

Understanding Git with Alloy

Milestone 2

Cláudio Lourenço Renato Neves

University of Minho
Formal Methods in Software Engineering

June 9, 2012



Table of contents

Where were we?

Progress

Current Model

The operations

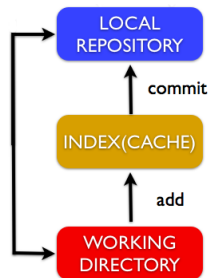
The properties

Future work

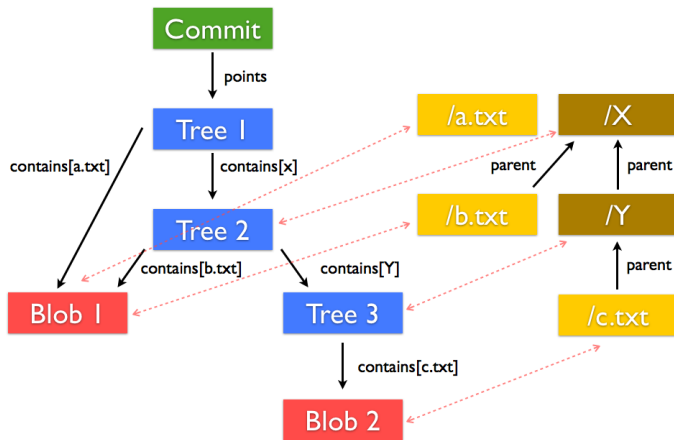


Where were we?

- Focusing on Index + Object Model
- No problem with add and rm operations
- Commit - Big Problem



The problem



How did we solve it?

Abstraction

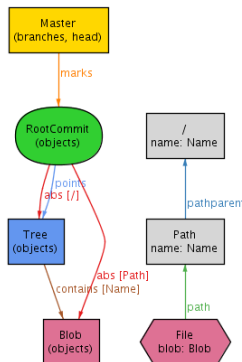
- In each commit there is relation between Objects and Paths
- Facts to reflect the parent relationship

```
sig Commit extends Object {
  points : Tree,
  parent : set Commit,
  abs: Path -> Object
}
```

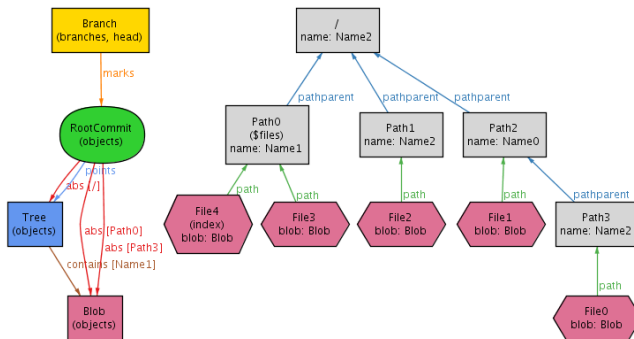
```
all c: Commit{
  let objs = c.points.*(contents.Name){
    c.abs in Path some -> lone objs
    (c.abs).(c.points) in Root
    all p,q : (c.abs).univ | p -> q in pathparent
      implies q.(c.abs) -> p.(c.abs) -> p.name in contents
    all t,o : objs, n : Name | t -> o -> n in contents
      implies all y : c.abs.t | some x : c.abs.o |
        x -> y in pathparent and x.name = n
  }
}
```



The abstraction relation



The abstraction relation



Model

```

abstract sig Object {
  objects: set State
}

sig Blob extends Object {}

sig Tree extends Object {
  contains : Name  $\rightarrow$  lone (Tree+Blob)
}

sig Commit extends Object {
  points : Tree,
  parent : set Commit,
  abs: Path  $\rightarrow$  Object
}

sig RootCommit extends Commit {}

sig Branch{
  marks: Commit one  $\rightarrow$  State,
  branches: set State,
  head: set State
}

lone sig Master extends Branch{}

```

```

sig Path {
  pathparent : lone Path,
  name : Name
}

sig File{
  path: Path,
  blob: Blob,
  index: set State
}

```



Operations - add and rm

add

Add a file with the current content to the index

```
...
index.s' = index.s +
          f - ((f.path) ~ path - f)
...
```

