Recap

- git: A tool for version control
- Files can be untracked, unmodified, modified, staged
- Collect changes for committing: Staging area
- Mark a set of changes as one “commit”
- Continue development in a secondary “branch”

UNIVERSITÄT
ZU KÖLN

Session 3: Git Merging

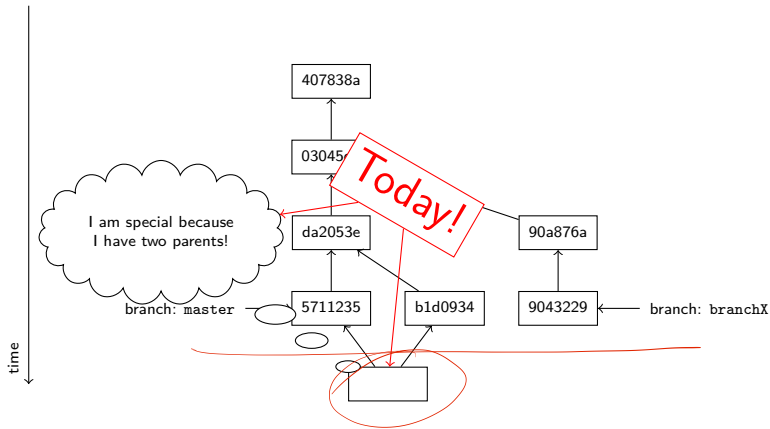
Fortgeschrittene Programmierung (Java 2)

Nils Reiter
nils.reiter@uni-koeln.de

24. April 2024

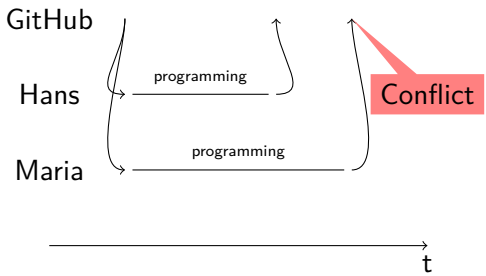
INSTITUT FÜR
DIGITAL HUMANITIES
UNIVERSITÄT ZU KÖLN

Branches


time
407838a
03045
Today!
I am special because
I have two parents!
da2053e
90a876a
branch: master
5711235
b1d0934
9043229
branch: branchX

Merging

Situations



Conflict resolution options

- Ignore, let Maria overwrite Hans' code (this is bad!)
- Create a second copy (this is what Dropbox does)
- Force Maria to *explicitly* merge the code (this is what git does)

demo

Merging

- Merging is done line-wise
- Merging is done *into* a branch
  - I.e., you checkout the branch you want to merge into

Merging

- Merging is done line-wise
- Merging is done *into* a branch
  - I.e., you checkout the branch you want to merge into
- First attempt (by git): Automatic merging
  - Changes that are non-conflicting can be merged automatically
  - Non-conflicting: In different files, or different regions of a file
- Second step (by you): Look at conflicting changes
  - Conflicting: Lines that are close have been changed independently
  - Close: 2-3 lines apart

Merging

- Merging is done line-wise
- Merging is done *into* a branch
  - I.e., you checkout the branch you want to merge into
- First attempt (by git): Automatic merging
  - Changes that are non-conflicting can be merged automatically
  - Non-conflicting: In different files, or different regions of a file
- Second step (by you): Look at conflicting changes
  - Conflicting: Lines that are close have been changed independently
  - Close: 2-3 lines apart
- GUI tools really help with merging

Merging

- Merging is done line-wise
- Merging is done *into* a branch
  - I.e., you checkout the branch you want to merge into
- First attempt (by git): Automatic merging
  - Changes that are non-conflicting can be merged automatically
  - Non-conflicting: In different files, or different regions of a file
- Second step (by you): Look at conflicting changes
  - Conflicting: Lines that are close have been changed independently
  - Close: 2-3 lines apart
- GUI tools really help with merging

Conflicts when merging are **not** not signs that you made mistakes, but are a naturally occurring phenomenon.

