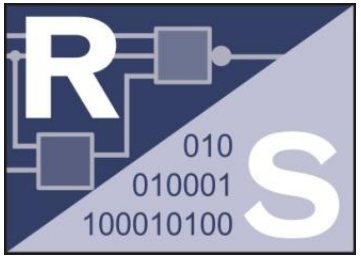
# A Formal Method to Identify Deficiencies of Functional Requirements for Product Lines of Embedded Systems

TECHNISCHE UNIVERSITÄT DARMSTADT


Florian Markert




Computer Systems Group
Technische Universität Darmstadt

Sebastian Oster


Real-Time Systems Group
Technische Universität Darmstadt

# Überblick

TECHNISCHE UNIVERSITÄT DARMSTADT

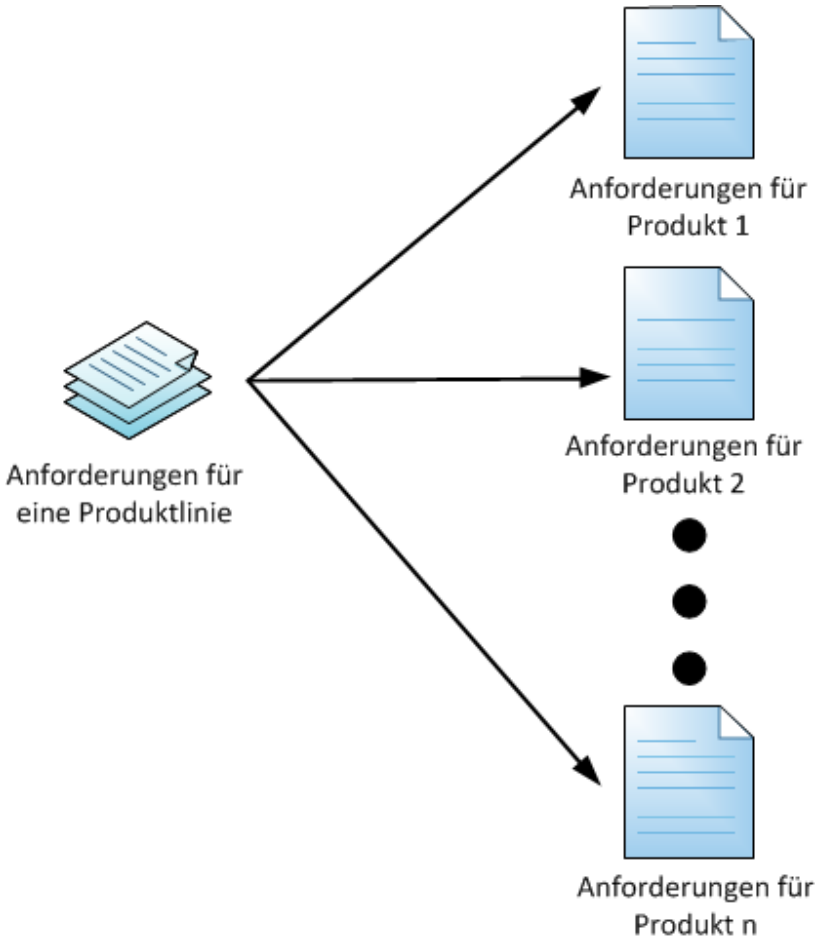
- Motivation
- Einführung
- Lösungsansatz
- Beispiel
- Zusammenfassung & Ausblick

# Motivation – Anforderungen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Die Qualität eines Produkts hängt stark von der Qualität der zu Grunde liegenden Anforderungen ab.
- Anforderungen sollten zu Beginn des Entwicklunbgsprozesses möglichst „gut“ sein.
- Um „gut“ zu sein, müssen Anforderungen nach IEEE 830 u.a. vollständig, widerspruchsfrei und modifiziertbar sein.

# Motivation – Anforderungen




Sind die Anforderungen für jedes Produkt

- vollständig?
- widerspruchsfrei (konsistent)?
- frei von Redundanz (modifizierbar)?

