

(Pro-)Seminar Softwaresystemtechnik (SST)



TECHNISCHE
UNIVERSITÄT
DARMSTADT

(SS 17, Proseminar 2 CP | Seminar 4 CP)



Einführungsveranstaltung



ES Real-Time Systems Lab

Prof. Dr. rer. nat. Andy Schürr

Dept. of Electrical Engineering and Information Technology

Dept. of Computer Science (adjunct Professor)

www.es.tu-darmstadt.de

Sebastian Ruland

sebastian.ruland@es.tu-darmstadt.de

Herzlich Willkommen!

Als Proseminar (mit reduzierten Ansprüchen, 2 CP):

- B.Sc. ETiT (5. Sem.)

Als Seminar (4 CP):

- B.Sc. (5. Sem.) und M.Sc. Informatik (2. Sem.)
- B.Sc. Informationssystemtechnik (5. Sem.)
- Dipl. ETiT (DT, Hauptstudium)
- entsprechende Wirtschaftsstudiengänge
- Sonstige passende Fachrichtung

Was wir von den Teilnehmern erwarten...

- Interesse am Thema + Motivation
- Wille zur Zusammenarbeit mit
 - Betreuer
 - Kommilitonen bei einer Gruppenarbeit
- Wissenschaftliches Vorgehen (unter Anleitung)
- Fristgerechte Abgabe der geforderten Arbeiten
- Teilnahme an *allen* Pflichtveranstaltung



ACHTUNG: Das Seminar ist inhaltlich und vom Umfang her anspruchsvoll!
→ Wir geben uns Mühe bei der Betreuung, und erwarten im Gegenzug von allen Teilnehmern ebenfalls **vollen Einsatz!!!**

- Grundfertigkeiten zur Erstellung einer **wissenschaftlichen Arbeit**
 - Selbständiges Erarbeiten eines Themengebietes (unter Anleitung)
 - Literaturrecherche
 - finden, lesen, verstehen, bewerten
 - Wissenschaftliches Schreiben
 - Gliedern, Zitieren, Formulieren
- Mitwirken am **Reviewprozess**
 - Verwendbares Feedback zu fremden Arbeiten geben
 - Gegenseitige Unterstützung, Schwachstellen identifizieren
- **Präsentation**
 - Aufbereiten, bewerten der Ergebnisse
 - Vorstellen der Ergebnisse
 - Techniken, Stil, Zeiteinteilung, Reden vor der Gruppe



- Heute
 - Themenvorstellung
 - Themenvergabe durch uns
- Während des Semesters
 - Erstellen einer Ausarbeitung
 - (Auf-)Schreiben von (Zwischen-) Ergebnissen
 - Vortrag vorbereiten
 - Regelmäßige Absprachen mit Betreuer!
 - Individuelle Absprachen
 - Fortschritt, Fragen, Feedback, Tipps
- Am Ende des Semesters (**10. + 11. Juli**)
 - Vortrag im Blockseminar → Präsentation + Ausarbeitung liegen bereits vor



Plagiatshinweis – „Abschreiben“ verboten!



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Wir messen der Einhaltung der Grundregeln der wissenschaftlichen Ethik großen Wert bei.
- Mit der Abgabe einer Lösung (Hausaufgabe, Programmierprojekt, Diplomarbeit, etc.) bestätigen Sie, dass (Sie/Ihre Gruppe) (der alleinige Autor/die alleinigen Autoren) des gesamten Materials sind. Falls Ihnen die Verwendung von Fremdmaterial gestattet war, so müssen Sie dessen Quellen deutlich zitiert haben.
- Weiterführende Informationen unter <http://www.es.tu-darmstadt.de/lehre/plagiat/>



Wichtige Links zur Veranstaltung

- **Seminarrichtlinien**
 - <http://www.es.tu-darmstadt.de/fileadmin/download/lehre/Seminarrichtlinien.pdf>
- **Plagiatshinweise**
 - <http://www.es.tu-darmstadt.de/lehre/plagiat/>
- **Schreibkurse**
 - http://www.owl.tu-darmstadt.de/owl_ueber_uns/ueber_uns_1.de.jsp
 - http://www.hda.tu-darmstadt.de/angebote_fuer_studierende_zentral/trainings_fuer_schluesselformen/angebot_startseite_stud_sk/startseite_stud_sk.de.jsp



Zeitplan

Datum	Topic	Abgabe
12.04.	Auftaktveranstaltung	-
07.05.	Gliederung der Ausarbeitung	Betreuer
04.06.	Erste Fassung der Ausarbeitung	Sebastian Ruland
11.06.	Abgabe der Reviews	Sebastian Ruland
25.06.	Endfassung der Ausarbeitung	Sebastian Ruland
02.07 - 06.07.	Probenvorträge	Betreuer
10. + 11.07.	Abschlussvortrag	-

Themenübersicht

Thema	PS
Build your own Graph Transformation Tutorial	X
From Markov Chains to Stochastic Graph Transformation	
In-Network Processing	
The Theory of Hybrid Automata	
Modeling and Testing Product Lines with Unbounded Parametric Real-Time Constraints	X
Grammar-based Blackbox Fuzzing	X
(Grammar-based) Whitebox Fuzzing	
Bidirectional Transformations using QVT	
Bidirectional Model Transformations with TGGs	X
Incremental Pattern Matching using the Rete algorithm	





Themenvorstellungen



Incremental Pattern Matching and Bidirectional Transformations



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)

www.es.tu-darmstadt.de

Sebastian Ruland

sebastian.ruland@es.tu-darmstadt.de

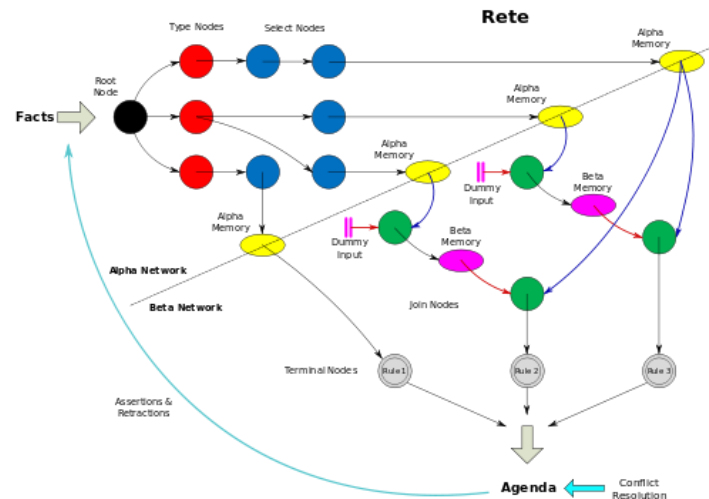
1. Topic – Incremental Pattern Matching with Rete

