Compiler Construction 2010/2011: Register allocation example

Konrad Anton

November 30, 2010

Architecture

Registers:

- `a0,a1` Arguments, caller-save
- `v0` Return value
- `s0` Callee-save temp
- `t0` Caller-save temp

And no `div` instruction – use function!

The Program

Compute $f(x, y) = \frac{x+y}{y}$.
After instruction selection,

```
f(x,y) {

  sum = x + y ;

  q = div( sum, y );

  return q;
}
```

The Program

Compute $f(x, y) = \frac{x+y}{y}$.
After instruction selection, apply calling convention

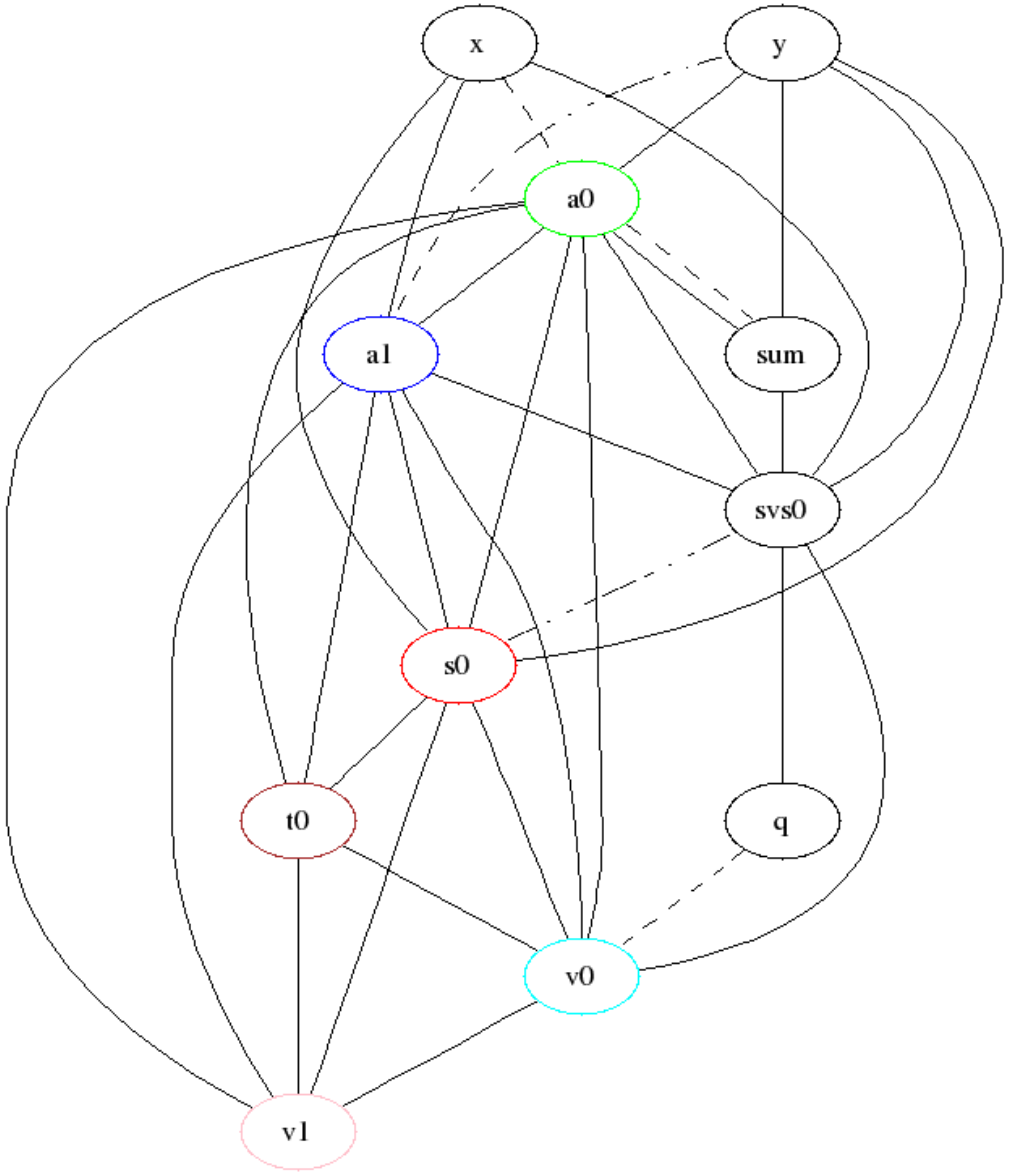
```
f: {
   x = %a0 ;          // retrieve parameters
   y = %a1 ;          // retrieve parameters
   svs0 = %s0 ;       // save callee-save reg
   sum = x + y ;
   %a0 = sum ;
   %a1 = y ;
   call div
   q = %v0;
   %s0 = svs0 ;
   %v0 = q;
}
```

Liveness analysis

(Live-out = successor's Live-in because the example has no branches)

