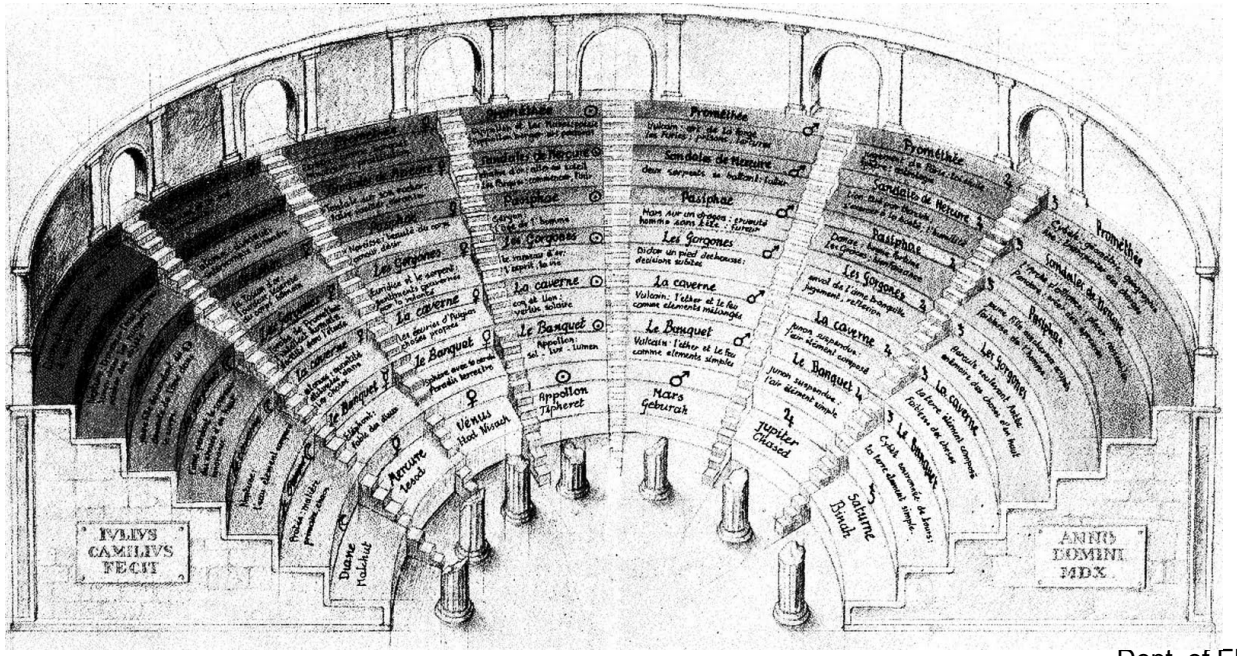
Projektseminar Softwaresysteme

Themenvorstellung und Kick-Off
Sommersemester 2019

TECHNISCHE UNIVERSITÄT DARMSTADT



ES Real-Time Systems Lab
Prof. Dr. rer. nat. Andy Schürr
Dept. of Electrical Engineering and Information Technology
Dept. of Computer Science (adjunct Professor)

Lars Luthmann

lars.luthmann@es.tu.darmstadt.de

www.es.tu-darmstadt.de

Überblick

TECHNISCHE UNIVERSITÄT DARMSTADT

- Für welche Studiengänge von Interesse
  - ETiT, DT, IST, Informatik
- Webseite der Veranstaltung
  - http://www.es.tu-darmstadt.de/lehre/aktuelle-veranstaltungen/projektseminar-softwaresysteme/
- Ziele
  - Praxisorientierte Erfahrungen in der Softwareentwicklung
  - Einblick in Modellierungs- und modellbasierten Softwareentwicklungstechniken

