =>      r2
        addI    r2, bss =>      r2
        load    r2      =>      r3
        cmp_GE  r3, r7  ->      r8
        cbr     r8      ->      L1, L2
L2:     loadI   0       =>      r6
        jumpI   ->      L0
L1:     nop
        loadI   0       =>      r4
        addI    r4, bss =>      r4
        load    r4      =>      r5
        cmp_GE  r5, r7  ->      r8
        cbr     r8      ->      L3, L4
L3:     loadI   1       =>      r6
        jumpI   ->      L0
L4:     loadI   0       =>      r6
L0:     nop
        cmp_GE  r6, r10 ->      r11
        cbr     r11     ->      L8, L9
L8:     loadI   1       =>      r9
        jumpI   ->      L5
L9:     loadI   0       =>      r9
L5:     nop
        loadI   1       =>      r15
        cmp_GE  r9, r15 ->      r14
        cbr     r14     ->      L10, L11
L10:    nop
        loadI   0       =>      r13
        loadI   0       =>      r12
        addI    r12, bss        =>      r12
        store   r13     =>      r12
L11:    nop
```

2.15 input/e14.iks

- Valgrind with input/e14.iks says (w,r,c,l)=(0,0,0,0)

```
/* global variable */
int:x[10][10];
int:y;
int:main()
{
  y = 0;
}
```

- Generated ILOC

```
loadI   0       =>      r1
loadI   400     =>      r0
addI    r0, bss =>      r0
store   r1      =>      r0
```

3 g3 ../../gc/compiladores/

3.1 input/e00.iks

- Valgrind with input/e00.iks says (w,r,c,l)=(0,0,0,392)

```
/* global variable */
int:x;
int:main()
{
  x = 0;
}
```

- Generated ILOC

```
        loadI 0 => r0
        storeAI r0 => bss, 0
l1:
```

3.2 input/e01.iks

- Valgrind with input/e01.iks says (w,r,c,l)=(0,0,0,392)

```
/* local variable */
int:main()
int:x;
{
  x = 0;
}
```

- Generated ILOC

```
        loadI 0 => r0
        storeAI r0 => rarp, 0
l1:
```

3.3 input/e02.iks

- Valgrind with input/e02.iks says (w,r,c,l)=(0,0,0,392)

```
/* global variable with offset */
int:x;
int:y;
int:main()
{
  y = 0;
}
```

- Generated ILOC

```
        loadI 0 => r0
        storeAI r0 => bss, 4
l1:
```

3.4 input/e03.iks

- Valgrind with input/e03.iks says (w,r,c,l)=(0,0,0,392)

```
/* local variable with offset */
int:main()
int:x;
int:y;
{
  y = 0;
}
```

- Generated ILOC

```
        loadI 0 => r0
        storeAI r0 => rarp, 4
l1:
```

3.5 input/e04.iks

- Valgrind with input/e04.iks says (w,r,c,l)=(0,2,0,1,104)

```
/* global arranjo */
int:x;
int:y[10];
int:main()
{
  y[0] = 0;
  y[1] = 0;
}
```

- Generated ILOC

```
        loadI 0 => r0
        loadI 0 => r1
        loadI 4 => r2
        mult r1, r2 => r1
        loadI 4 => r3
        add r3, r1 => r1
        storeAI r0 => bss, r1
l1:
        loadI 0 => r4
        loadI 1 => r5
        loadI 4 => r6
        mult r5, r6 => r5
        loadI 4 => r7
        add r7, r5 => r5
        storeAI r4 => bss, r5
l2:
```

3.6 input/e05.iks

- Valgrind with input/e05.iks says (w,r,c,l)=(0,2,0,1,688)

```
/* global arranjo with expression */
int:x;
int:y[10];
int:main()
{
  y[x+1] = 0;
  y[x*x] = 0;
}
```

- Generated ILOC

