Sichere Zukunftsvorhersagen durch Modellierung und Approximation

Jan Reineke @ SAARLAND UNIVERSITY
COMPUTER SCIENCE

Forschungstage Informatik

24. Juni 2016

Der Kontext: “Harte” Echtzeitsysteme

Sicherheitskritische Anwendungen:

- Luft- und Raumfahrt, Automobilindustrie, Eisenbahnbau, Produktion

Airbag
Reaktion in < 10 Millisekunden

kurbelwellensynchrone Berechnungen
Reaktion in < 45 Mikrosekunden

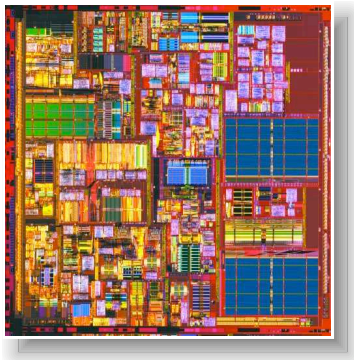
- Eingebettete Software muss
  - korrekte Steuersignale,
  - innerhalb fixer Zeitschranken berechnen.

Das “Zeitanalyseproblem”

```
// Perform the convolution.
for (int i=0; i<10; i++) {
  x[i] = a[i]*b[j-i];
  // Notify listeners.
  notify(x[i]);
}
```

Software

+



Mikroarchitektur

?

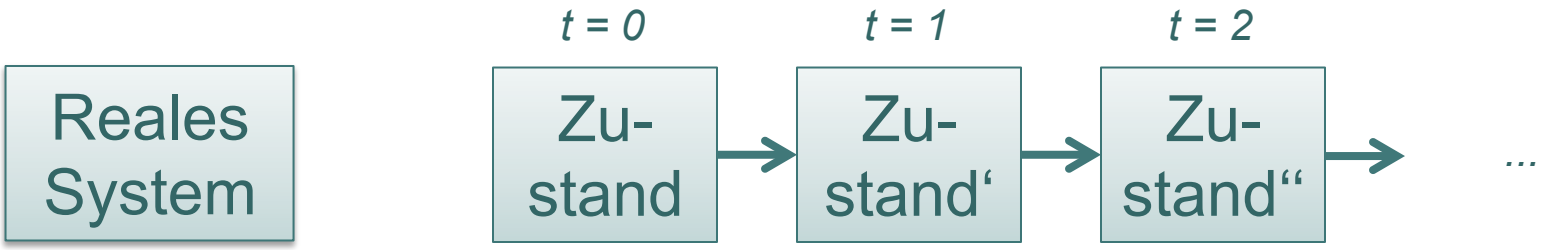
“worst-case execution time” (WCET)

Zukunftsvorhersage Allgemein

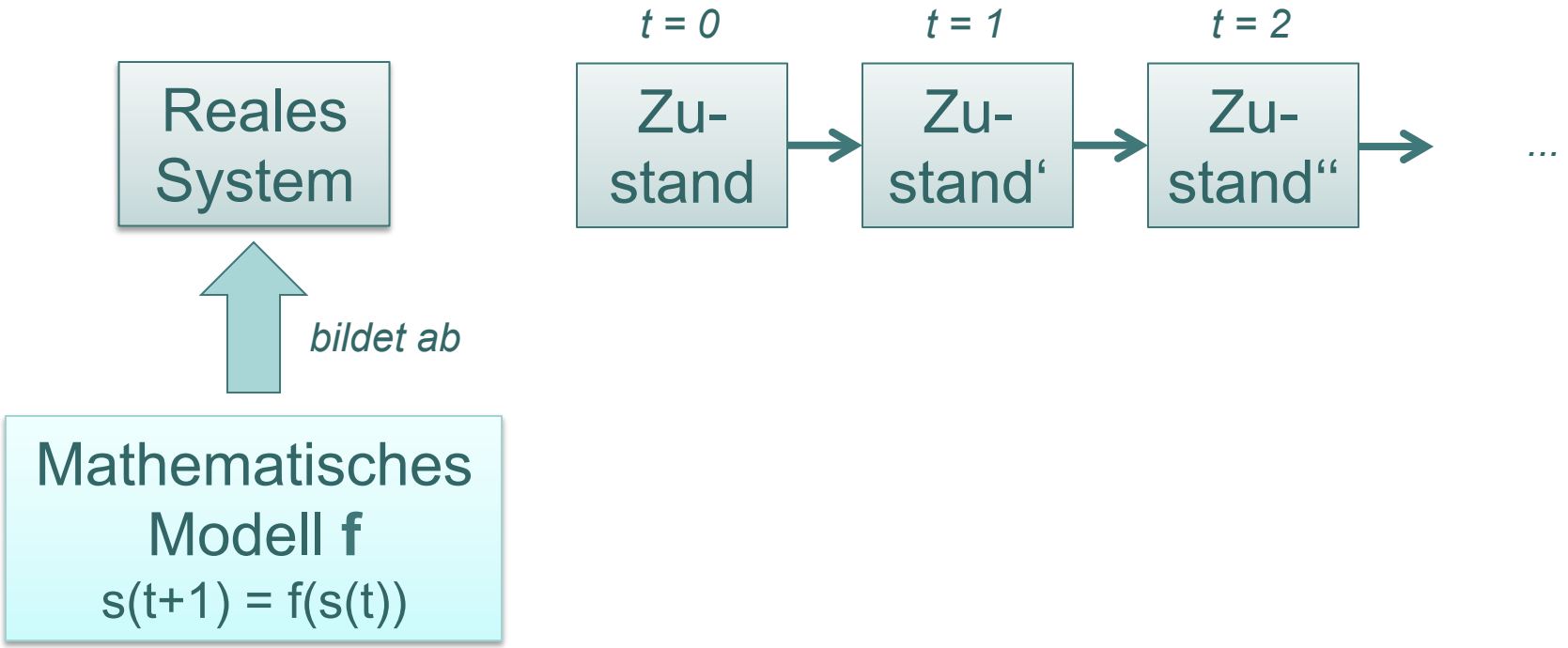
Prediction is very difficult,
especially if it‘s about the future.

Niels Bohr (Physiker)

