Hinter den Kulissen von Git

Friedrich Gräter

Motivation

Motivation

- Fehler: „Reference is not a tree”

Motivation

- Fehler: „Reference is not a tree”
- Sync-Verfahren?

Motivation

- Fehler: „Reference is not a tree”
- Sync-Verfahren?
- Elegante Lösung

Git?

Git?

- Versionierung von Quelltexten

Git?

- Versionierung von Quelltexten
- Mehrere getrennte Versionsverläufe

Git?

- Versionierung von Quelltexten
- Mehrere getrennte Versionsverläufe
- Unabhängig von zentralem Server

Git?

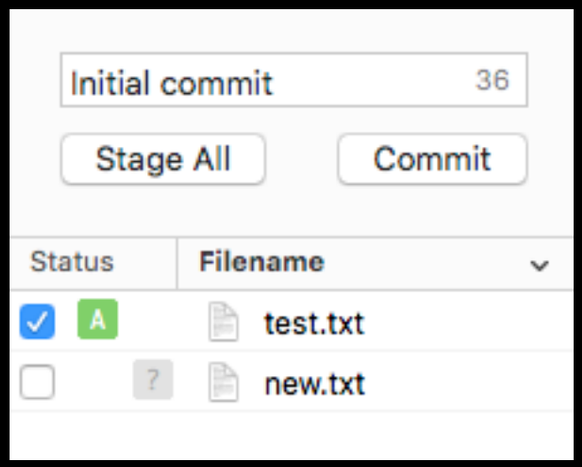
- Versionierung von Quelltexten
- Mehrere getrennte Versionsverläufe
- Unabhängig von zentralem Server
- Abgleich durch Merge / Push / Pull

Aufbau

Aufbau

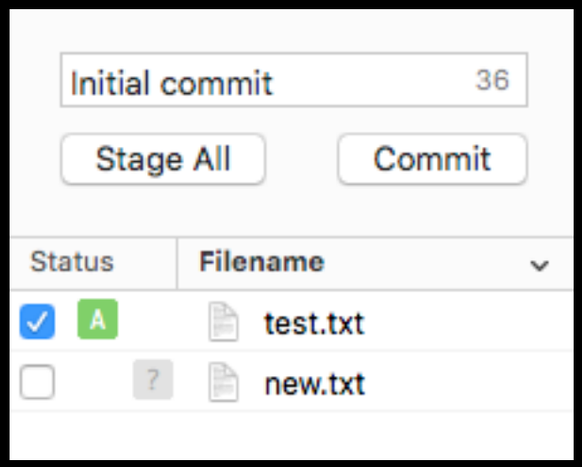
Working Directory

Aufbau



Working Directory

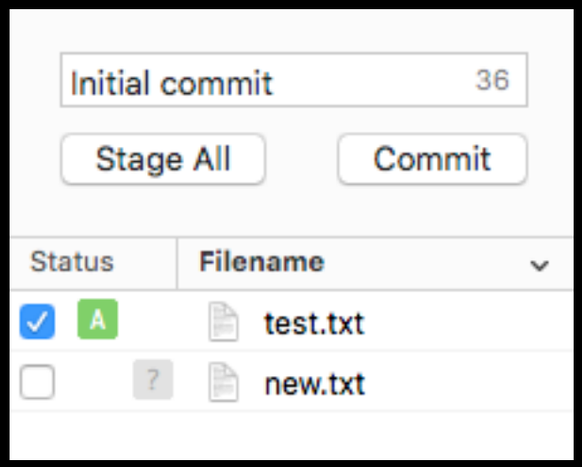
Aufbau



`git add`

Working Directory

Aufbau

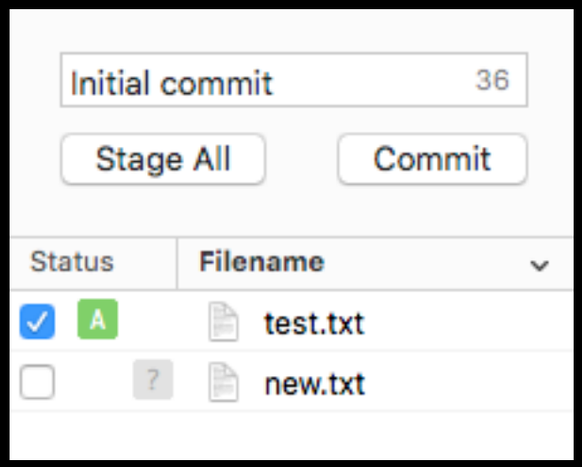


`git add`

Working Directory

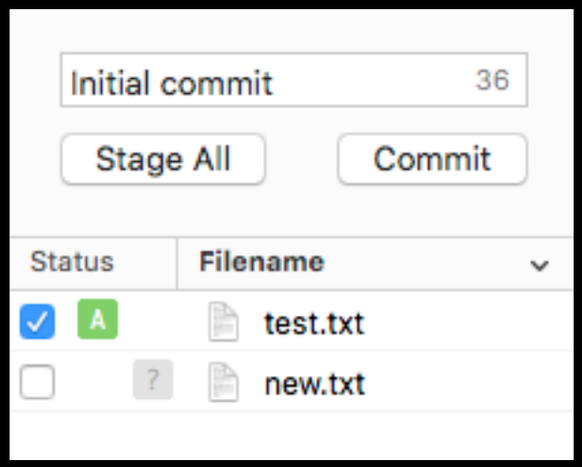
Staging Area

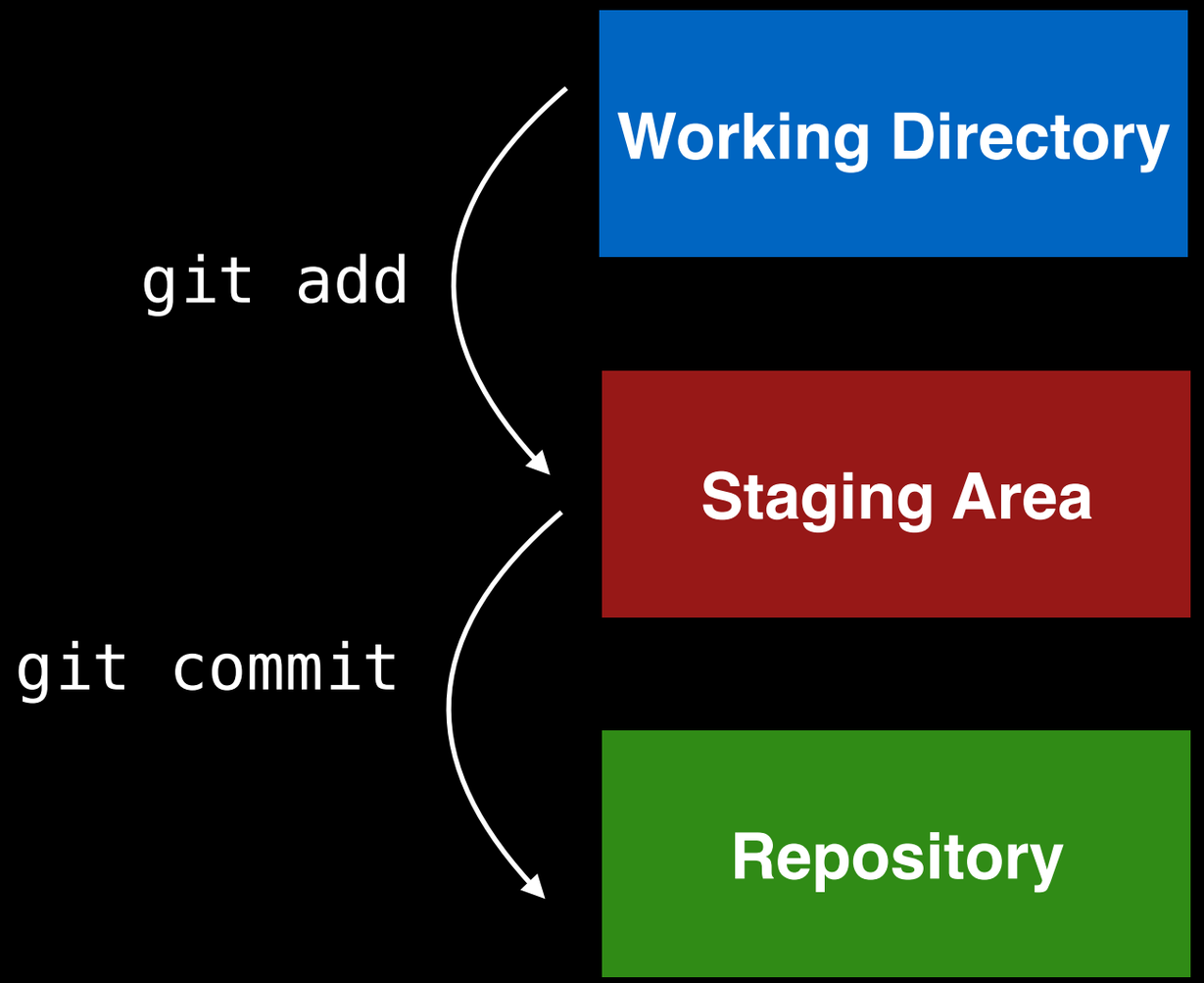
Aufbau


Initial commit
36
Stage All
Commit
Status
Filename
A
test.txt
?
new.txt

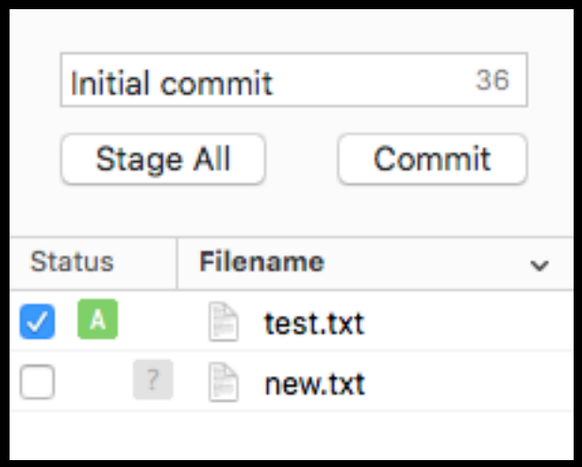
Working Directory
git add
Staging Area
git commit

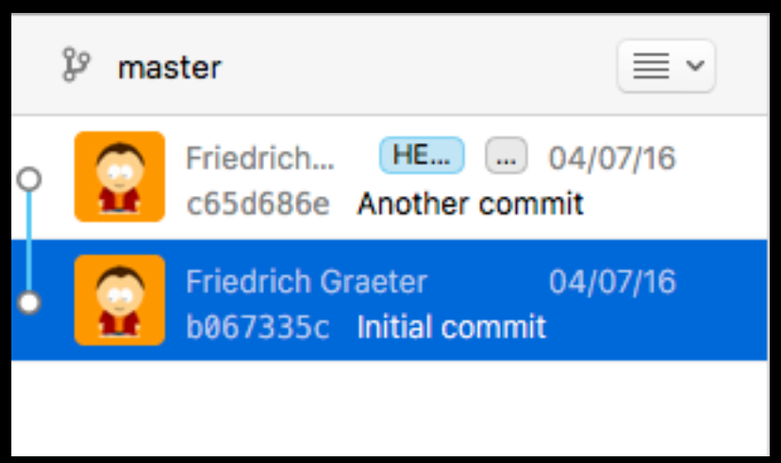
Aufbau

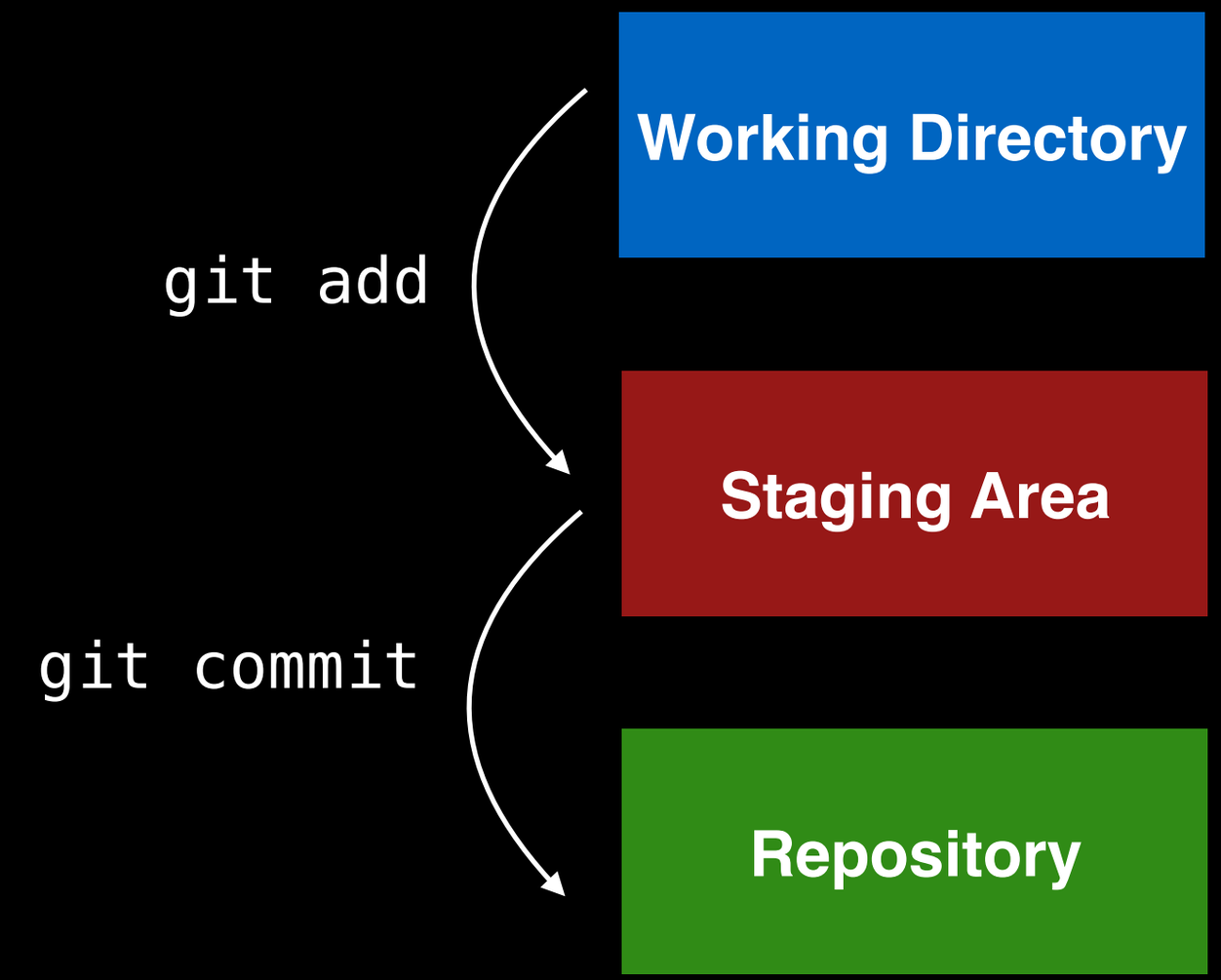

Initial commit
36
Stage All
Commit
Status
Filename
test.txt
new.txt


Working Directory
git add
Staging Area
git commit
Repository

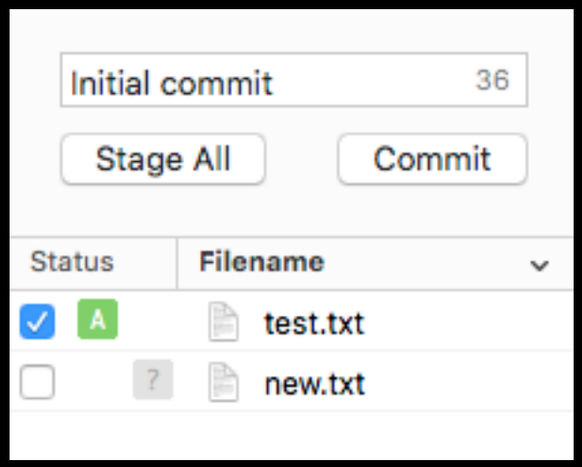
Aufbau

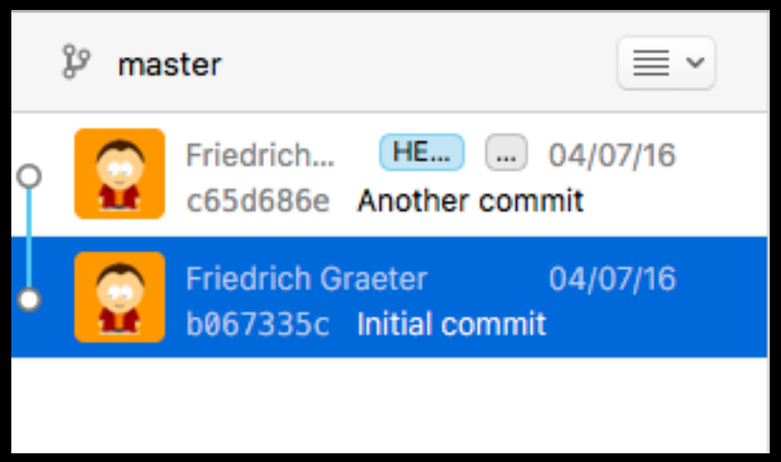

Initial commit
36
Stage All
Commit
Status
Filename
A
test.txt
?
new.txt

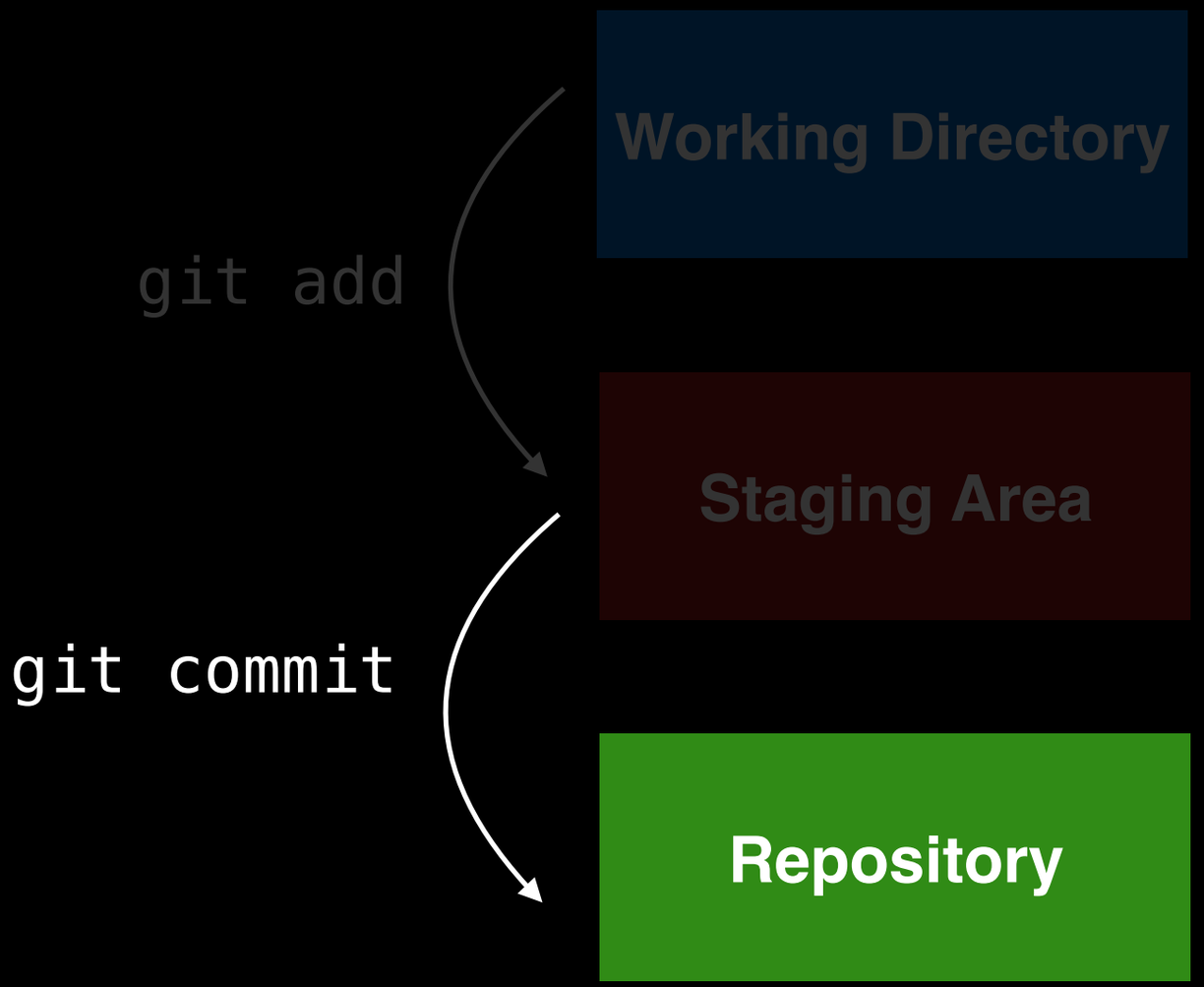

master
Friedrich...
HE...
...
04/07/16
c65d686e
Another commit
Friedrich Graeter
04/07/16
b067335c
Initial commit


Working Directory
git add
Staging Area
git commit
Repository

Aufbau


Initial commit
36
Stage All
Commit
Status
Filename
A
test.txt
?
new.txt


master
Friedrich...
HE...
...
04/07/16
c65d686e
Another commit
Friedrich Graeter
04/07/16
b067335c
Initial commit

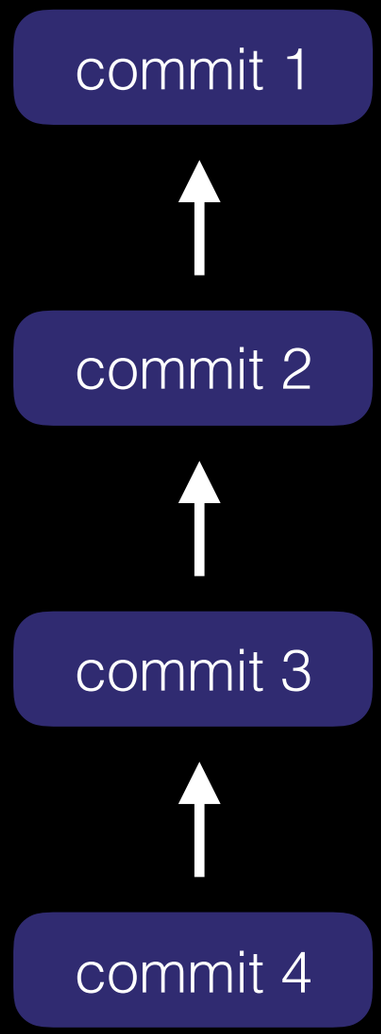

Working Directory
git add
Staging Area
git commit
Repository

Commit

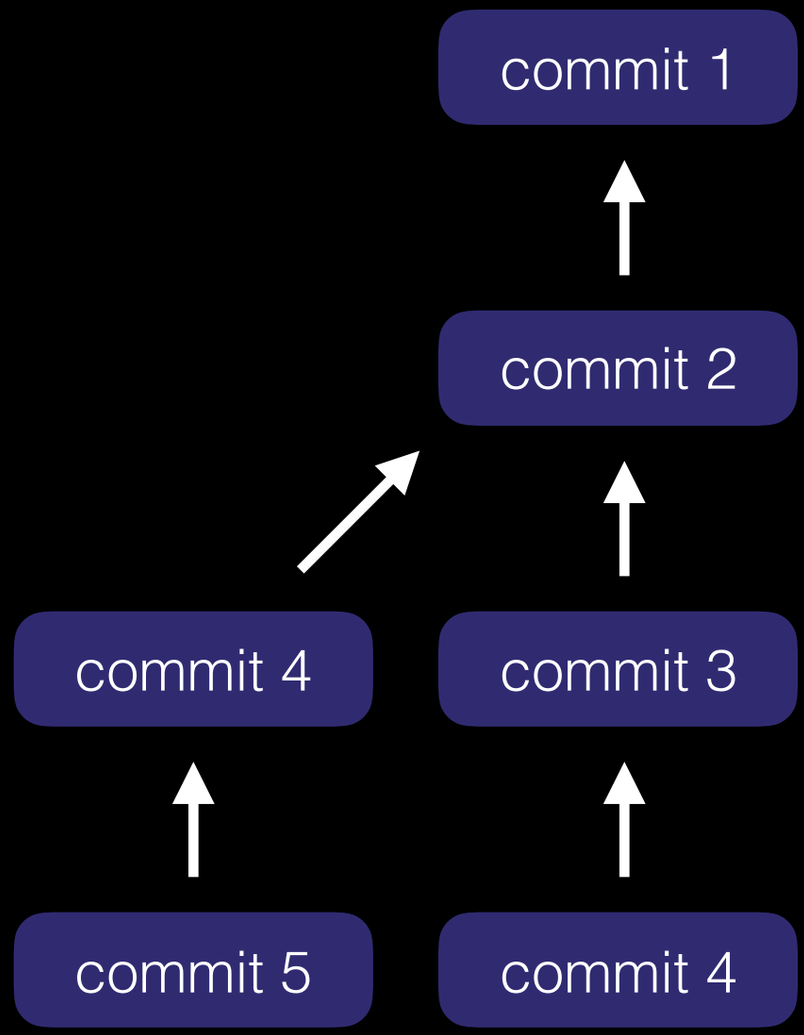
Commit

commit 1

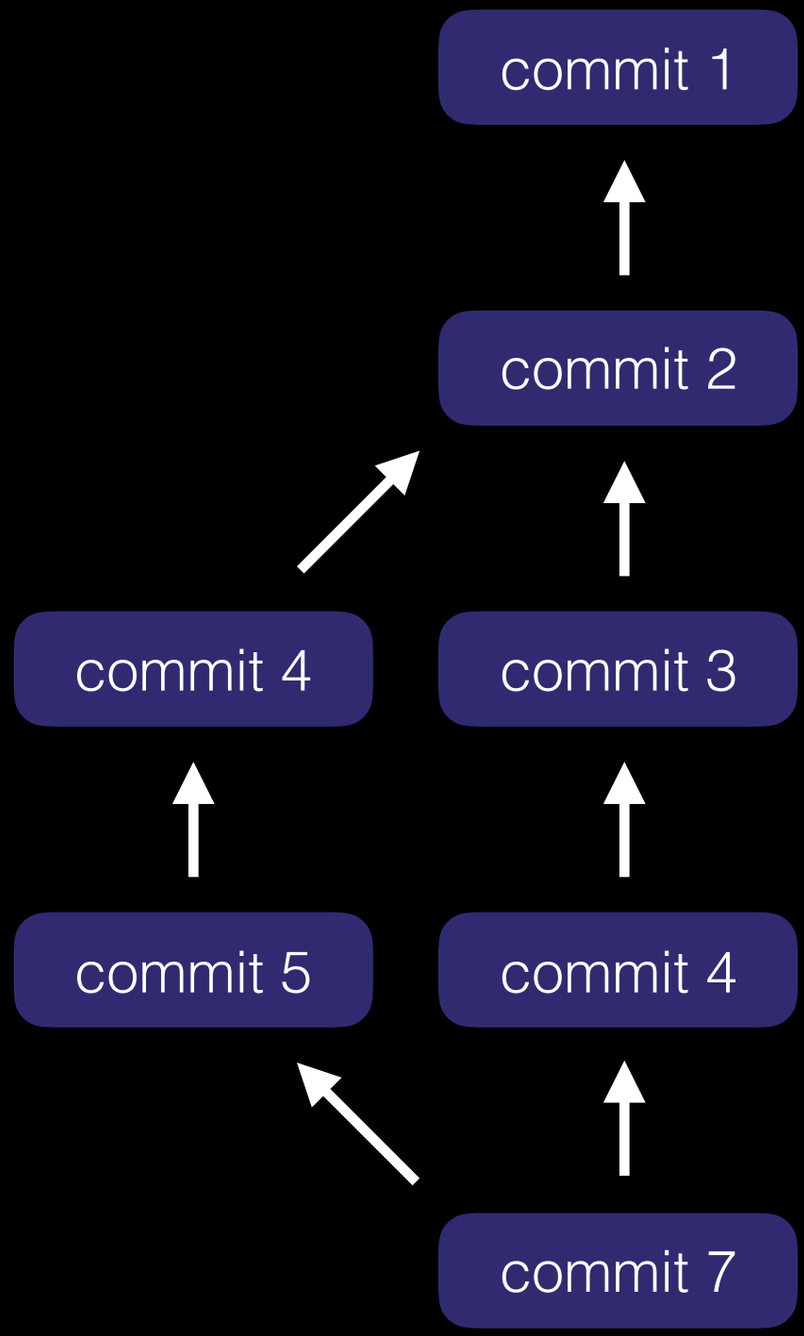
Commit



Commit

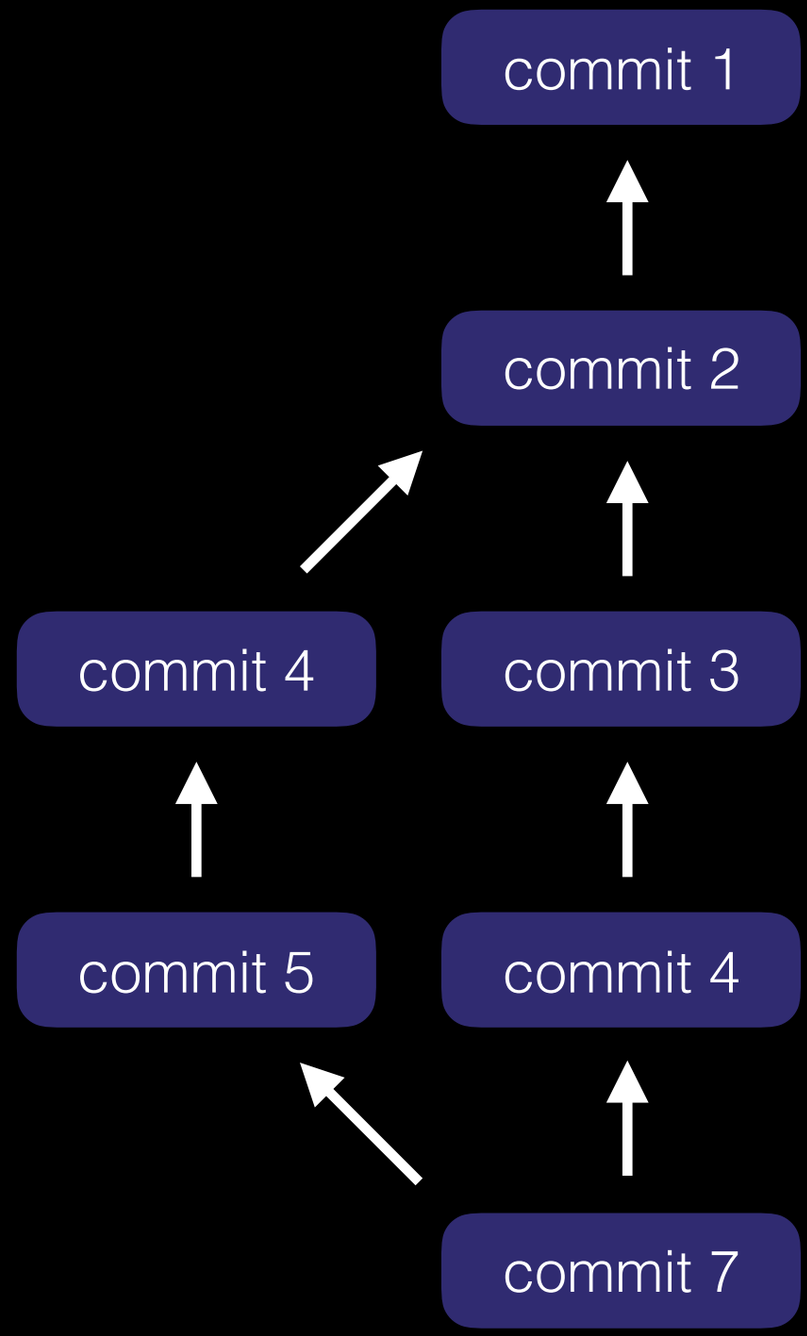


Commit



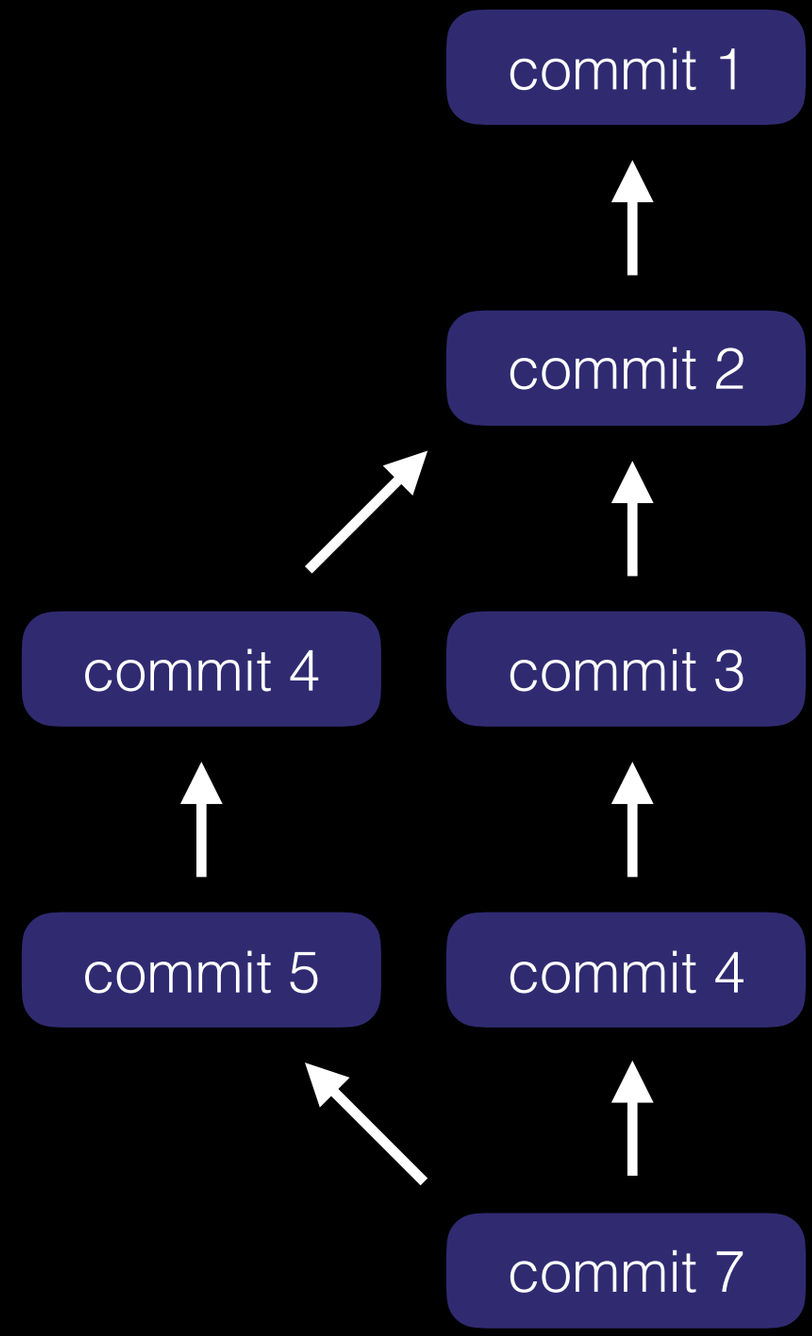
Commit

- Referenz zu Vorgängerversion(en)



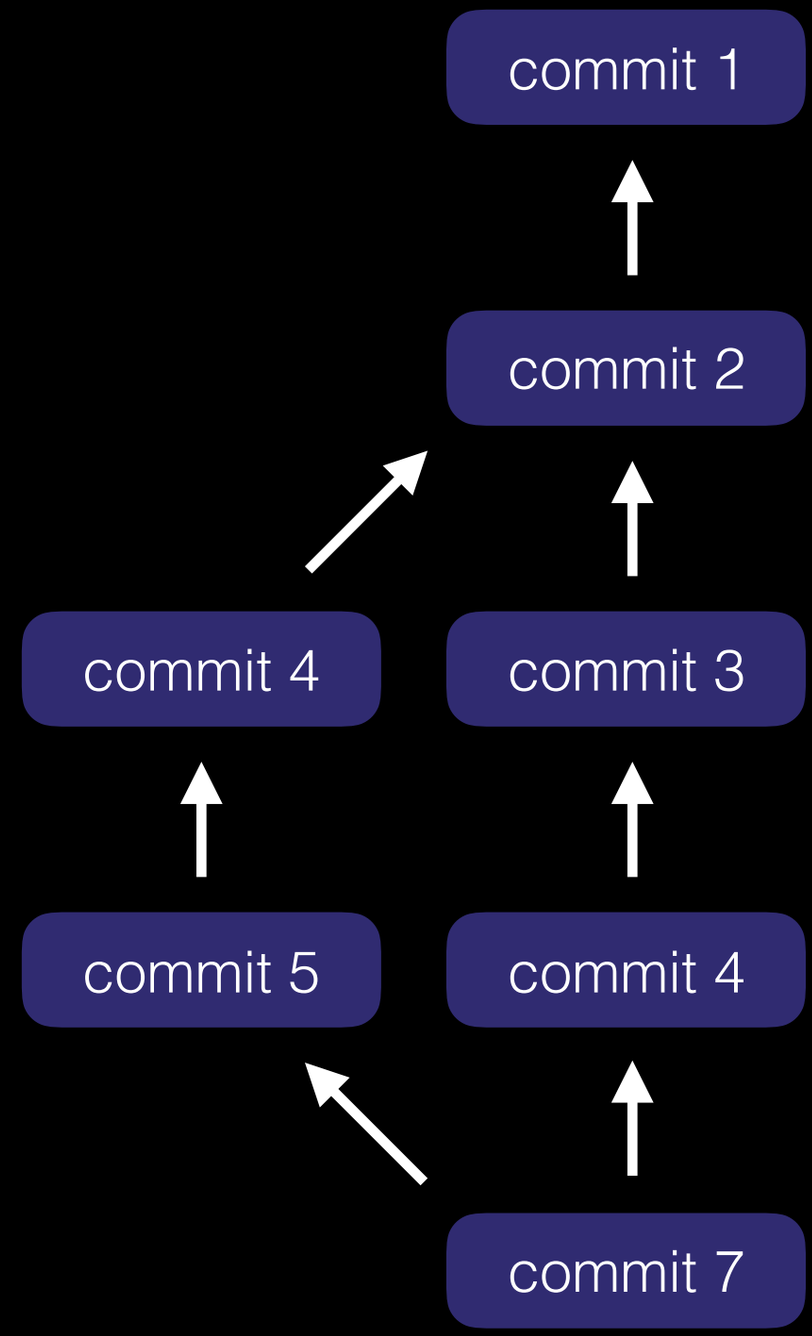
Commit

- Referenz zu Vorgängerversion(en)
- Metadaten (Autor, Kommentar, Datum, …)



Commit

- Referenz zu Vorgängerversion(en)
- Metadaten (Autor, Kommentar, Datum, …)
- Snapshot des Dateibaums (keine Diffs!)