```
        loadI 0 => r0
        loadAI bss, 0 => r2
        load r2 => r1
        loadI 1 => r3
        add r1, r3 => r4
        loadI 4 => r5
        mult r4, r5 => r4
        loadI 4 => r6
        add r6, r4 => r4
        storeAI r0 => bss, r4
l1:
        loadI 0 => r7
        loadAI bss, 0 => r9
        load r9 => r8
        loadAI bss, 0 => r11
        load r11 => r10
        mult r8, r10 => r12
        loadI 4 => r13
        mult r12, r13 => r12
        loadI 4 => r14
        add r14, r12 => r12
        storeAI r7 => bss, r12
l2:
```

3.7 input/e06.iks

- Valgrind with input/e06.iks says (w,r,c,l)=(0,0,0,800)

```
/* arithm. */
int:x;
int:main()
{
  x = 0*1+2*3;
}
```

- Generated ILOC

```
        loadI 0 => r0
        loadI 1 => r1
        mult r0, r1 => r2
        loadI 2 => r3
        loadI 3 => r4
        mult r3, r4 => r5
        add r2, r5 => r6
        storeAI r6 => bss, 0
l1:
```

3.8 input/e07.iks

- Valgrind with input/e07.iks says (w,r,c,l)=(0,0,0,800)

```
/* arithm. */
int:x;
int:main()
{
  x = 0*(1+2)*3;
}
```

- Generated ILOC

```
        loadI 0 => r0
        loadI 1 => r1
        loadI 2 => r2
        add r1, r2 => r3
        mult r0, r3 => r4
        loadI 3 => r5
        mult r4, r5 => r6
        storeAI r6 => bss, 0
l1:
```

3.9 input/e08.iks

- Valgrind with input/e08.iks says (w,r,c,l)=(0,0,0,392)

```
int:x;
int:main()
{
  x = 0;
}
```

- Generated ILOC

```
        loadI 0 => r0
        storeAI r0 => bss, 0
l1:
```

3.10 input/e09.iks

- Valgrind with input/e09.iks says (w,r,c,l)=(0,0,0,760)

```
int:x;
int:main()
{
  if (x) then {
    x = 0;
  }
}
```

- Generated ILOC

```
        loadAI bss, 0 => r1
        load r1 => r0
        cmp_GE r0, 1 -> r2
        cbr r2 -> l2, l1
l2:     loadI 0 => r3
        storeAI r3 => bss, 0
l3:
l1:
```

3.11 input/e10.iks

- Valgrind with input/e10.iks says (w,r,c,l)=(0,0,0,1,024)

```
int:x;
int:main()
{
  if (x) then {
    x = 0;
  }else{
    x = 1;
  }
}
```

- Generated ILOC

```
        loadAI bss, 0 => r1
        load r1 => r0
        cmp_GE r0, 1 -> r2
        cbr r2 -> l2, l3
l2:     loadI 0 => r3
        storeAI r3 => bss, 0
l4:
        jumpI -> l1
l3:     loadI 1 => r4
        storeAI r4 => bss, 0
l5:
l1:
```

3.12 input/e11.iks

- Valgrind with input/e11.iks says (w,r,c,l)=(0,0,0,696)

```
int:x;
int:main()
{
  do {
    x = 0;
  }while (x);
}
```

- Generated ILOC

```
l2:     loadI 0 => r0
        storeAI r0 => bss, 0
l4:
l3:     loadAI bss, 0 => r2
        load r2 => r1
        cmp_GE r1, 1 -> r3
        cbr r3 -> l2, l1
l1:
```

3.13 input/e12.iks

- Valgrind with input/e12.iks says (w,r,c,l)=(0,0,0,768)

```
int:x;
int:main()
{
  while (x) do {
    x = 0;
  }
}
```

- Generated ILOC

```
l3:     loadAI bss, 0 => r1
        load r1 => r0
        cmp_GE r0, 1 -> r2
        cbr r2 -> l2, l1
l2:     loadI 0 => r3
        storeAI r3 => bss, 0
l4:
        jumpI -> l3
l1:
```

3.14 input/e13.iks

- Valgrind with input/e13.iks says (w,r,c,l)=(0,0,0,1,288)

```
int:x;
int:main()
{
  if (x || x && x) then {
    x = 0;
  }
}
```

- Generated ILOC

