Verifizierte Implementierung einer Mapping-Datenbank in Coq

besondere Lernleistung 2016

Benno Fünfstück

12. Mai 2016

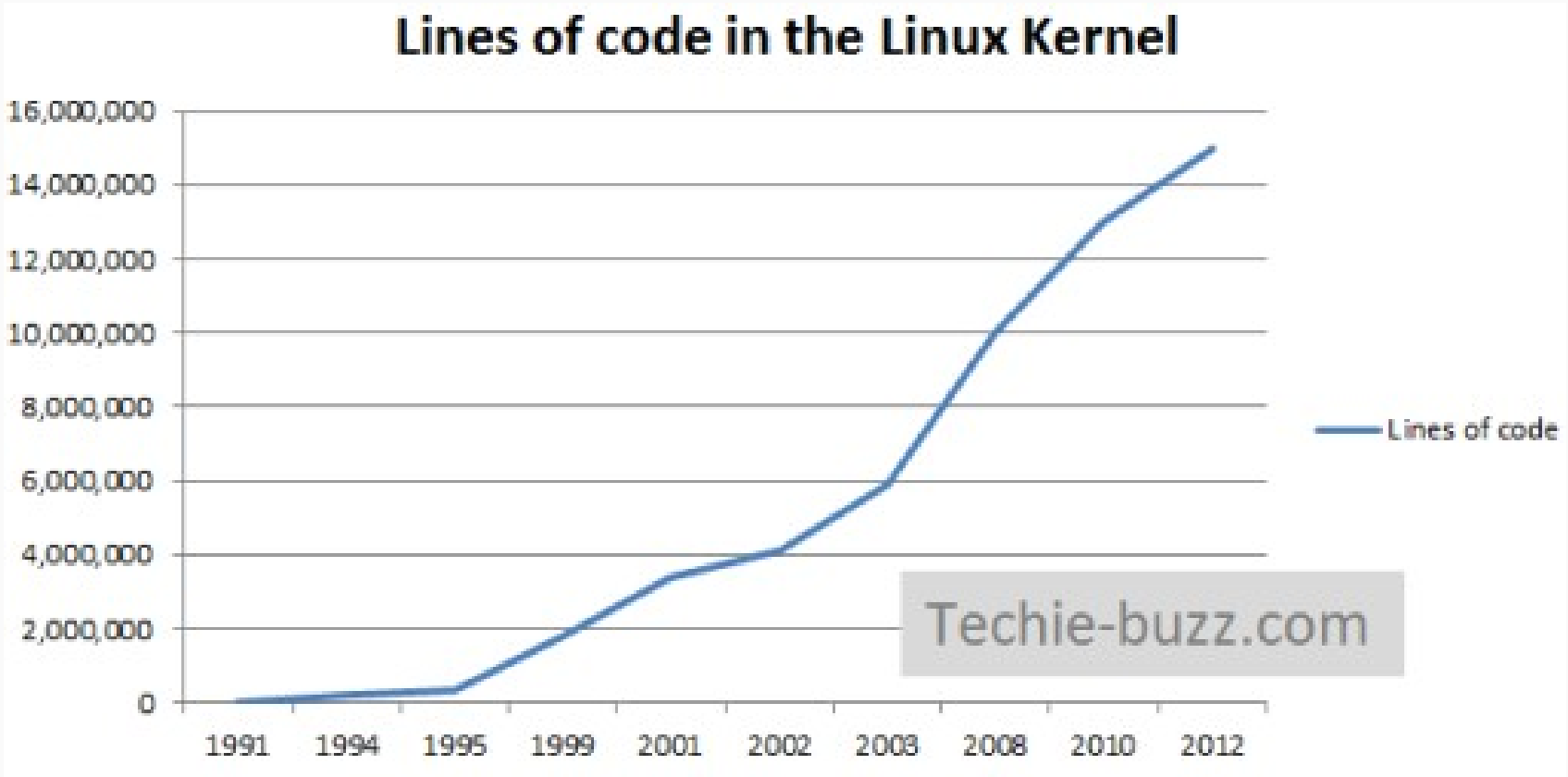
Betreuer: Dr. Hendrik Tews & Dr. Thomas Türk
FireEye Technologie Deutschland GmbH