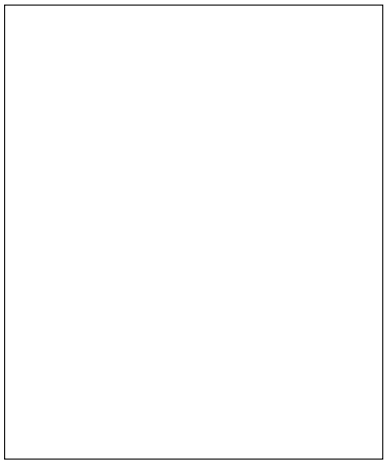
Merging

original

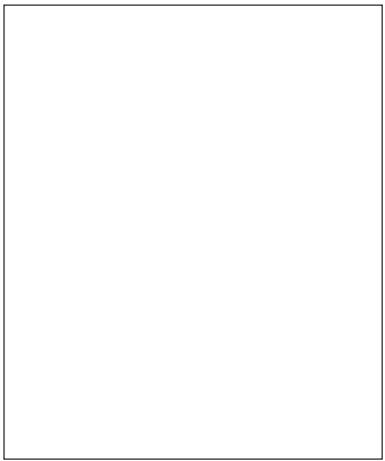
```
def add(x,y):
    return x+y

for i in range(10):
    d = add(i,i*2)
    print(d)
```

Merging

maria



hans



1. Create branches

Merging

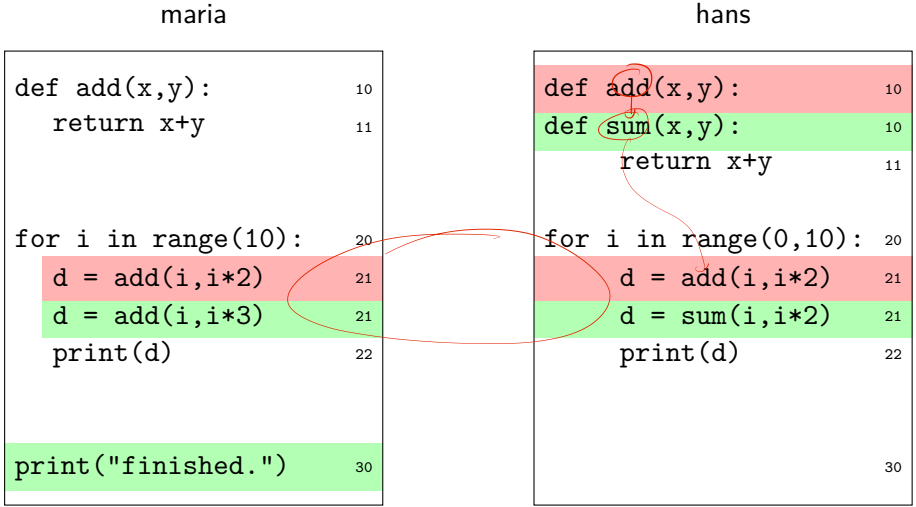
maria

```
def add(x,y):                10
   return x+y                11


for i in range(10):          20
   d = add(i,i*2)            21
   d = add(i,i*3)            21
   print(d)                  22


print("finished.")           30
```

hans

1. Create branches
2. M. codes & comm

Merging



1. Create branches
2. M. codes & comm
3. H. codes & commi

Merging

maria

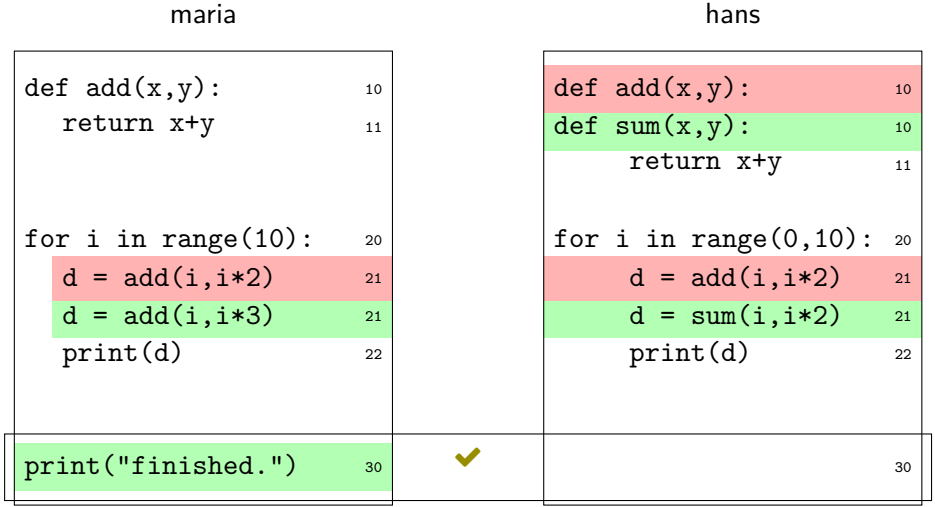
```
def add(x,y):                    10
   return x+y                    11


for i in range(10):              20
   d = add(i,i*2)                21
   d = add(i,i*3)                21
   print(d)                      22


print("finished.")               30
```

hans

```
def add(x,y):                    10
def sum(x,y):                    10
      return x+y                 11

for i in range(0,10):            20
      d = add(i,i*2)             21
      d = sum(i,i*2)             21
      print(d)                   22


                                 30
```