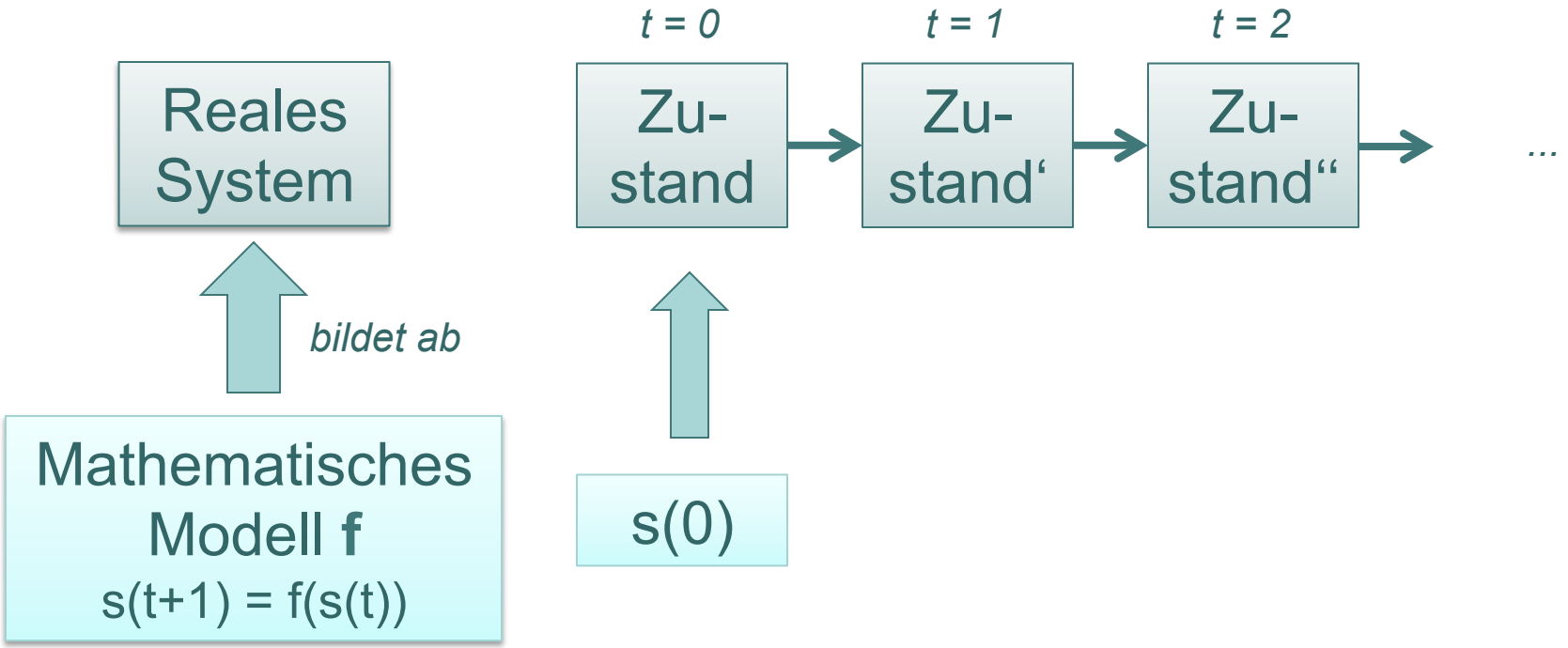
Vorhersagen basieren auf Modellen



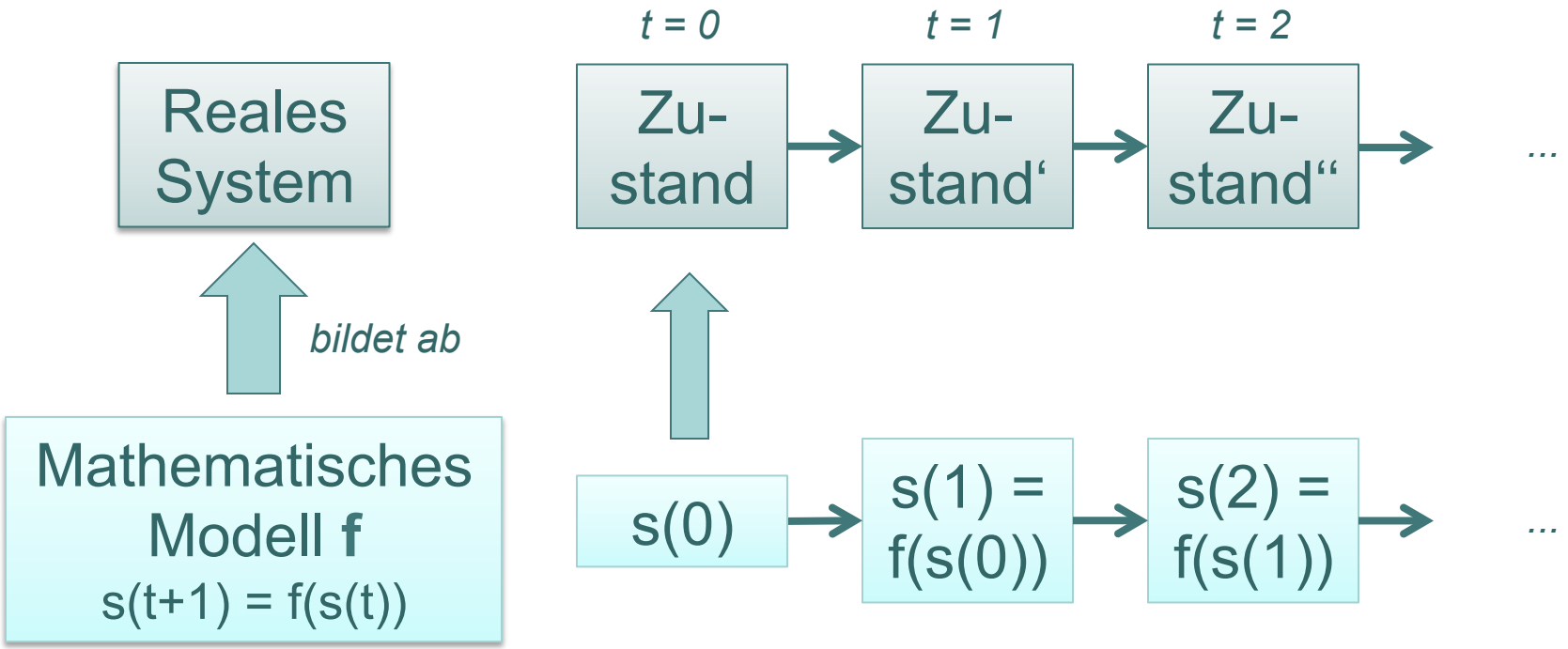
Vorhersagen basieren auf Modellen



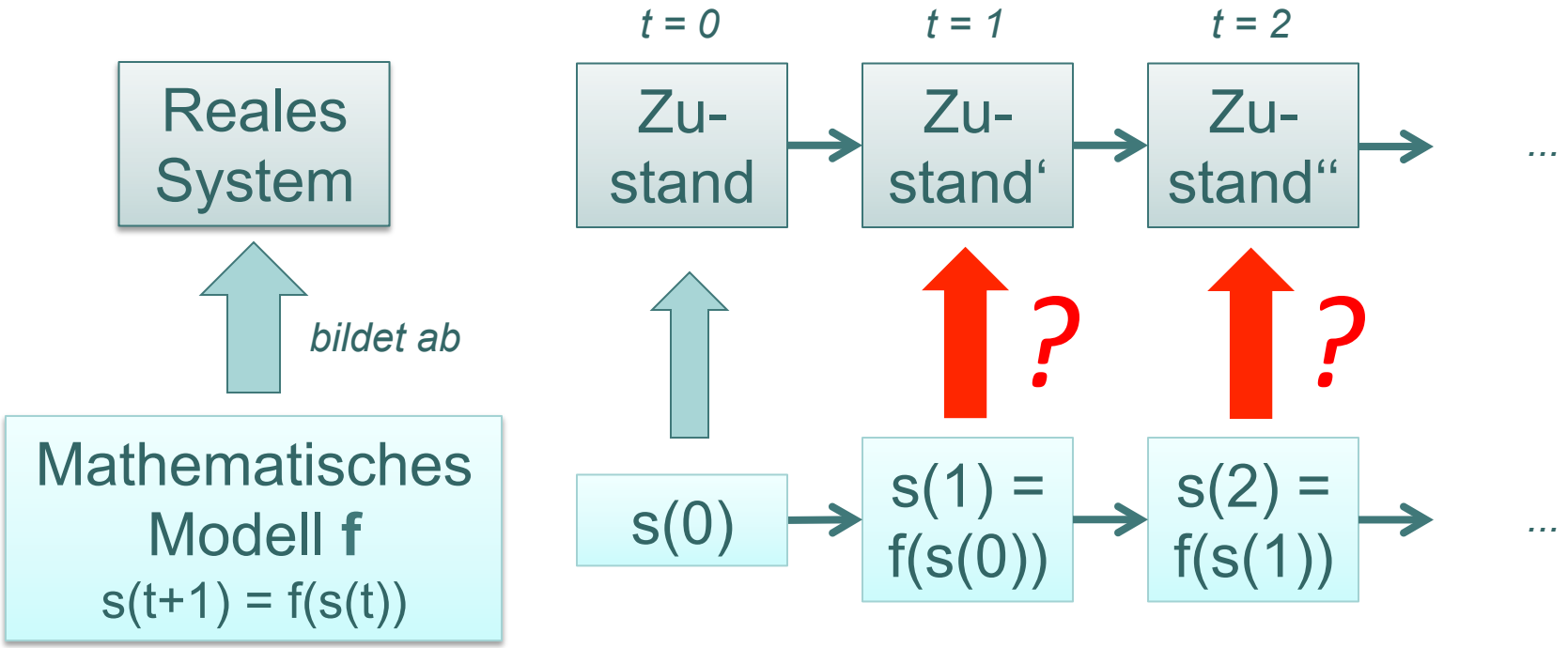
Vorhersagen basieren auf Modellen



Vorhersagen basieren auf Modellen

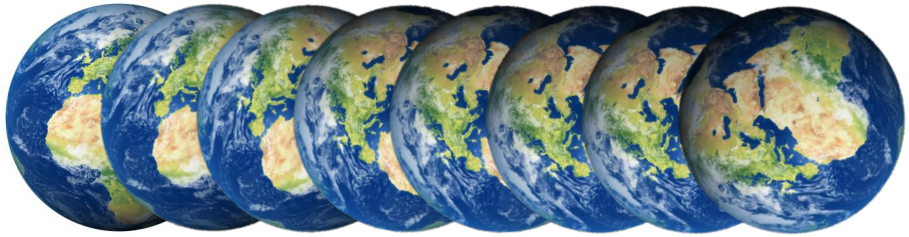


Vorhersagen basieren auf Modellen



Beispiel: Wettervorhersage

Reale Welt

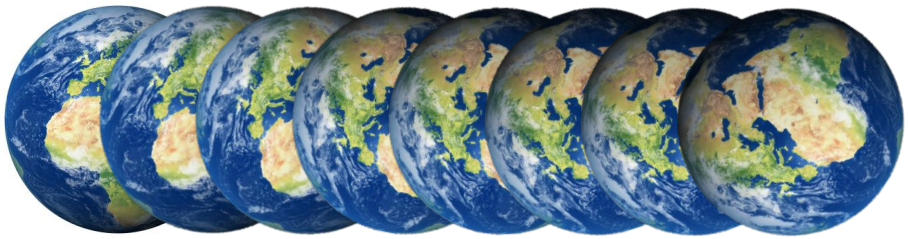


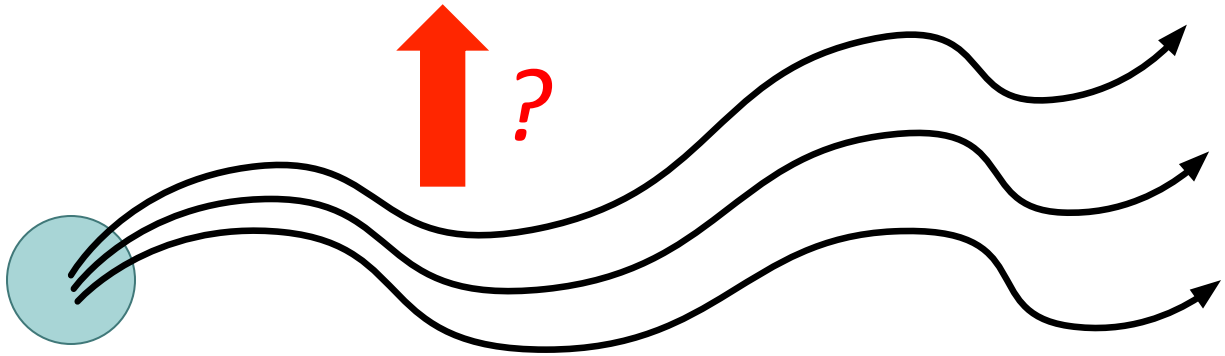
Beispiel: Wettervorhersage

Reale Welt

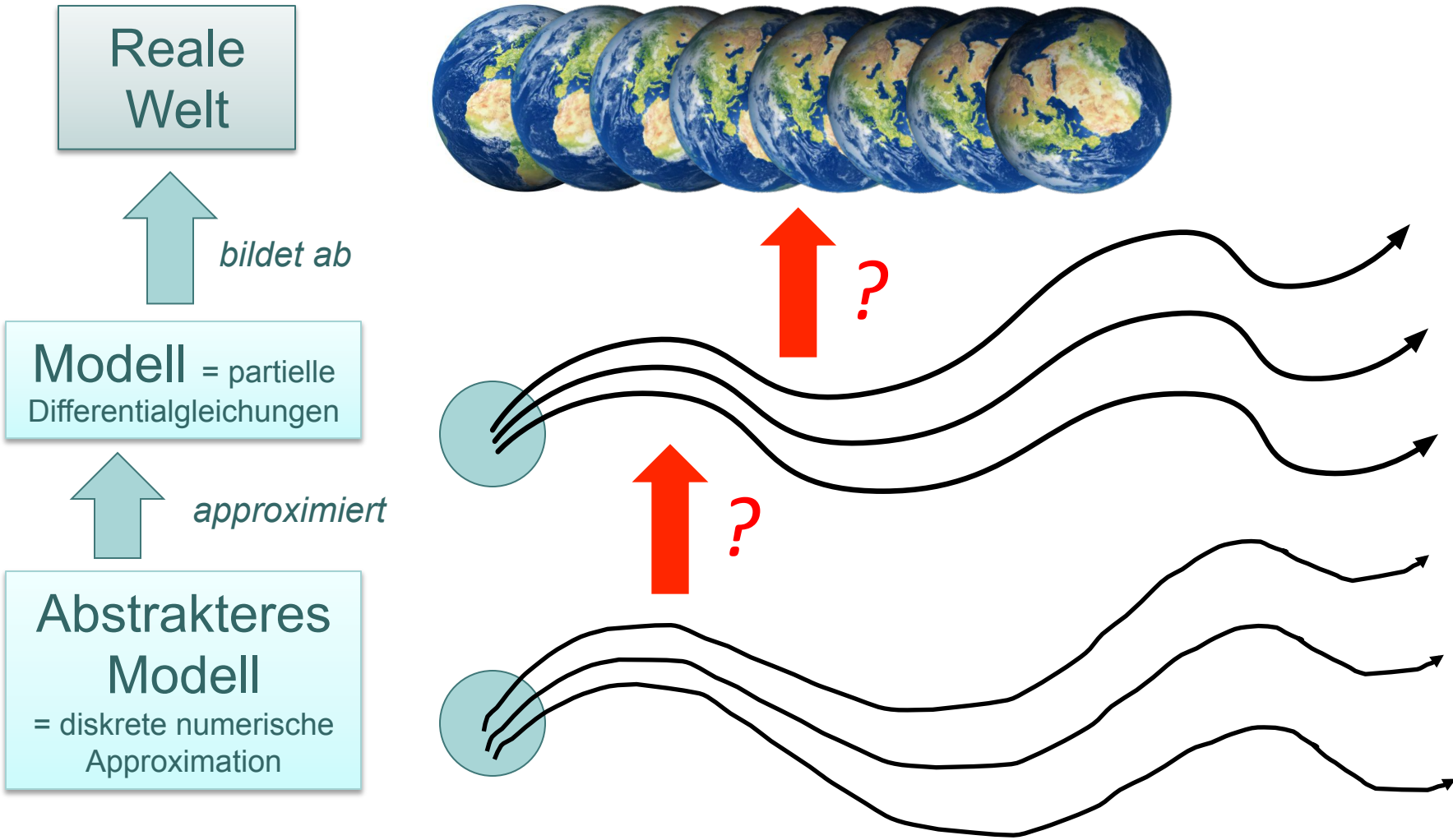
bildet ab

Modell = partielle Differentialgleichungen





Beispiel: Wettervorhersage



Zur Güte der Vorhersagen

Hängt ab von:

- Beziehung des Modells zur Realität
- Unsicherheit über initialen Zustand
- Inhärente Robustheit des realen Systems

All models are wrong, but some are useful.

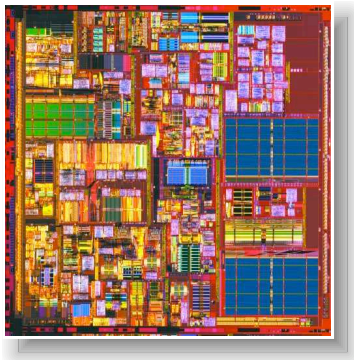
George Box (Statistiker)

Zurück zum “Zeitanalyseproblem”

```
// Perform the convolution.
for (int i=0; i<10; i++) {
  x[i] = a[i]*b[j-i];
  // Notify listeners.
  notify(x[i]);
}
```

Software

+



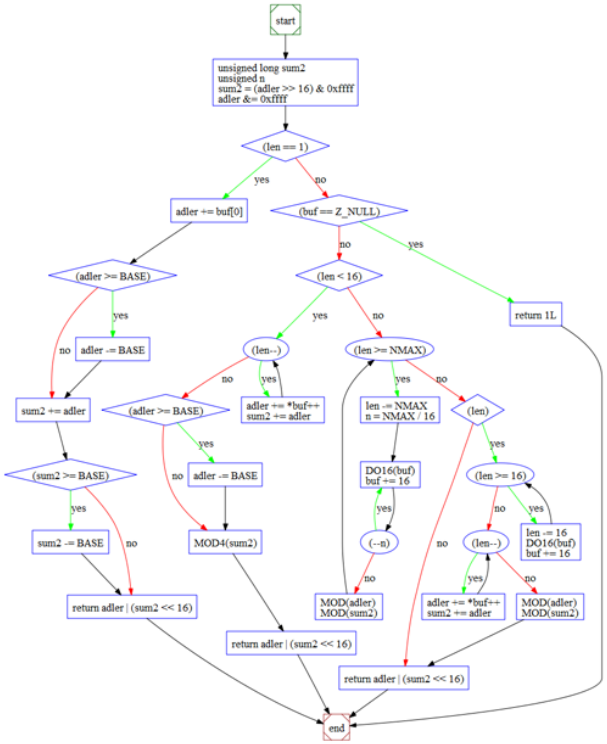
Mikroarchitektur

?

“Worst-Case Execution Time” (WCET)
= maximale Laufzeit unter allen möglichen initialen Bedingungen

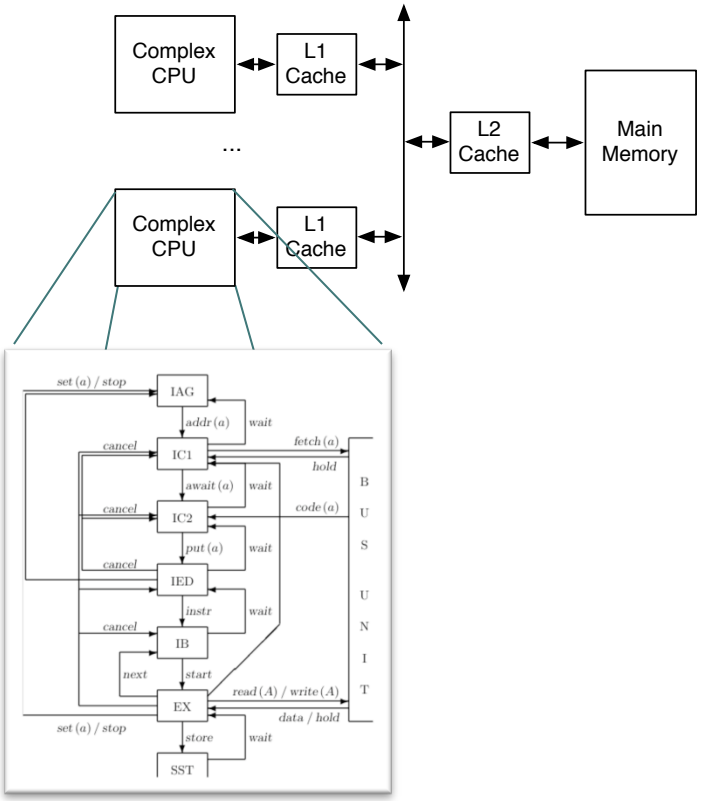
Wovon hängt die Ausführungszeit eines Programms ab?

Eingabeabhängiger Kontrollfluss



+

Mikroarchitektureller Zustand

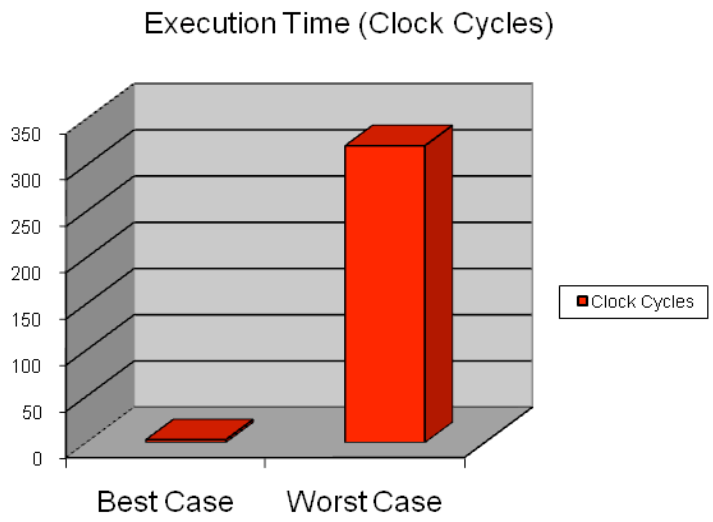


Einfluss der Mikroarchitektur

x=a+b; →

```
LOAD   r2, _a
LOAD   r1, _b
ADD    r3,r2,r1
```

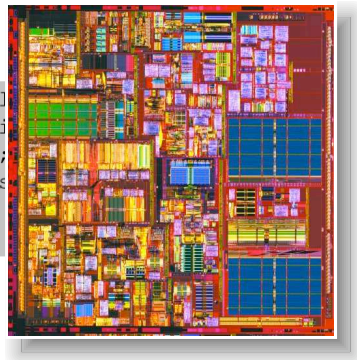
PowerPC 755

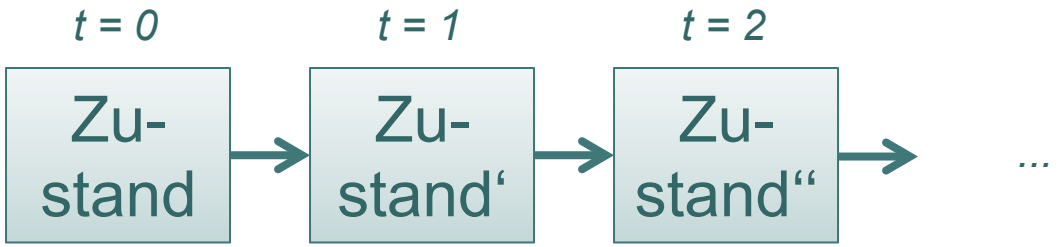


Naiver Ansatz zur Zeitanalyse

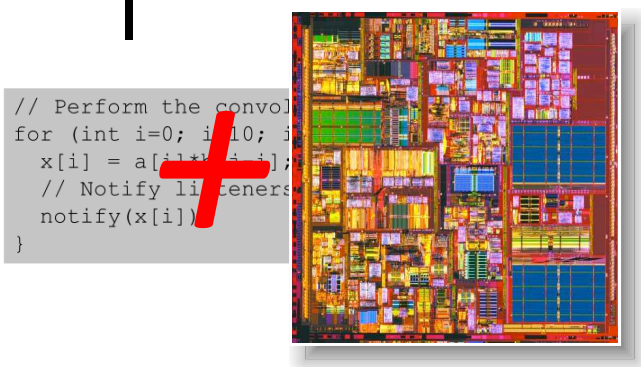
```
// Perform the convol
for (int i=0; i 10; i
  x[i] = a[i]t  i  i];
  // Notify li teners
  notify(x[i])
}
```

