# *Wissensbasierte Systeme*

Vorlesung 5 vom 17.11.2004

Sebastian Iwanowski
FH Wedel

# Wissensbasierte Systeme

1. Motivation
2. Prinzipien und Anwendungen
3. Logische Grundlagen
4. ➔ Suchstrategien
5. Modellbasierte Diagnose
    - Kandidatengenerierung
    - Konfliktgenerierung
    - Wertpropagierung
    - Gesamtarchitektur
    - Komponentenmodellierung
6. Andere Diagnosemethoden
7. Weitere Wissensrepräsentationsformen
8. Bewertung wissensbasierter Systeme

# Suchen in Suchgraphen

## Suchgraph:

- **Knoten: beschreibt Zustand in der Suchdomäne**
- **Kante: Übergang von einem Zustand in den nächsten**
  (in der Regel mit Richtung)
- **Startknoten: Anfangszustand**
  (ist immer eindeutig)
- **Zielknoten: gewünschter Endzustand (Lösung des Problems)**
  (es darf mehrere geben)

wünschenswert:

- **Suchgraph ist Suchbaum**
  (Pfad vom Startknoten zu jedem Zielknoten ist eindeutig)

## Verschiedene Suchziele:

1) **Alle Lösungen eines Problems finden**
2) **Die beste Lösung finden**
3) **Ein paar gute Lösungen finden**

# Lösen von CSP mit Suchbäumen

- **Zustand: Belegung von Variablen mit Werten**
- **Folgezustand: Belegung einer weiteren Variable mit einem Wert unter Beibehaltung der Werte für die bisher belegten Variablen**
- **Startknoten: keine Variable hat einen Wert**
- **Zielknoten: gültige Lösung**
- **Expansion eines Knotens: Berechnen aller Folgeknoten**

**Verschiedene Suchstrategien unterscheiden sich in:**

**Welcher Knoten wird als nächstes expandiert ?**

# Allgemeine Suchstrategien für CSP

Im allgemeinen für CSP nur *blinde (uninformierte) Suche* möglich:

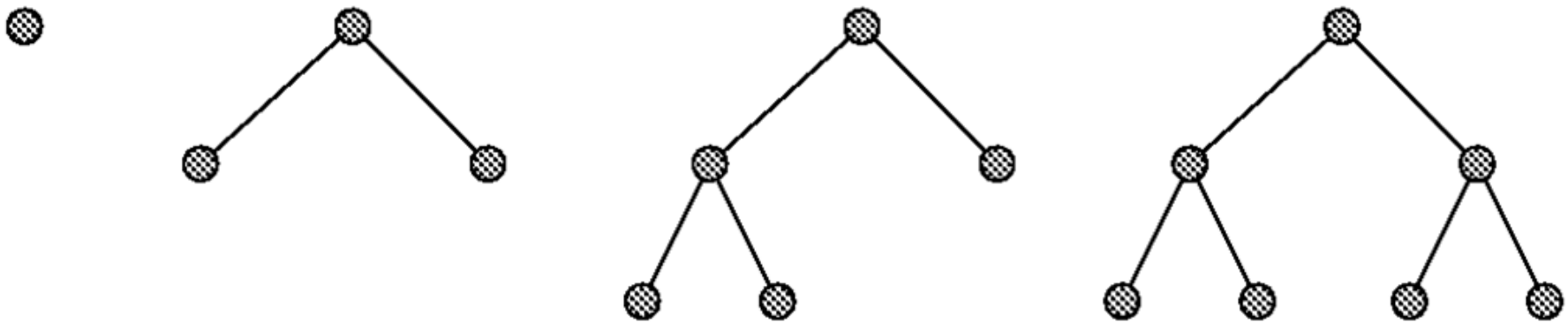
Es gibt keine Information über günstige Suchrichtungen (das Ziel wird erst bei Erreichen erkannt)

Die wichtigsten Suchstrategien:

1. Breitensuche (breadth-first-search)
2. Tiefensuche (depth-first-search)
3. Bestensuche (best-first-search)

# Allgemeine Suchstrategien für CSP

## Breitensuche (breadth-first-search):

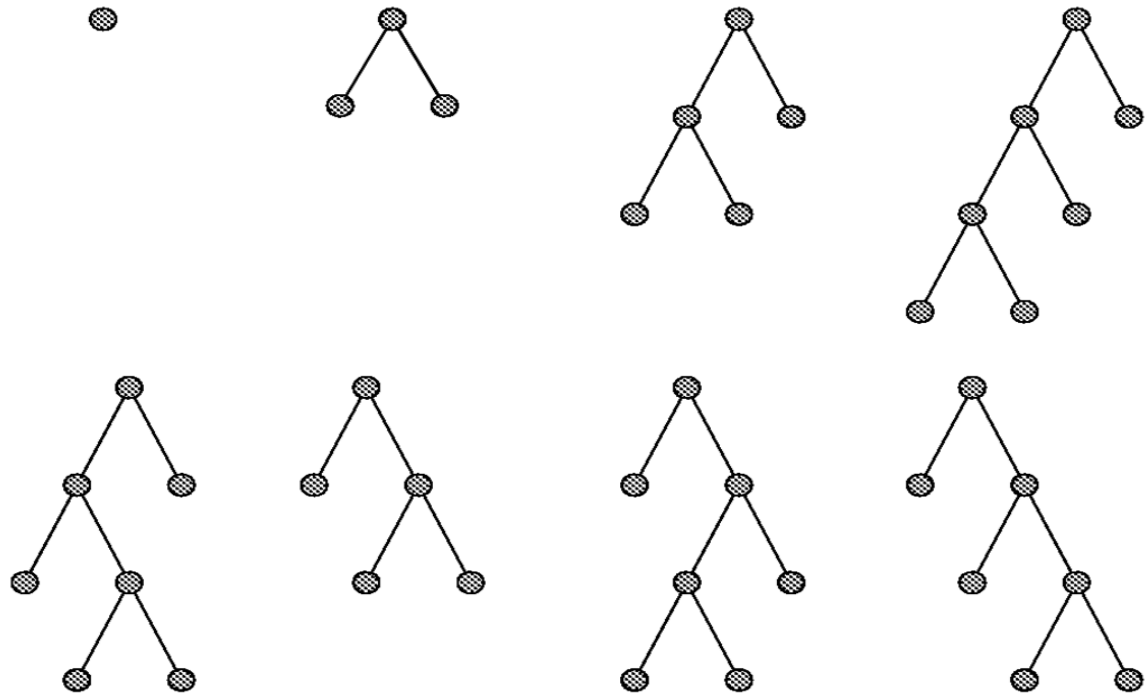


**Exponentieller Aufwand für Zeit und Platz**

**Für CSP uninteressant**

# Allgemeine Suchstrategien für CSP

## Tiefensuche (depth-first-search)



**Exponentieller Aufwand für Zeit**

**Linearer Aufwand für Platz**

**Der „Normalfall“ für allgemeine CSP**

# Allgemeine Suchstrategien für CSP

## Bestensuche (best-first-search)

- zusätzlich sei gegeben: Bewertungsfunktion für die Zustände
- Expandiere jeweils den Zustand mit bester Kostenbewertung

**Im *schlechtesten Fall* ist das nicht besser als Tiefensuche:**

**Exponentieller Aufwand für Zeit**

**Linearer Aufwand für Platz**

**Bei guten Bewertungsfunktionen ist das *Durchschnittsverhalten* viel besser!**

**In Spezialfällen ebenfalls:**

**Bsp.: Spezialfall „Kürzeste-Wege-Problem“:**

**Algorithmus von Dijkstra (nur noch quadratischer Aufwand für Zeit)**

# Spezielle Suchstrategien für CSP

## Zurücksetzen (Backtracking)

- Teste alle Constraints auch bei unvollständigen Variablenbelegungen
- Zustände, die irgendwelche Constraints bereits verletzen, werden nicht weiter expandiert