Idea:

- Split up patterns and create a pattern network
- Define how changes propagate through the network

Advantages:

- Less redundant pattern evaluations
- React quickly to changes without evaluating all patterns from scratch



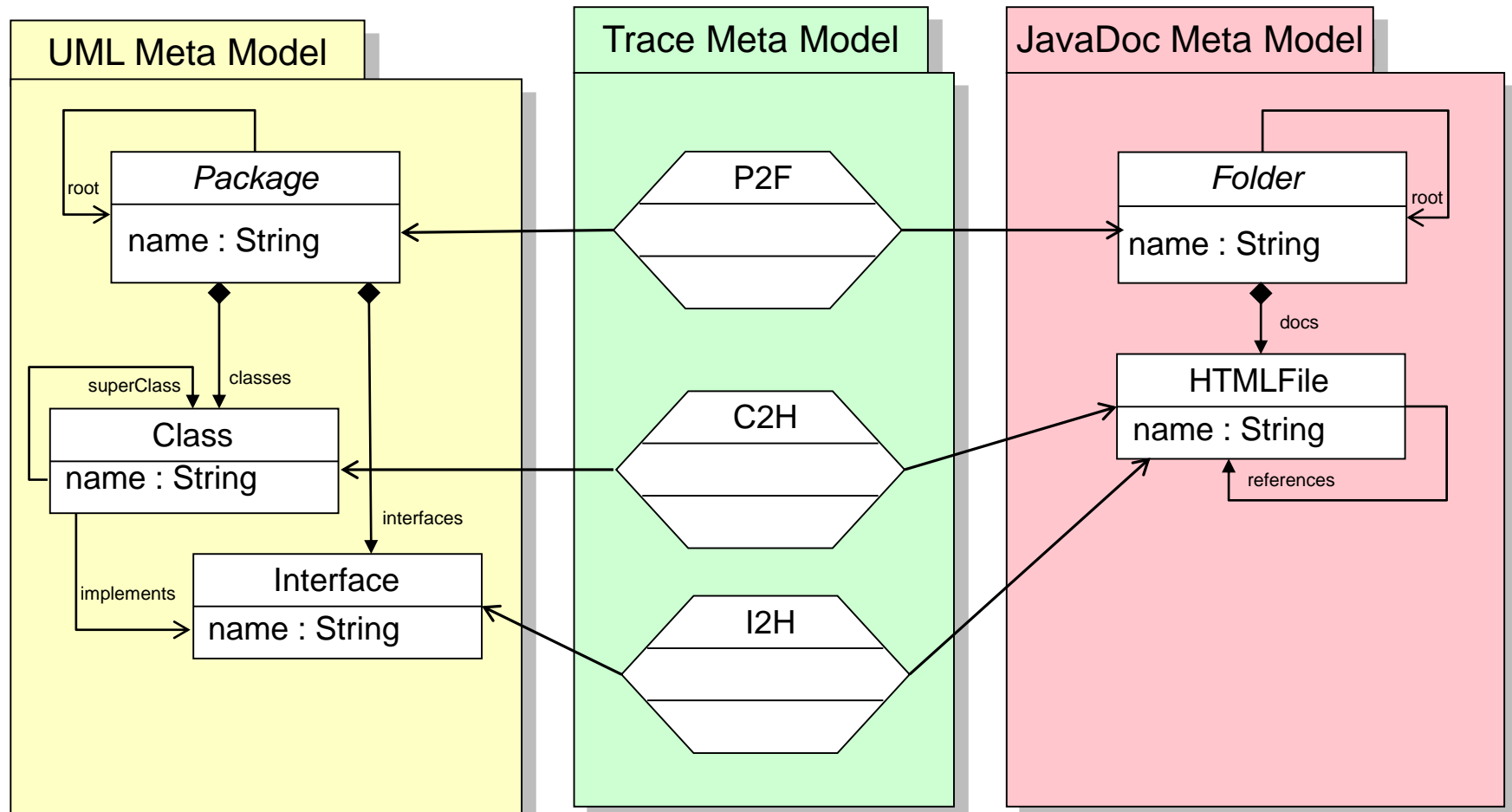
Getting familiar with Pattern Matching and Rete (and Groove)

- What are patterns?
 - What are the components of Rete?
- How do the components of Rete interact?
 - What are the limitations of Rete?
 - What is Groove?
- How does Groove implement Rete?

