Modelling the Git core system with Alloy

Cláudio Lourenço, Renato Neves
Universidade do Minho

April 7, 2012

1 Abstract

2 Modelling Git Object Model

2.1 What says in the book [1]

For each part of a textual specification we will associate an Alloy specification. The specification comes from [1].

"All the information needed to represent the history of a project is stored in files referenced by a 40-digit "object name"...."

```
sig Sha{}
```

"..and there are four different types of objects: "blob", "tree", "commit", and "tag"."

```
abstract sig Object {
        namedBy : one Sha
}

sig Blob extends Object{}

sig Tree extends Object {}

sig Commit extends Object{}

sig Tag extends Object{}
```

"A "blob" is used to store file data - it is generally a file."

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

"A "tree" is basically like a directory - it references a bunch of other trees and/or blobs..."

```
sig Tree extends Object {
        references : set (Tree+Blob)
}
```

"A "commit" points to a single tree...."

```
sig Commit extends Object{
        points : one Tree
}
```

"A "tag" is a way to mark a specific commit..."

```
sig Tag extends Object{
        marks : one Commit
}
```

Next, as the book [1] says, a "tree" acts like a directory, so it or it's descendents cannot point to itself.

```
no ^references & iden
```

"...two "trees" have the same SHA1 name if and only if their contents (including, recursively, the contents of all subdirectories) are identical."

```
all t,t' : Tree | t.namedBy = t'.namedBy <=> t.references = t'.references
```

"What that means to us is that is virtually impossible to find to different objects with the same name"

```
namedBy.~namedBy - (Tree->Tree) in iden
```

2.2 What we assume...

Blobs must have at least one parent

```
Blob in Tree.references
```

A root tree can only have one parent commit

```
points.~points in iden
```

A tree or is root or must have a parent

```
Tree in Tree.references + Commit.points
```

A sha must be generated from an object

```
Sha in Object.namedBy
```

3 Conclusions

References

[1] *Git Community Book.*