

All programming is an exercise
in caching — Terje Mathisen

The rest is mostly bookkeeping
— Me.

The problem

- * Developing a pipeline
- * Expensive computations based on various parameters
- * Adjust script.
 - what needs to be recomputed?
 - "n11l4_scan11_11sta_4sb_unb1_3904d.nofd.dat".
- * Can we tool up and do better?

example1.toy

```
a := 1;  
fn := "datafile";  
if a == 1 then
```

```
    c := 1;
```

```
    b :=  $\emptyset$ 
```

```
else
```

```
    b := 2
```

```
end;
```

```
f := a + b;
```

```
do
```

```
    b := b + 1
```

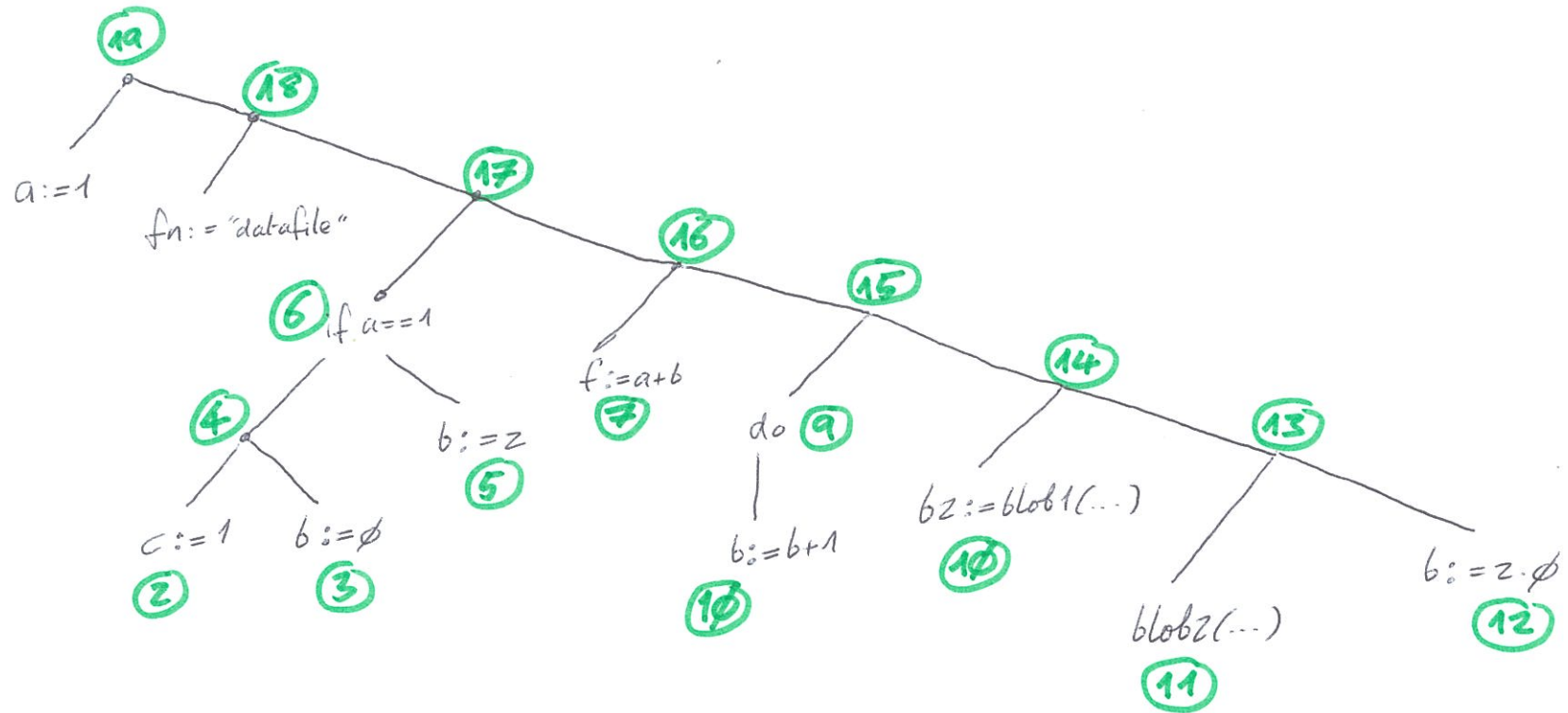
```
while b < 10
```

```
    b2 := blob1( $\emptyset$ .7, "Half", b, f);
```

