

Wissensrepräsentation

Vorlesung

Sommersemester 2008

5. Sitzung

Dozent

Nino Simunic M.A.

Computerlinguistik, Campus DU



Wissensbasierte Systeme:

- **Regelbasierte Expertensysteme bzw. Produktionssysteme**

Rückblick: WBS, XPS

- Ein XPS ist ein wissensbasiertes System
 - Das **Wissen** ist **getrennt** von seiner **Anwendung** bzw. **Interpretation**
 - Das Wissen ist i.A. **deklarativ**, die Verarbeitung **prozedural**
- In XPS wird Wissen häufig in Form von *WENN-DANN* (**Produktions-**)**Regeln** dargestellt
 - Produktionssysteme

Wissensbasis, Inferenzmaschine

■ Wissensbasis

- Datenbank mit **Regeln**
 - Repräsentiert das Wissen des Systems
- Datenbank mit **Fakten** (»Faktenbasis«)
 - System-Input führt zu Schlussfolgerungen, und ggf. Handlungen

■ Inferenzmaschine, Interpreter

- Kontrolliert den Prozess des Schlussfolgerns

Typische Szenarien für XPS

- **Interpretation**
Analyse von Daten mit Hinblick auf ihre Bedeutung
- **Diagnose**
Erkennung von Fehlern in komplexen Systemen
- **Überwachung**
Kontinuierliche Prüfung von Signalen und ggf. Reaktion
- **Vorhersage**
Schlussfolgerung auf zukünftige Gegebenheiten
- **Planung**
Zielgerichtete Konstruktion von Handlungsabläufen
- **Entwurf**
Erzeugung von Objekten mit bestimmten Eigenschaften

Expertensysteme

■ DENTRAL

- Stanford Universität: Strukturanalyse organischer Substanzen.
- Interpretierende Daten aus Massenspektrogrammen

■ MYCIN

- Stanford University: Diagnose, Therapie bakterieller Infektionskrankheiten
- > 450 Regeln
- Inferenzmaschine unter **EMYCIN** Grundlage für andere Expertensysteme (Pionier bzgl. Expertensystem-Shell).

■ PROSPECTOR

- (Geologie) System zur Interpretation von Messdaten aus Probebohrungen. Ziel Ob ein Mineral vorhanden ist.

■ XCON

- Konfiguration von Rechneranlagen: Sicherstellung (ausgehend von der Bestellung einer Rechneranlage), dass alle erforderlichen Komponenten vorhanden sind und und zusammenspielen.
- >2000 Regeln. Meileinstein (damals) in dieser Größenordnung.

Rückblick: Regeln

IF A ... THEN B ...
A → B

Interpretationen je nach Kontext

Schlussfolgerung allgemein:

Prämisse → Konklusion

Hypothese (bspw. Diagnose):

Ursache → Wirkung

Produktion:

Bedingung → Aktion

Wie schlussfolgert die Inferenzmaschine?

- Kombination von Regeln und Fakten, um zu einem (im Idealfall) gültigen Schluss zu kommen.
- Häufig: **Modus ponens**

if A then B (Regel)

A true (Faktum)

B true (Schluss)

Deduktion als Mittel zur Schlussfolgerung auf Basis einer Menge von Antezedenten wird auch als **Vorwärtsverkettung** bezeichnet.



Simple Beispiel: Regelbasiertes System zur Lift-Steuerung

Rule 1

IF on first floor
AND button is pressed on first floor
THEN open door

Rule 2

IF on first floor
AND button is pressed on second floor
THEN go to second floor

Rule 3

IF on first floor
AND button is pressed on third floor
THEN go to third floor

Rule 4

IF on second floor
AND button is pressed on first floor
AND already going to third floor
THEN remember to go to first floor later