+

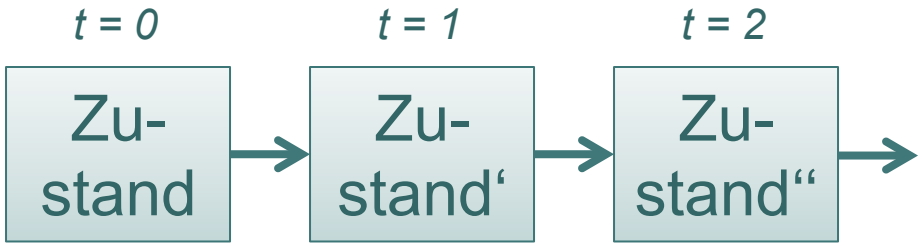




Naiver Ansatz zur Zeitanalyse

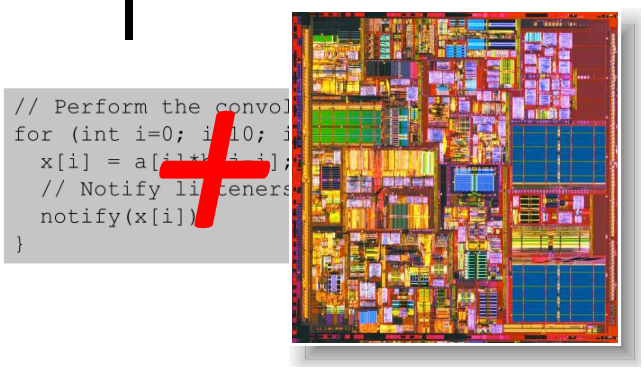


bildet ab

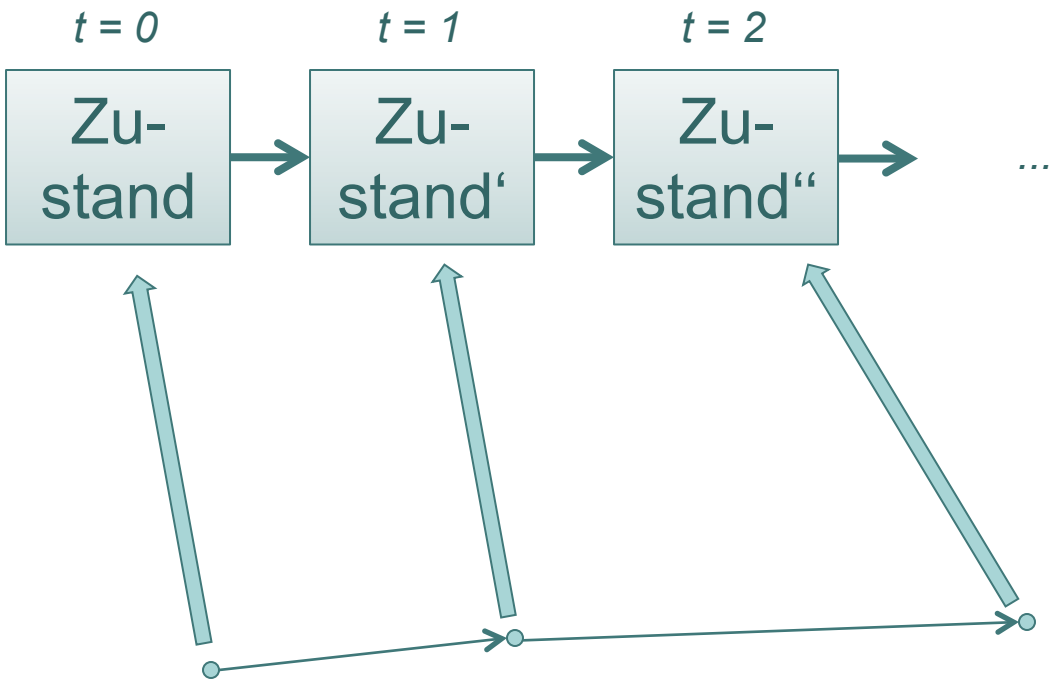


Konkretes Modell:
Zyklengenaues Modell der Hardware und Software

Naiver Ansatz zur Zeitanalyse



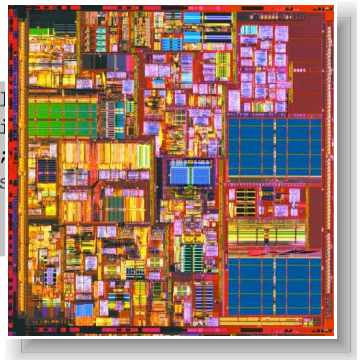
bildet ab



Konkretes Modell:
Zyklengenaues Modell der Hardware und Software

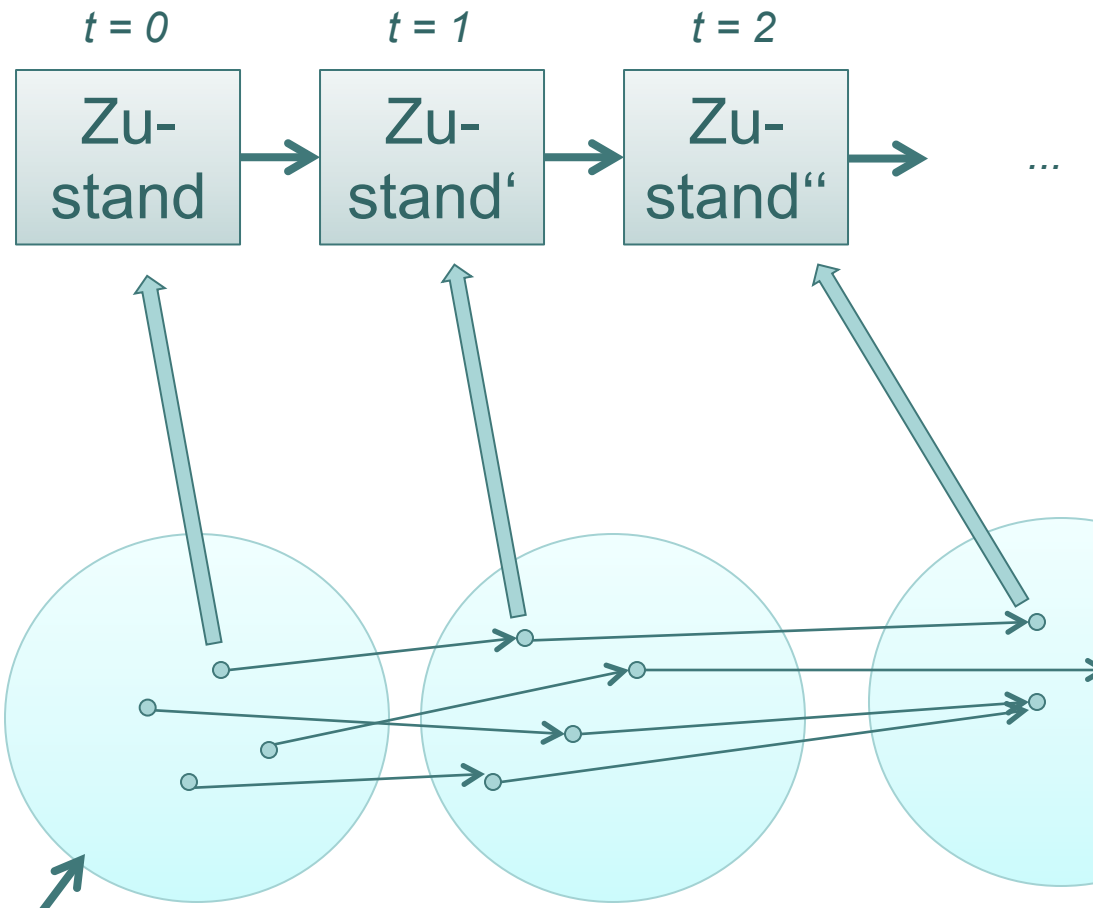
Naiver Ansatz zur Zeitanalyse

```
// Perform the convol
for (int i=0; i  10; i
  x[i] = a[i]t  i  i];
  // Notify li teners
  notify(x[i])
}
```



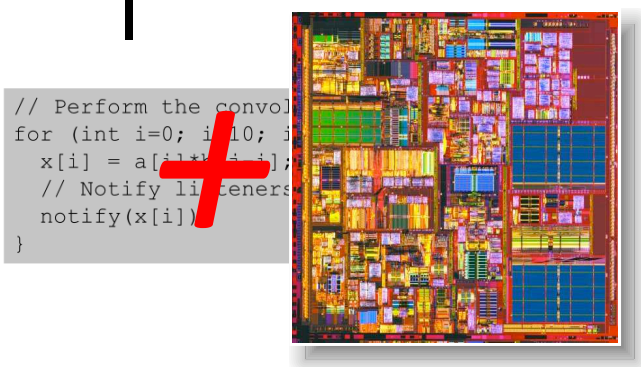
bildet ab

Konkretes Modell:
Zyklengenaues Modell der Hardware und Software



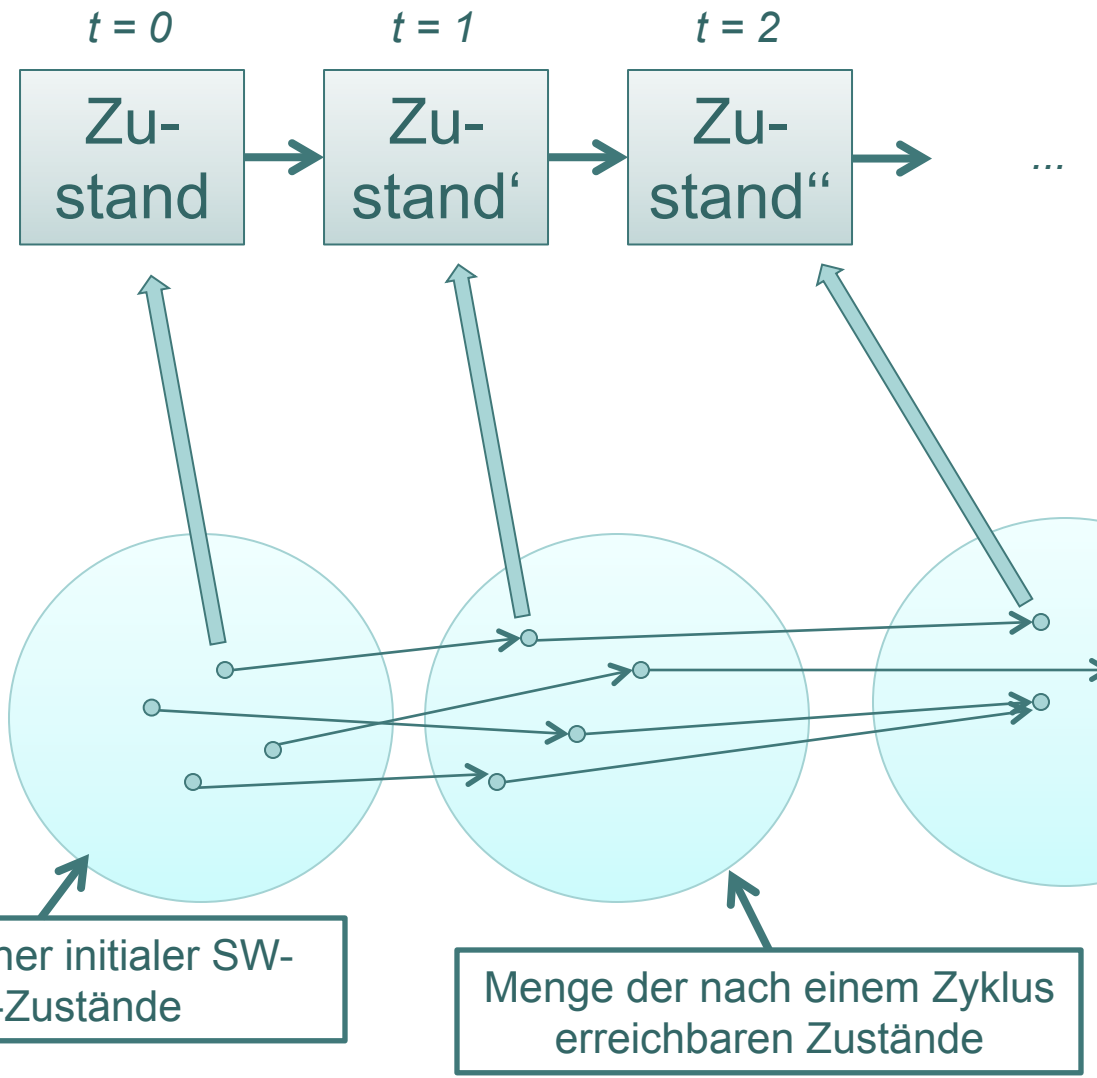
Menge möglicher initialer SW- und HW-Zustände

Naiver Ansatz zur Zeitanalyse



bildet ab

Konkretes Modell:
Zyklengenaues Modell der Hardware und Software



Naiver Ansatz zur Zeitanalyse

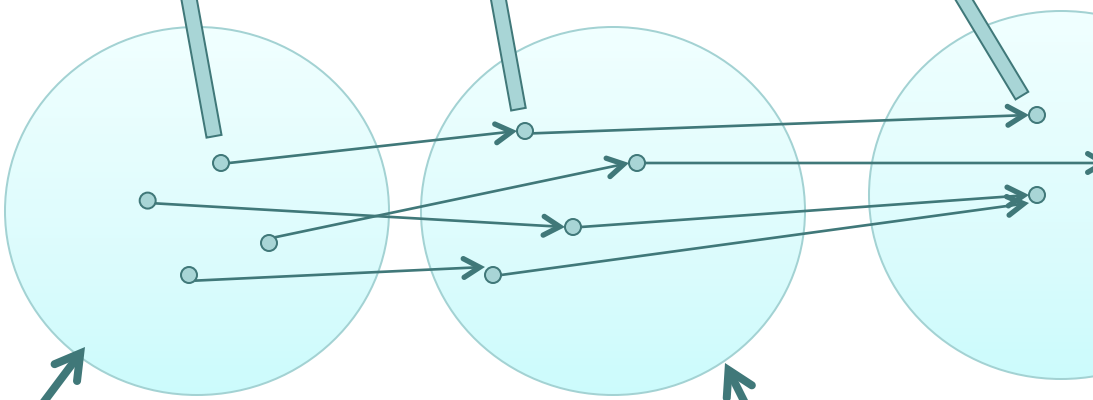
t = 0 | t = 1 | t = 2

Zu- | Zu- | Zu-

...

Positiv: Modell bildet Realität im Normalfall sehr präzise ab!
Ausnahmen?

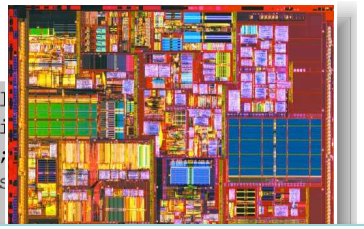
Konkretes Modell:
Zyklengenaues Modell der Hardware und Software



Menge möglicher initialer SW- und HW-Zustände

Menge der nach einem Zyklus erreichbaren Zustände

Naiver Ansatz zur Zeitanalyse



```
// Perform the convol
for (int i=0; i  10; i
  x[i] = a[i]
  // Notify li teners
  notify(x[i])
}
```

t = 0 t = 1 t = 2

Zu- Zu- Zu-

...