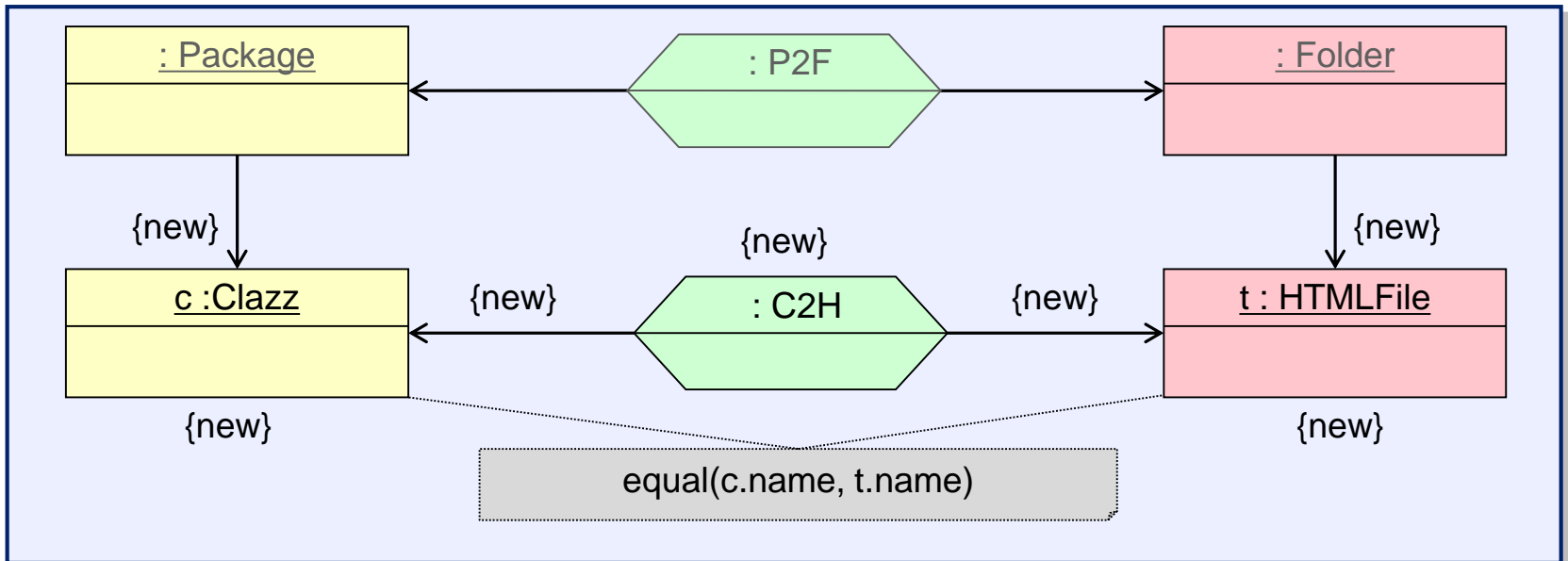
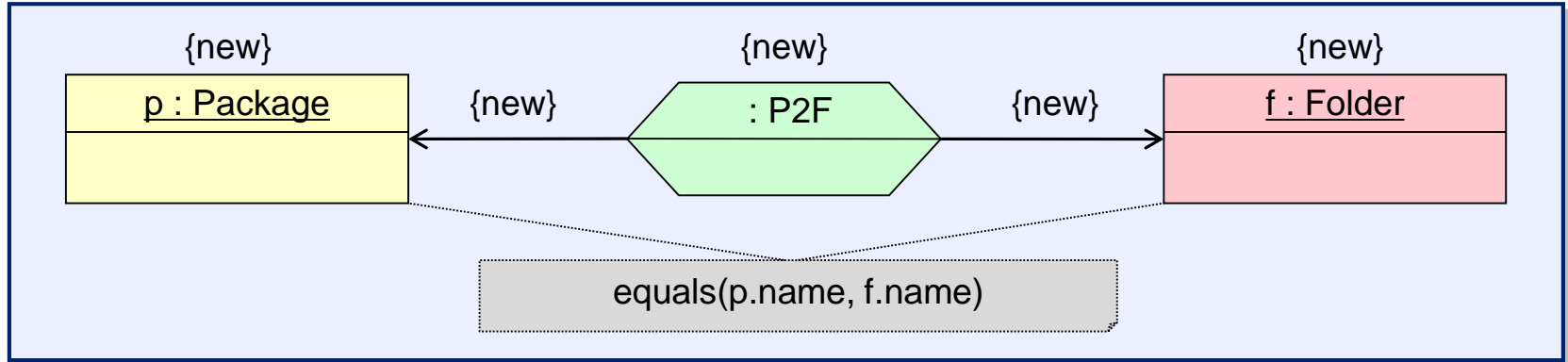
Literature: Lara, Juan & Varro, Daniel. (2010). Preface of the Fourth International Workshop on Graph-Based Tools, GraBaTs 2010.



2. Topic – Bidirectional Model Transformations with TGGs



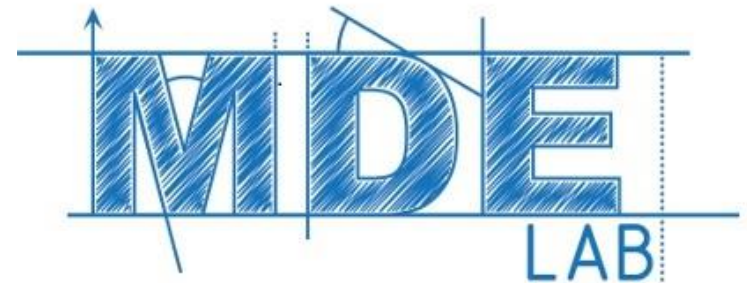
2. Topic – Bidirectional Model Transformations with TGGs



2. Topic – Goals

Getting familiar with Triple Graph Grammars

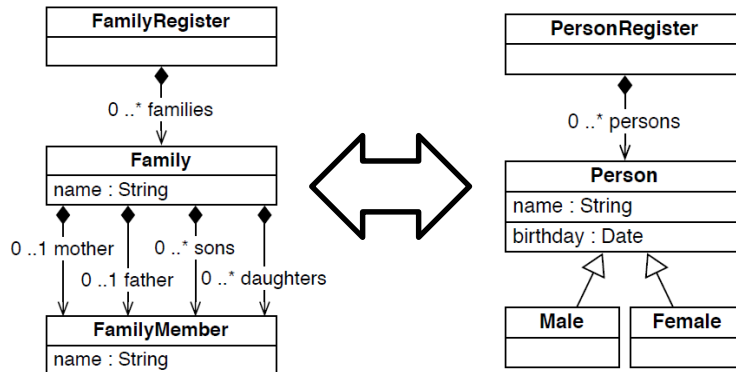
- What are patterns?
- What are graph transformations?
- How to implement a TGG? (using [MoTE](#))
- How to efficiently propagate minor changes?



Literature: Stephan Hildebrandt. *On the Performance and Conformance of Triple Graph Grammar Implementations*. PhD thesis, Hasso Plattner Institute at the University of Potsdam, 2014.