## Vorwärtstest (Forward Checking)

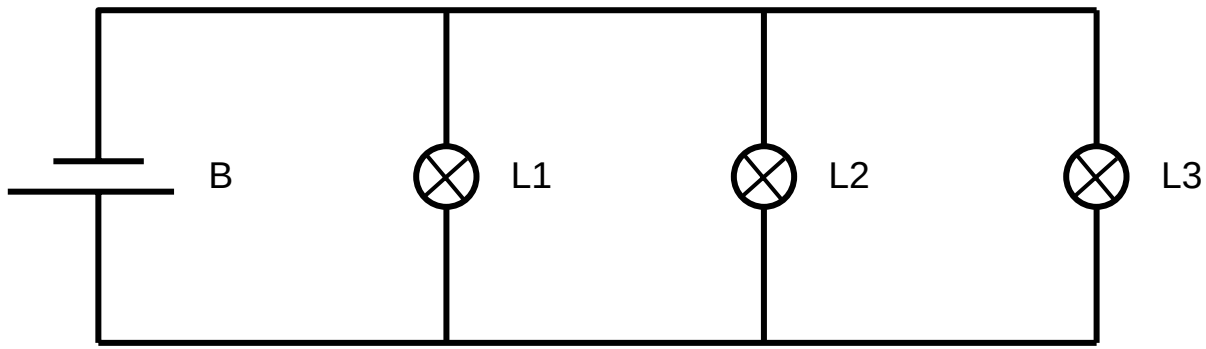
- Reduziere alle Domains für alle noch nicht belegten Variablen, sodass keine Konflikte zwischen Constraints mehr entstehen.
- Setze zurück, wenn die Domains dadurch leer werden.

# Wissensbasierte Systeme

1. Motivation
2. Prinzipien und Anwendungen
3. Logische Grundlagen
4. Suchstrategien
5. Modellbasierte Diagnose
   - Kandidatengenerierung
   - Konfliktgenerierung
   - Wertpropagierung
   - Gesamtarchitektur
   - Komponentenmodellierung
6. Andere Diagnosemethoden
7. Weitere Wissensrepräsentationsformen
8. Bewertung wissensbasierter Systeme

# Modellbasierte Diagnose

**Beispiel, warum Addierer-/Multiplizierer-Beispiel nicht alle Schwierigkeiten des GDE-Ansatzes aufzeigt:**




**Beobachtung:**

**L1, L2 leuchten nicht, L3 leuchtet**

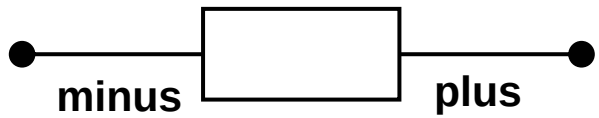
**GDE-Diagnosen:**

1. **(B ok, L1 defekt, L2 defekt, L3 ok)**
2. **(B defekt, L1 ok, L2 ok, L3 defekt)** **???**

# Modellbasierte Diagnose

## Modellierung der elektrischen Komponenten:

**Batterie:**



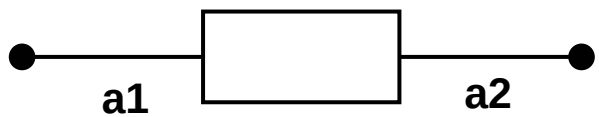

**Wertebereiche:** minus, plus ∈ { `Masse`, `Versorgungsspannung` }

**Regeln:**

ok ⇒ (minus = `Masse`)

ok ⇒ (plus = `Versorgungsspannung`)

**Kabel:**




**Wertebereiche:** a1, a2 ∈ { `Masse`, `Versorgungsspannung` }

**Regeln:**

ok ∧ (a1 = `Masse`) ⇒ (a2 = `Masse`)

ok ∧ (a1 = `Versorgungsspannung`) ⇒ (a2 = `Versorgungsspannung`)

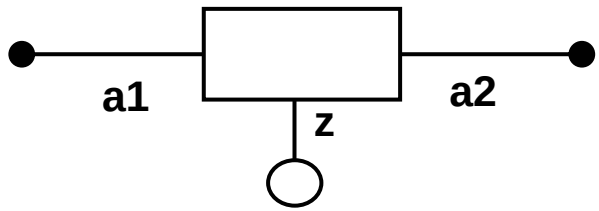
ok ∧ (a2 = `Masse`) ⇒ (a1 = `Masse`)

ok ∧ (a2 = `Versorgungsspannung`) ⇒ (a1 = `Versorgungsspannung`)