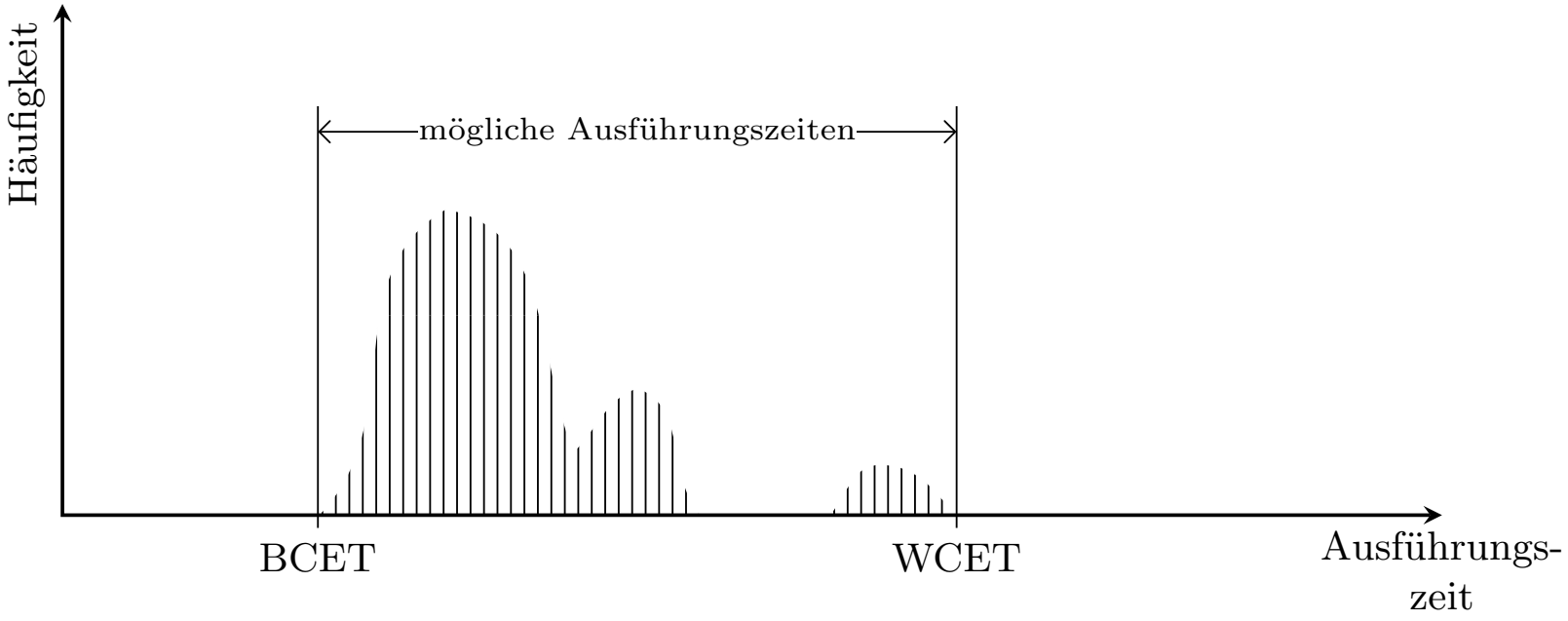
Positiv: Modell bildet Realität im Normalfall sehr präzise ab! Ausnahmen?

Problem: Menge der initialen SW- und HW-Zustände zu groß um alle Fälle explizit durchzuspielen!

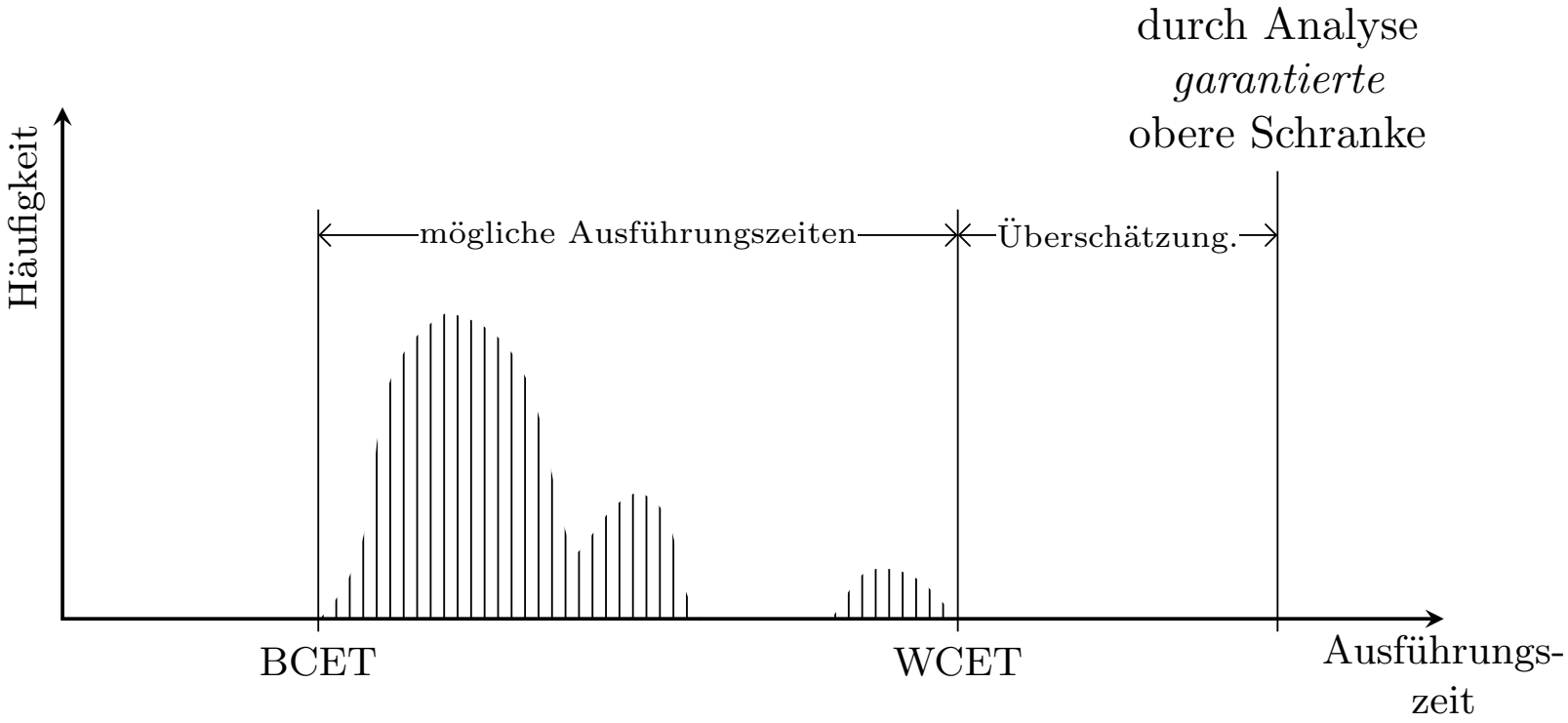
Menge möglicher initialer SW- und HW-Zustände

Menge der nach einem Zyklus erreichbaren Zustände

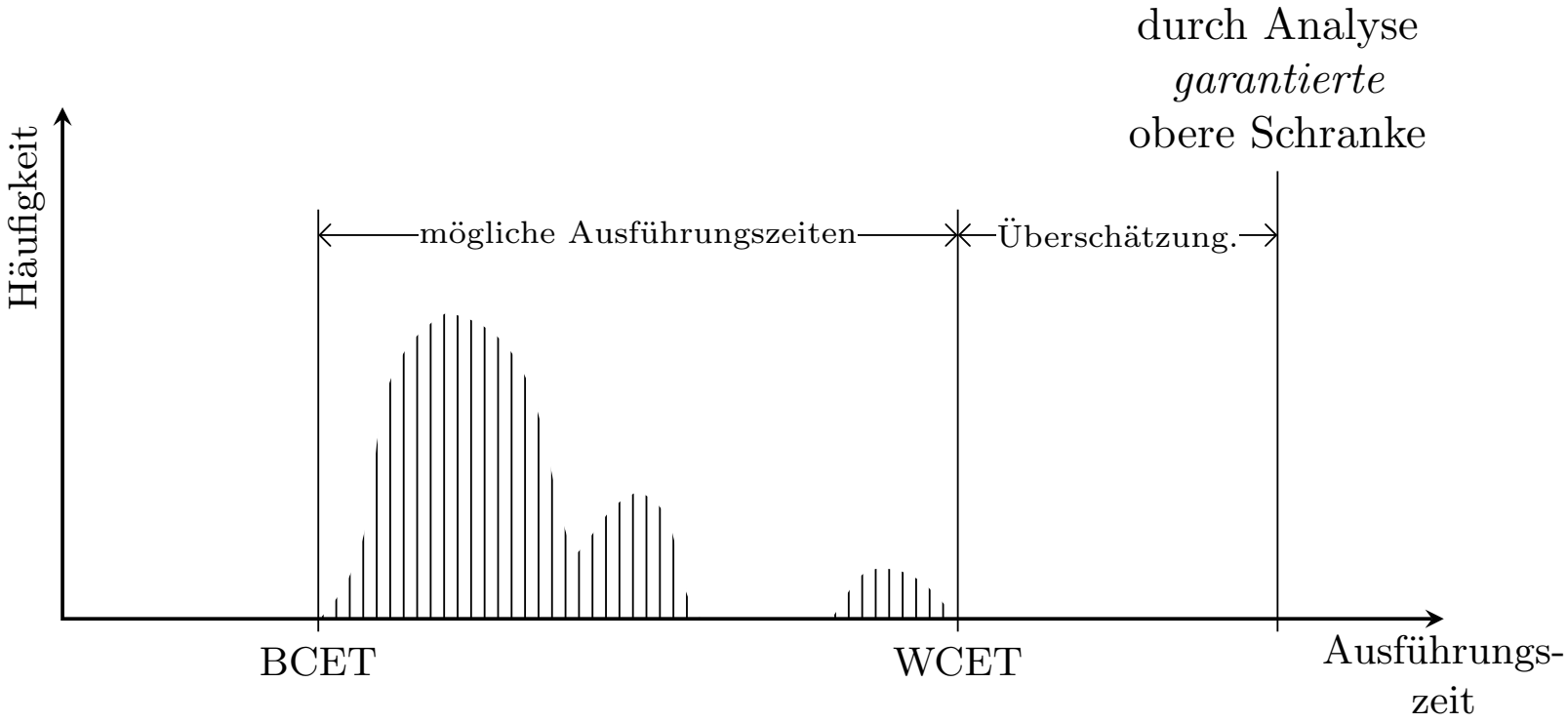
Ziel der Zeitanalyse



Ziel der Zeitanalyse



Ziel der Zeitanalyse



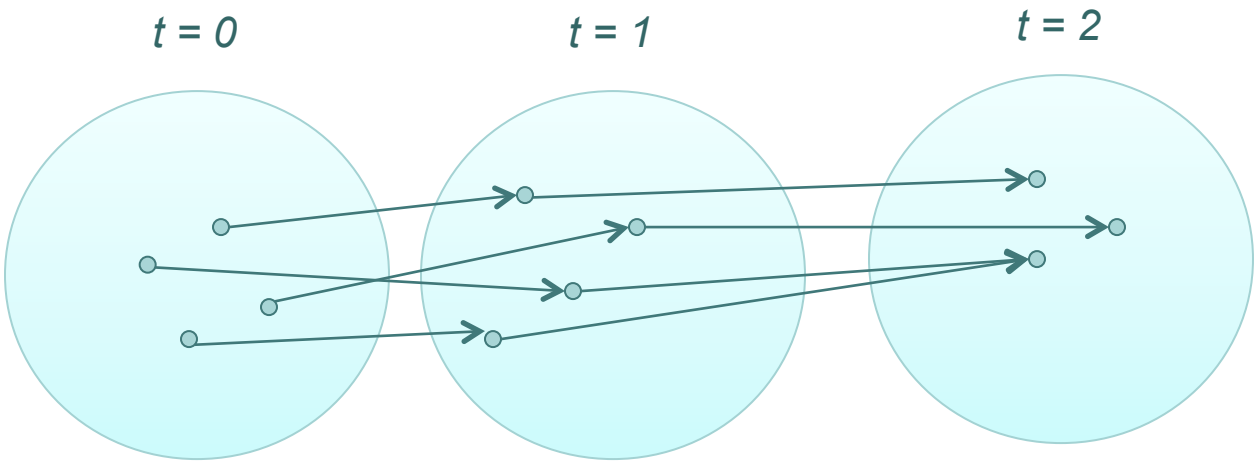
Unterschied zur Wettervorhersage:
Ausführungszeit darf auf keinen Fall unterschätzt werden!

