[320] Branching and Merging
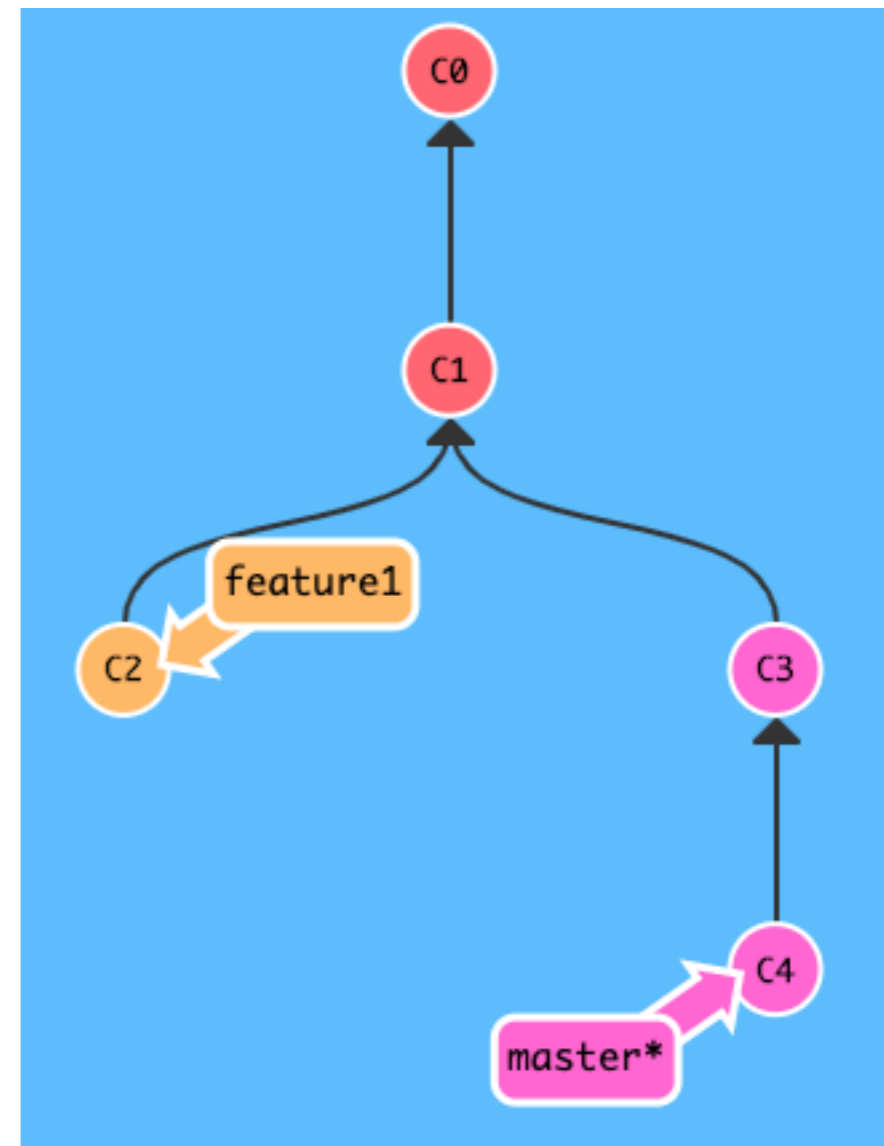
Tyler Caraza-Harter