test.txt

```
git add
git commit
```

test.txt

git add
git commit

test.txt

blob

git add
git commit

test.txt

blob

„version 1”

git add
git commit

test.txt

blob

„version 1”

git add
git commit

.git/objects/

test.txt

83baae

blob

„version 1”

git add
git commit

.git/objects/

test.txt

SHA-1("version 1") → 83baae

83baae

blob

„version 1”

git add
git commit

.git/objects/

test.txt

83baae

blob

„version 1”

git add
git commit

.git/objects/

test.txt

83baae

blob

„version 1”

git add
git commit

.git/objects/

- 83/baae… blob

test.txt

tree

83baae

blob

„version 1”

git add
git commit

.git/objects/

- 83/baae… blob

test.txt

tree

test.txt, 83baae

83baae

blob

„version 1”

git add
git commit

.git/objects/

- 83/baae… blob

test.txt

d3b7a1

tree

test.txt, 83baae

83baae

blob

„version 1”

git add
git commit

.git/objects/

- 83/baae… blob

test.txt

SHA-1("test.txt:83ba…") → d3b7a1

d3b7a1

tree

test.txt, 83baae

83baae

blob

„version 1”

git add
git commit

test.txt

.git/objects/

- 83/baae… blob

d3b7a1

tree

test.txt, 83baae

83baae

blob

„version 1”

git add
git commit

.git/objects/

- 83/baae… blob

test.txt

d3b7a1

tree

test.txt, 83baae

83baae

blob

„version 1”

```
git add
git commit
```

test.txt

```
.git/objects/

    - 83/baae… blob
    - d3/b7a1… tree
```

commit

d3b7a1

tree

test.txt, 83baae

83baae

blob

„version 1”

git add
git commit

```
.git/objects/

    - 83/baae… blob
    - d3/b7a1… tree
```

test.txt

commit

Comment

d3b7a1

tree

test.txt, 83baae

83baae

blob

„version 1”

```
git add
git commit
```

test.txt

```
.git/objects/

    - 83/baae… blob
    - d3/b7a1… tree
```

commit

Comment

d3b7a1

tree

test.txt, 83baae

83baae

blob

„version 1”

```
git add
git commit
```

test.txt

```
.git/objects/

    - 83/baae… blob
    - d3/b7a1… tree
```

fdf4fc

commit

Comment

d3b7a1

tree

test.txt, 83baae

83baae

blob

„version 1”

```
git add
git commit
```

test.txt

```
.git/objects/

    - 83/baae… blob
    - d3/b7a1… tree
```

SHA-1("Comment:d3b7a1") → fdf4c

fdf4fc

commit

Comment

d3b7a1

tree

test.txt, 83baae

83baae

blob

„version 1”

git add
git commit

test.txt

```
.git/objects/

    - 83/baae… blob
    - d3/b7a1… tree
```

fdf4fc

commit

Comment

d3b7a1

tree

test.txt, 83baae

83baae

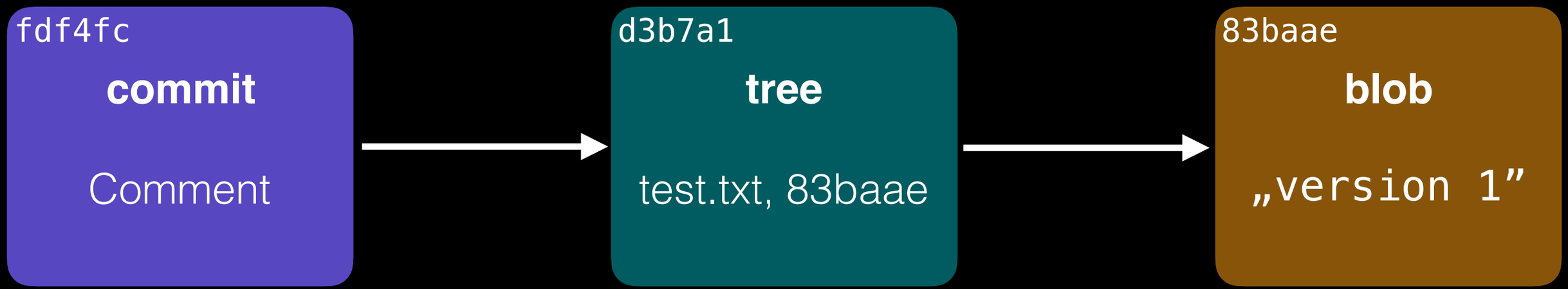
blob

„version 1”

```
git add
git commit
```

test.txt

```
.git/objects/

    - 83/baae… blob
    - d3/b7a1… tree
```

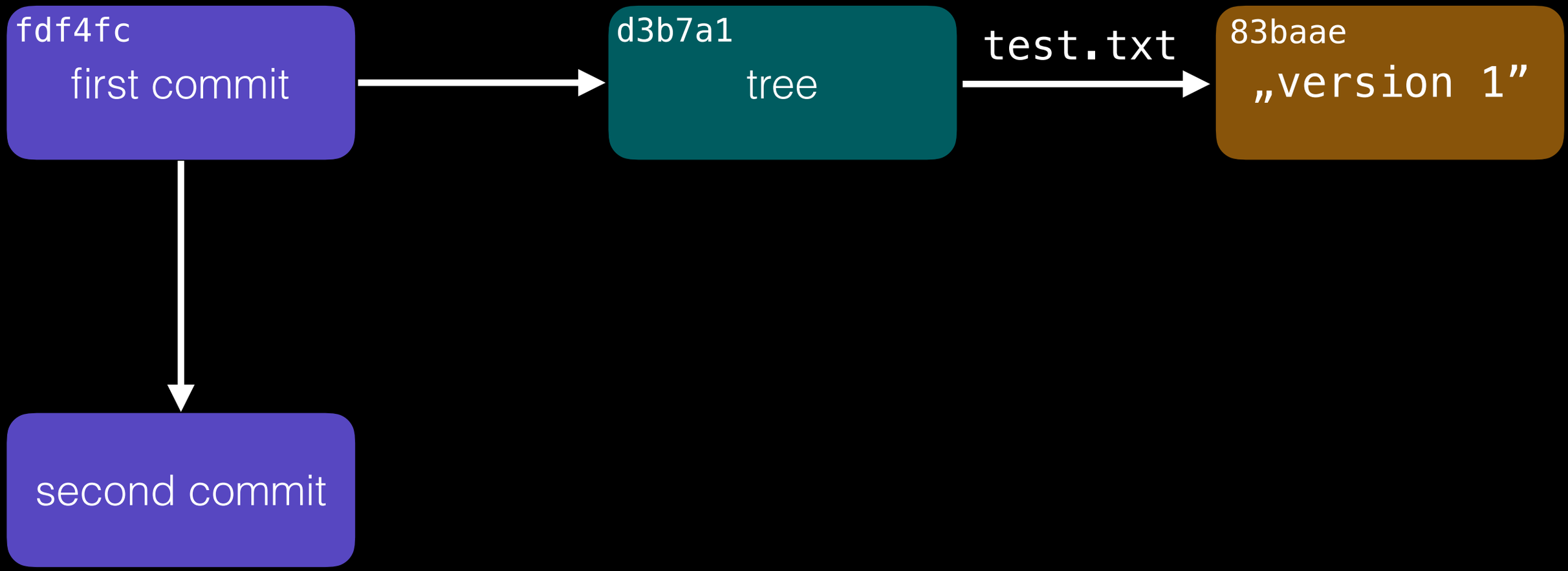


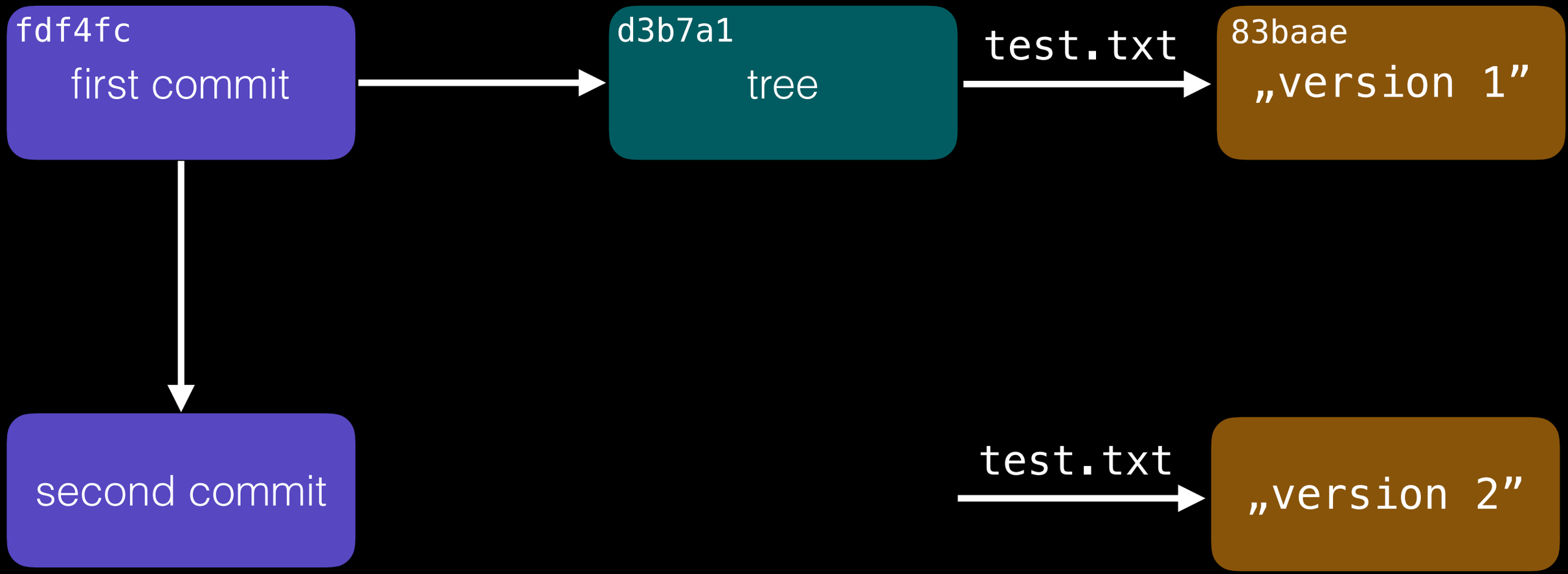

```
.git/objects/

    - 83/baae… blob
    - d3/b7a1… tree
    - fd/f4fc… commit
```

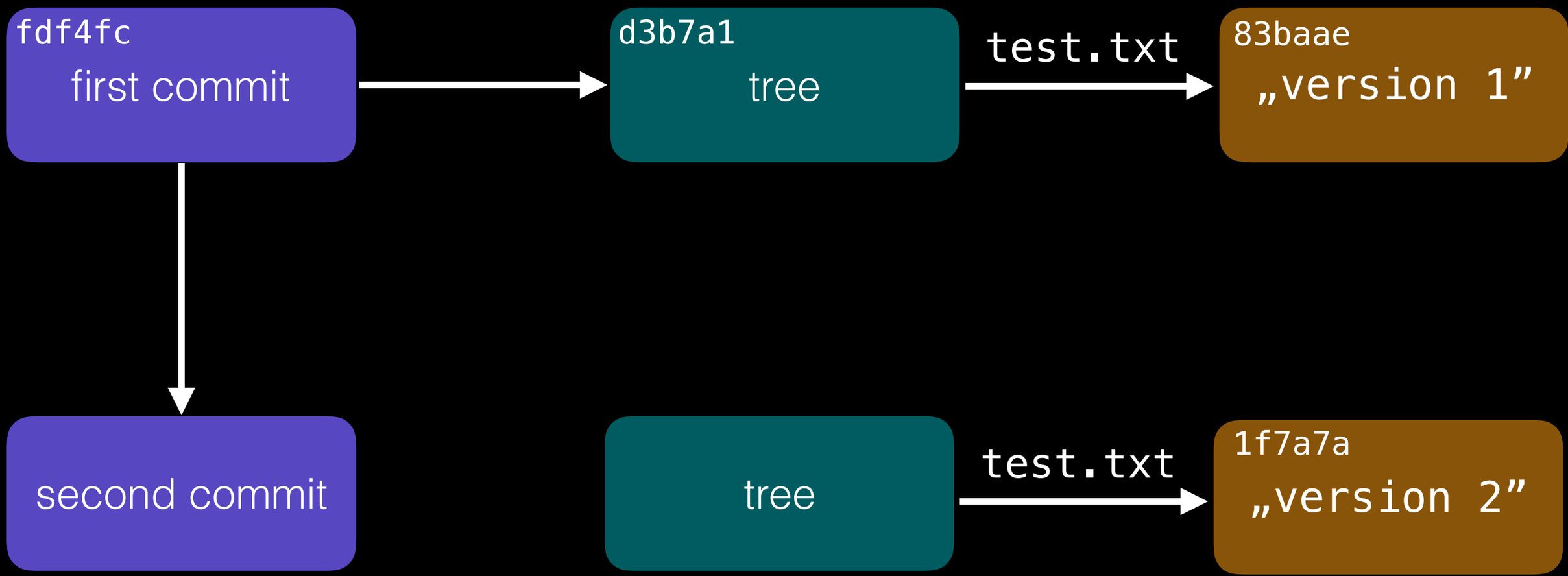
```
git add
git commit
```

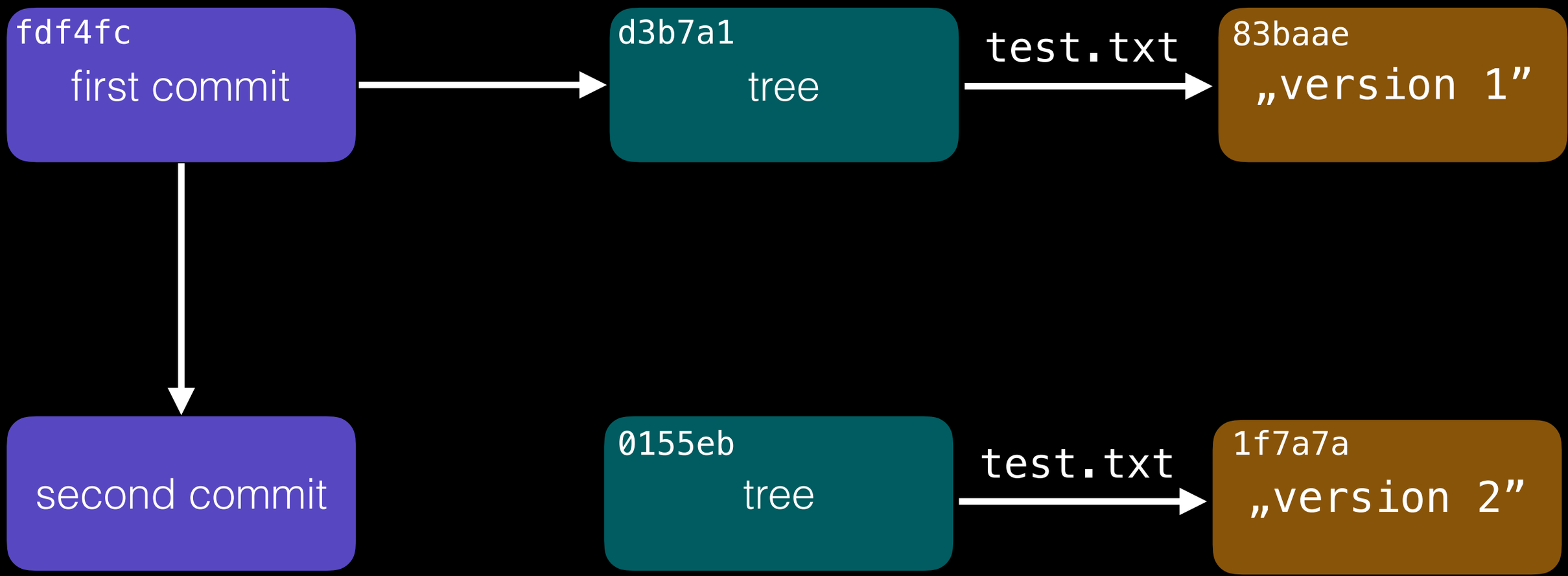
test.txt

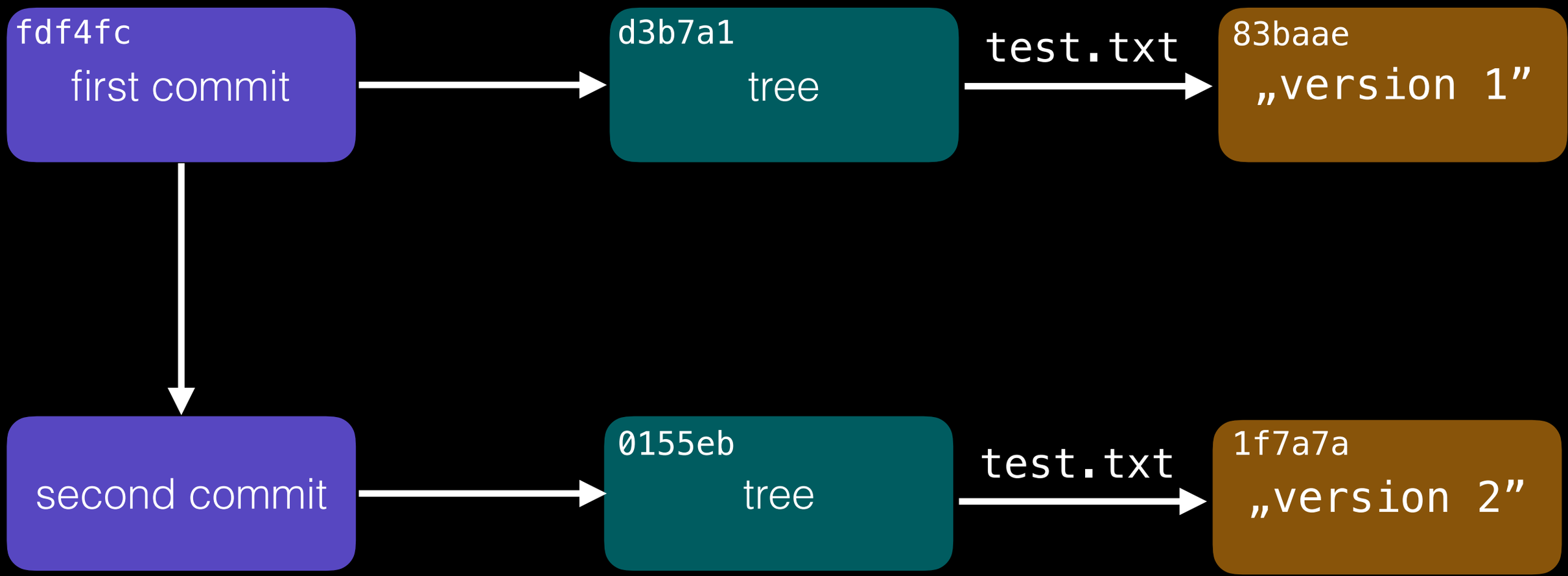

fdf4fc
first commit
d3b7a1
tree
test.txt
83baae
„version 1”

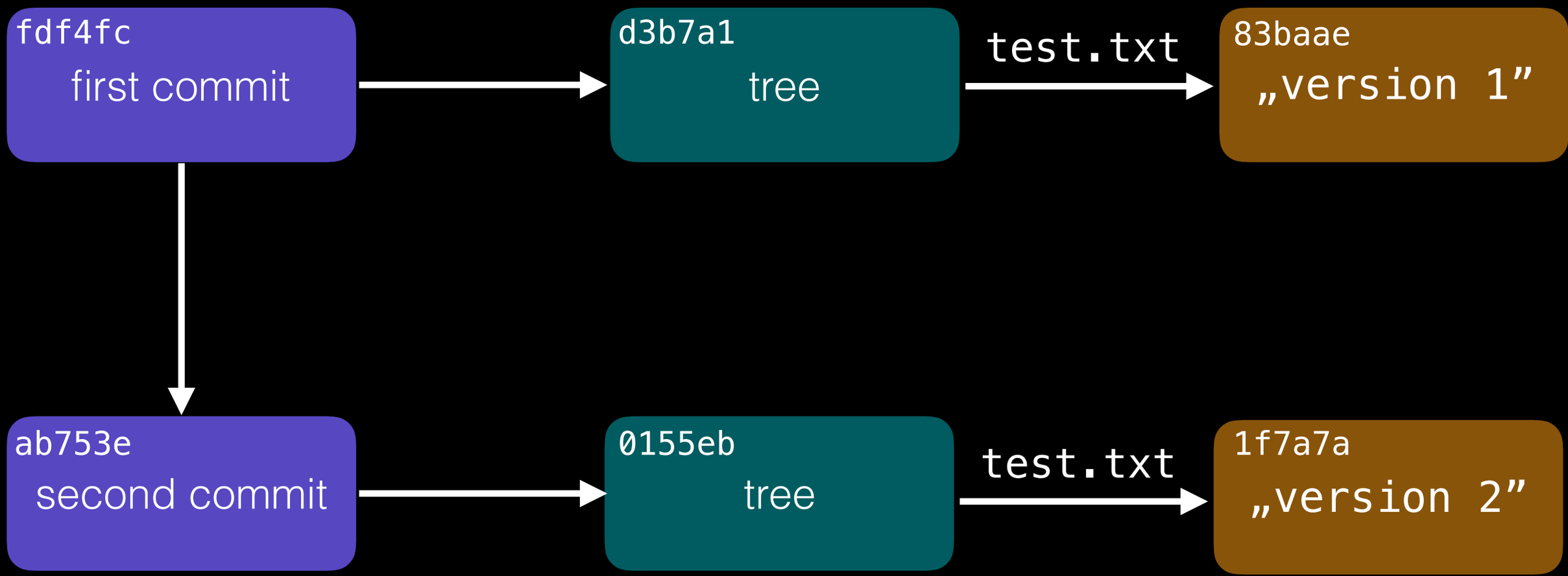

fdf4fc
first commit
d3b7a1
tree
test.txt
83baae
„version 1”
second commit

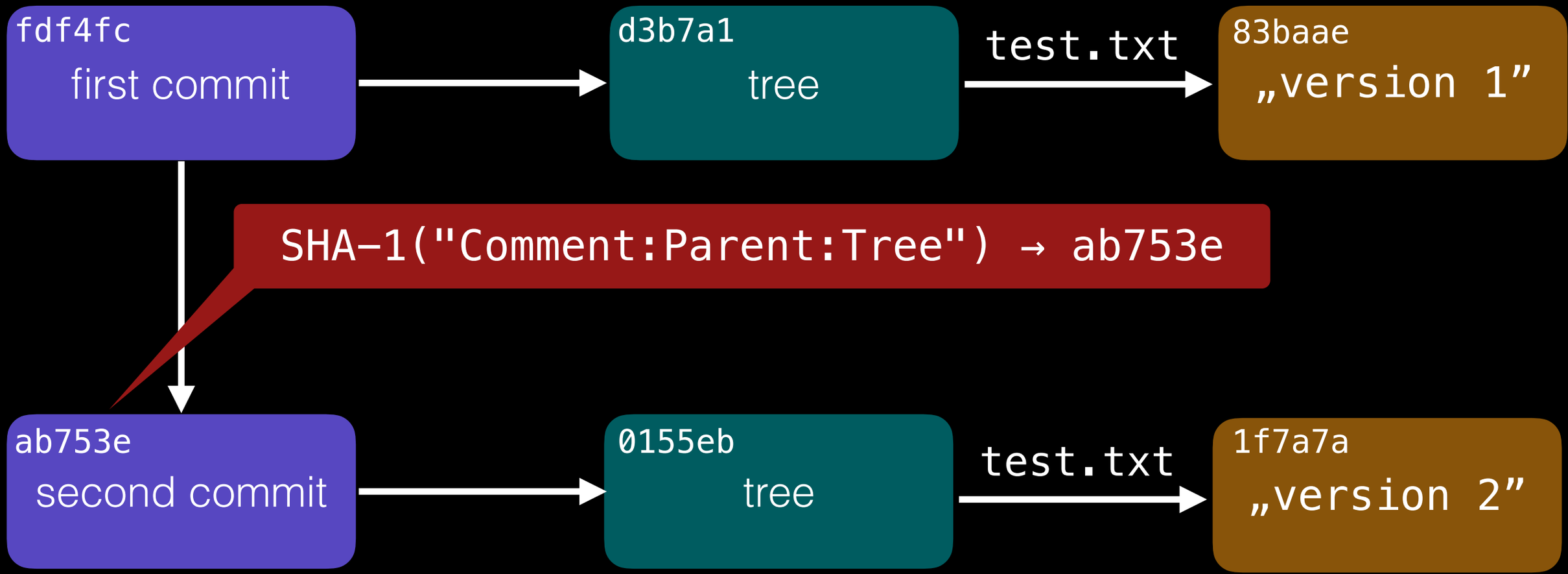

fdf4fc
first commit
d3b7a1
tree
test.txt
83baae
„version 1”
second commit
test.txt
„version 2”

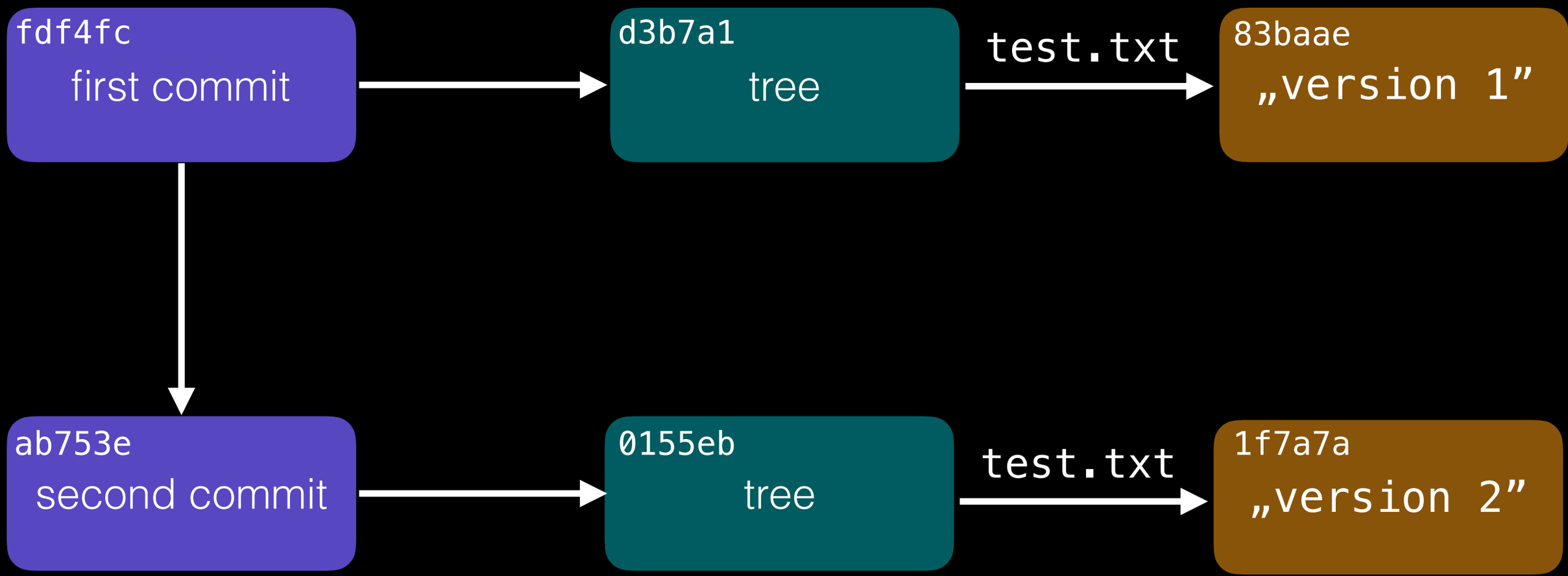

fdf4fc
first commit
d3b7a1
tree
test.txt
83baae
„version 1”
second commit
tree
test.txt
1f7a7a
„version 2”

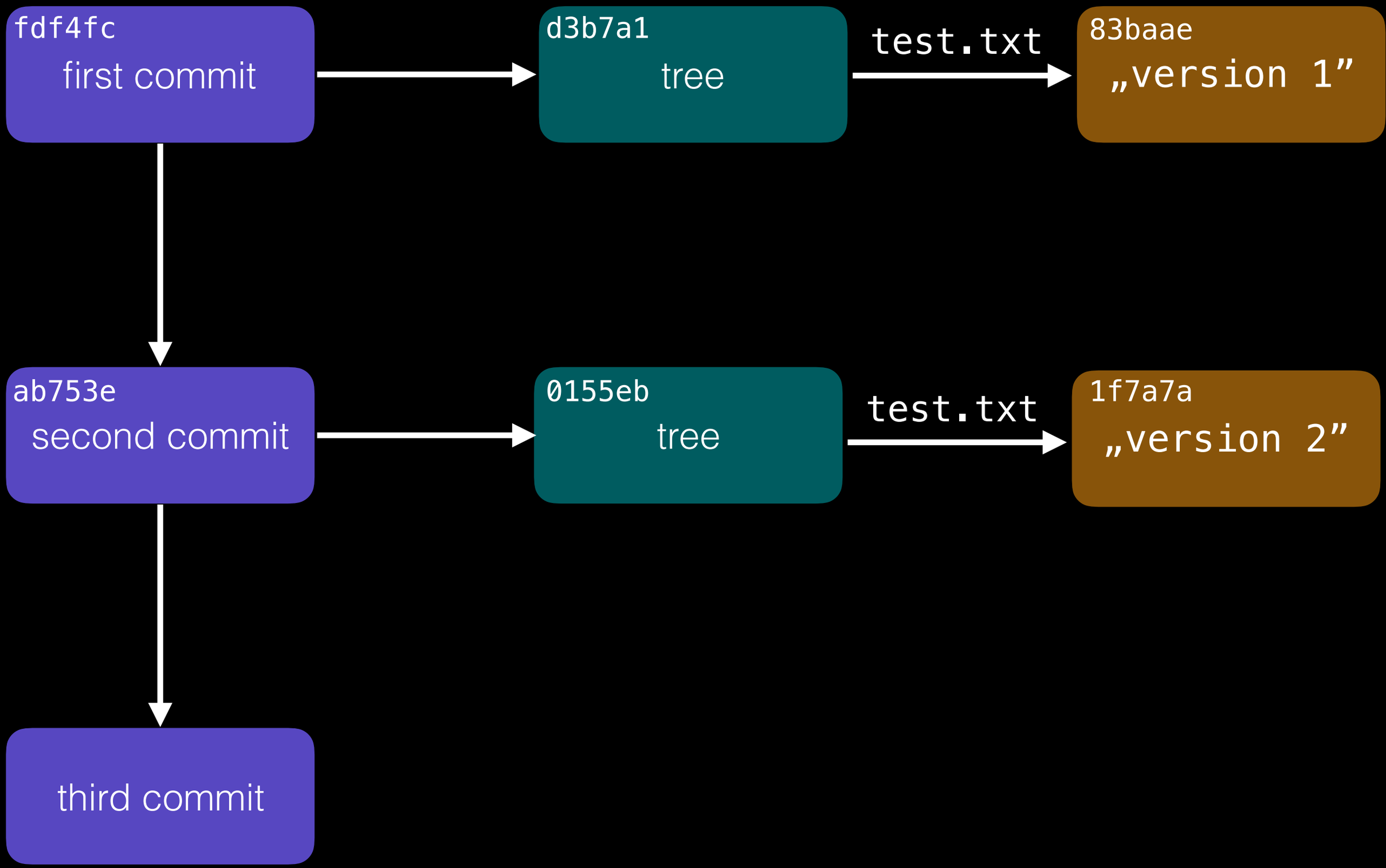

fdf4fc
first commit
d3b7a1
tree
test.txt
83baae
„version 1”
second commit
0155eb
tree
test.txt
1f7a7a
„version 2”

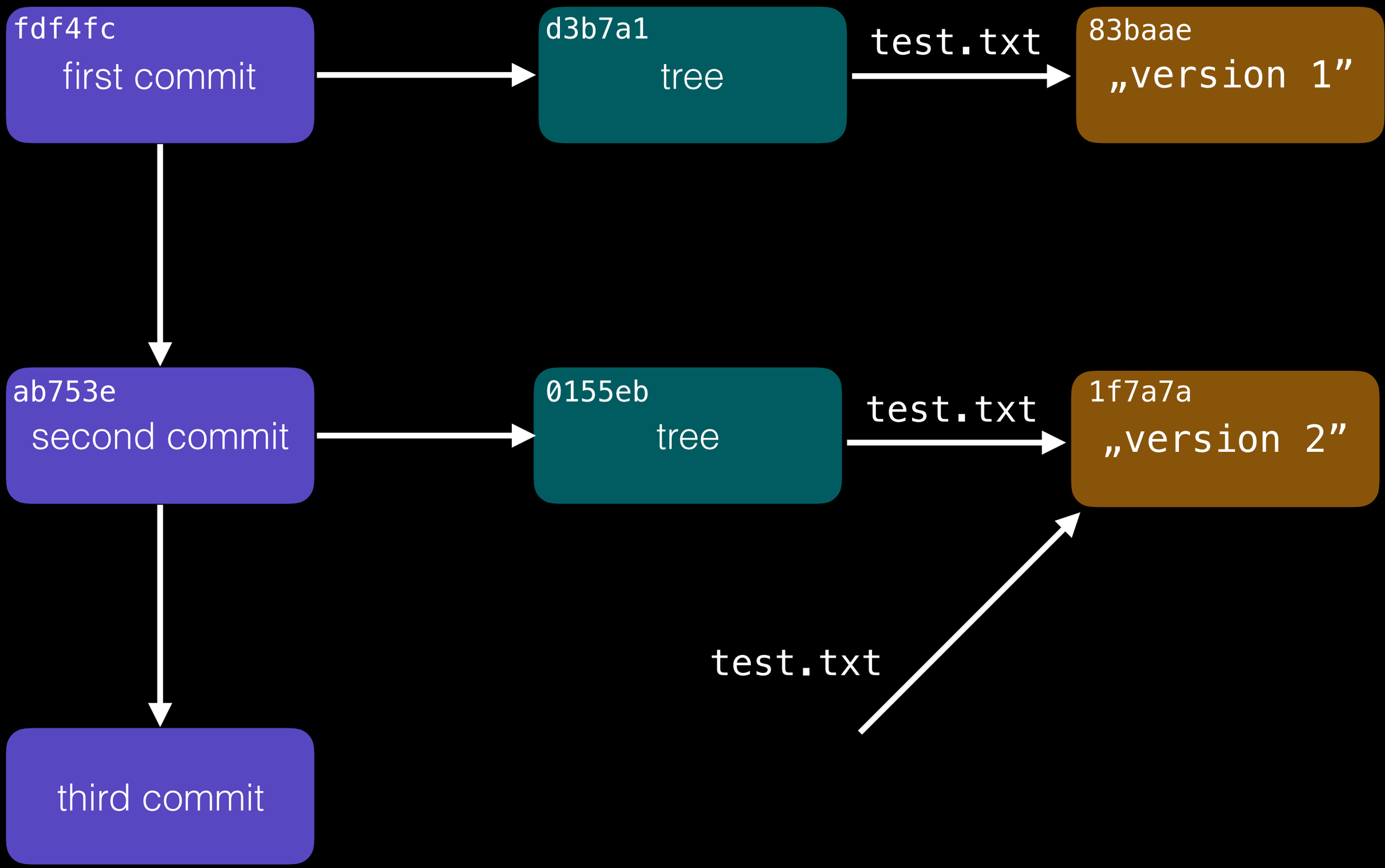

fdf4fc
first commit
d3b7a1
tree
test.txt
83baae
„version 1”
second commit
0155eb
tree
test.txt
1f7a7a
„version 2”

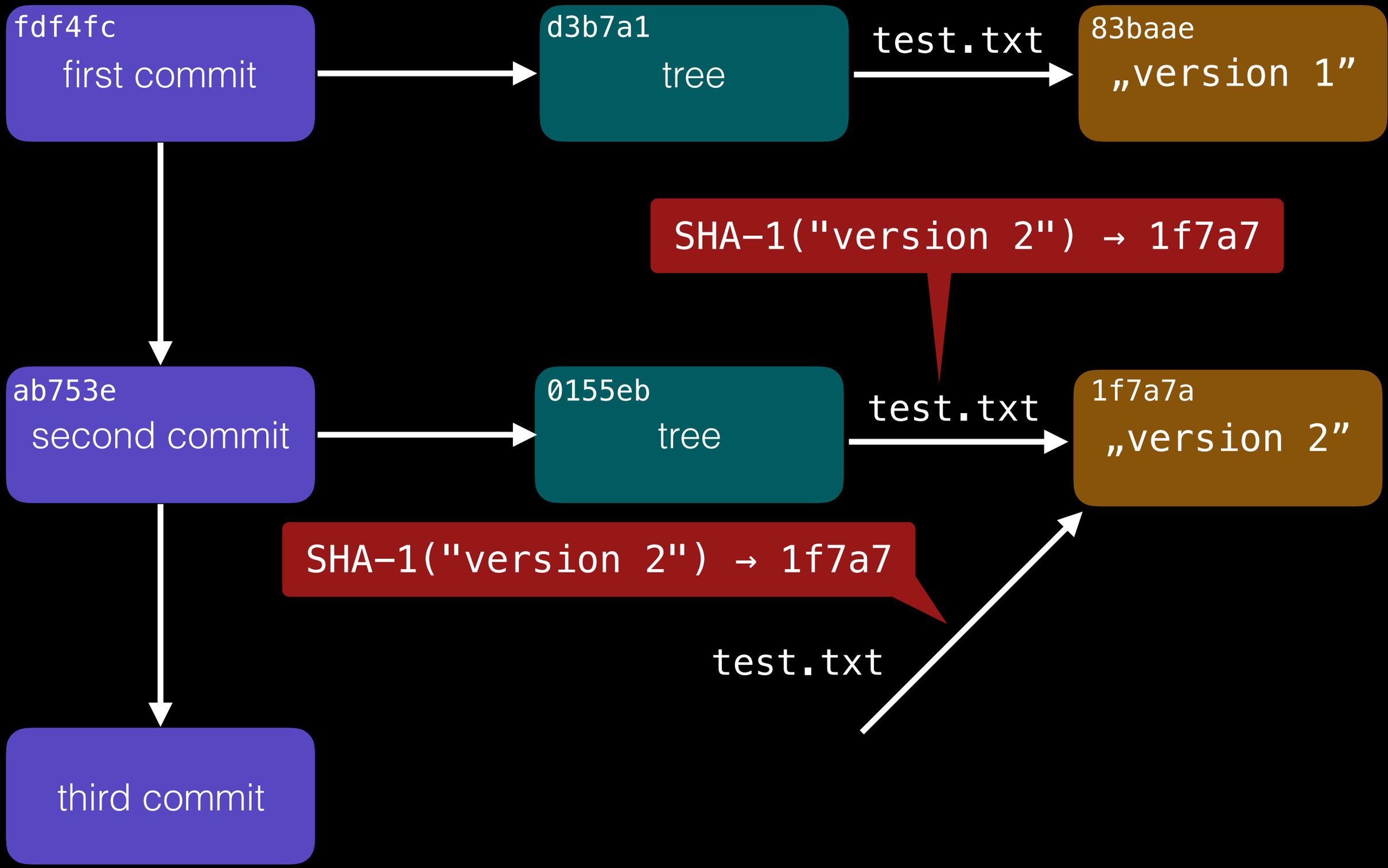

fdf4fc
first commit
d3b7a1
tree
test.txt
83baae
„version 1”
ab753e
second commit
0155eb
tree
test.txt
1f7a7a
„version 2”