Fact 1

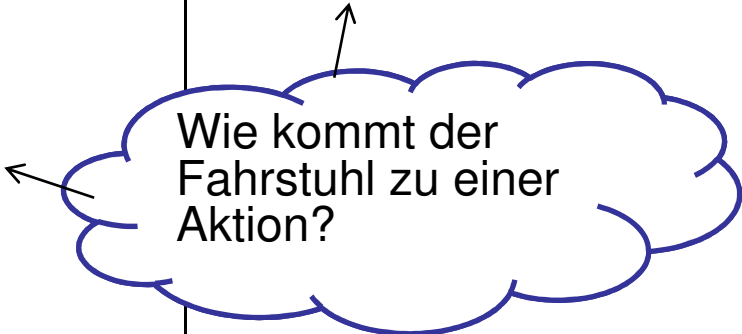
at first floor

Fact 2

button pressed on third floor

Fact 3

today is Tuesday



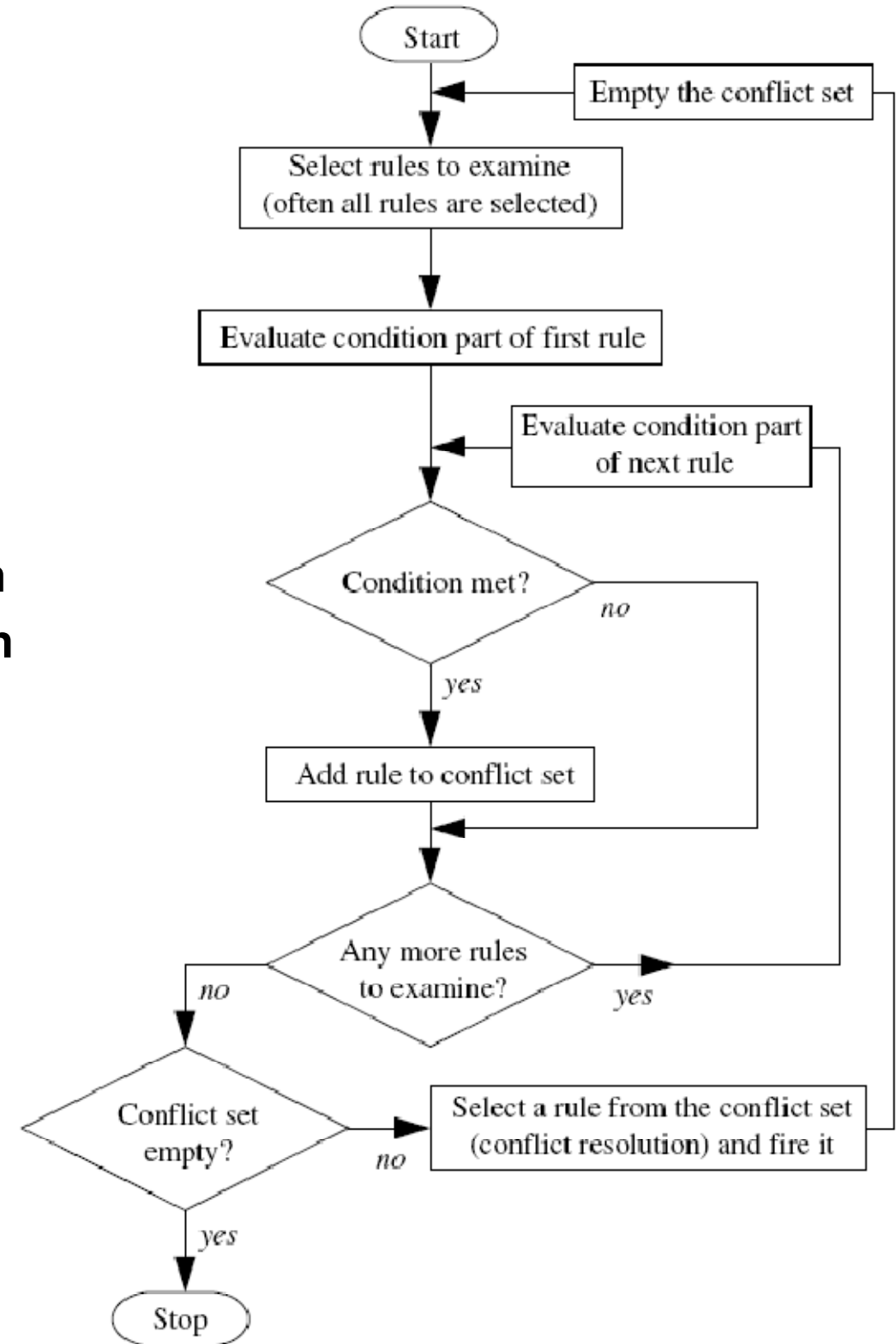
Wie kommt der
Fahrstuhl zu einer
Aktion?

