

# ***Wissensbasierte Systeme***

Vorlesung 6 vom 24.11.2004  
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

# 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 (= konsistenter Kandidat):

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

### Präferenz zwischen Kandidaten:

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)

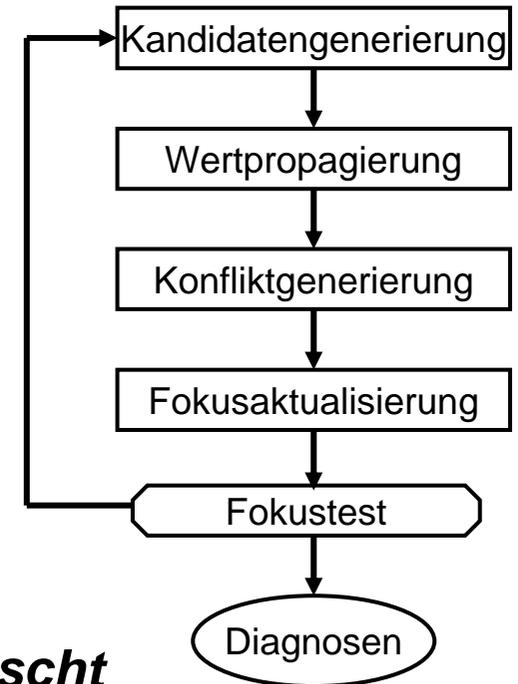
**Weitere Ziele von MDS:**

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

# Kandidatengenerierung

## INPUT:

- **Alte Konflikte und die für diese Konflikte präferierten und konsistenten Kandidaten**
- **Neue Konflikte**

## OUTPUT:

- **Menge der präferierten Kandidaten, die auch für die neuen Konflikte konsistent sind**

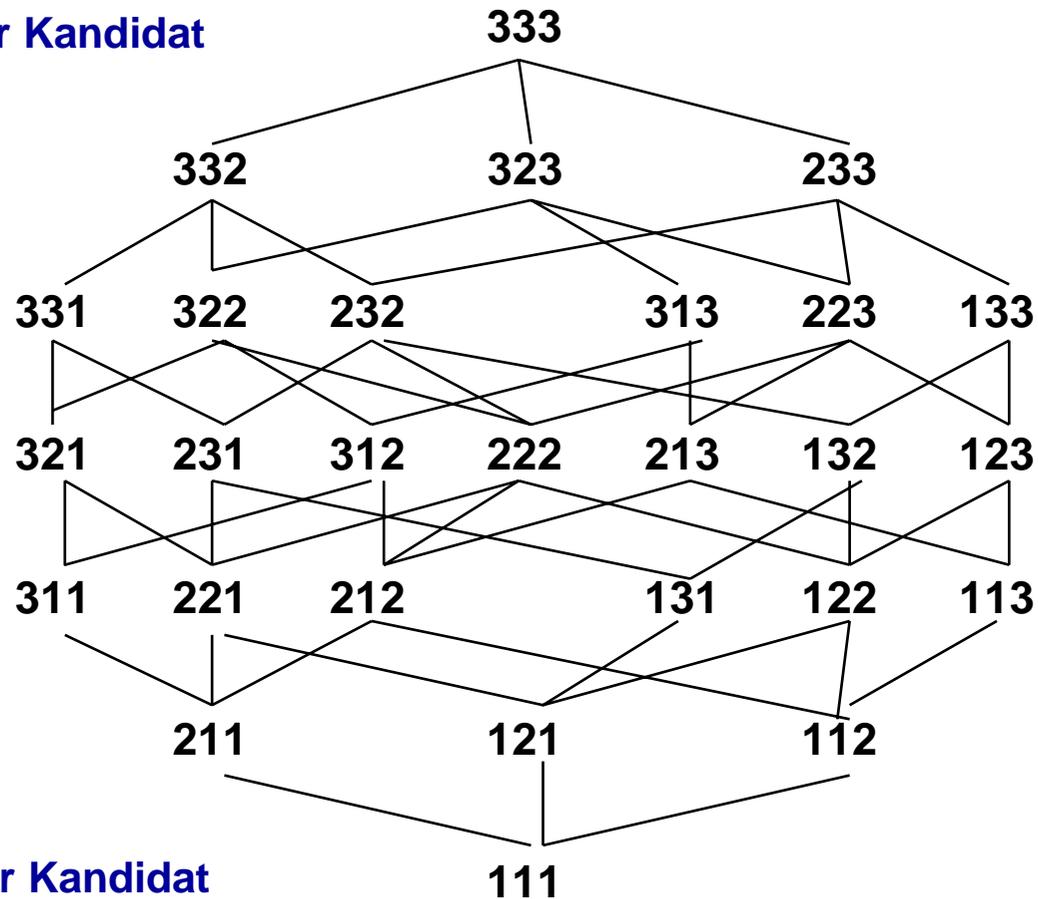
## Einbettung der Kandidatengenerierung in den Diagnoseprozess:

- Output der Kandidatengenerierung wird in den Fokus genommen
- Durch Wertepropagierung werden neue Konflikte gefunden
- Diese Konflikte werden als Input für eine neue Runde der Kandidatengenerierung genommen
- Wenn keine neuen Konflikte gefunden werden, ist der Diagnoseprozess beendet.

# Die Kandidaten im Präferenznetz

**Beispiel:** 3 Komponenten  
Für jede Komponente 3 Verhaltensmodi

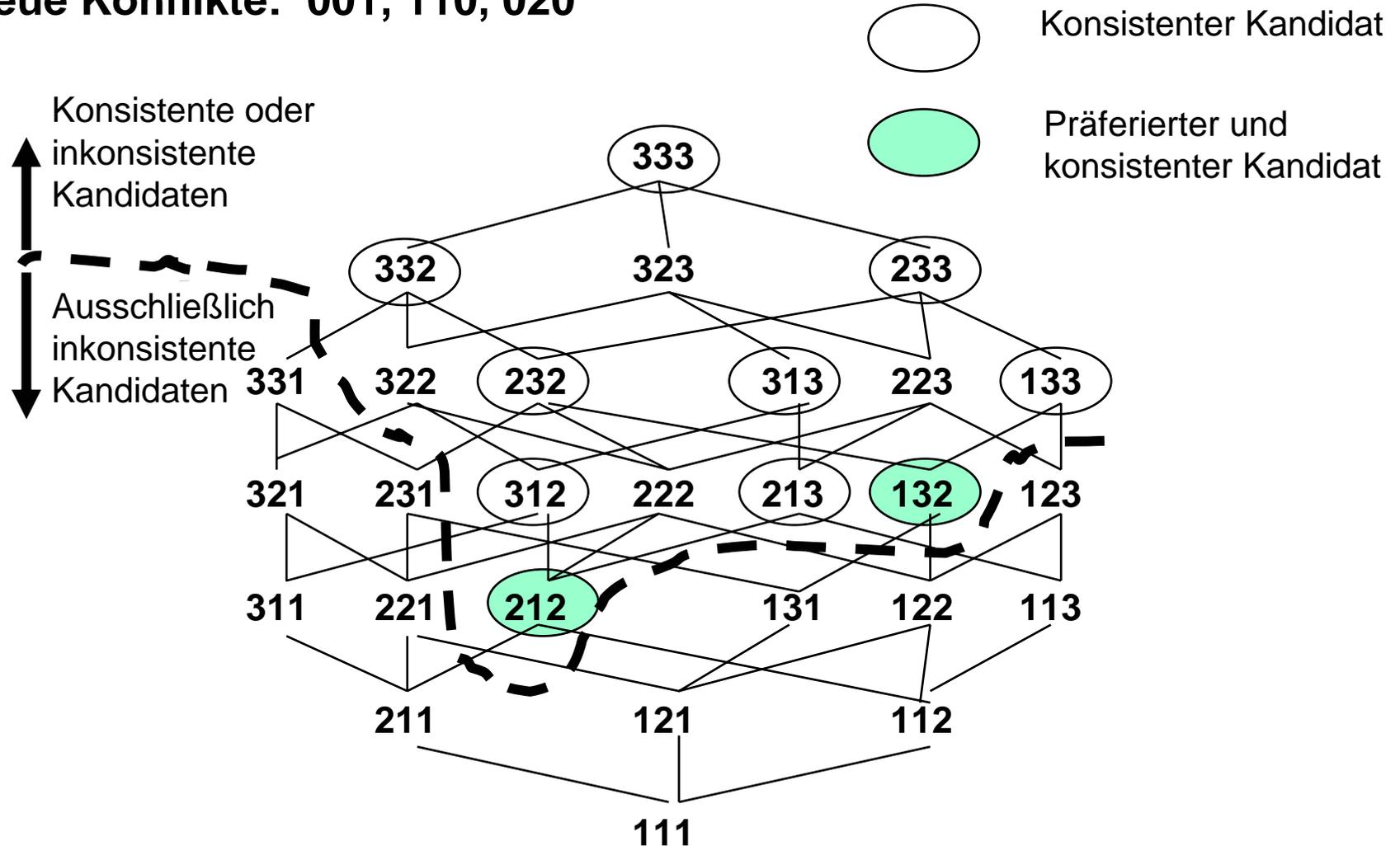
minimal präferierter Kandidat



maximal präferierter Kandidat

# Die Kandidaten im Präferenznetz

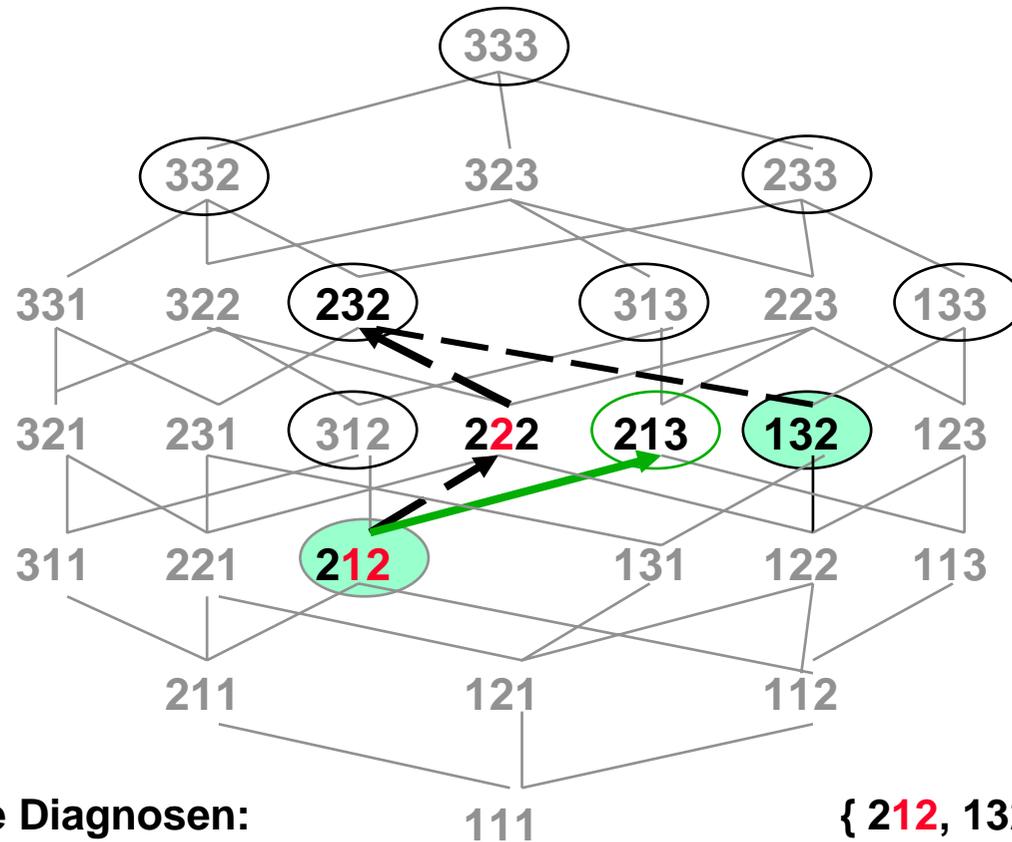
Neue Konflikte: 001, 110, 020



# Kandidatenaktualisierung

Alte Konflikte: 001, 110, 020

Neuer Konflikt: 012



Bisherige präferierte Diagnosen:

Neue präferierte Diagnosen (Stufe 1):

Neue präferierte Diagnosen (Stufe 2):

{ 212, 132 }

{ 222, 213, 132 }

{ 213, 132 }

# Kandidatenaktualisierung

## Aktionen bei Entdeckung eines neuen Konflikts:

- 1) Konsistenzcheck aller präferierten Diagnosen
- 2) Entfernen aller jetzt als inkonsistent erkannten Kandidaten
- 3) Bilden der Präferenznachfolger jedes eben entfernten Kandidaten
- 4) Aufnehmen der Präferenznachfolger, die folgende Bedingung erfüllen:
  - Der Nachfolger ist nicht von einer anderen noch als konsistent erkannten Diagnose präferiert.
  - Der Nachfolger ist selbst konsistent.

# Kandidatenaktualisierung

## 3) Bilden der Präferenznachfolger:

- Wenn C der Konflikt ist, der in der alten Diagnose enthalten ist, dann Bilden nur der Nachfolger, die den Verhaltensmodus **genau einer Komponente** ändern, **die in C enthalten** ist

(Bildung nur von direkten Nachfolgern, Nachfolgebildung bzgl. Konflikt C)

**Anmerkung: Auf diese Weise geht keine Diagnose verloren:**

**Satz:** Jede Nachfolgediagnose, die C nicht enthält, ist Nachfolgediagnose eines direkten Nachfolgers, der C nicht enthält.

- Wenn einer dieser direkten Nachfolger einen Konflikt C' enthält, dann Bildung weiterer direkter Nachfolger bzgl. C'

# Optimierungen der Kandidatengenerierung

## 1. Fokussierte Vorgehensweise:

- Ordne die präferierten und konsistenten Kandidaten nach Wahrscheinlichkeit
- Teile diese Kandidaten in 2 Mengen auf:
  - **focus:** Die besten k Kandidaten: Ihre vorhergesagten Werte werden nachfolgend vollständig weiterpropagiert, bis endgültige Konsistenz oder Inkonsistenz feststeht.
  - **candidates:** Die restlichen präferierten und konsistenten Kandidaten: Ihre vorhergesagten Werte werden vorerst nicht weiter propagiert

# Optimierungen der Kandidatengenerierung

## Aktionen bei Entdeckung eines neuen Konflikts: (fokussierte Vorgehensweise)

- Konsistenzcheck nur für die Fokusdiagnosen
- Entfernen aller inkonsistenten Kandidaten aus dem Fokus
- Bilden der direkten Präferenznachfolger jedes entfernten Fokuskandidaten bzgl. des jeweiligen Konflikts (siehe Detail)
- Einfügen dieser Nachfolger in [candidates](#), sofern sie dort präferiert sind (anderenfalls werden sie verworfen)
- Auffüllen von focus mit den wahrscheinlichsten Kandidaten aus [candidates](#)

# Optimierungen der Kandidatengenerierung

## 2. Eliminierung irrelevanter Konflikte:

- Konflikte sind nur relevant, wenn sie einen Nachfolger der gegenwärtig präferierten Diagnosen zu Fall bringen könnten:
- Berücksichtige zum Kandidatentest nur die relevanten Konflikte !