Practice Branching
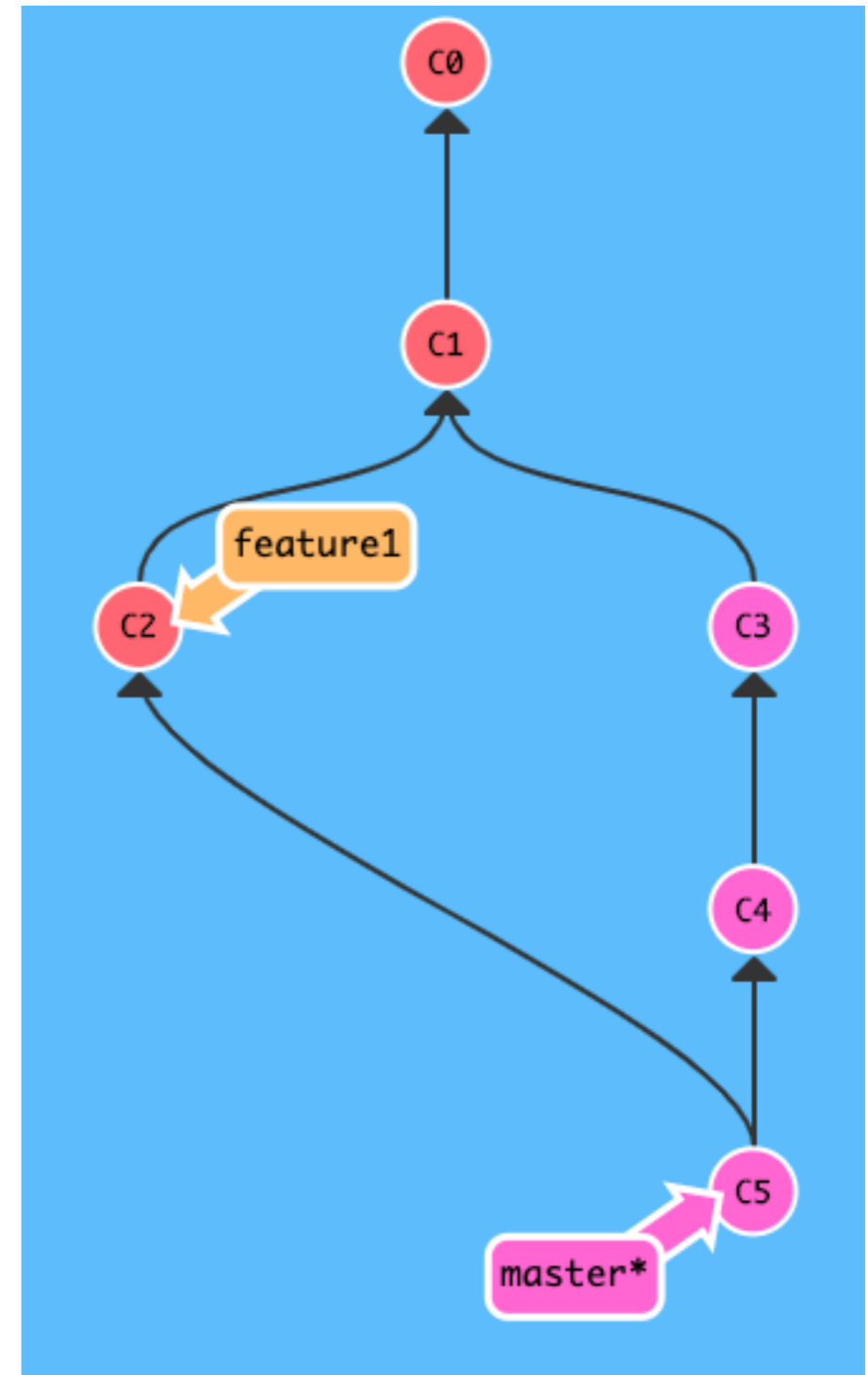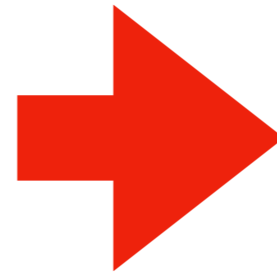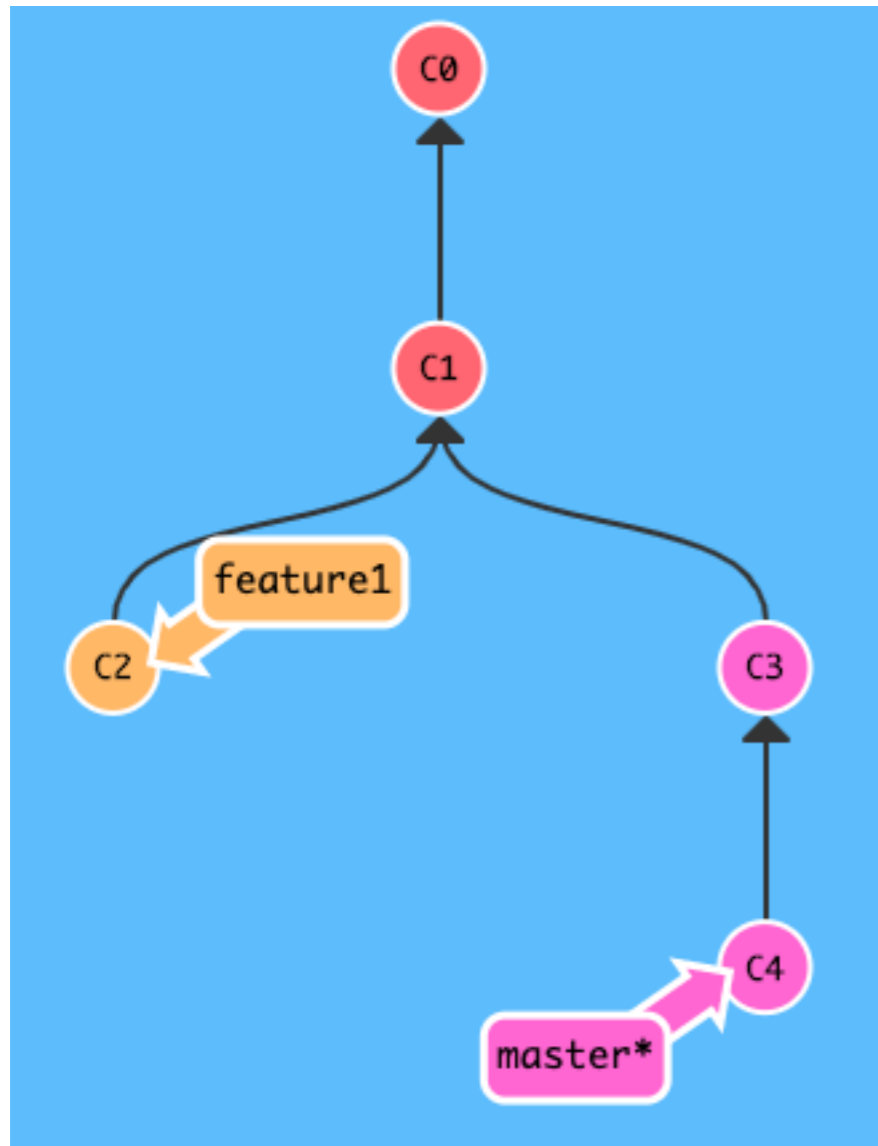
Git equivalent of PythonTutor:

