Understanding Git with Alloy

Milestone 3

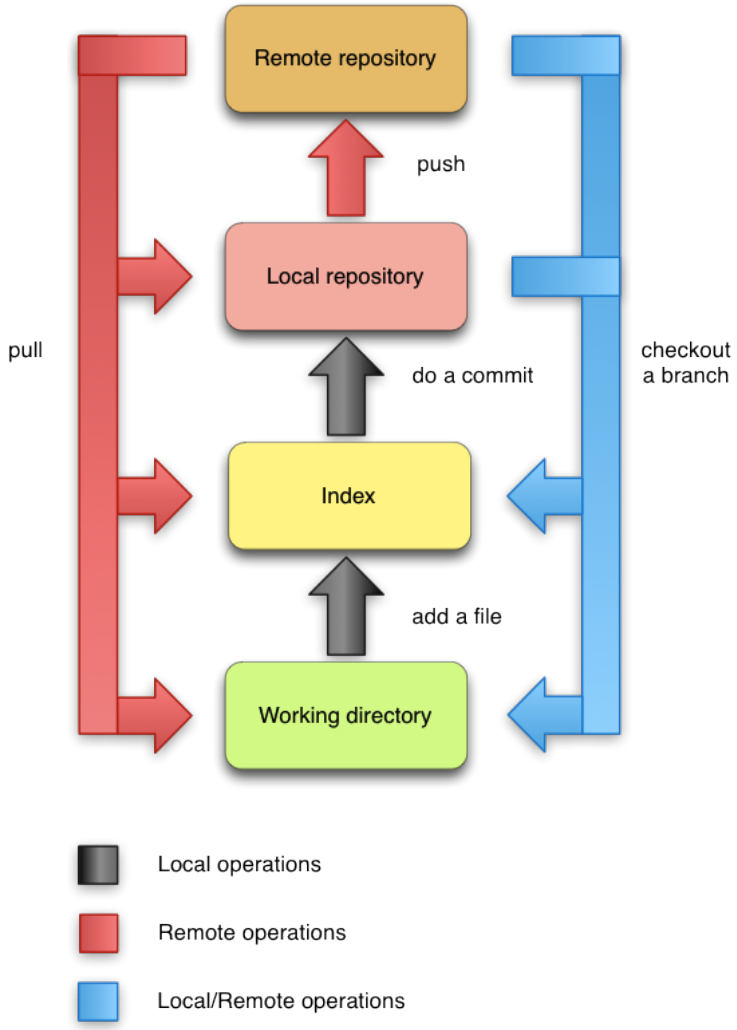
Cláudio Lourenço Renato Neves

University of Minho
Formal Methods in Software Engineering

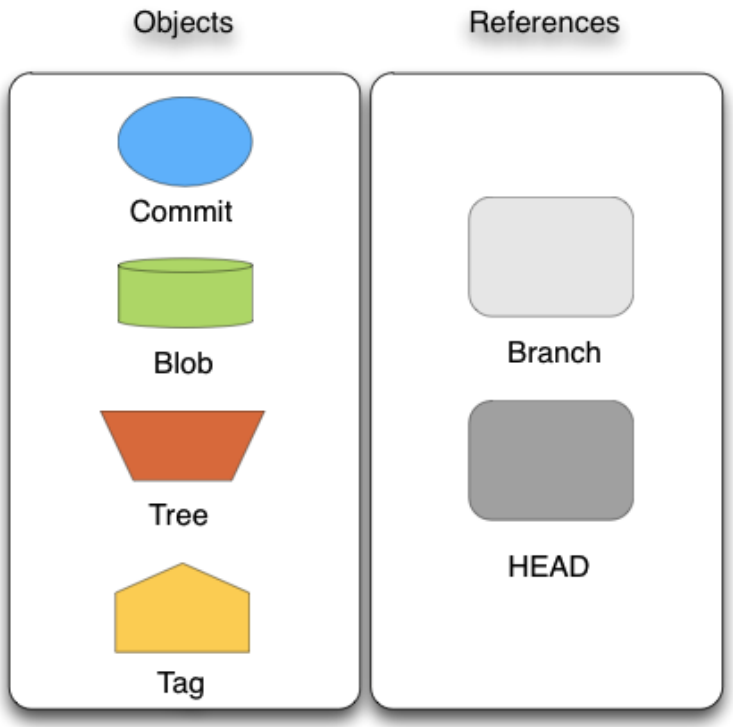
July 10, 2012

Table of contents

The Git Structure



Repository



Blob and Tree

Blob

- Represents the content of a file;
- The name is calculated from its content;

```
sig Blob extends Object {}
```