Gliederung

1. Microkern und Mapping-Datenbank
2. Implementierung
3. Verifikation
4. Ergebnisse und Ausblick

Microkern und Mapping-Datenbank

Entwicklung des Linux-Kerns

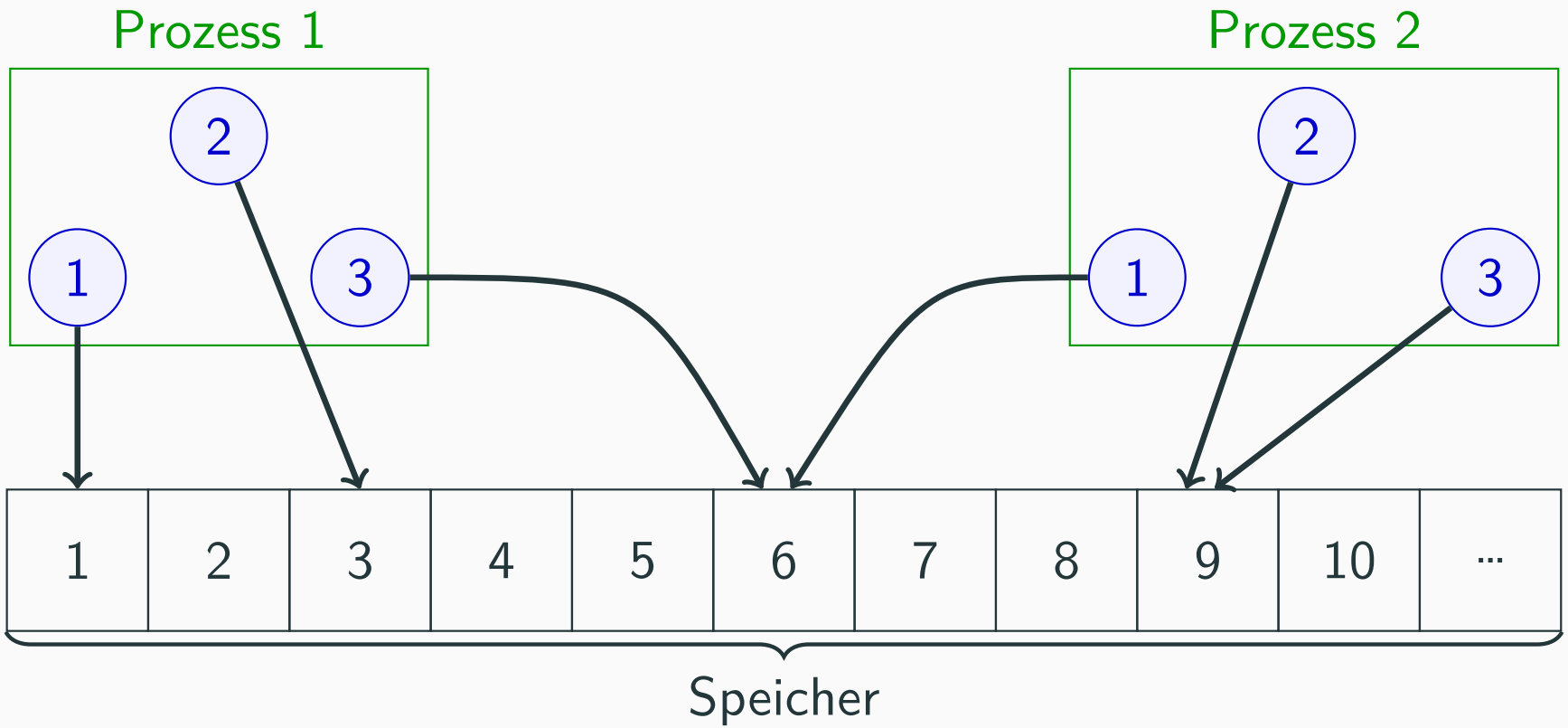


Quelle: http://cache.techie-buzz.com/images4/chinmoy/linux-kernel-rise.jpg

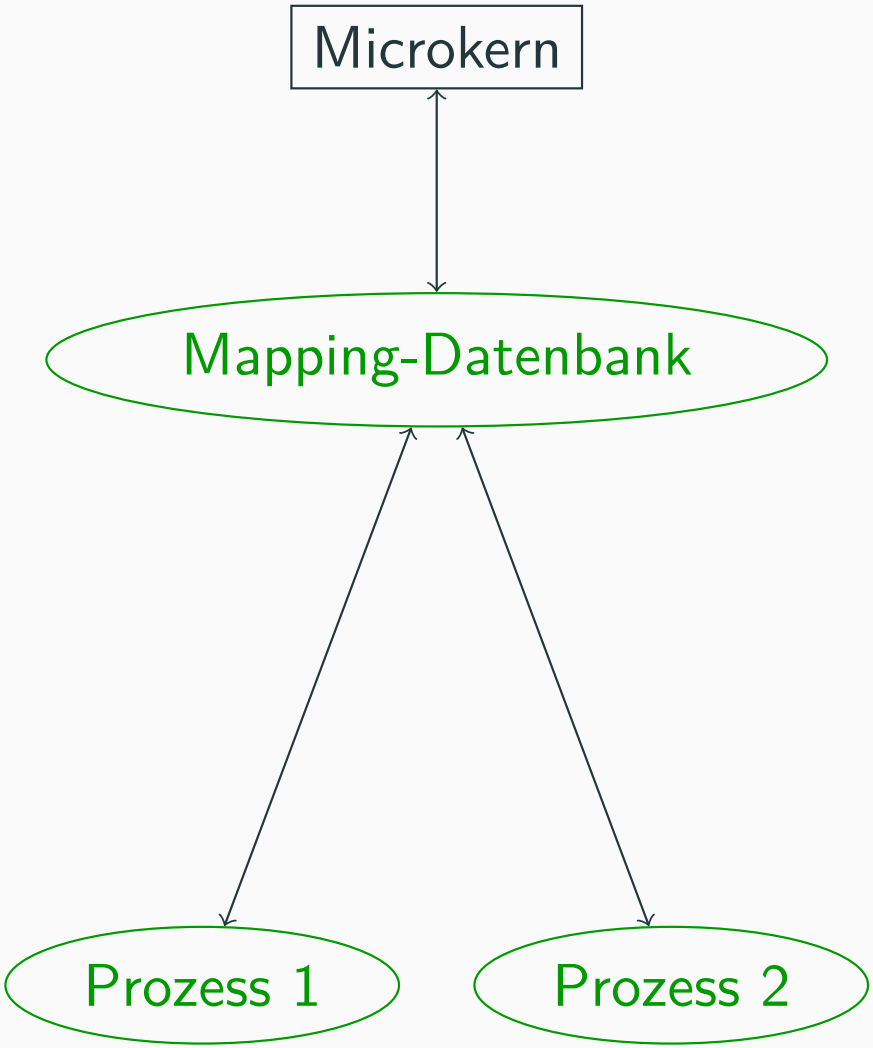
Lösungsansatz: Microkern

- Nur wesentliche Funktionen im Kern
- NOVA-Microkern: ca. 10000 Zeilen Quelltext
- weniger Fehler im Kern