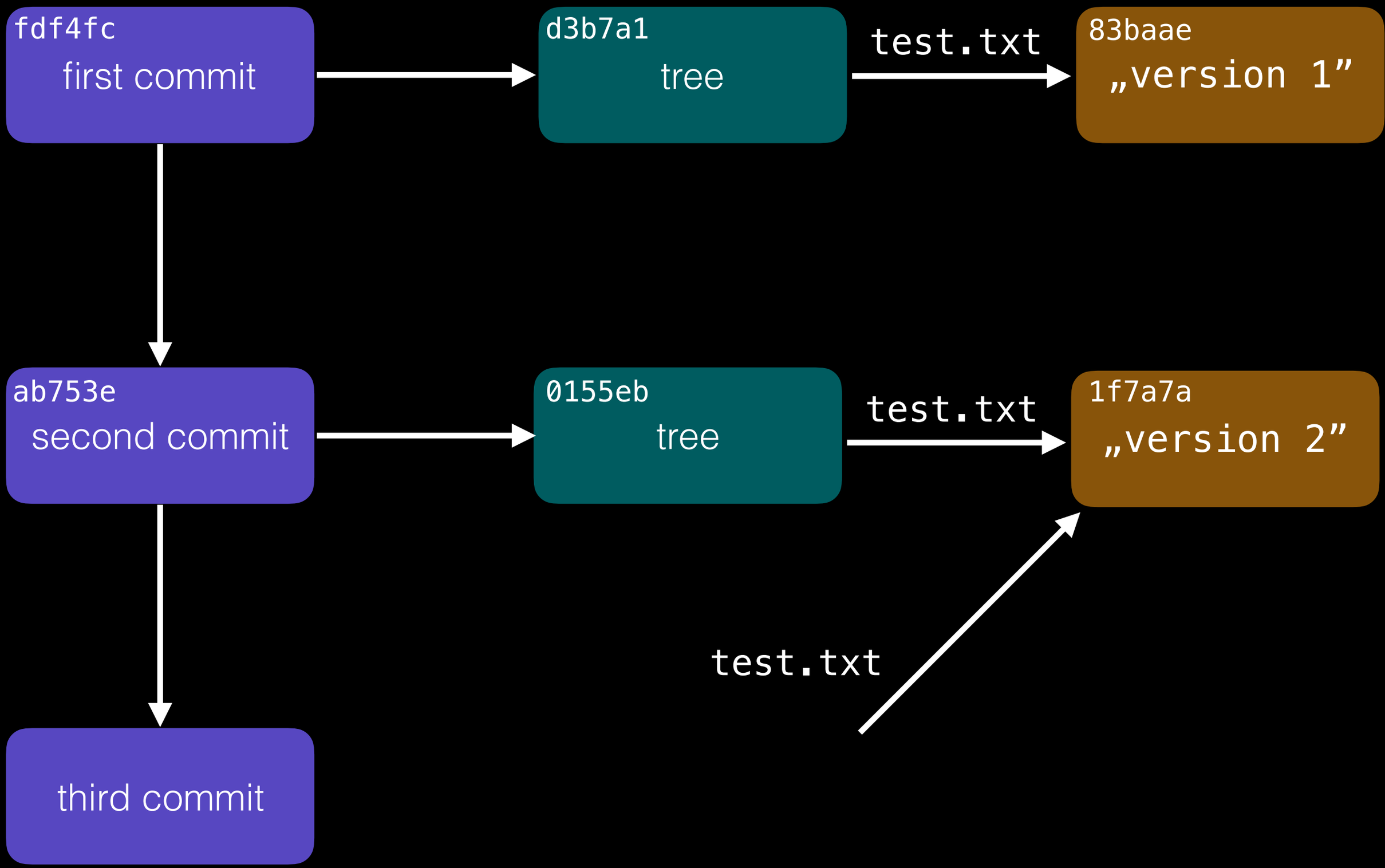

fdf4fc
first commit
d3b7a1
tree
test.txt
83baae
„version 1”
SHA-1("Comment:Parent:Tree") → ab753e
ab753e
second commit
0155eb
tree
test.txt
1f7a7a
„version 2”

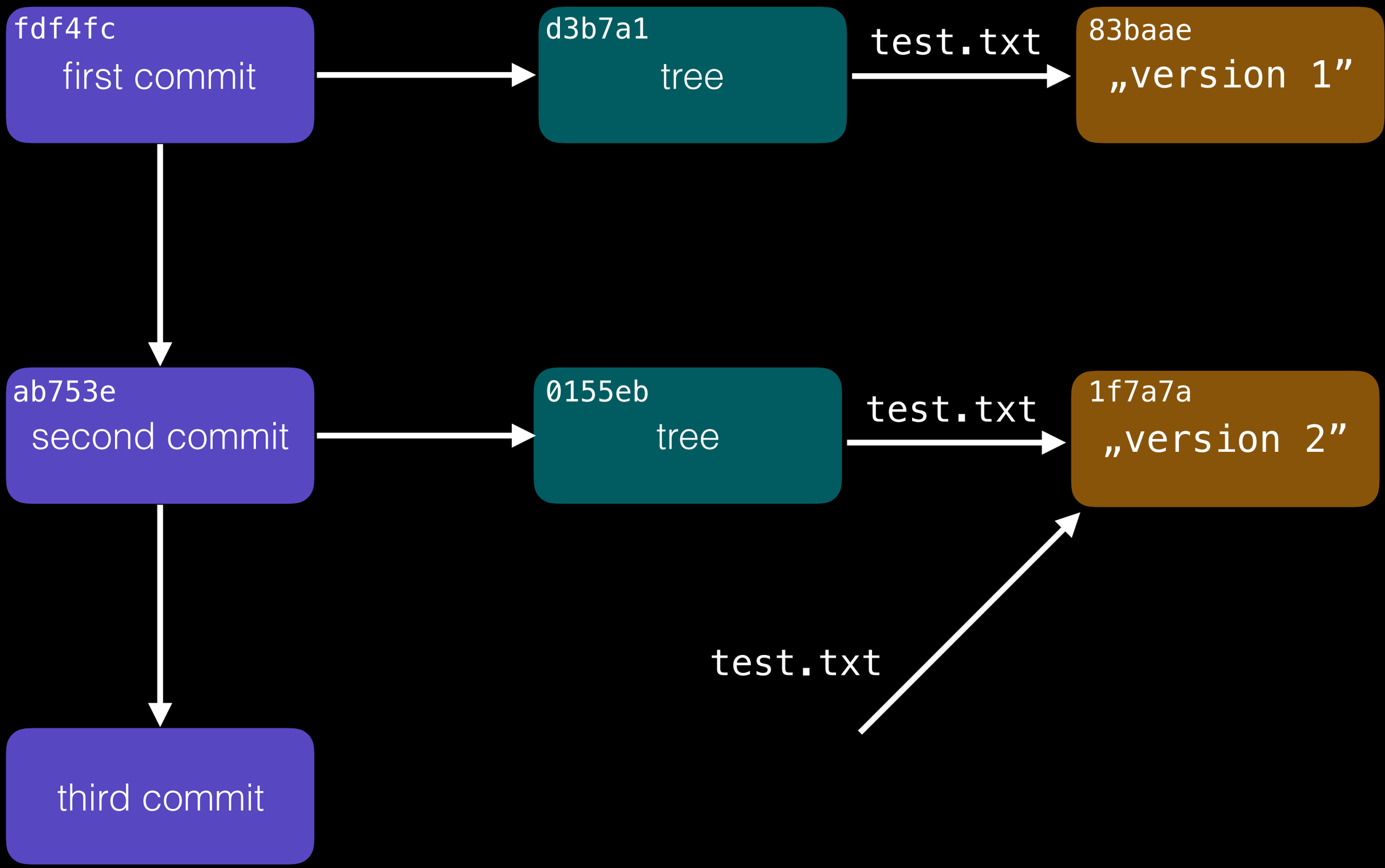

fdf4fc
first commit
d3b7a1
tree
test.txt
83baae
„version 1”
ab753e
second commit
0155eb
tree
test.txt
1f7a7a
„version 2”

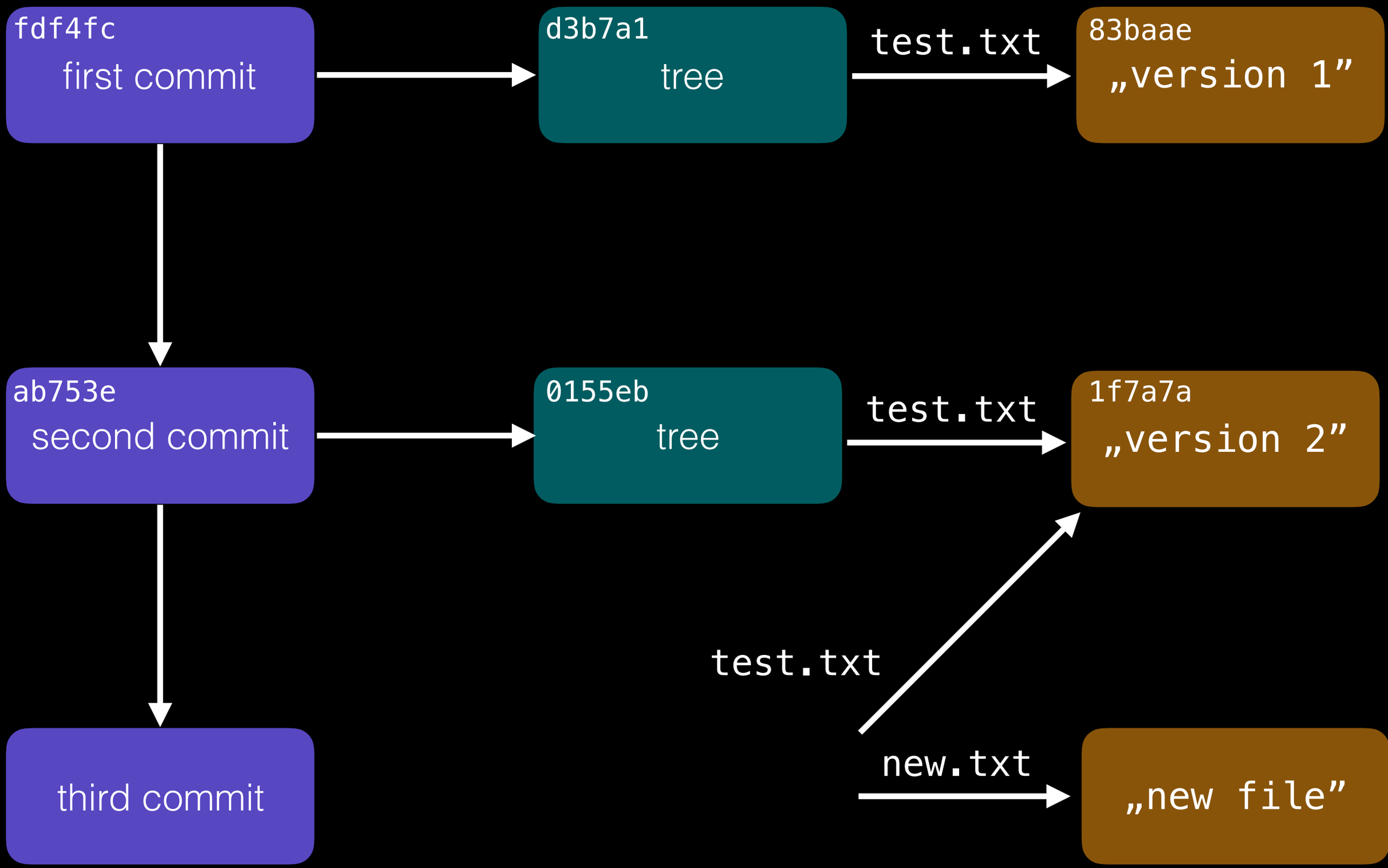

fdf4fc
first commit
d3b7a1
tree
test.txt
83baae
„version 1”
ab753e
second commit
0155eb
tree
test.txt
1f7a7a
„version 2”
third commit

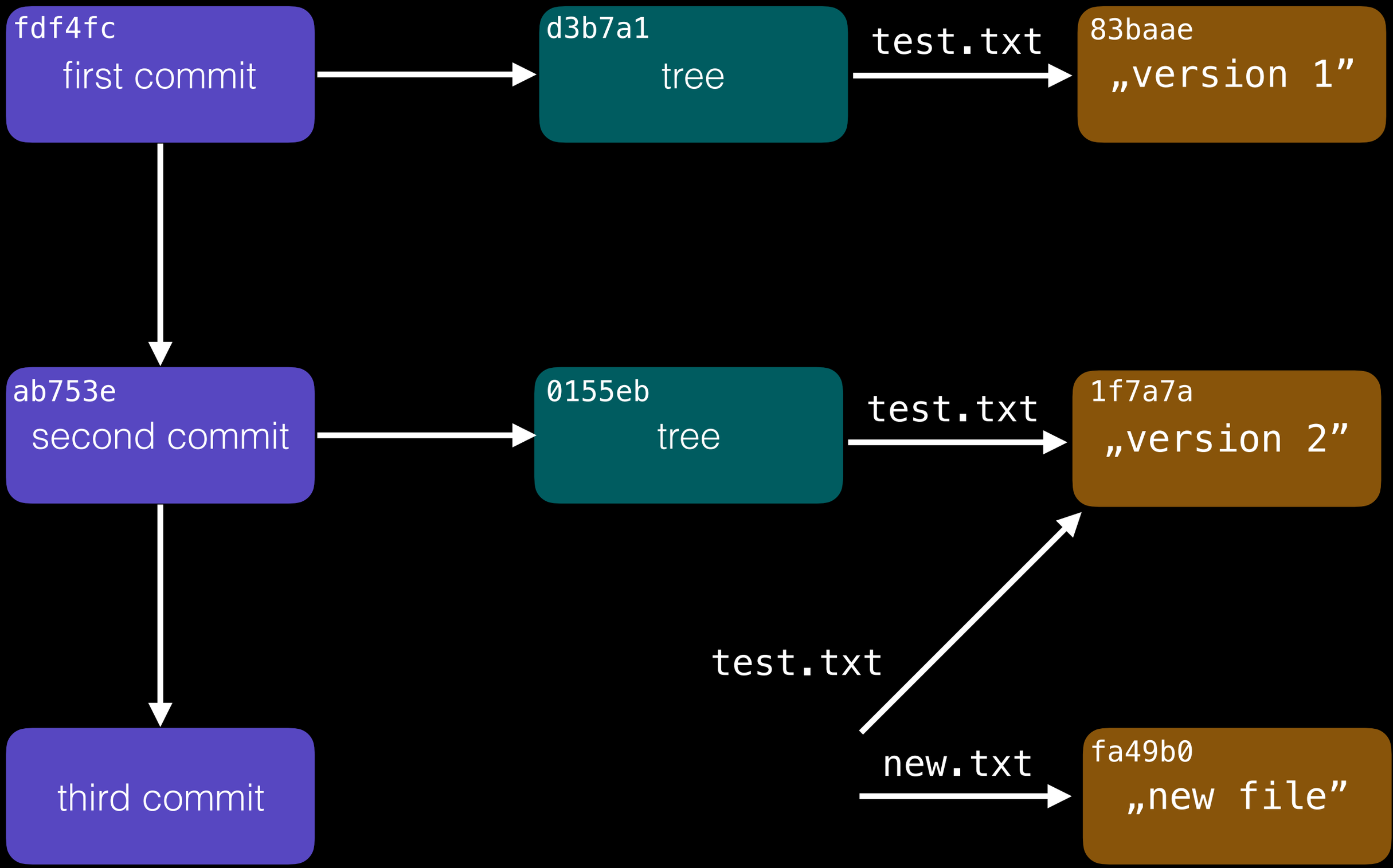

fdf4fc
first commit
d3b7a1
tree
test.txt
83baae
„version 1”
ab753e
second commit
0155eb
tree
test.txt
1f7a7a
„version 2”
test.txt
third commit

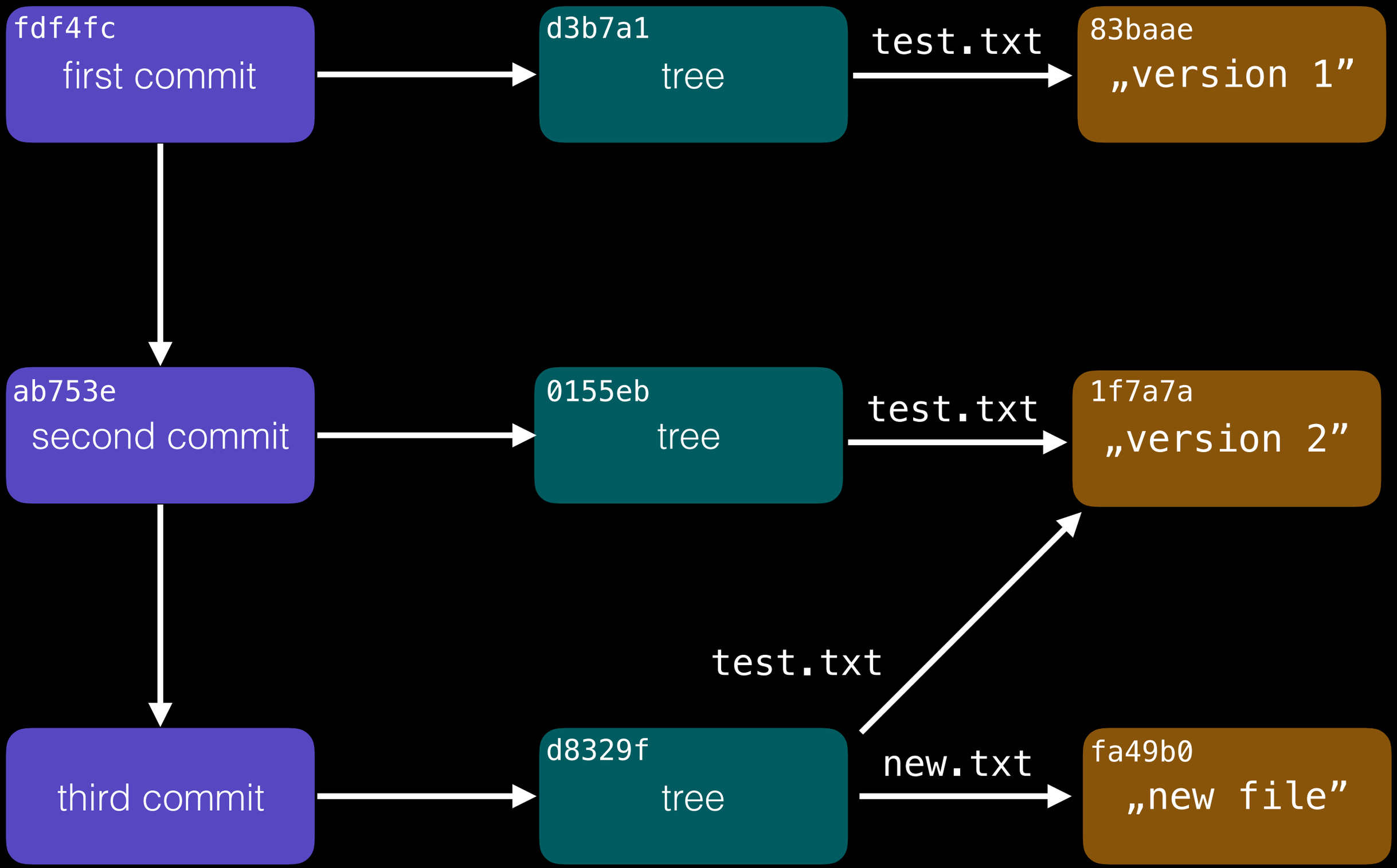

fdf4fc
first commit
d3b7a1
tree
test.txt
83baae
„version 1”
SHA-1("version 2") → 1f7a7
ab753e
second commit
0155eb
tree
test.txt
1f7a7a
„version 2”
SHA-1("version 2") → 1f7a7
test.txt
third commit

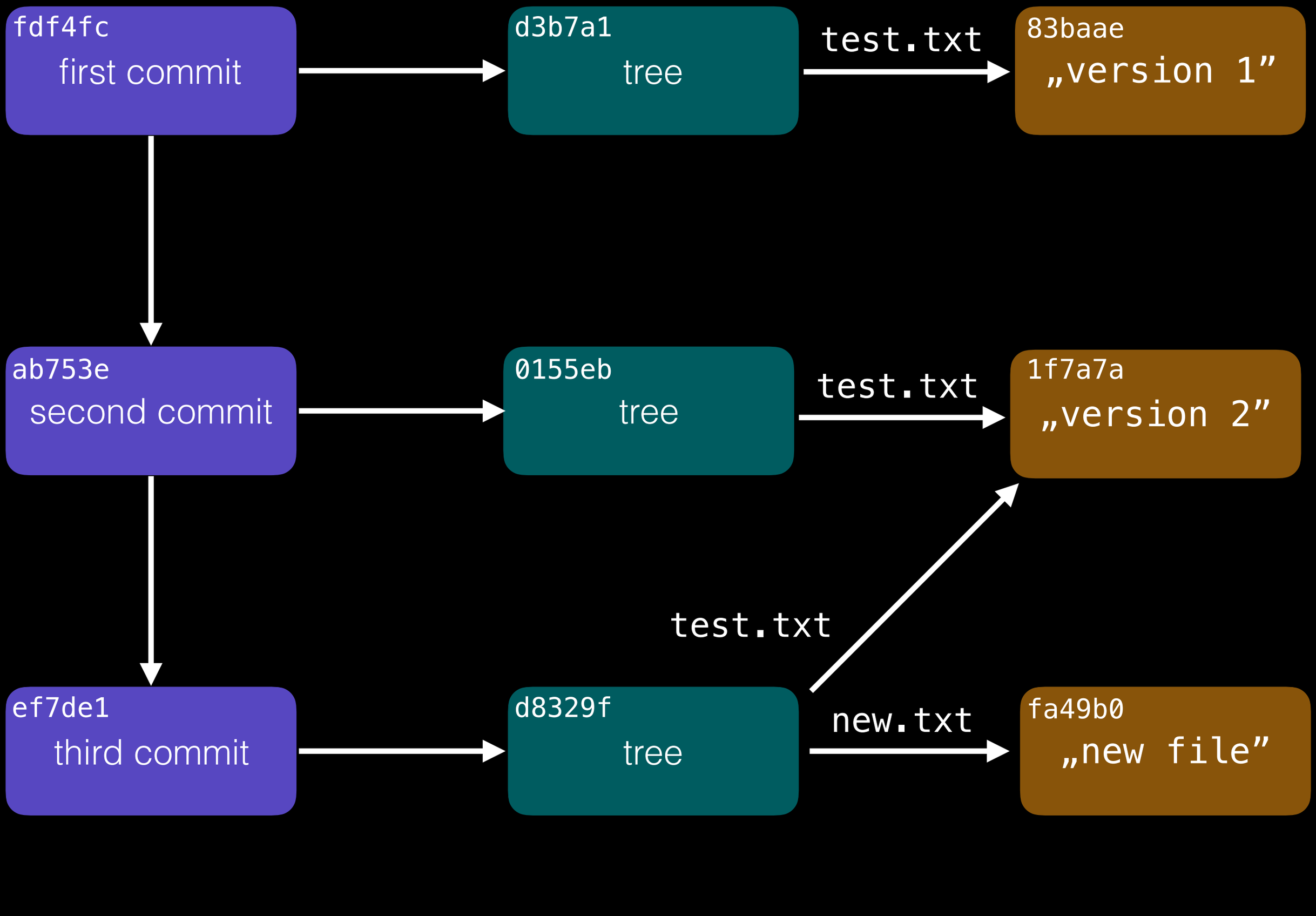

fdf4fc
first commit
d3b7a1
tree
test.txt
83baae
„version 1”
ab753e
second commit
0155eb
tree
test.txt
1f7a7a
„version 2”
test.txt
third commit


fdf4fc
first commit
d3b7a1
tree
test.txt
83baae
„version 1”
ab753e
second commit
0155eb
tree
test.txt
1f7a7a
„version 2”
test.txt
third commit


fdf4fc
first commit
d3b7a1
tree
test.txt
83baae
„version 1”
ab753e
second commit
0155eb
tree
test.txt
1f7a7a
„version 2”
test.txt
third commit
new.txt
„new file”

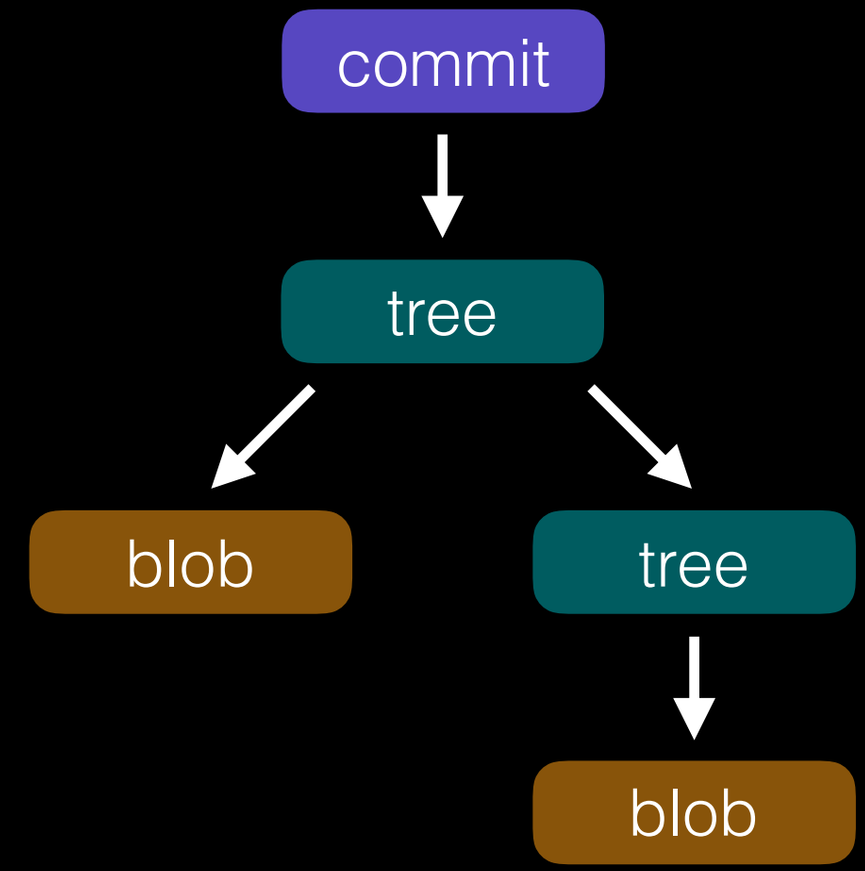

fdf4fc
first commit
d3b7a1
tree
test.txt
83baae
„version 1”
ab753e
second commit
0155eb
tree
test.txt
1f7a7a
„version 2”
test.txt
third commit
new.txt
fa49b0
„new file”


fdf4fc
first commit
d3b7a1
tree
test.txt
83baae
„version 1”
ab753e
second commit
0155eb
tree
test.txt
1f7a7a
„version 2”
test.txt
third commit
d8329f
tree
new.txt
fa49b0
„new file”


fdf4fc
first commit
d3b7a1
tree
test.txt
83baae
„version 1”
ab753e
second commit
0155eb
tree
test.txt
1f7a7a
„version 2”
ef7de1
third commit
d8329f
tree
test.txt
new.txt
fa49b0
„new file”

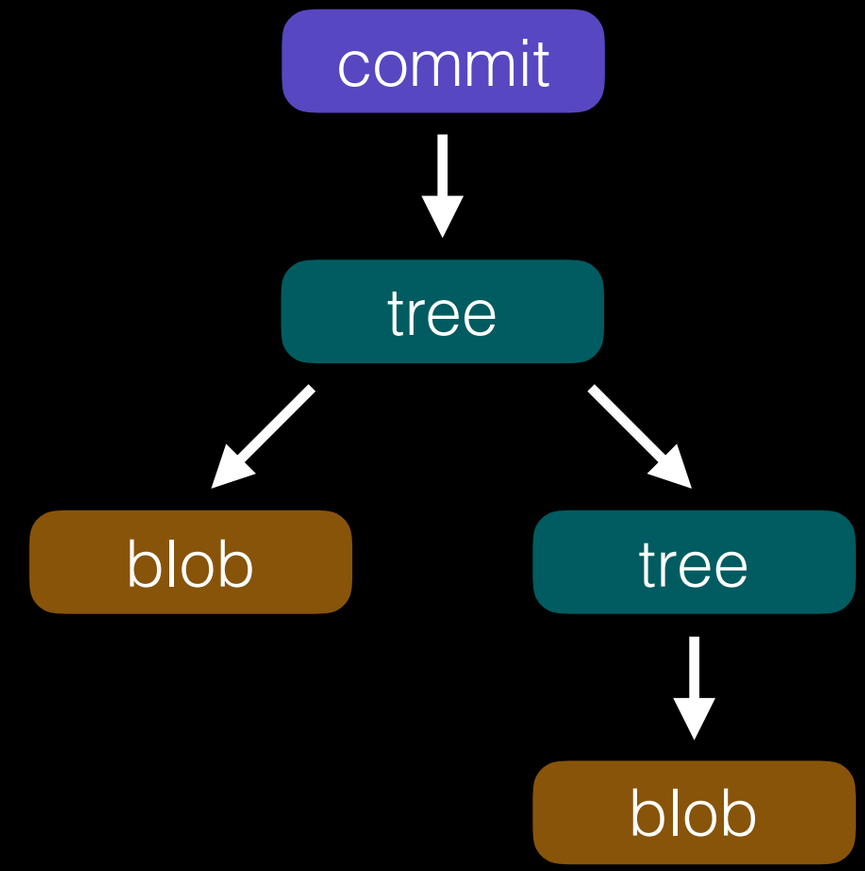
Objekte und Referenzen

Objekte und Referenzen



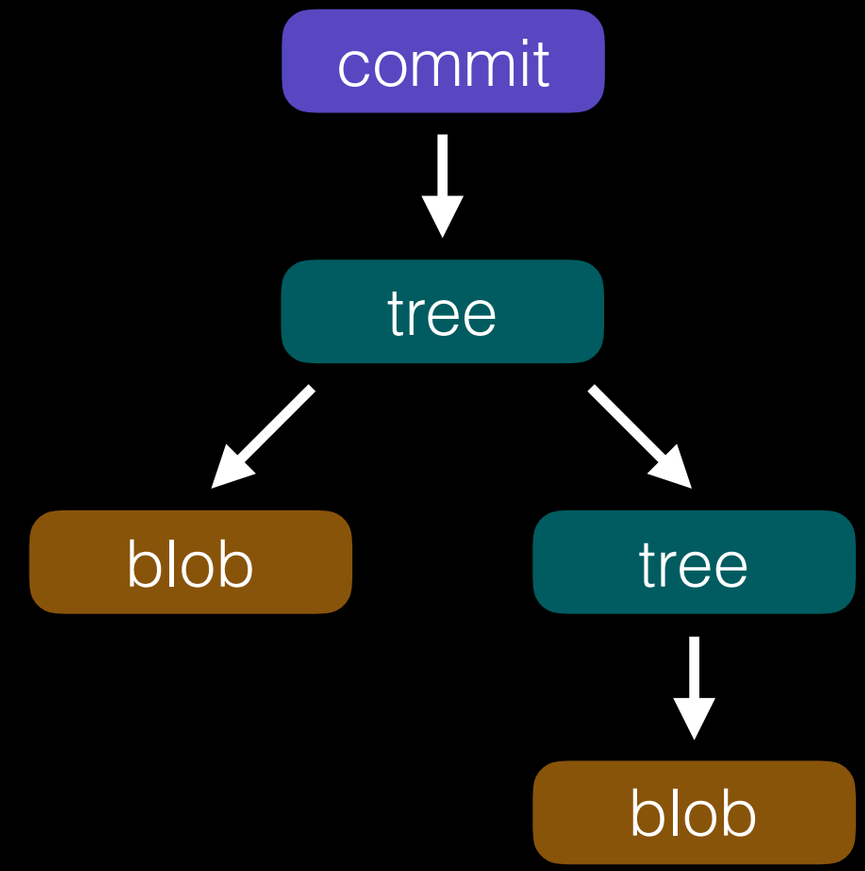
Objekte und Referenzen

- **Objekte:**
  Commits, Trees, BLOBs



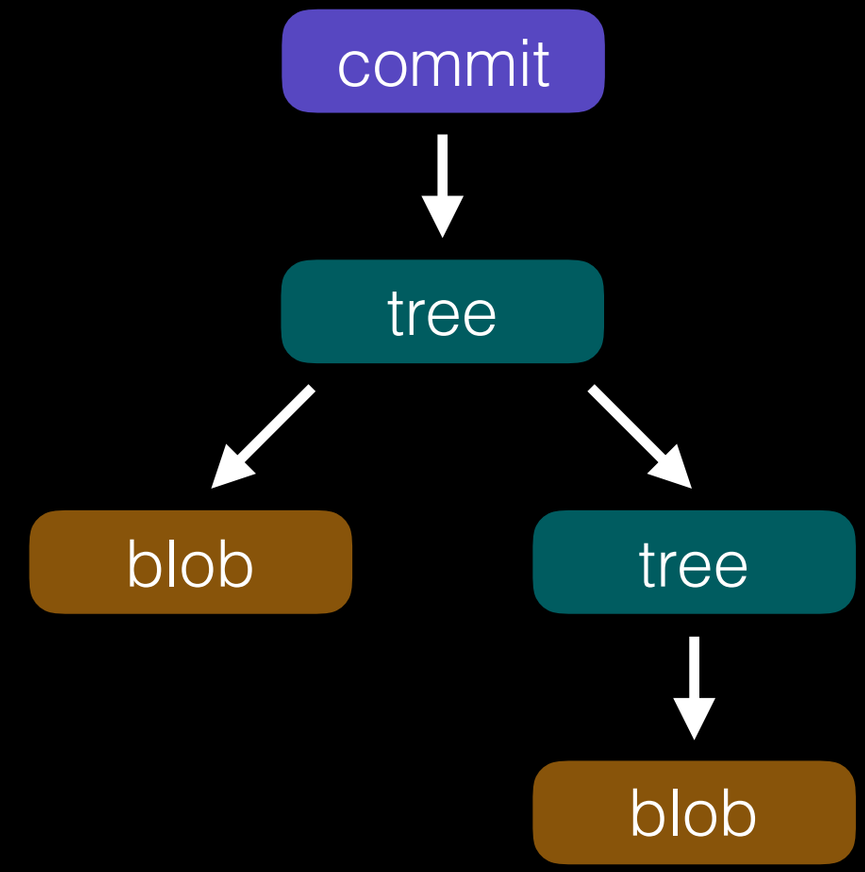
Objekte und Referenzen

- **Objekte:**
  Commits, Trees, BLOBs
- `.git/objects`



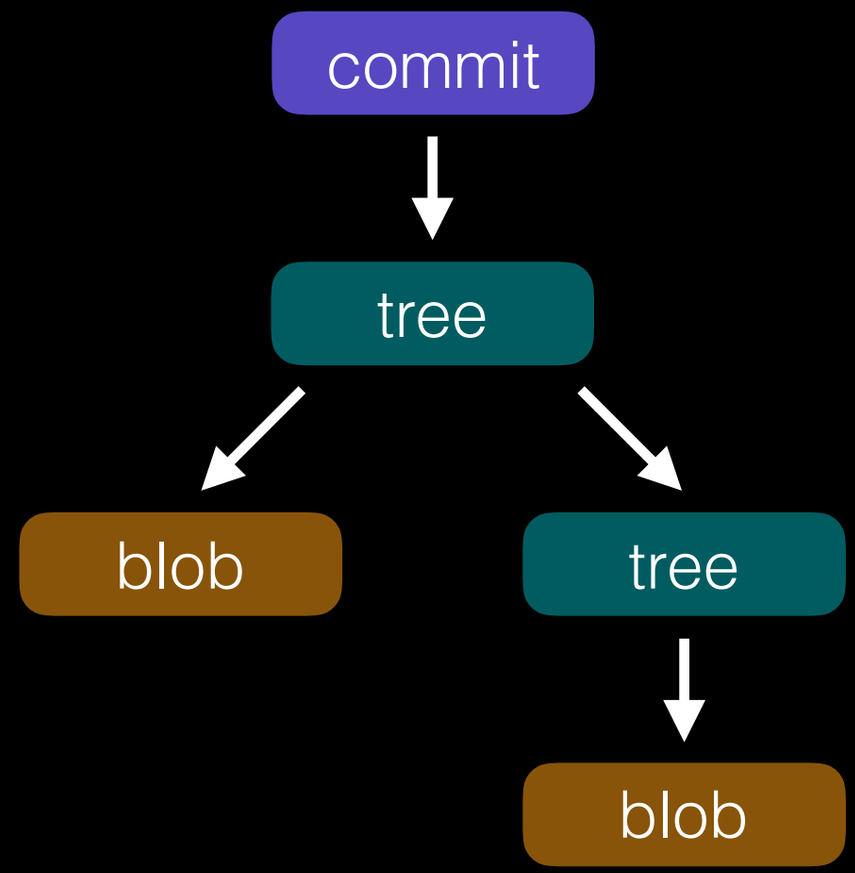
Objekte und Referenzen

- **Objekte:**
  Commits, Trees, BLOBs
- `.git/objects`
- **Referenzen:**
  SHA-1 Hash des Inhalts



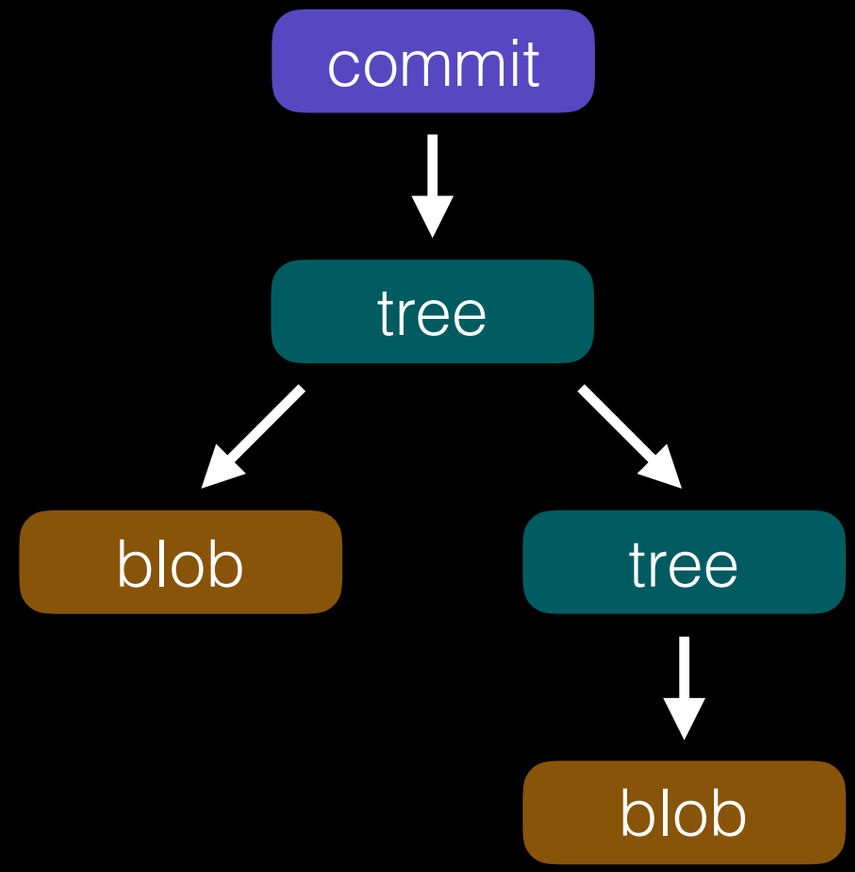
Objekte und Referenzen

- **Objekte:**
  Commits, Trees, BLOBs
- `.git/objects`
- **Referenzen:**
  SHA-1 Hash des Inhalts
- **Vorteil von Hashes:**



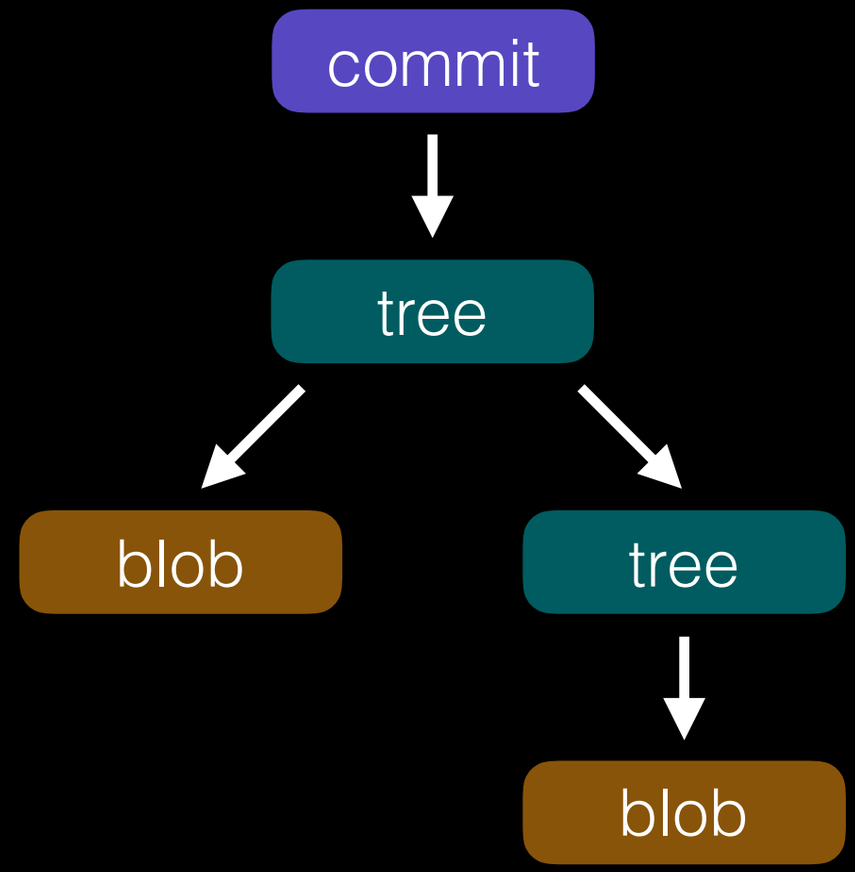
Objekte und Referenzen

- **Objekte:**
  Commits, Trees, BLOBs
- `.git/objects`
- **Referenzen:**
  SHA-1 Hash des Inhalts
- **Vorteil von Hashes:**
  - Duplikate finden