Effiziente Zeitanalyse durch Abstraktion

Konkretes Modell:
Zyklengenaues Modell der Hardware und Software

sichere Approximation

Abstrakteres Modell:
Abstraktion der HW + SW

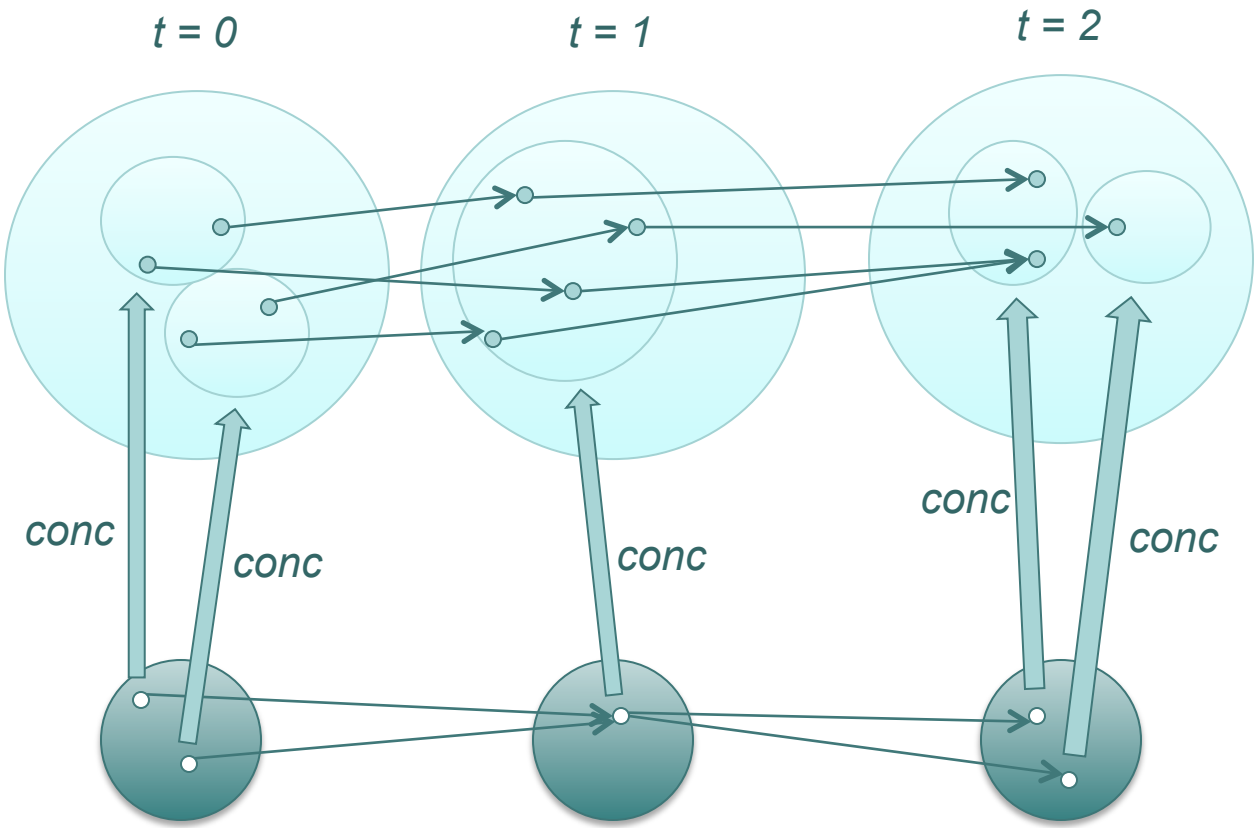


Effiziente Zeitanalyse durch Abstraktion

Konkretes Modell:
Zyklengenaues Modell der Hardware und Software

sichere Approximation

Abstrakteres Modell:
Abstraktion der HW + SW



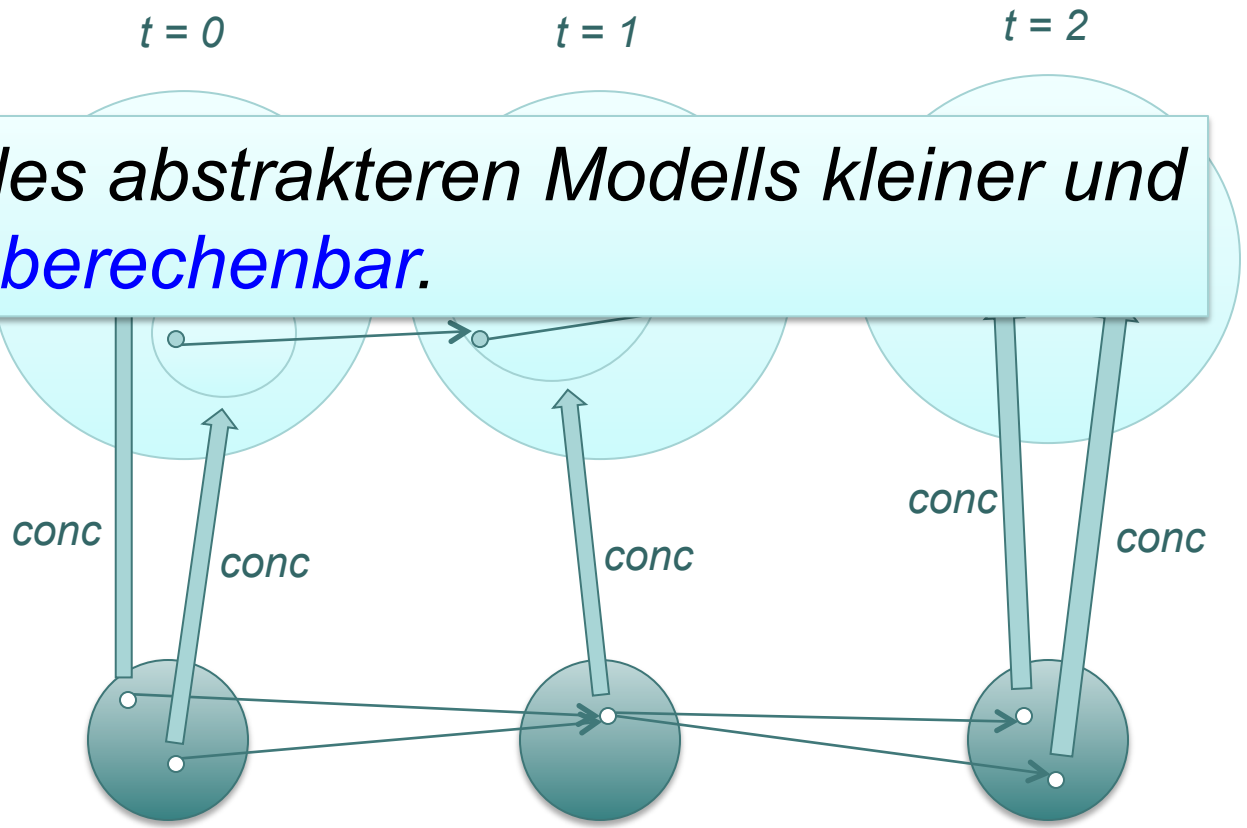
Effiziente Zeitanalyse durch Abstraktion

Zustandsraum des abstrakteren Modells kleiner und daher praktisch berechenbar.

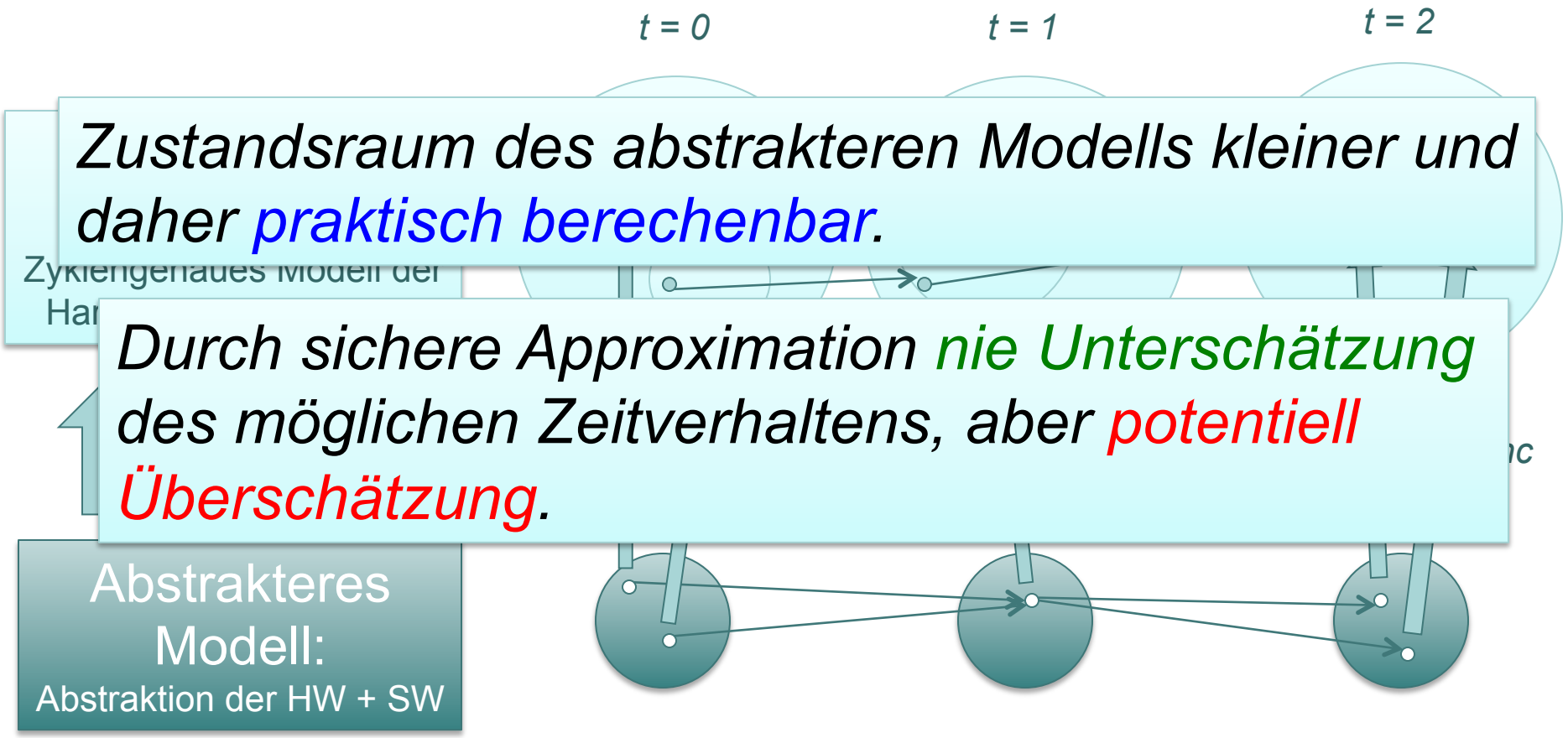
Zyklengenaues Modell der Hardware und Software

sichere Approximation

Abstrakteres Modell:
Abstraktion der HW + SW



Effiziente Zeitanalyse durch Abstraktion

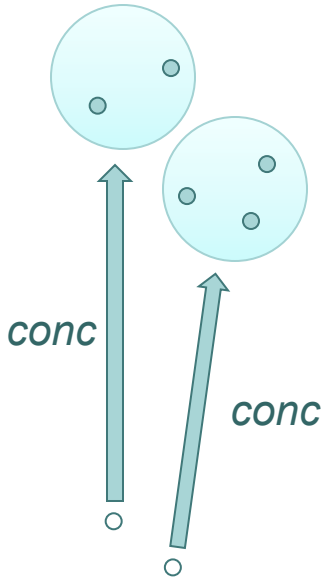


Zustandsraum des abstrakteren Modells kleiner und daher praktisch berechenbar.

Durch sichere Approximation nie Unterschätzung des möglichen Zeitverhaltens, aber potentiell Überschätzung.

Wie wird „sichere Approximation“ erreicht?

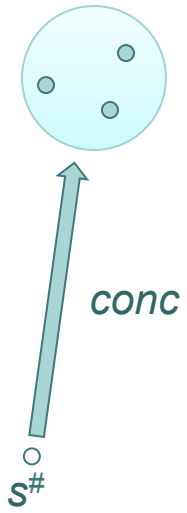
1. Jeder abstrakte Zustand $s^{\#}$ repräsentiert eine Menge *conc($s^{\#}$)* konkreter Zustände:



Wie wird „sichere Approximation“ erreicht?

2. ***Lokale Konsistenz***:

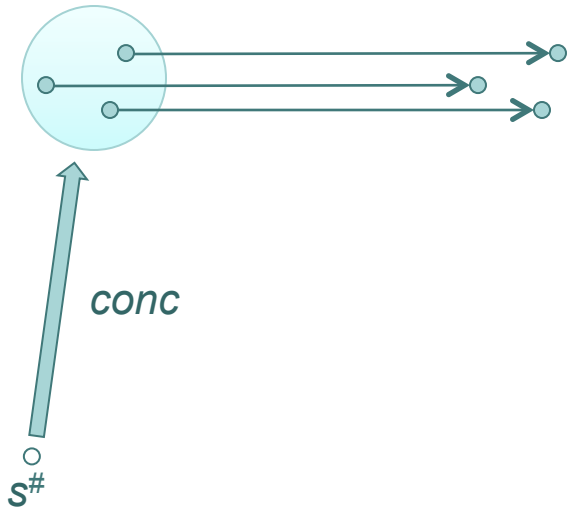
Die Nachfolger der Konkretisierung eines abstrakten Zustands $s^{\#}$ werden durch seine Nachfolger repräsentiert:



Wie wird „sichere Approximation“ erreicht?

2. ***Lokale Konsistenz***:

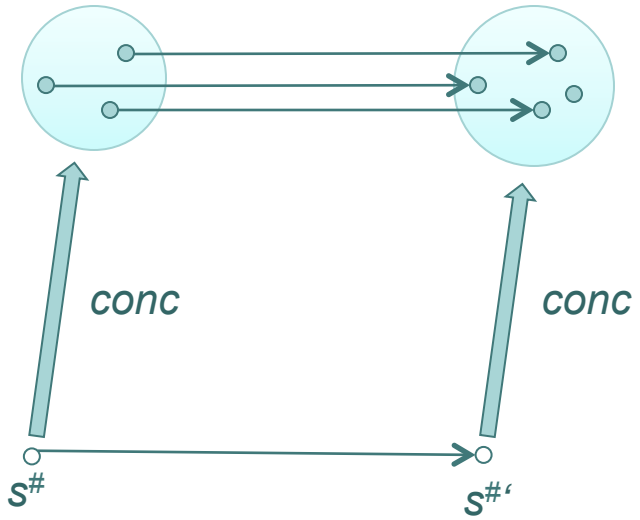
Die Nachfolger der Konkretisierung eines abstrakten Zustands $s^{\#}$ werden durch seine Nachfolger repräsentiert:



Wie wird „sichere Approximation“ erreicht?

2. ***Lokale Konsistenz***:

Die Nachfolger der Konkretisierung eines abstrakten Zustands $s^{\#}$ werden durch seine Nachfolger repräsentiert:

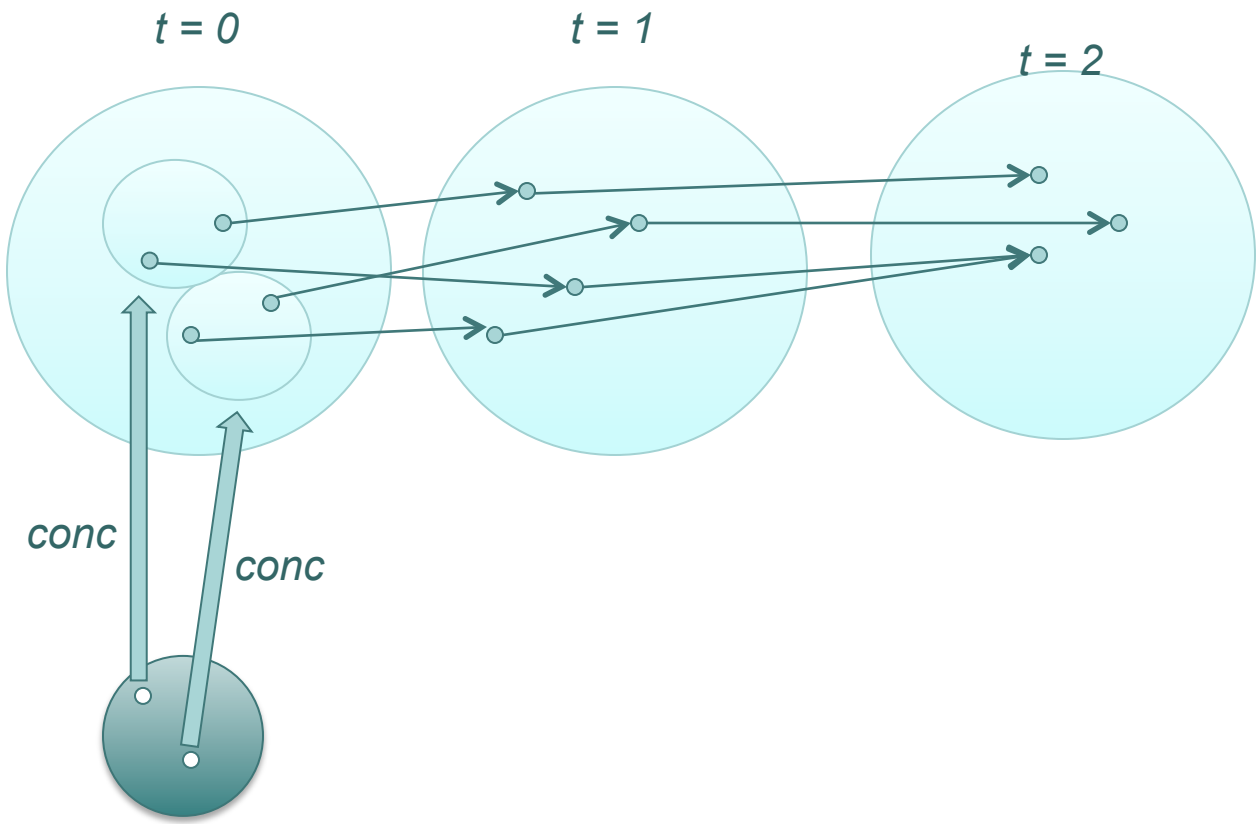


Wie wird „sichere Approximation“ erreicht?

