Script **generated by TTT**

Title:　　　Seidl: Programmoptimierung (16.12.2015)

Date:　　　Wed Dec 16 10:19:38 CET 2015

Duration:　09:32 min

Pages:　　　4

Example

int a, ret;　　　　　f () {
main () {　　　　　　　int b;
　　$a = 3$;　　　　　　if $(a \leq 1)$ {ret $= 1$; goto exit;}
　　f();　　　　　　　$b = a$;
　　$M[17] =$ ret;　　　$a = b - 1$;
　　ret $= 0$;　　　　　f();
}　　　　　　　　　　　ret $= b \cdot$ ret;
　　　　　　　　　　exit :
　　　　　　　　　　}

Such programs can be represented by a set of CFGs:　one for
each procedure ...

518

We extend our mini-programming language by procedures without
parameters and procedure calls.

For that, we introduce a new statement:
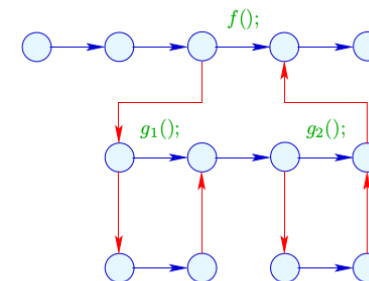
$$f();$$

Every procedure f has a definition:

$$f\,() \ \{ \ stmt^* \ \}$$

Additionally, we distinguish between global and local variables.

Program execution starts with the call of a procedure main () .

517

In order to optimize such programs, we require an extended
operational semantics.

Program executions are no longer paths, but forests:

520

... in the Example:

main()

0 → $a = 3;$
1 → $f();$
2 → $M[17] = \text{ret};$
3 → $\text{ret} = 0;$
4

$f()$

5
Neg $(a \leq 1)$ / Pos $(a \leq 1)$
6 — $b = a;$
7 — $a = b - 1;$ / $\text{ret} = 1;$
8 — $f();$
9 — $\text{ret} = b * \text{ret};$
10
11