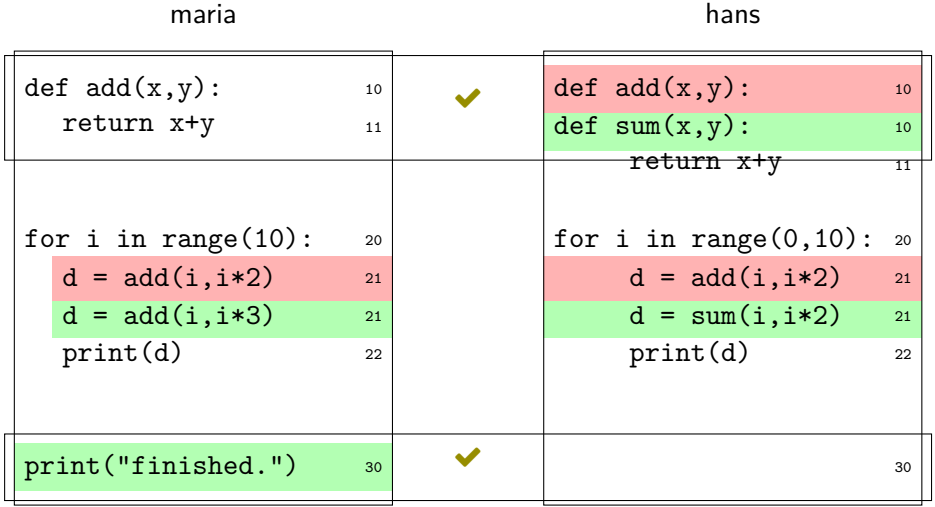
1. Create branches
2. M. codes & comm
3. H. codes & commi
4. Someone merges

Merging



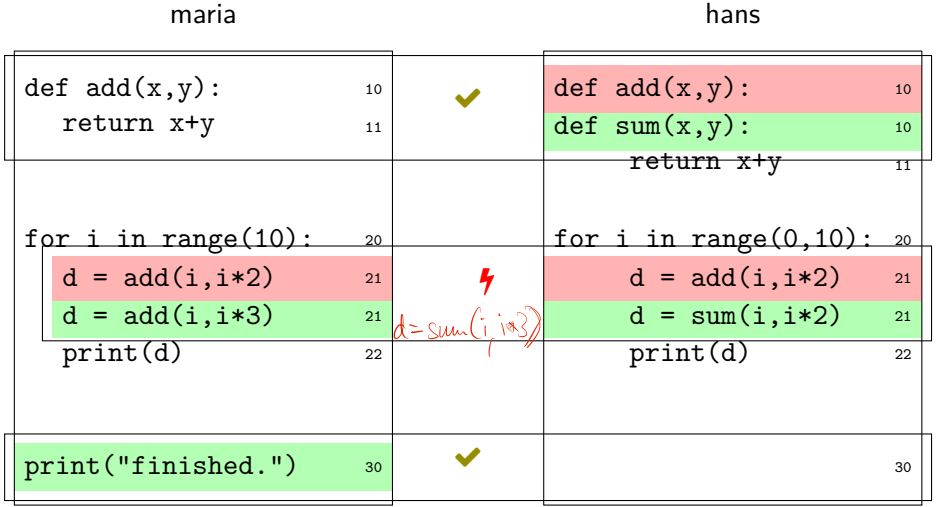
1. Create branches
2. M. codes & comm
3. H. codes & comm
4. Someone merges
   a. No problem

Merging



1. Create branches
2. M. codes & comm
3. H. codes & comm
4. Someone merges
   a. No problem
   b. No problem

Merging



1. Create branches
2. M. codes & comm
3. H. codes & comm
4. Someone merges
   a. No problem
   b. No problem
   c. Conflict

Merging

with local branches

Fred runs: `git merge maria`

```
1  def sum(x,y):
2      return x+y
3
4
5  for i in range(0,10):
6  <<<<<<< HEAD
7      d = sum(i,i*2)
8  =======
9      d = add(i,i*3)
10 >>>>>>> maria
11     print(d)
12
13
14 print("finished.")
```

← merged automatically

conflict, we need to take care of this manually

← merged automatically

Section 1

Remotes

Decentralized

- “Git is decentralized”: What does this mean exactly?