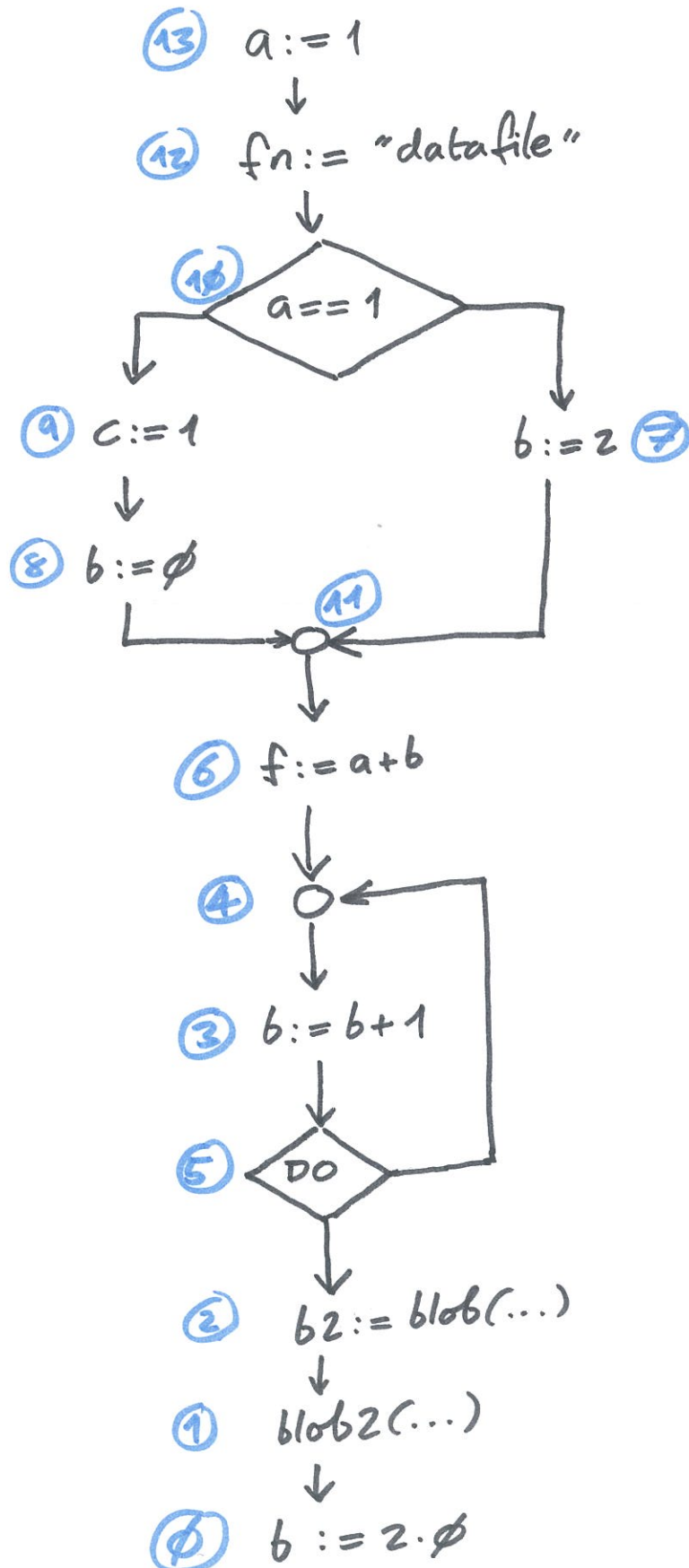
```
    blob2(data = b2, opcode = "CAL");
```

```
    b := 2.0
```