Die große Frage:

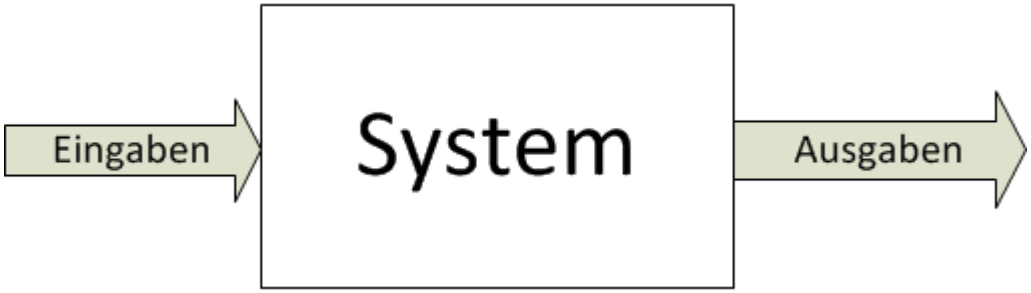
**Kann man nur anhand der Anforderungen für die Produktlinie zeigen, dass für alle ableitbaren Produkte Vollständigkeit, Widerspruchsfreiheit und Freiheit von Redundanz gilt?**

# Einführung – Problembeschreibung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Eingebettete Systeme bestehen aus Hardware und Software, was die Prüfung ihrer Anforderungen erschwert.
- Produktlinien eingebetteter Systeme zeichnen sich durch Variabilität aus.
- Die Beschreibung des Verhaltens eines eingebetteten Systems in den Anforderungen unterscheidet nicht, welcher Teil in Software und welcher in Hardware realisiert wird.

# Einführung – Eingebettetes System



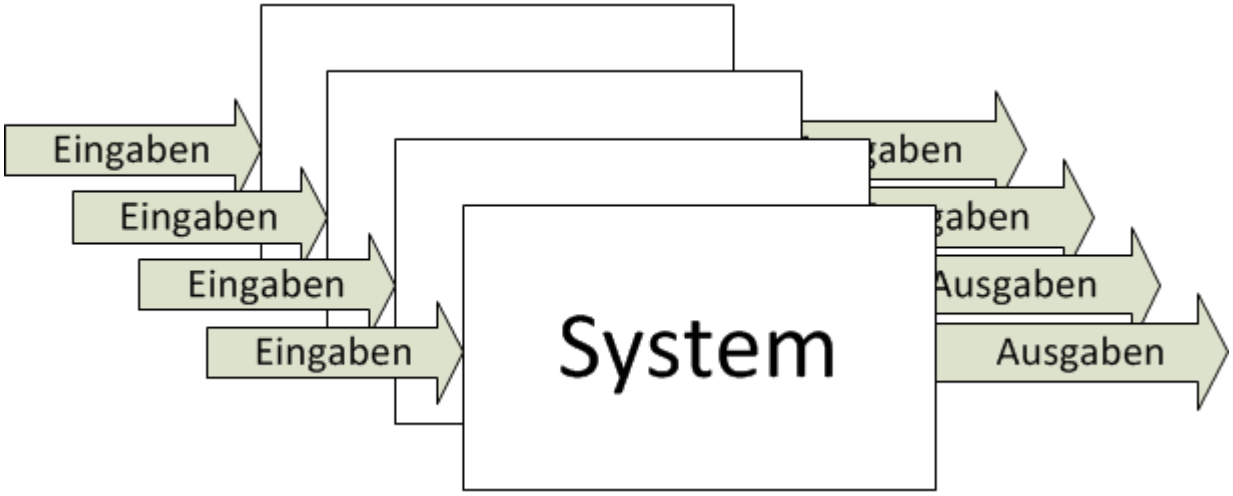

- Sensoren
- Tester
- Nachrichten
- Unterbrechungen

Temporale/logische Abbildung der Eingaben und internen Zustände auf die Ausgaben

- Aktoren
- Motoren
- Nachrichten
- Anzeigen

Beschreibbar durch Linear Temporal Logic (LTL)

# Einführung – Produktlinie



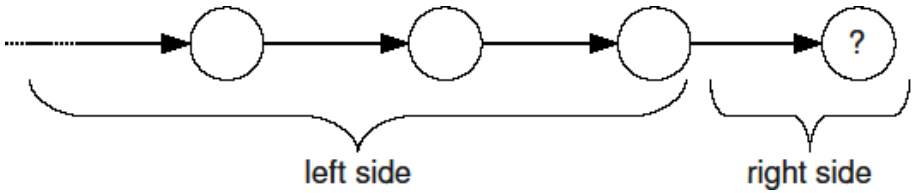

Wie kann eine Produktlinie eingebetteter Systeme

- durch LTL beschrieben werden?
- formal überprüft werden?
- formal bewertet werden?

