Technische Universität Kaiserslautern

o1o1seda o1o1oo

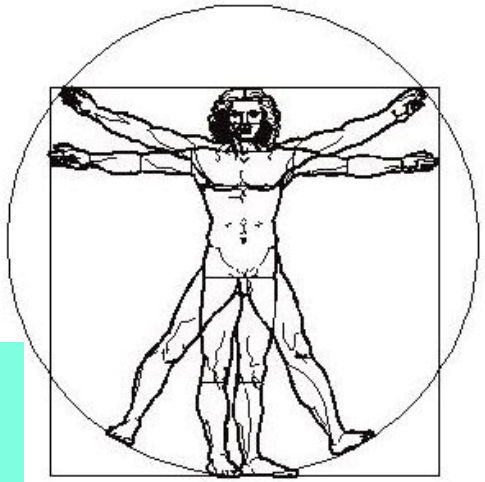
software engineering dependability

Qualitative Analyse der funktionalen Sicherheit software-intensiver Systeme mittels SEFT‘s

Michael Roth, Peter Liggesmeyer
Technische Universität Kaiserslautern

Motivation

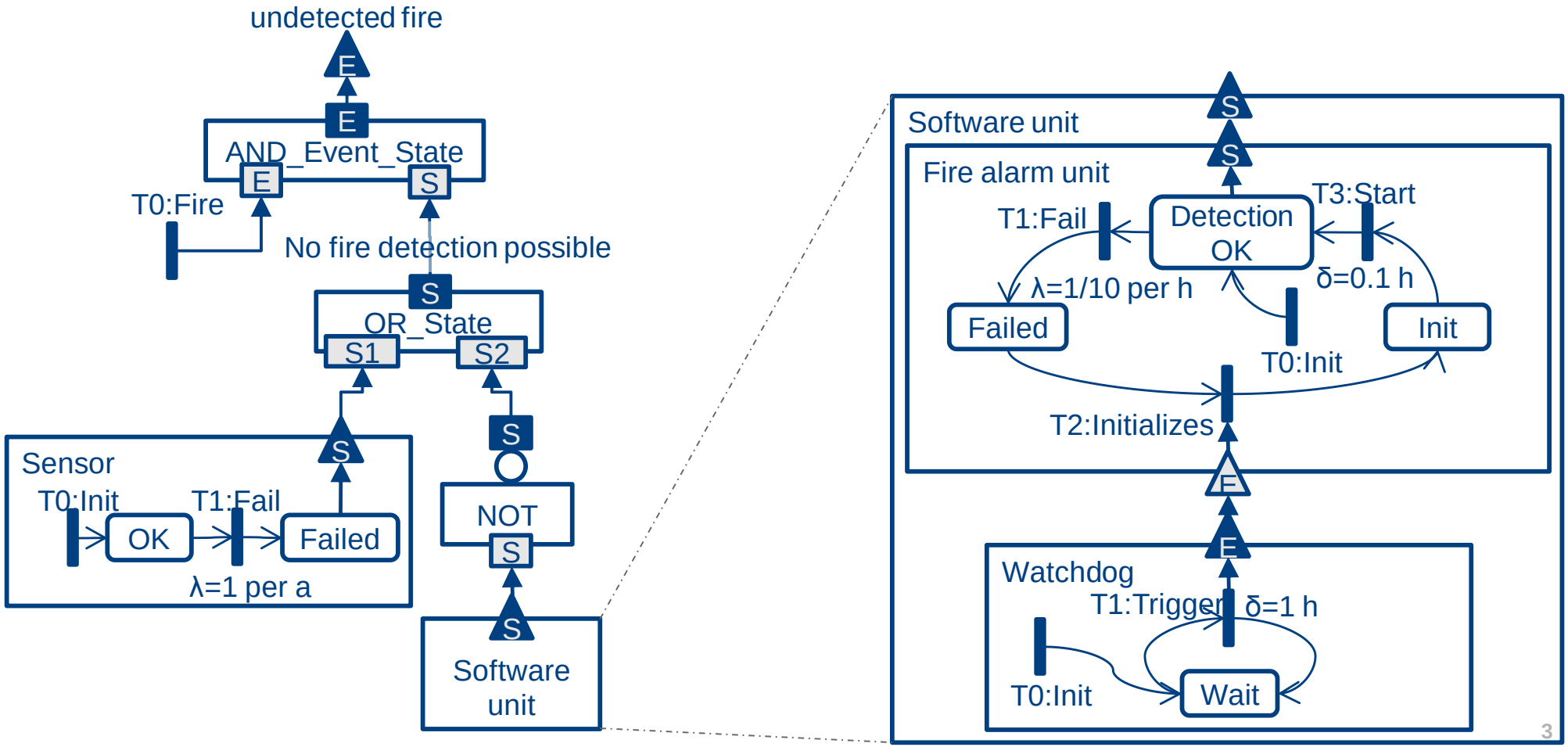

AIRBUS
A350 XWB Prestige




HW
Eingebettetes System
SW
Netzwerk

State/Event Fault Trees*

Kombination von Fehlerbäumen mit Zustandsdiagrammen:



*) B. Kaiser, C. Gramlich, M. Förster: State/event fault trees - A safety analysis model for software-controlled systems. In: Proceedings of the 23rd Int. Conference on Computer Safety, Reliability and Security, 2007