Tree

- Relation from names to Blobs or/and Trees;
- Used to represent the file system structure;

```
sig Tree extends Object {
   contains: Name -> lone(Tree+Blob)
}
```

Commit

- It is like a snapshot of the project on a certain moment in time;
- Author, Committer, Comment - Not important for us;
- Parent - The Commit which originated the current;
- Tree - Pointer to a Tree Object;

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

sig RootCommit extends Commit {}
```

Branch and HEAD

Branch

- It is just a pointer to a commit;

HEAD

- Special reference that identifies the current Branch;

```
sig Branch{
    marks: Commit lone -> State,
    branches: set State,
    head: set State
}

lone sig Master extends Branch{}
```

Working Directory

- Subset of a file system with the content of a project;
- These files can be the current files or files retrieved from the repository.

```
sig Path {
    pathparent: lone Path,
    name: Name,
    unmerge: set State
}

one sig Root extends Path{}
```

Index

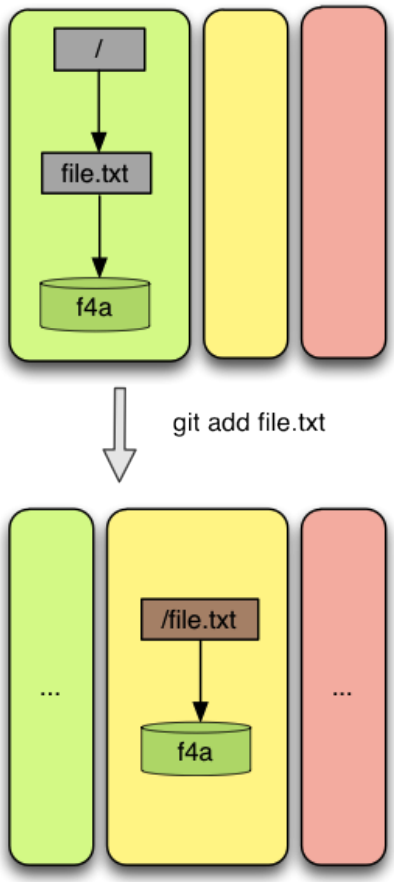
- Something in between the working directory and repository;
- It keeps a relation from file to content;
- The files in index will be in the next commit;

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

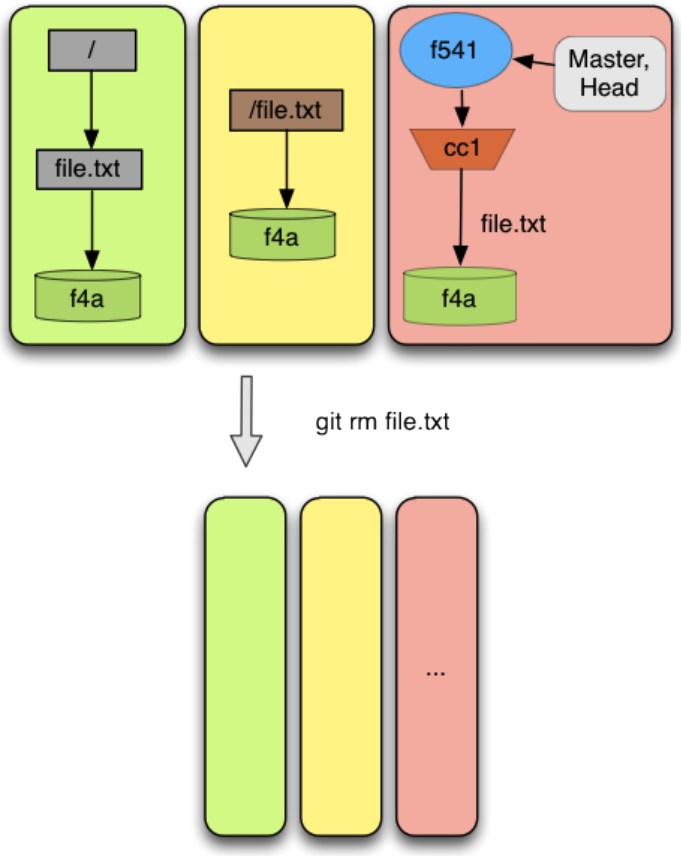
Modeled Operations

- Add and Remove
- Commit
- Branch and Branch Remove
- Checkout
- Merge

Add



Remove


/
file.txt
f4a
/file.txt
f4a
f541
Master,
Head
cc1
file.txt
f4a
git rm file.txt
...