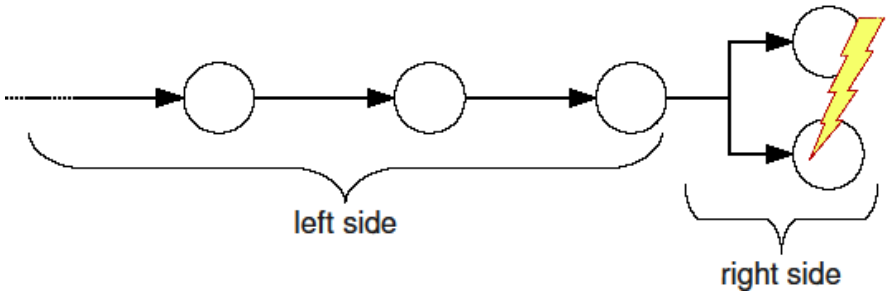
# Einführung – Definitionen

TECHNISCHE UNIVERSITÄT DARMSTADT

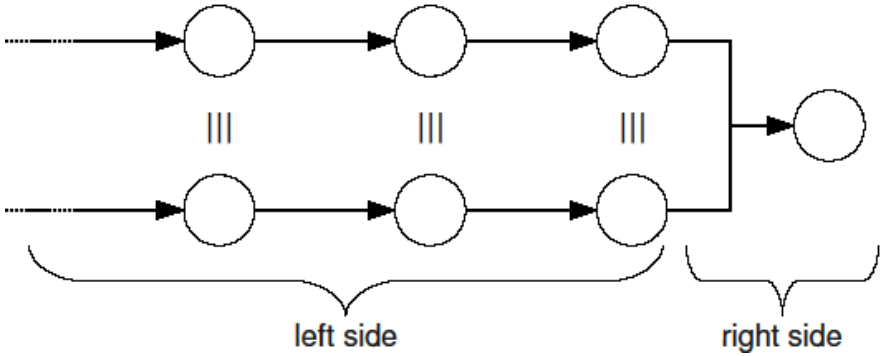
Unvollständigkeit:




Widerspruch:




Redundanz:

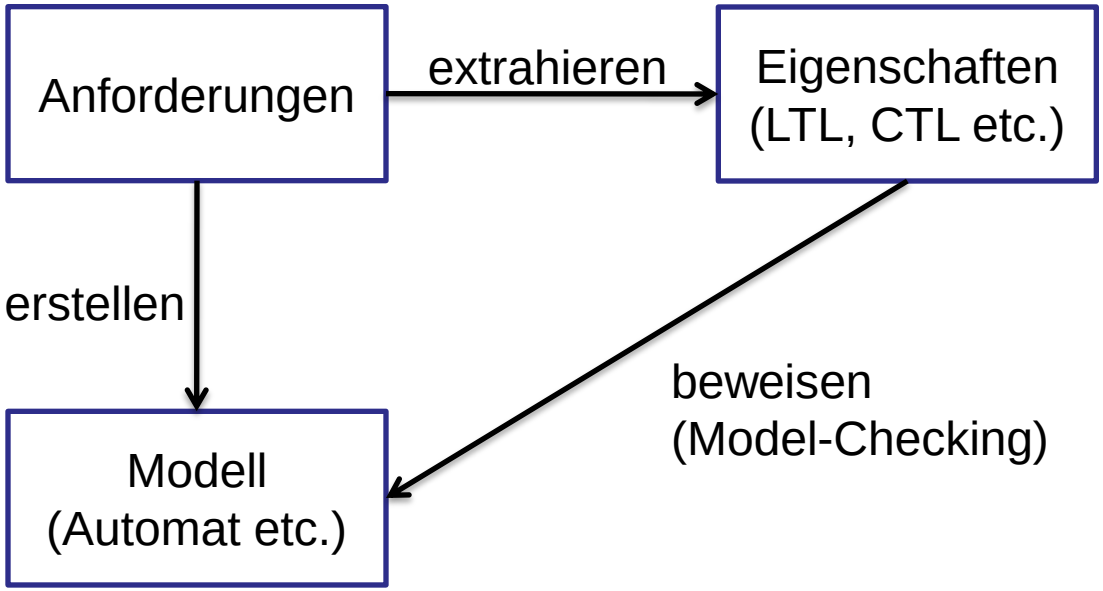


# Lösungsansatz – Formale Methode

TECHNISCHE UNIVERSITÄT DARMSTADT

- Übernahme einer Methode aus der formalen Hardwareverifikation
- Methode wurde ursprünglich zum Prüfen von Eigenschaftssätzen verwendet
- Anpassung an die Bedürfnisse von Produktlinien eingebetteter Systeme

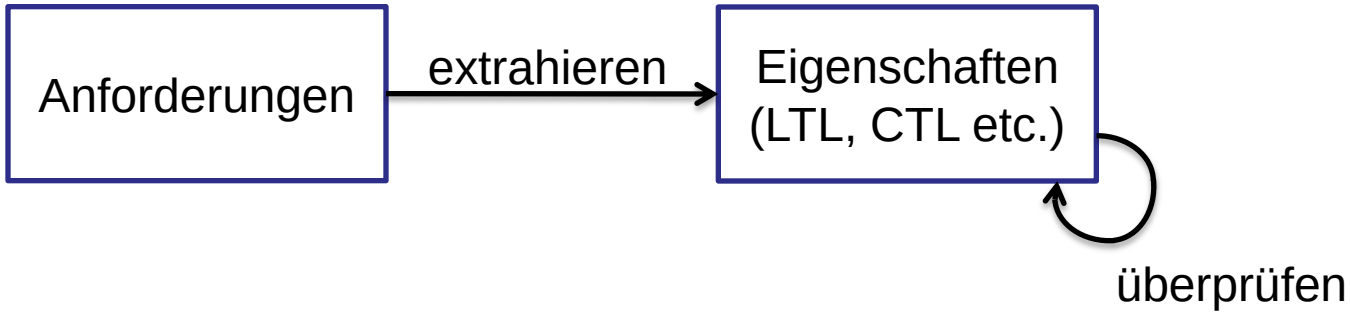
# Lösungsansatz – Ursprüngliches Problem




Problem:

**Ein Eigenschaftsprüfer beweist nur die in den Eigenschaften enthaltene Funktionalität auf dem Modell.**

# Lösungsansatz – Ursprüngliches Problem




Wie kann die Vollständigkeit der Eigenschaften nur mit Bezug auf sich selbst

- definiert werden?
- gezeigt werden?
- gemessen werden?

# Lösungsansatz – Algorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

- Anforderungen werden in eine Menge von Linear Temporal Logic (LTL) Ausdrücken (Eigenschaften) übersetzt.

  z.B. $G(\neg seat\ belt \rightarrow X(\neg seat\_heater))$

- Eigenschaften werden mit Hilfe des Algorithmus aus [1] normalisiert und bewertet.

[1] Martin Oberkönig, Martin Schickel, and Hans Eveking. A Quantitative Completeness Analysis for Property-Sets. In *FMCAD '07, pages 158–161, Washington, DC, USA,* 2007. IEEE Computer Society.

# Lösungsansatz – Algorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

- Alle Ein- und Ausgänge werden als boolesche Werte interpretiert.
- Integer-Werte werden als Bitvektoren dargestellt.
- Der Eigenschaftssatz wird so verändert (normalisiert), dass alle Situationen, in denen ein Augangswert ‘0‘ (off-set $v_0$) und alle, in denen er ‘1‘ wird (on-set $v_1$) erkennbar werden

# Lösungsansatz – Vollständigkeitsmaß

Beispiel:

$c_1 : a \rightarrow c$ (on-set von c)

$c_0 : \neg a \wedge b \rightarrow \neg c$ (off-set von c)

| a | b | on-set c | off-set c |
|---|---|---|---|
| 0 | 0 | | |
| 0 | 1 | | x |
| 1 | 0 | x | |
| 1 | 1 | x | |

Der Ausgang c ist zu 75% vollständig beschrieben. Für den Fall $\neg a \wedge \neg b$ fehlt jede Information über das Verhalten von c.