Objekte und Referenzen

- **Objekte:**
  Commits, Trees, BLOBs
- `.git/objects`
- **Referenzen:**
  SHA-1 Hash des Inhalts
- **Vorteil von Hashes:**
  - Duplikate finden
  - Integrität sicherstellen



Optimierungen

Optimierungen

- Einzelne Objekte: Zip-Kompression

Optimierungen

- Einzelne Objekte: Zip-Kompression
- Packfiles:

Optimierungen

- Einzelne Objekte: Zip-Kompression
- Packfiles:
  - Bei Bedarf (> 7000 Objekte)

Optimierungen

- Einzelne Objekte: Zip-Kompression
- Packfiles:
  - Bei Bedarf (> 7000 Objekte)
  - Neuste Version

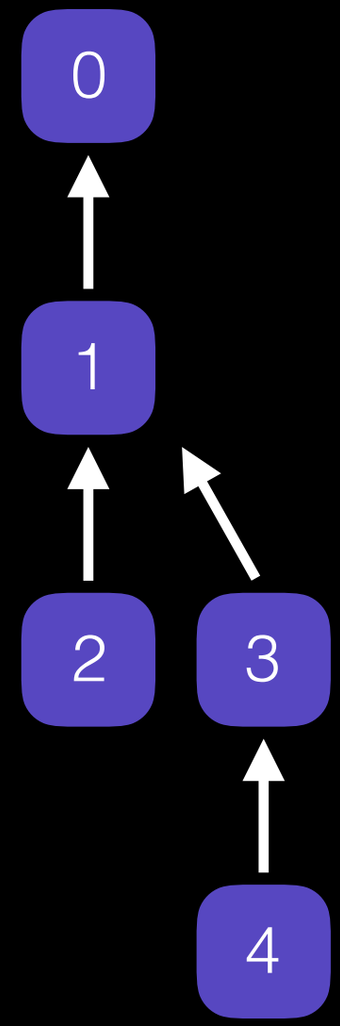
Optimierungen

- Einzelne Objekte: Zip-Kompression
- Packfiles:
  - Bei Bedarf (> 7000 Objekte)
  - Neuste Version
  - Deltas von neueren zu älteren Versionen

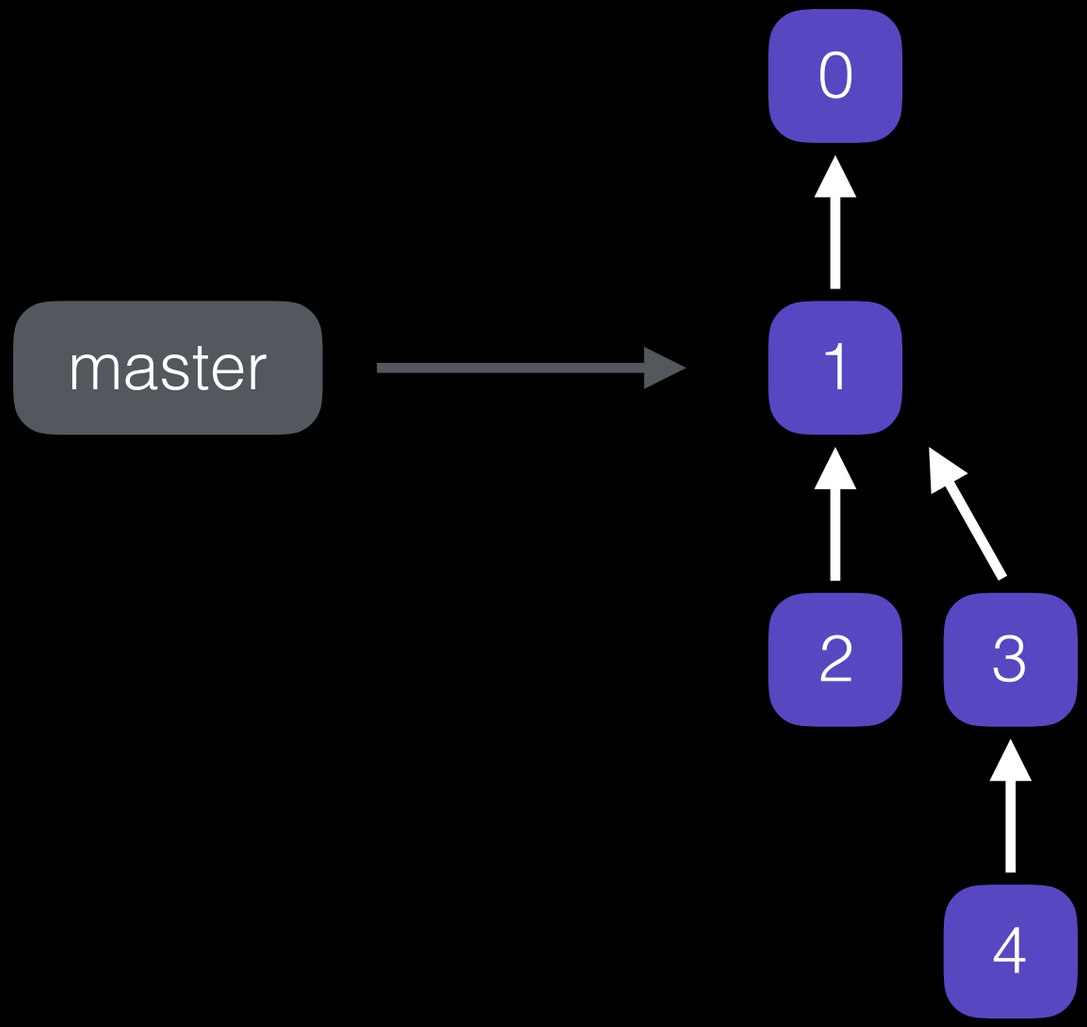
Branching & Merging

Branches

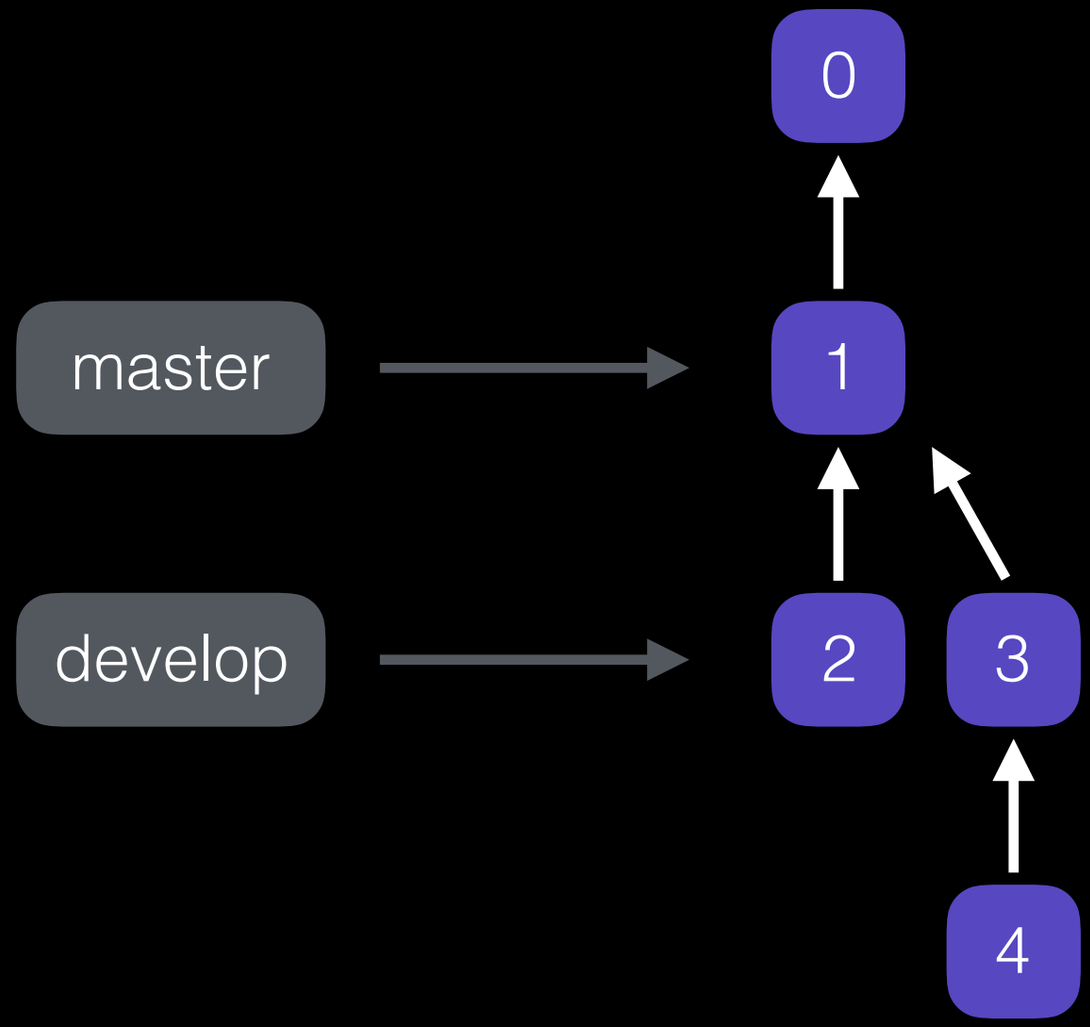
Branches



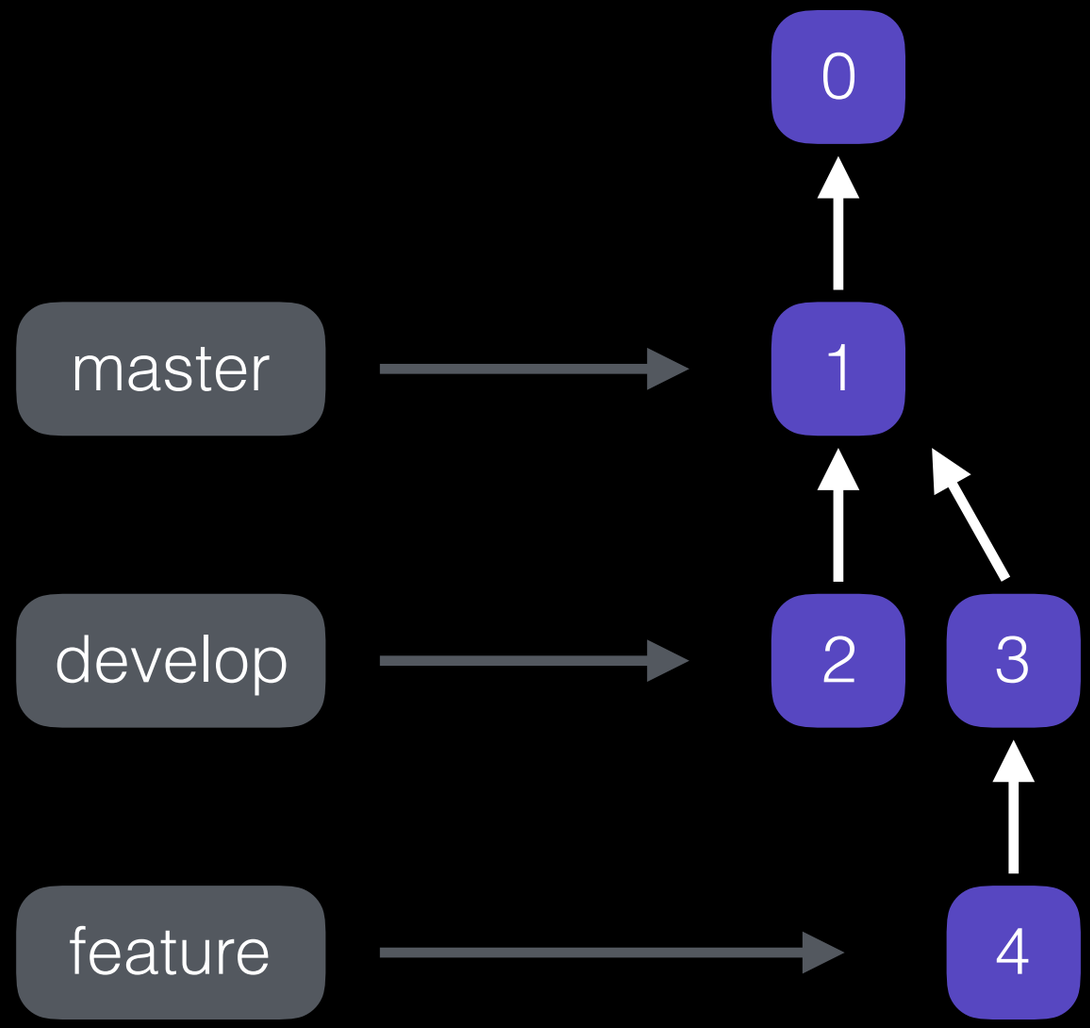
Branches



Branches

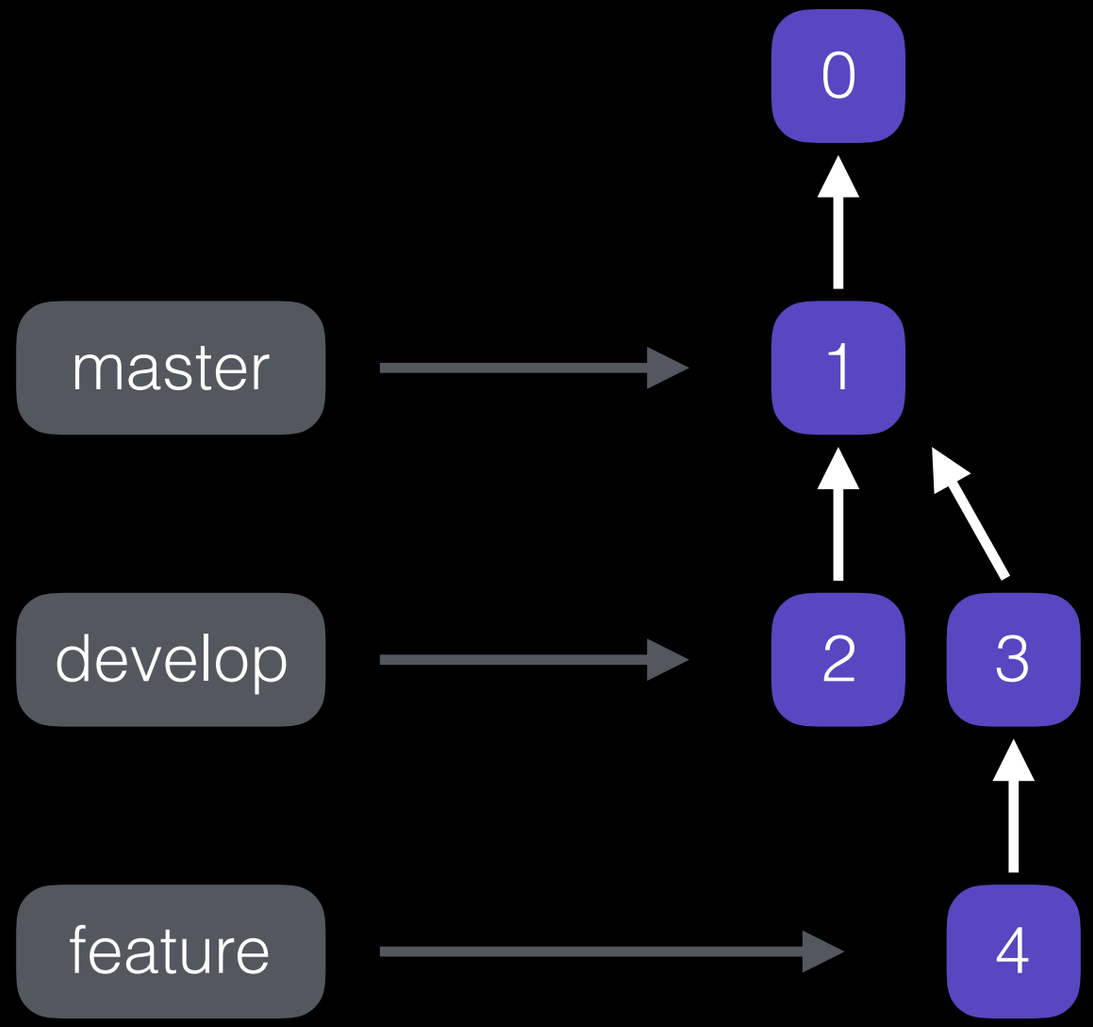


Branches



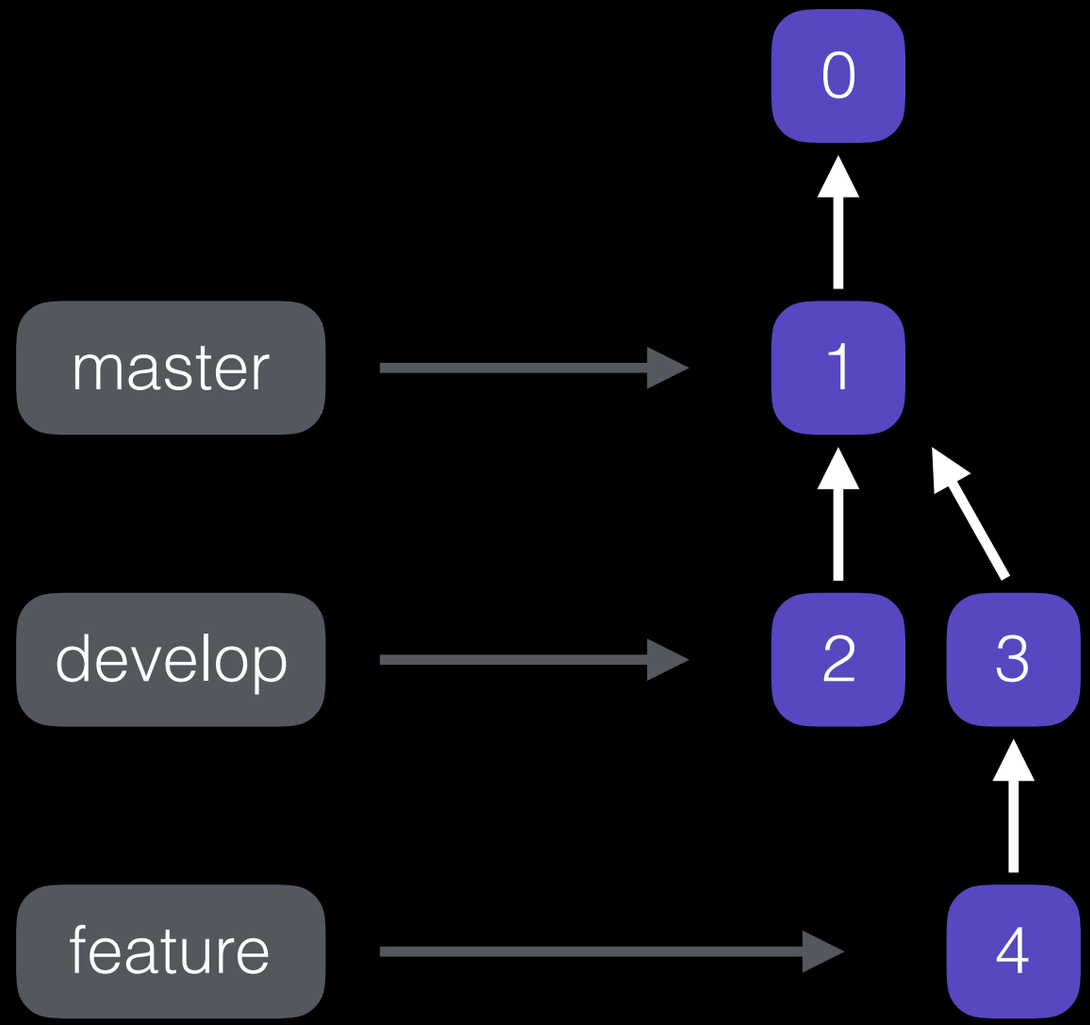
Branches

- Referenz auf Commit



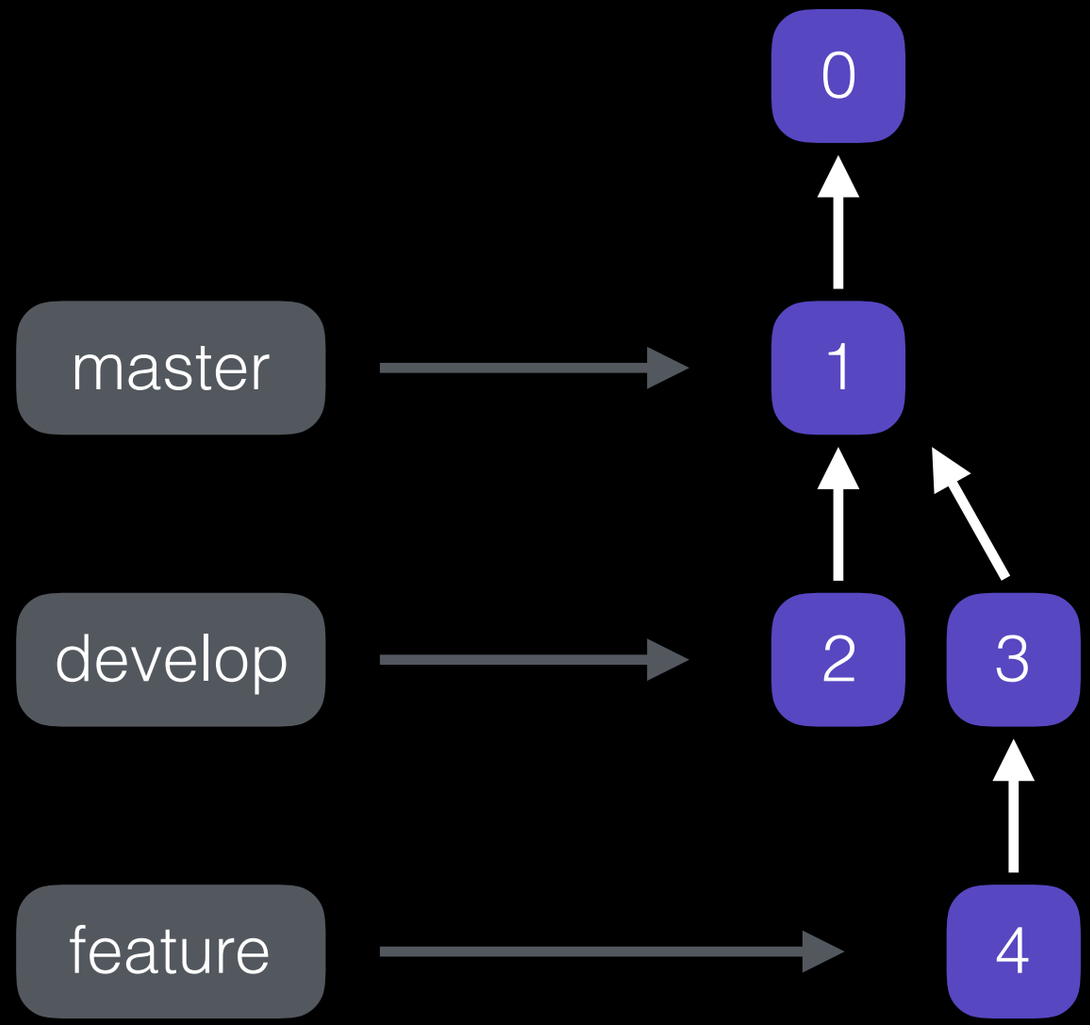
Branches

- Referenz auf Commit
- Pro Branch eine Datei:
  `.git/refs/heads`



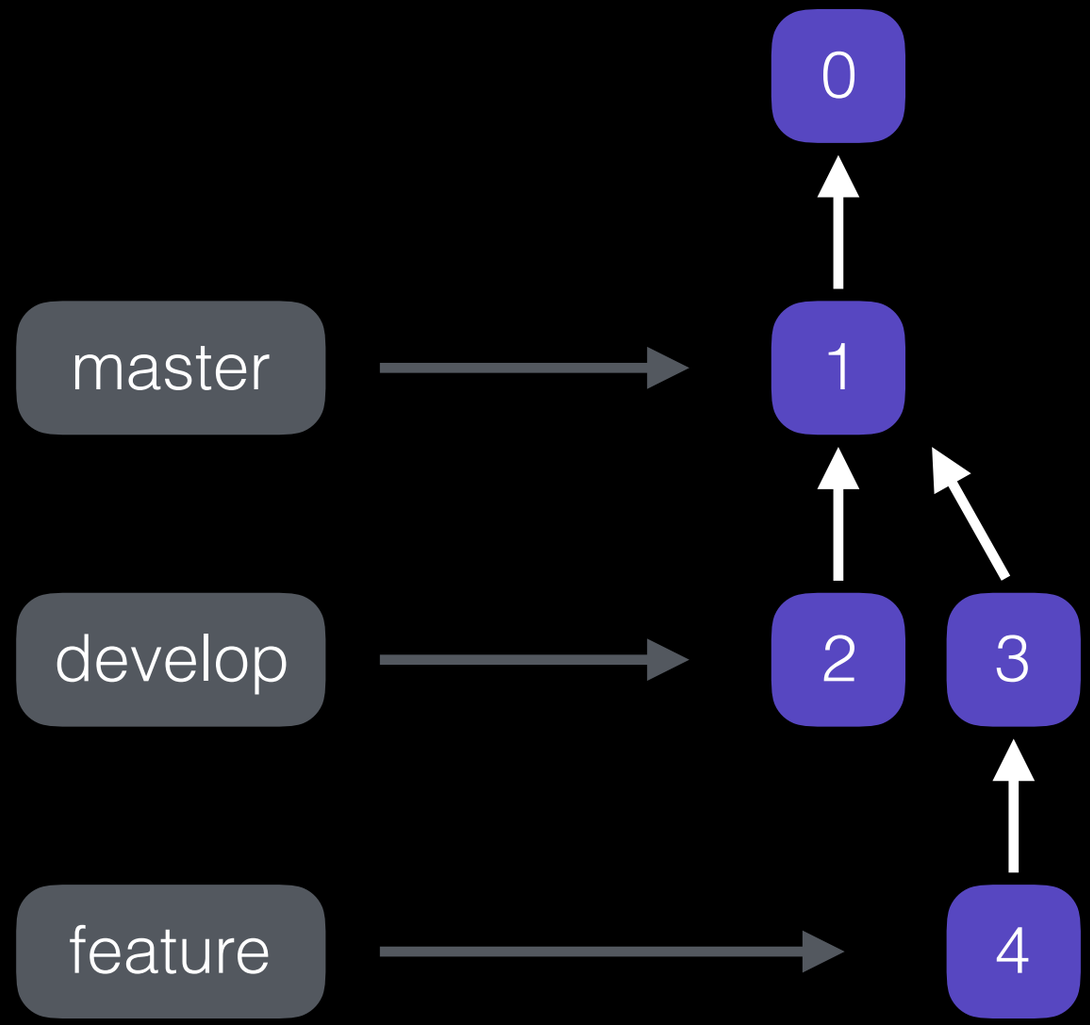
Branches

- Referenz auf Commit
- Pro Branch eine Datei:
  `.git/refs/heads`
- Branch des Working Directory:
  `.git/HEAD`



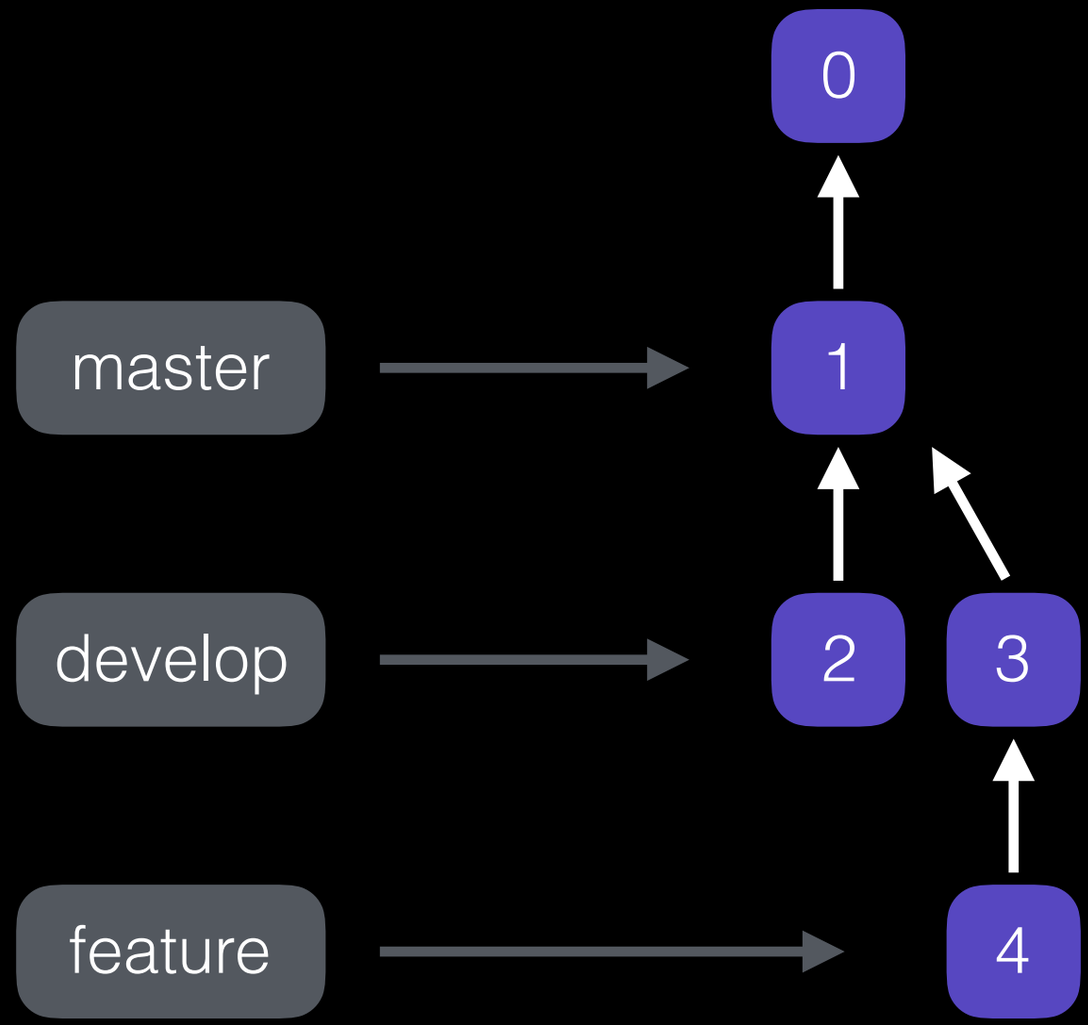
Branches

- Referenz auf Commit
- Pro Branch eine Datei:
  `.git/refs/heads`
- Branch des Working Directory:
  `.git/HEAD`
- Branch-Metadaten
  unversioniert (Force Push…)



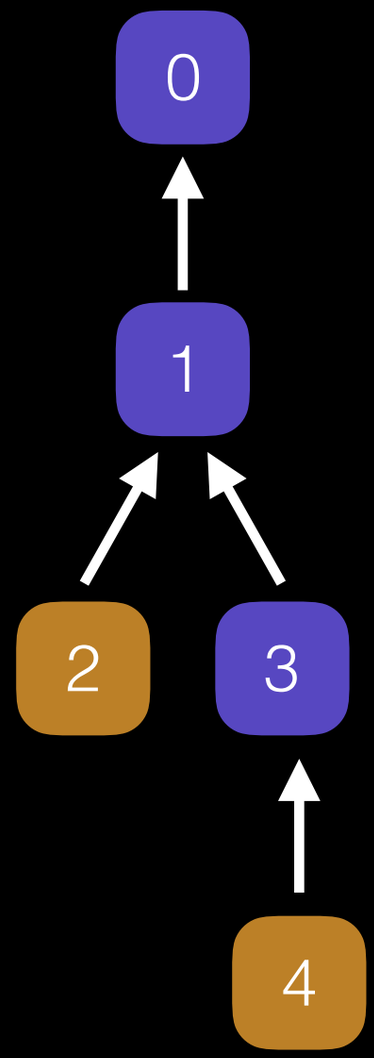
Branches

- Referenz auf Commit
- Pro Branch eine Datei:
  `.git/refs/heads`
- Branch des Working Directory:
  `.git/HEAD`
- Branch-Metadaten
  unversioniert (Force Push…)
- **Reflog:**
  Log aller Branch-Änderungen



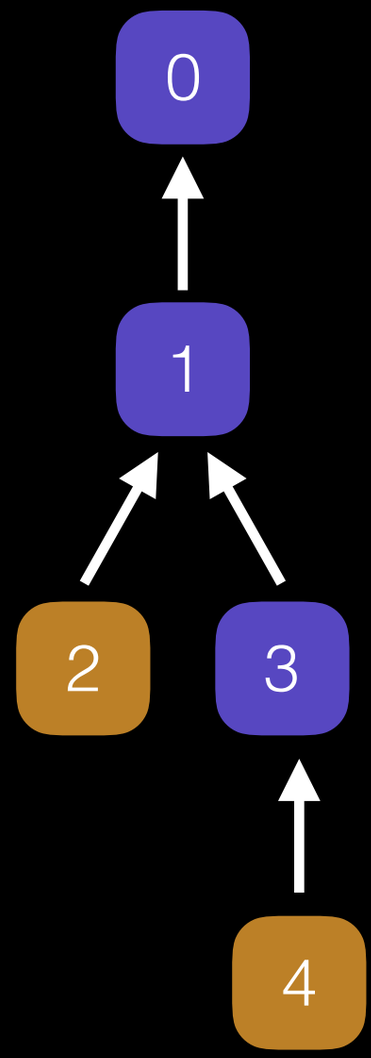
Merging

Merging



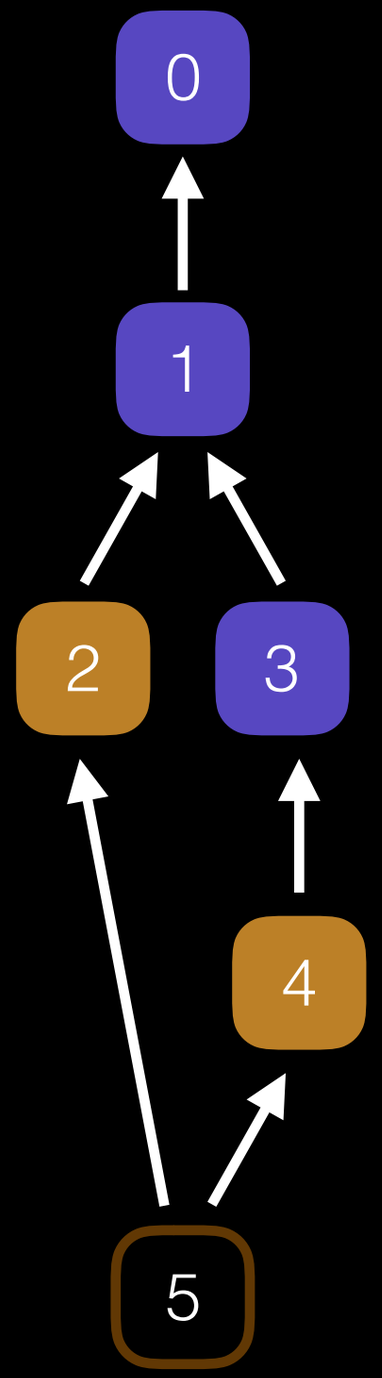
Merging

- Vereinigen zweier Branches



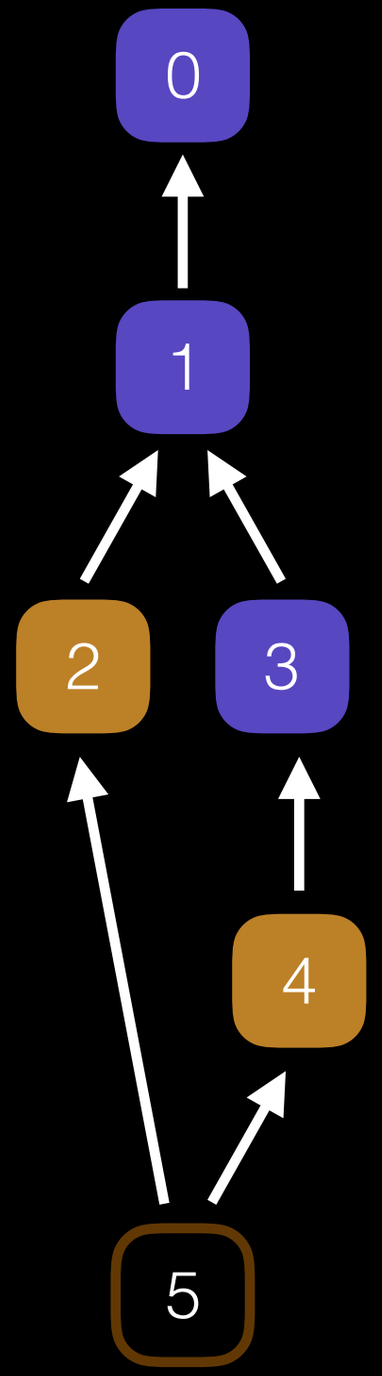
Merging

- Vereinigen zweier Branches
- Merge-Commit:



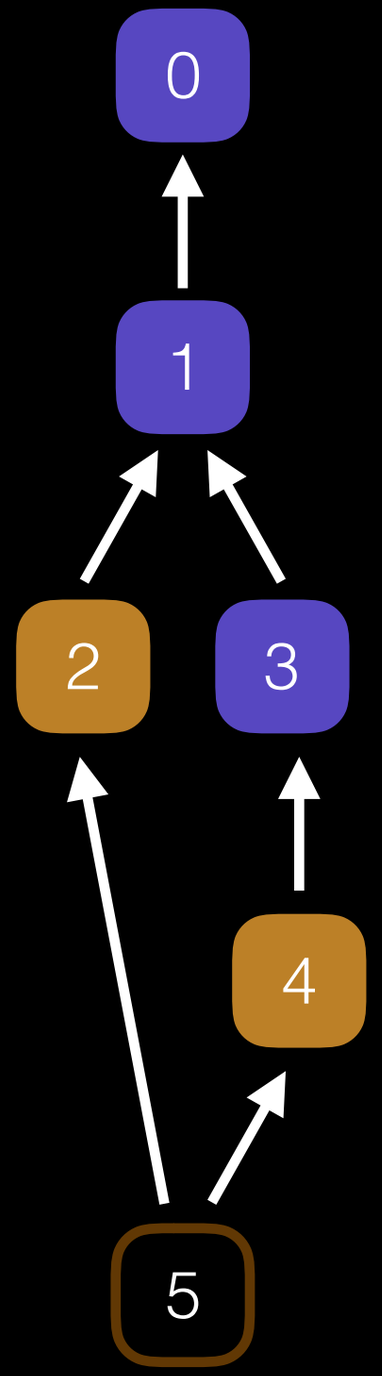
Merging

- Vereinigen zweier Branches
- Merge-Commit:
- Eindeutiger Zustand nach Merge



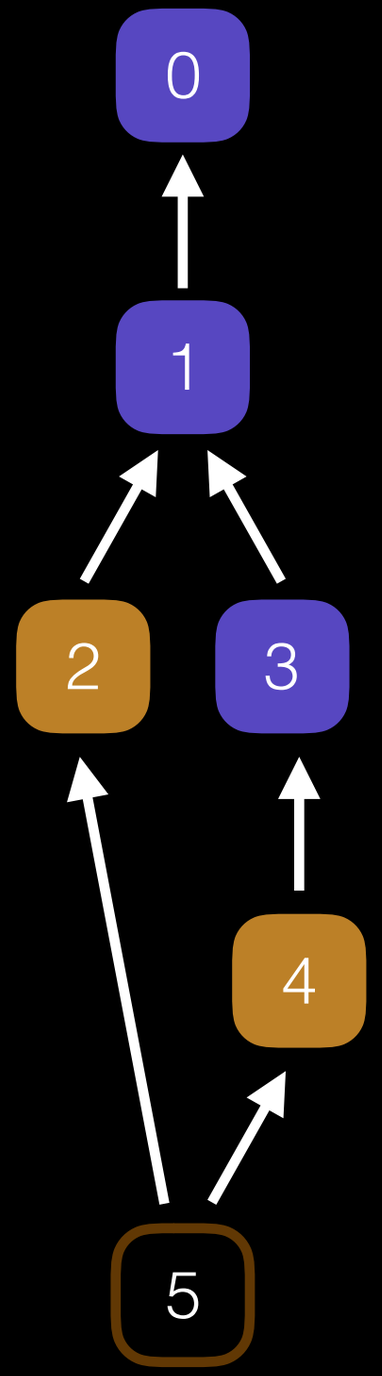
Merging

- Vereinigen zweier Branches
- Merge-Commit:
  - Eindeutiger Zustand nach Merge
  - Ausgewählte Änderungen