3. Topic – Bidirectional Transformations using QVT-R



```

transformation families2persons ( famDB : Families , perDB : Persons ) {
  top relation FamilyRegister2PersonRegister {
    enforce domain famDB familyRegister : Families :: FamilyRegister {};
    enforce domain perDB personRegister : Persons :: PersonRegister {};
    where {
      Father2Male ( familyRegister , personRegister );
      Son2Male ( familyRegister , personRegister );
      Mother2Female ( familyRegister , personRegister );
      Daughter2Female ( familyRegister , personRegister );
    }
  }
  relation Mother2Female {
    familyName , firstName , fullName : String ;
    enforce domain famDB familyRegister : Families :: FamilyRegister {
      families = family : Families :: Family {
        name = familyName ,
        mother = mother : Families :: FamilyMember { name = firstName }
      }
    };
    enforce domain perDB personRegister : Persons :: PersonRegister {
      persons = female : Persons :: Female { name = fullName }
    };
    where {
      fullName = familyName + ' , ' + firstName ;
      firstName = firstName ( fullName );
      familyName = familyName ( fullName );
    }
  }
  relation Daughter2Female {
    ... -- Like ' Mother2Female ', replace 'mother ' with 'daughters '
    when {
      female . oclIsUndefined ();
    } -- Prevents application in backward direction
    ... -- See ' Mother2Female '
  }
}

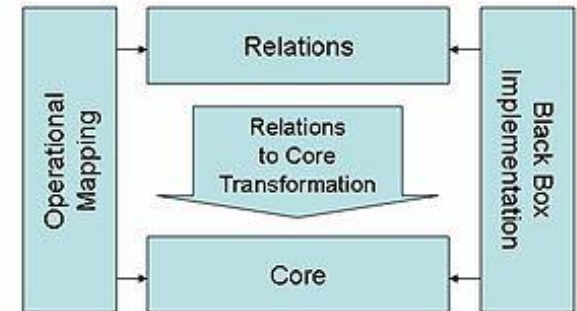
```



3. Topic – Goals

Getting familiar with Query/View/Transformation Relations

- How is this standard defined?
- How to implement a QVT specification?
- How to implement bidirectional transformations?
 - What are the limitations?
 - Point out interesting applications



Literature: Bernhard Westfechtel. Case-based exploration of bidirectional transformations in qvt relations. *Software & Systems Modeling*, pages 1–41, 2016.

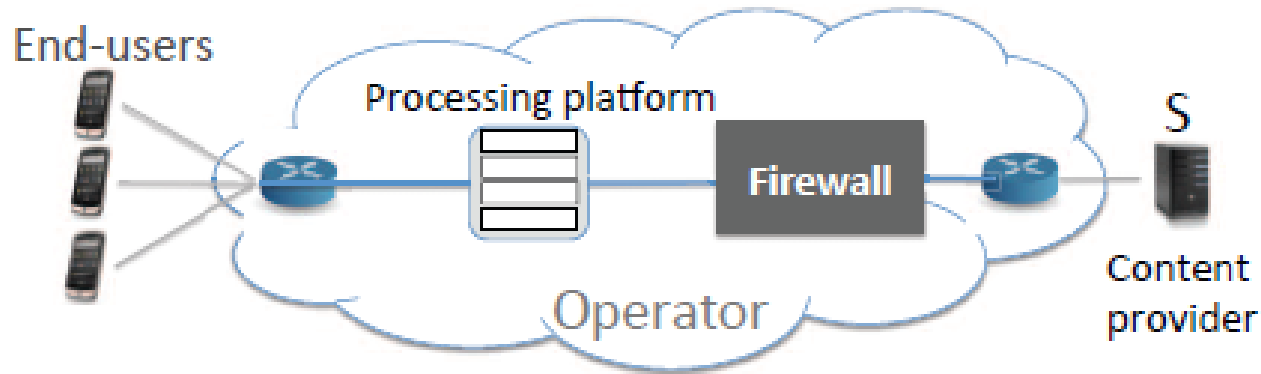
In-Network Processing

SST Seminar – SS 18

Stefan Tomaszek



In-Network Processing



Aufgabe: Beschreibung von der Idee hinter In-Network Processing und der Architektur IN-NET.

Paper:

Radu Stoenescu, Vladimir Andrei Olteanu, Matei Popovici, Mohamed Ahmed, João Martins, Roberto Bifulco, Filipe Manco, Felipe Huici, Georgios Smaragdakis, Mark Handley, Costin Raiciu: In-Net: in-network processing for the masses. EuroSys 2015: 23:1-23:15