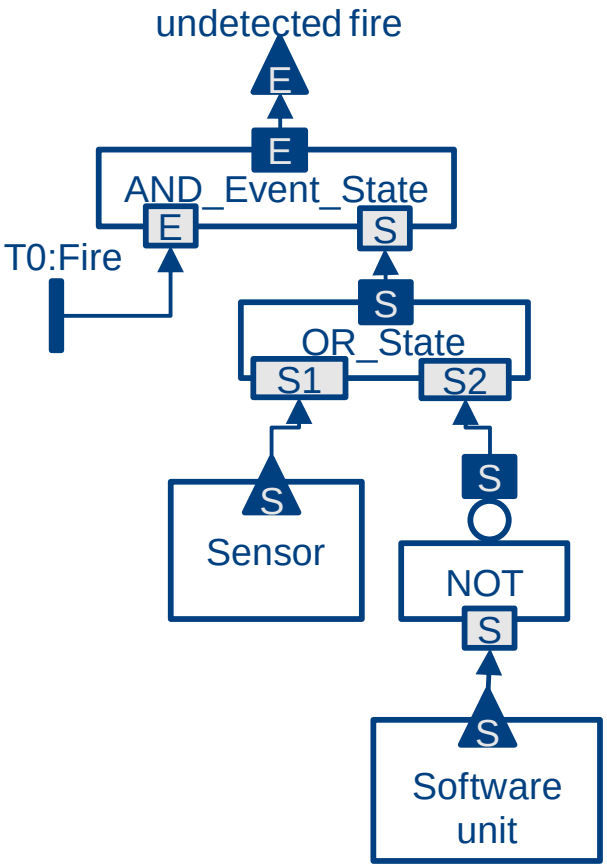
State/Event Fault Tree Eigenschaften

Kombination von Fehlerbäumen mit Zustandsdiagrammen:

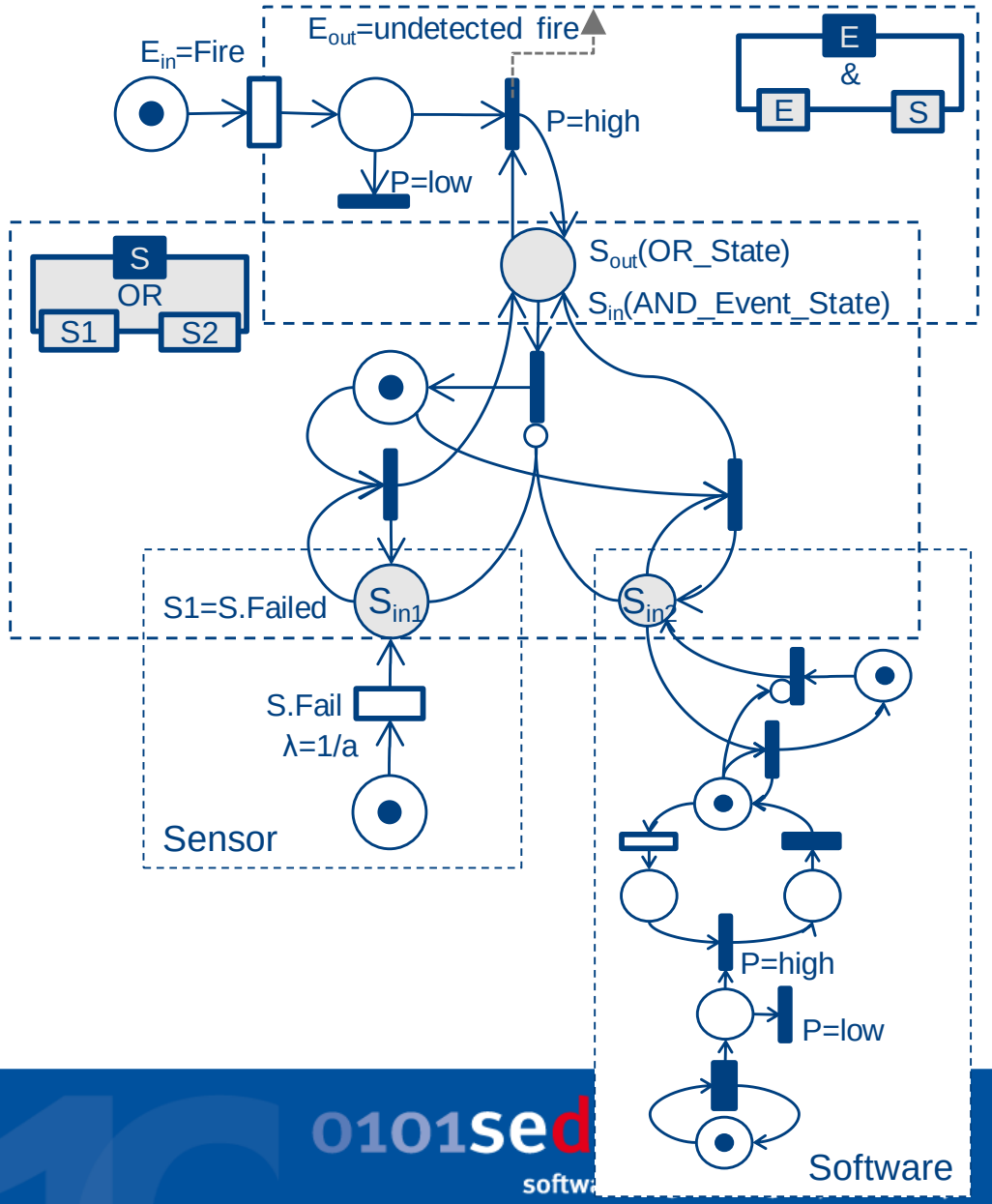
- Modellierung der Dekompositions Struktur des zugehörigen Systems
- Fehlerbaum ähnliche Gatter (vertraut für Sicherheits-Ingenieure)
  - Boolean- und zustandsbasierte Gatter
- Zustandsdiagramme beinhalten das zeitliche Verhalten der Komponenten
- Gut geeignet für software-intensive Systeme*
- **Keine qualitativen Analysemethoden**

*) M. Steiner, P. Keller, P. Liggesmeyer: Modeling the Effects of Software on Safety and Reliability in Complex Embedded Systems. Computer Safety, Reliability and Security (SafeComp Workshops), 2012

State/Event Fault Tree Übersetzung



DSPN



State/Event Fault Tree Analyse

Quantitative Analyse:

TimeNet*:

- Transiente Simulation
- Stationäre Simulation
- Transiente Analyse
- Stationäre Analyse

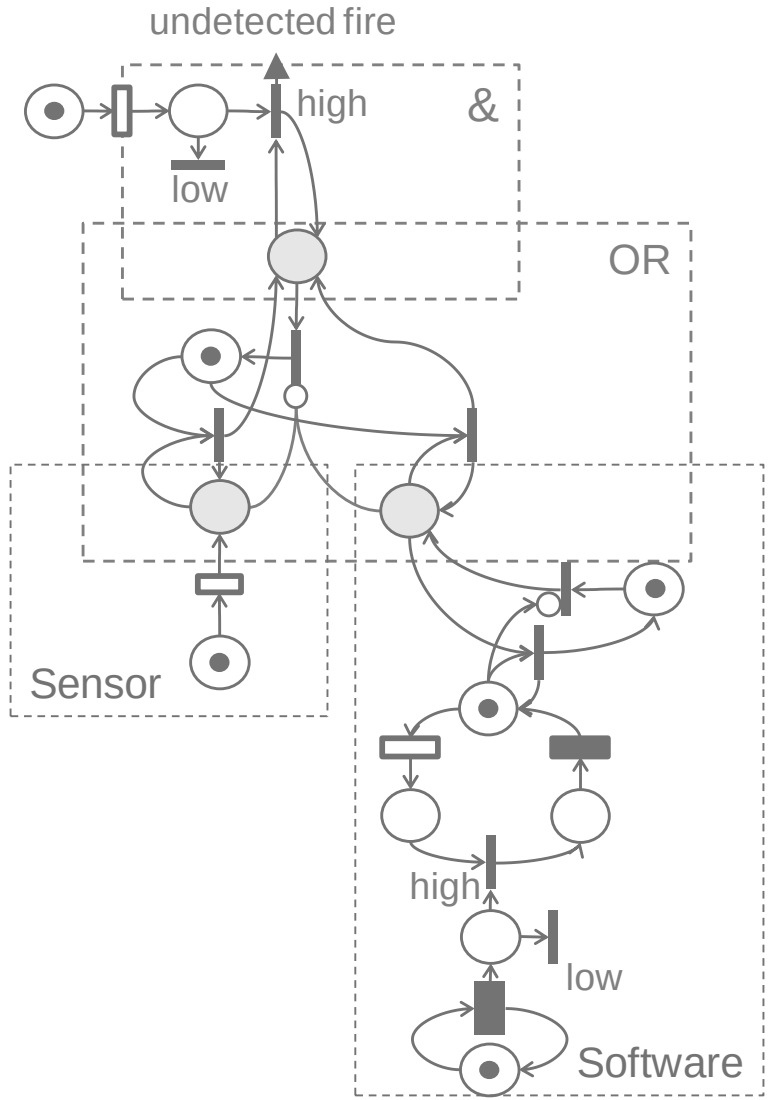
Qualitative Analyse:

Erstellen eines Erreichbarkeitsgraphs (RG) für das übersetzte DSPN:

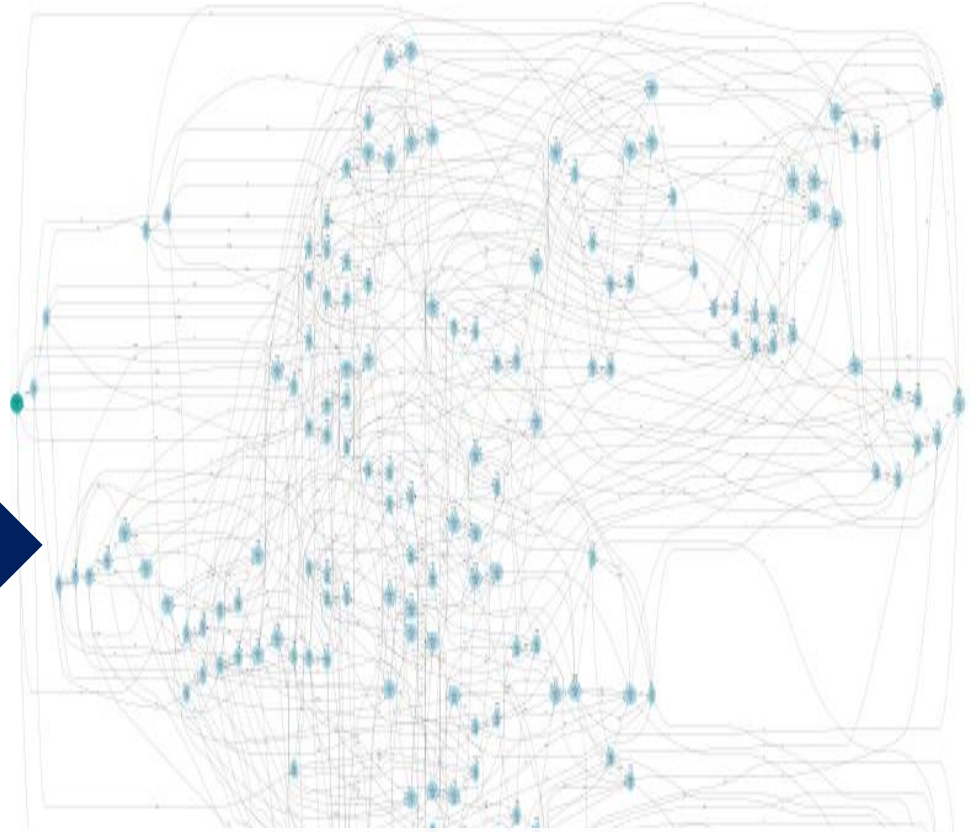
- *Minimale Ereignis Sequences*
- RG is immer endlich, da das zugehörige DSPN immer beschränk ist

*) A. Zimmermann, R. German, J. Freiheit, G. Hommel: TimeNET 3.0 Tool Description. Int. Conf. on Petri Nets and Performance Models (PNPM'99), 1999