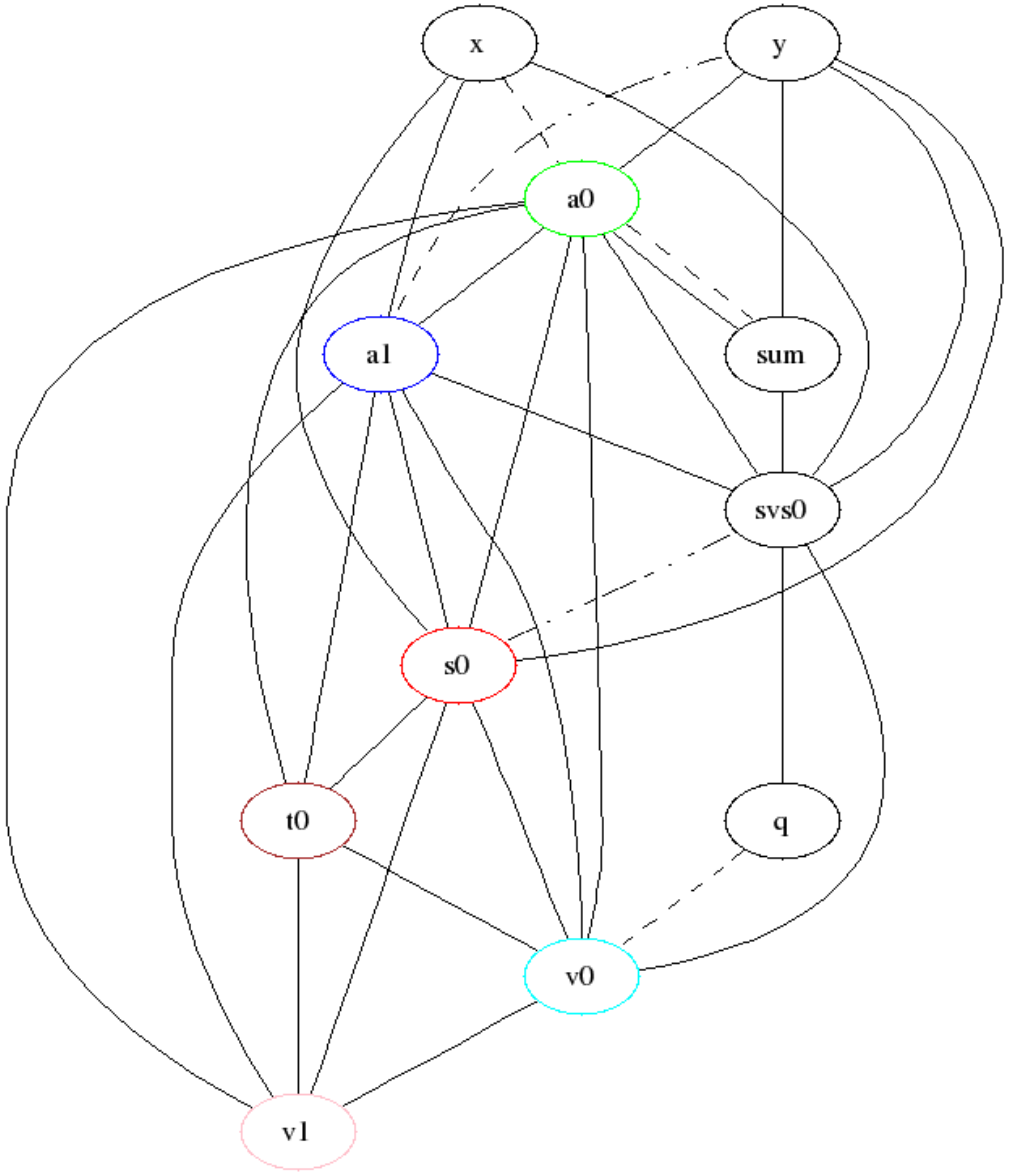
Nr	Stmt	Def	Use	Live-in
1	x=%a0	x	%a0	%a0,%a1,%s0
2	y=%a1	y	%a1	x,%a1,%s0
3	svs0=%s0	svs0	%s0	x,y,%s0
4	sum=x+y	sum	x,y	x,y,svs0
5	%a0=sum	%a0	sum	y,svs0,sum
6	%a1=y	%a1	y	%a0,y,svs0
7	call div	%v0,%t0,%a0,%a1	%a0,%a1	%a0,%a1,svs0
8	q=%v0	q	%v0	%v0,svs0
9	%s0=svs0	%s0	svs0	q,svs0
10	%v0=q	%v0	q	q