# Modellbasierte Diagnose

## Modellierung der elektrischen Komponenten:

**Lampe**:




**Wertebereiche:**

a1, a2 ∈ { Masse, Versorgungsspannung }

z ∈ { hell, dunkel }

**Regeln:**

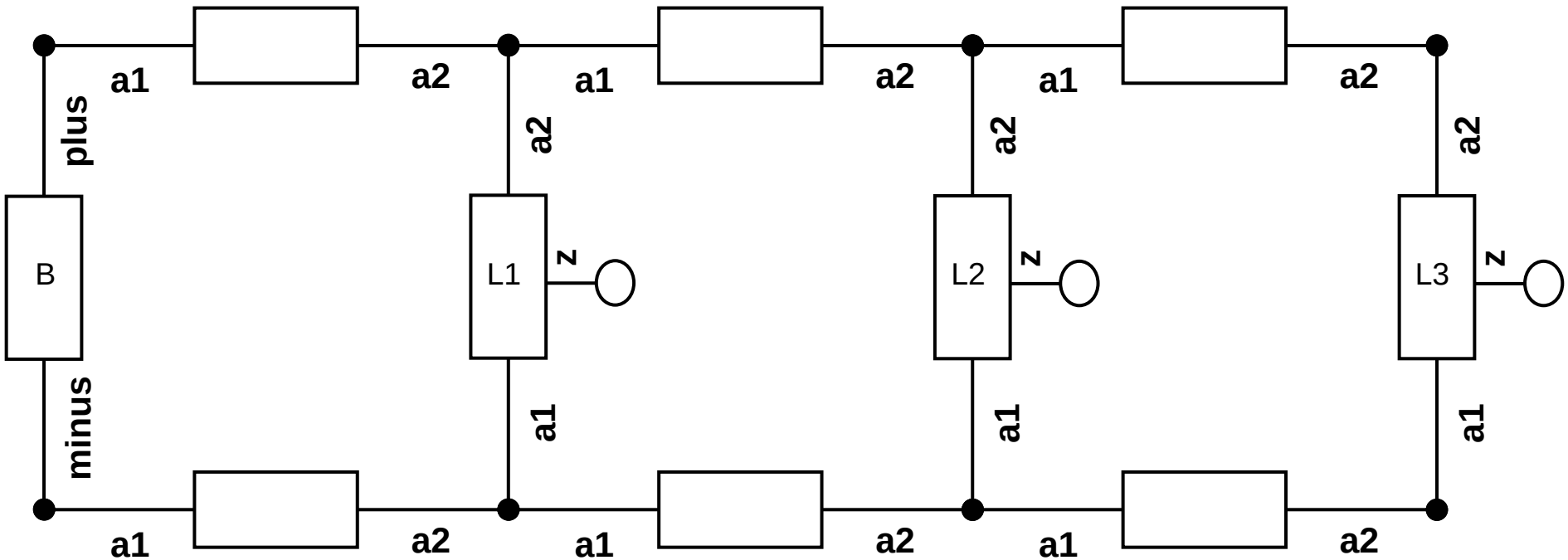
ok ∧ (a1 = Versorgungsspannung) ∧ (a2 = Masse) ⇒ (z = hell)
ok ∧ (a2 = Versorgungsspannung) ∧ (a1 = Masse) ⇒ (z = hell)
ok ∧ (a1 = Versorgungsspannung) ∧ (a2 = Versorgungsspannung) ⇒ (z = dunkel)
ok ∧ (a1 = Masse) ∧ (a2 = Masse) ⇒ (z = dunkel)

ok ∧ (a1 = Masse) ∧ (z = hell) ⇒ (a2 = Versorgungsspannung)
ok ∧ (a1 = Versorgungsspannung) ∧ (z = hell) ⇒ (a2 = Masse)
ok ∧ (a1 = Masse) ∧ (z = dunkel) ⇒ (a2 = Masse)
ok ∧ (a1 = Versorgungsspannung) ∧ (z = dunkel) ⇒ (a2 = Versorgungsspannung)

ok ∧ (a2 = Masse) ∧ (z = hell) ⇒ (a1 = Versorgungsspannung)
ok ∧ (a2 = Versorgungsspannung) ∧ (z = hell) ⇒ (a1 = Masse)
ok ∧ (a2 = Masse) ∧ (z = dunkel) ⇒ (a1 = Masse)
ok ∧ (a2 = Versorgungsspannung) ∧ (z = dunkel) ⇒ (a1 = Versorgungsspannung)