Qualitative Analysis: Erreichbarkeitsgraph



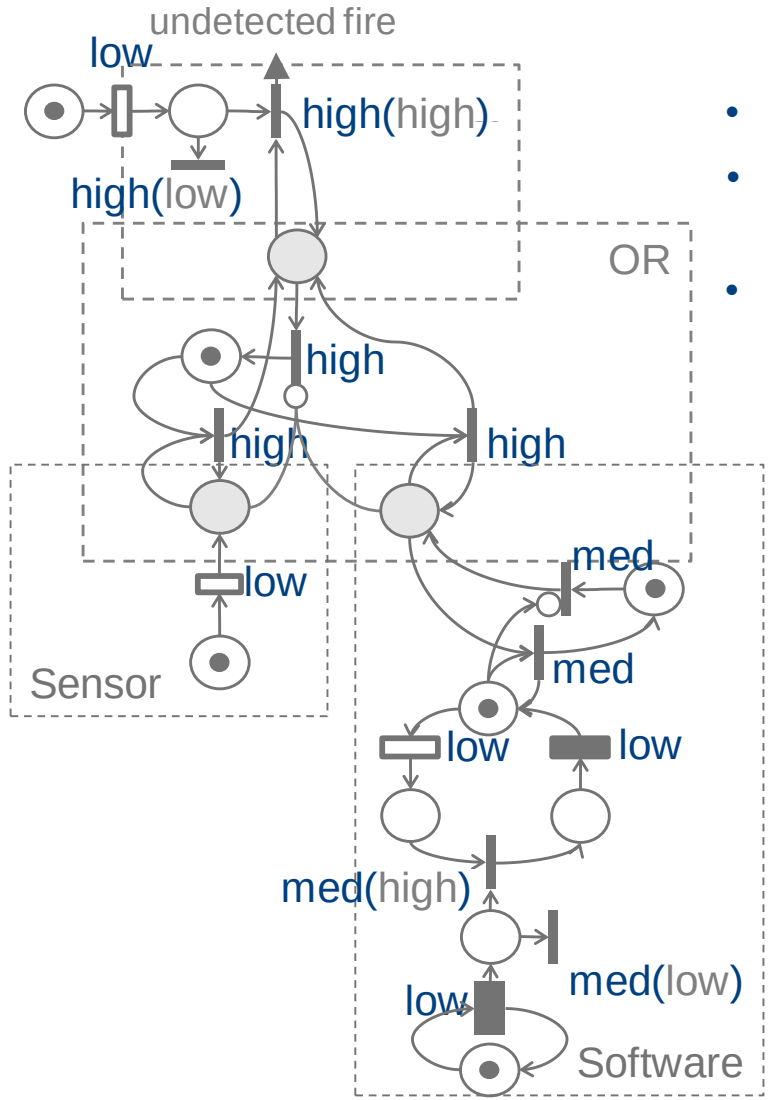
RG(DSPN)



- Wie kann die Zustandsexplosion verhindert werden?
- Wie kann der resultierende Graph einfach und analysierbar gehalten werden?