### Beispiel für relevante Konflikte:

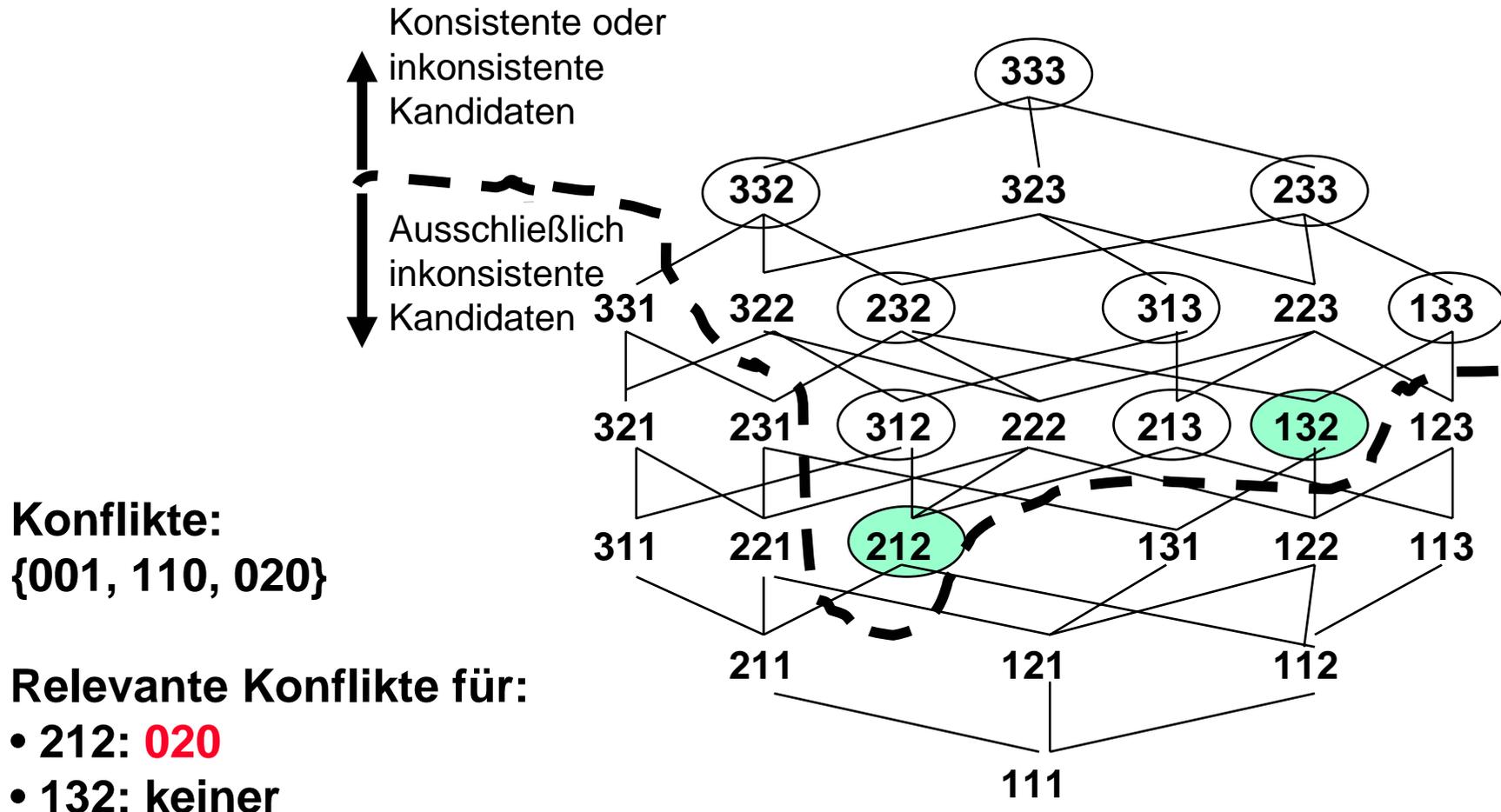
Konflikt:      0 2 2          2 0 2          2 0 2

Kandidat:      3 1 2          2 1 1          1 1 3

Relevant ?

# Optimierungen der Kandidatengenerierung

## 2. Eliminierung irrelevanter Konflikte:



# Optimierungen der Kandidatengenerierung

## 3. Effiziente Repräsentationsformen für:

- Kandidaten zum schnellen Test der Präferenzbeziehung
- Konflikte zum Testen der Relevanz für die gegenwärtigen Präferenzdiagnosen

***MDS enthält weitere Optimierungen zur Beschleunigung der Kandidatengenerierung***

***Beim nächsten Mal:  
Konfliktgenerierung***