Conflict graph



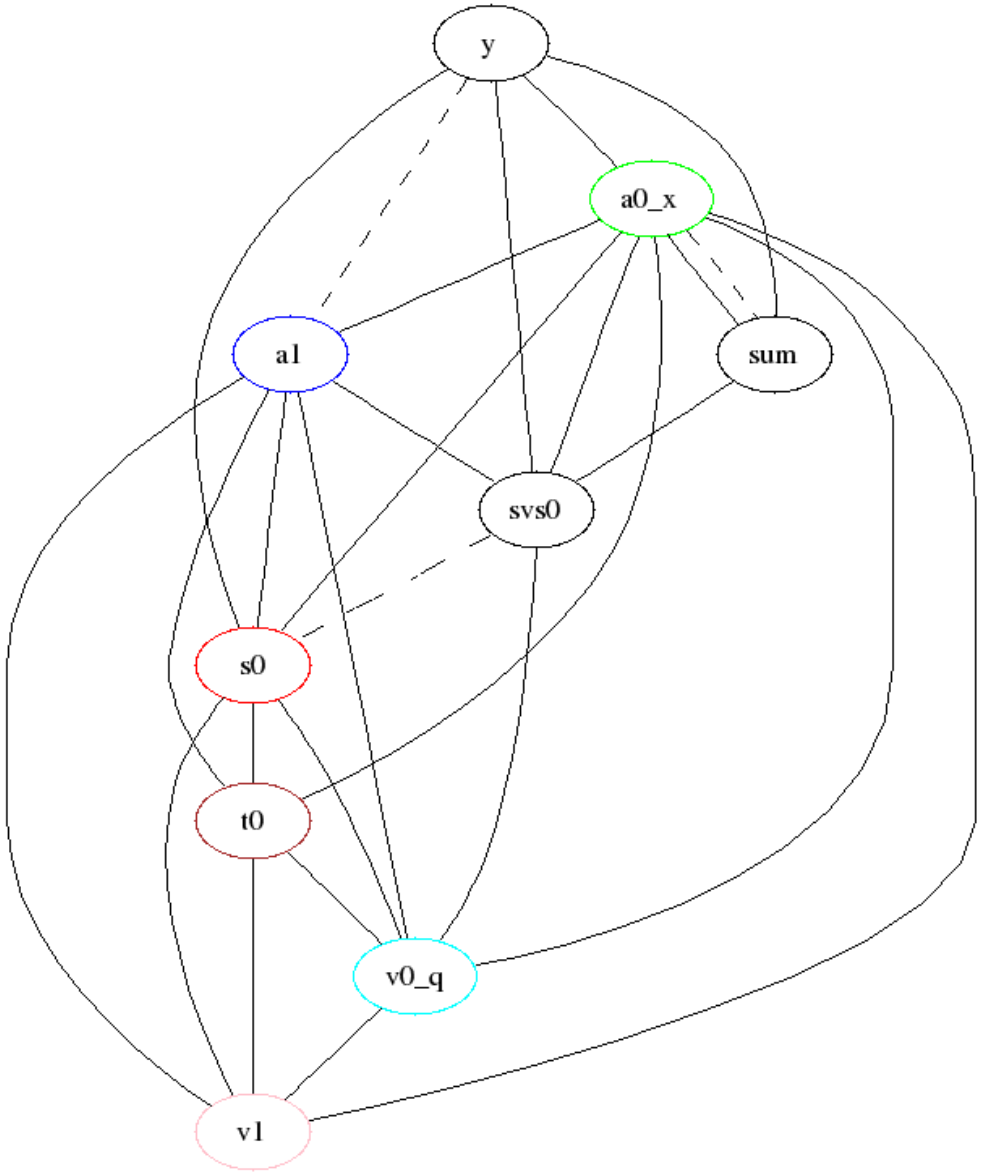
- $N = 5$ registers.
- Real registers precolored.