Ein Widerspruch läge vor, wenn sowohl $c_0$ als auch $c_1$ für die selbe Kombination von $a$ und $b$ gesetzt sein müssten.

# Lösungsansatz – Definitionen

TECHNISCHE UNIVERSITÄT DARMSTADT

Definition der Determiniertheitsfunktion: $v_0 \vee v_1$

Definition der Konsistenzfunktion: $v_0 \wedge v_1$

Definition der Vollständigkeit: $v_0 \vee v_1 \equiv 1$

Definition der Widerspruchsfreiheit: $v_0 \wedge v_1 \equiv 0$

Grad der Vollständigkeit: $\frac{\#Mintherme \equiv 1}{2^{\#Variablen}}$

# Lösungsansatz – Anforderungen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Es gibt verschiedene Möglichkeiten, Anforderungen zu formulieren (Text, Tabelle, Automat…).
- Anforderungen müssen in LTL-Ausdrücke übersetzt werden.
- Zur einfacheren Übersetzbarkeit strukturieren wir diese bisher in Listenform und übersetzen diese in LTL-Ausdrücke

# Lösungsansatz – Produktlinien

TECHNISCHE UNIVERSITÄT DARMSTADT

- Das Feature-Modell wird zur Repräsentation der variablen und gemeinsamen Teile der Produktlinie verwendet.
- Im Feature-Modell enthaltene Informationen müssen zu den Eigenschaften hinzugefügt werden.
- Feature-Modelle müssen in Boolesche Ausdrücke überführt werden.

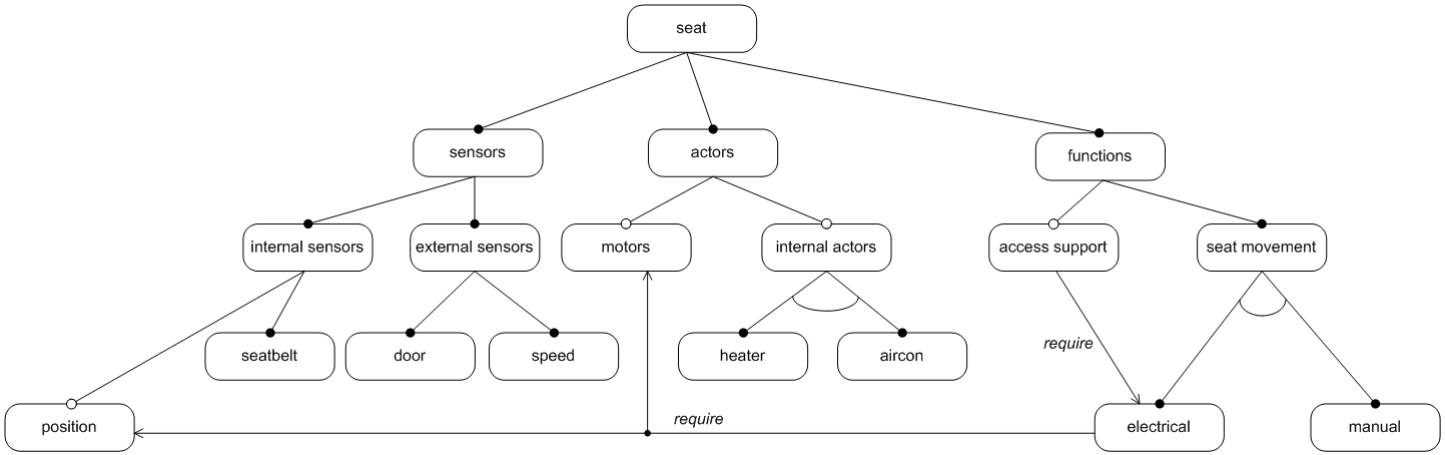
# Lösungsansatz – Produktlinien

TECHNISCHE UNIVERSITÄT DARMSTADT

- Durch das Hinzufügen des Feature-Modells und der Features zu den LTL-Ausdrücken entsteht ein Eigenschaftssatz für alle ableitbaren Produkte.
- Sollte mindestens ein Produkt, das ableitbar ist, unvollständig oder widersprüchlich sein, so gibt der Algorithmus dieses aus.
- Daraufhin können die Eigenschaften analysiert und angepasst werden.