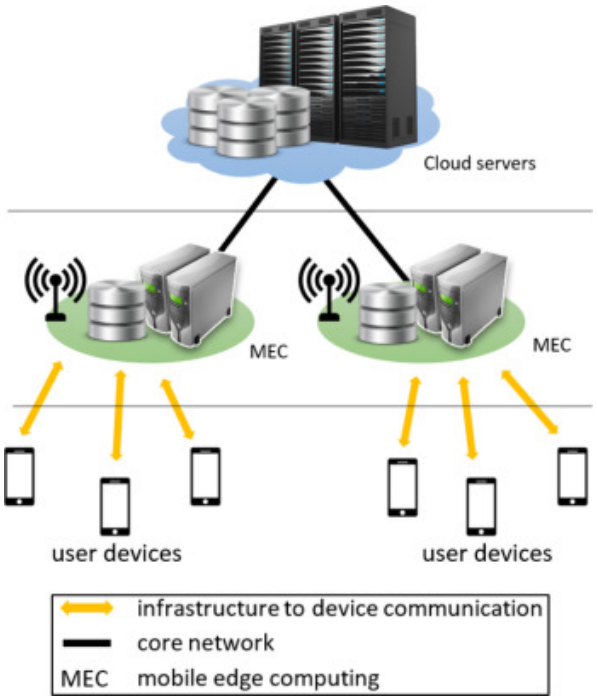
Thema 1

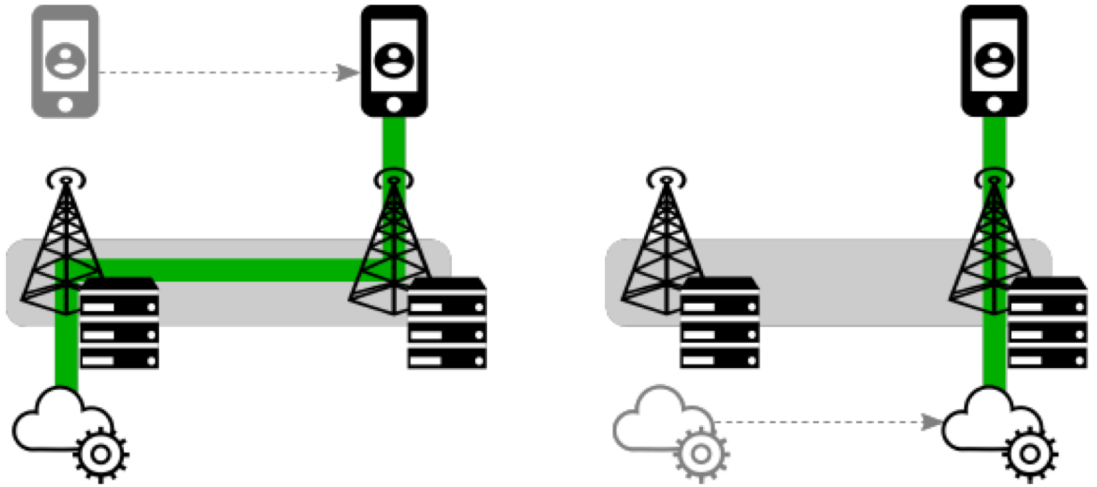
TECHNISCHE UNIVERSITÄT DARMSTADT

Mobile Edge Computing Scenario

Betreuer:
Stefan Tomaszek

Mobile Edge Computing (MEC)





[1] Brandherm, F., Wang, L., & Mühlhäuser, M. (Accepted/In press). *A Learning-based Framework for Optimizing Service Migration in Mobile Edge Clouds*. 1-6. Paper presented at The 2nd International Workshop on Edge Systems, Analytics and Networking, Dresden, Germany.

Ziel

TECHNISCHE UNIVERSITÄT DARMSTADT

- Erstellung eines Metamodells für MEC Szenarien
- Implementierung der Modelltransformationen für das Metamodell
  - Z.B. Erstellung eines Knotens oder Kante
- Implementierung eines Generators für das MEC Szenario aus dem Paper von Brandherm et al. [1]
  - Erstellung einer Spezifikation für dieses Szenario

Generell:

- Tests für alle Implementierungen
- Ausführliche Dokumentation und Spezifikation

[1] Brandherm, F., Wang, L., & Mühlhäuser, M. (Accepted/In press). *A Learning-based Framework for Optimizing Service Migration in Mobile Edge Clouds*. 1-6. Paper presented at The 2nd International Workshop on Edge Systems, Analytics and Networking, Dresden, Germany.