Conflict graph

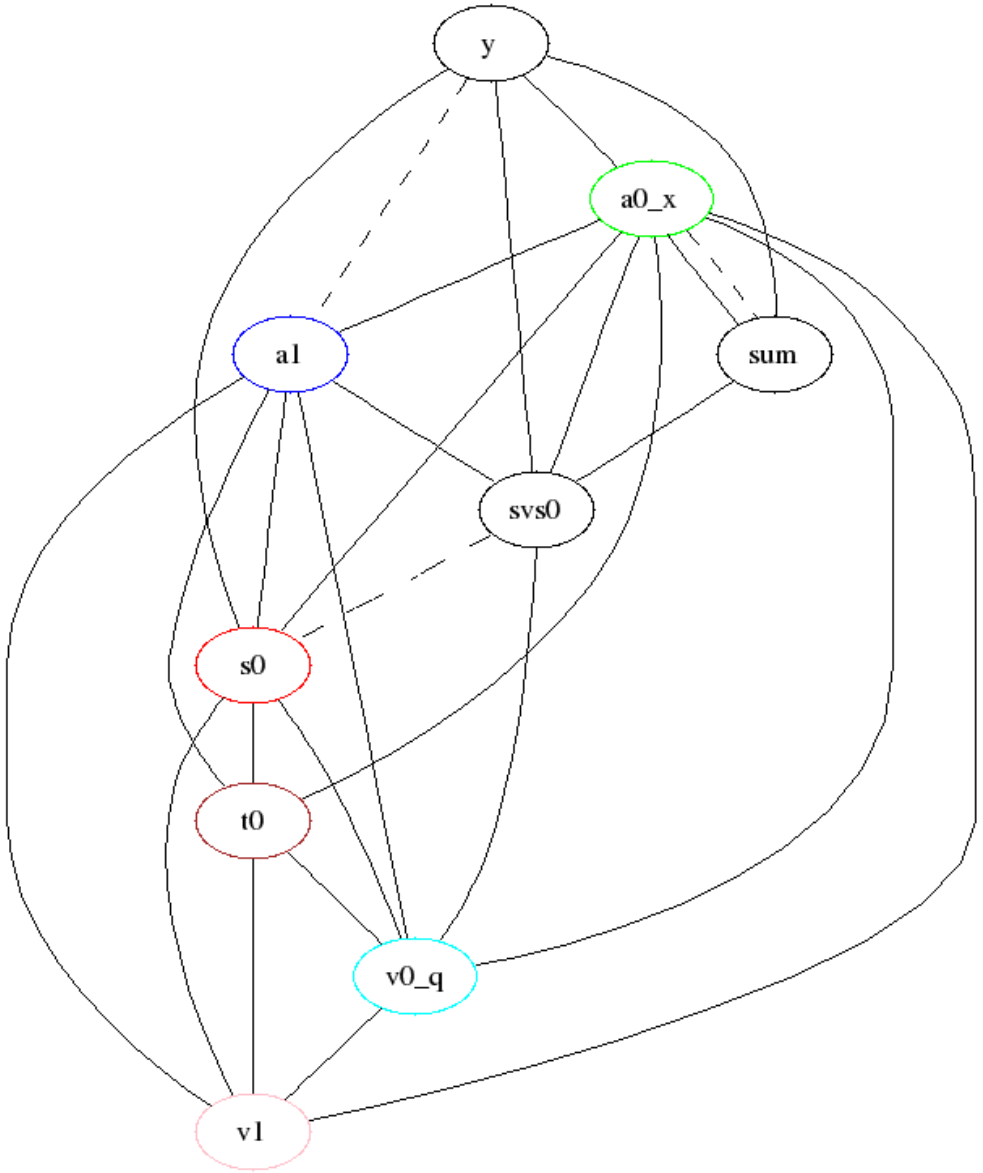


- $N = 5$ registers.
- Real registers precolored.
- No simplify possible.
- coalesce *q* and *v*0 by George.
- coalesce *x* and *a*0 by George.