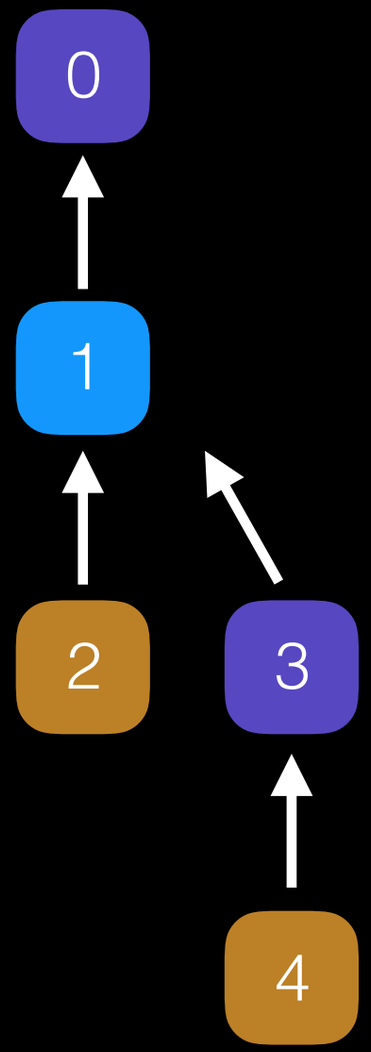


Merging

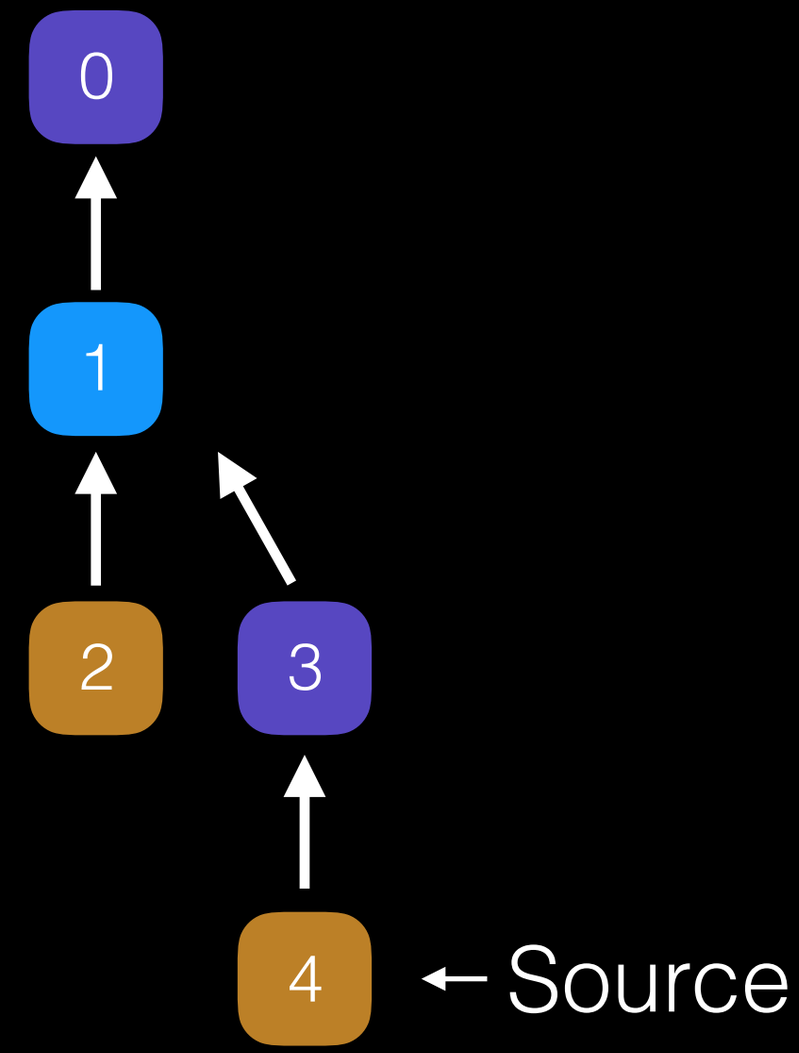
- Vereinigen zweier Branches
- Merge-Commit:
  - Eindeutiger Zustand nach Merge
  - Ausgewählte Änderungen
  - Behebung von Konflikten



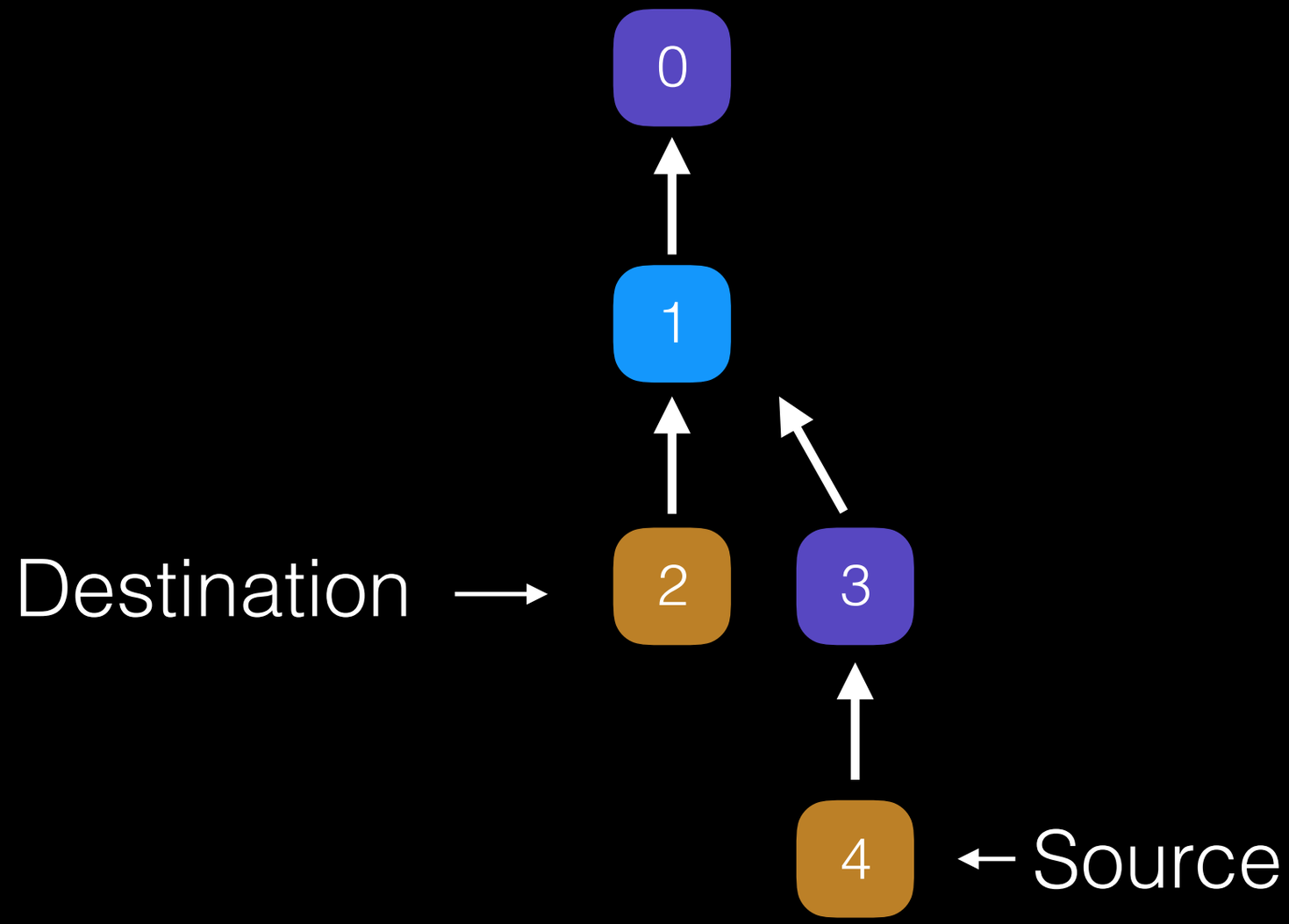
Three-Way-Merge



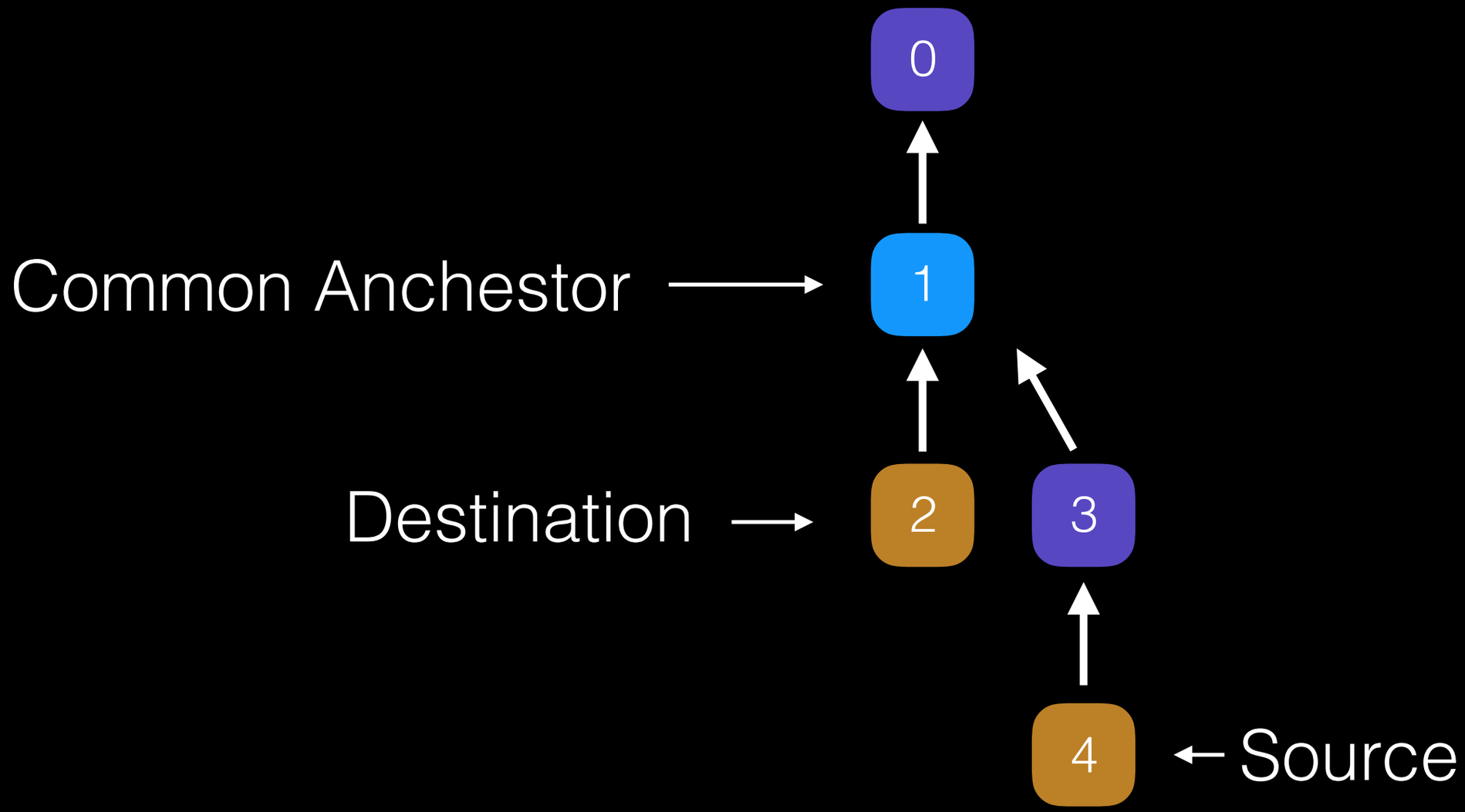
Three-Way-Merge



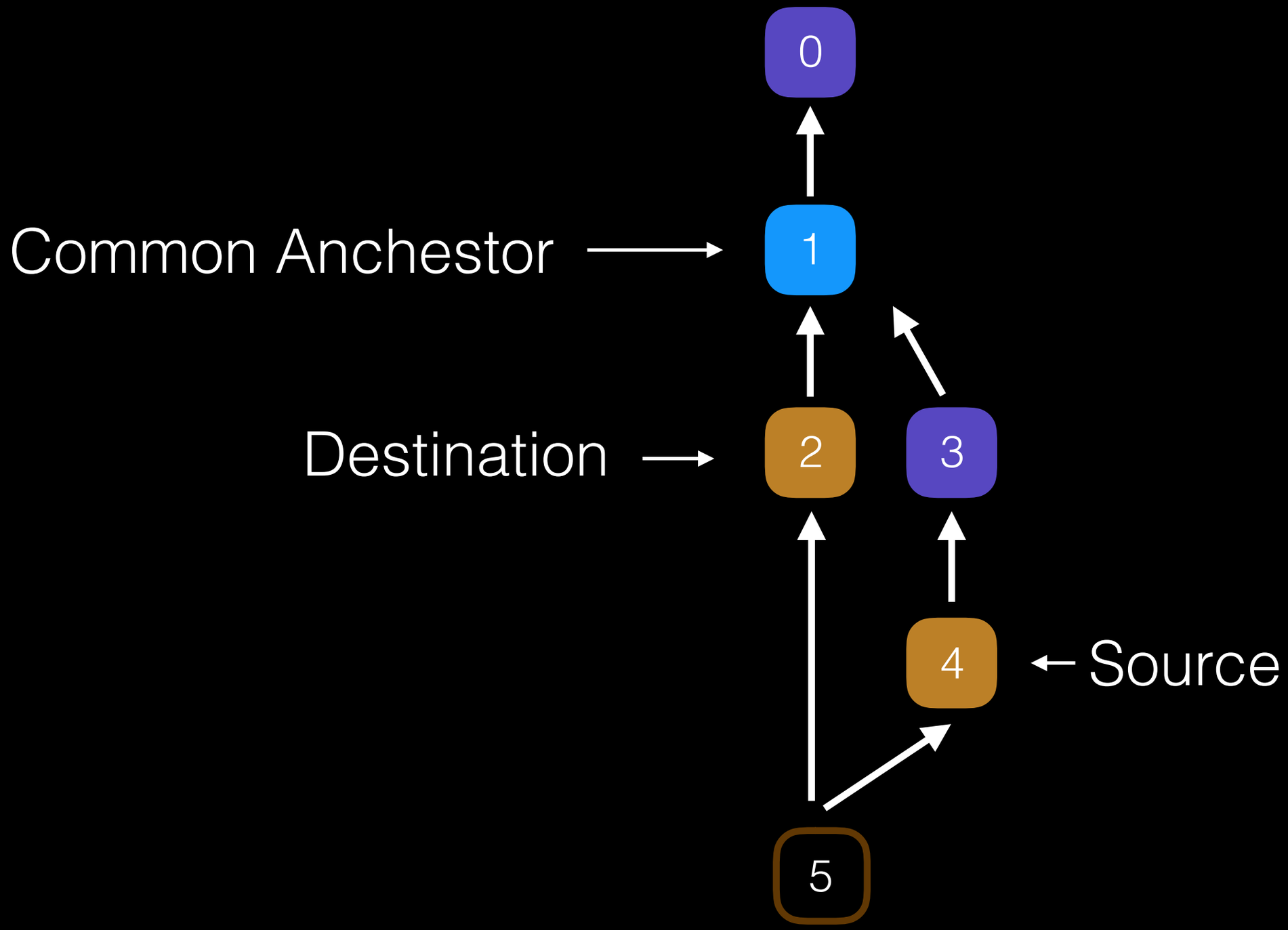
Three-Way-Merge



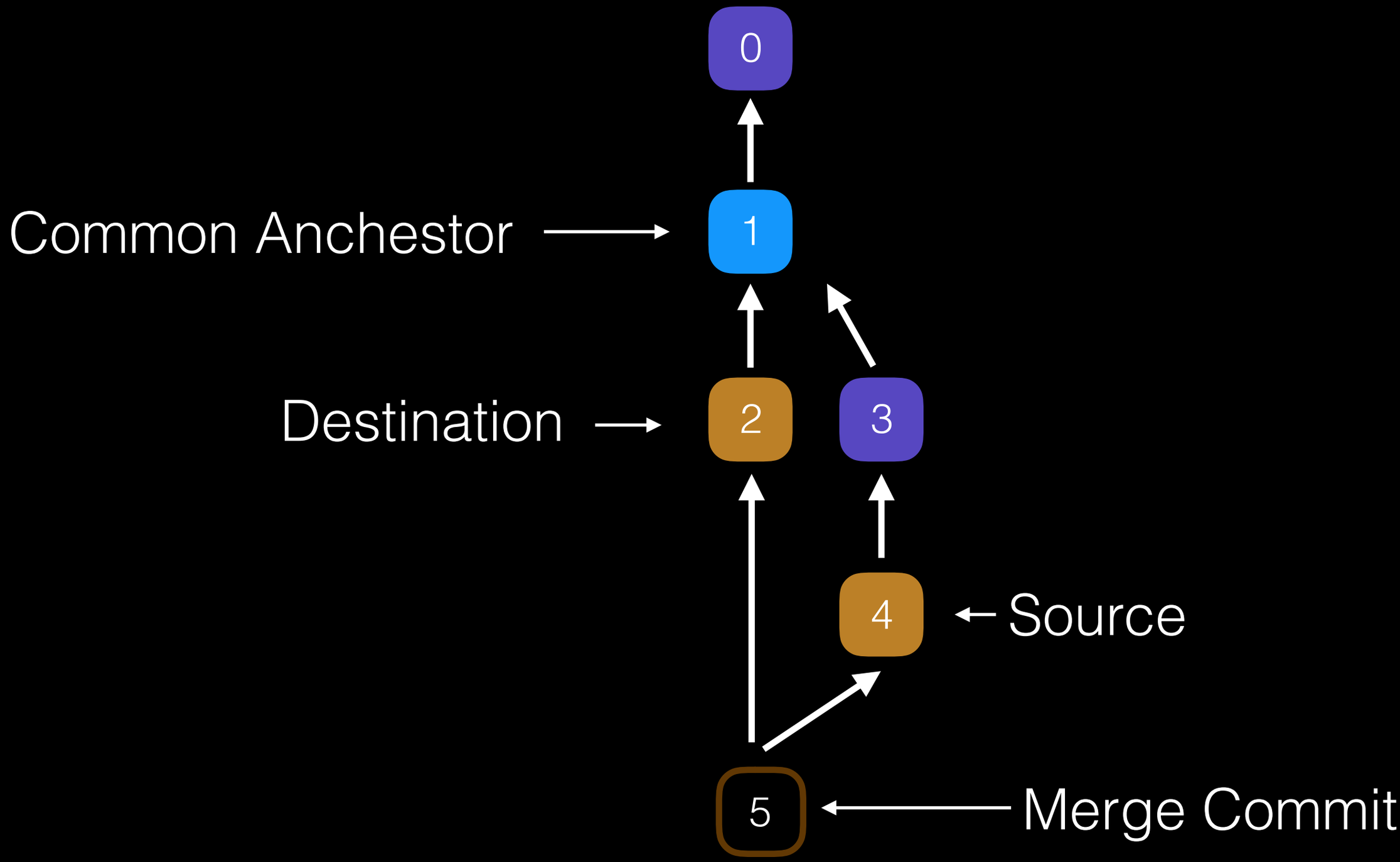
Three-Way-Merge



Three-Way-Merge



Three-Way-Merge



Three-Way-Merge

Three-Way-Merge

Three-Way-Merge

Zeile	Anchestor	Source	Dest	Merge
1	a	a	a	a

Three-Way-Merge

Zeile	Anchestor	Source	Dest	Merge
1	a	a	a	a
2	b	b	B	B

Three-Way-Merge

Zeile	Anchestor	Source	Dest	Merge
1	a	a	a	a
2	b	b	B	B

Three-Way-Merge

Zeile	Anchestor	Source	Dest	Merge
1	a	a	a	a
2	b	b	B	B
35			x	x

Three-Way-Merge

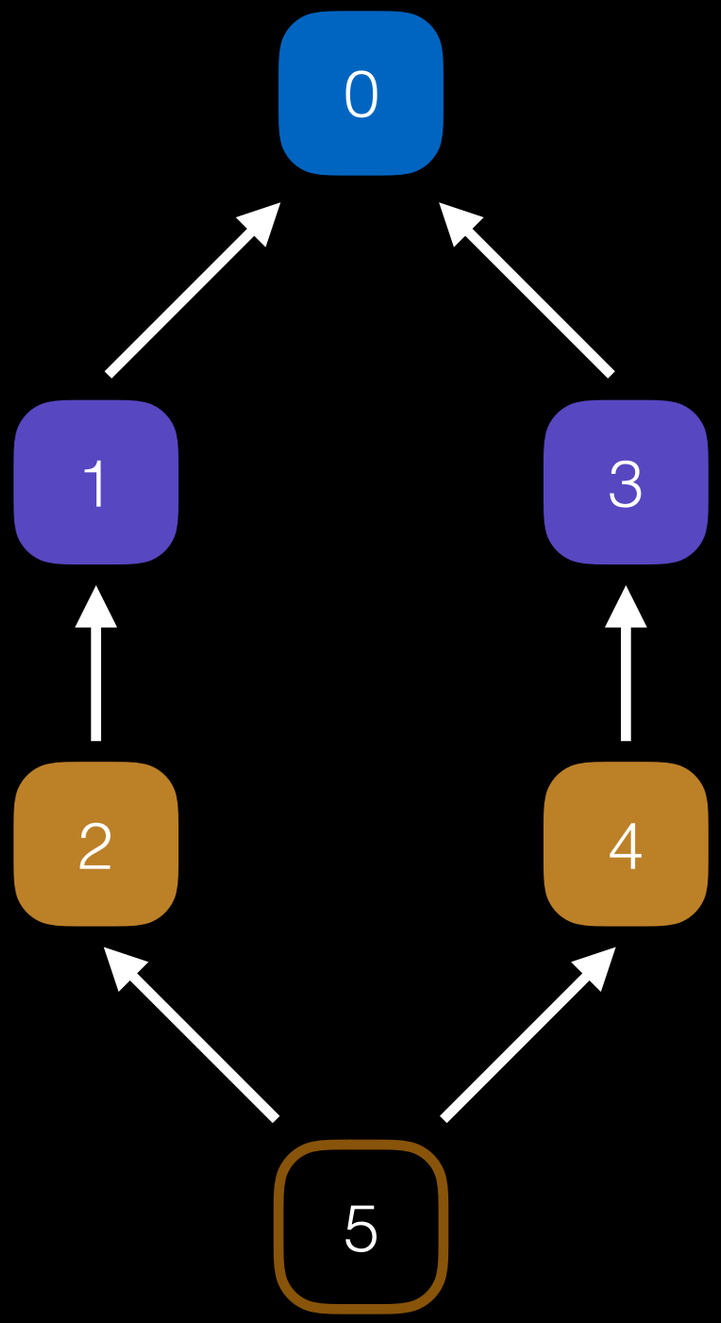
Zeile	Anchestor	Source	Dest	Merge
1	a	a	a	a
2	b	b	B	B
35			x	x

Three-Way-Merge

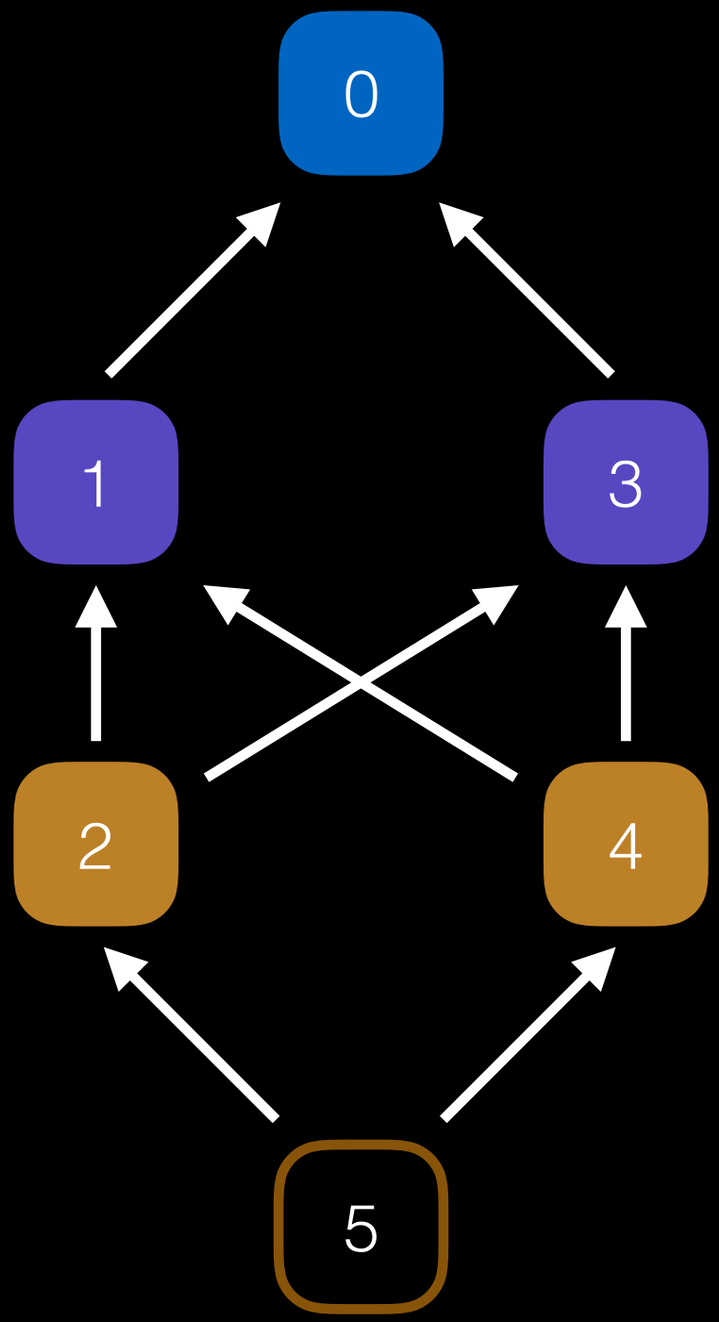
Zeile	Anchestor	Source	Dest	Merge
1	a	a	a	a
2	b	b	B	B
35			x	x
42	d	e	f	**Konflikt**

Common Anchestor?

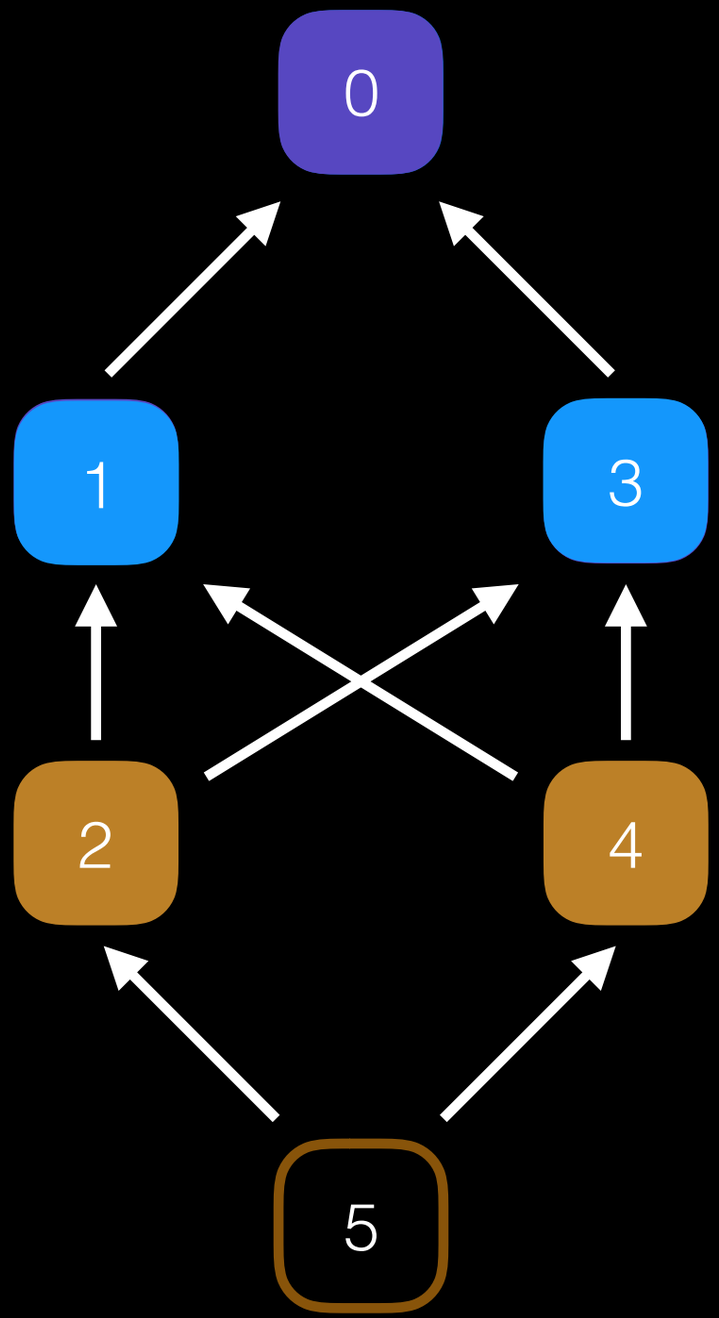
Common Anchestor?



Common Anchestor?



Common Anchestor?



Common Anchestor?

Common Anchestor?

develop

Common Anchestor?

0: ab

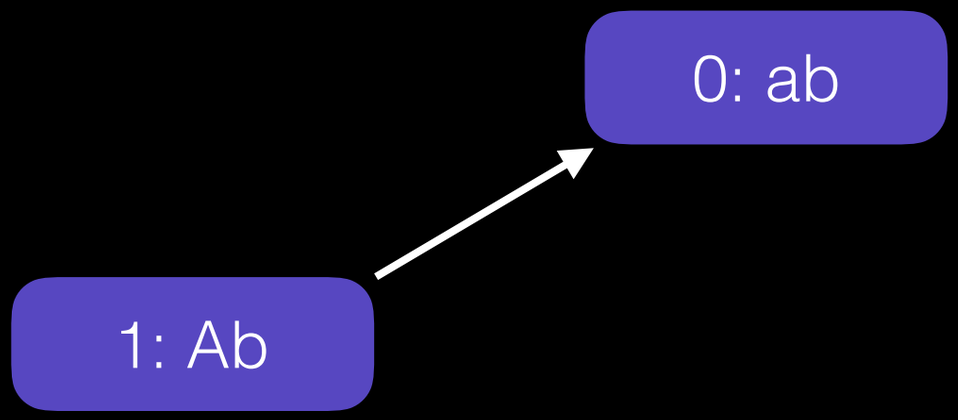
develop

Common Anchestor?

0: ab

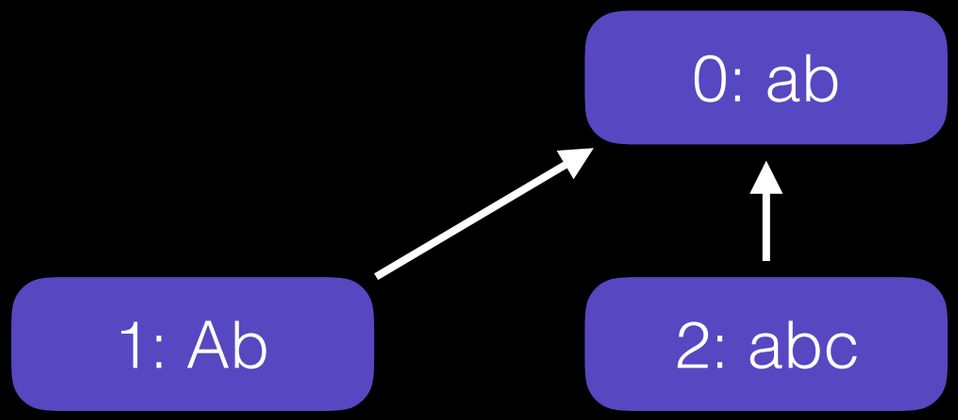
feature A develop

Common Anchestor?



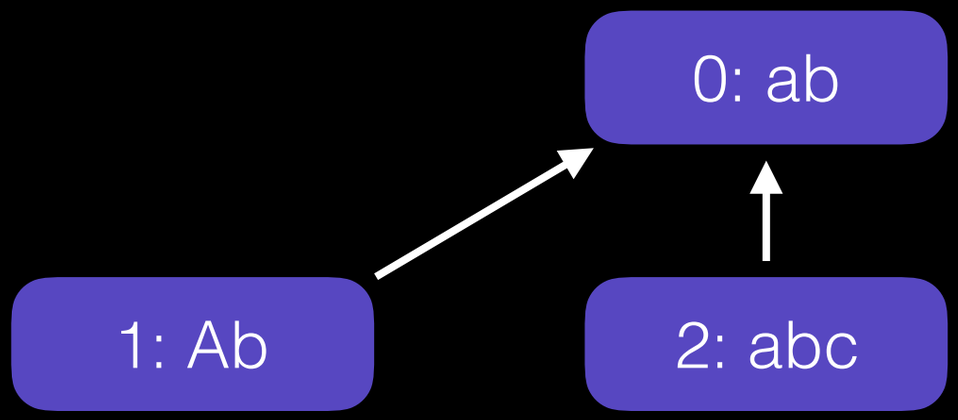
feature A develop

Common Anchestor?



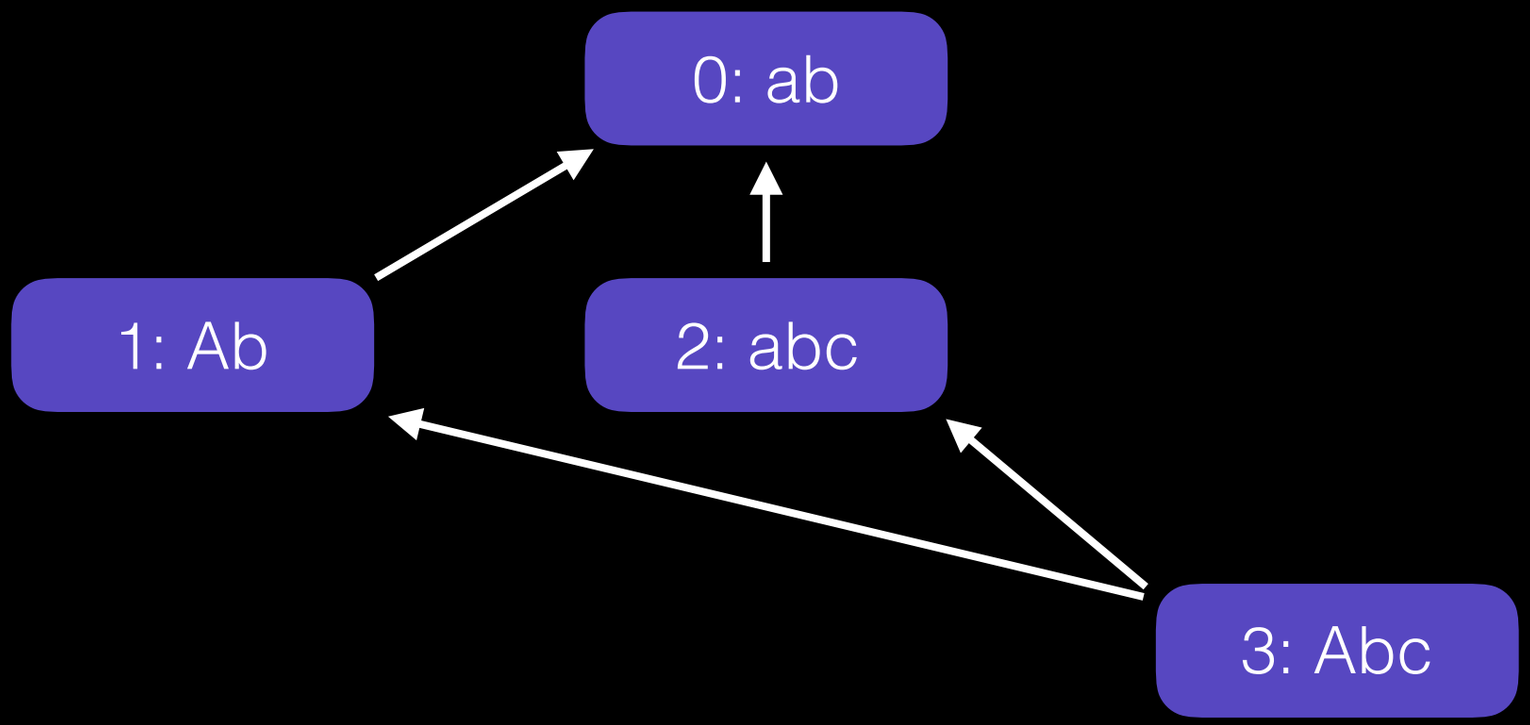
feature A develop

Common Anchestor?



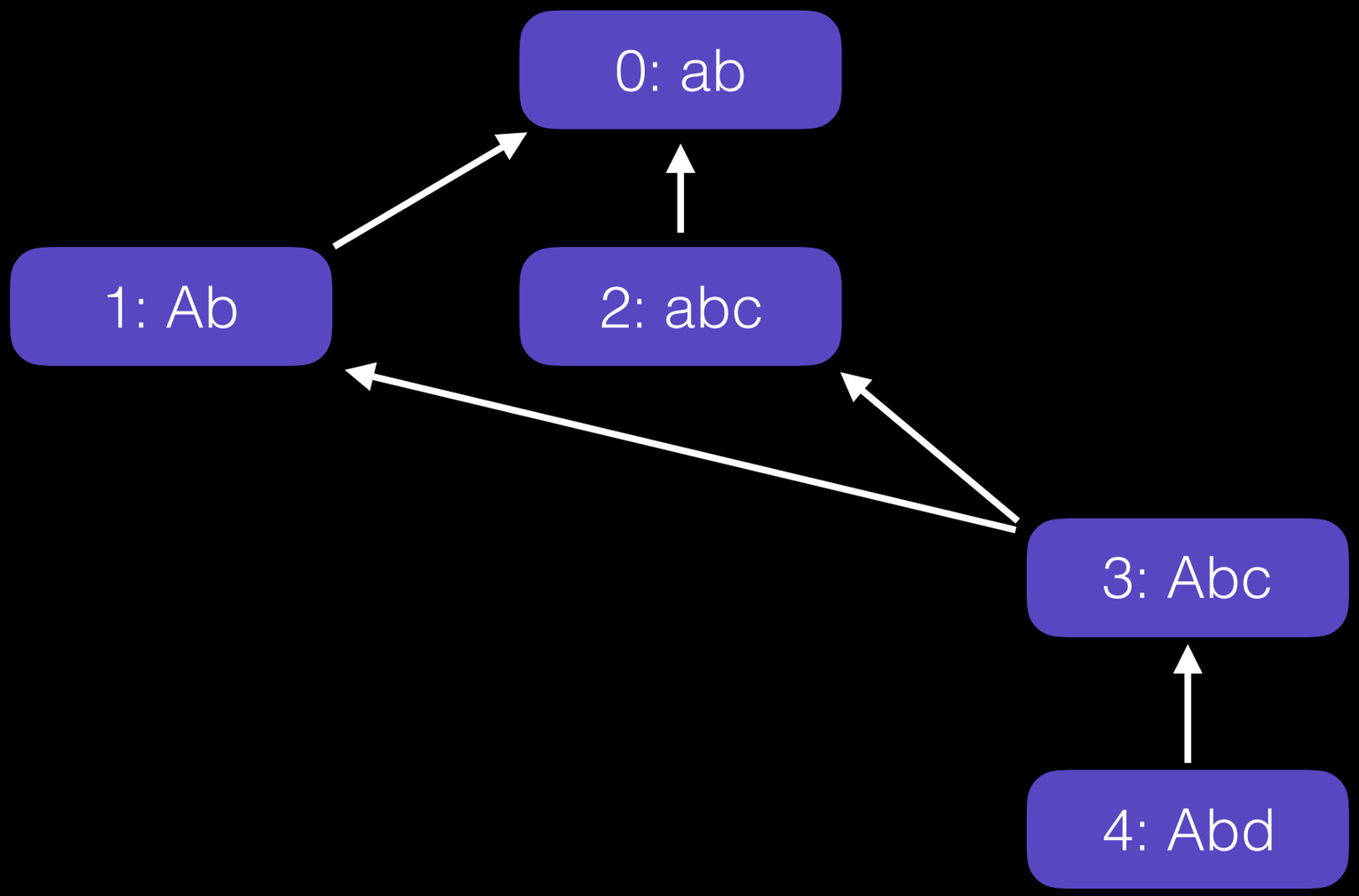
feature A develop feature B

Common Anchestor?