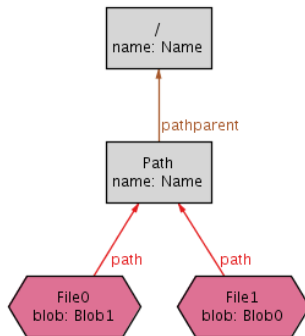
rm

Remove the file from index

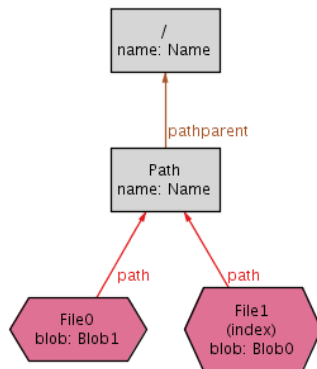
```
...
index.s' = index.s - f
...
```



Operations - add and rm



Operations - add and rm



Operations - commit

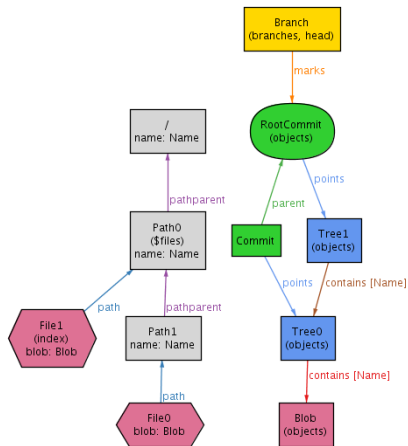
commit

Creates a commit, from the index

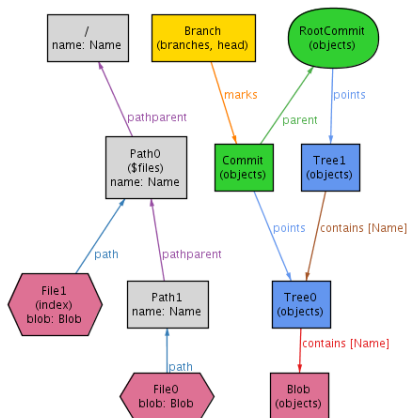
```
...  
(head.s').(marks.s').parent = (head.s).(marks.s)  
...  
(index.s).path.*pathparent = (head.s').(marks.s').abs.univ  
all f:index.s | f.path → f.blob in (head.s').(marks.s').abs  
...
```



Operations - commit



Operations - commit



Operations - branch

branch

Creates a new branch pointing to the current one

```
...  
branches.s' = branches.s + b  
marks.s' = marks.s + b  $\rightarrow$  (head.s).(marks.s)  
...
```

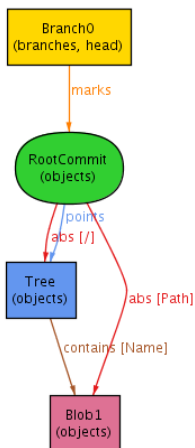
branch -d

Removes a branch if it is not pointed by the head

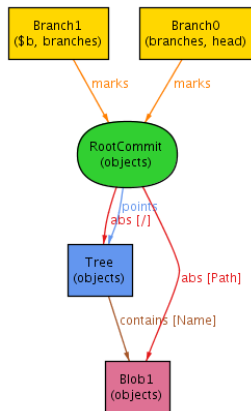
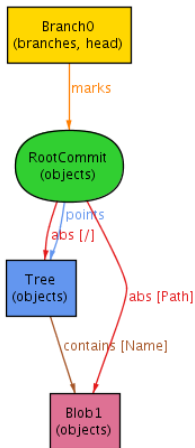
```
...  
branches.s' = branches.s - b  
marks.s' = marks.s - b  $\rightarrow$  Commit  
...
```



Operations - branch



Operations - branch



Operations - checkout

Updates the head and the index to reflect a certain commit pointed by a branch

Problems

- Difficult to understand the pre conditions
- There are no specifications

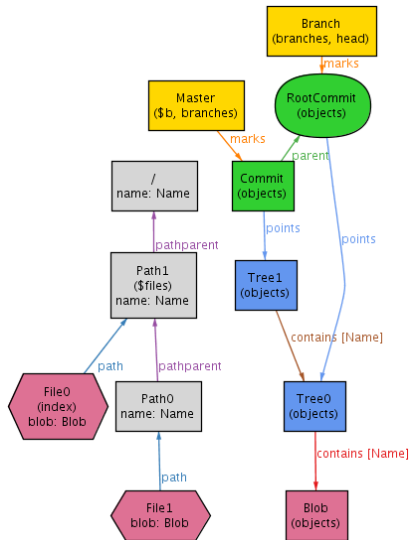


Pre conditions found relatively to the index

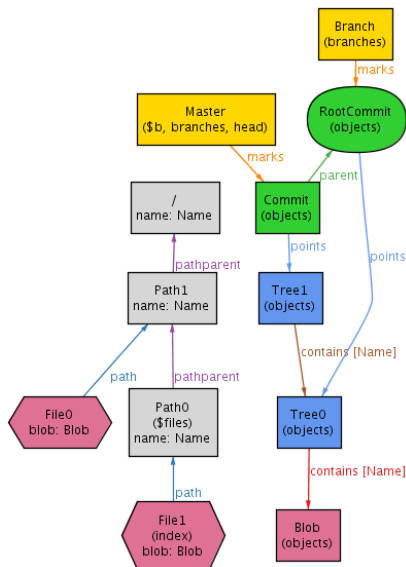
- Removed files are ignored
- Not modified files since last commit are ignored
- New files or Modified files:
 - if file exists on destination commit then:
 - the content is the same as in index, or
 - the content in the destination commit is the same as in the current commit
 - if the files does not exists in the destination commit
 - it also does not exist on the current commit



Operations - checkout



Operations - checkout



Properties



Future work



Understanding Git with Alloy

Milestone 2

Cláudio Lourenço Renato Neves

University of Minho
Formal Methods in Software Engineering

June 9, 2012