Grober, generischer Ablauf des Schließens:

- (1) Auswahl der zu **evaluierenden Regeln**
 - Verfügbare Regeln (*engl.* available rules)
- (2) Bestimmung der **anwendbaren Regeln**
 - *Engl.* to **trigger a rule**
 - **Konfliktmenge** (*engl.* conflict set)
- (3) Auswahl der zu **feuernden Regel**
 - *Engl.* to **fire a rule**
 - **Konfliktresolution** (*engl.* conflict resolution)

Vorwärtsverkettung: Generischer Ablauf

- (1) Auswahl der zu **evaluierenden Regeln**
- (2) Bestimmung der **anwendbaren Regeln**
- (3) Auswahl der zu **feuernden Regel**



Auswahl der zu **evaluierenden Regeln**

Rule 1

IF on first floor
AND button is pressed on first floor
THEN open door

Rule 2

IF on first floor
AND button is pressed on second floor
THEN go to second floor

Rule 3

IF on first floor
AND button is pressed on third floor
THEN go to third floor

Rule 4

IF on second floor
AND button is pressed on first floor
AND already going to third floor
THEN remember to go to first floor later

Fact 1

at first floor

Fact 2

button pressed on third floor

Fact 3

today is Tuesday

Auswahl der zu **evaluierenden Regeln**

Auswahl der anwendbaren Regeln

Rule 1

IF
AND
THEN

on first floor
button is pressed on first floor
open door

Rule 2

IF
AND
THEN

on first floor
button is pressed on second floor
go to second floor

Rule 3

IF
AND
THEN

on first floor
button is pressed on third floor
go to third floor

Rule 4

IF
AND
AND
THEN

on second floor
button is pressed on first floor
already going to third floor
remember to go to first floor later