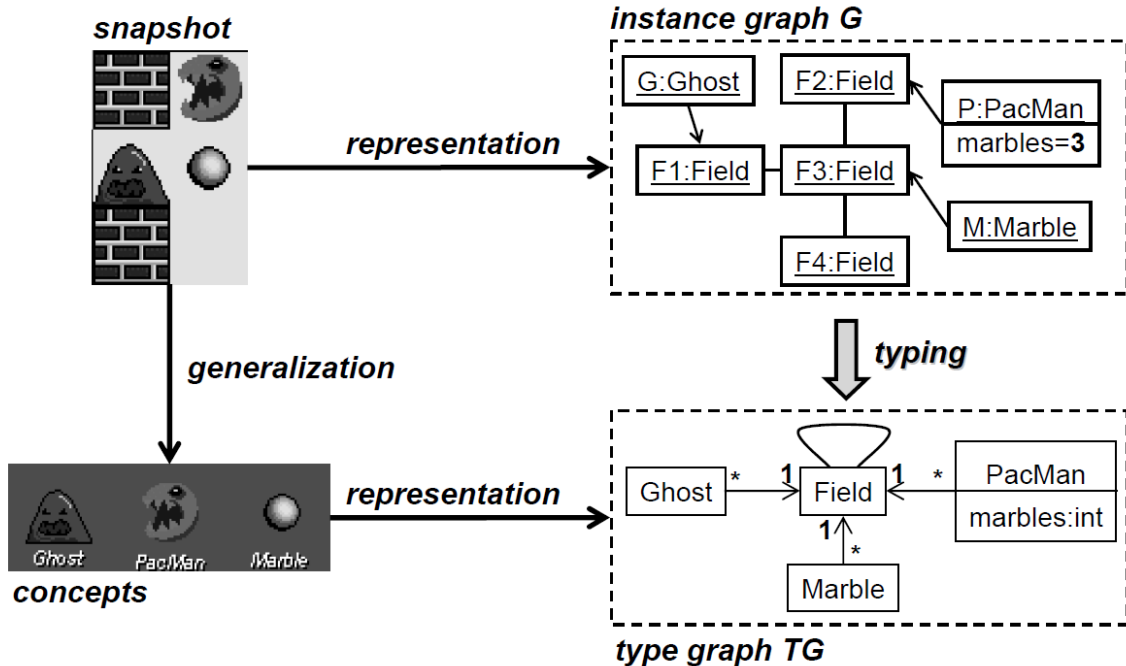
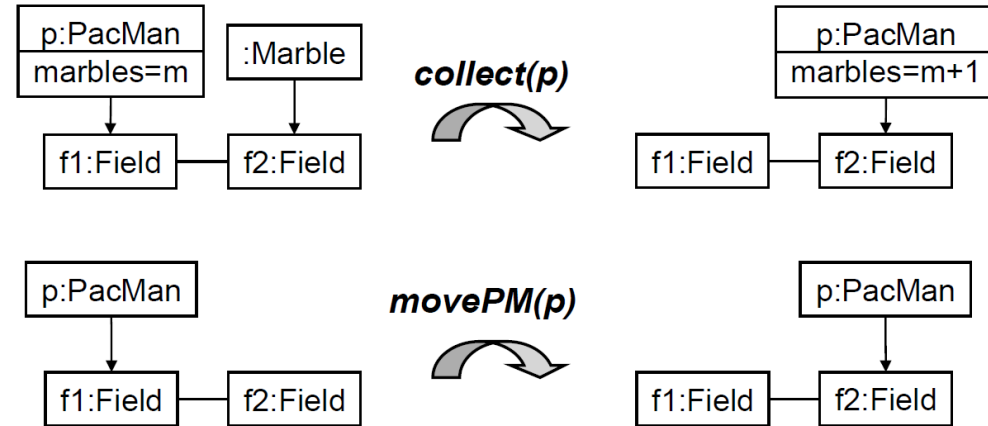
Build your own Graph Transformation Tutorial

Roland Kluge

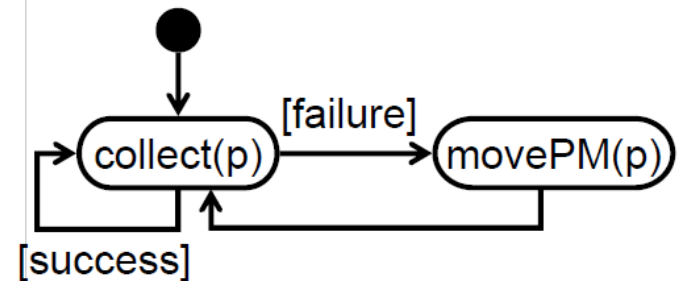


Motivation and goal

- **Goal: Create a tutorial for basic graph transformation concepts**
- **Suitable for:**
Proseminar, Seminar.



Behavior(`p:PacMan`):



(Figures taken from [Heckel 2006])



Work packages and expectations

Work packages

- **Understand the tutorials** by König (2018), Heckel (2006), ...
- Draft your own **running example** (Ideally: **Jointly** with other student)
- **(Short-)paper** illustrates **core concepts of graph transformation**:
metamodel/type graph, match, rules, rule applications, constraints,
(attributes/attribute constraints),...
- **Presentation: Step-by-step** tutorial, maybe with **tool demo** (eMoflon, Henshin,...)

Expectations from my side

- Profound interest in topic
- Self-organization (esp. own example)

Literature

König B., Nolte D., Padberg J., Rensink A. (2018) A Tutorial on Graph Transformation. In: Heckel R., Taentzer G. (eds) Graph Transformation, Specifications, and Nets. Lecture Notes in Computer Science, vol 10800. Springer, Cham (https://doi.org/10.1007/978-3-319-75396-6_5)

Heckel R. (2006) Graph Transformation in a Nutshell. ENTCS, 148(1), pp. 187-198 (<https://doi.org/10.1016/j.entcs.2005.12.018>)

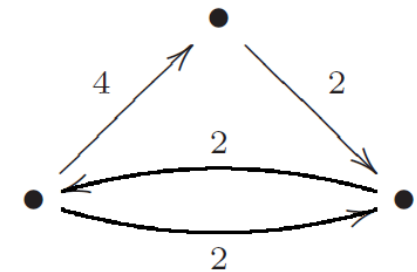
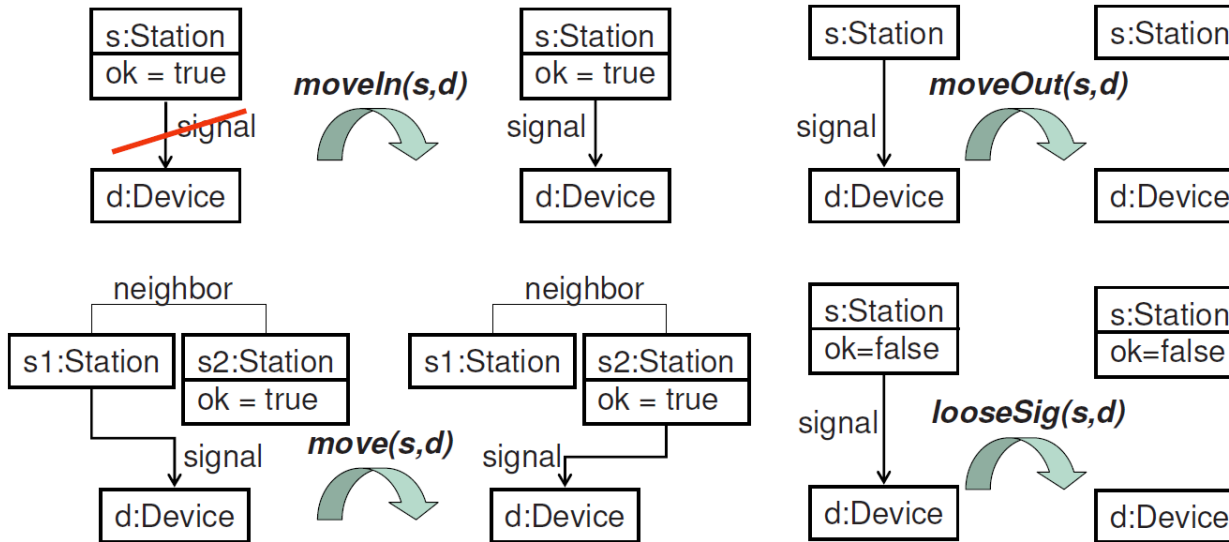
From (Time-Continuous) Markov Chains to Stochastic Graph Transformation