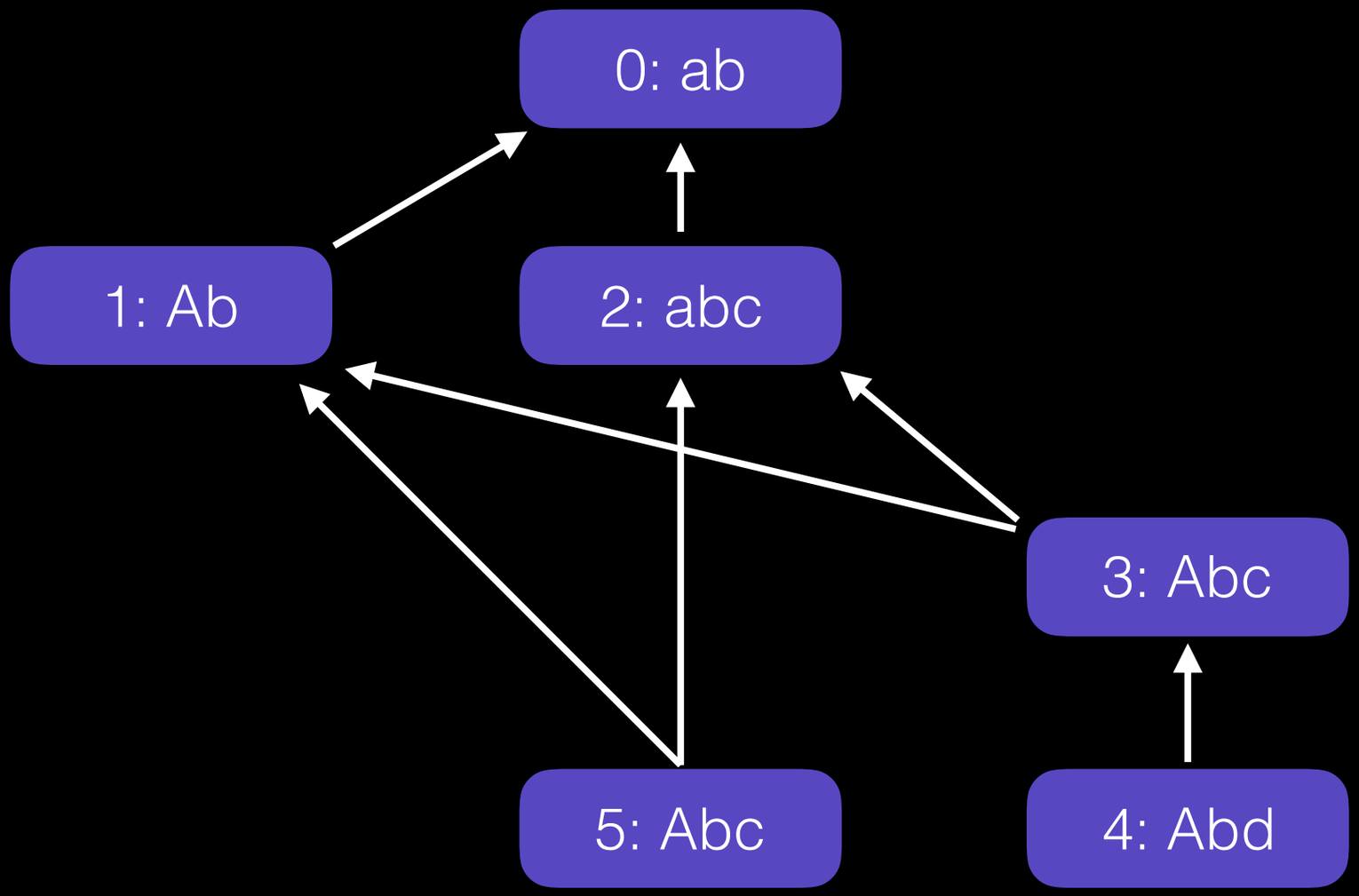
feature A　　develop　　feature B

Common Anchestor?



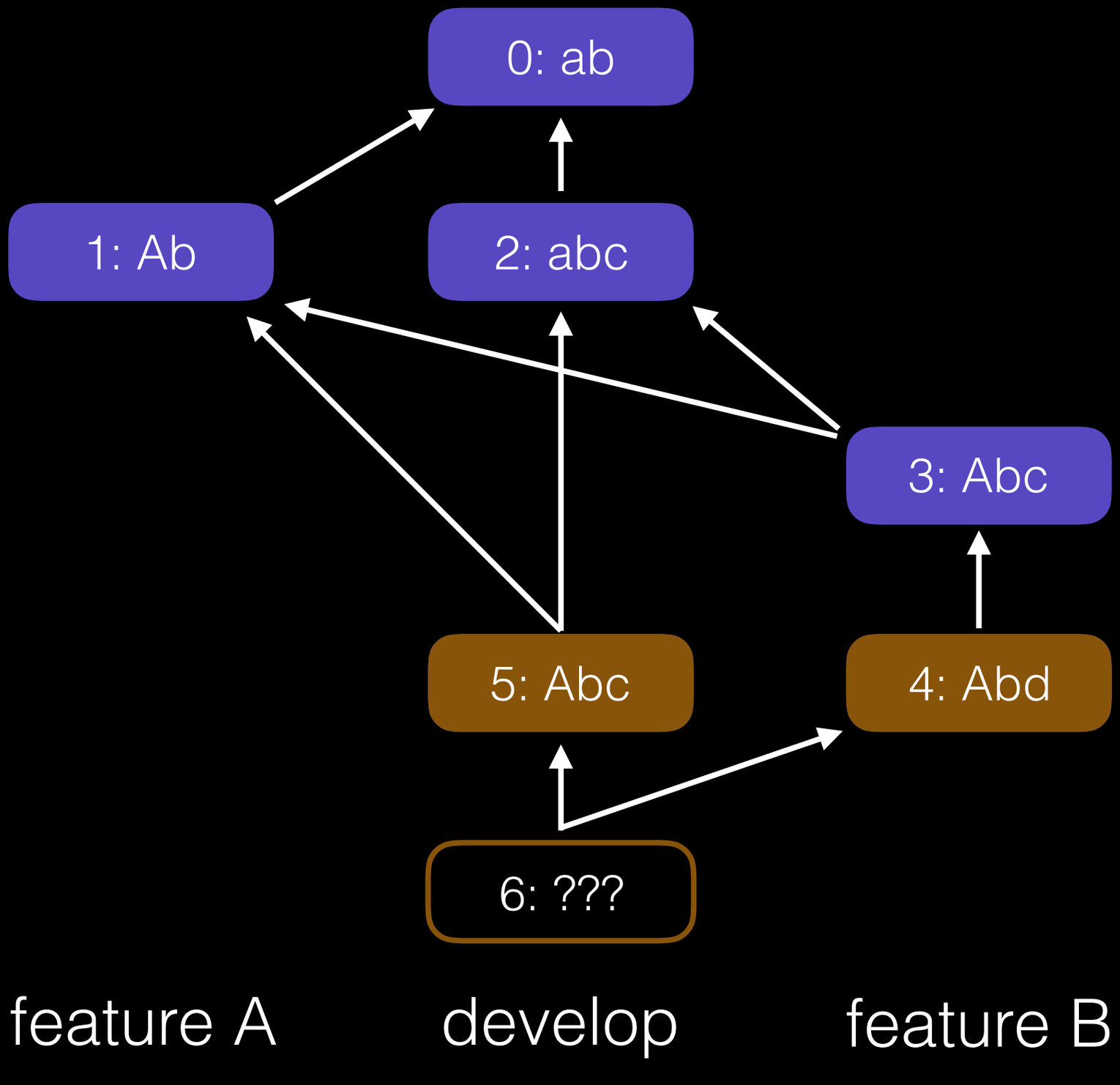
feature A develop feature B

Common Anchestor?

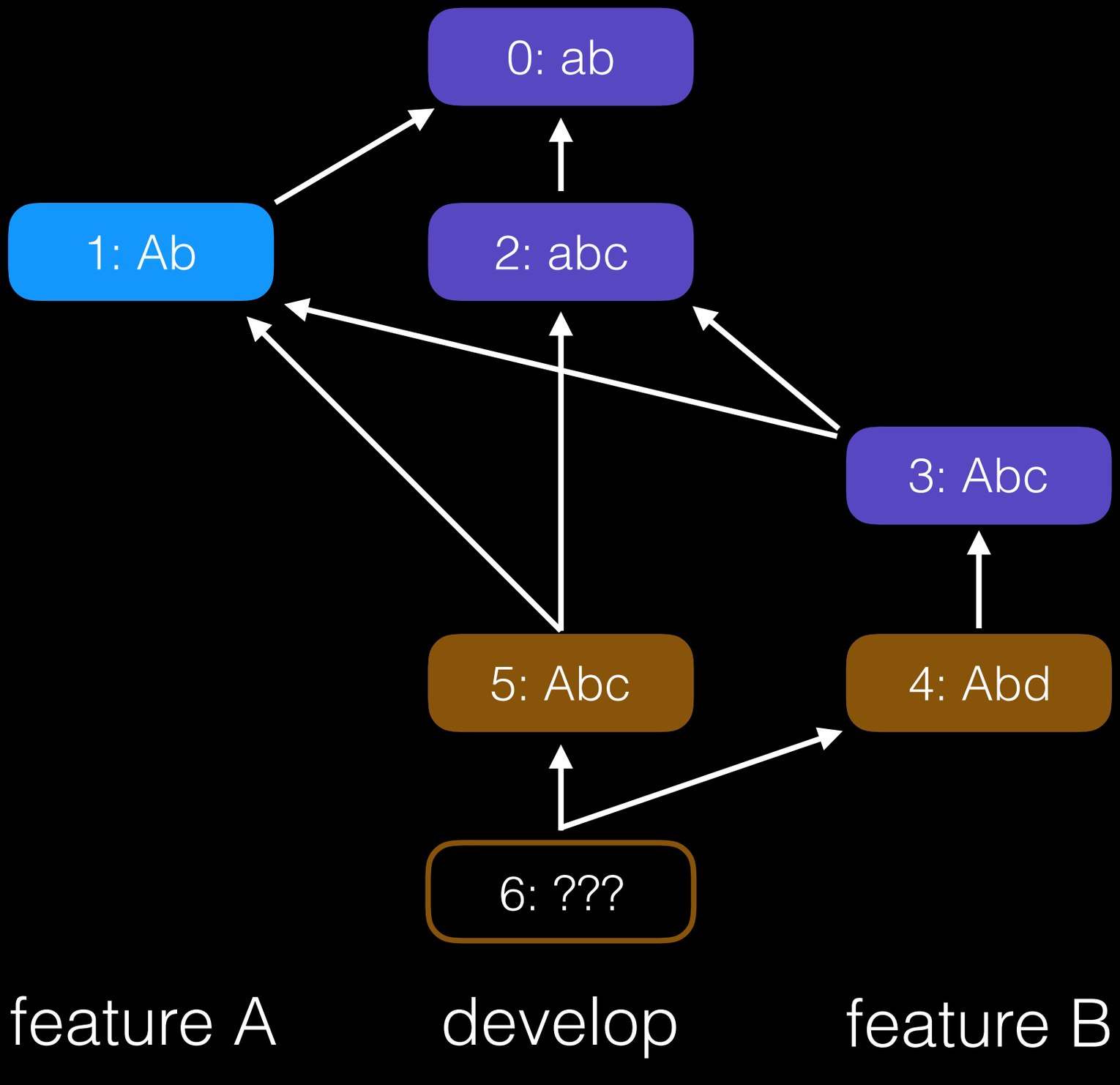


feature A develop feature B

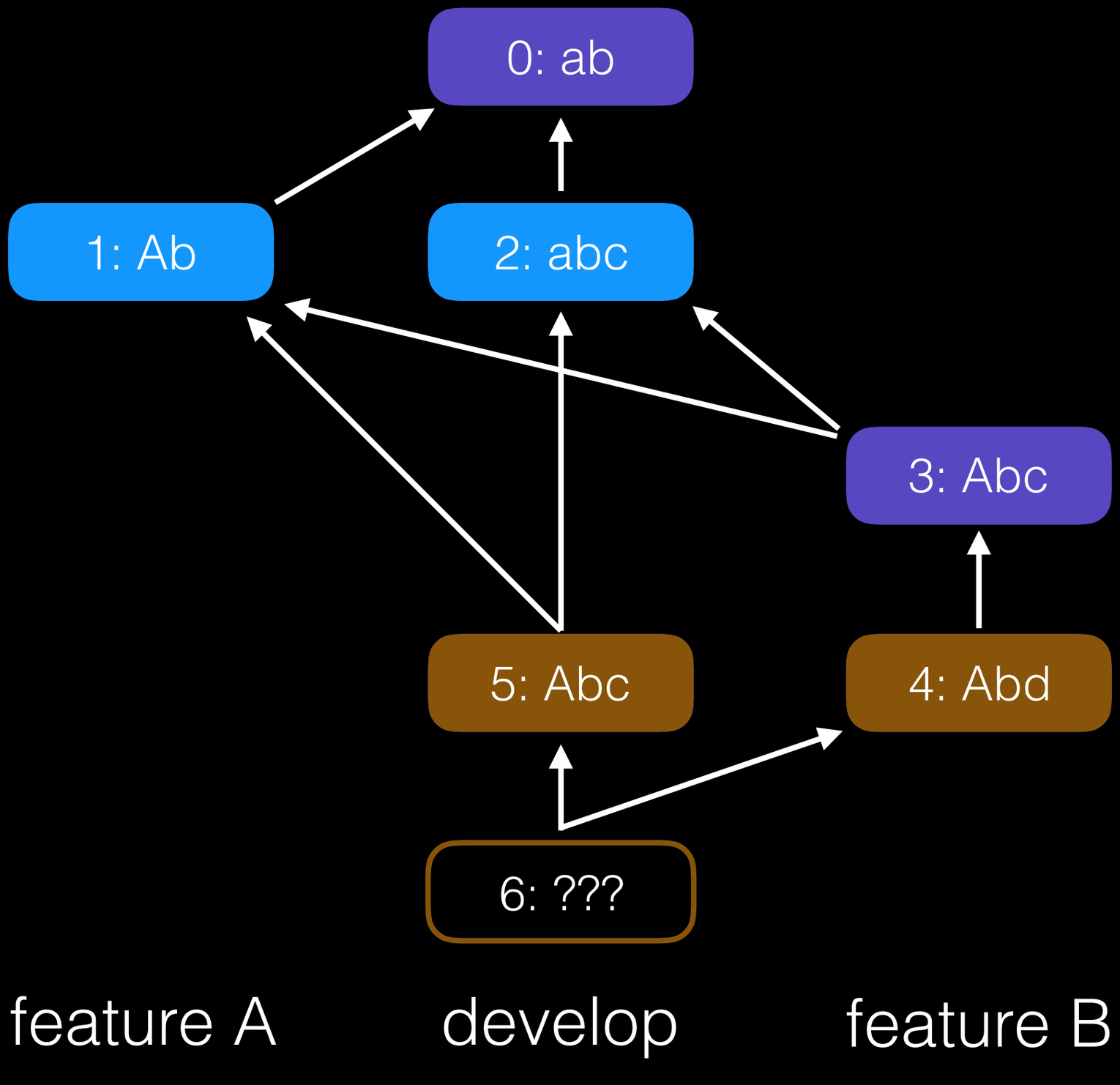
Common Anchestor?



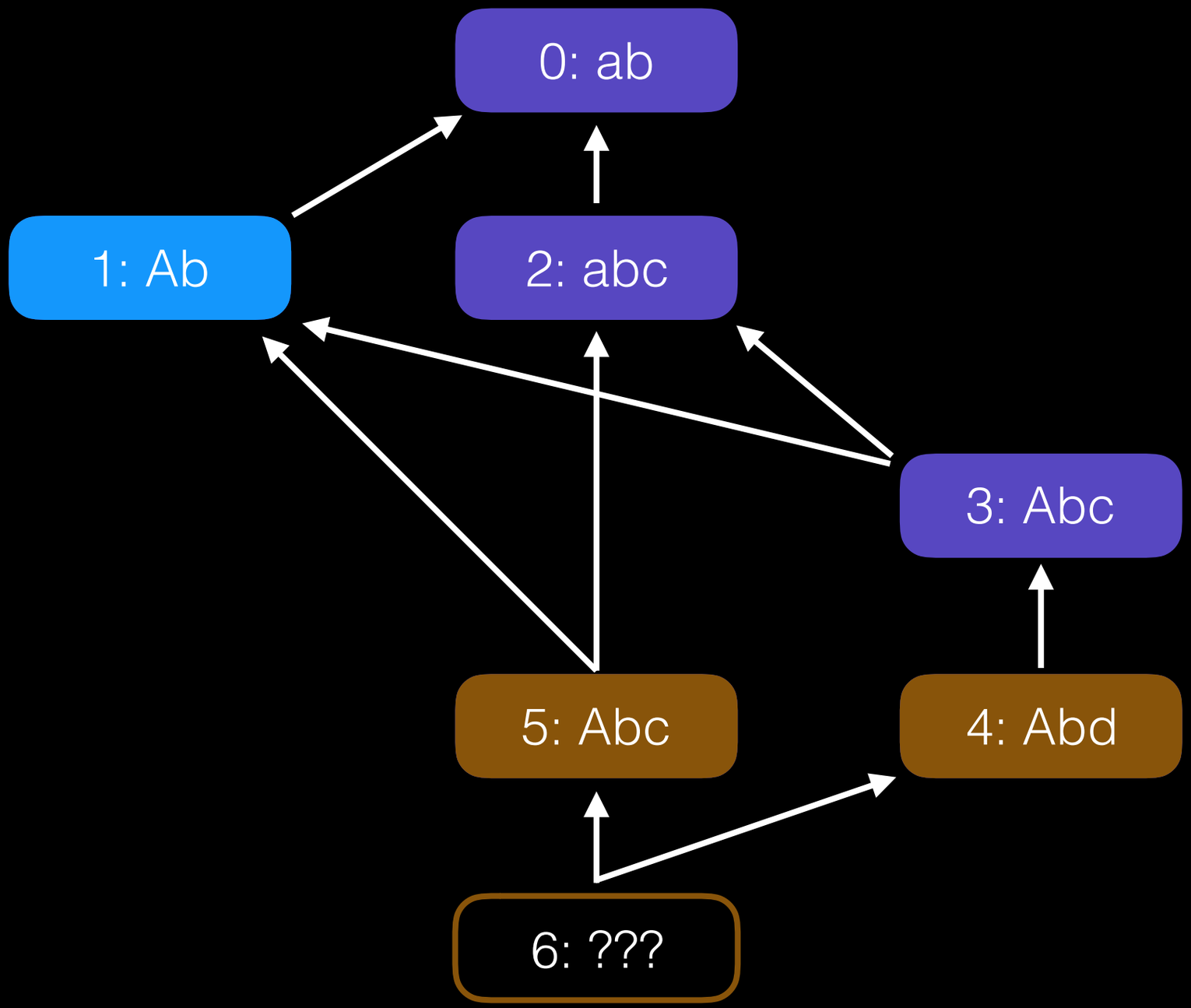
Common Anchestor?



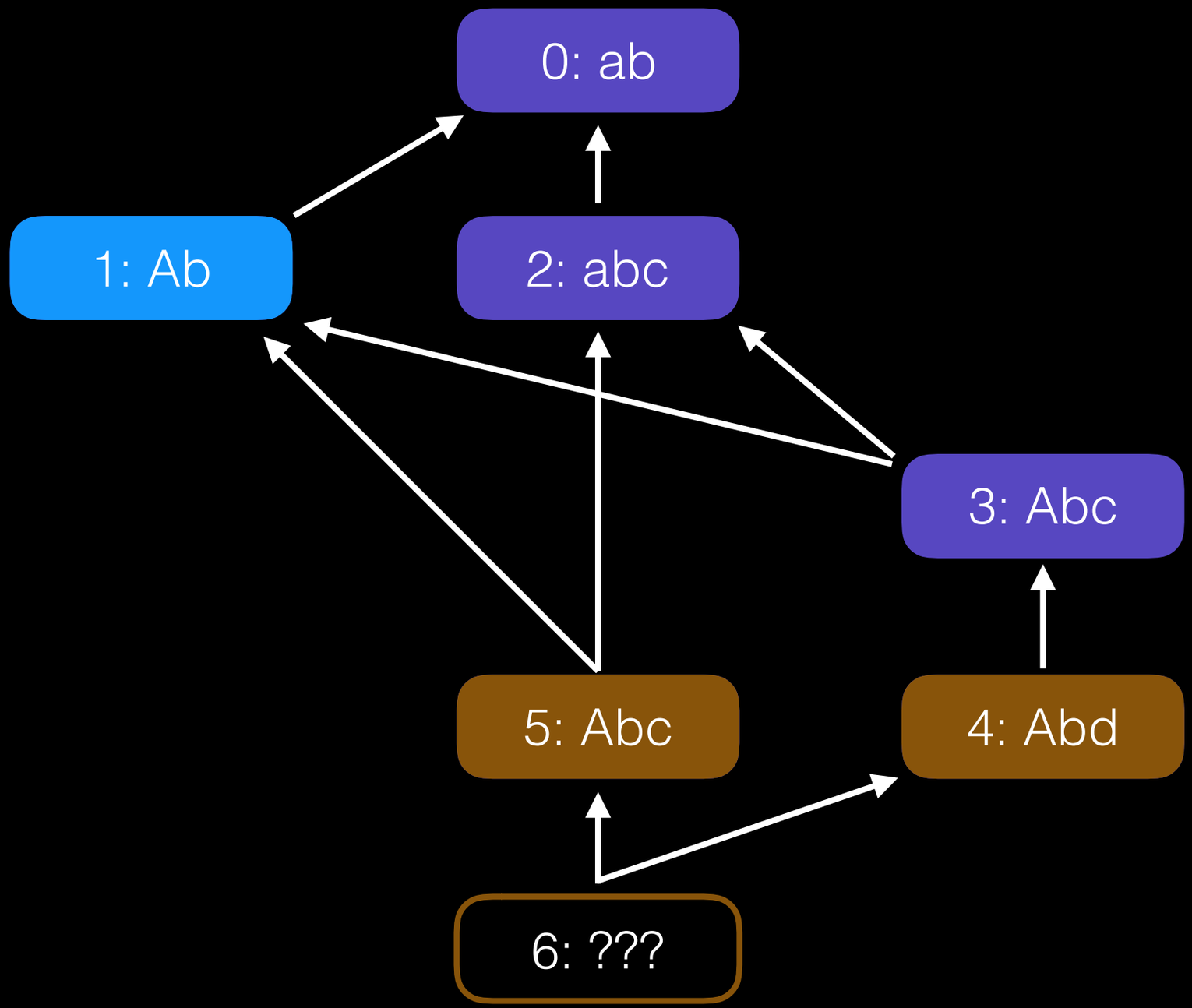
Common Anchestor?



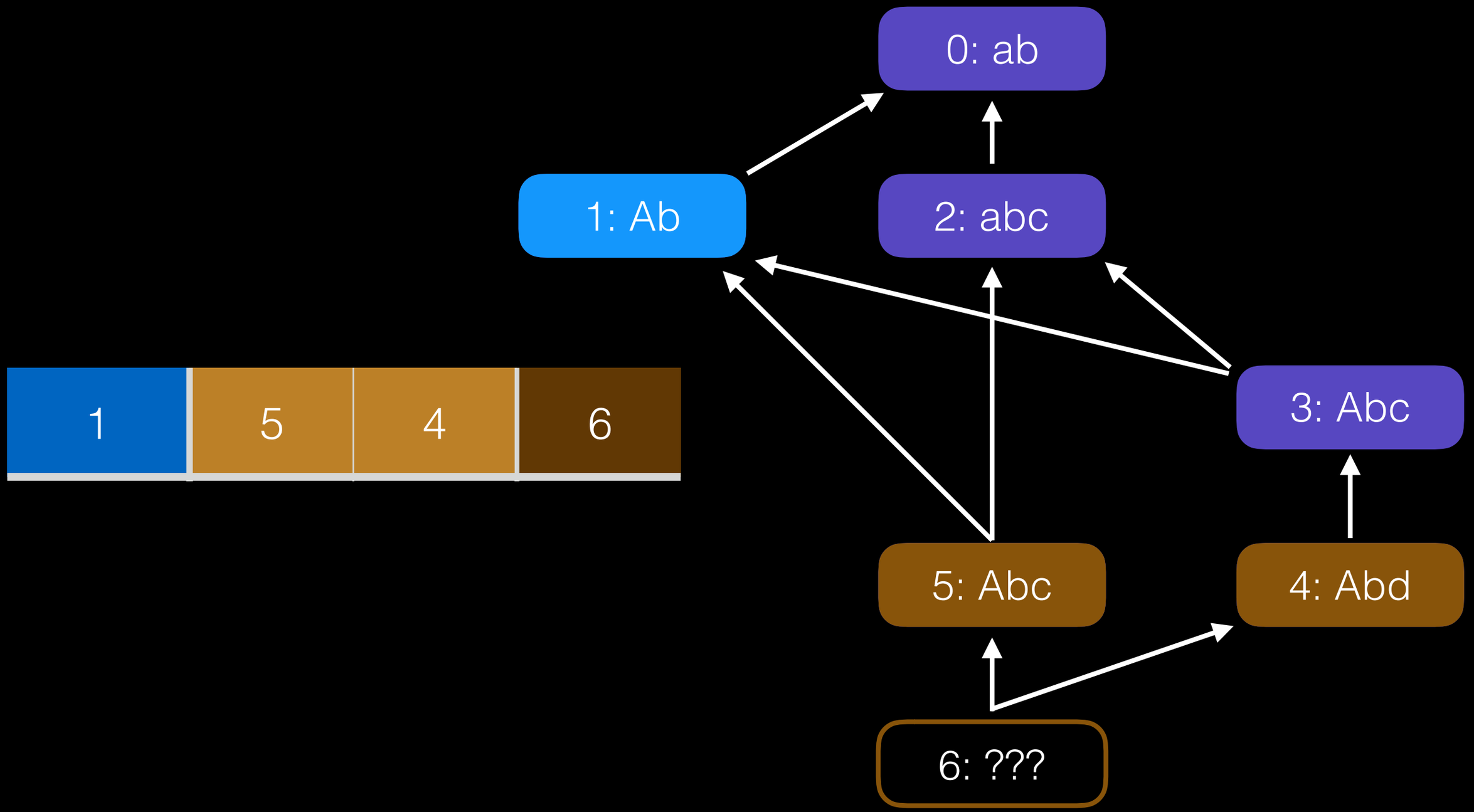
Würfeln?



Würfeln?

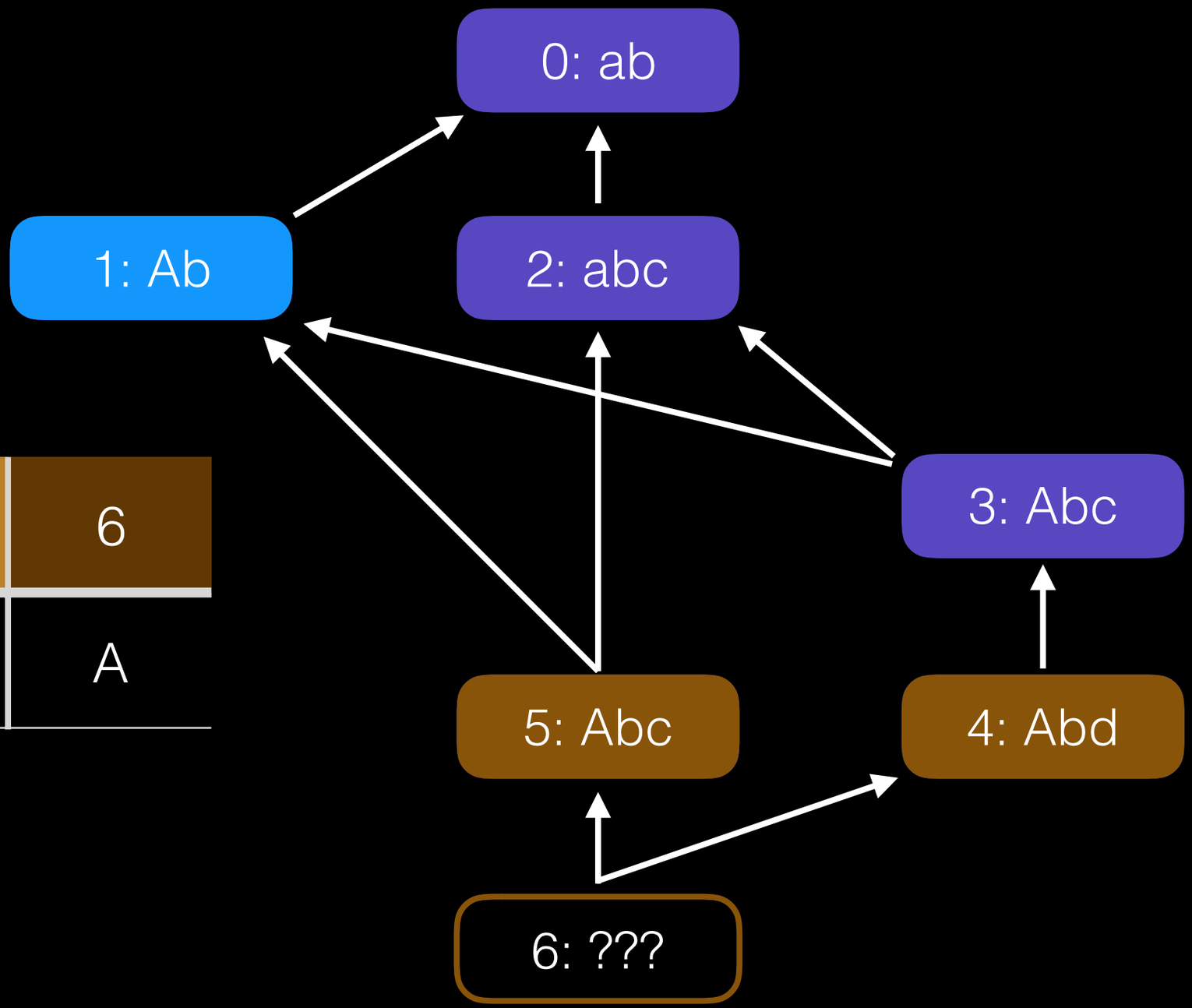


Würfeln?

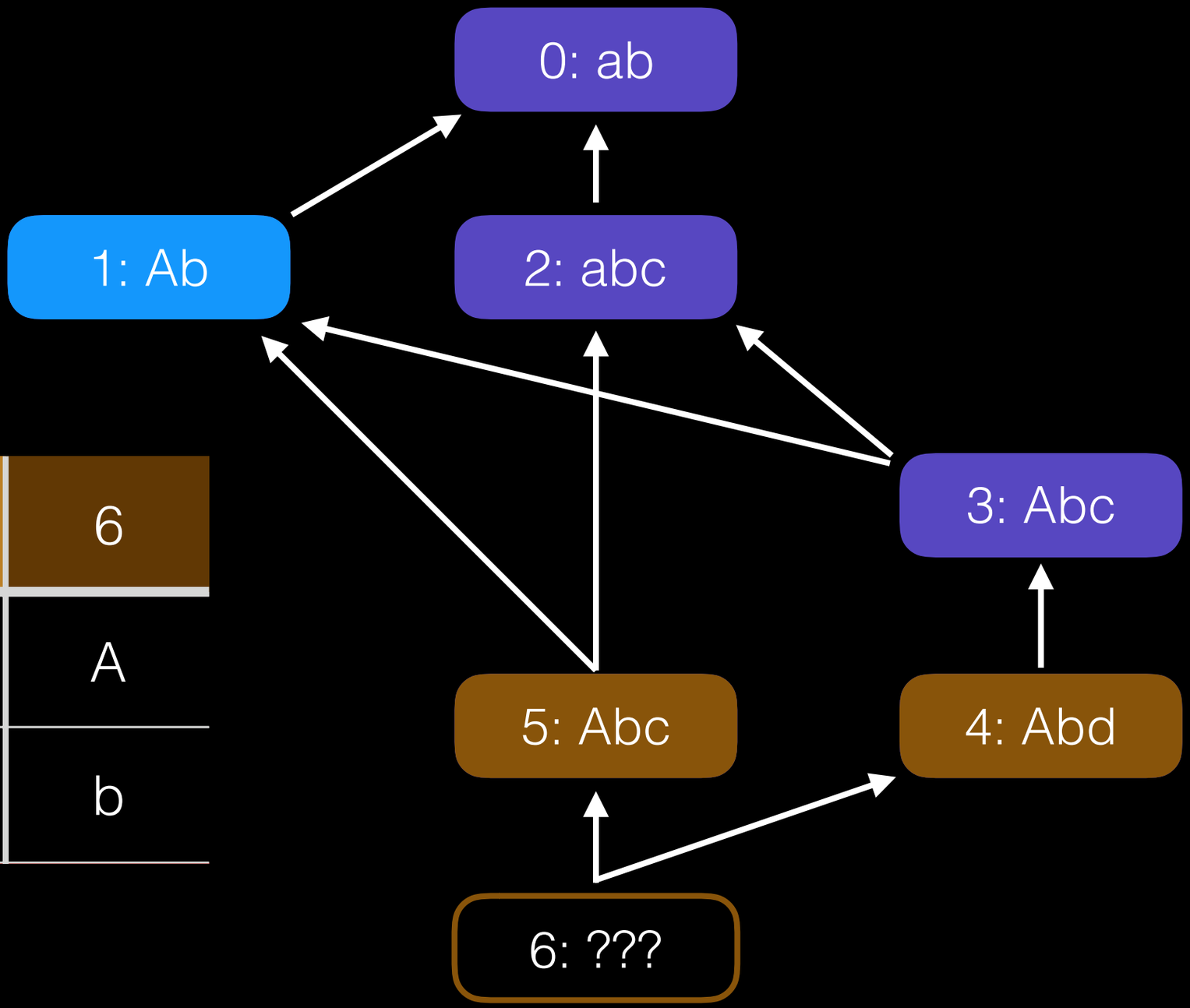


Würfeln?

1	5	4	6
A	A	A	A

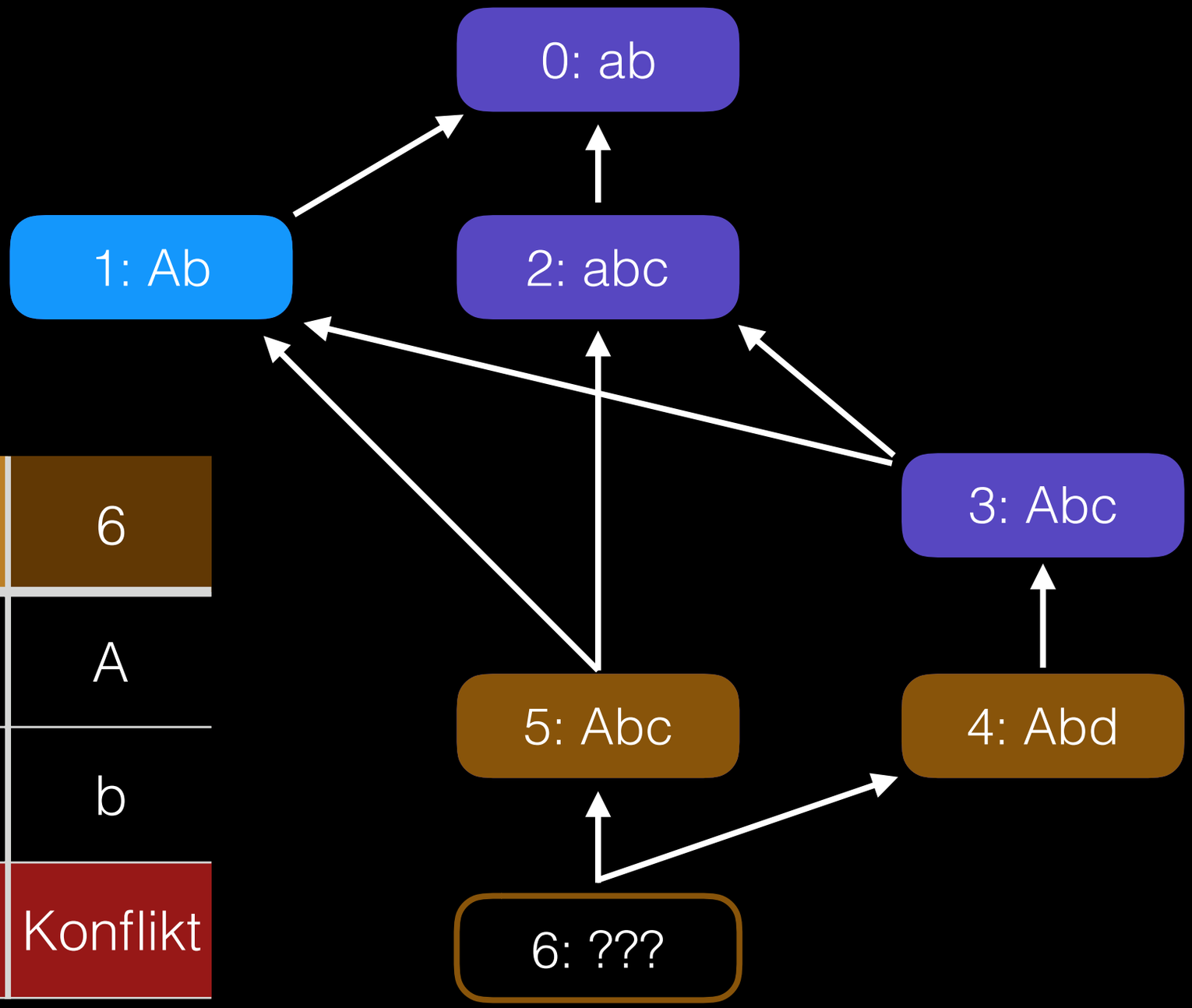


Würfeln?