Fact 1

at first floor

Fact 2

button pressed on third floor

Fact 3

today is Tuesday

Keine gültiges Antezedens
gefunden

Regel feuert

Conflict set: { go(ing) to third floor }

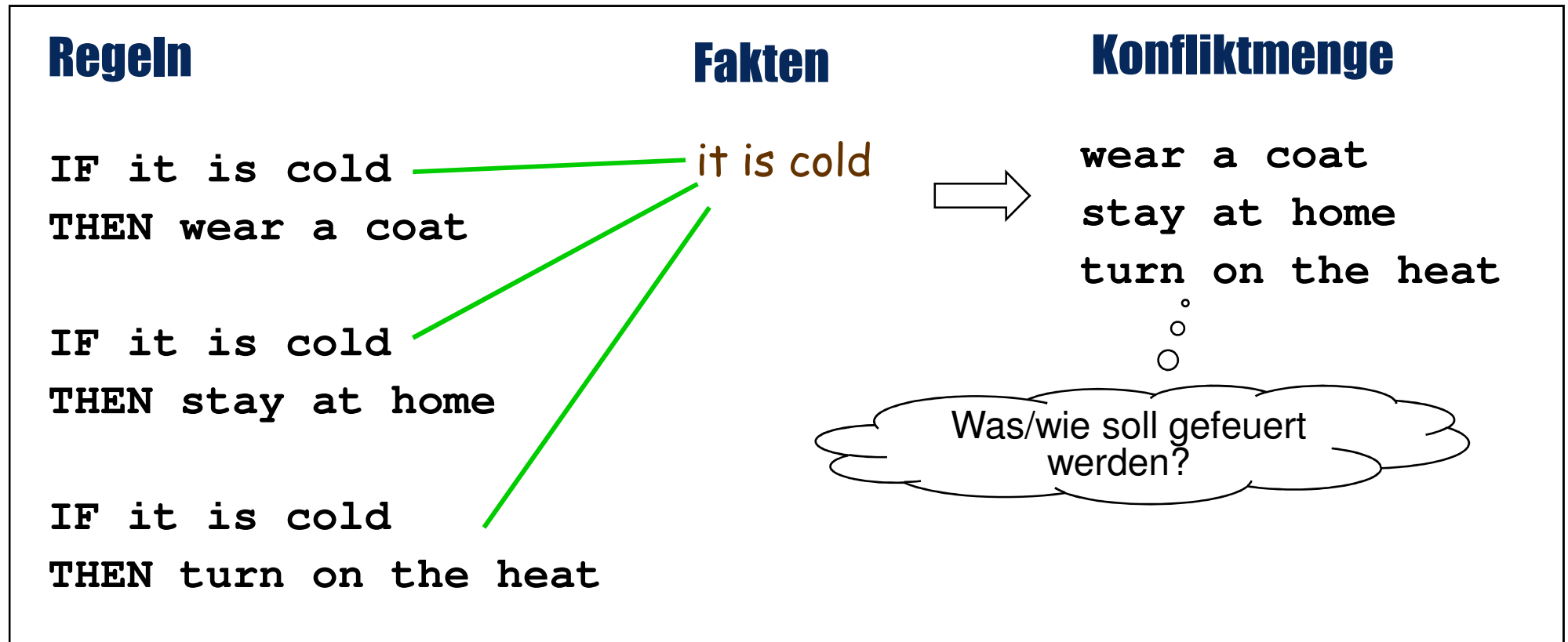
Die Konfliktmenge (**Forward Chaining**)

- Auswahl der feuernden Regel aus d. Konfliktmenge trivial im letzten Fall
 - *go(ing) to third floor* als einzige anwendbare Regel (triggered rule)
 - Wird zu den Fakten hinzugefügt
 - Führt dazu, dass der Lift die Aktion ausführt (in den dritten Stock fahren).
- Denkbar: Mehrere feuerbereite Regeln in der Konfliktmenge → Konfliktresolutionsstrategie

Gängige Konfliktresolutionsstrategien

- Reihenfolge
 - **Erste anwendbare Regel** feuert (Trivialstrategie)
 - **Aktuellste Regel** feuert, d.h. die Regel, deren Vorbedingungen sich auf möglichst neue Einträge in der Faktenmenge bezieht.
- Auswahl nach der syntaktischen Struktur der Regel
 - Die **spezifischste Regel** feuert: Die Regel mit den meisten Aussagen in der Vorbedingung ist die spezifischste Regel.
- Auswahl mittels **Zusatzwissen (Metawissen)**
 - Die **Regel mit der höchsten Priorität** feuert. Dazu muss jeder Regel eine Priorität, die i.d.R. numerisch ist, zugeordnet werden.
 - Zusätzliche Regeln (**Meta-Regeln**) steuern den Auswahlvorgang.