Zusammenfassung

TECHNISCHE UNIVERSITÄT DARMSTADT

Voraussetzungen:

- Sehr gute Java Kenntnisse
- JUnit
- (Sehr hilfreich) eMoflon, Metamodellierung
- (Wünschenswert) Kenntnisse: Maven, Git

Aufgabe / Ziele:

1. **Einarbeitung** in das Paper/Szenario und das Rahmenwerk
2. **Spezifikationen festlegen**
   - Metamodell und Spezifikation für dieses Szenario festlegen
   - JUnit-Tests für die Modelltransformationen und das Szenario erstellen
3. **Entwicklung und Implementierung**
   - Metamodell und Modelltransformationen implementieren
   - Generator für dieses Szenario implementieren

Thema 2

TECHNISCHE UNIVERSITÄT DARMSTADT

Implementierung eines Benchmark-Moduls für eMoflon

Betreuer:
Lars Fritsche

Implementierung eines Benchmark-Moduls für eMoflon

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Einarbeitung in das regelbasierte Werkzeug eMoflon (www.emoflon.org)

2. Einarbeitung in EMF und modellbasierte Entwicklung

3. Implementierung des Benchmark-Moduls
   - GUI zur Auswahl und Konfiguration der Transformationen
   - Automatische Ausführung
   - Daten aufbereiten (Plotten, speichern, ….)

Thema 3

TECHNISCHE UNIVERSITÄT DARMSTADT

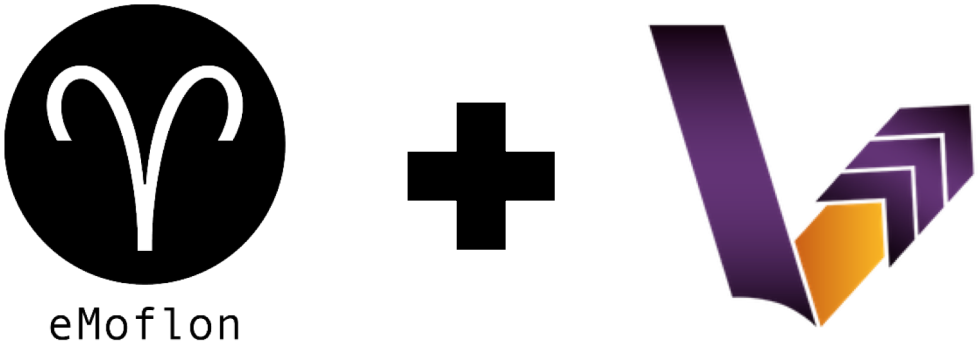
Inkrementelles Pattern Matching
Erweiterung von eMoflon mit VIATRA

Betreuer:
Sebastian Ehmes

Inkrementelles Pattern Matching
Erweiterung von eMoflon mit VIATRA

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Einarbeitung in das regelbasierte Werkzeug Viatra
2. Einarbeitung in das regelbasierte Werkzeug eMoflon (www.emoflon.org)
3. Einarbeitung in EMF und modellbasierte Entwicklung
4. Integrierung von Viatra in eMoflon