# Modellbasierte Diagnose

## Zusammenbau des Systemmodells aus den Komponentenmodellen:




**Werte an den Verbindungsknoten müssen gleich sein**

**Bei Widerspruch: Konflikt der den Werten zugrunde liegenden Verhaltensmodelle**

**Diagnosen sind Mengen von Verhaltensmodellen, die keinen Konflikt enthalten**

# Modellbasierte Diagnose

## Fazit aus der bisher vorgenommenen Modellierung:

### Es besteht kein logischer Widerspruch zu folgender Diagnose:

**2. (B defekt, L1 ok, L2 ok, L3 defekt)**

### Grund:

**L3 darf im Fehlerfall auch leuchten, wenn keine Spannungsdifferenz besteht**

***Unvollständigkeit der Wissensbasis !***

### Noch schlimmer:

**Wenn eine Verhaltensregeln nur ausgewertet wird, wenn für ihre Voraussetzungen konkrete Werte vorliegen, dann kann auch kein Widerspruch zu folgender Diagnose gefunden werden:**

**3. (B defekt, L1 ok, L2 ok, L3 ok)**

### Grund:

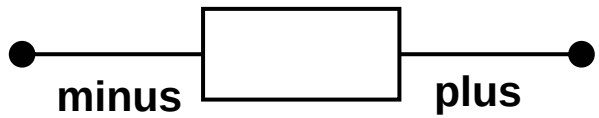
**Es werden nirgendwo Werte berechnet**

***Mangelnde Beweisfähigkeit der Problemlösungskomponente !***

# Modellbasierte Diagnose

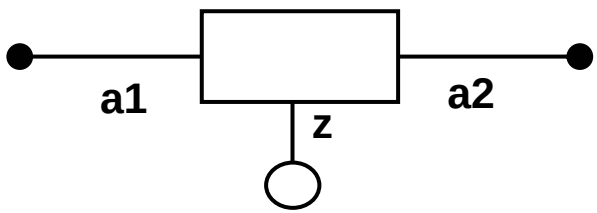
## Zusätzliche Regeln für den Ausschluss von Diagnose 2:

**Batterie:**




defekt ⇒ (minus = `Masse`)

**Lampe:**




defekt ∧ (a1 = `Versorgungsspannung`) ∧ (a2 = `Versorgungsspannung`) ⇒ (z = `dunkel`)

defekt ∧ (a1 = `Masse`) ∧ (a2 = `Masse`) ⇒ (z = `dunkel`)

***Es müssen also auch Verhaltensmodelle für Fehler angegeben werden, um physikalisch unmögliches Verhalten auszuschließen.***

# Modellbasierte Diagnose

## Begriffswelt GDE-Diagnose:

### Komponente:

**Einheit, deren Verhalten diagnostiziert werden soll**

**üblicherweise nummeriert von 1 bis n**

### Komponententyp:

**fasst Komponenten gleichartigen Verhaltens zusammen**

### Verhaltensmodus:

**Einem Komponententyp zugeordnete Verhaltensbeschreibung**

**üblicherweise nummeriert von 1 bis k:**

**1 steht für ok**

**2 bis k sind die Fehlermodi (geordnet nach Wahrscheinlichkeit)**

### (Diagnose-)Kandidat:

**Zuweisung von genau einem Verhaltensmodus an jede Komponente des Systems**

# Modellbasierte Diagnose

## Begriffswelt GDE-Diagnose:

### Kandidat:

**(2 1 3 1 1 2 1) bedeutet:** **Komponente Nr. 1 ist in Verhaltensmodus 2**
**Komponente Nr. 2 ist in Verhaltensmodus 1**
**Komponente Nr. 3 ist in Verhaltensmodus 3**
**Komponente Nr. 4 ist in Verhaltensmodus 1**
**Komponente Nr. 5 ist in Verhaltensmodus 1**
**Komponente Nr. 6 ist in Verhaltensmodus 2**
**Komponente Nr. 7 ist in Verhaltensmodus 1**

### Konflikt:

**Zuweisung von genau einem Verhaltensmodus an einige Komponente des Systems**

**(0 1 0 0 0 2 0) bedeutet:** **Komponente Nr. 2 ist in Verhaltensmodus 1**
**Komponente Nr. 6 ist in Verhaltensmodus 2**
**über die anderen Komponenten wird keine Aussage gemacht**