Konkretes Modell:
Zyklengenaues Modell der Hardware und Software

sichere Approximation

Abstrakteres Modell:
Abstraktion der HW + SW

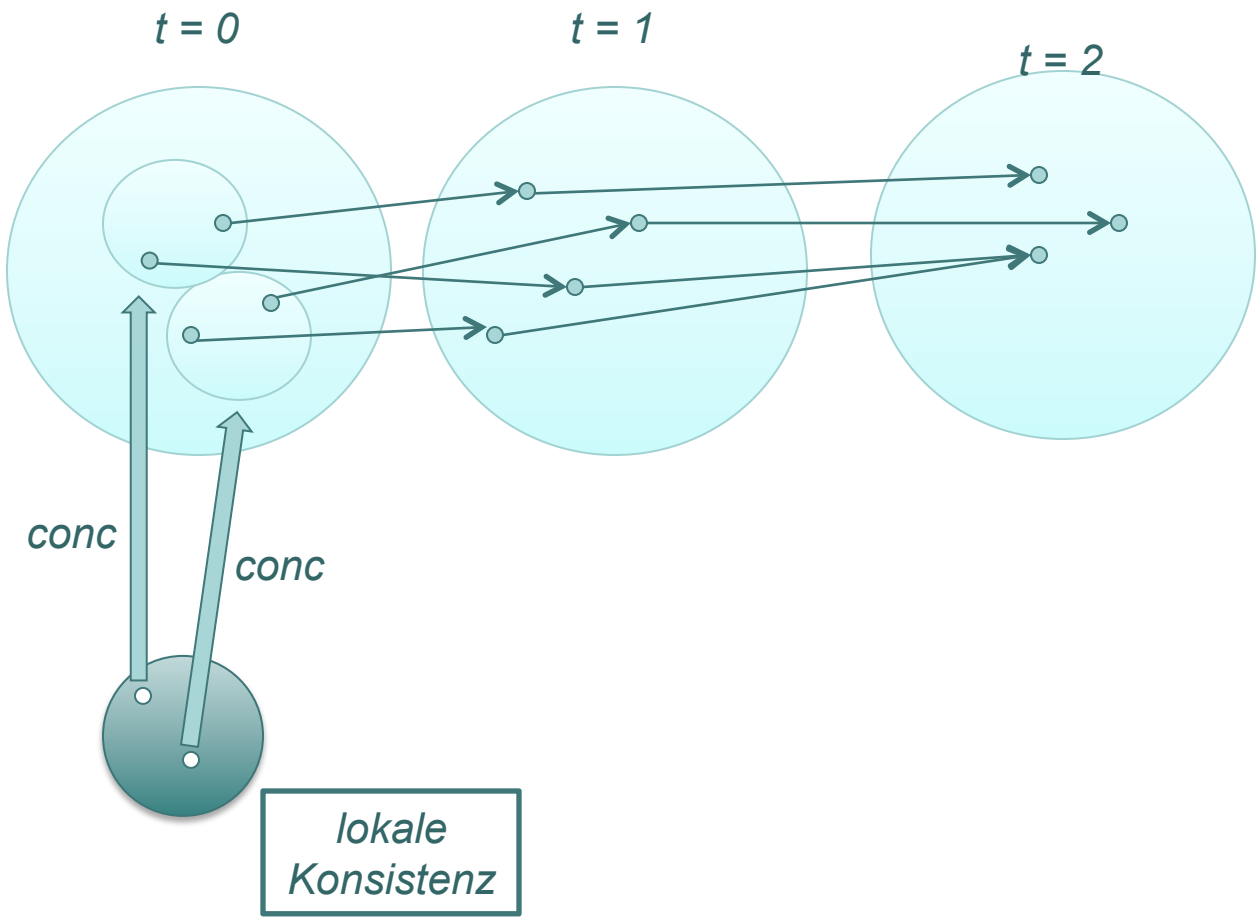


Wie wird „sichere Approximation“ erreicht?

Konkretes Modell:
Zyklengenaues Modell der Hardware und Software

sichere Approximation

Abstrakteres Modell:
Abstraktion der HW + SW

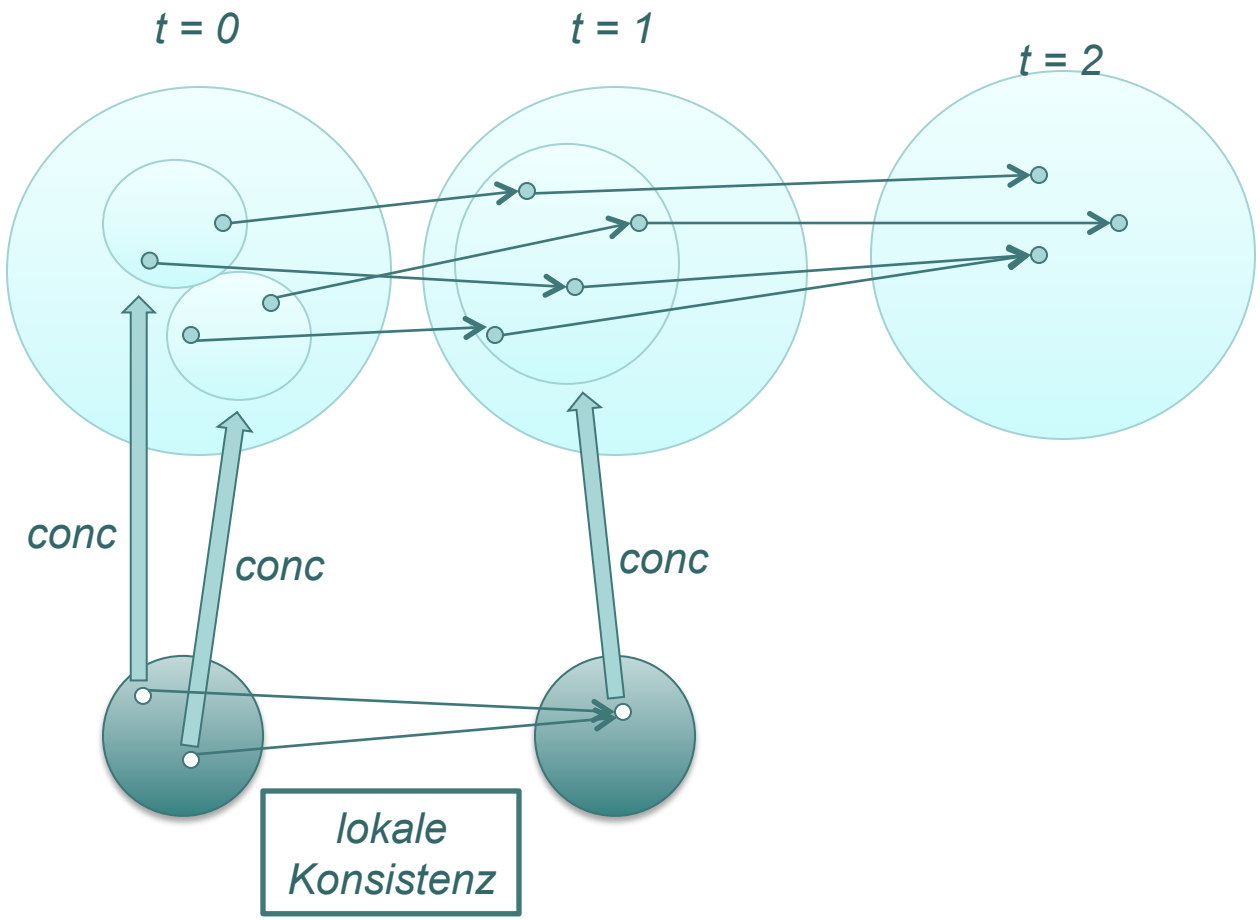


Wie wird „sichere Approximation“ erreicht?

Konkretes Modell:
Zyklengenaues Modell der Hardware und Software

sichere Approximation

Abstrakteres Modell:
Abstraktion der HW + SW

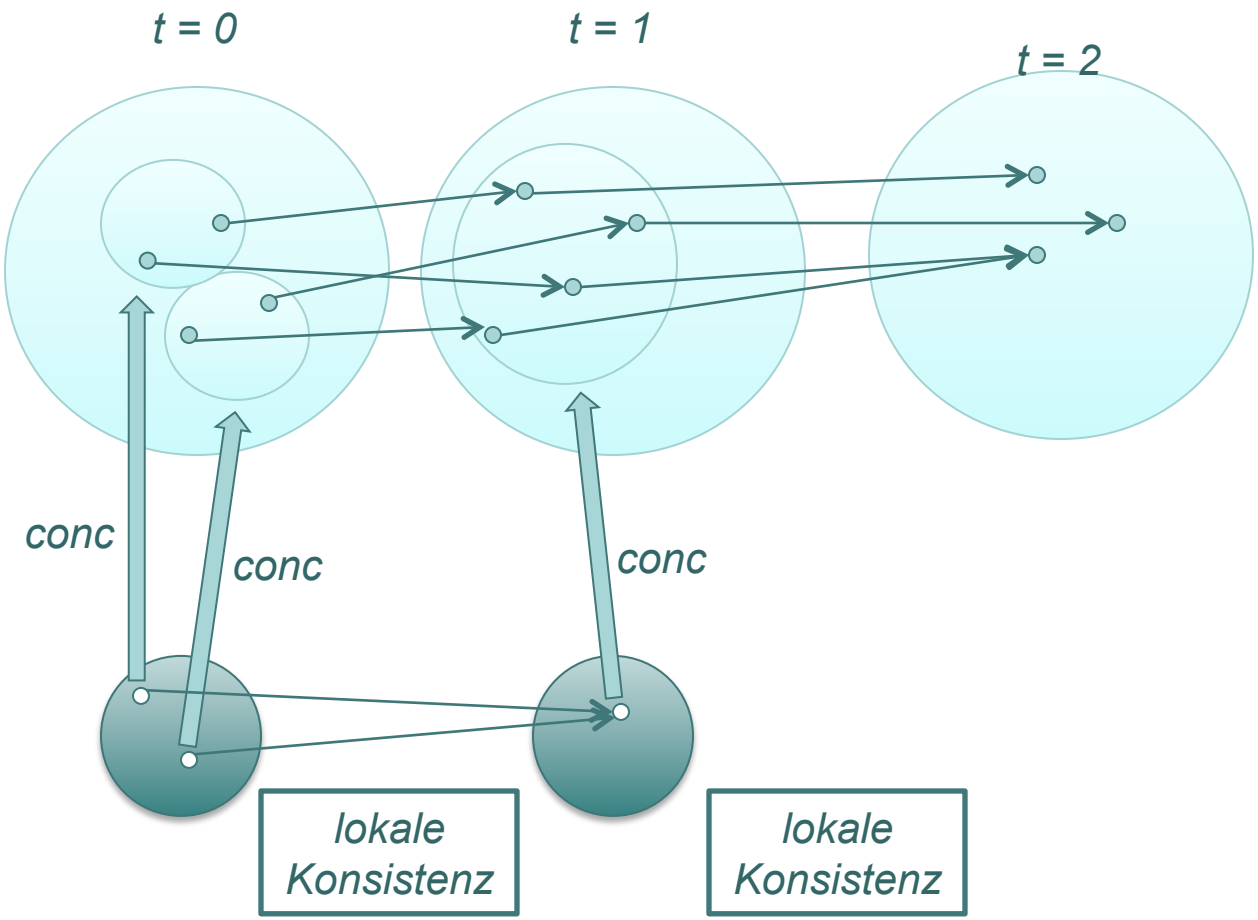


Wie wird „sichere Approximation“ erreicht?

Konkretes Modell:
Zyklengenaues Modell der Hardware und Software

sichere Approximation

Abstrakteres Modell:
Abstraktion der HW + SW

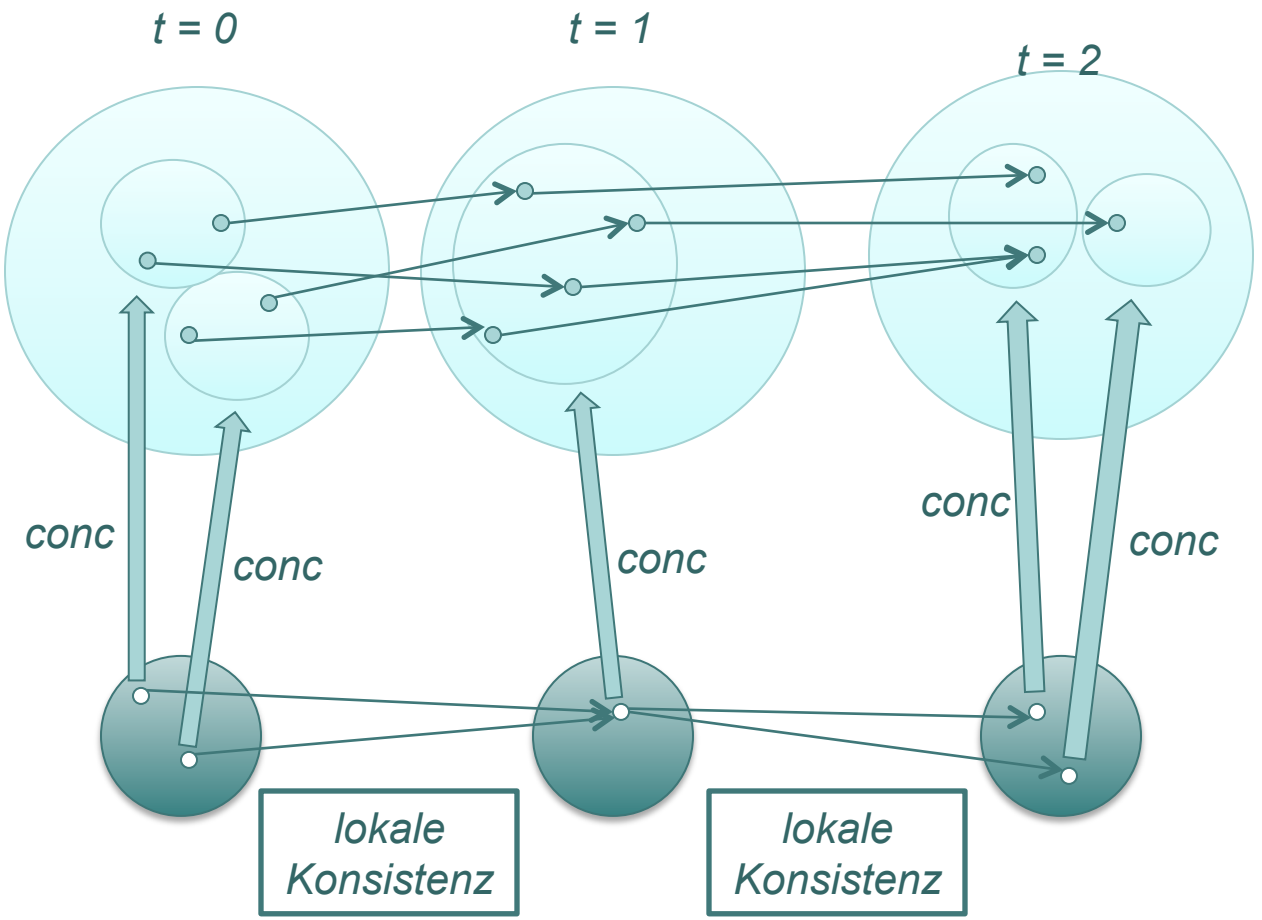


Wie wird „sichere Approximation“ erreicht?