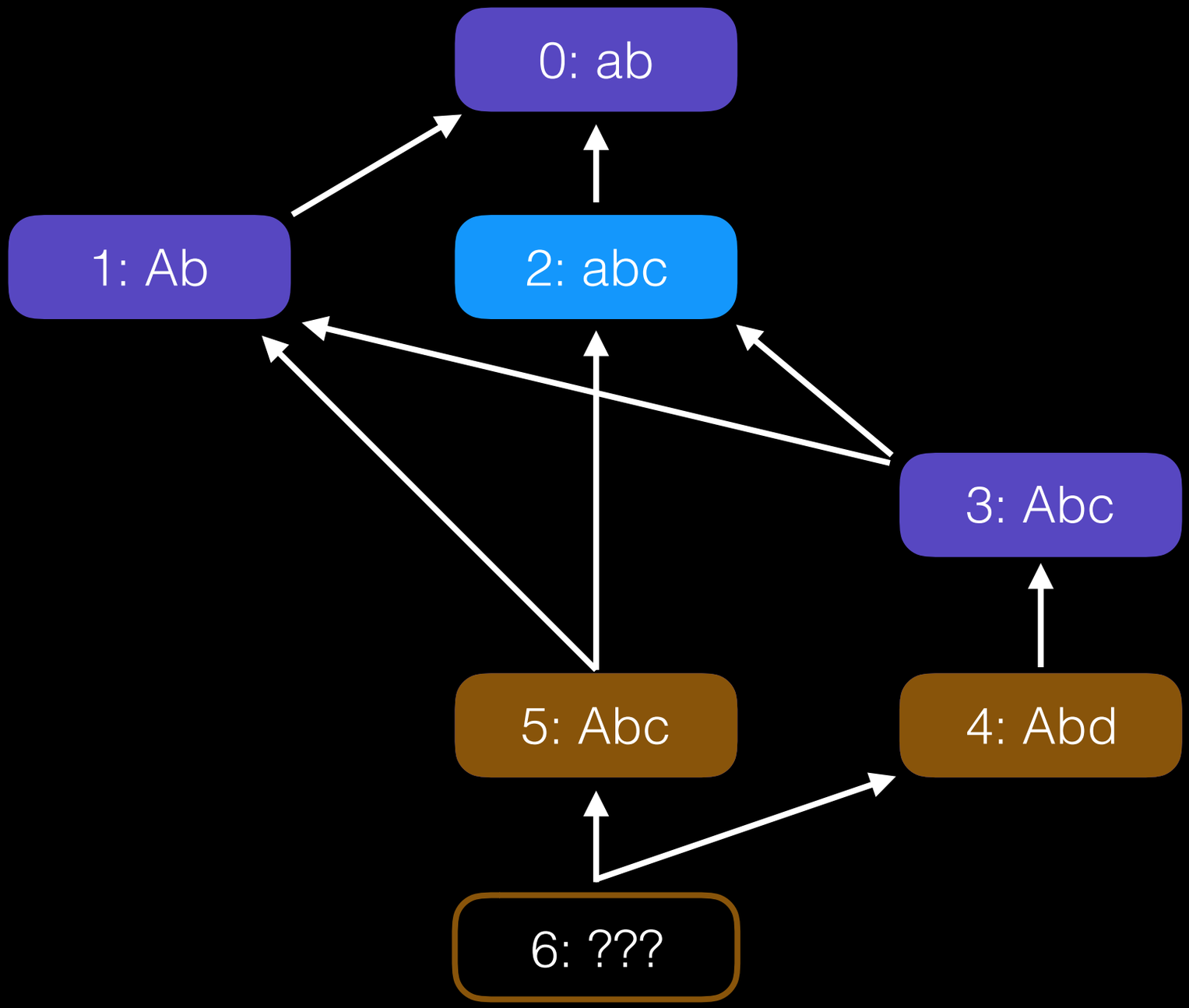
1	5	4	6
A	A	A	A
b	b	b	b

Würfeln?

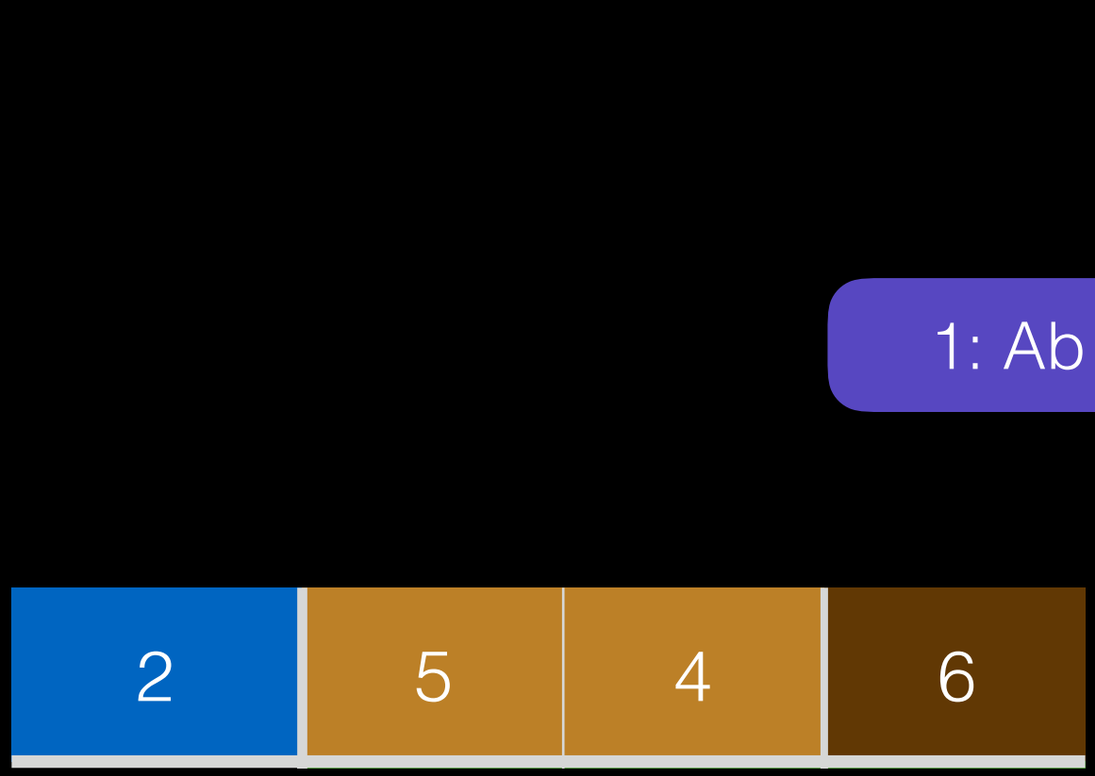


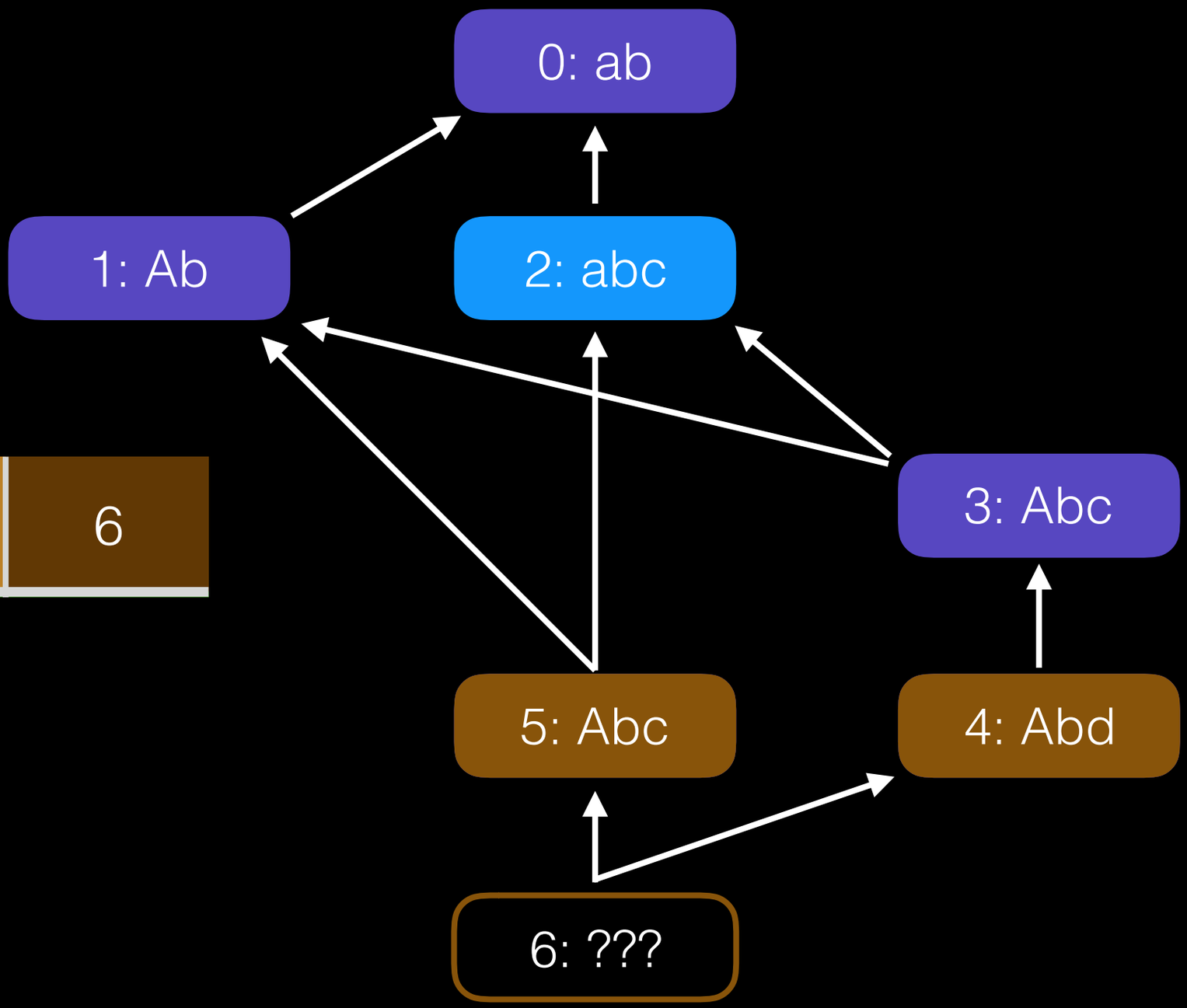
1	5	4	6
A	A	A	A
b	b	b	b
	c	d	Konflikt

Würfeln?

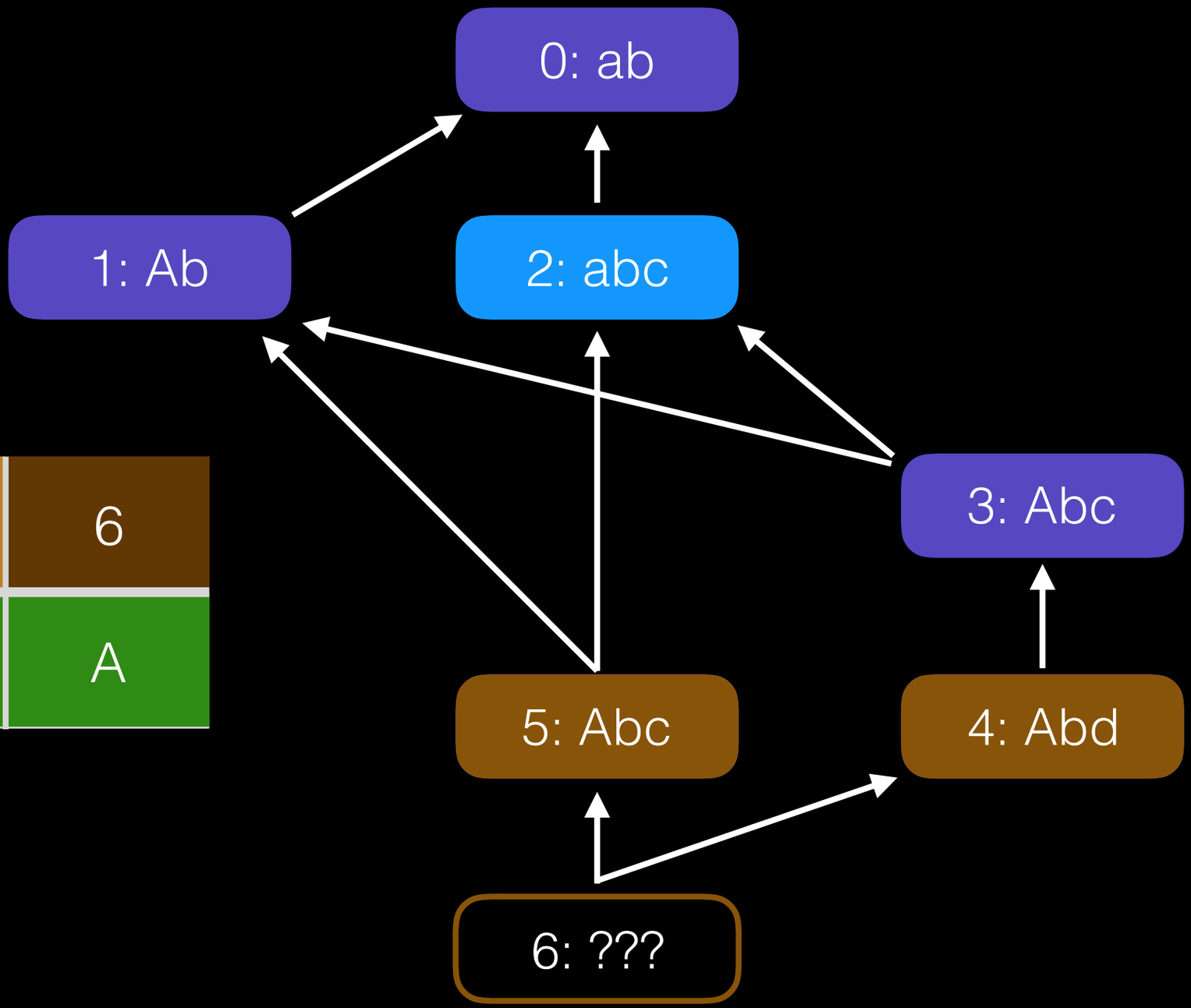


Würfeln?



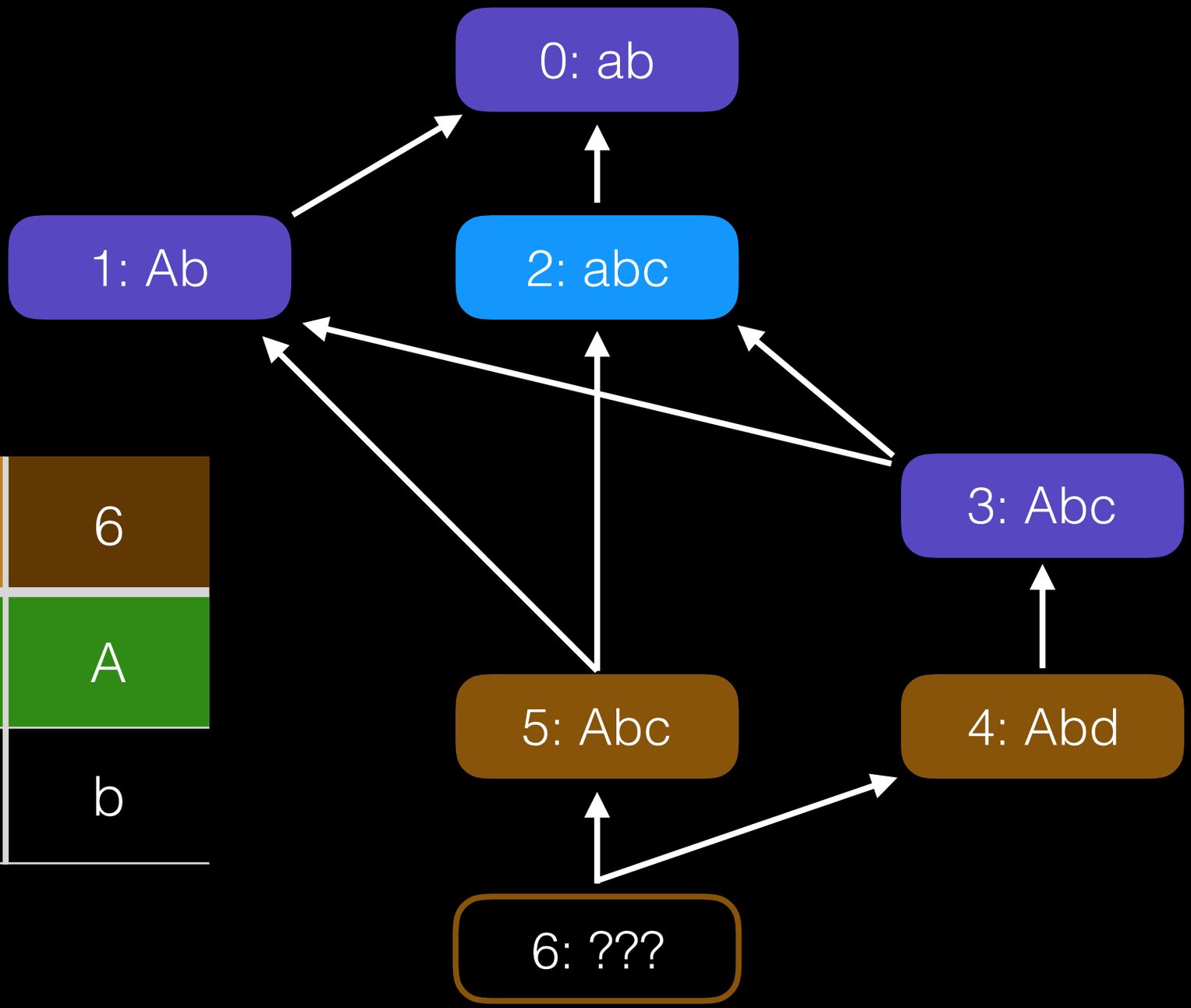


Würfeln?



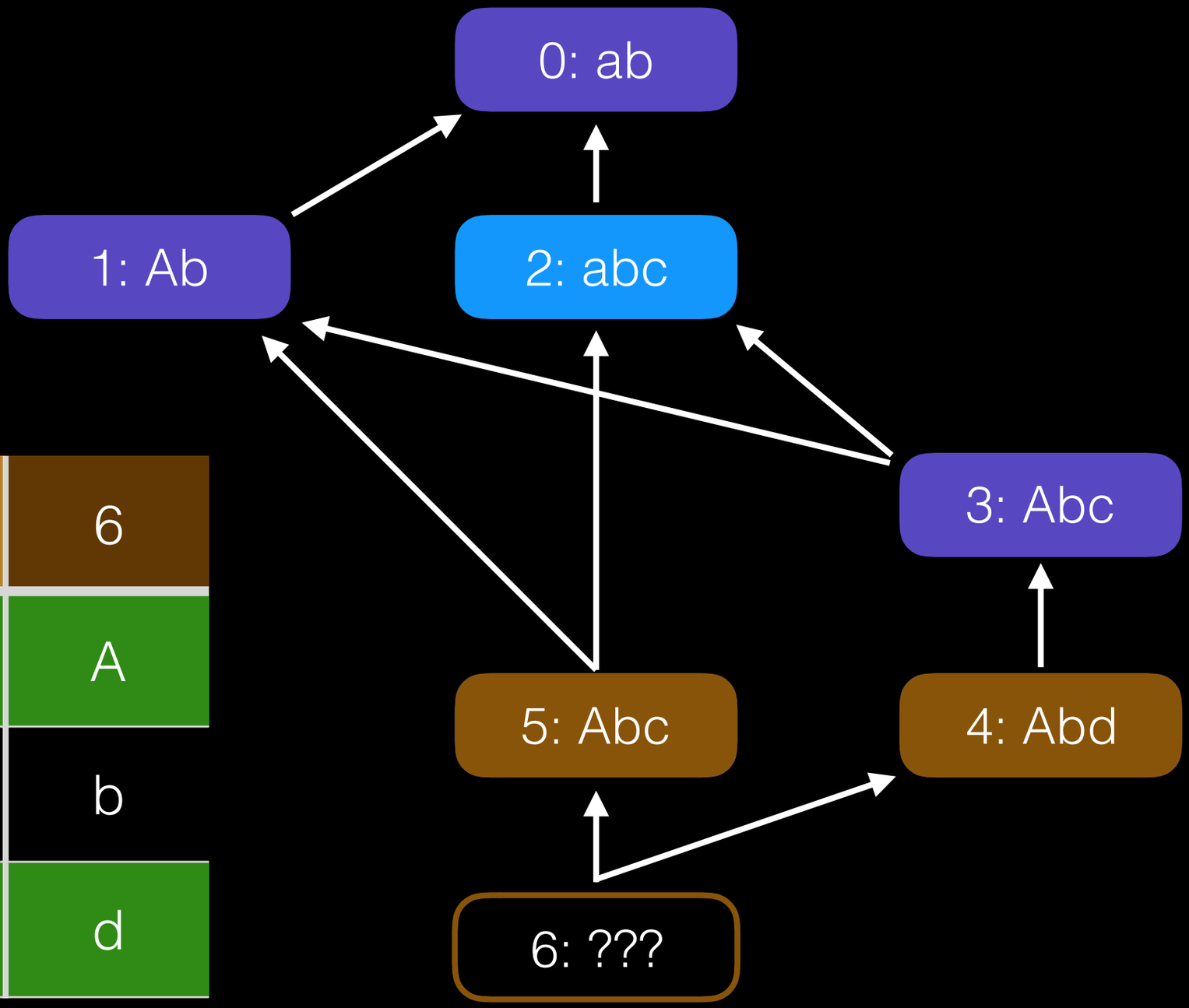
2	5	4	6
a	A	A	A

Würfeln?



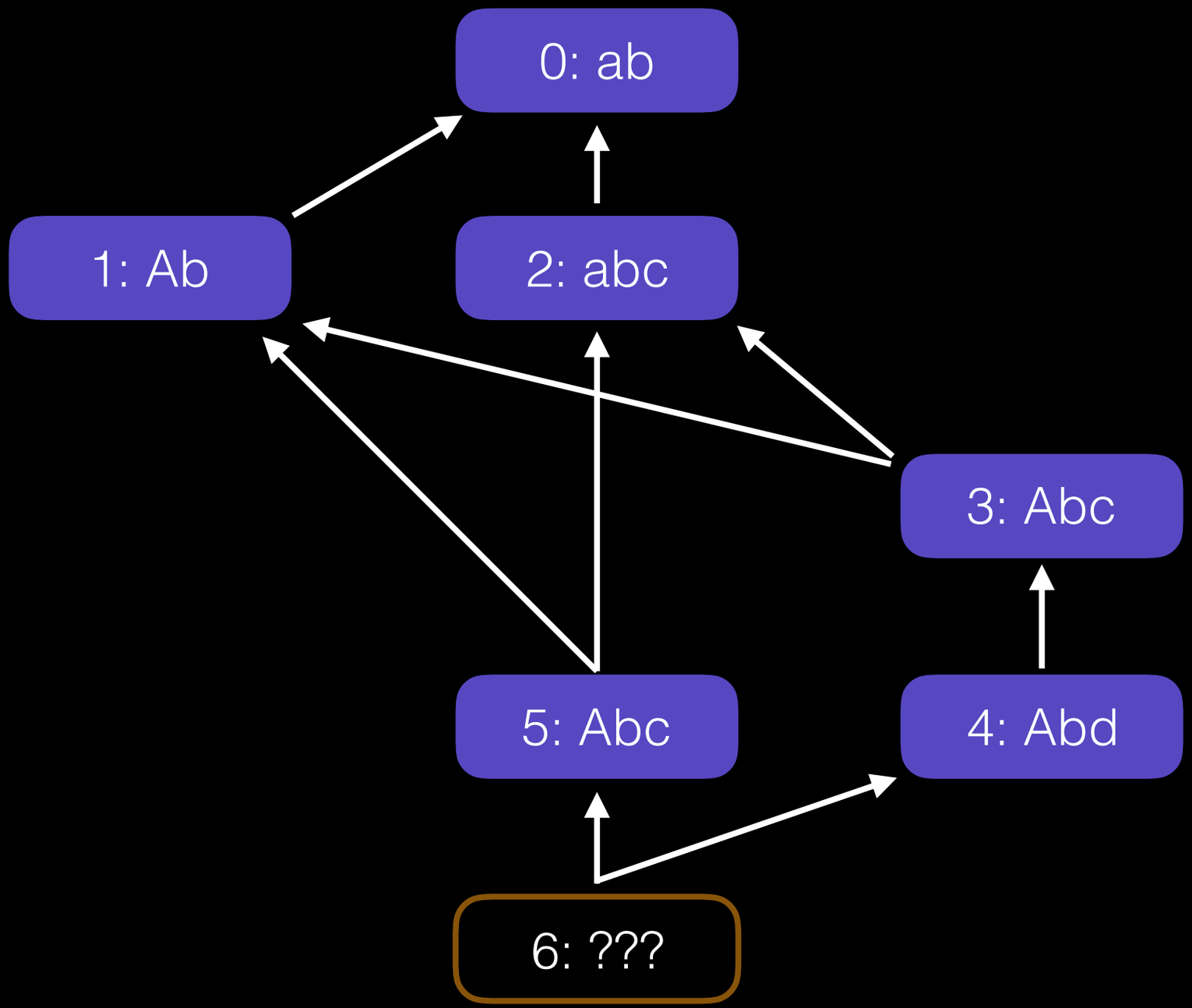
2	5	4	6
a	A	A	A
b	b	b	b

Würfeln?

