Compiler Construction

Linear Scan

Yet another register allocation algorithm?

Linear scan

- Popular algorithm in industrial quality compilers
- Based on the optimal greedy coloring of interval graphs

Problem statement

Given a sequence of intervals, we want the minimum number of colors necessary to paint them

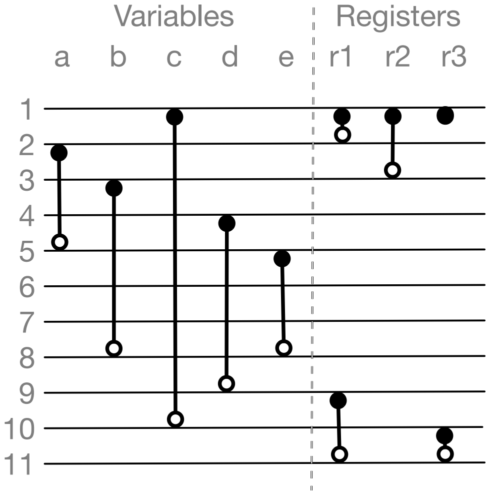
There are optimal algorithms [2004]for this but here we don't want to count the minimum number of color but rewrite the program so it can fit in fixed number of colors

Finding interval lines

An interval for a variable starts at the first program point a variable is alive and ends at the last point this variable is alive

```
L1:    c := r3                   1
       a := r1                   2
       b := r2                   3
       d := 0                    4
       e := a                    5
L1:    d := d + b                6
       e := e - 1                7
       cjump e > 0, L6           8
       r1 := d                   9
       r3 := c                   10
       return                    11
```

Intervals



Main Idea

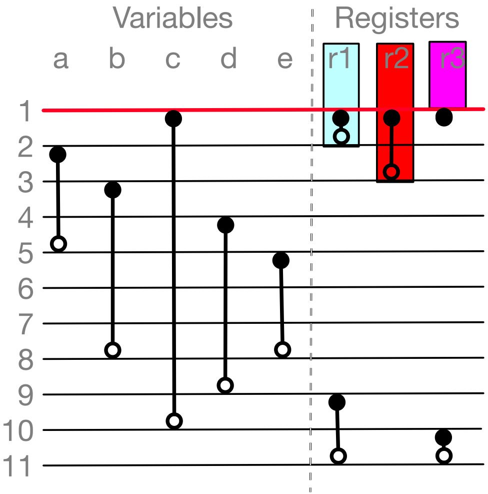
1. walk all intervals in order of increasing start points
2. Release no longer used registers
3. If all registers are used, spill a variable
4. Else associate register to a variable and remove register from the pool of free registers.

What is the best variable to spill?

Many heuristics!.

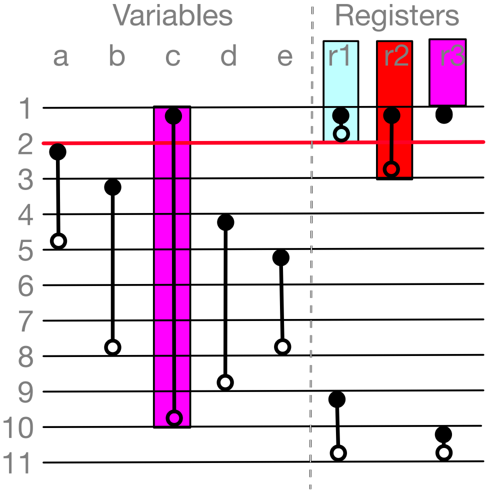
Original paper spills the interval that the farthest end point from the spilling point

Full example



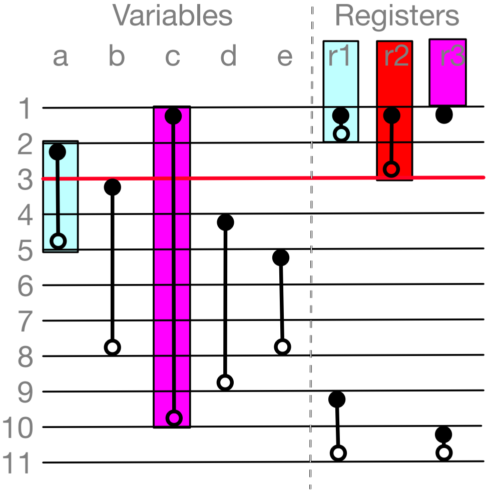
assign r3 to c since this is the only available register