Roland Kluge



Motivation and goal

- **Goal: Introduce fundamentals of Stochastic Graph Transformation (based on Time-Continuous Markov Chains)**
- **Suitable for: Seminar only**



$$\begin{pmatrix} -2 & 0 & 2 \\ 4 & -6 & 2 \\ 0 & 2 & -2 \end{pmatrix}$$

(Figures taken from [Heckel 2004])

rule name	p	rate	$\rho(p)$	rule name	p	rate	$\rho(p)$	rule name	p	rate	$\rho(p)$
repair		500		fail		1		connect		10000	
moveIn		1		moveOut		1		disconnect		10000	
move		100		looseSig		100000		handOver		100000	



Work packages and expectations

Work packages

- Start with **Heckel et al. (2004)** to understand the "mission goal"
 - Work into **graph transformation**
 - Work into **time-continuous Markov chains**
 - Work into **stochastic graph transformation**
- Draft your **own running example** (Ideally: **Jointly** with other student)
- **Paper/presentation**: Guided tour to stochastic graph transformation
 - You may assume a basic understanding of graph transformation.

Expectations from my side

- Profound interest in topic
- Self-organization (esp. own literature research, own running example)



Literature

Heckel R., Lajos G., Menge S. (2004) Stochastic Graph Transformation Systems. In: Ehrig H., Engels G., Parisi-Presicce F., Rozenberg G. (eds) Graph Transformations. ICGT 2004. Lecture Notes in Computer Science, vol 3256. Springer, Berlin, Heidelberg (https://doi.org/10.1007/978-3-540-30203-2_16)

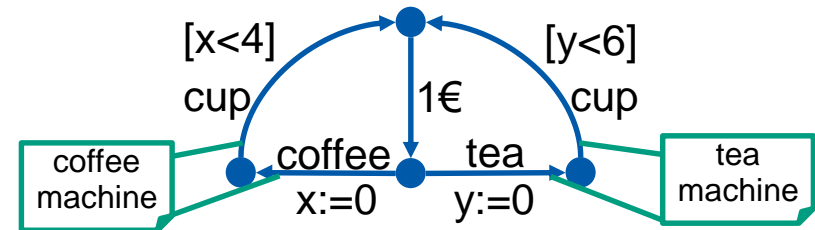
Modeling and Testing Product Lines with Unbounded Parametric Real-Time Constraints

SST Seminar – SS 18
(Proseminar geeignet)

Lars Luthmann



Modeling and Testing Product Lines with Unbounded Parametric Real-Time Constraints



- Modellierung gezeiteter Produktlinien mit potenziell unendlich vielen Varianten
- Automatisierte Testfallgenerierung für gezeitete Produktlinien

Aufgabe:

- Beschreiben des Modells
- Beschreiben der Testfallgenerierung (nur Seminar)



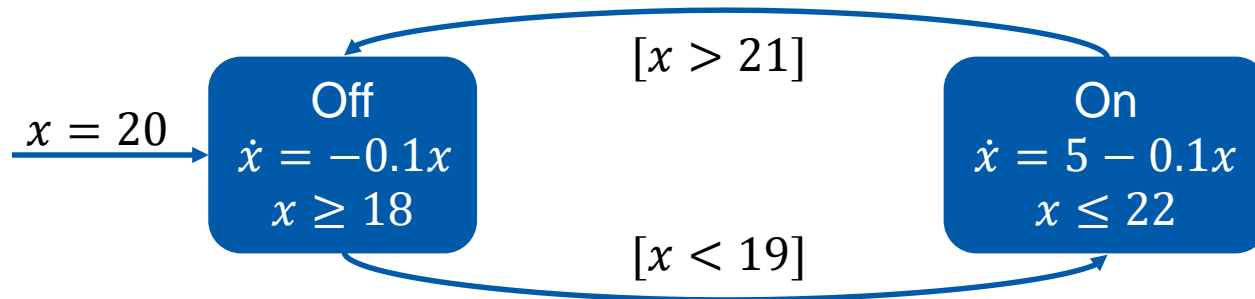
The Theory of Hybrid Automata

SST Seminar – SS 18

Lars Luthmann



The Theory of Hybrid Automata



- Verallgemeinerung von Timed Automata
- Modellierung von kontinuierlichem Verhalten (z.B. Heizung)

Aufgabe:

- Beschreiben des Modells
- Beschreiben von Eigenschaften/Operatoren



Grammar-based Blackbox Fuzzing

SST Seminar – SS 18
(Proseminar geeignet)