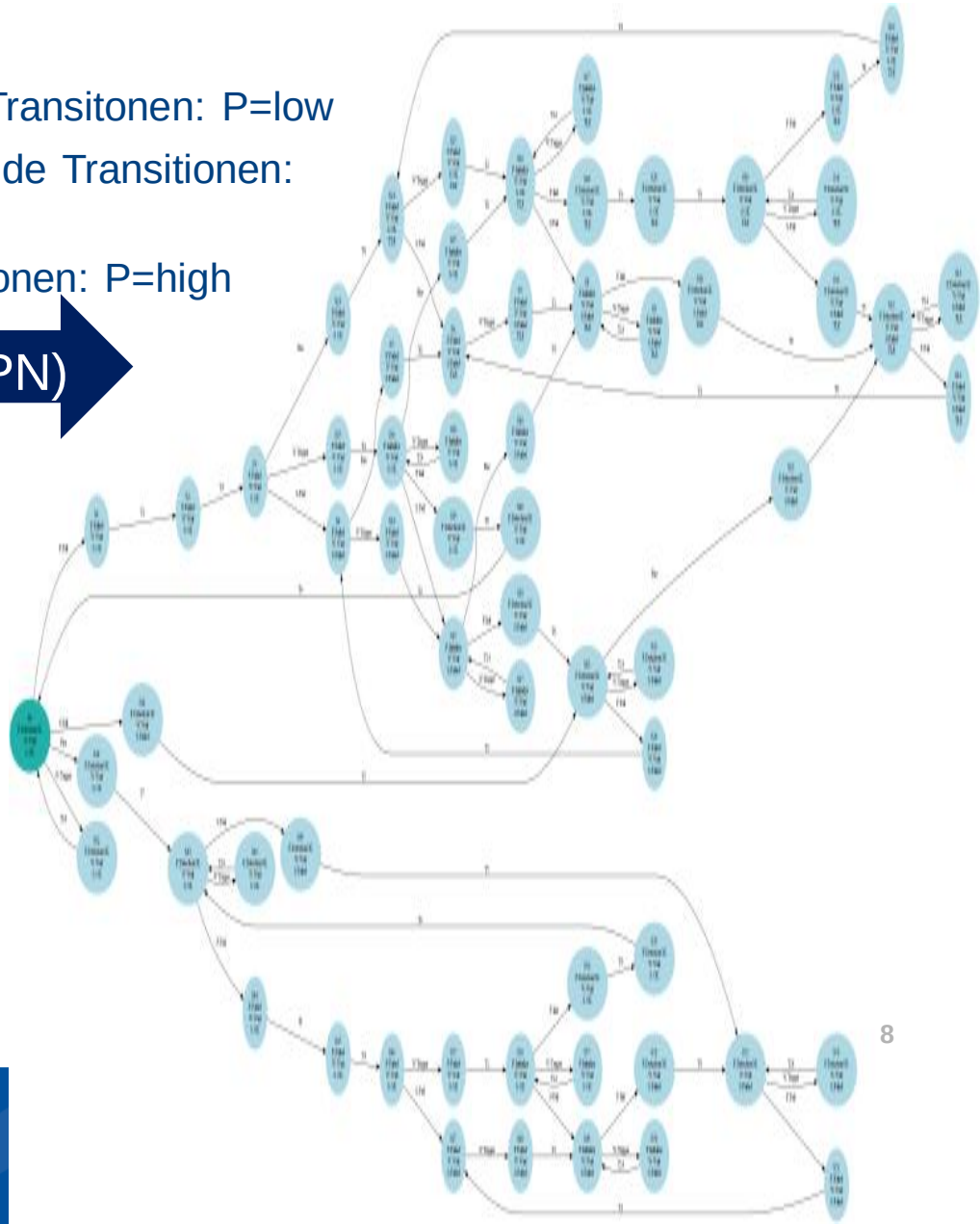
➢ **4 Reduktionsschritte**

1.) Transitions Priorisierung

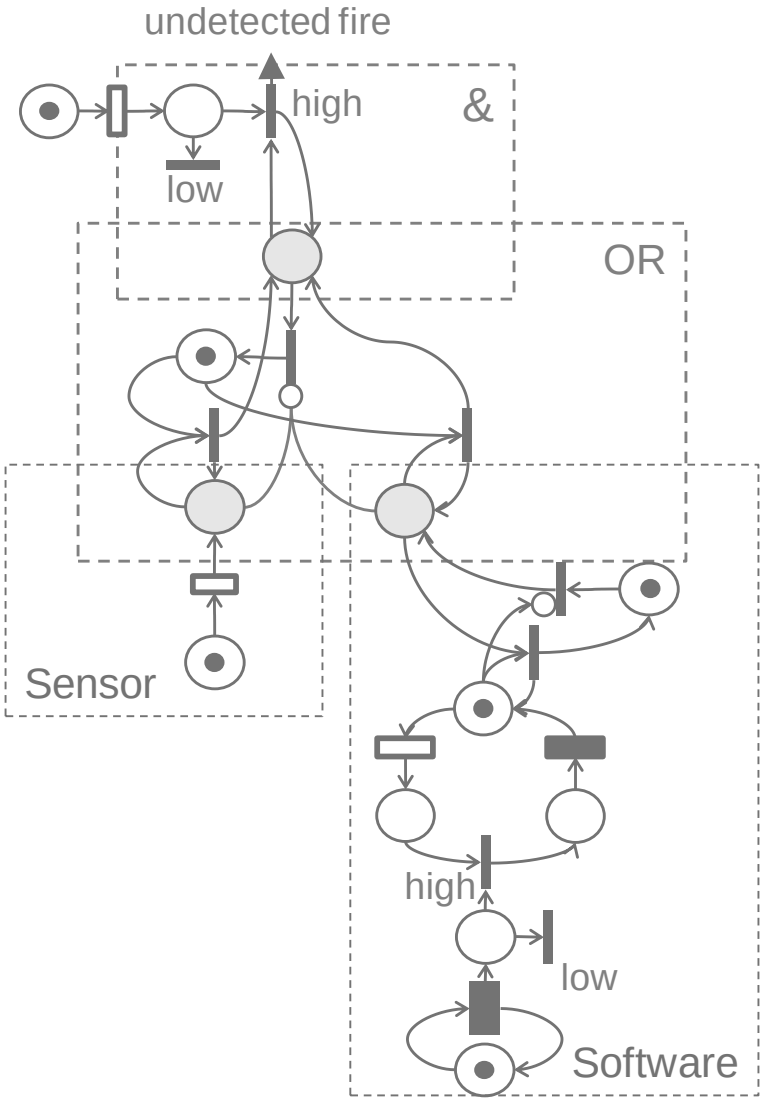


- Zeitbehaftete Transitonen: P=low
- Direkt schaltende Transitionen: P=medium
- Gatter Transitionen: P=high