Conflict graph

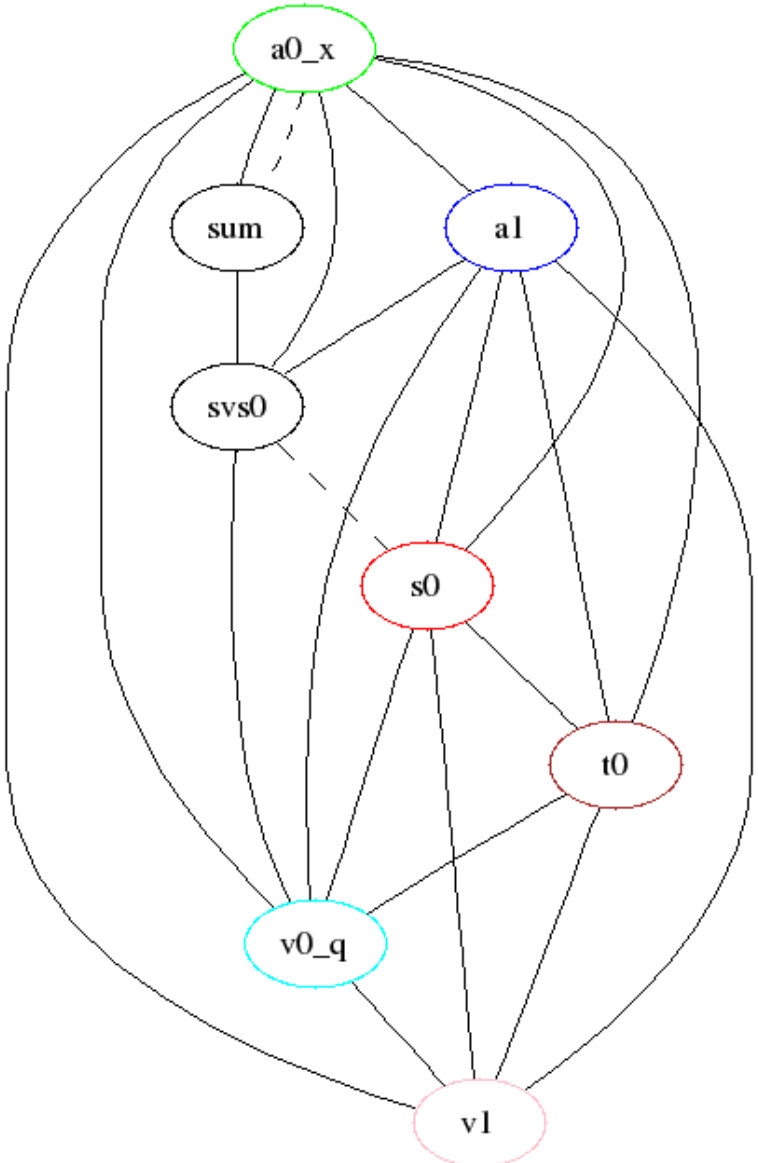


Conflict graph

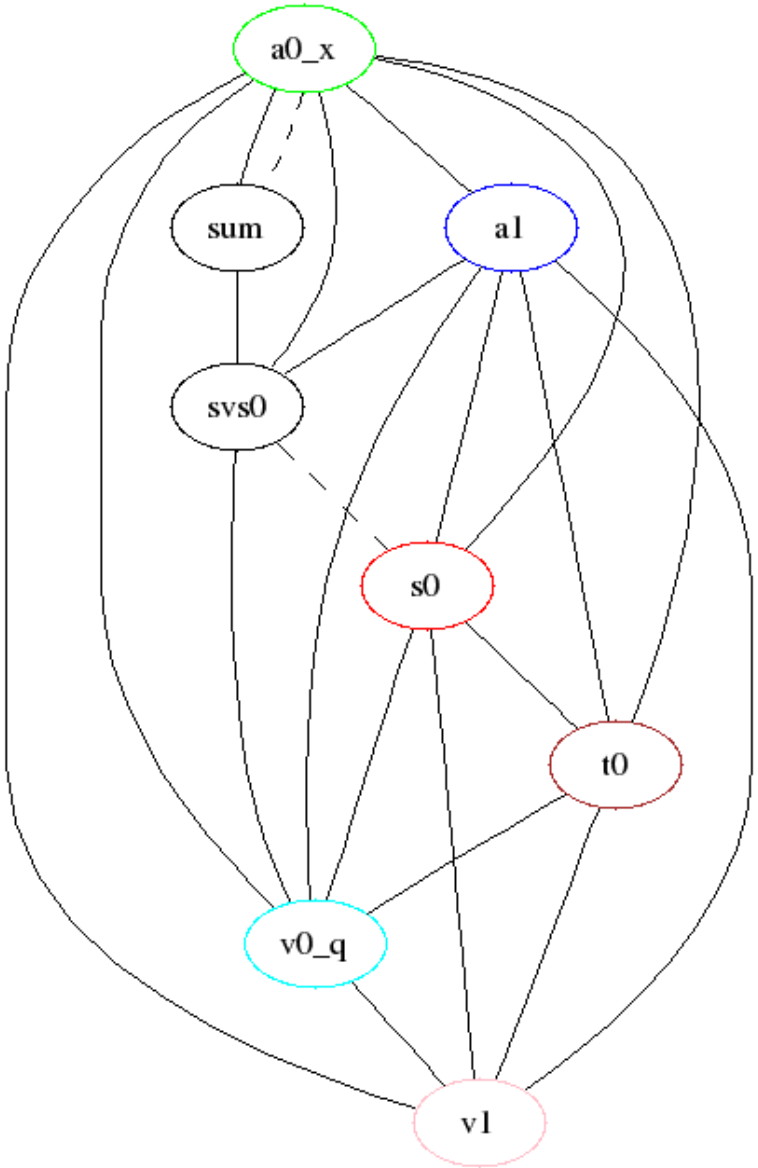


- No simplify possible.
- No coalesce possible.
- Freeze y, then simplify y