Capabilities



Mapping-Datenbank

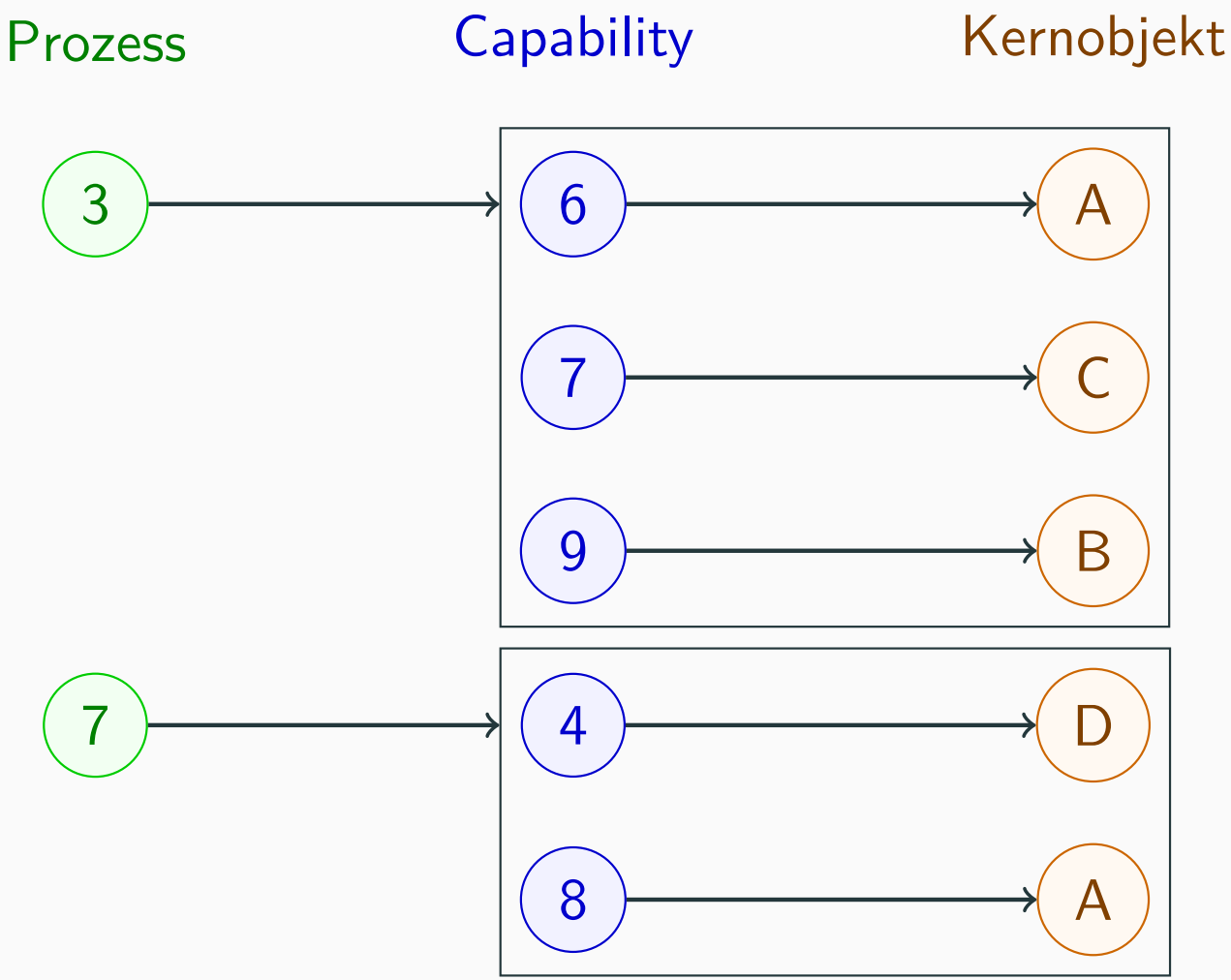


Implementierung

Operationen der Mapping-Datenbank

- Eintrag anlegen
- Eintrag entfernen
- Zugriff auf ein bestimmtes Kernobjekt entziehen