Sebastian Ruland

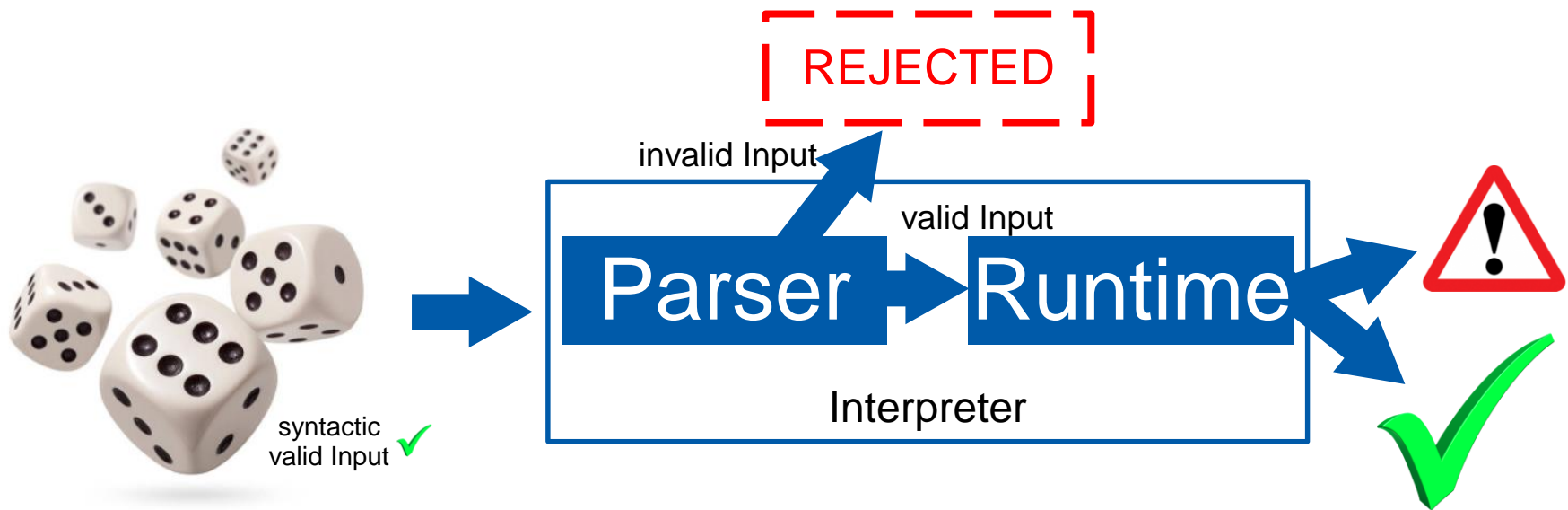


Grammar-based Blackbox Fuzzing

- Problem: Interpreter (z.b. Javascript Interpreter) soll getestet werden
- Methoden: Intelligente, zufällige Generierung von Eingaben
- Praktische Anwendung: Bug Bounties etc.

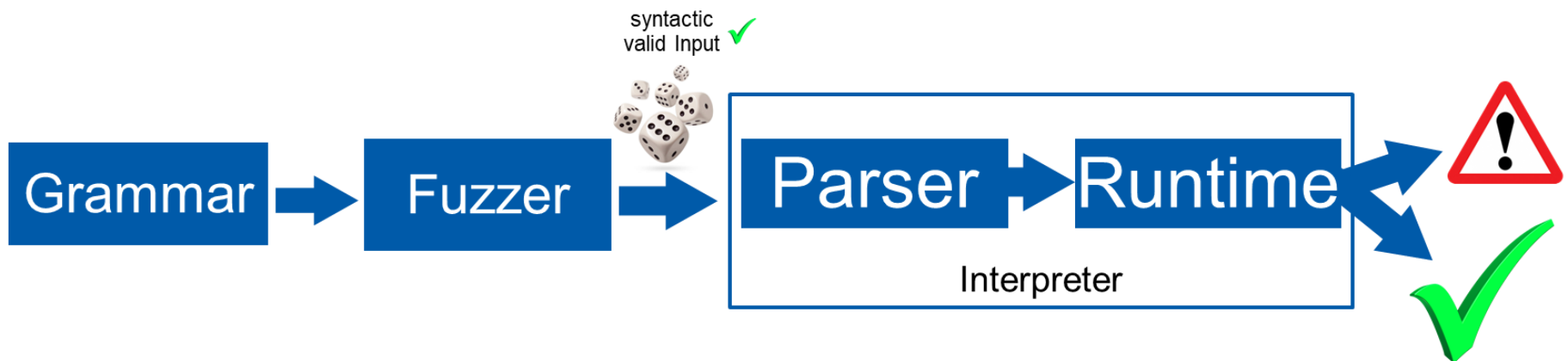


Grammar-based Blackbox Fuzzing



Grammar-based Blackbox Fuzzing

Fuzzing with Code Fragments; Holler et. al



Grammar-based Blackbox Fuzzing

Aufgabe:

- Einführung in Fuzzing
- Einführung in generative Grammatiken
- Beschreibung von Fuzzing Strategien



(Grammar-based) Whitebox Fuzzing

SST Seminar – SS 18

Sebastian Ruland

Grammar-based Whitebox Fuzzing

- Problem: komplexe Strukturen sollen getestet werden
- Methoden: Gesteuerte Generierung von Eingaben
- Praktische Anwendung: Bug Bounties etc.



(Grammar-based) Whitebox Fuzzing

```
void top(char input[4]) {  
    int cnt = 0;  
    if (input[0] == 'b') cnt++;  
    if (input[1] == 'a') cnt++;  
    if (input[2] == 'd') cnt++;  
    if (input[3] == '!') cnt++;  
    if (cnt >= 4) crash();  
}
```