```
        loadAI bss, 0 => r1
        load r1 => r0
        cmp_GE r0, 1 -> r2
        cbr r2 -> l2, l3
l3:     loadAI bss, 0 => r4
        load r4 => r3
        cmp_GE r3, 1 -> r5
        cbr r5 -> l4, l1
l4:     loadAI bss, 0 => r7
        load r7 => r6
        cmp_GE r6, 1 -> r8
        cbr r8 -> l2, l1
l2:     loadI 0 => r9
        storeAI r9 => bss, 0
l5:
l1:
```

3.15 input/e14.iks

- Valgrind with input/e14.iks says (w,r,c,l)=(0,0,0,560)

```
/* global variable */
int:x[10][10];
int:y;
int:main()
{
  y = 0;
}
```

- Generated ILOC

```
        loadI 0 => r0
        storeAI r0 => bss, 4
l1:
```

4 g6 ../../gf/compilador/

4.1 input/e00.iks

- Valgrind with input/e00.iks says (w,r,c,l)=(5,2,1,864)

```
/* global variable */
int:x;
int:main()
{
  x = 0;
}
```

- Generated ILOC

```
loadAI table, 0 => r1
loadI 0 => r2
i2i r2 => r3
store r3 => r1
```

4.2 input/e01.iks

- Valgrind with input/e01.iks says (w,r,c,l)=(5,2,1,888)

```
/* local variable */
int:main()
int:x;
{
  x = 0;
}
```

- Generated ILOC

```
loadAI fp, 0 => r1
loadI 0 => r2
i2i r2 => r3
store r3 => r1
```

4.3 input/e02.iks

- Valgrind with input/e02.iks says (w,r,c,l)=(6,2,1,1,064)

```
/* global variable with offset */
int:x;
int:y;
int:main()
{
  y = 0;
}
```

- Generated ILOC

```
loadAI table, 0 => r1
loadAI table, 4 => r2
loadI 0 => r3
i2i r3 => r4
store r4 => r2
```

4.4 input/e03.iks

- Valgrind with input/e03.iks says (w,r,c,l)=(6,2,1,1,088)

```
/* local variable with offset */
int:main()
int:x;
int:y;
{
  y = 0;
}
```

- Generated ILOC

```
loadAI fp, 0 => r1
loadAI fp, 4 => r2
loadI 0 => r3
i2i r3 => r4
store r4 => r2
```

4.5 input/e04.iks

- input/e04.iks causes a segfault (exit code is 139)

4.6 input/e05.iks

- input/e05.iks causes a segfault (exit code is 139)

4.7 input/e06.iks

- Valgrind with input/e06.iks says (w,r,c,l)=(5,2,3,2,208)

```
/* arithm. */
int:x;
int:main()
{
  x = 0*1+2*3;
}
```

- Generated ILOC

```
loadAI table, 0 => r1
loadI 0 => r2
loadI 1 => r3
mult r2, r3 => r4
loadI 2 => r5
loadI 3 => r6
mult r5, r6 => r7
add r4, r7 => r8
i2i r8 => r9
store r9 => r1
```

4.8 input/e07.iks

- Valgrind with input/e07.iks says (w,r,c,l)=(5,2,3,2,256)

```
/* arithm. */
int:x;
int:main()
{
  x = 0*(1+2)*3;
}
```

- Generated ILOC

```
loadAI table, 0 => r1
loadI 0 => r2
loadI 1 => r3
loadI 2 => r4
add r3, r4 => r5
mult r2, r5 => r6
loadI 3 => r7
mult r6, r7 => r8
i2i r8 => r9
store r9 => r1
```

4.9 input/e08.iks

- Valgrind with input/e08.iks says (w,r,c,l)=(5,2,1,864)

```
int:x;
int:main()
{
  x = 0;
}
```

- Generated ILOC

```
loadAI table, 0 => r1
loadI 0 => r2
i2i r2 => r3
store r3 => r1
```

4.10 input/e09.iks

- Valgrind with input/e09.iks says (w,r,c,l)=(5,2,2,1,032)

```
int:x;
int:main()
{
  if (x) then {
    x = 0;
  }
}
```

- Generated ILOC

```
loadAI table, 0 => r1


cbr r1 => L2, L1
L2:
loadI 0 => r2
i2i r2 => r3
store r3 => r1

L1:
```

4.11 input/e10.iks

- Valgrind with input/e10.iks says (w,r,c,l)=(5,2,2,1,600)