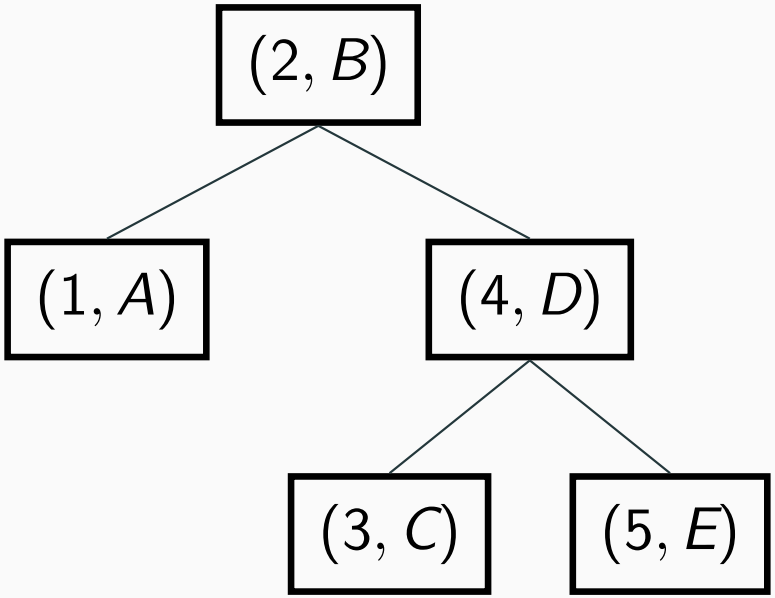
Datenstruktur der Mapping-Datenbank



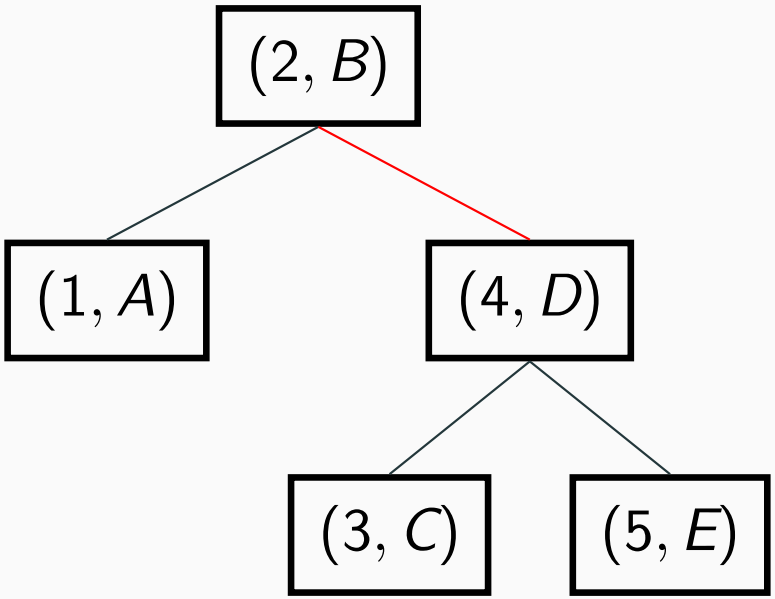
Darstellung der Zuordnung

Capability	Kernobjekt
2	B
5	E
1	A
4	D
3	C
...	...