https://learngitbranching.js.org/?NODEMO

Merging without Conflicts

Switch branch

```
git merge frombranch
```
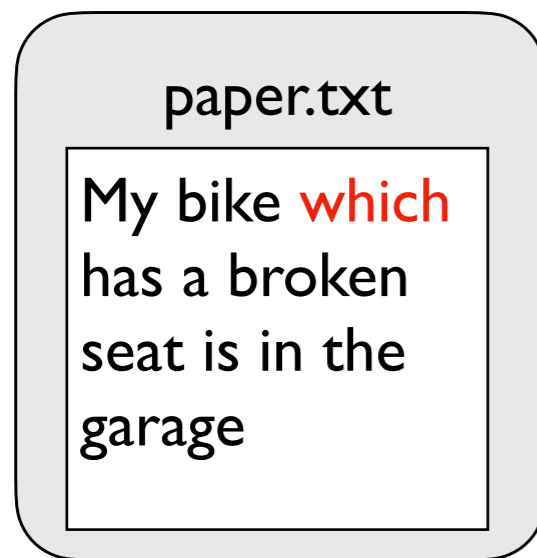
add whatever is there to the current branch

tip (or learn vim):
```
export EDITOR=nano
```

Merging with Conflicts

What happens when two people try to fix the same issue, in two different (incompatible) ways?

master **branch**

paper.txt

My bike which has a broken seat is in the garage

time

Merging with Conflicts

What happens when two people try to fix the same issue, in two different (incompatible) ways?

master **branch**

paper.txt

My bike, which has a broken seat, is in the garage.