Full example



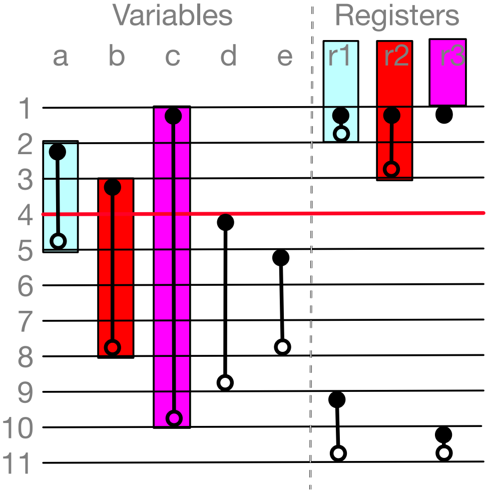
assign r1 to a since this is the only available register

Full example



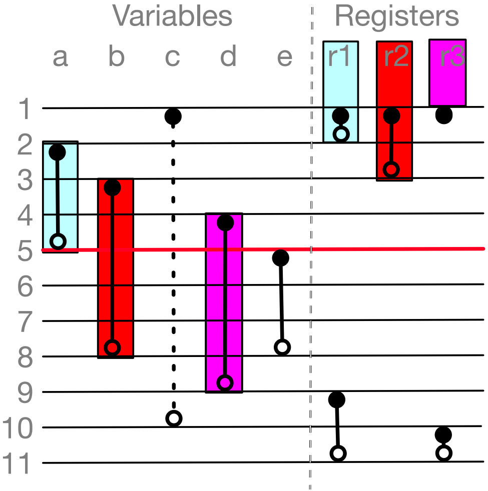
assign r2 to b since this is the only available register

Full example



There is no free register. Spill c as its interval as the farthest end-point.

Full example



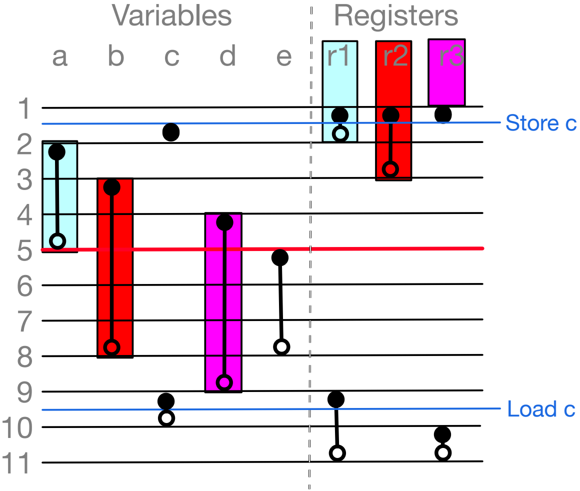
Spill Everywhere in the interval

In RISC, every variable needs to be used in a register.

The spilled interval is splitted into many small intervals.

- Insert a store operation just after the definition
- and suround each use with load/store opérations.

Full example



Interval of c is splitted into two micro-intervals

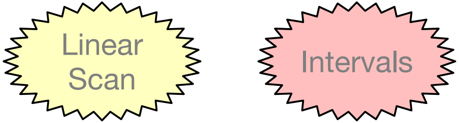
Spill Everywhere in the interval

Linear scan is sub-optimal, results are not guaranteed to be optimal, neither in term of minimum register assignment, neither in term of spilling/coalescing.

The spilled interval is splitted into many small intervals.

- Insert a store operation just after the definition
- and suround each use with load/store opérations.

Summary



Linear Scan

Intervals