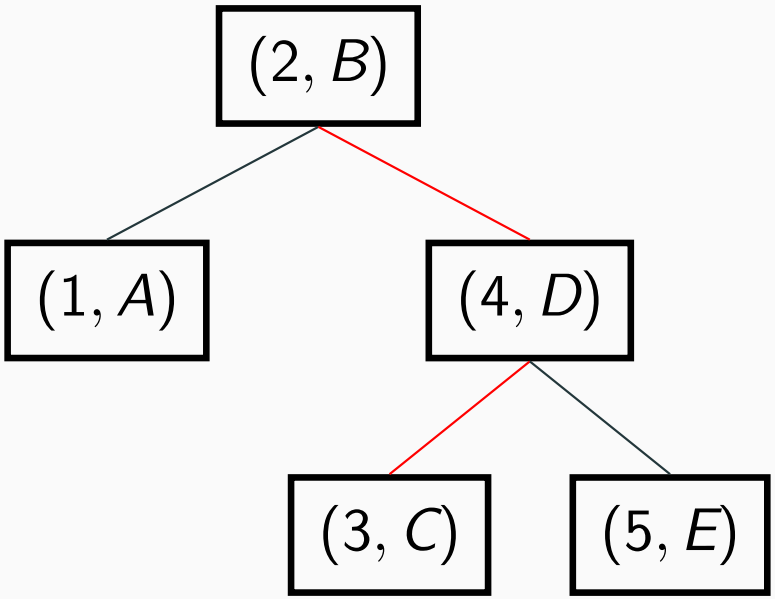
Binärer Baum



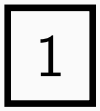
Binärer Baum



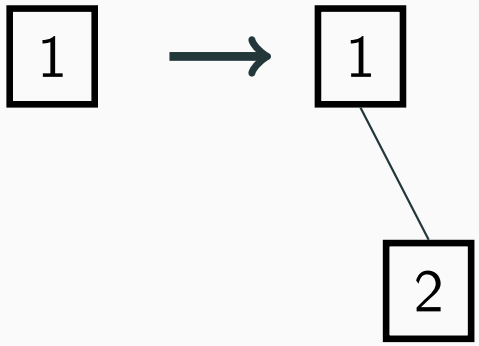
Binärer Baum



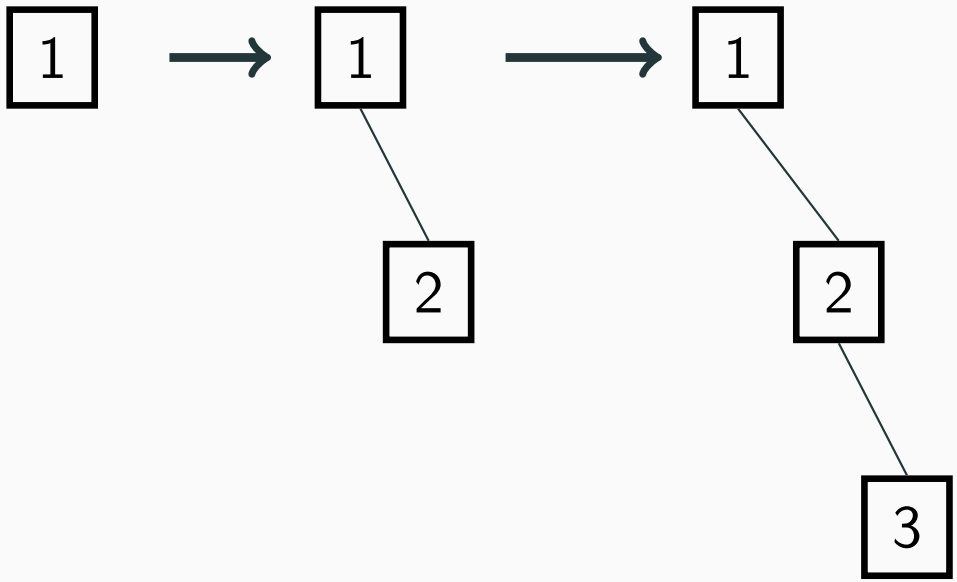
Entartung



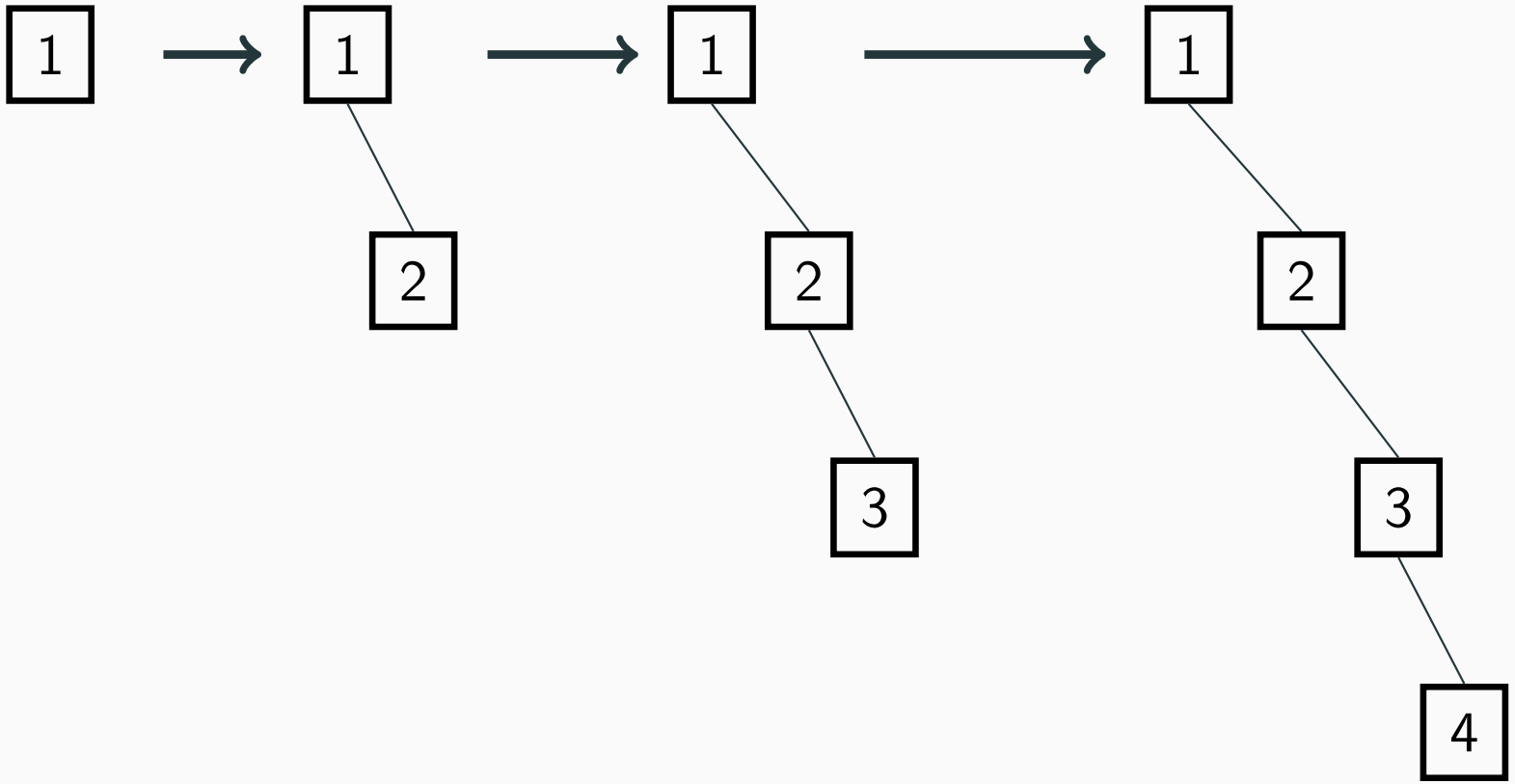
Entartung



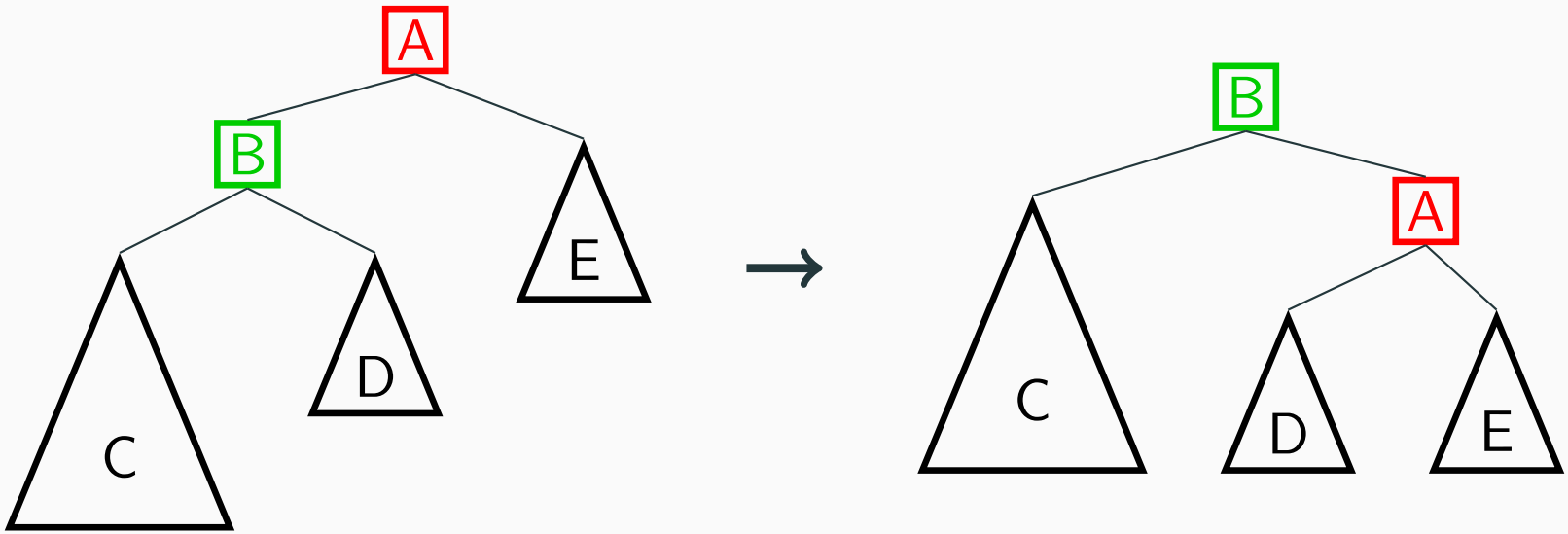
Entartung



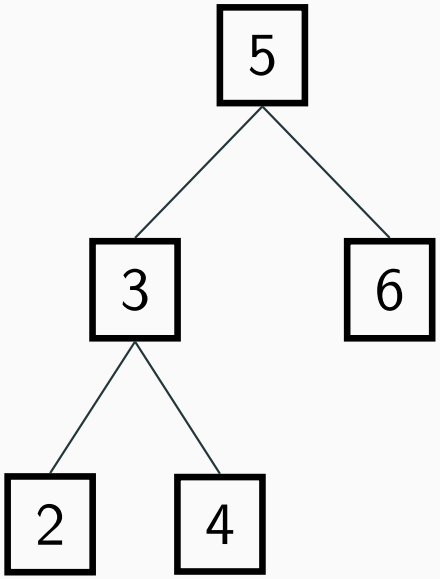
Entartung



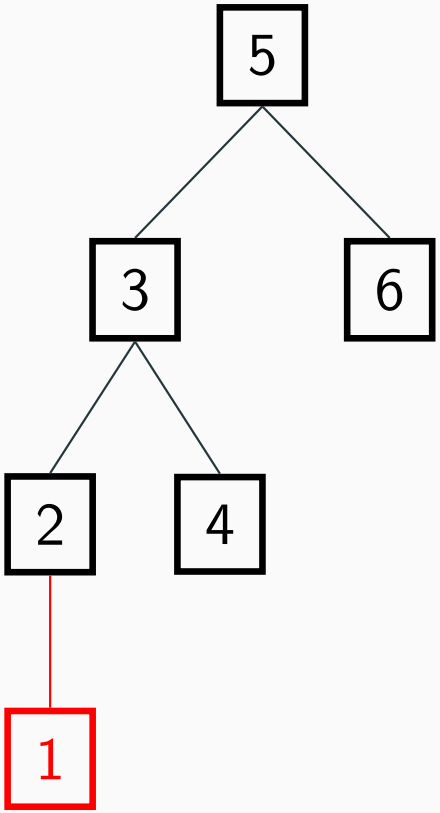
Rotation



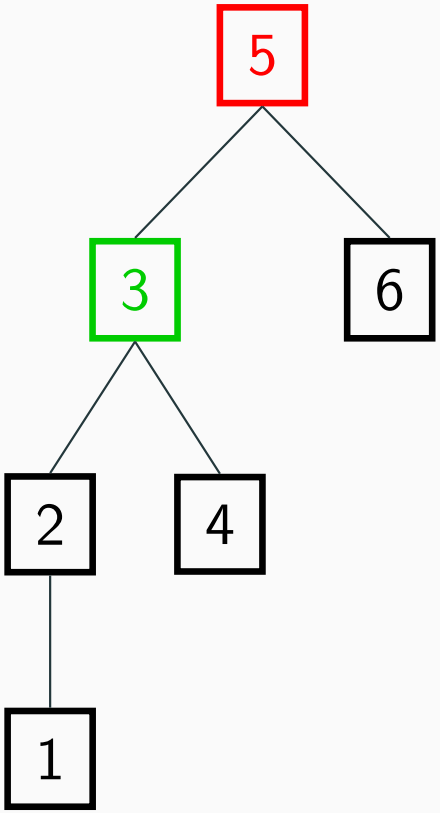
Rotation: Beispiel



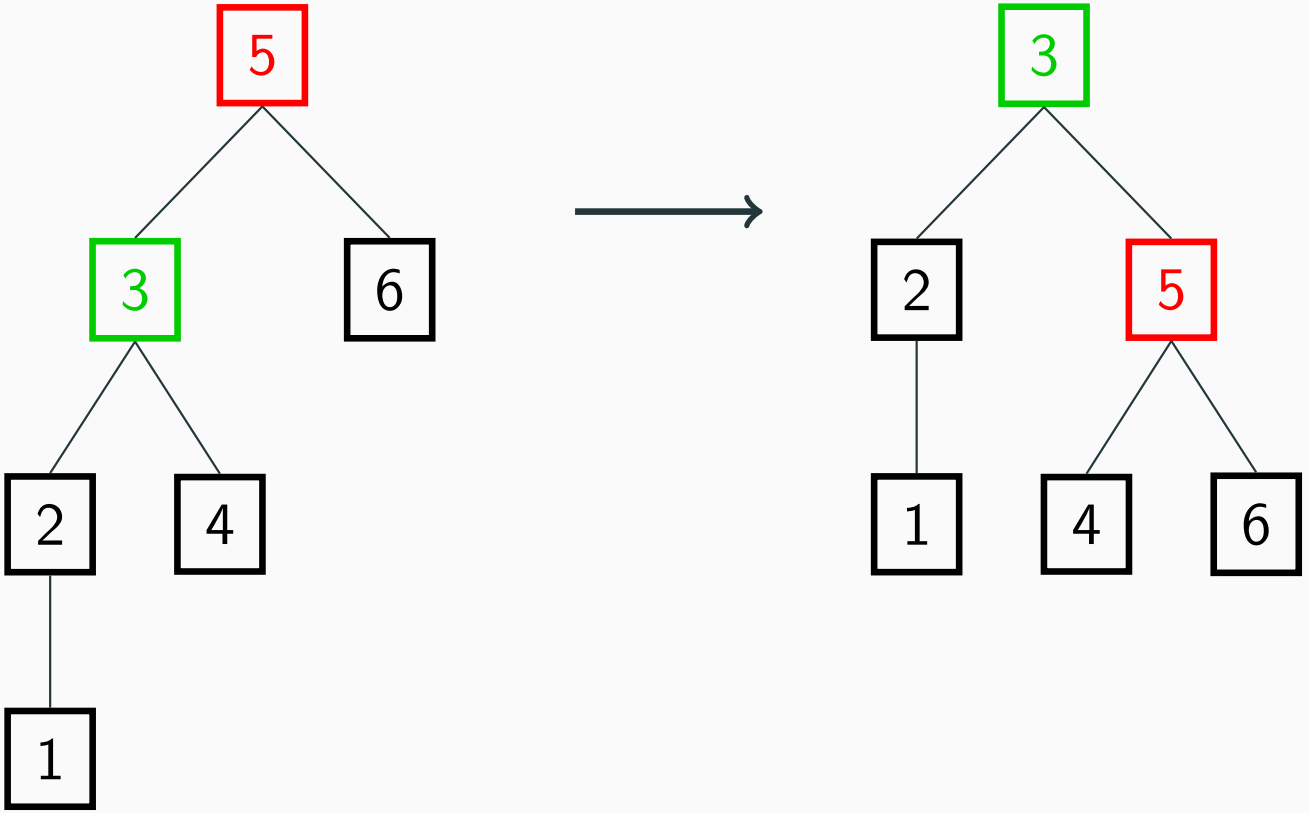
Rotation: Beispiel



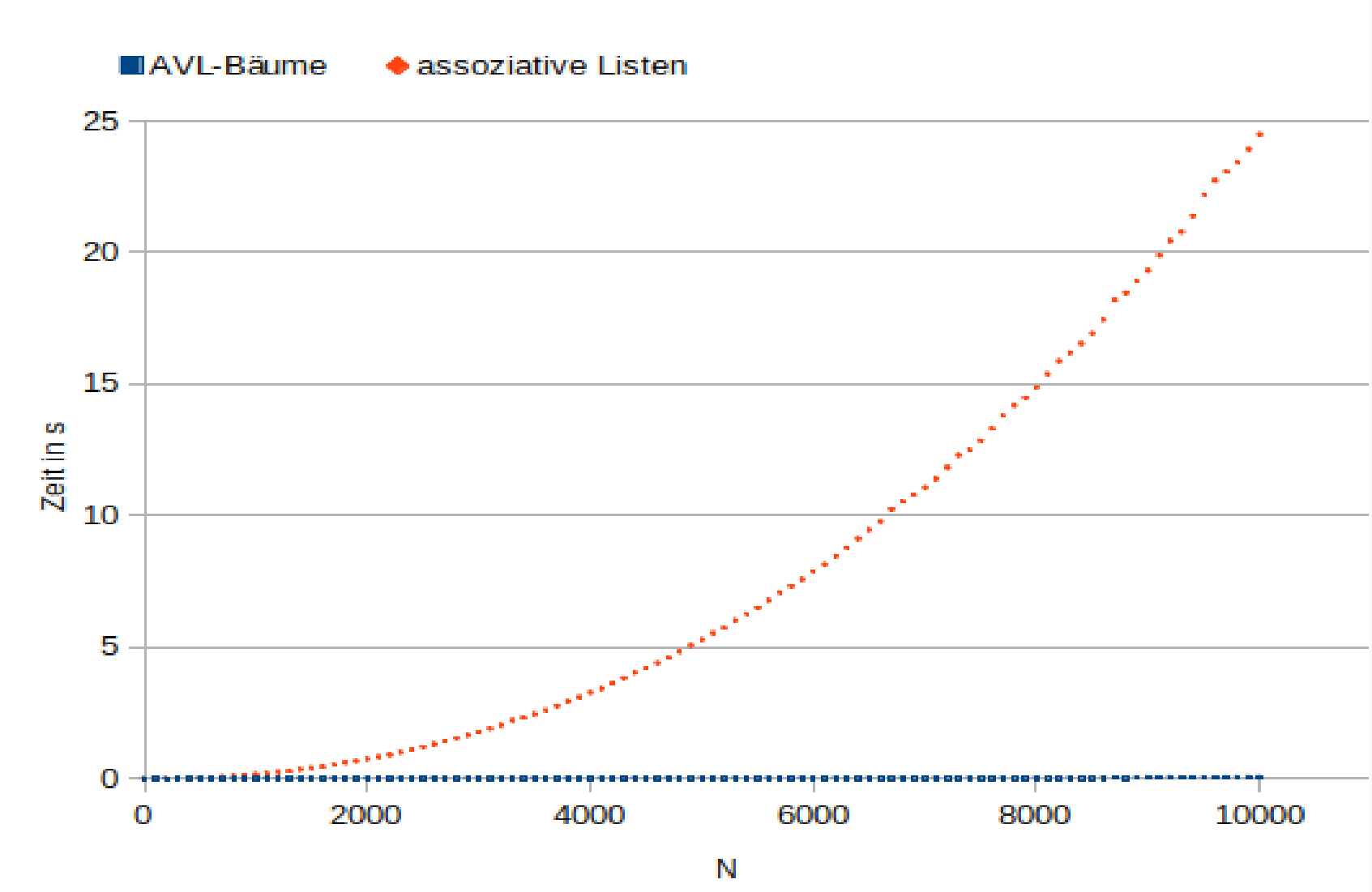
Rotation: Beispiel



Rotation: Beispiel