```
int:x;
int:main()
{
  if (x) then {
    x = 0;
  }else{
    x = 1;
  }
}
```

- Generated ILOC

```
loadAI table, 0 => r1

cbr r1 => L2, L3
L2:
loadI 0 => r2
i2i r2 => r3
store r3 => r1

jumpI => L1
L3:
loadI 1 => r4
i2i r4 => r5
store r5 => r1

L1:
```

4.12 input/e11.iks

- Valgrind with input/e11.iks says (w,r,c,l)=(5,2,2,1,056)

```
int:x;
int:main()
{
  do {
    x = 0;
  }while (x);
}
```

- Generated ILOC

```
loadAI table, 0 => r1

L2:
loadI 0 => r2
i2i r2 => r3
store r3 => r1


cbr r1 => L2, L1
L1:
```

4.13 input/e12.iks

- Valgrind with input/e12.iks says (w,r,c,l)=(5,2,2,1,032)

```
int:x;
int:main()
{
  while (x) do {
    x = 0;
  }
}
```

- Generated ILOC

```
loadAI table, 0 => r1

L2:

cbr r1 => L3, L1
L3:
loadI 0 => r2
i2i r2 => r3
store r3 => r1

jumpI => L2
L1:
```

4.14 input/e13.iks

- Exit code for input/e13.iks is not zero (exit code is 1)
- Valgrind with input/e13.iks says (w,r,c,l)=(2,0,1,632)

```
int:x;
int:main()
{
  if (x || x && x) then {
    x = 0;
  }
}
```

- Generated ILOC

```
                    ^
line 4: syntax error
```

4.15 input/e14.iks

- Valgrind with input/e14.iks says (w,r,c,l)=(5,2,1,1,312)

```
/* global variable */
int:x[10][10];
int:y;
int:main()
{
  y = 0;
}
```

- Generated ILOC

```
loadAI table, 0 => r1
loadI 0 => r2
i2i r2 => r3
store r3 => r1
```

5 g8 ../../gh/compiladores/

5.1 input/e00.iks

- Valgrind with input/e00.iks says (w,r,c,l)=(0,0,0,310)

```
/* global variable */
int:x;
int:main()
{
  x = 0;
}
```

- Generated ILOC

```
loadI 0 => r1
storeAI r1 => bss, 0
```

5.2 input/e01.iks

- Valgrind with input/e01.iks says (w,r,c,l)=(0,0,0,310)

```
/* local variable */
int:main()
int:x;
{
  x = 0;
}
```

- Generated ILOC

```
loadI 0 => r1
storeAI r1 => rarp, 0
```

5.3 input/e02.iks

- Valgrind with input/e02.iks says (w,r,c,l)=(0,0,0,312)

```
/* global variable with offset */
int:x;
int:y;
int:main()
{
  y = 0;
}
```

- Generated ILOC

```
loadI 0 => r1
storeAI r1 => bss, 4
```

5.4 input/e03.iks

- Valgrind with input/e03.iks says (w,r,c,l)=(0,0,0,312)

```
/* local variable with offset */
int:main()
int:x;
int:y;
{
  y = 0;
}
```

- Generated ILOC

```
loadI 0 => r1
storeAI r1 => rarp, 4
```

5.5 input/e04.iks

- Valgrind with input/e04.iks says (w,r,c,l)=(0,0,0,1,238)

```
/* global arranjo */
int:x;
int:y[10];
int:main()
{
  y[0] = 0;
  y[1] = 0;
}
```

- Generated ILOC

```
loadI 0 => r1
loadAO r1 => r3
multI r3, 4 => r4
loadI 0 => r5
storeAO r5 => bss, r4
loadI 1 => r7
loadAO r7 => r9
multI r9, 4 => r10
loadI 0 => r11
storeAO r11 => bss, r10
```

5.6 input/e05.iks

- Valgrind with input/e05.iks says (w,r,c,l)=(0,0,0,1,402)

```
/* global arranjo with expression */
int:x;
int:y[10];
int:main()
{
  y[x+1] = 0;
  y[x*x] = 0;
}
```

- Generated ILOC