$RG_{Prio}(DSPN)$

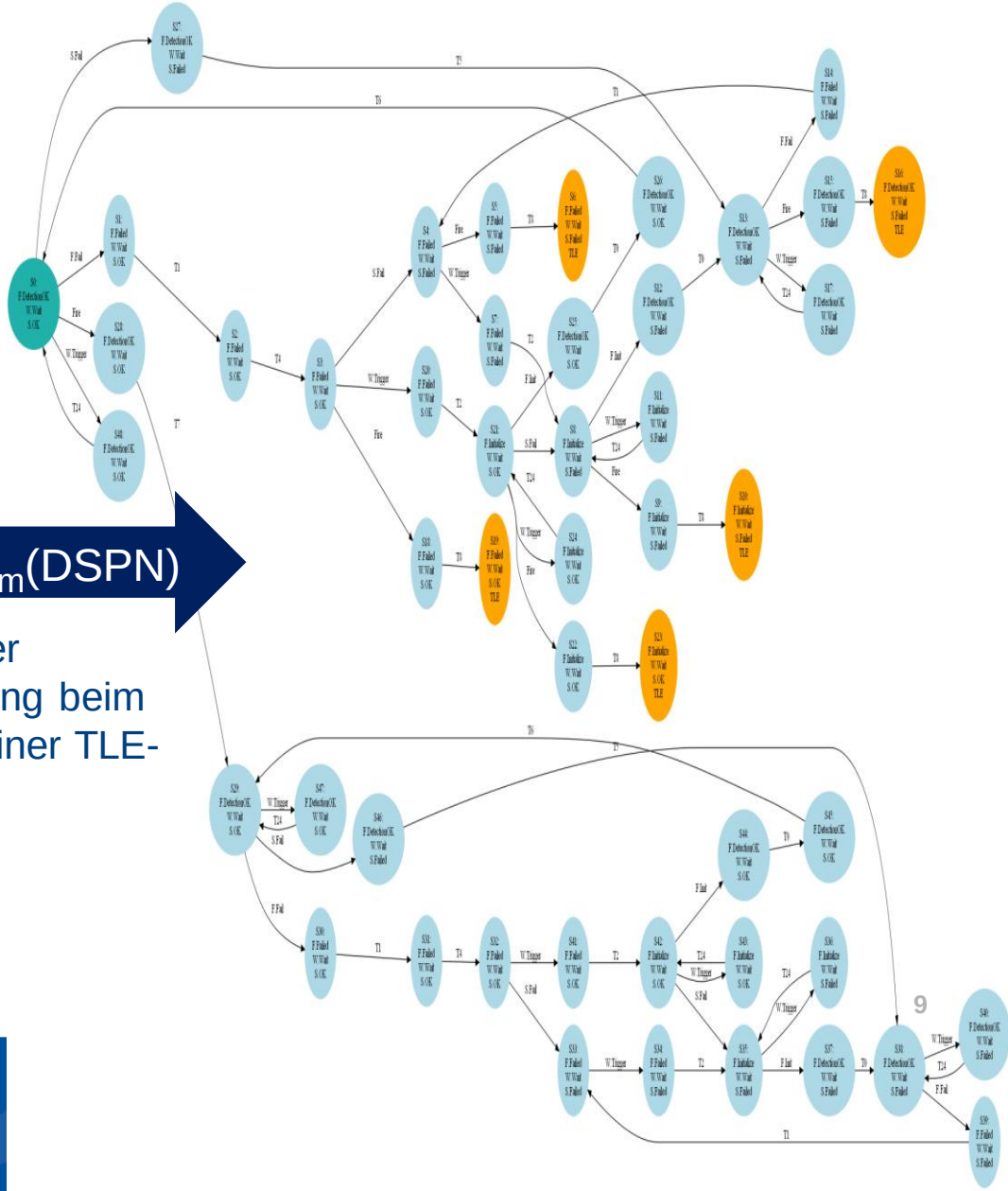


2.) TLE Termination

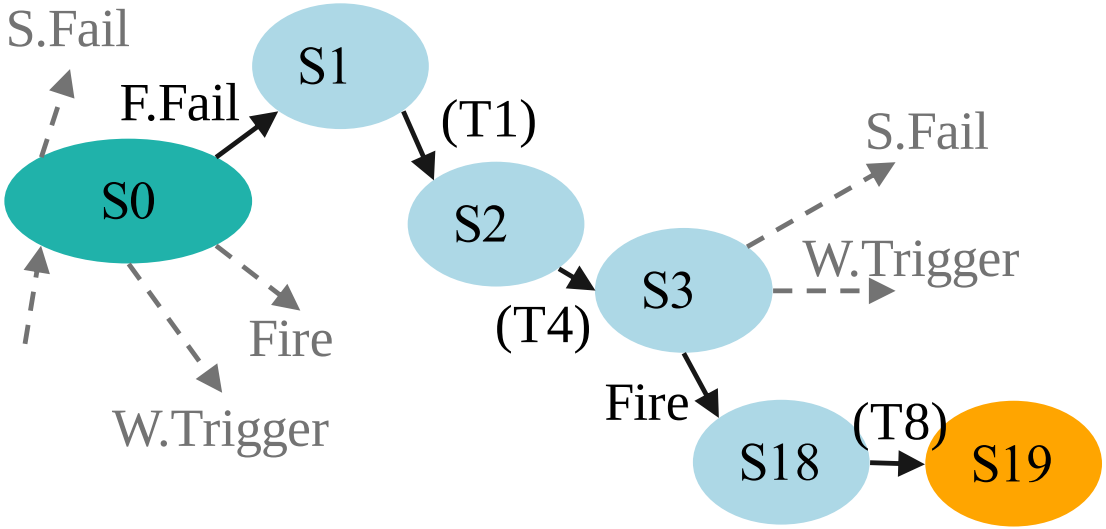


$RG_{Prio\&Term}(DSPN)$

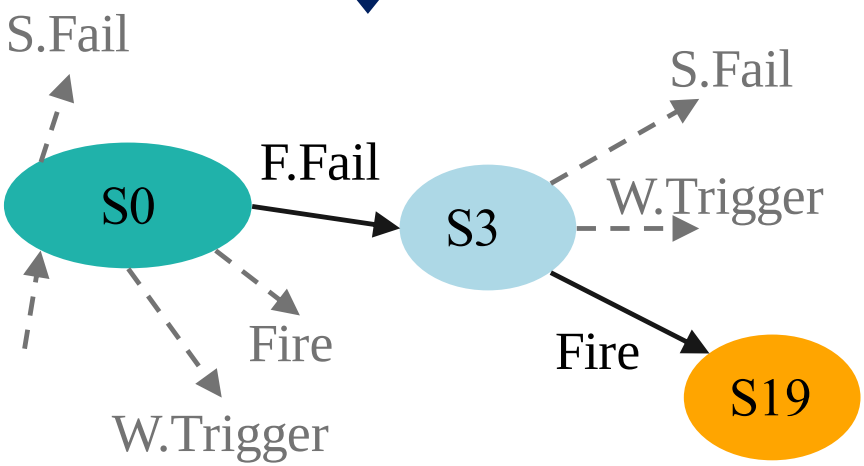
Beenden der Pfaderstellung beim Erreichen einer TLE-Stelle!