Decentralized

- “Git is decentralized”: What does this mean exactly?
- No central server required
- A local git repository stores the entire history, all branches and tags
- Every clone of the repository has the entire history
  - Offline working galore!

Remotes

- Each repository can be associated with multiple ‘remotes’
  - Usually, one remote is called ‘origin’
- `clone` makes a local clone *and* sets one remote to point to the source

Remotes

- Each repository can be associated with multiple ‘remotes’
  - Usually, one remote is called ‘origin’
- `clone` makes a local clone *and* sets one remote to point to the source
- Merging works across remote repositories
  - E.g., you can merge something from a remote branch into your local branch

Downloading stuff

- A branch can be set to 'track' a remote branch
  - Typically, you want the branches to have the same name
- `git fetch` downloads all tracked branches to your local repository, but keeps your working copy as it is
- `git pull` fetches the changes from the server *and* merges them into your working copy
  - Merge conflicts can occur!
- `git push` pushes your local changes to the tracking branch on the server
  - If the remote branch moved on, you'll be forced to pull and merge first

Section 2

Summary

Summary

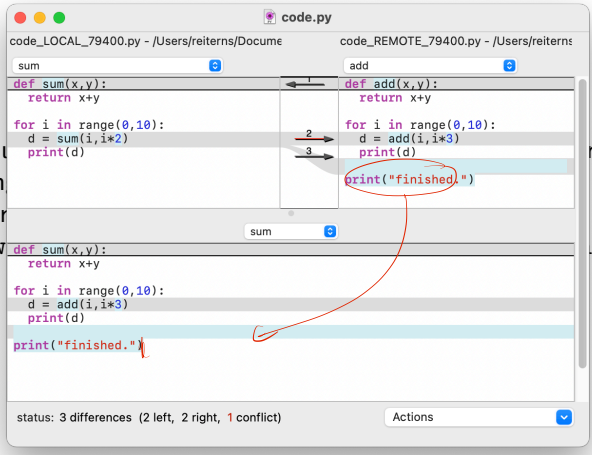
- Merging
  - Git attempts automatic merging of changed lines in different files or file sections
  - Manual merging requires attention and care, but is doable

Summary

- Merging
  - Git attempts automatic merging of changed lines in different files or file sections
  - Manual merging requires attention and care, but is doable
  - GUIs help (search for 'diff tools') or check `https://en.wikipedia.org/wiki/Comparison_of_file_comparison_tools`

Summary

- Merging
  - Git attempts au … r file sections
  - Manual mergin…
  - GUIs help (sear… https://en.w… …rison_tools

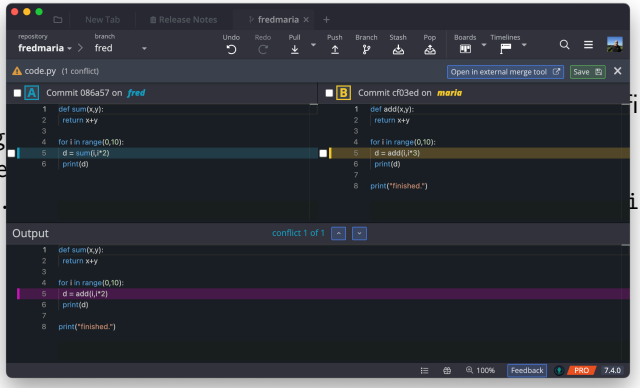


FileMerge on Mac OS (part of XCode)

Summary

- Merging
  - Git attempts
  - Manual merg
  - GUIs help (se

  `https://en.`



GitKraken

Summary

- Merging
  - Git attempts automatic merging of changed lines in different files or file sections
  - Manual merging requires attention and care, but is doable
  - GUIs help (search for ‘diff tools’) or check `https://en.wikipedia.org/wiki/Comparison_of_file_comparison_tools`
  - Coordination in a group helps to minimize merge effort

Summary

- Merging
  - Git attempts automatic merging of changed lines in different files or file sections
  - Manual merging requires attention and care, but is doable
  - GUIs help (search for 'diff tools') or check `https://en.wikipedia.org/wiki/Comparison_of_file_comparison_tools`
  - Coordination in a group helps to minimize merge effort
- Remotes
  - Entire repository can be synchronized with remote repositories
  - If a branch tracks a remote branch, changes can be pulled directly
    - This may result in a merge conflict, if the remote branch has been altered!

Exercise

https://github.com/idh-cologne-java-2-summer-2024/exercise-03