Thema 4

TECHNISCHE UNIVERSITÄT DARMSTADT

Metamorphic Testing

Betreuer:
Sebastian Ruland

Metamorphic Testing

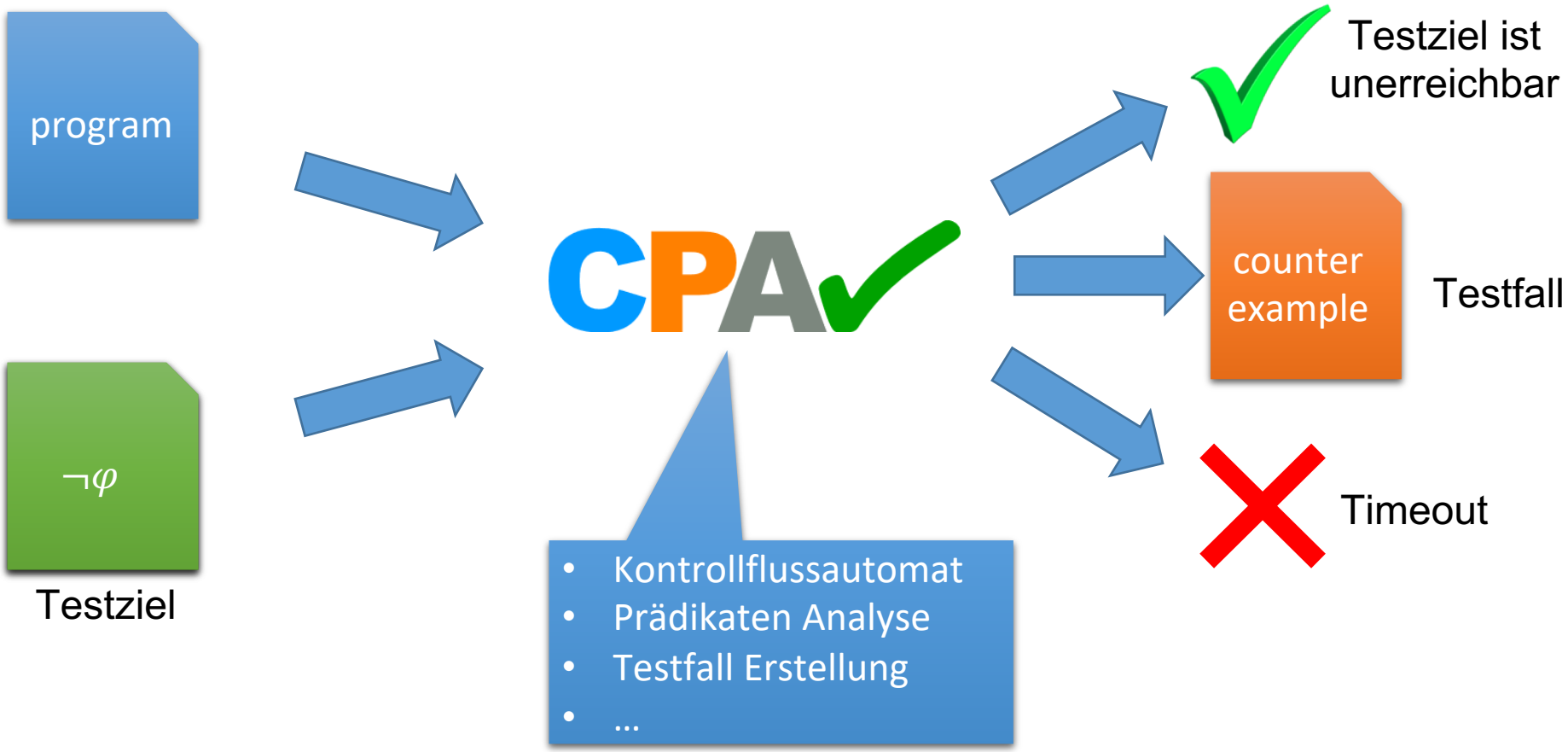
TECHNISCHE UNIVERSITÄT DARMSTADT

- Bei der Testfall Generierung fehlt oft ein sogenanntes Orakel
  - D.h. es kann für einen Test nicht bestimmt werden, ob dieser korrekt durchläuft
- Metamorphic Relations können genutzt werden, um ein Orakel zu erzeugen
- Z.B. wenn ein Fehler im Program durch einen Test erreichbar ist, können weitere Tests erstellt werden, die diesen Fehler ebenfalls erreichen

```
1  if(x > 0 || y < 0){
2  printf("error");
3  }
```