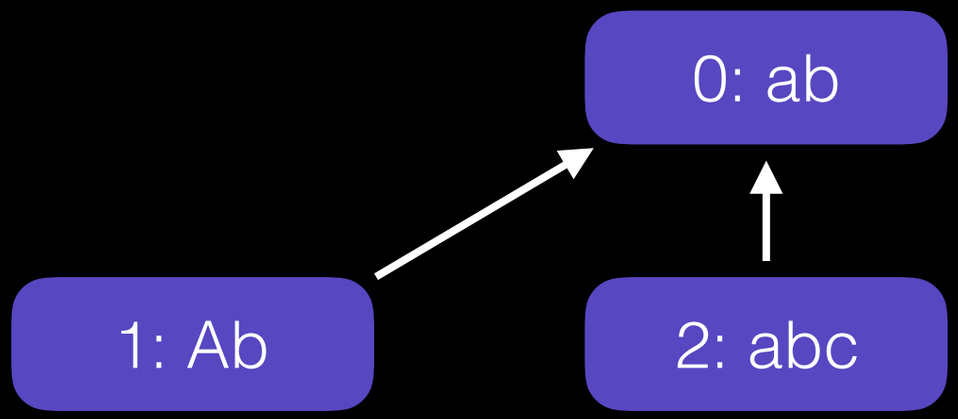


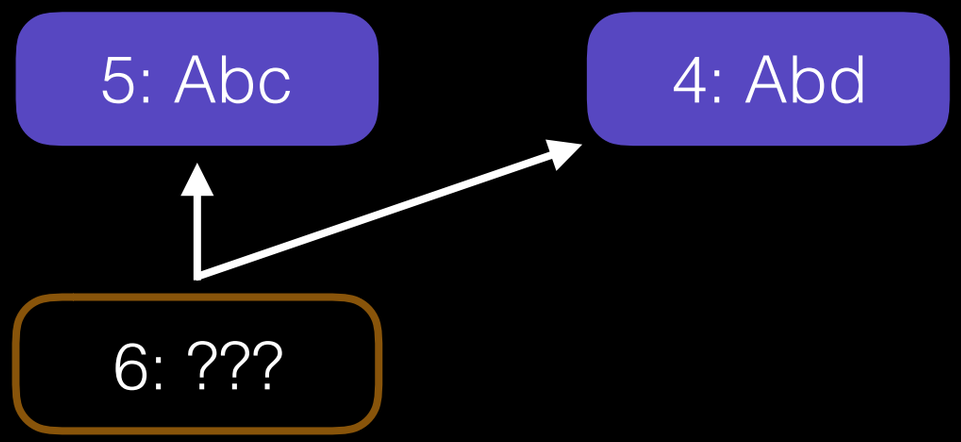
2	5	4	6
a	A	A	A
b	b	b	b
c	c	d	d

Recursive Three-Way-Merge

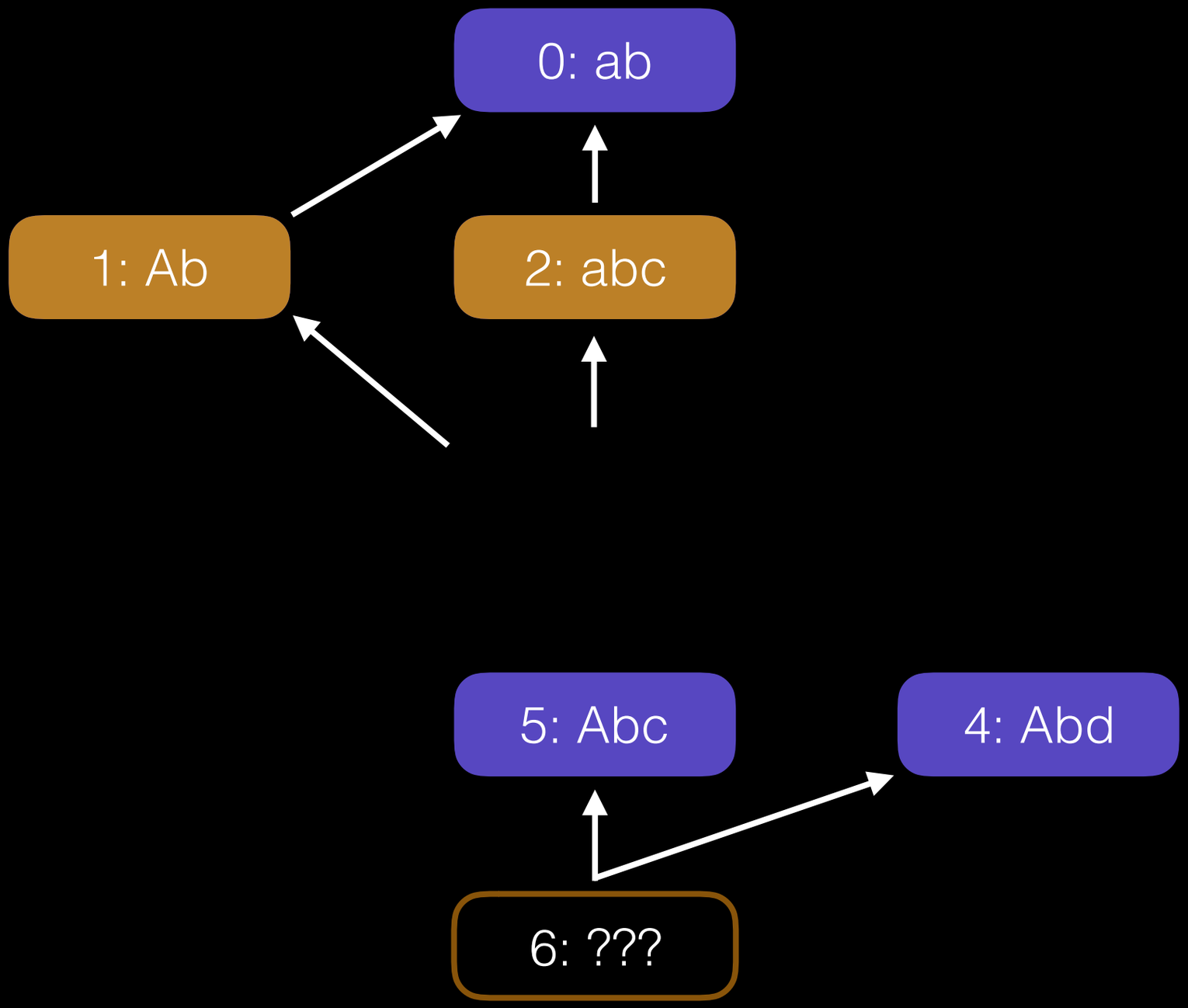


Recursive Three-Way-Merge

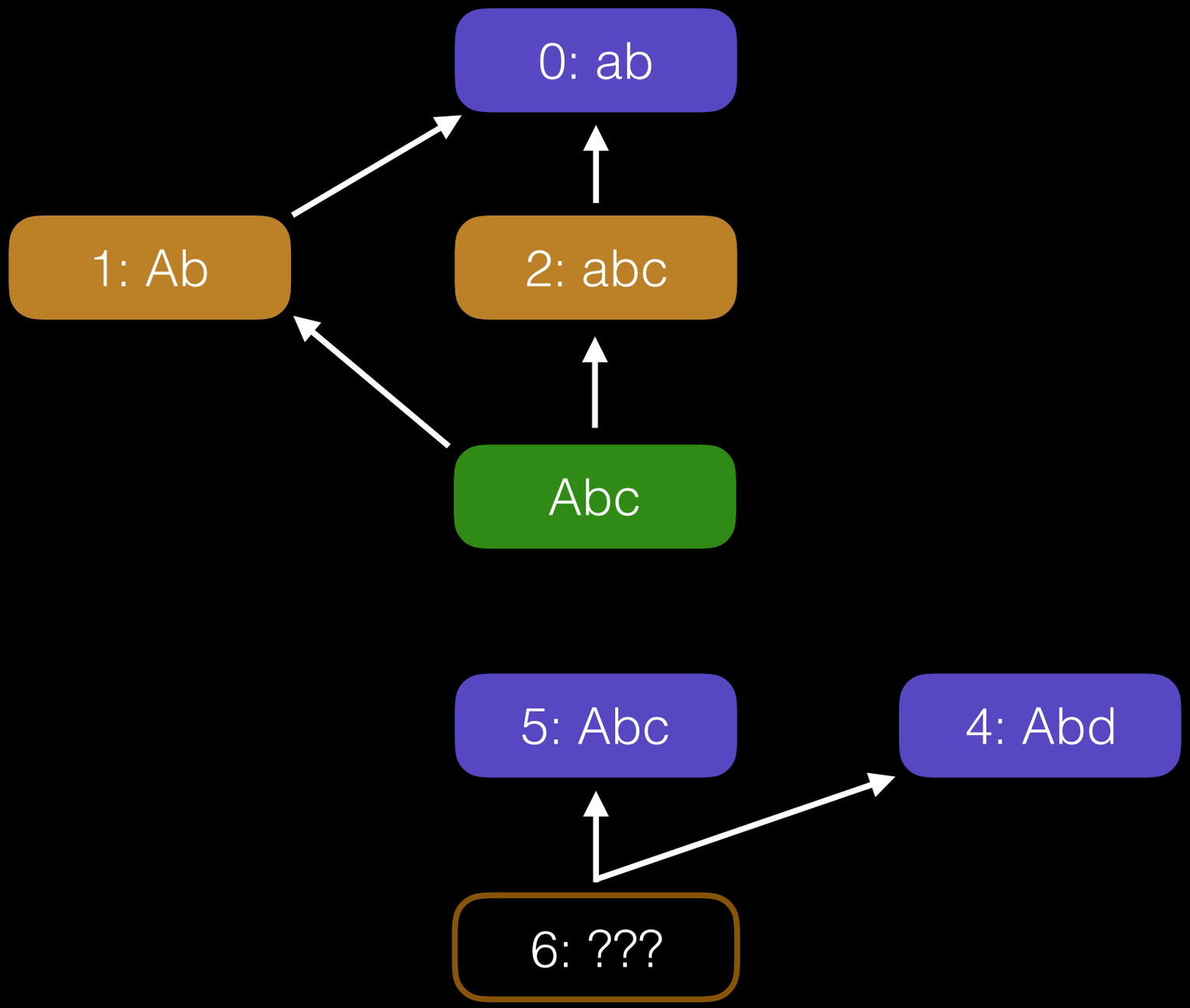




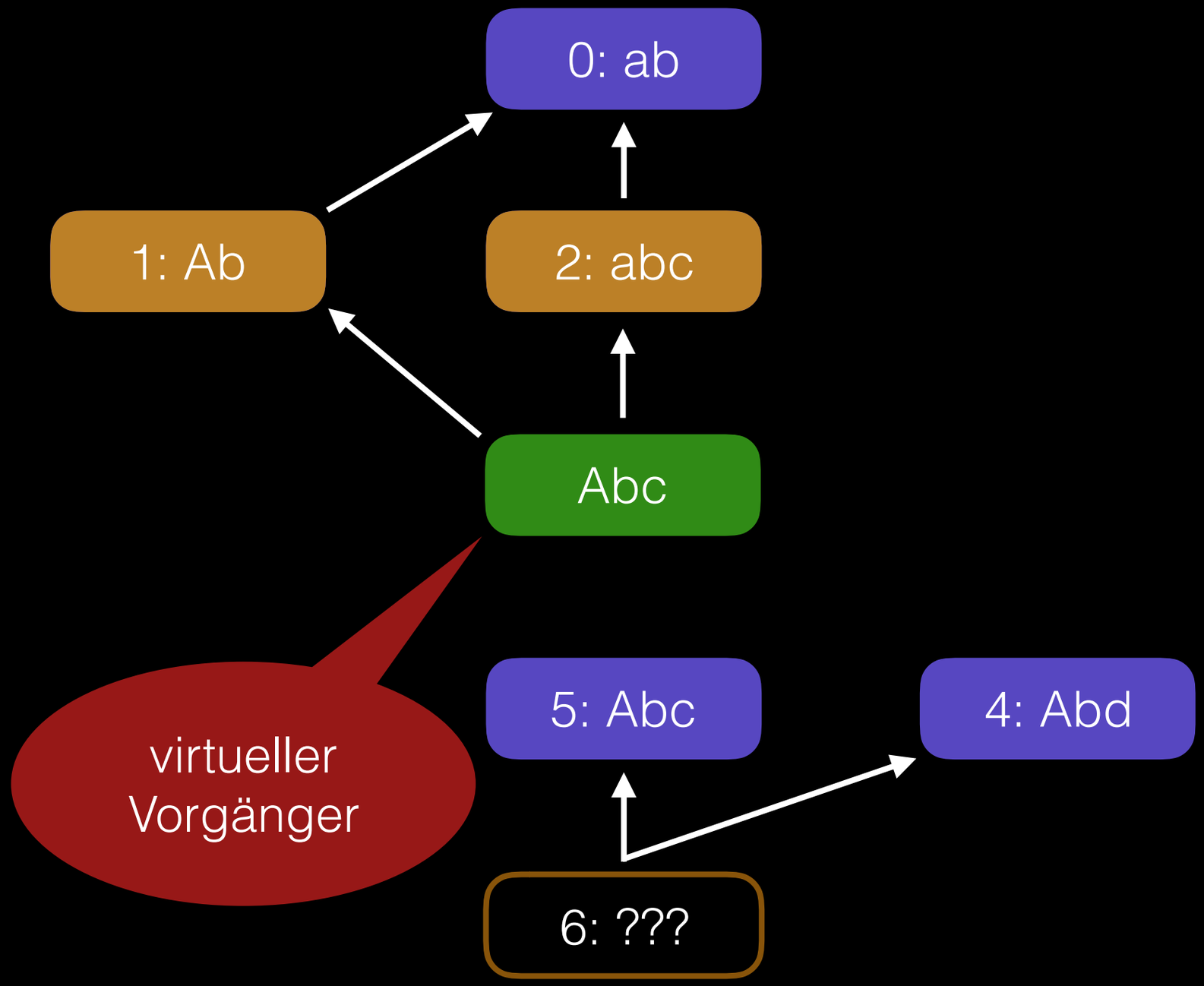
Recursive Three-Way-Merge



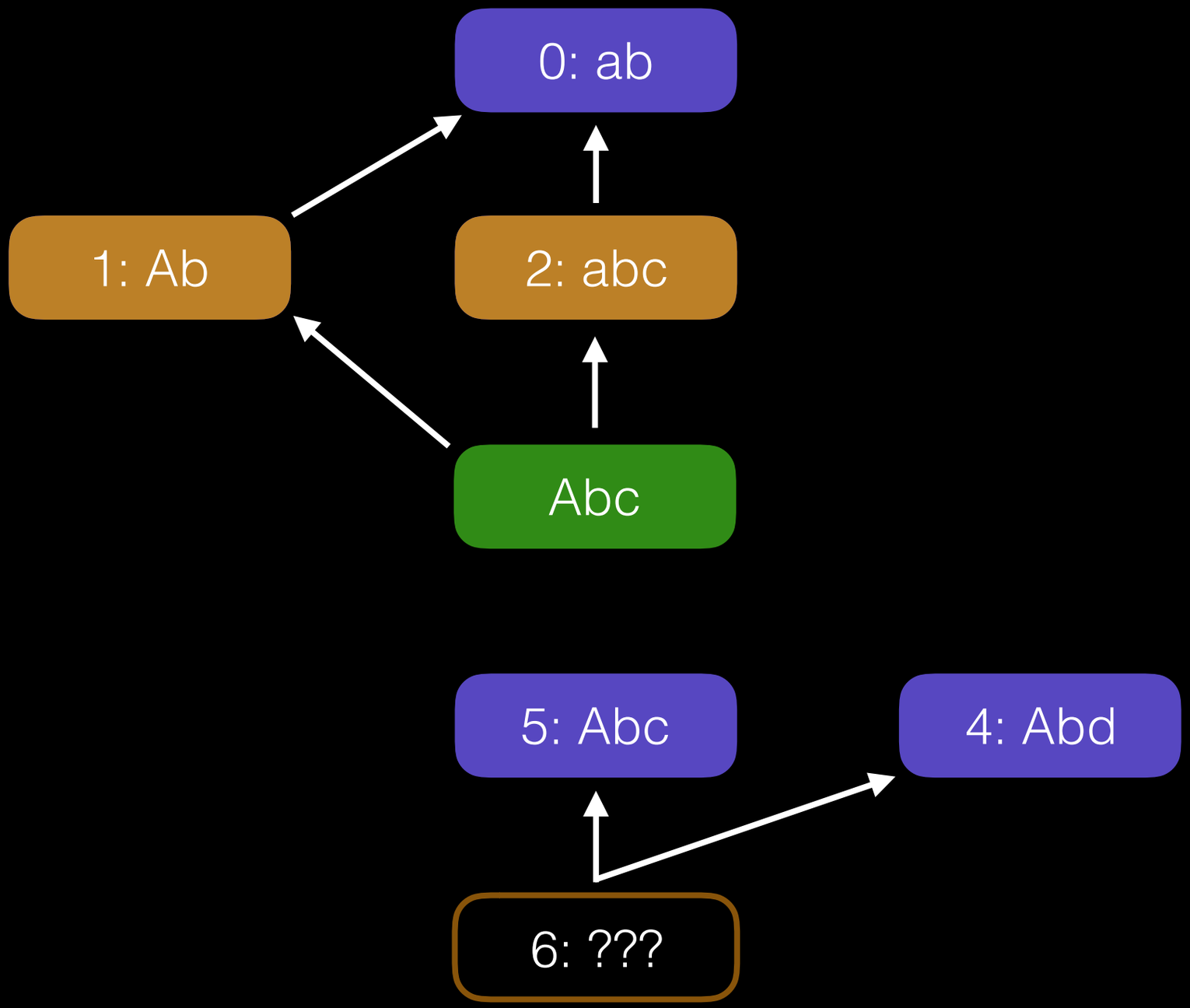
Recursive Three-Way-Merge



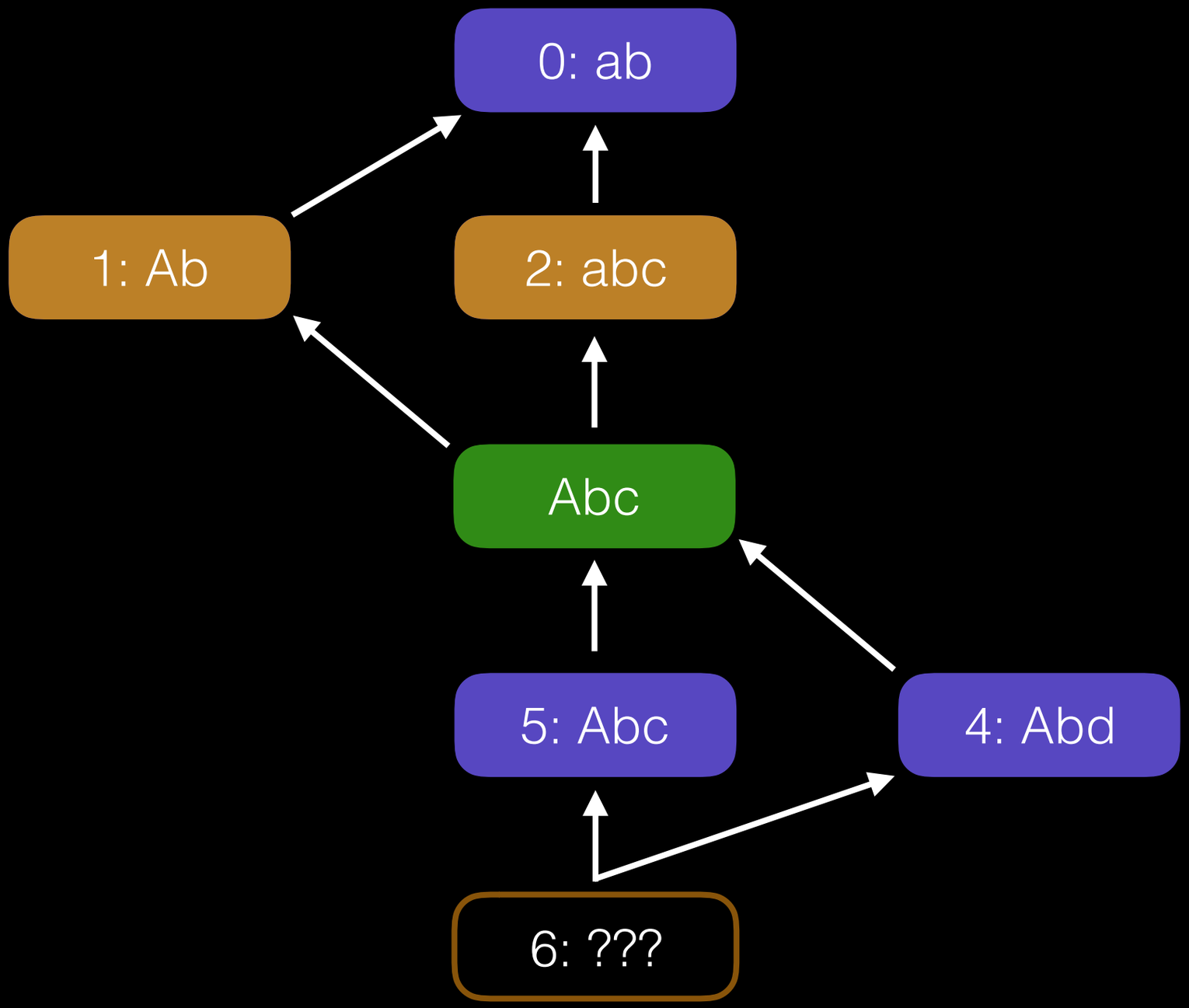
Recursive Three-Way-Merge



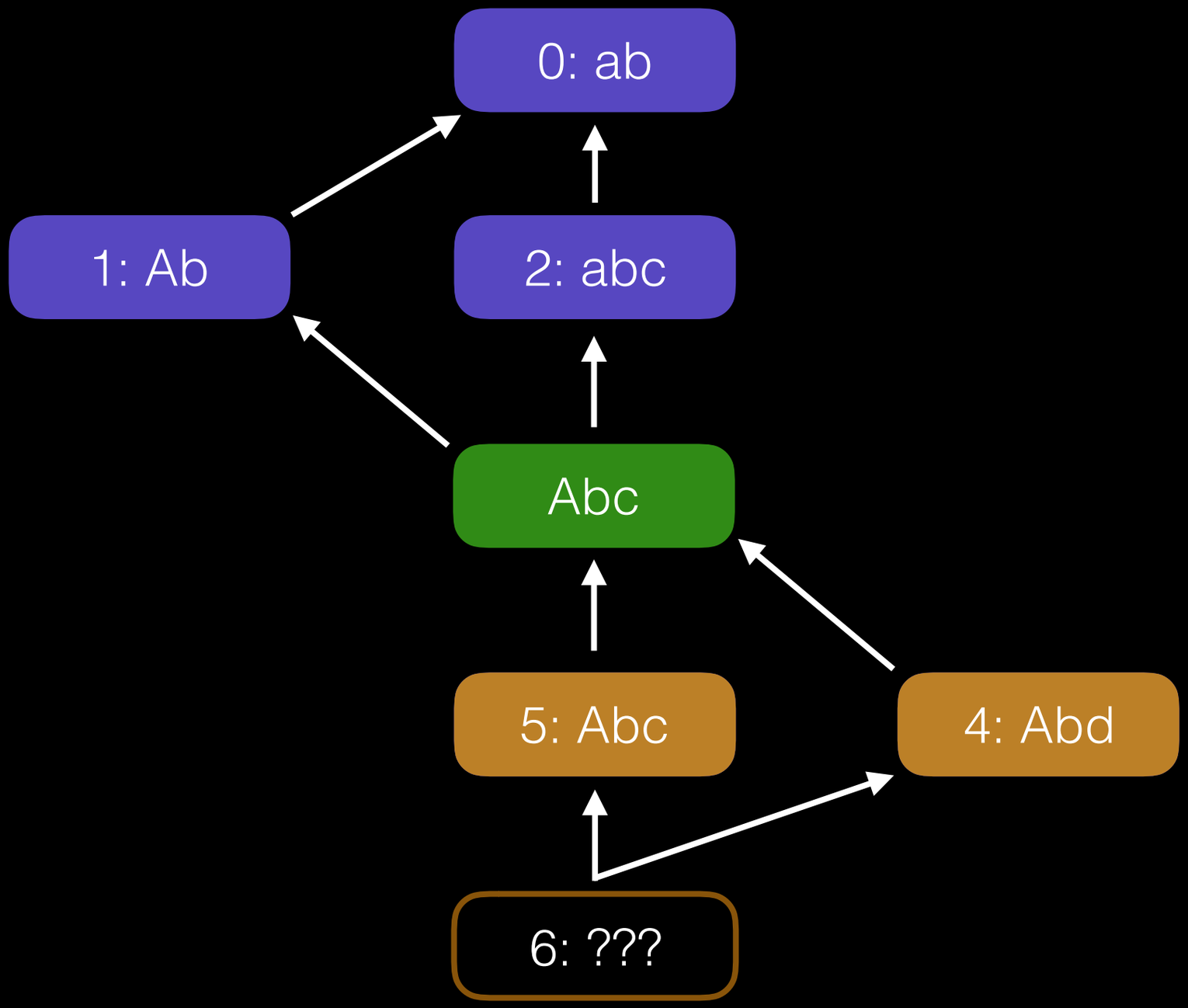
Recursive Three-Way-Merge



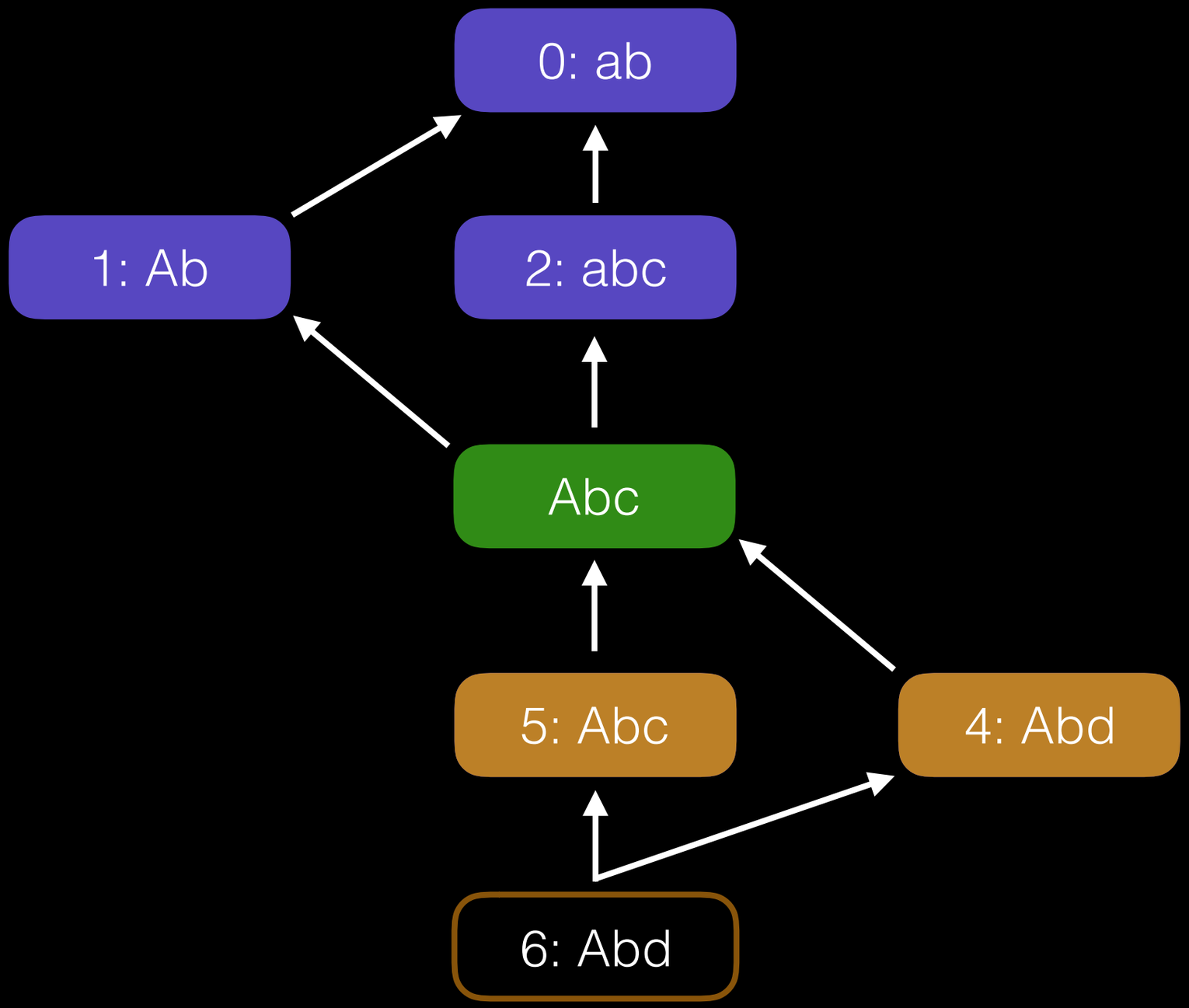
Recursive Three-Way-Merge



Recursive Three-Way-Merge



Recursive Three-Way-Merge

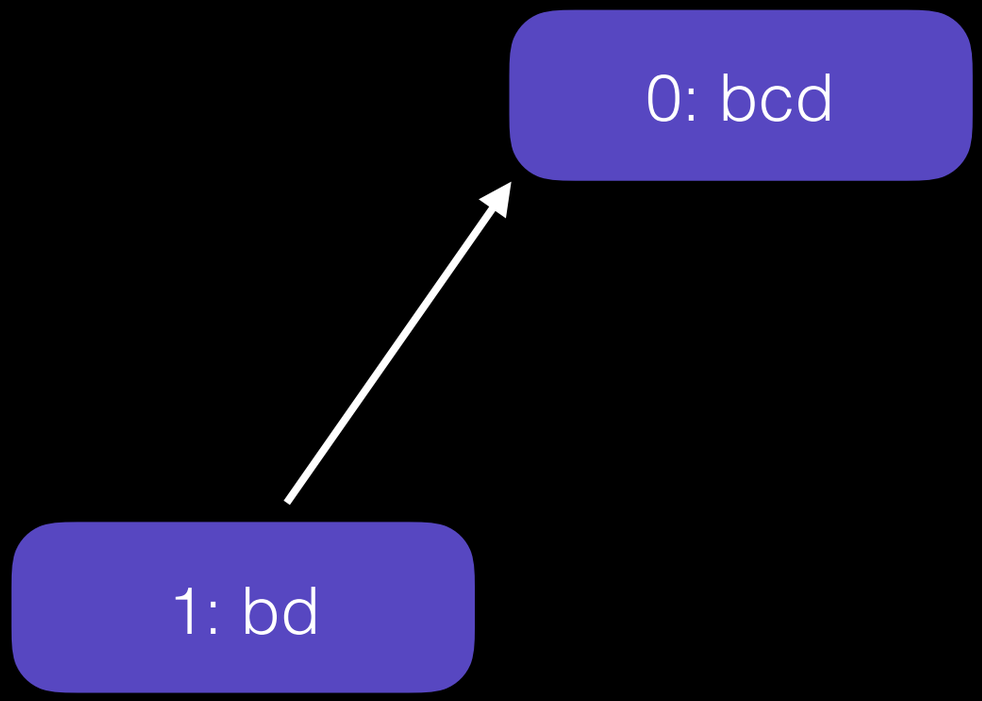


Intention erhalten?

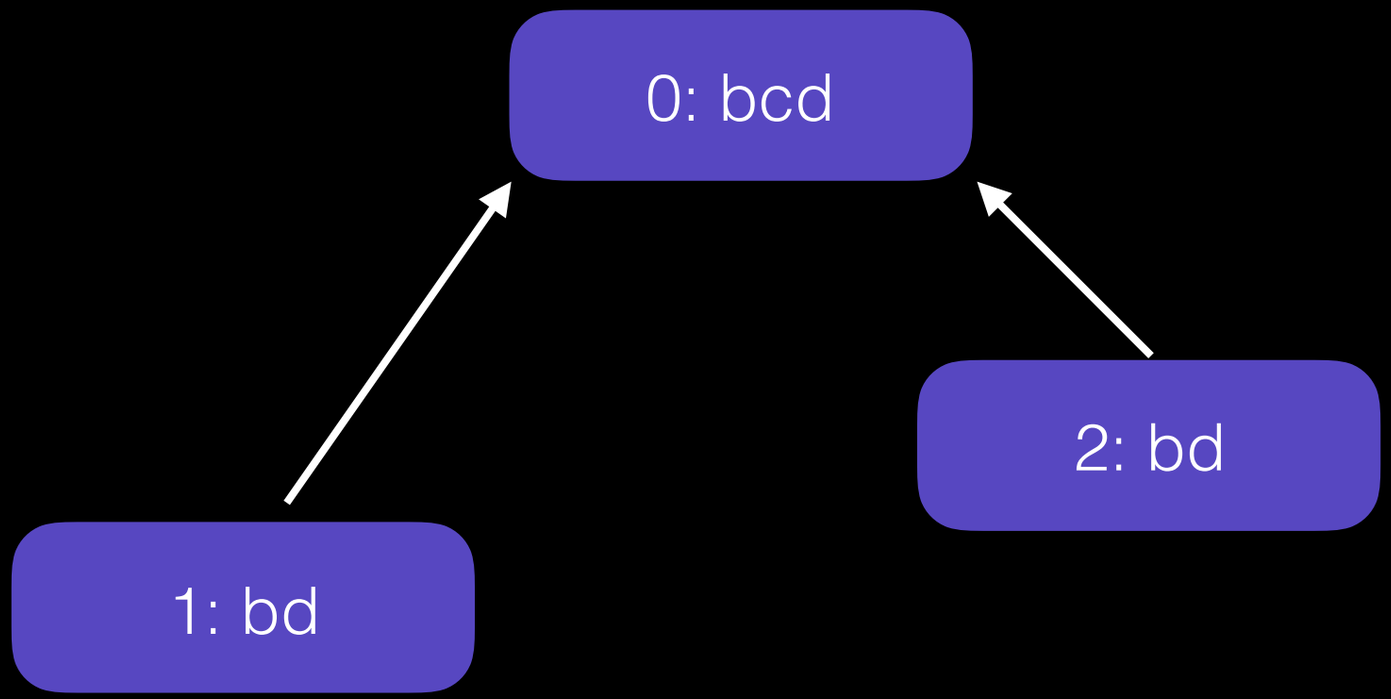
Intention erhalten?

0: bcd

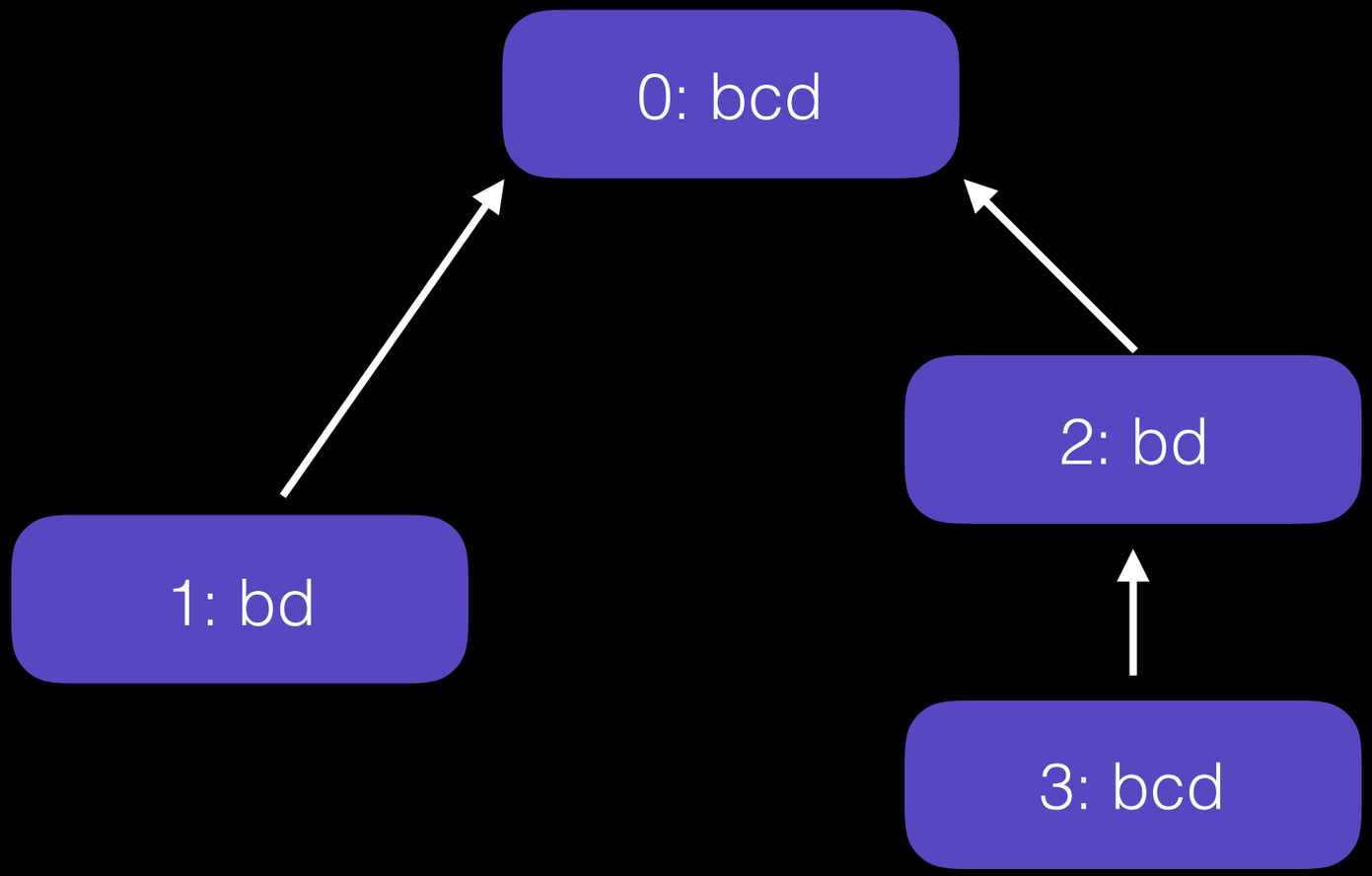
Intention erhalten?



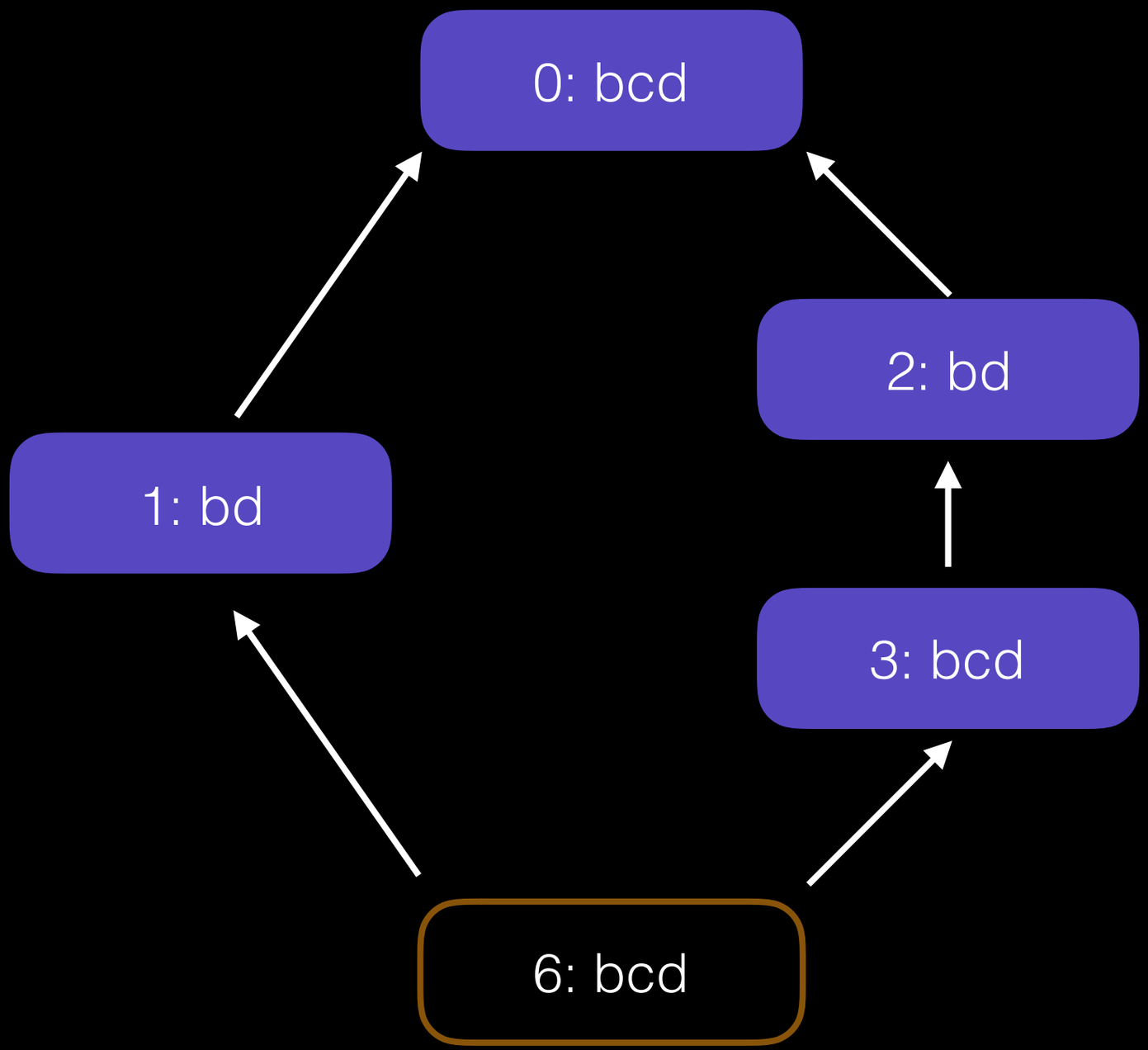
Intention erhalten?



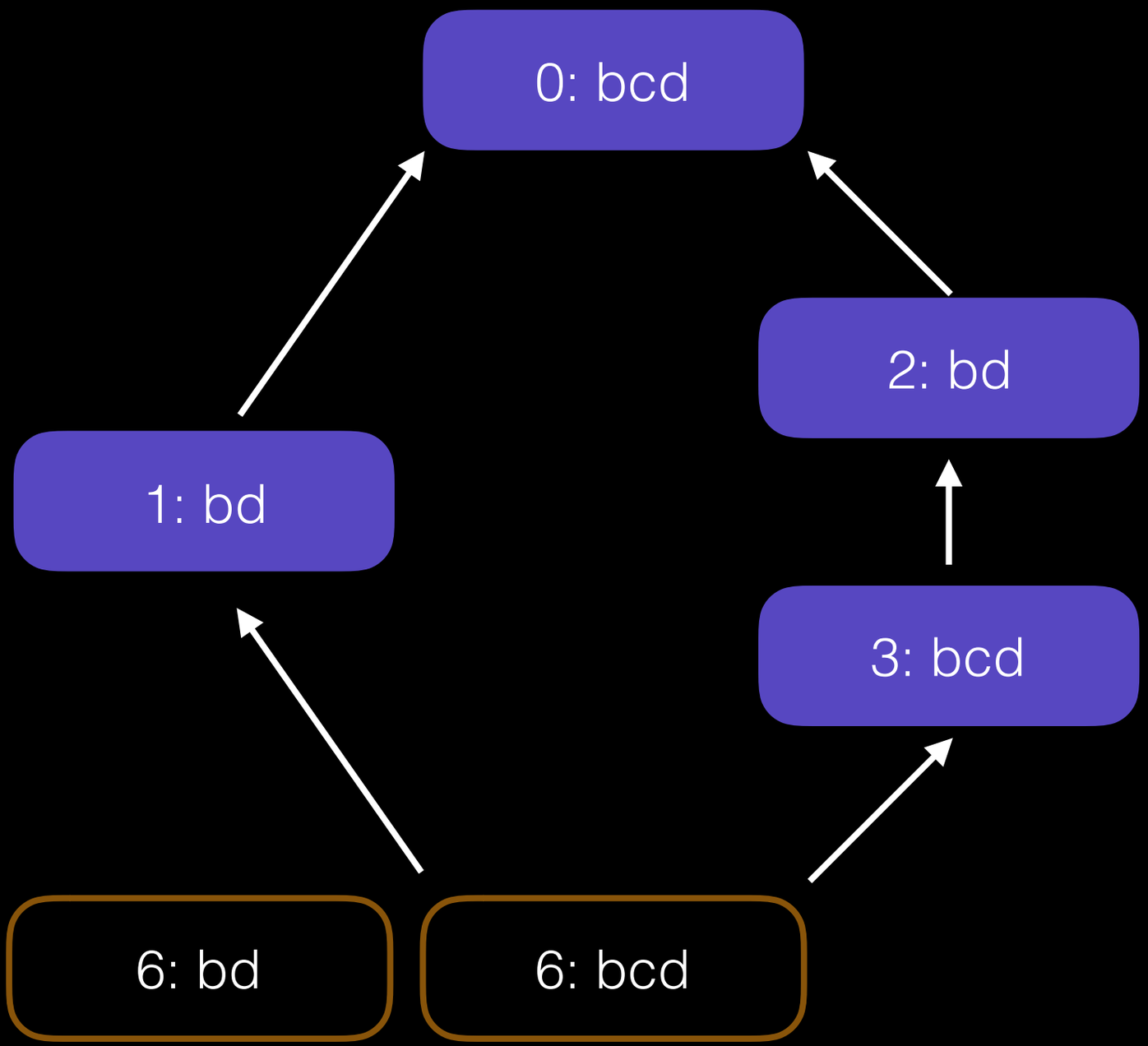
Intention erhalten?



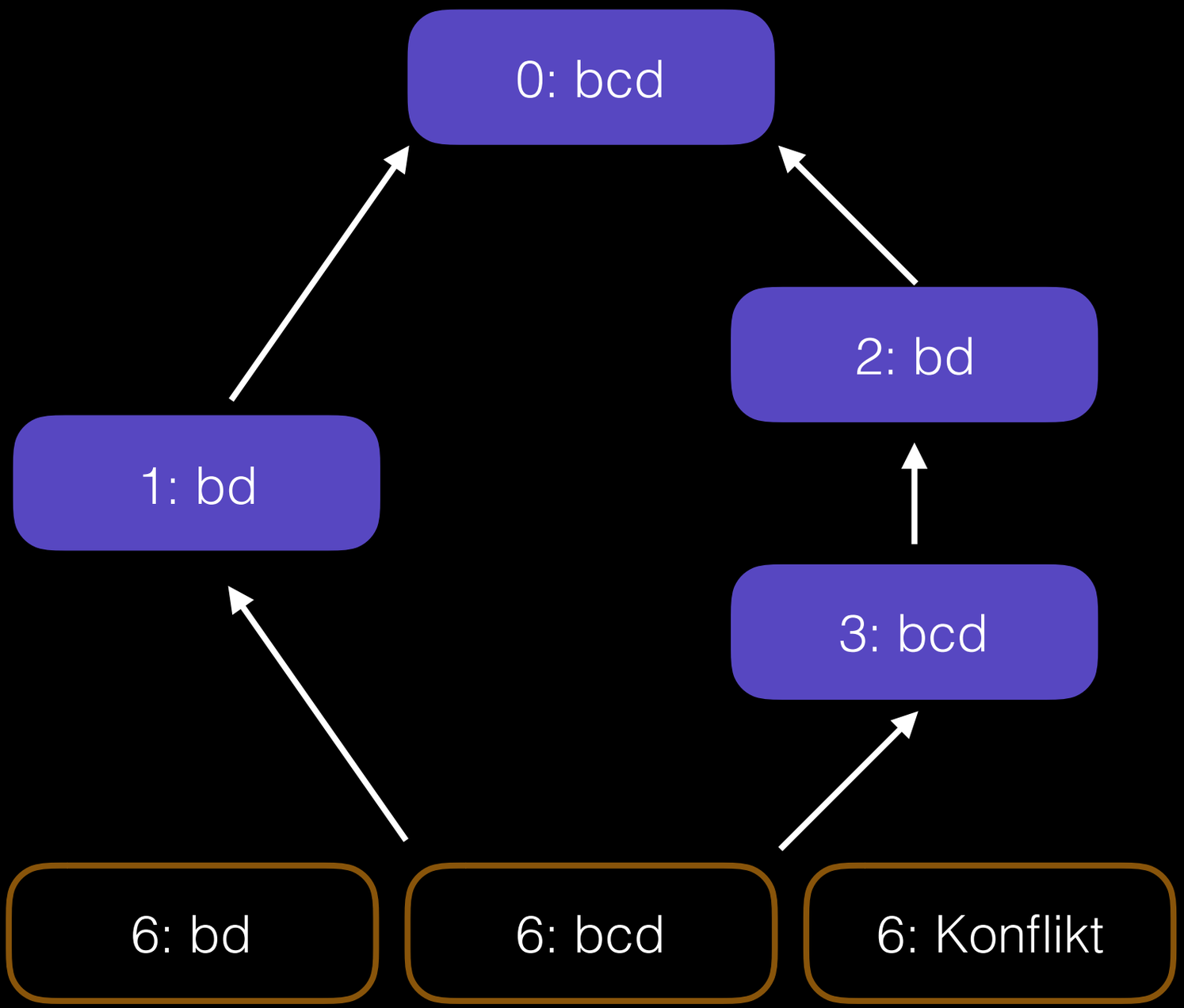
Intention erhalten?



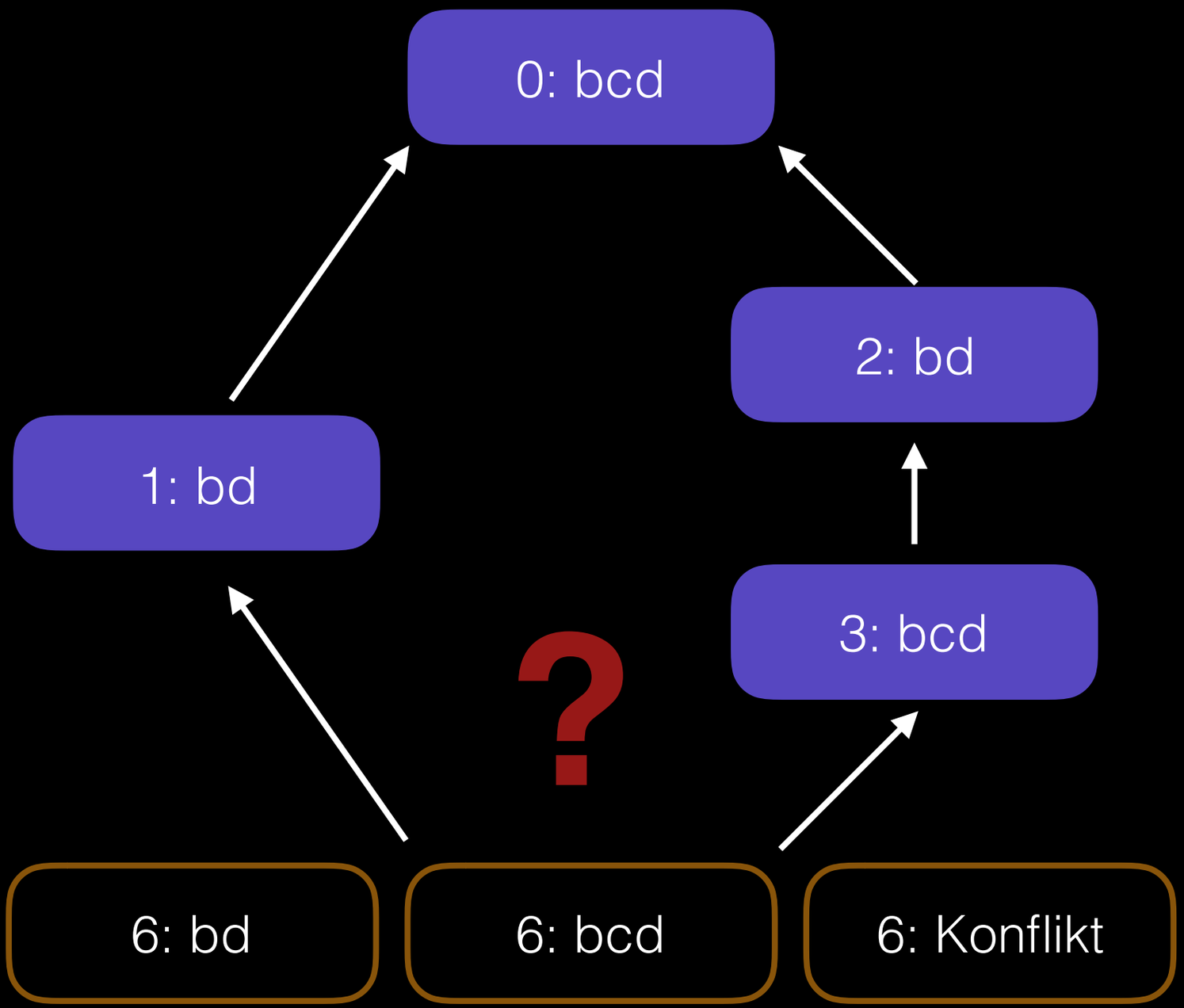
Intention erhalten?



Intention erhalten?



Intention erhalten?



Three-Way-Merge

Three-Way-Merge

- Änderungen über gemeinsamen Vorgänger

Three-Way-Merge

- Änderungen über gemeinsamen Vorgänger
- Recursive Three-Way-Merge:
  Verbesserte Automatik über virtuellen Vorgänger

Three-Way-Merge

- Änderungen über gemeinsamen Vorgänger
- Recursive Three-Way-Merge:
  Verbesserte Automatik über virtuellen Vorgänger
- Nicht konfliktfrei / vollautomatisch
  (→ CRDTs, Operational Transformations, …)

Three-Way-Merge

- Änderungen über gemeinsamen Vorgänger
- Recursive Three-Way-Merge:
  Verbesserte Automatik über virtuellen Vorgänger
- Nicht konfliktfrei / vollautomatisch
  (→ CRDTs, Operational Transformations, …)
- Nur Text
  Keine Sets, Trees, …

Git

Git

- Verteilte Versionsverwaltung

Git

- Verteilte Versionsverwaltung
- Jeder Commit: Kompletter Zustand

Git

- Verteilte Versionsverwaltung
- Jeder Commit: Kompletter Zustand
- Commits, BLOBs, Trees über SHA-1 adressiert

Git

- Verteilte Versionsverwaltung
- Jeder Commit: Kompletter Zustand
- Commits, BLOBs, Trees über SHA-1 adressiert
- Merging mittels neuer Commits

Git

- Verteilte Versionsverwaltung
- Jeder Commit: Kompletter Zustand
- Commits, BLOBs, Trees über SHA-1 adressiert
- Merging mittels neuer Commits
- Recursive Three-Way-Merge reduziert Konflikte

Quellen

- Git Internals – Plumbing & Porcelain
  https://git-scm.com/book/en/v2/Git-Internals-Plumbing-and-Porcelain

- More on Recursive Merge Strategy
  http://blog.plasticscm.com/2012/01/more-on-recursive-merge-strategy.html