Wetter-Beispiel: Mehrere anwendbare Regeln



Lift-Beispiel: Mehrere anwendbare Regeln

Rule 1

IF on first floor
AND button is pressed on first floor
THEN open door

Rule 2

IF on first floor
AND button is pressed on second floor
THEN go to second floor

Rule 3

IF on first floor
AND button is pressed on third floor
THEN go to third floor

Rule 4

IF on second floor
AND button is pressed on first floor
AND already going to third floor
THEN remember to go to first floor later

Fact 1

At first floor

Fact 2

Button pressed on second floor

Fact 3

Button pressed on third floor

Lift-Beispiel: Mehrere anwendbare Regeln

Rule 1

IF on first floor
AND button is pressed on first floor
THEN open door

Rule 2

IF on first floor
AND button is pressed on second floor
THEN go to second floor

Rule 3

IF on first floor
AND button is pressed on third floor
THEN go to third floor

Rule 4

IF on second floor
AND button is pressed on first floor
AND already going to third floor
THEN remember to go to first floor later

Fact 1

At first floor

Fact 2

Button pressed on second floor

Fact 3

Button pressed on third floor

Konfliktresolutionsstrategie: *Priorität*

- Regeln erhalten Prioritäten zur Konfliktauflösung.
 - Regel mit höchster Priorität wird immer zuerst gefeuert.
- Beispiel: Folgende Regeln werden in die Konfliktmenge genommen.

```
IF patient has pain  
THEN prescribe painkillers priority 10
```

```
IF patient has chest pain  
THEN treat for heart disease priority 100
```

Gefeuerte Regel

Konfliktresolutionsstrategie: *Longest Matching* / Syntaktische Länge

- Die **spezifischste Regel** wird gefeuert
 - Annahme: Die längsten Regeln sind die spezifischsten Regeln und haben Vorrang.
- Beispiel: Folgende Regeln werden in die Konfliktmenge genommen.

```
IF patient has pain  
THEN prescribe painkiller
```

```
IF patient has chest pain  
AND patient is over 60  
AND patient has history of heart conditions  
THEN take to emergency room
```

Gefeuerte Regel

Metaregeln

- **Metawissen** – Wissen über Wissen.
- **Metaregeln**: Strategie zum Feuern der Regeln in einer Konfliktmenge.
 - »Expert A is to be trusted more than expert B«
 - »Any rule that involves drug X is not to be trusted as much as rules that do not involve X«
- Metaregeln werden in XPS wie herkömmliche Regeln behandelt, erhalten jedoch eine höhere Priorität als »normale« Regeln.

Backward chaining

- Beweis der Gültigkeit einer Schlussfolgerung, bzw. **Hypothese**
 - Zu prüfende Schlussfolgerung: Ziel (*goal*)
 - Ziel-orientiertes Schließen (vs. Daten-orientiertes Schließen)
 - »Welche Regeln müssen feuern, um zum Ziel zu kommen«?
- Vorwärtsverkettung und Rückwärtsverkettung konkurrieren bei der Problemlösung
 - Vergleich der Verfahren am Beispiel

Beispiel Regeln und Fakten

Regeln:

Regel 1	$A \wedge B \rightarrow C$
Regel 2	$A \rightarrow D$
Regel 3	$C \wedge D \rightarrow E$
Regel 4	$B \wedge E \wedge F \rightarrow G$
Regel 5	$A \wedge E \rightarrow H$
Regel 6	$D \wedge E \wedge H \rightarrow I$

Fakten:

Faktum 1	A
Faktum 2	B
Faktum 3	F

Ziel: Der Beweis von H .

Vorwärtsverkettung (1)

Regeln:

Regel 1	$A \wedge B \rightarrow C$
Regel 2	$A \rightarrow D$
Regel 3	$C \wedge D \rightarrow E$
Regel 4	$B \wedge E \wedge F \rightarrow G$
Regel 5	$A \wedge E \rightarrow H$
Regel 6	$D \wedge E \wedge H \rightarrow I$

Regeln 1 und 2 werden getriggert.
(Resolution: Reihenfolge)

1. Regel 1 feuert $\rightarrow C$ wird den Fakten hinzugefügt.
2. Regel 2 feuert $\rightarrow D$ wird den Fakten hinzugefügt.

Initiale Fakten:

Faktum 1	A
Faktum 2	B
Faktum 3	F

Ziel: Der Beweis von **H**.