Abstract Syntax Tree 1



Control Flow Graph 1



use-def 1

Use

Def

⑩ If Test $a == 1$ → ⑬ $a := 1$

⑥ $f := a + b$ → ⑬ $a := 1$
→ ⑧ $b := \emptyset$
→ ⑦ $b := 2$

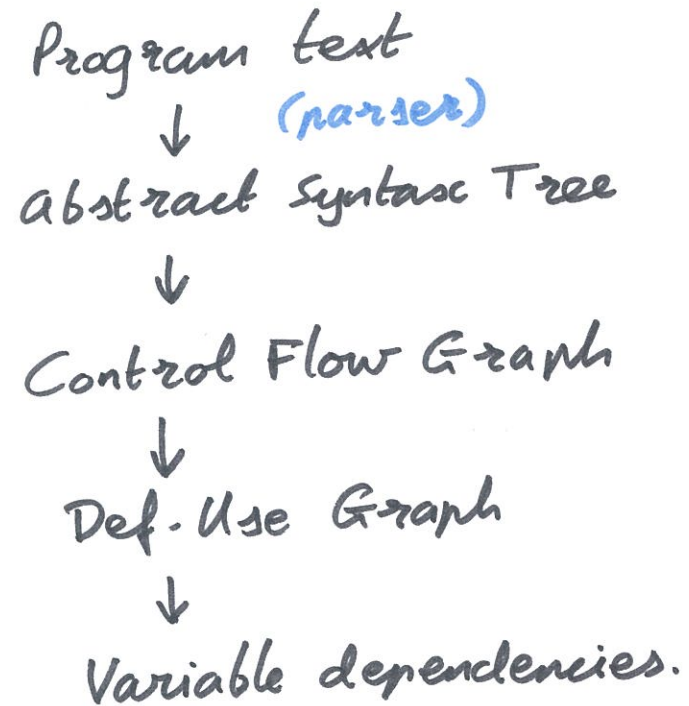
⑤ DoEnd → ③ $b := b + 1$

③ $b := b + 1$ → ⑧ $b := \emptyset$
→ ⑦ $b := 2$
→ ③ $b := b + 1$


② $b := blob1(\dots)$ → ③ $b := b + 1$
→ ⑥ $f := a + b$

① $blob2(\dots)$ → ② $b2 := blob1(\dots)$
→ ④ $b := 2.\emptyset$

"Compiler" stages



example 2. toy

```
a :=  $\emptyset$ ;  
fn := "datafile";  
if a == 1 then  
    b :=  $\emptyset$ ;   
    c := 1  
else  
    b := 2  
end;  
f := a + b;  
bz := blob( $\emptyset$ , 7, "FAIL", b, f);  
blob2(data = bz, opcode = "CAL");  
b := 3. $\emptyset$ 
```

Deleted "do"

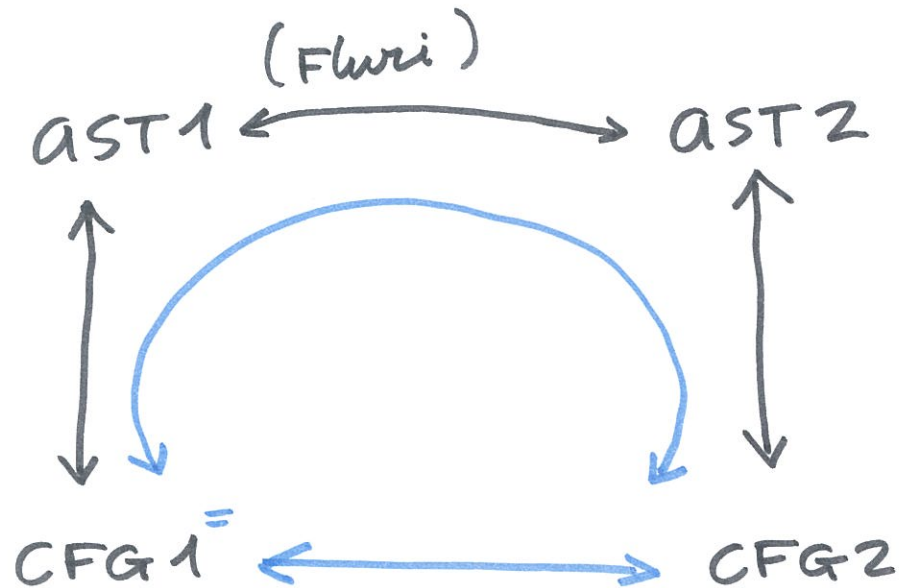
Comparing programs (1)

- * Want to compare ASTs!
(not text)
- * Compute "edit script" between ASTs.
 - Insert
 - Delete
 - align (= reorder children)
 - Move
 - Update
- * Algorithm by Fluri et al.
- * Computes "minimal edit script"
- * Fair amount of Heuristics.

Comparing programs (2)

- * Apply edit script to AST 1
 - new copy of AST 2
 - numbering consistent with AST 1.
- * Run dependency analysis on both ASTs.

Comparing programs (3)



- * Can deduce provenance of variables.
 - * Can infer which data products will be unchanged
- (- Conservative approximation)