paper.txt

My bike which has a broken seat is in the garage

paper.txt
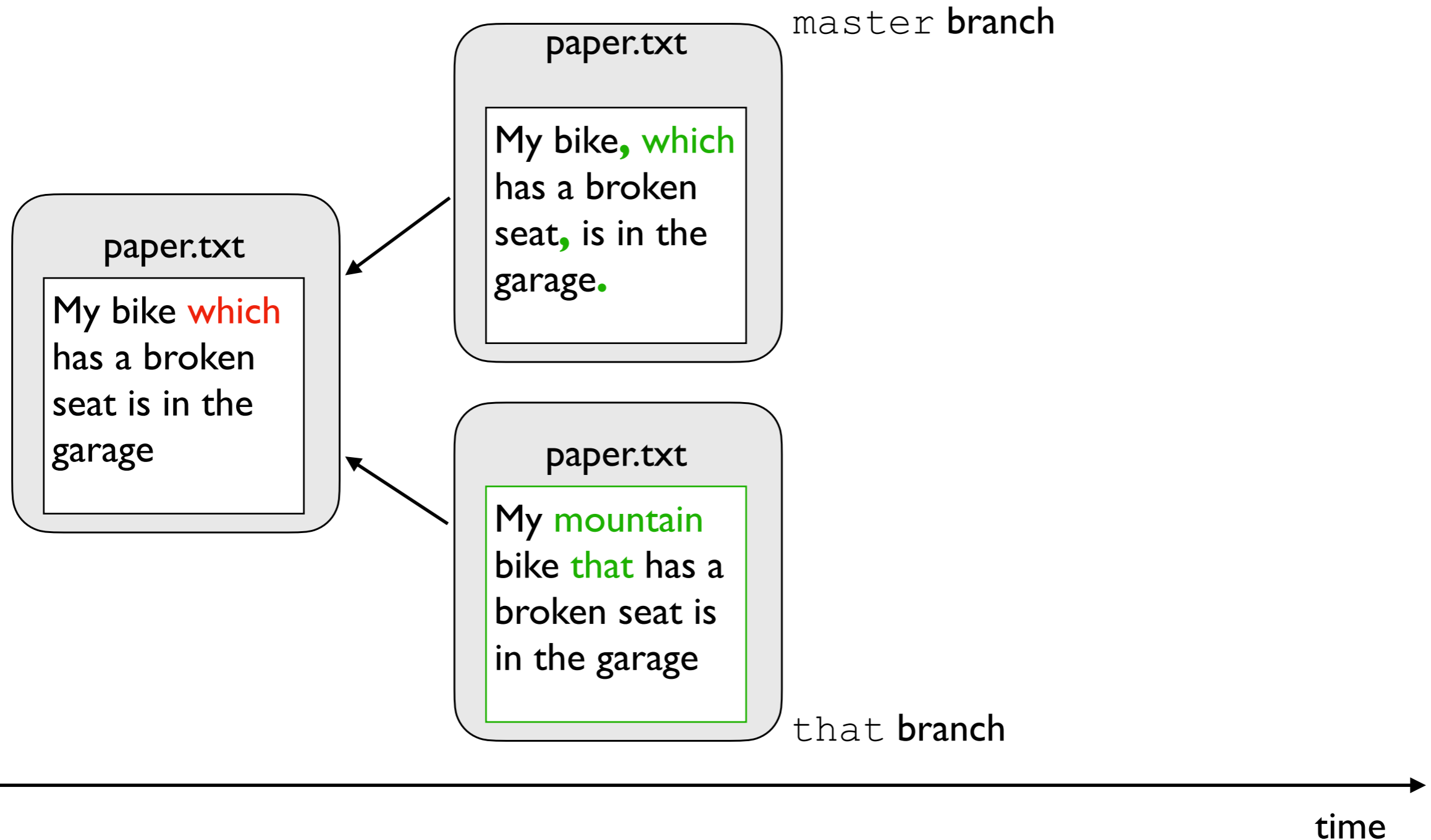
My mountain bike that has a broken seat is in the garage