Konkretes Modell:
Zyklengenaues Modell der Hardware und Software

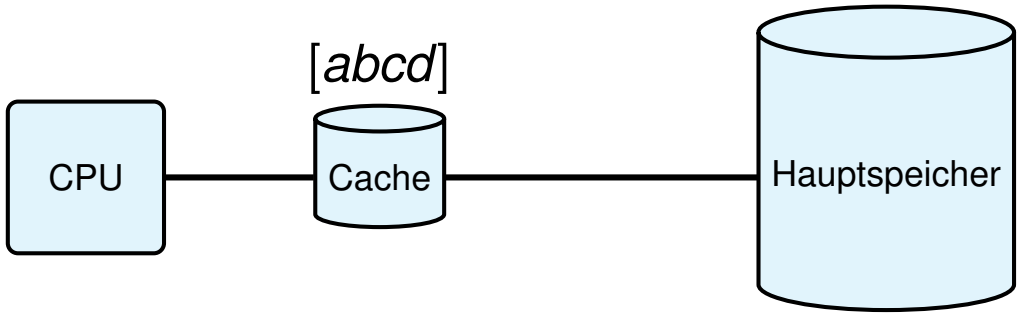
sichere Approximation

Abstrakteres Modell:
Abstraktion der HW + SW



Sichere Approximation am Beispiel von Caches

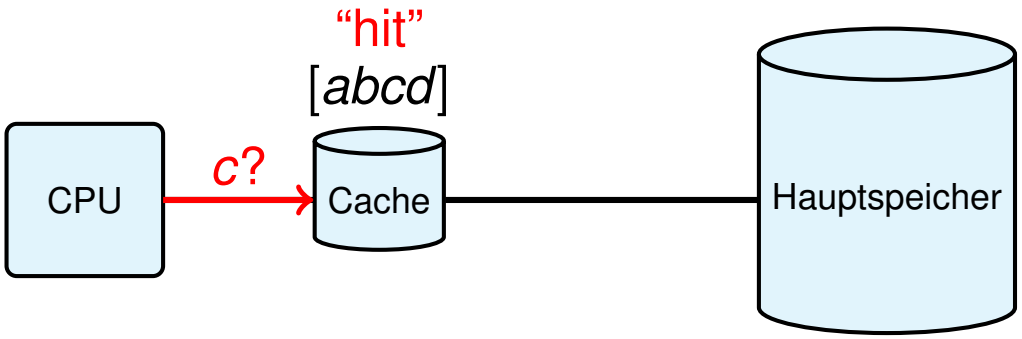
- Cache: Kleiner, aber schneller Speicher der einen Teil des Hauptspeichers zwischenspeichert



- Ersetzungsstrategie entscheidet welche Speicherblöcke im Cache gehalten werden
  Hier: Least-Recently-Used

Sichere Approximation am Beispiel von Caches

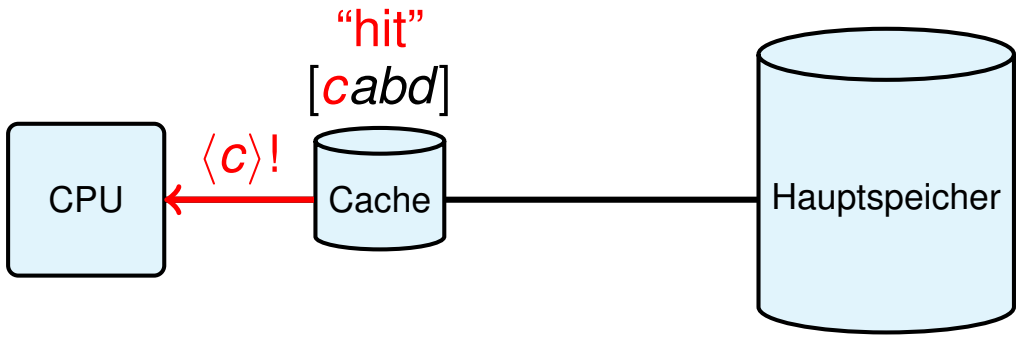
- Cache: Kleiner, aber schneller Speicher der einen Teil des Hauptspeichers zwischenspeichert



- Ersetzungsstrategie entscheidet welche Speicherblöcke im Cache gehalten werden
  Hier: Least-Recently-Used

Sichere Approximation am Beispiel von Caches

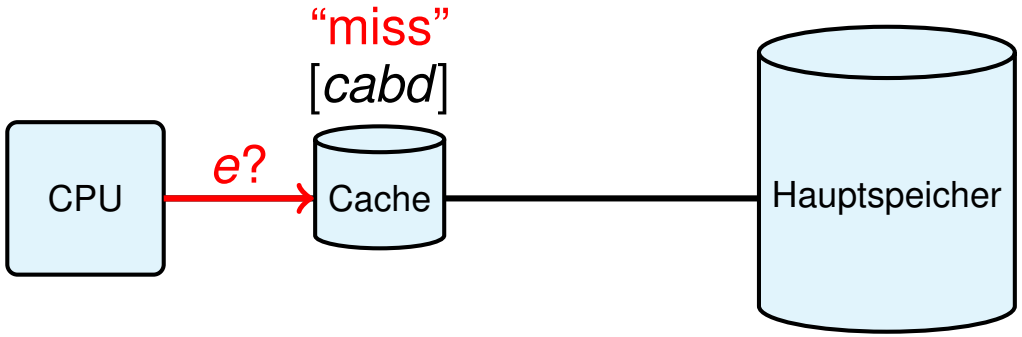
- Cache: Kleiner, aber schneller Speicher der einen Teil des Hauptspeichers zwischenspeichert



- Ersetzungsstrategie entscheidet welche Speicherblöcke im Cache gehalten werden
  Hier: Least-Recently-Used

Sichere Approximation am Beispiel von Caches

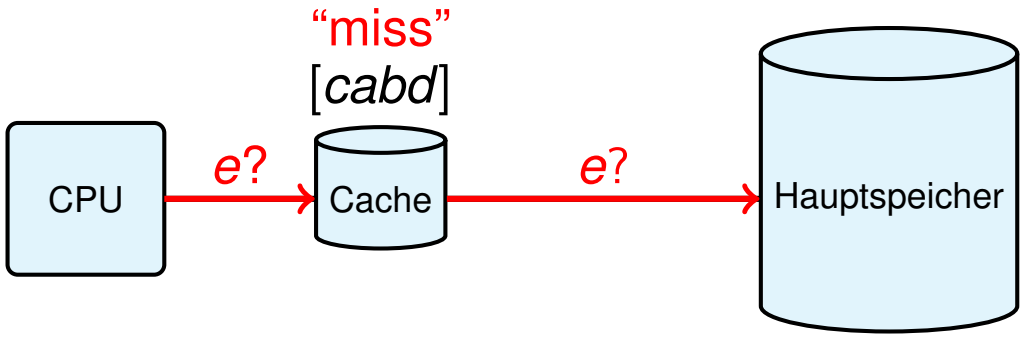
- Cache: Kleiner, aber schneller Speicher der einen Teil des Hauptspeichers zwischenspeichert



- Ersetzungsstrategie entscheidet welche Speicherblöcke im Cache gehalten werden
  Hier: Least-Recently-Used

Sichere Approximation am Beispiel von Caches

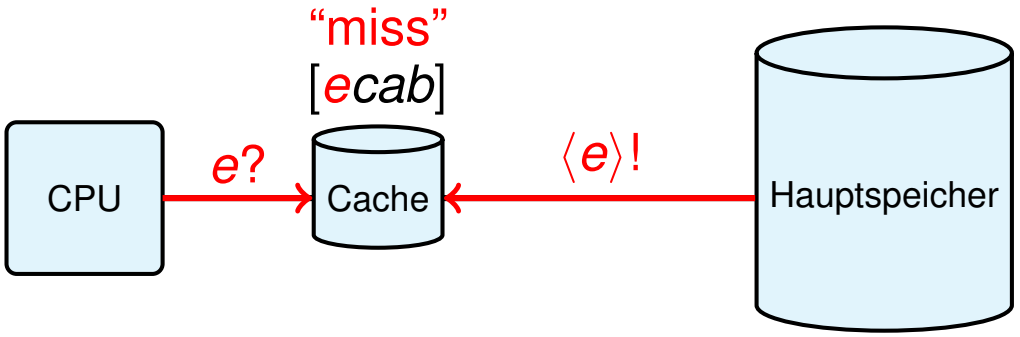
- Cache: Kleiner, aber schneller Speicher der einen Teil des Hauptspeichers zwischenspeichert



- Ersetzungsstrategie entscheidet welche Speicherblöcke im Cache gehalten werden
  Hier: Least-Recently-Used

Sichere Approximation am Beispiel von Caches

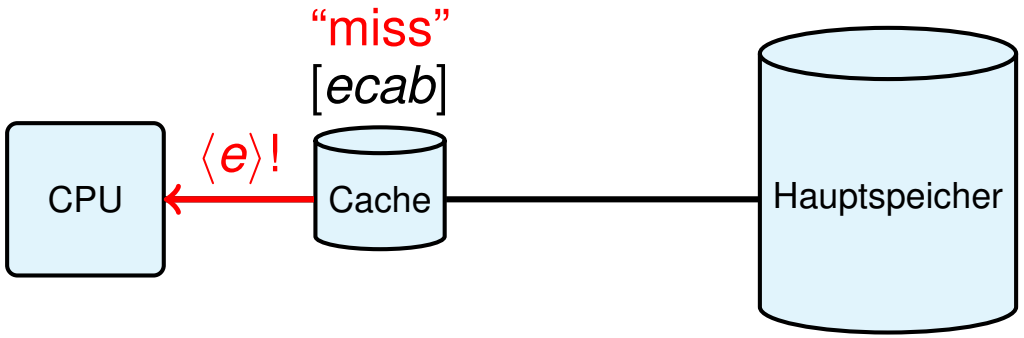
- Cache: Kleiner, aber schneller Speicher der einen Teil des Hauptspeichers zwischenspeichert



- Ersetzungsstrategie entscheidet welche Speicherblöcke im Cache gehalten werden
  Hier: Least-Recently-Used

Sichere Approximation am Beispiel von Caches

- Cache: Kleiner, aber schneller Speicher der einen Teil des Hauptspeichers zwischenspeichert



- Ersetzungsstrategie entscheidet welche Speicherblöcke im Cache gehalten werden
  Hier: Least-Recently-Used

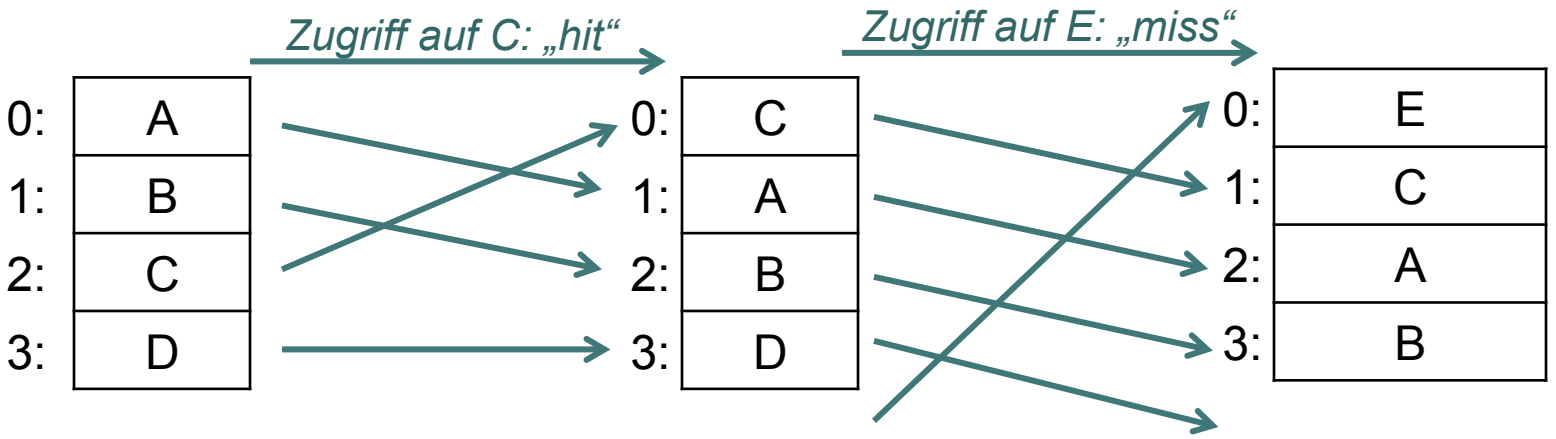
Konkrete Cache-Zustände

Most-recently-used:	0:	A
	1:	B
	2:	C
Least-recently-used:	3:	D

Most-recently-used:	0:	E
	1:	B
	2:	F
Least-recently-used:	3:	D

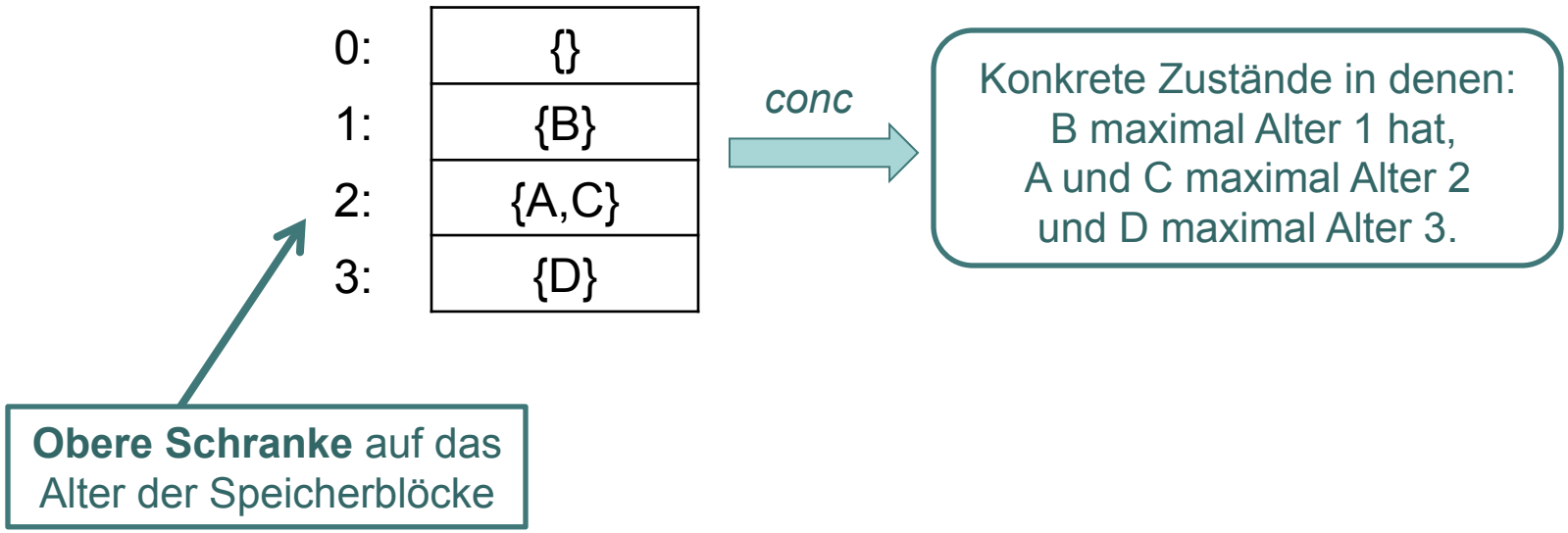
Cache-Verhalten

Wie ändert sich ein Cache-Zustand bei einem Speicherzugriff?