Conflict graph

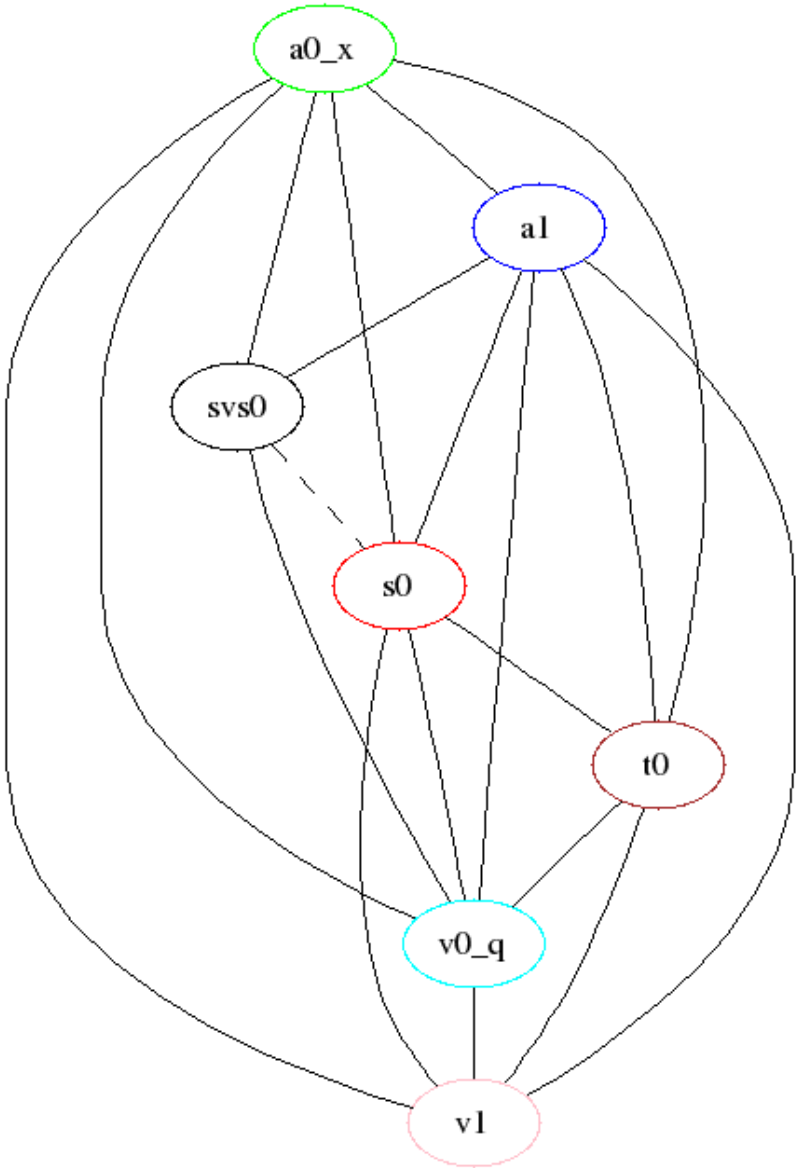


Conflict graph

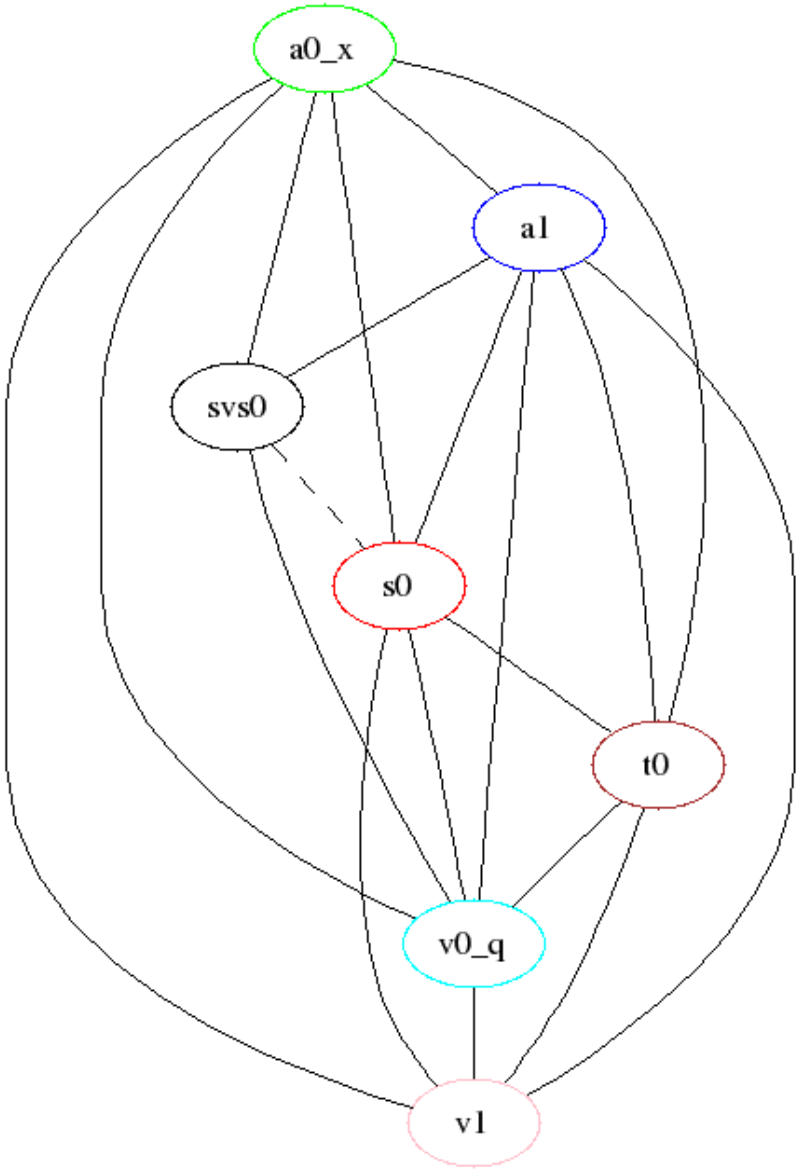


- No simplify
- No coalesce
- Freeze sum, then simplify sum.