Facts	Rules triggered	Rule fired
A, B, F	1, 2	1
A, B, C, F	2	2
A, B, C, D, F	3	3

Vorwärtsverkettung (2)

Regeln:

Regel 1	$A \wedge B \rightarrow C$
Regel 2	$A \rightarrow D$
Regel 3	$C \wedge D \rightarrow E$
Regel 4	$B \wedge E \wedge F \rightarrow G$
Regel 5	$A \wedge E \rightarrow H$
Regel 6	$D \wedge E \wedge H \rightarrow I$

**Regel 3 wird getriggert
und gefeuert. $\rightarrow$ E zu den Fakten.**

Ziel: Der Beweis von **H**.

Facts	Rules triggered	Rule fired
A, B, F	1, 2	1
A, B, C, F	2	2
A, B, C, D, F	3	3
A, B, C, D, E, F	4, 5	4

Vorwärtsverkettung (3)

Regeln:

Regel 1	$A \wedge B \rightarrow C$
Regel 2	$A \rightarrow D$
Regel 3	$C \wedge D \rightarrow E$
Regel 4	$B \wedge E \wedge F \rightarrow G$
Regel 5	$A \wedge E \rightarrow H$
Regel 6	$D \wedge E \wedge H \rightarrow I$

Regeln 4 and 5 werden getriggert.

1. Regel 4 feuert zuerst $\rightarrow$ Faktum G zu den Fakten
2. Regel 5 feuert $\rightarrow$ Faktum H wird den Fakten hinzugefügt.

Ziel: Der Beweis von H.

**Ziel bewiesen,
H existiert in den Fakten.**

Facts	Rules triggered	Rule fired
A, B, F	1,2	1
A, B, C, F	2	2
A, B, C, D, F	3	3
A, B, C, D, E, F	4,5	4
A, B, C, D, E, F, G	5	5
A, B, C, D, E, F, G, H	6	STOP

Rückwärtsverkettung (1)

Regeln:

Regel 1	$A \wedge B \rightarrow C$
Regel 2	$A \rightarrow D$
Regel 3	$C \wedge D \rightarrow E$
Regel 4	$B \wedge E \wedge F \rightarrow G$
Regel 5	$A \wedge E \rightarrow H$
Regel 6	$D \wedge E \wedge H \rightarrow I$

Initiale Fakten:

Faktum 1	A
Faktum 2	B
Faktum 3	F

Ziel: Der Beweis von **H**.

Welche Regeln müssen feuern, damit wir zum Ziel **H** kommen?

Regel 5 hat als einzige Regel **H** als Konklusion.

Um **H** zu beweisen, müssen wir die Antezedenten von **H** beweisen:
A und **E**.

Facts	Goals	Matching rules
A, B, F	H	5

Neben Regel-, und Faktenbasis auch »Ziel-Basis« nötig

Rückwärtsverkettung (2)

Regeln:

- Regel 1 $A \wedge B \rightarrow C$
Regel 2 $A \rightarrow D$
Regel 3 $C \wedge D \rightarrow E$
Regel 4 $B \wedge E \wedge F \rightarrow G$
Regel 5 $A \wedge E \rightarrow H$
Regel 6 $D \wedge E \wedge H \rightarrow I$

Faktum A bereits in den Fakten vorhanden – es muss nur E bewiesen werden.
→ E wird den Zielen hinzugefügt.

Facts	Goals	Matching rules
A, B, F	H	5
A, B, F	E	3

Der Beweis von E ist ausreichend, um H zu beweisen, und H aus der Ziel-Liste zu streichen. Warum?

Rückwärtsverkettung (3)

Regeln:

- Regel 1 $A \wedge B \rightarrow C$
Regel 2 $A \rightarrow D$
Regel 3 $C \wedge D \rightarrow E$
Regel 4 $B \wedge E \wedge F \rightarrow G$
Regel 5 $A \wedge E \rightarrow H$
Regel 6 $D \wedge E \wedge H \rightarrow I$

Jetzt der Beweis von Faktum E.