- => erstelle Testfälle, die sowohl durch eine Belegung von „x“ als auch von „y“ den Fehler auslösen

Testfallgenerierung mit CPAchecker



Ziel

TECHNISCHE UNIVERSITÄT DARMSTADT

- Einarbeitung in CPAchecker
- Erweiterung der Testausgabe um Prädikate
- Implementierung eines Tools zur geeigneten Darstellung von Metamorphic Relations

Generell:

- Tests für alle Implementierungen
- Ausführliche Dokumentation und Spezifikation

Voraussetzungen:

- gute Java Kenntnisse
- (wünschenswert) grundlegende C Kenntnisse

Thema 5

TECHNISCHE UNIVERSITÄT DARMSTADT

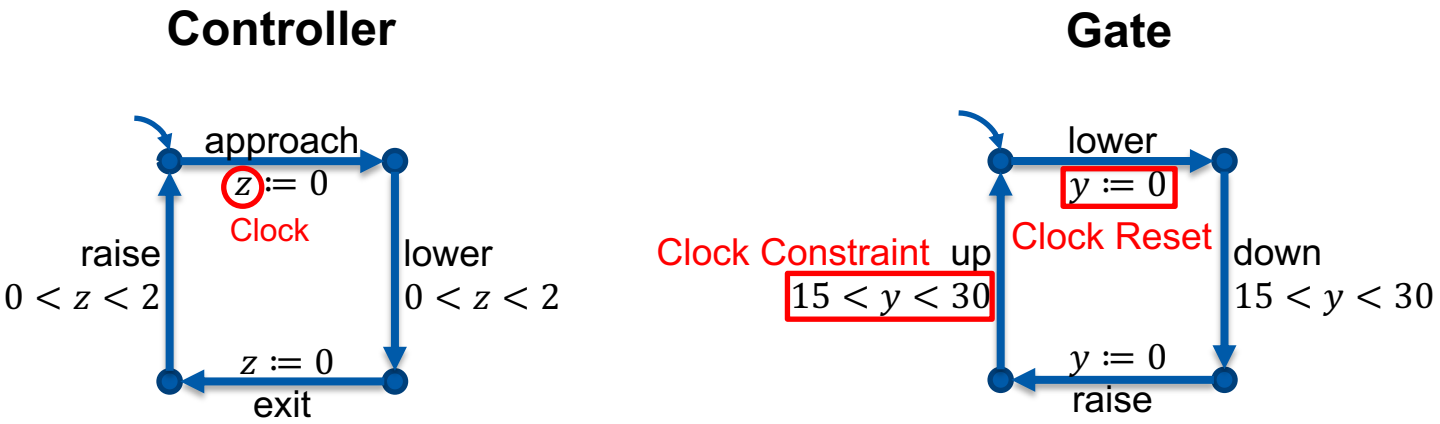
Symbolische Testfallausführung für echtzeitkritische Software-Produktlinien

Betreuer:
Lars Luthmann

Timed Automata

[Alur and Dill (1990): Automata for modeling real-time systems]

TECHNISCHE UNIVERSITÄT DARMSTADT

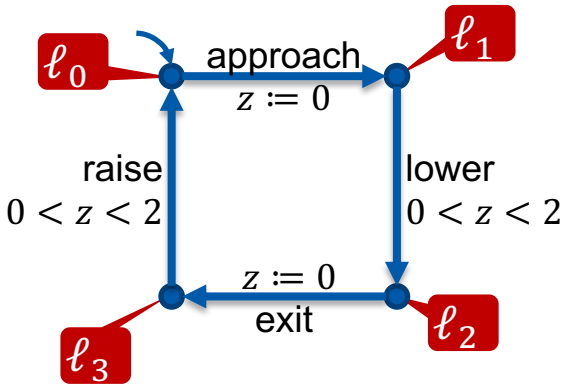


- *Controller* und *Gate* eines Bahnübergangs
- Modellierung von echtzeitkritischem Verhalten mit Timed Automata