Conflict graph

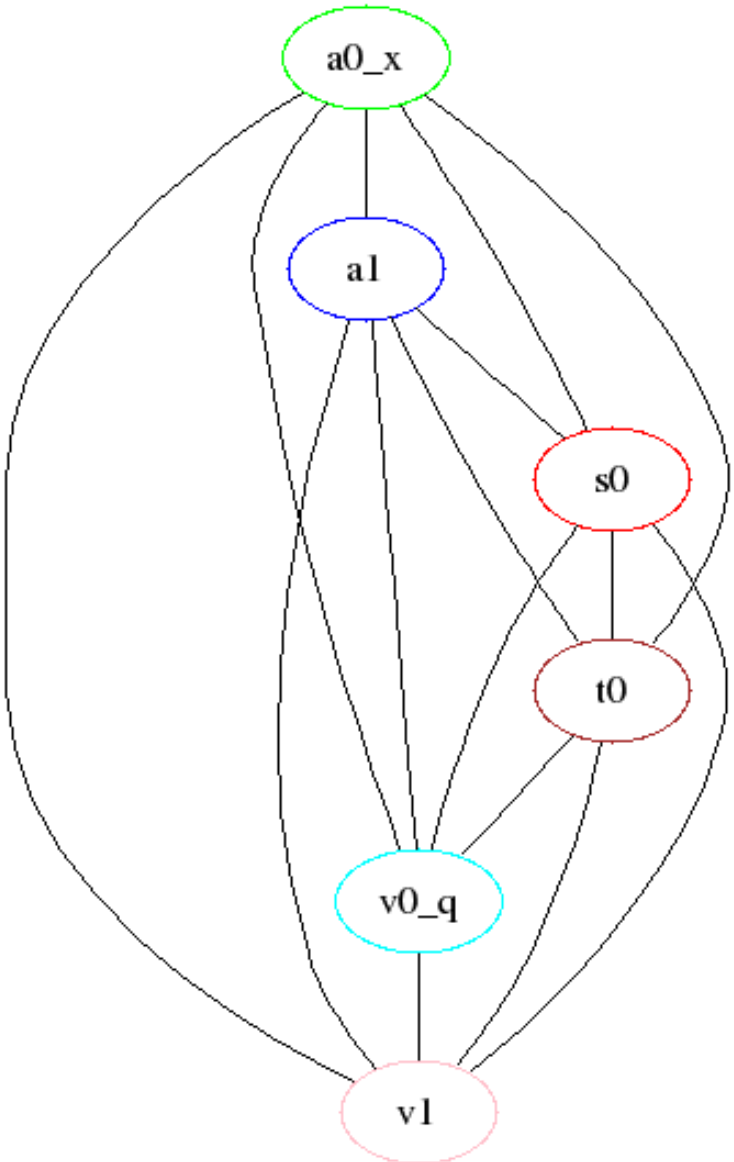


Conflict graph

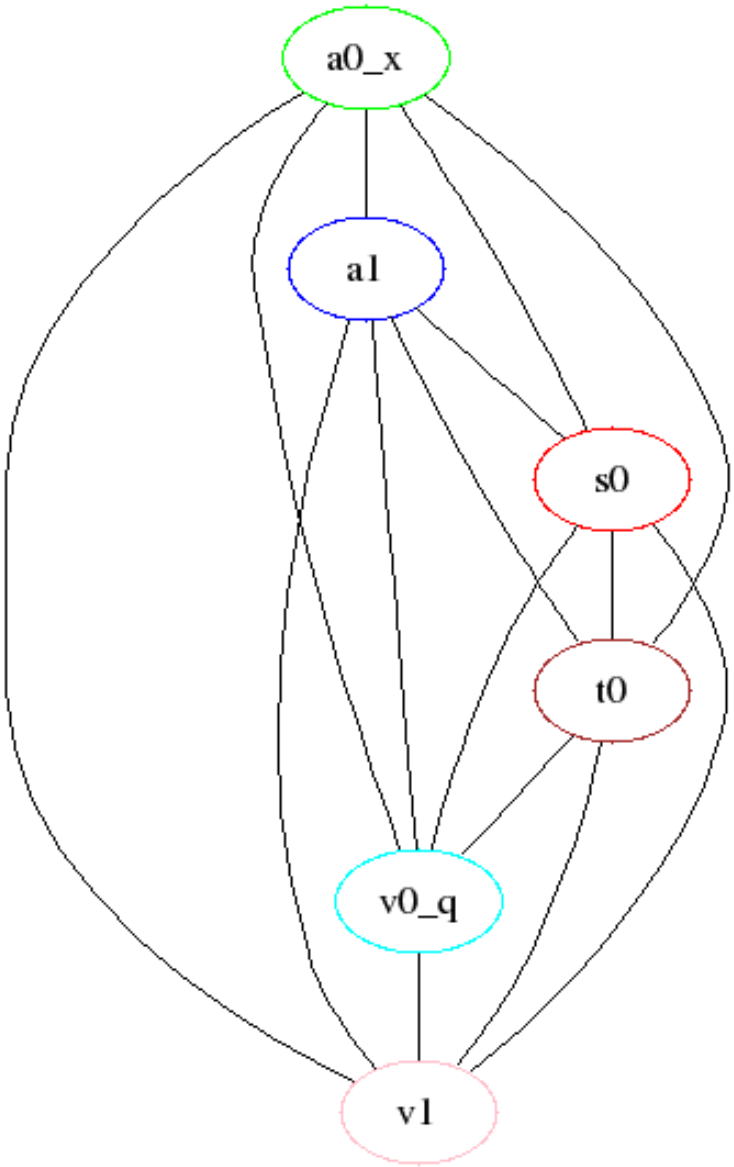


- No simplify, no coalesce.
- Freeze svs0, then simplify svs0.