```
loadI 1 => r3
add r2, r3 => r1
loadAO r1 => r5
multI r5, 4 => r6
loadI 0 => r7
storeAO r7 => bss, r6
mult r10, r11 => r9
loadAO r9 => r13
multI r13, 4 => r14
loadI 0 => r15
storeAO r15 => bss, r14
```

5.7 input/e06.iks

- Valgrind with input/e06.iks says (w,r,c,l)=(0,0,0,1,006)

```
/* arithm. */
int:x;
int:main()
{
  x = 0*1+2*3;
}
```

- Generated ILOC

```
loadI 0 => r3
loadI 1 => r4
mult r3, r4 => r2
loadI 2 => r6
loadI 3 => r7
mult r6, r7 => r5
add r2, r5 => r1
storeAI r1 => bss, 0
```

5.8 input/e07.iks

- Valgrind with input/e07.iks says (w,r,c,l)=(0,0,0,1,006)

```
/* arithm. */
int:x;
int:main()
{
  x = 0*(1+2)*3;
}
```

- Generated ILOC

```
loadI 0 => r3
loadI 1 => r5
loadI 2 => r6
add r5, r6 => r4
mult r3, r4 => r2
loadI 3 => r7
mult r2, r7 => r1
storeAI r1 => bss, 0
```

5.9 input/e08.iks

- Valgrind with input/e08.iks says (w,r,c,l)=(0,0,0,310)

```
int:x;
int:main()
{
  x = 0;
}
```

- Generated ILOC

```
loadI 0 => r1
storeAI r1 => bss, 0
```

5.10 input/e09.iks

- Valgrind with input/e09.iks says (w,r,c,l)=(0,0,0,766)

```
int:x;
int:main()
{
  if (x) then {
    x = 0;
  }
}
```

- Generated ILOC

```
loadAI bss, 0 => r0
cbr r0 -> L1, L0
L1:
loadI 0 => r2
storeAI r2 => bss, 0
L0:
```

5.11 input/e10.iks

- Valgrind with input/e10.iks says (w,r,c,l)=(0,0,0,1,234)

```
int:x;
int:main()
{
  if (x) then {
    x = 0;
  }else{
    x = 1;
  }
}
```

- Generated ILOC

```
loadAI bss, 0 => r0
cbr r0 -> L1, L2
L1:
loadI 0 => r2
storeAI r2 => bss, 0
jumpI -> L0
L2:
loadI 1 => r4
storeAI r4 => bss, 0
L0:
```

5.12 input/e11.iks

- Valgrind with input/e11.iks says (w,r,c,l)=(0,0,0,867)

```
int:x;
int:main()
{
  do {
    x = 0;
  }while (x);
}
```

- Generated ILOC

```
L1:
loadI 0 => r1
storeAI r1 => bss, 0
L2:
loadAI bss, 0 => r2
cbr r2 -> L1, L0
L0:
```

5.13 input/e12.iks

- Valgrind with input/e12.iks says (w,r,c,l)=(0,0,0,967)

```
int:x;
int:main()
{
  while (x) do {
    x = 0;
  }
}
```

- Generated ILOC

```
L2:
loadAI bss, 0 => r0
cbr r0 -> L1, L0
L1:
loadI 0 => r2
storeAI r2 => bss, 0
jumpI -> L2
L0:
```

5.14 input/e13.iks

- Valgrind with input/e13.iks says (w,r,c,l)=(0,0,0,1,132)

```
int:x;
int:main()
{
  if (x || x && x) then {
    x = 0;
  }
}
```

- Generated ILOC

```
loadAI bss, 0 => r0
L2:
loadAI bss, 0 => r1
L3:
loadAI bss, 0 => r2
L1:
loadI 0 => r4
storeAI r4 => bss, 0
L0:
```

5.15 input/e14.iks

- Valgrind with input/e14.iks says (w,r,c,l)=(0,0,0,318)

```
/* global variable */
int:x[10][10];
int:y;
int:main()
{
  y = 0;
}
```

- Generated ILOC

```
loadI 0 => r1
storeAI r1 => bss, 400
```