Testfallausführung

TECHNISCHE UNIVERSITÄT DARMSTADT

Controller

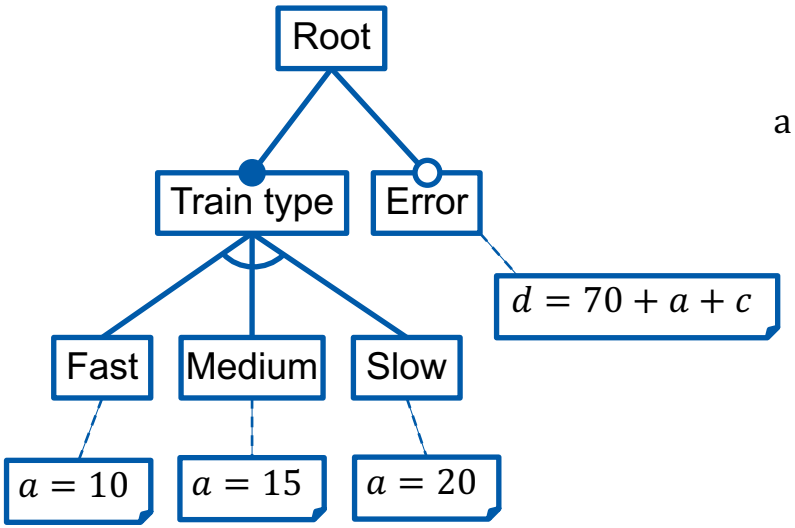


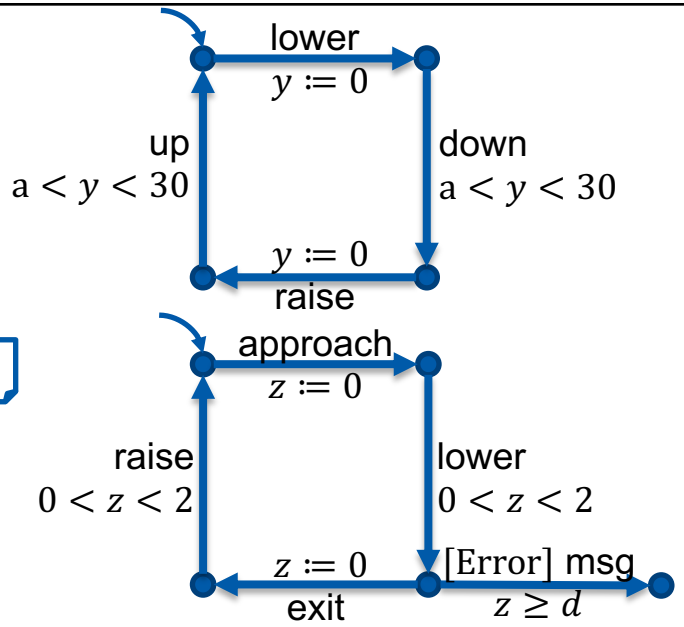
$\langle \ell_0, z = 0 \rangle$
↓ 14
$\langle \ell_0, z = 14 \rangle$
↓ approach
$\langle \ell_1, z = 0 \rangle$
↓ 2

- Testfall: Abfolge von Delays und Aktionen
- Beispiel: (14, approach), (2, lower)
- Testfall schlägt fehl!