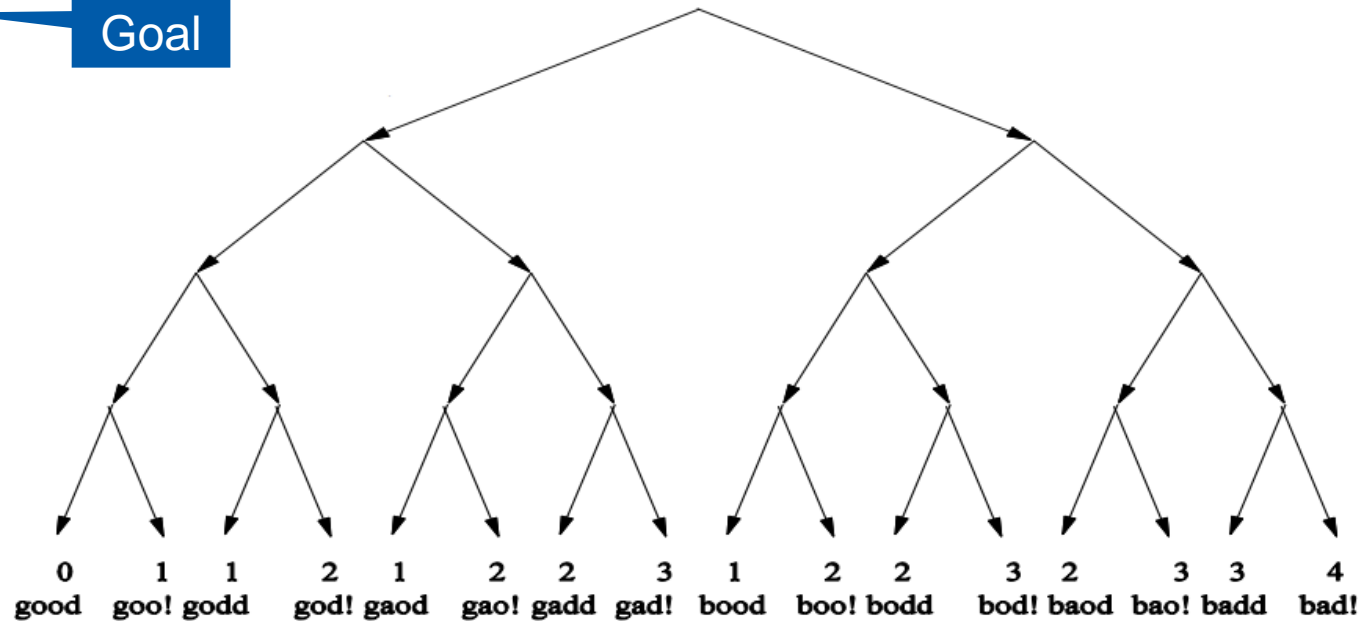

(Grammar-based) Whitebox Fuzzing

Input: „good“

```
void top(char input[4]) {  
    int cnt = 0;  
    if (input[0] == 'b') cnt++;  
    if (input[1] == 'a') cnt++;  
    if (input[2] == 'd') cnt++;  
    if (input[3] == '!') cnt++;  
    if (cnt >= 4) crash();  
}
```

$I_0 \neq 'b'$
 $I_1 \neq 'a'$
 $I_2 \neq 'd'$
 $I_3 \neq '!'$

Goal



(Grammar-based) Whitebox Fuzzing

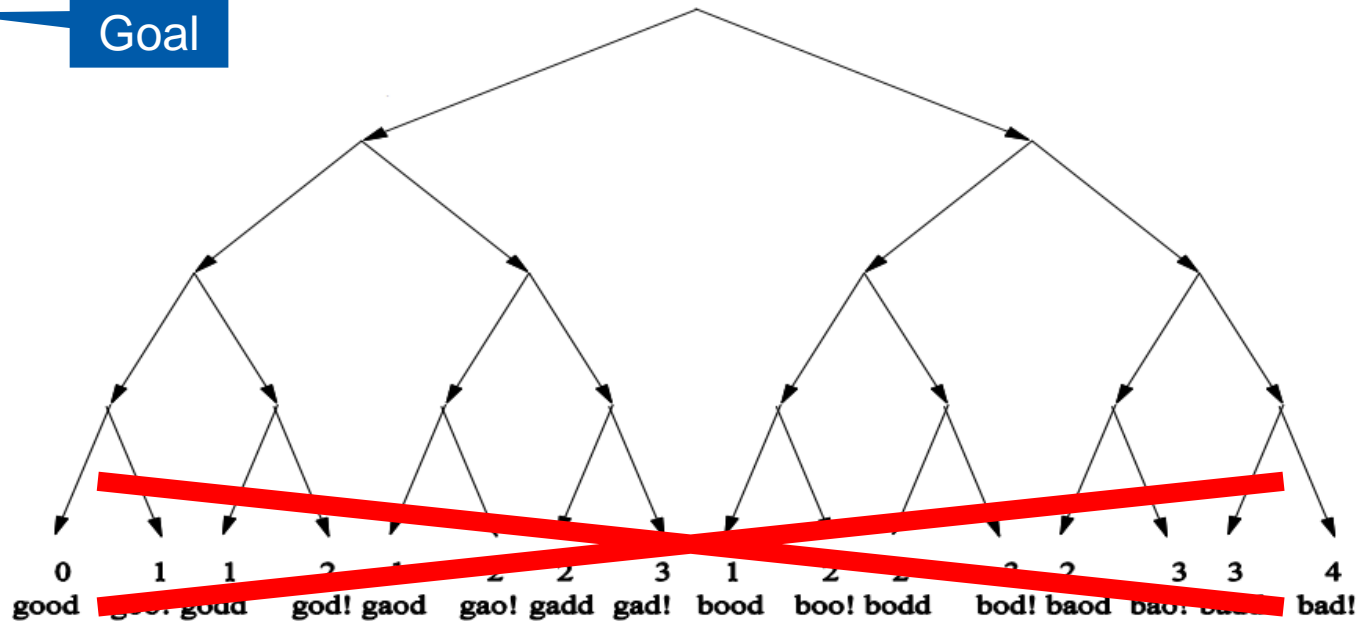
Input: „good“

Grammar:
input: good | bad! | poor

```
void top(char input[4]) {  
    int cnt = 0;  
    if (input[0] == 'b') cnt++;  
    if (input[1] == 'a') cnt++;  
    if (input[2] == 'd') cnt++;  
    if (input[3] == '!') cnt++;  
    if (cnt >= 4) crash();  
}
```

$I_0 \neq 'b'$
 $I_1 \neq 'a'$
 $I_2 \neq 'd'$
 $I_3 \neq '!'$

Goal



(Grammar-based) Whitebox Fuzzing

Aufgabe:

- Einführung in Fuzzing
- Einführung in symbolische Ausführung
- Beschreibung von Whitebox Fuzzing Strategien
- (Optional: Whitebox Fuzzing von Interpretern)



Wie geht es weiter...!?

E-Mail an sebastian.ruland@es.tu-darmstadt.de bis heute 24 Uhr mit

- eurem Erst-, Zweit- und Drittwunsch und
- Studiengang + Fachsemester

Unsere Aufgaben

- Wir verteilen die Themen schnellstmöglich auf die Interessenten
- Geben das Ergebnis bekannt (→ E-Mail)
- Bereiten alles vor, damit anschließend die Bearbeitung unmittelbar starten kann

Eure Aufgaben

- Warten auf Ergebnisse der Zuteilung
- Anschließend meldet ihr euch bitte unmittelbar beim Betreuer
→ Termin für ein erstes, persönliches Treffen



Fragen?