that **branch**

time

Merging with Conflicts

What happens when two people try to fix the same issue, in two different (incompatible) ways?
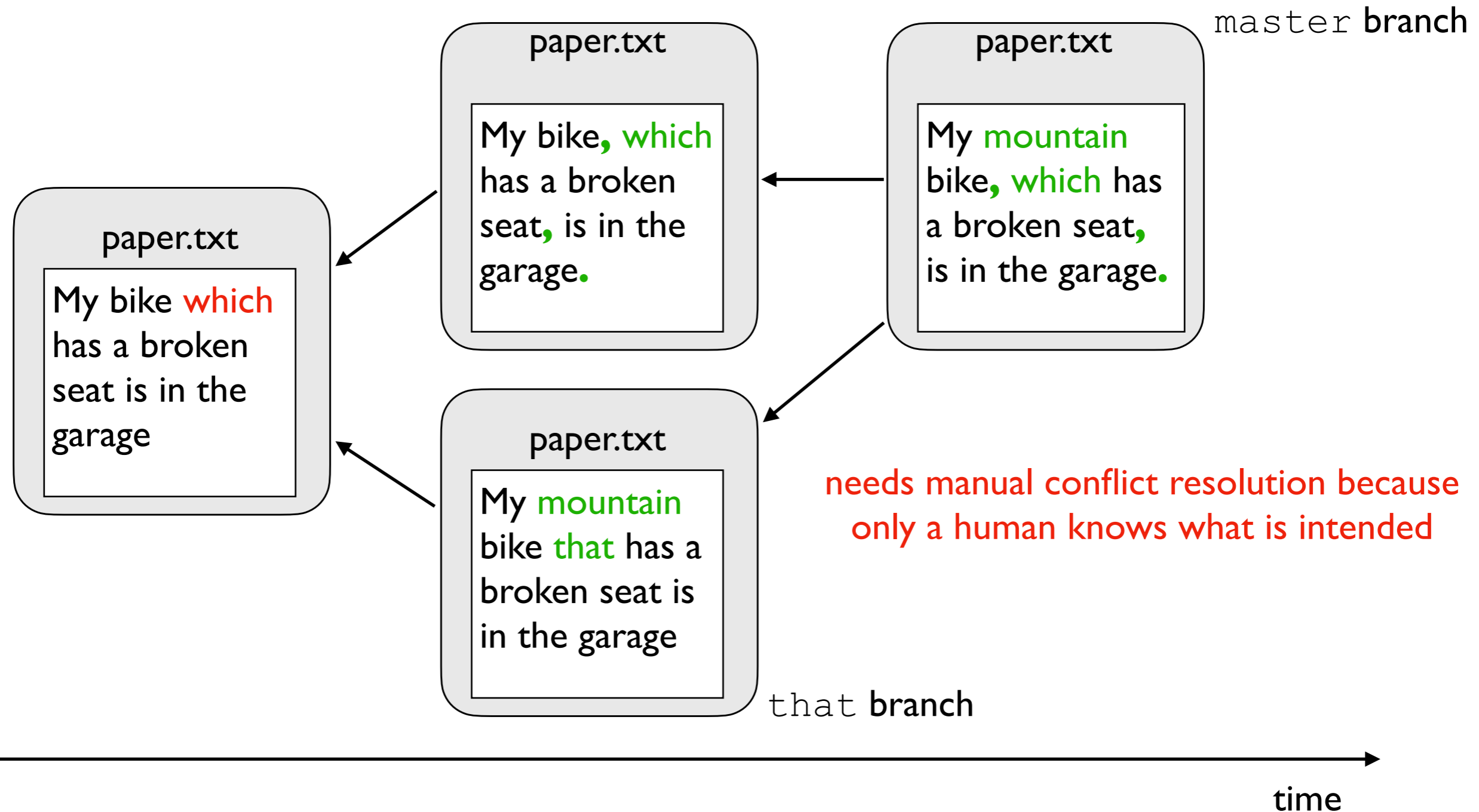

master **branch**

paper.txt

My bike, which has a broken seat, is in the garage.

paper.txt

My mountain bike, which has a broken seat, is in the garage.

paper.txt

My bike which has a broken seat is in the garage

paper.txt

My mountain bike that has a broken seat is in the garage

needs manual conflict resolution because only a human knows what is intended

that **branch**

time
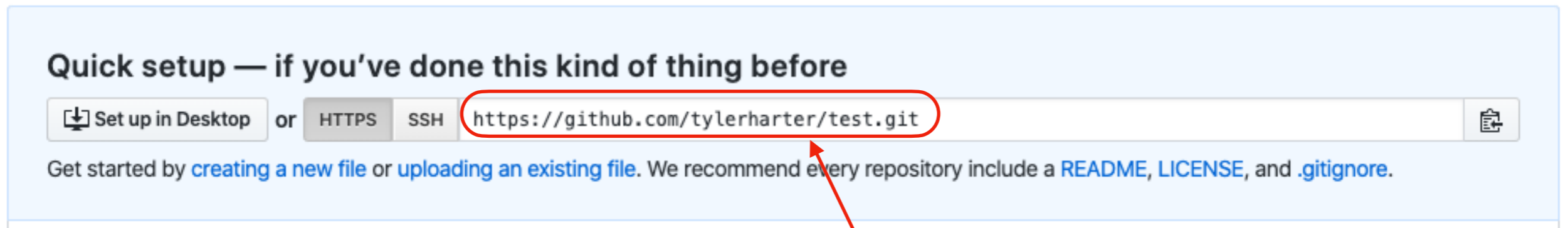
Remotes

We will often want to work on our laptops, but also have our repositories on GitHub (or similar)

Create GitHub account, go here: https://github.com/new

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/tylerharter/test.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

Pushing a branch to GitHub

```
git remote add github https://github.com/tylerharter/test.git
```

```
git push github master
```

remote name

branch name

see reading about naming in software

Summary of Terms

commit: a snapshot of files at a point in time

HEAD: a convenient label for the current commit

tag: a long-term label associated with a commit

branch: a label attached to a commit that re-attaches to new commits

merge: to combine changes on another branch into the current branch

conflict: differences that cannot automatically be merged

Challenges: https://learngitbranching.js.org/?NODEMO