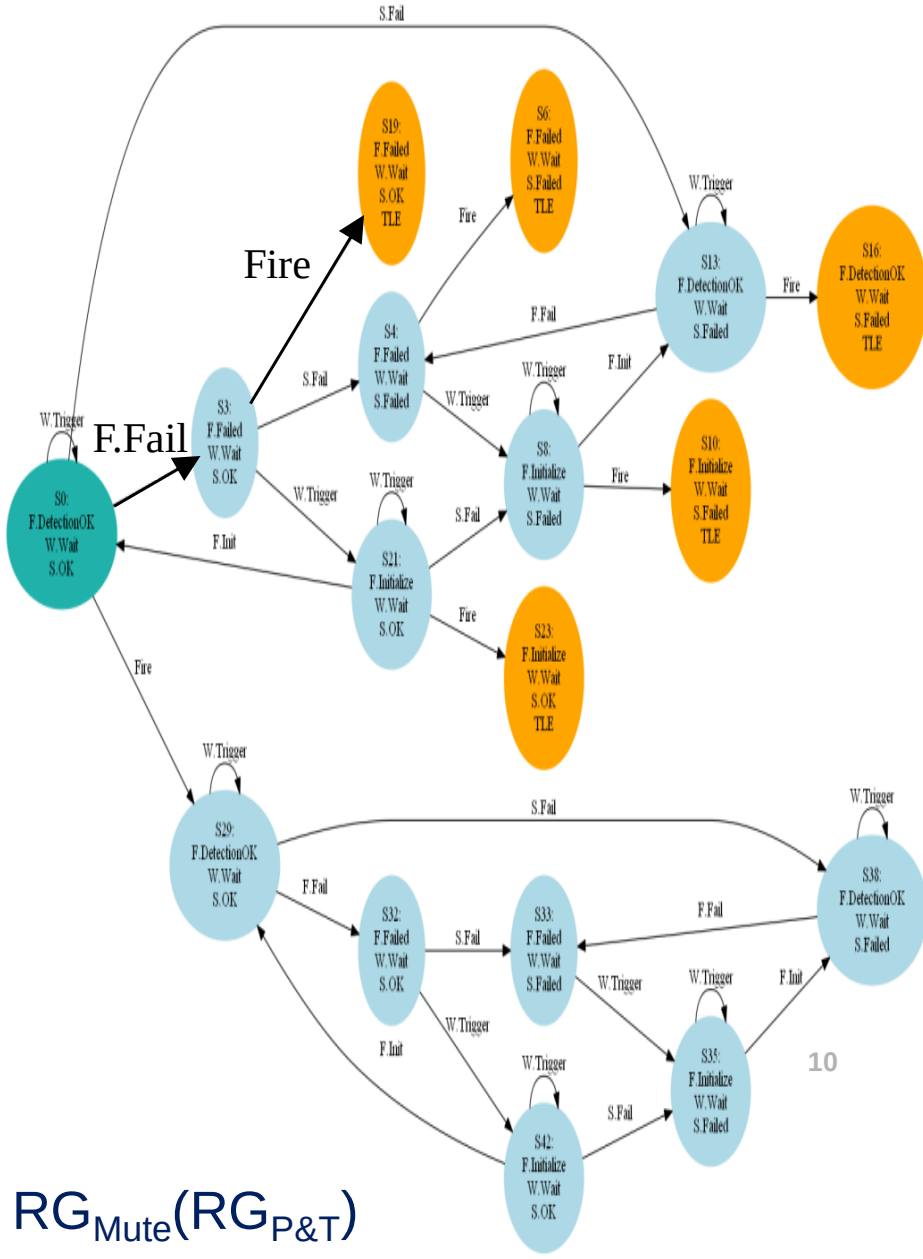


3.) “Stille” Transitionen



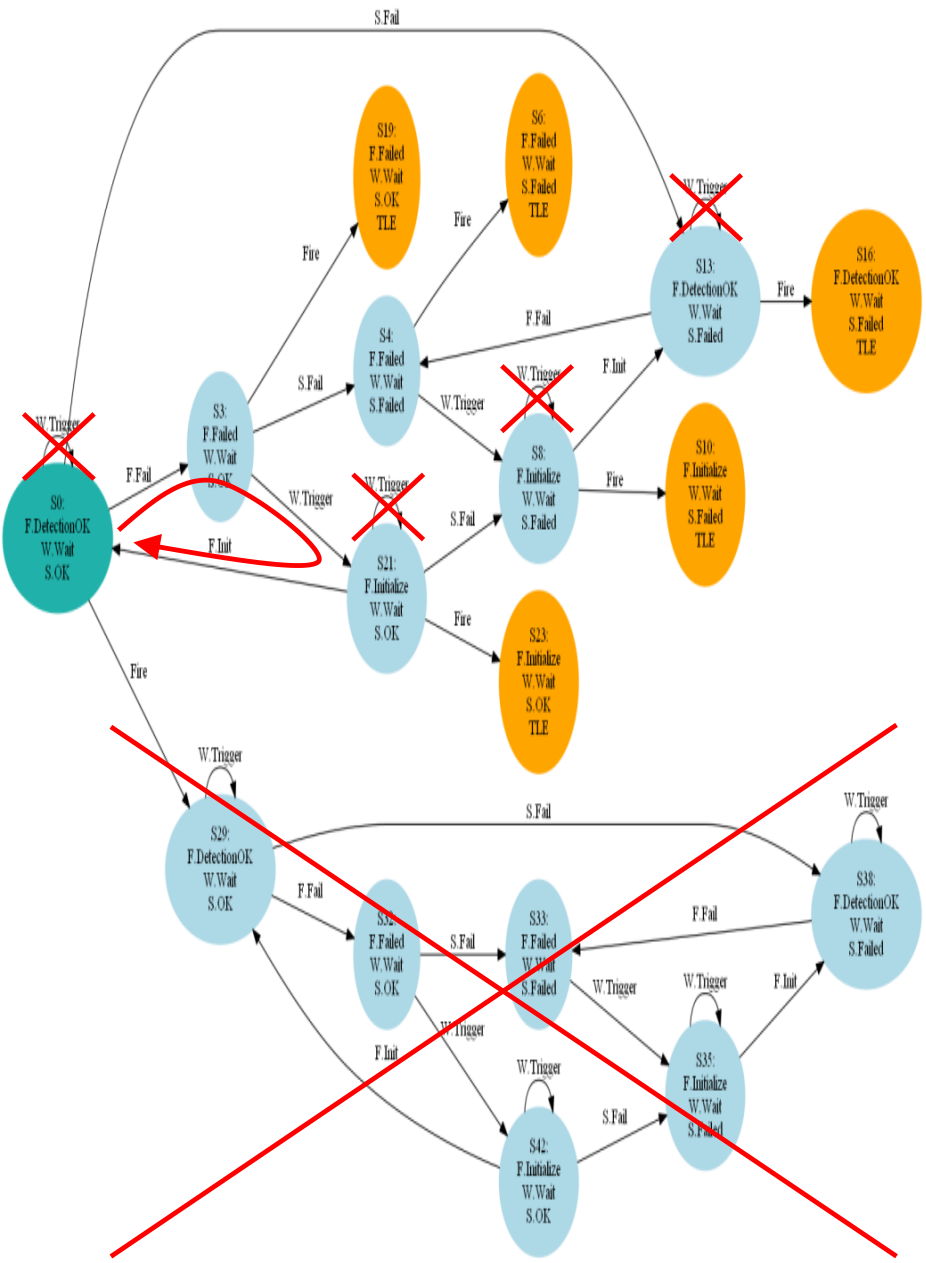
Substitution durch Folgezustand





$RG_{Mute}(RG_{P\&T})$

4.) Minimalisierung

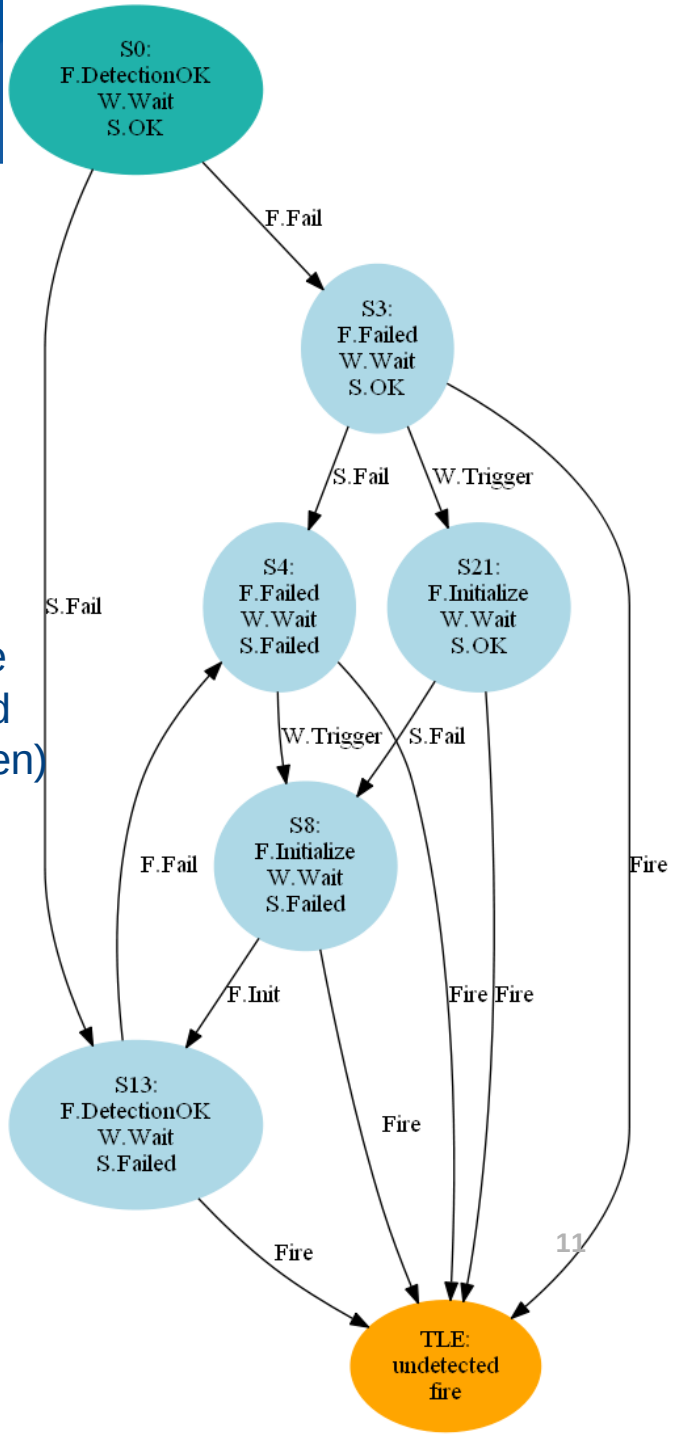


$RG_{Min}(RG_{Mute})$

Entfernen aller Pfade, die nie in einem TLE-Zustand enden (nicht-triviale Zyklen)

Entfernen der trivialen Zyklen

Zusammenführung von TLE-Zuständen



Ergebnisse & Evaluation

	SEFT	DSPN	RG_{Full}	RG_{Prio}	RG_{Term}	RG_{Mute}	RG_{Min}
Zustände	6	13	192	63	49	17	7
Transitionen	9	14	880	90	64	32	13
Gatter	2	-	-	-	-	-	-
Reduktion	-	-	-	85%	90%	95.5%	98%

Zusammenfassung

Zusammenfassung

- 4 stufiger Prozess der eine qualitative Analyse bei SEFTs erlaubt
- Anwendung des Reduktionsprozess auf ein Beispiel System
- Es konnte eine signifikante Reduktion des Erreichbarkeitsgraphen gezeigt werden

➢ Dadurch wird der Erreichbarkeitsgraph verständlich gehalten, was eine qualitative Analyse von SEFTs ermöglicht

Aussicht

- Verbessern der Werkzeug-Performance
- Entwickeln weiterer Analysemethoden die auf den resultierenden Graphen angewendet werden können
  - kürzester Pfad
  - Importance-Analyse
  - …