Produktlinien von Timed Automata

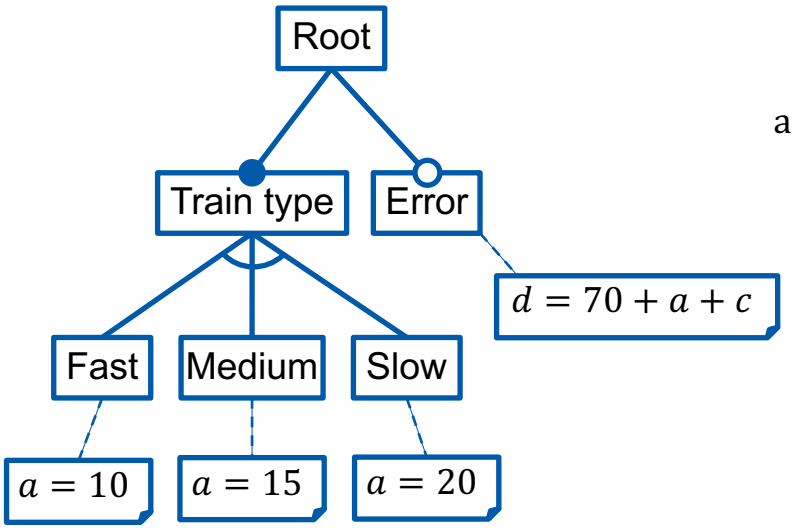
[Luthmann et al. (2017): Modeling and Testing Product Lines with Unbounded Parametric Real-Time Constraints]

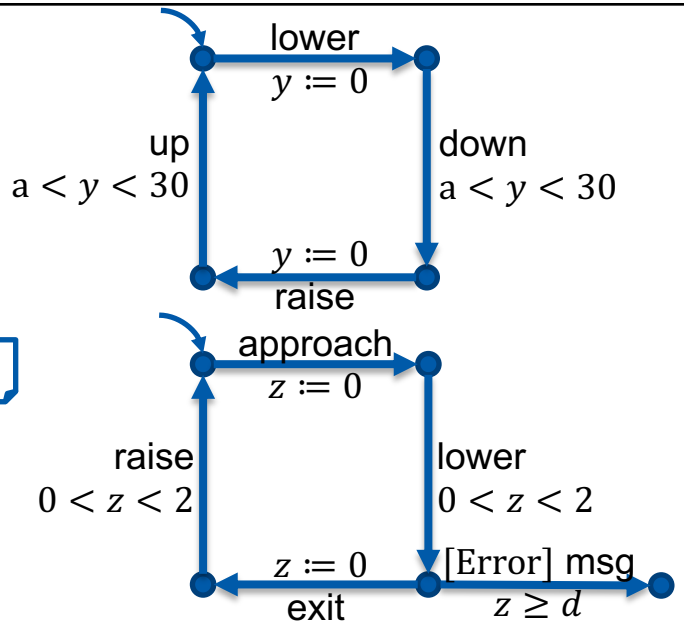




- Produktlinie fasst ähnliche Varianten in einem Modell zusammen
- Feature-Modell beschreibt gültige Konfigurationen
- Configurable Parametric Timed Automata (CoPTA)

Symbolische Testfallausführung





- Symbolischer Testfall: TA-Testfall mit Presence Condition
- Beispiel: (14, approach), (1, lower), (5, msg), $\text{Error} \wedge \text{Fast} \wedge d \geq 0$
- „Aufsammeln“ der Constraints liefert: $\text{Error} \wedge d \geq 0$
- Zusätzliche Prüfung: $(\text{Error} \wedge d \geq 0) \Rightarrow (\text{Error} \wedge \text{Fast} \wedge d \geq 0)$

Aufgaben

TECHNISCHE UNIVERSITÄT DARMSTADT

1. In CoPTA einarbeiten
2. Ausführen von Testfällen auf TA
3. Erweiterung um Features
4. Erweiterung um Parameter
5. ...

- Voraussetzungen: gute Java-Kenntnisse

Nächste Schritte

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Mail an lars.luthmann@es.tu-darmstadt.de bis heute Abend (17.04.)**
  - mit einer priorisierten Liste von **allen Themen**, die in Frage kommen

Nach der Themenverteilung:

- Regelmäßige Treffen mit dem Betreuer
- Inhalt und Format wird vom Betreuer und dem Studenten bestimmt
- Optional: Zwischenvortrag
- Gemeinsame Abschlussveranstaltung (voraussichtlich ~Mitte September)