**Interpretation:** **Es ist unvereinbar, dass sich Komponente 2 in Verhaltensmodus 1 und Komponente Nr. 6 in Verhaltensmodus 2 befindet.**

# Modellbasierte Diagnose

## Begriffswelt GDE-Diagnose:

### Diagnose:

**Kandidat, der keinen Konflikt enthält**

**Beispiele:** **(2 1 3 1 1 2 1) enthält den Konflikt (0 1 0 0 0 2 0), ist also keine Diagnose**

**Wenn (0 1 0 0 0 2 0) der einzige Konflikt ist, ist (1 1 1 1 1 1 1) eine Diagnose**

**Wenn (0 1 0 0 0 2 0) und (1 1 0 0 0 0 0) die Konflikte sind, ist (1 2 1 1 1 1 1) eine Diagnose**

### Präferierter Kandidat:

**Ein Kandidat A ist einem anderen Kandidaten B präferiert, wenn A für jede Komponente maximal den Verhaltensmodus von B zuweist.**

**Beispiel:** **(1 1 1 1 1 1 1) ist präferiert zu (1 2 1 1 1 1 1)**

### Präferierte Diagnose:

**Eine Diagnose ist präferiert, wenn alle ihr präferierten Kandidaten Konflikte enthalten, sie also bezüglich der Präferenz maximal ist.**

**Beispiel:** **Wenn (0 1 0 0 0 2 0) und (1 1 0 0 0 0 0) die Konflikte sind, sind (1 2 1 1 1 1 1) und (2 1 1 1 1 1 1) die beiden einzigen präferierten Diagnosen.**

# Modellbasierte Diagnose

**Ziel von MDS (Daimler-Chrysler-Weiterentwicklung der GDE):**

1) **Finde die wahrscheinlichsten präferierten Diagnosen !**
   **(aus Komplexitätsgründen wird die Stückzahl stark begrenzt)**

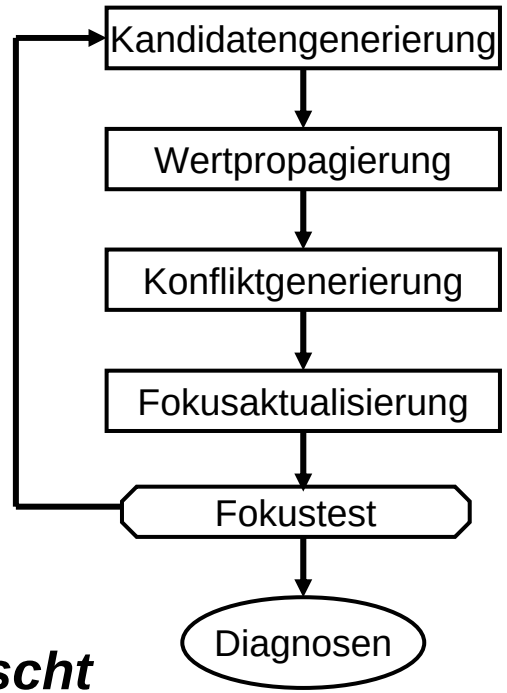
**Weitere Ziele von MDS:**

2) **Schlage Aktionen und Tests vor, um die möglichen Diagnosen weiter einzuschränken !**

# Modellbasierte Diagnose

## Algorithmus für Ziel 1):

1. **Nimm Kandidaten in den Fokus auf.**
2. **Generiere und propagiere alle Werte, die sich aus den Verhaltensmodi der Kandidaten im Fokus ergeben.**
3. **Finde die minimalen Konflikte aus den propagierten Werten.**
4. **Schließe die Kandidaten aus, die Konflikte enthalten.**
5. **Falls Fokus noch genügend groß, dann Ziel erreicht, anderenfalls weiter bei 1.**




***In der Realisierung werden die Schritte 1 bis 4 vermischt***

**(erreicht durch ereignisorientierte Programmierung)**

In den folgenden Vorlesungen werden die Verfahren für **Kandidatengenerierung**, **Konfliktgenerierung** und **Wertpropagierung** getrennt beschrieben.

***Beim nächsten Mal:***

***Kandidatengenerierung***