# Beispiel – Autositz

- Gemeinsamkeiten und Unterschiede werden als Feature Model dargestellt



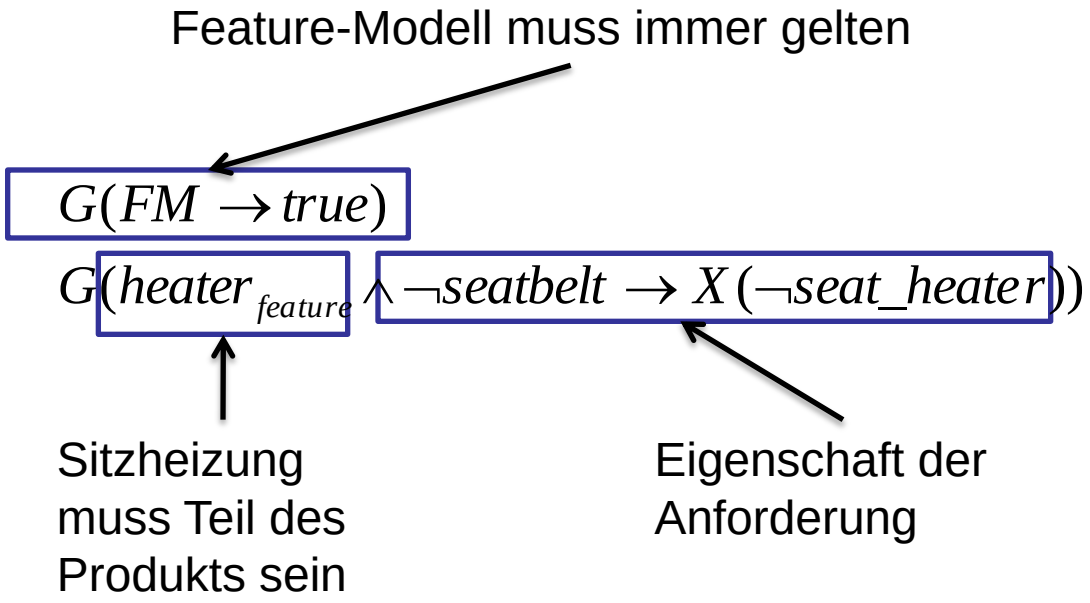

- Feature-Modell wird in einen booleschen Ausdruck übersetzt

$$FM = seat \wedge sensors \wedge actors \wedge \ldots$$

# Beispiel – Autositz

Beispielanforderung:

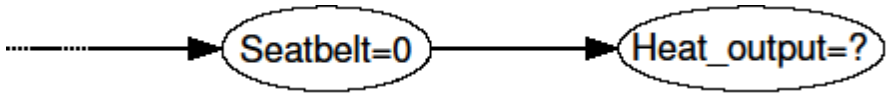
*Die Sitzheizung darf nur aktiv sein, wenn der Sitzgurt angelegt ist*



# Beispiel – Unvollständigkeit

LTL: $G(heater_{feature} \wedge seatbelt \wedge heat\_request \rightarrow X(heat\_output))$

Beispiellücke:




LTL vervollständigen:

$$G((heater_{feature} \wedge seatbelt \wedge heat\_request \rightarrow X(heat\_output)) \vee$$
$$(heater_{feature} \wedge \neg seatbelt \rightarrow X(\neg heat\_output)) \vee$$
$$(heater_{feature} \wedge seatbelt \wedge \neg heat\_request \rightarrow X(\neg heat\_output)))$$

# Beispiel – Widerspruch

TECHNISCHE UNIVERSITÄT DARMSTADT

Verhalten des Motors zur Sitzbewegung:

LTL:

$$M1: e\_movement_{feature} \wedge (engine2\_request = 0) \wedge (engine3\_request = 0) \wedge (speed\_sensor < 16)$$

$$M2: e\_movement_{feature} \wedge (engine1\_request = 1) \wedge (seat\_back > 0)$$

$$M3: e\_movement_{feature} \wedge (engine1\_request = 2) \wedge (seat\_back < 50)$$