Regel 3 hat E als
Schlussfolgerung:

Um E zu beweisen, müssen wir
demnach die Antezedenten von
Regel 3 beweisen: C, D.

C, D sind noch nicht bewiesen
(nicht in den Fakten) → Beide
sind zu prüfen und werden den
Zielen hinzugefügt.

Facts	Goals	Matching rules
A, B, F	H	5
A, B, F	E	3
A, B, F	C, D	1

Rückwärtsverkettung (4)

Regeln:

- Regel 1 $A \wedge B \rightarrow C$
Regel 2 $A \rightarrow D$
Regel 3 $C \wedge D \rightarrow E$
Regel 4 $B \wedge E \wedge F \rightarrow G$
Regel 5 $A \wedge E \rightarrow H$
Regel 6 $D \wedge E \wedge H \rightarrow I$

Facts	Goals	Matching rules
A, B, F	H	5
A, B, F	E	3
A, B, F	C, D	1
A, B, C, F	D	2
A, B, C, D, F		STOP

D ist der Schluss in Regel 2. Zu beweisen ist A.

A ist bereits bewiesen (siehe Fakten), somit auch D.
→ D zu den Fakten.

Analog dazu: C ist der Schluss in Regel 1. Seine Antezedenten sind bereits bewiesen (siehe Fakten).
→ C zu den Fakten.

Alle Ziele der Ziel-Basis wurden bewiesen. Somit wurde auch H bewiesen.
→ Inferenz-Stop.

Diskussion

- Rückwärtsverkettung benötigte *hier* einen Schritt weniger.
- Generell:
 - Rückwärtsverkettung sinnvoll, wenn wenige Schlussfolgerungen und viele Fakten vorhanden sind.
 - **Backtracking-Problem** (→ Vorlesungssitzung zu natürlichsprachlichen System/Syntaktisches Wissen)
 - Vorwärtsverkettung sinnvoll bei vielen möglichen Schlussfolgerungen.
- Die Wahl des Vorgehens hängt von den Problem(-felder-)n ab.
- In XPS ist es i.d.R. der Experte, der auf Basis seiner Erfahrungen die bessere Methode auswählt.

CLIPS: C Language Integrated Production System

- Freies XPS Shell, implementiert in C.
 - Sprache, um Regeln auszudrücken.
 - Hauptsächlich vorwärtsverkettend.
 - Notation ähnlich zu LISP
 - <http://clipsrules.sourceforge.net/>

```
(defrule birthday
  (firstname ?r1 John)
  (surname ?r1 Smith)
  (haircolor ?r1 Red)
=>
  (assert (is-boss ?r1)))
```

?r1 ist eine Variable (Person)

assert wird verwendet, um
Fakten zur Faktenbasis
hinzuzufügen.

Hier: Regel, um aus drei Fakten
über eine Person zu
schlussfolgern.

Wenn die Person einen Vornamen John hat, und der Nachname Smith ist, und er rotes Haar ist, ist er der Boss.

CLIPS: assert, facts

- Hinzufügen von Fakten

```
(assert (firstname x John))  
(assert (surname x Smith))  
(assert (haircolor x Red))  
(run)
```

- Befehl (*facts*), um die Fakten der Faktenbasis anzuzeigen:

```
CLIPS> (facts)  
f-0 (firstname x John)  
f-1 (surname x Smith)  
f-2 (haircolor x Red)  
f-3 (is-boss x)
```

- Neben den gegebenen Fakten, hat CLIPS folglich ein neues Faktum (via Boss-Regel-Schlussfolgerung) generiert.

(Self-)Reminder

- Übungsgruppe im Anschluss: Nächste Woche
- Freitagsgruppe: Bitte diese Woche kommen, die nächste Woche dafür nicht.

Referenzen