Conflict graph

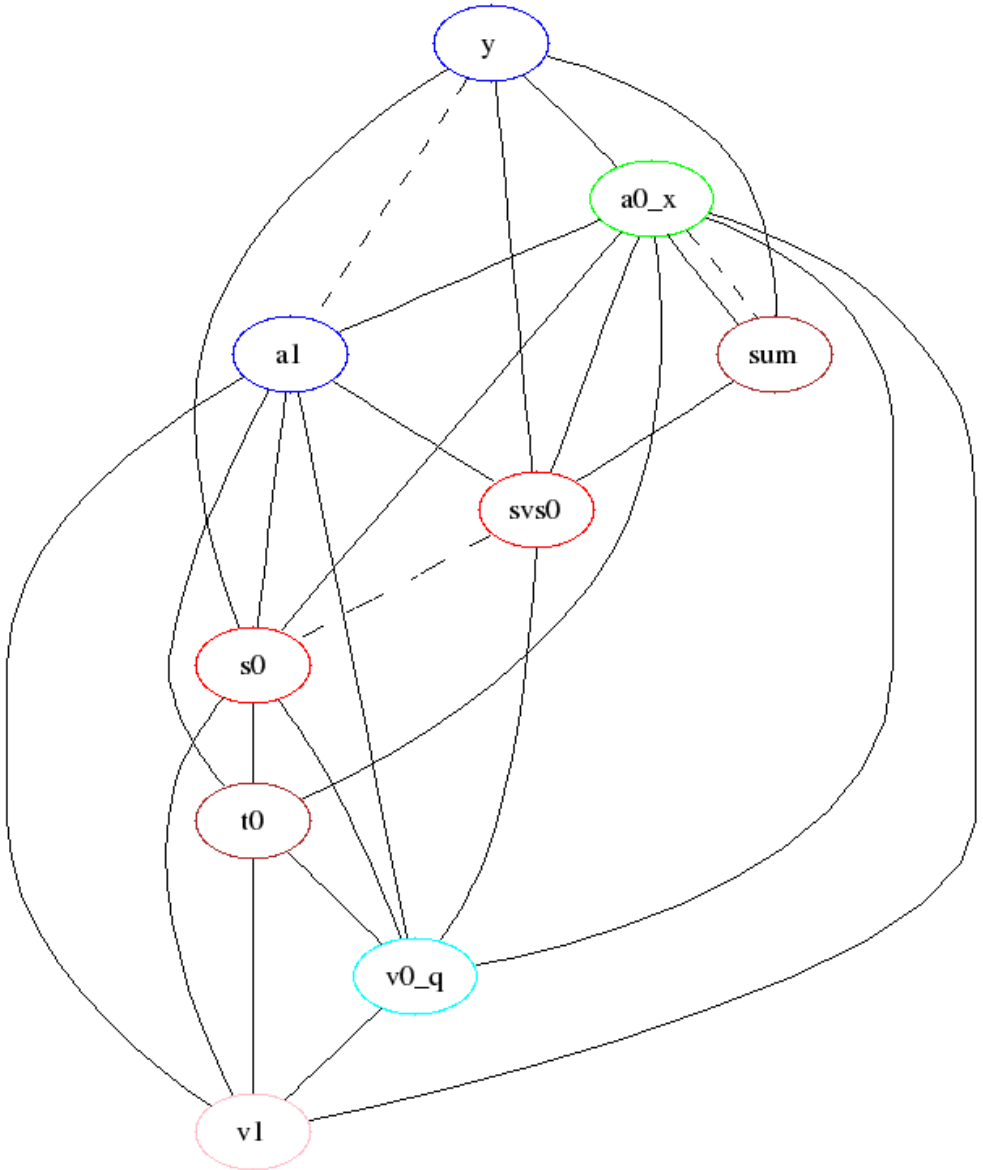


Conflict graph

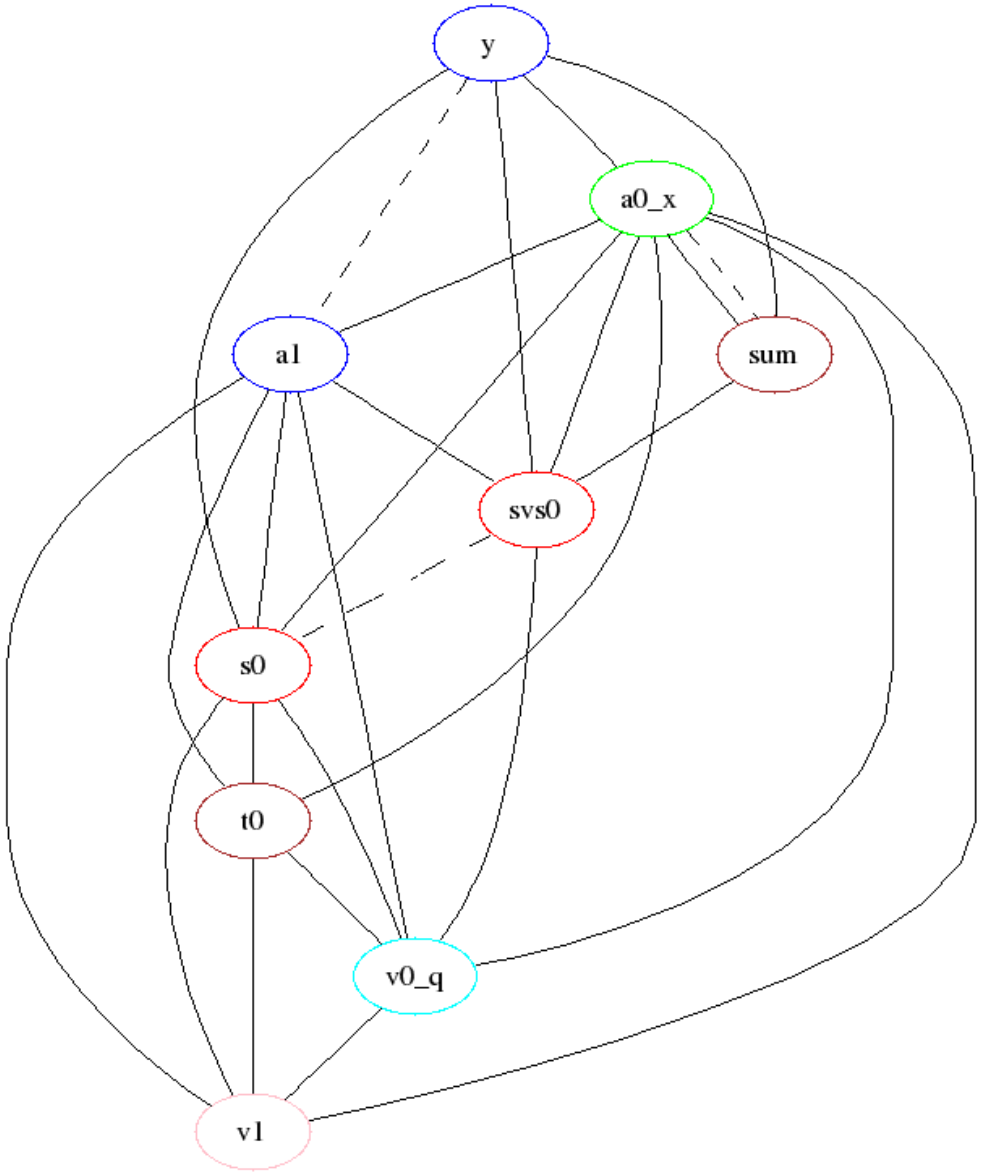


- Only precolored nodes remain.
- Assign colors in reverse order: svs0, sum, y.

Conflict graph



Conflict graph



- That's it.
- (This example didn't spill.)

The Program

Replace temps with registers, remove boring moves

```
f(x,y) {
  x = %a0 ;
  y = %a1 ;
  svs0 = %s0 ;
  sum = x + y ;
  %a0 = sum ; %a0 = %t0 ;
  %a1 = y ;
  call div
   q = %v0;
  %s0 = svs0 ;
  %v0 = q;
}
```

The Program

Replace temps with registers, remove boring moves

```
f: {


  %t0 = %a0 + %a1;


  call div


}
```