Static Single Assignment Form

CMPT 379: Compilers Instructor: Anoop Sarkar anoopsarkar.github.io/compilers-class

Control Flow Graph (CFG)

```
int main() {
extern int f(int);
int i;
int *a;
for (i = 0;
     i < 10;
     i = i + 1
{ a[i] = f(i); }
}
```



Control Flow Graph in 3-address code



3

- *def-use* chains keep track of where variables were defined and where they were used
- Consider the case where each variable has only one definition in the intermediate representation
- One static definition, accessed many times
- Static Single Assignment Form (SSA)

- SSA is useful because
 - Dataflow analysis and optimization is simpler when each variable has only one definition
 - If a variable has N uses and M definitions (which use N+M instructions) it takes N*M to represent def-use chains
 - Complexity is the same for SSA but in practice it is usually linear in number of definitions
 - SSA simplifies the register interference graph

• Original Program

• SSA Form

a := x + y	a1 := x + y
b := a - 1	b1 := a1 - 1
a := y + b	a2 := y + b1
b := x * 4	b2 := x * 4
a := a + b	a3 := a2 + b2

what about conditional branches?