$$G((M1 \wedge M2) \rightarrow X(engine1\_output = 1))$$

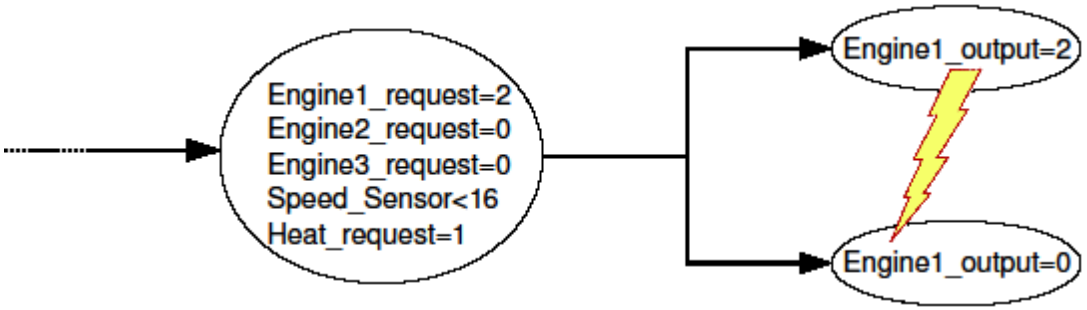
$$G((M1 \wedge M3) \rightarrow X(engine1\_output = 2))$$

$$G((M1 \wedge \neg M2 \wedge \neg M3) \rightarrow X(engine1\_output = 0))$$

$$G(((heater \wedge e\_movement)_{feature} \wedge (heat\_request \neq 0) \wedge (engine1\_request \neq 0)) \rightarrow X((engine1\_output = 0) \wedge (invalid\_inputs = 1)))$$

# Beispiel – Widerspruch

Widerspruch:




Auflösen des Widerspruchs:

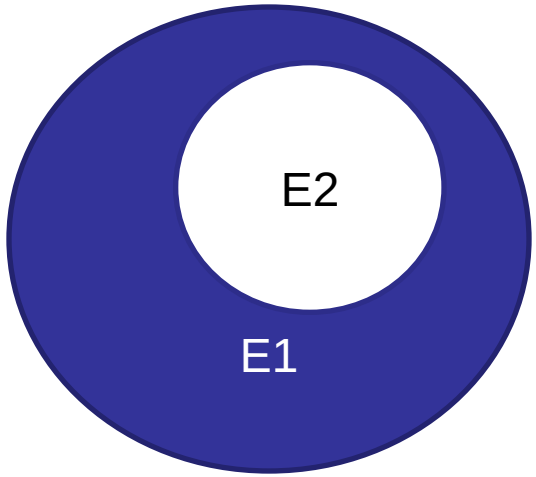
$$M1: e\_movement_{feature} \wedge (engine2\_request = 0) \wedge (heat\_request = 0)$$
$$(engine3\_request = 0) \wedge (speed\_sensor < 16)$$

# Beispiel – Redundanz

TECHNISCHE UNIVERSITÄT DARMSTADT

Wird das Verhalten einer Eigenschaft vollständig und widerspruchsfrei durch eine oder mehrere andere Eigenschaften des Eigenschaftssatzes beschrieben, so ist die Eigenschaft redundant.

Ändert sich der Grad der Vollständigkeit durch weglassen einer Eigenschaft nicht, so ist diese redundant.



# Beispiel – Ergebnisse

TECHNISCHE UNIVERSITÄT DARMSTADT

| Eigenschaften | Zerlegte Eigenschaften | Vollständigkeit | Ausführungszeit |
|---|---|---|---|
| Ursprünglich | 40.282 | 100% | 35s |
| Mit Lücke | 40.274 | 97% mit 2 Lücken | 55s |
| Mit Widerspruch | 40.354 | 100% mit 250 Widersprüchen | 34s |

# Zusammenfassung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Erkennung von Unvollständigkeit, Widersprüchen und Redundanz in Spezifikationen wünschenswert
- Formale Ansätze können für Teilprobleme geeignet sein
- Eindeutigkeit der Spezifikation durch Übersetzung in formale Eigenschaften sichergestellt

# Ausblick

TECHNISCHE UNIVERSITÄT DARMSTADT

- Erforschen der Grenzen des Ansatzes (Komplexität, Darstellbare Probleme)
- Verwendung weiterer Vollständigkeitsansätze
- Erkennung von Fehlern, die durch die manuelle Extraktion der Eigenschaften auftreten können

# Ende

TECHNISCHE UNIVERSITÄT DARMSTADT

Fragen????