Vergleich mit Listenimplementierung



Verifikation

Möglichkeiten zur Überprüfung

- Manuelles Testen $\Longrightarrow$ aufwändig, wenig Fälle abdeckbar
- Automatisiertes Testen $\Longrightarrow$ weniger aufwändig, aber immer noch nicht alle Fälle testbar
- Verifikation $\Longrightarrow$ Beweis, gilt für alle Fälle
- Coq ist ein interaktiver Theorembeweiser $\Longrightarrow$ Beweise werden durch den Computer überprüft.

Bewiesene Eigenschaften

1. Die angestrebte Veränderung wird durchgeführt
2. Es finden keine weiteren Veränderungen statt
3. Alle Invarianten werden beibehalten

$\Longrightarrow$ nur die angestrebte Veränderung wird durchgeführt.

Beispiel: Anlegen eines Mappings

```
Theorem create_has_mapping :
 forall (db:mapping_db) (pd:N) (sel:N) (ko:kernel_object),
  mapping_db_inv db ->
  has_mapping pd sel ko (create_mapping pd sel ko db).
```

Beispiel: Anlegen eines Mappings

```
Theorem create_preserve_other :
 forall (db:mapping_db) (pd pd':N) (sel sel':N)
        (ko ko':kernel_object),
  (pd' <> pd \/ sel' <> sel) -> mapping_db_inv db ->
  (has_mapping pd sel ko db
   <-> has_mapping pd sel ko (create_mapping pd' sel' ko' db)).
```

Beispiel: Anlegen eines Mappings

```
Theorem create_invariant :
 forall (db:mapping_db) (pd:N) (sel:N) (ko:kernel_object),
  mapping_db_inv db ->
  mapping_db_inv (create_mapping pd sel ko db).
```

Ergebnisse und Ausblick

Ergebnisse

- 2000 Zeilen Quelltext
- verifizierte Implementierung von AVL-Bäumen in Coq
- abstrakte Implementierung und Verifikation einer Mapping-Datenbank
- durch Verifikation wurden mehrere Fehler gefunden
  - Fehler bei der Implementierung des Löschens
  - Falsche Anpassung der Balance im Falle einer Rotation

Ausblick

- Verknüpfung mit Operationen des Microkerns
- Refinement auf C-Implementierung

Fragen?

- Bildquellen
  - http://cache.techie-buzz.com/images4/chinmoy/linux-kernel-rise.jpg
- Vielen Dank für die Aufmerksamkeit!