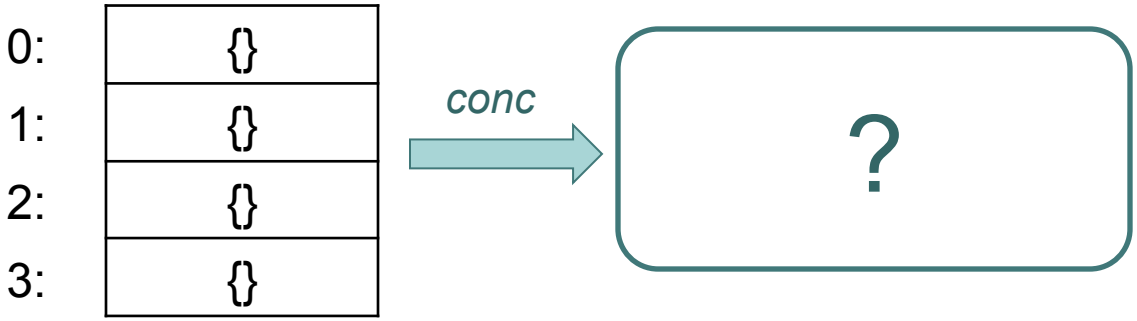
Abstrakte Cache-Zustände

Repräsentieren kompakt Mengen von konkrete Cache-Zuständen:



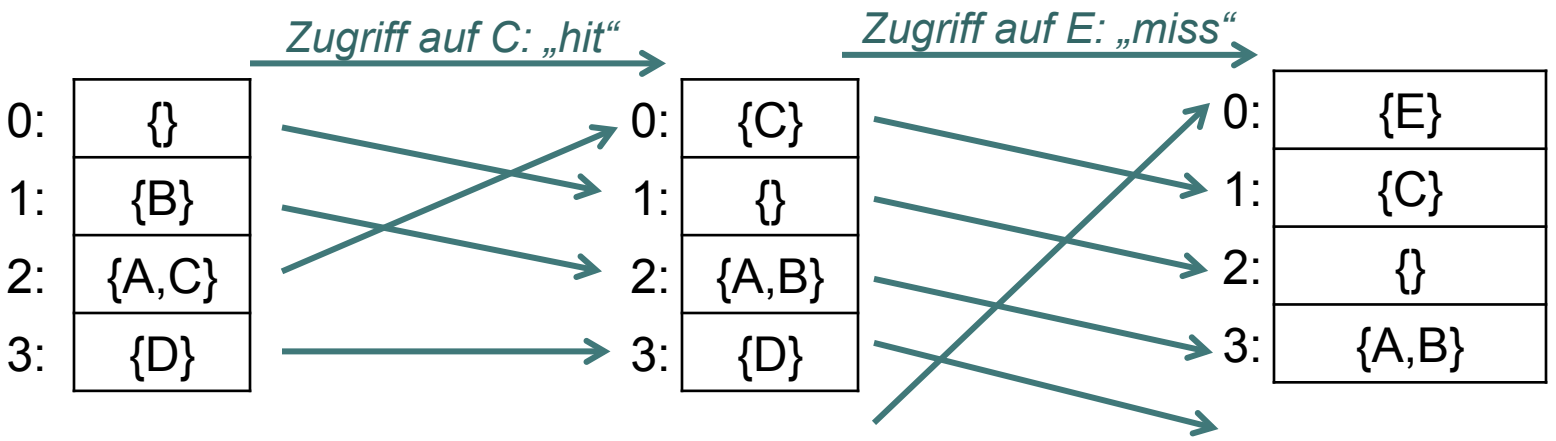
Abstrakte Cache-Zustände

Repräsentieren kompakt Mengen von konkrete Cache-Zuständen:



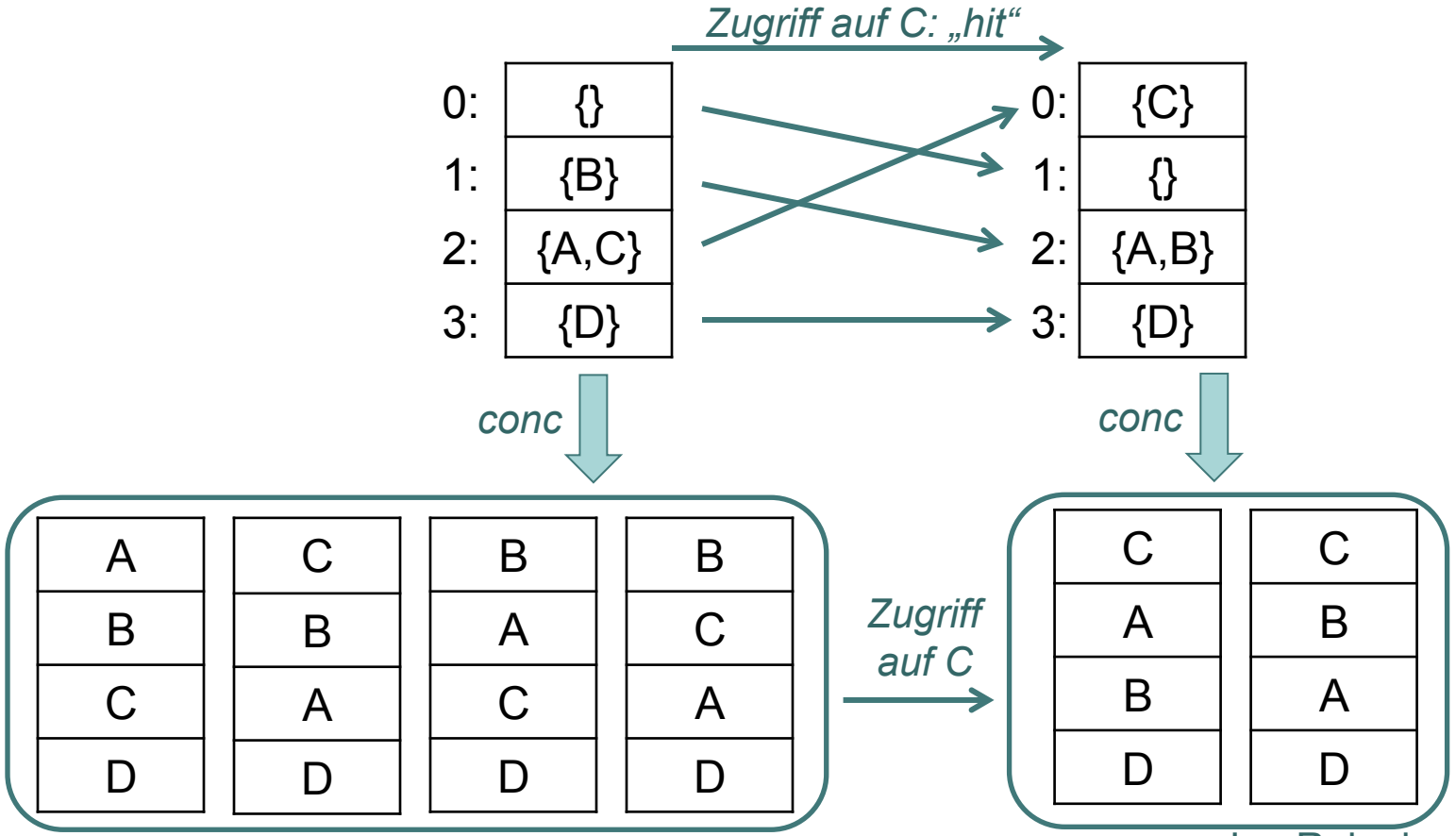
Abstraktes Cache-Verhalten

Wie ändert sich ein abstrakter Cache-Zustand bei einem Speicherzugriff?

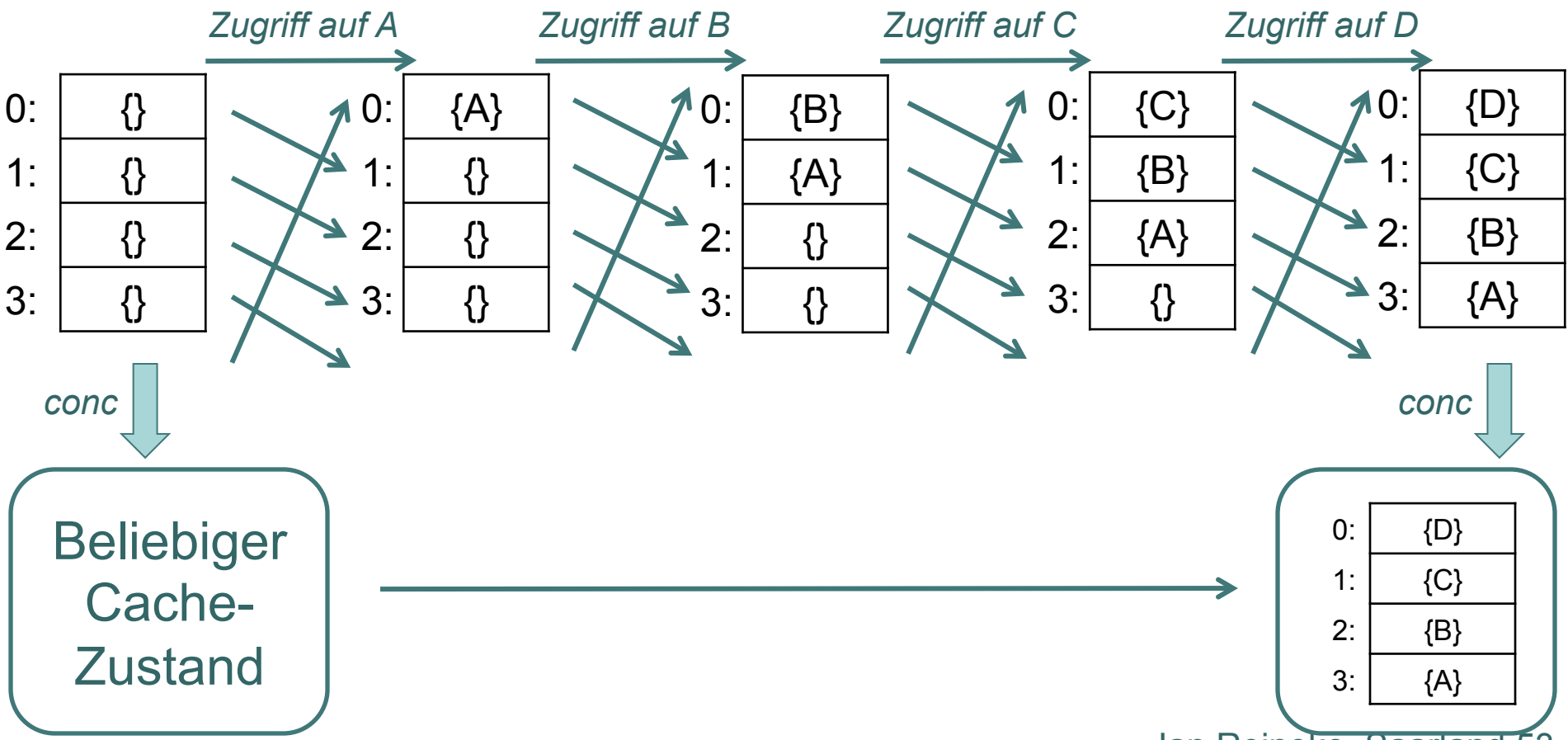


Abstraktes Cache-Verhalten

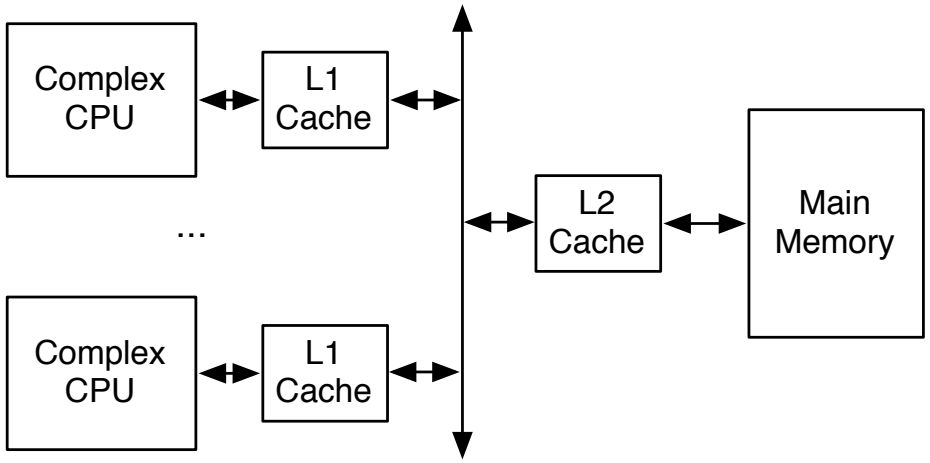
Das abstrakte Cache-Verhalten ist „lokal konsistent“ zum konkreten Cache-Verhalten.



Caches mit LRU-Ersetzungsstrategie sind sehr „robust“



Alles gut? Aktuelle Herausforderungen



Alles gut? Aktuelle Herausforderungen

Private Caches

Präzise & effiziente Abstraktionen für

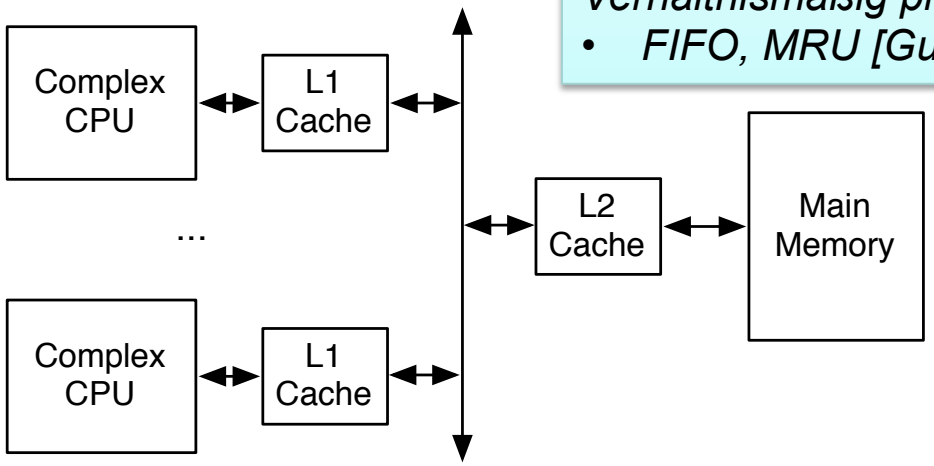
- *LRU [Ferdinand, 1999]*

Weniger präzise aber effiziente Abstraktionen für

- *FIFO, PLRU, MRU [Grund and Reineke, 2008-2011]*

Verhältnismäßig präzise quantitative Analysen für

- *FIFO, MRU [Guan et al., 2012-2014]*



Geteilte Resourcen in Multicores

Herausforderung: Interferenz auf geteilten Ressourcen

→ Ausführungszeit hängt von anderen Tasks ab

Komplexe Pipelines

Präzise aber sehr ineffiziente Analysen; wenige Abstraktion

Fazit

- Zukunftsvorhersagen über reale Systeme basieren auf Modellen:
  - sichere Vorhersagen erfordern wahrheitsgetreue Modelle
  - Approximation um Effizienz zu erreichen
    - Lokale Konsistenz garantiert sichere Approximation

Vielen Dank für die Aufmerksamkeit!