

Logische und funktionale Programmierung

Vorlesung 4: Such- und Beweisbäume in PROLOG

Babeş-Bolyai Universität, Department für Informatik, Cluj-Napoca
csacarea@cs.ubbcluj.ro

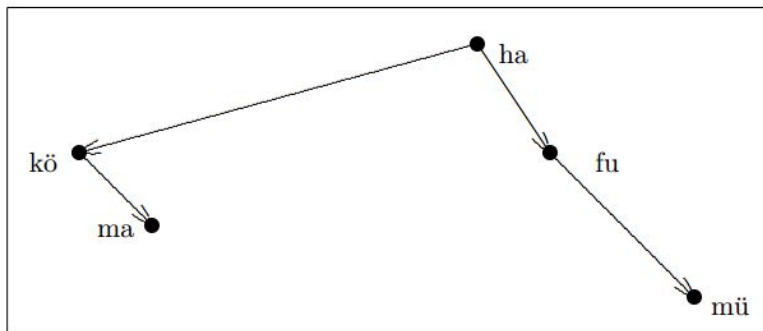
10. November 2017



WISSENSBASIERTE SYSTEME

Aufgabenstellung

Nach Bekanntgabe von Abfahrts- und Zielort soll sich ein Bahnreisender - als Anwender des Systems - anzeigen lassen können, ob es zwischen diesen beiden Orten eine Zugverbindung im Intercity (IC) Netz der Bundesbahn gibt.



WISSENSBASIERTE SYSTEME

Dieses Netz enthält die Stationen Hamburg (“ha”), Köln (“kö”), Fulda (“fu”), Mainz (“ma”) und München (“mü”). Der Zeichnung entnehmen wir, daß die folgende *Aussage* (engl.: assertion) zutrifft:

<es gibt eine IC-Verbindung von “ha” nach “mü” über “fu”>

Innerhalb dieser Aussage wird ein Sachverhalt in der Form

<es gibt eine IC-Verbindung von ... nach ... über ...>

als *Beziehung* zwischen den *Objekten* “ha”, “mü” und “fu” beschrieben. Wir bezeichnen diese Beziehung als *Prädikat* (engl.: predicate) und wählen für sie eine abkürzende Darstellung, indem wir sie wie folgt kennzeichnen:

ic_verbindung_über(ha,mü,fu)



AUFBAU DER WISSENSBASIS

es gibt eine IC-Verbindung von ... nach ...;

Kennzeichnen wir dieses Prädikat durch den Prädikatsnamen

ic , so geben die Fakten

$ic(ha, kö)$.

$ic(kö, ma)$.

$ic(ha, ma)$.

$ic(ha, fu)$.

$ic(fu, mü)$.

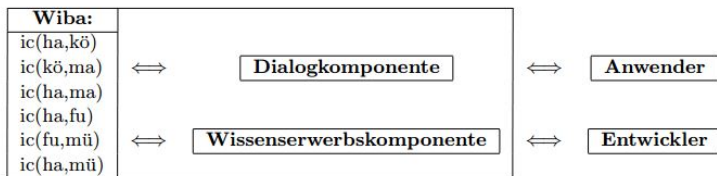
$ic(ha, mü)$. den in der Abbildung gesetzten Rahmen wider.

Wissensbasis = Sammlung von Fakten.

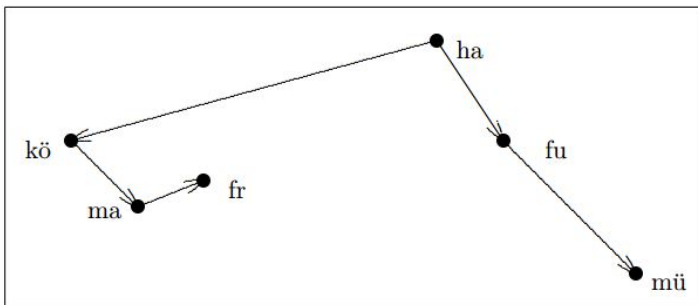


WISSENSERWERB KOMPONENTE

Knowledge Discovery



WISSENSBASIS AKTUALISIEREN



`ic(ha, fr).`

`ic(kö, fr).`

`ic(ma, fr).`

⇒ Nur Direktverbindungen sind ausreichend!

DIREKTE VERBINDUNGEN

`dic(ha,kö) .`

`dic(ha,fu) .`

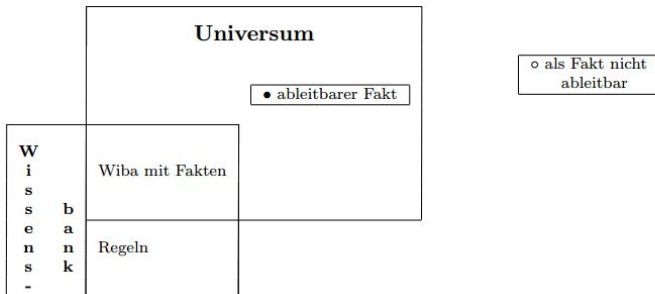
`dic(kö,ma) .`

`dic(fu,mü) .`



WISSENSABLEITUNG

Inferenz

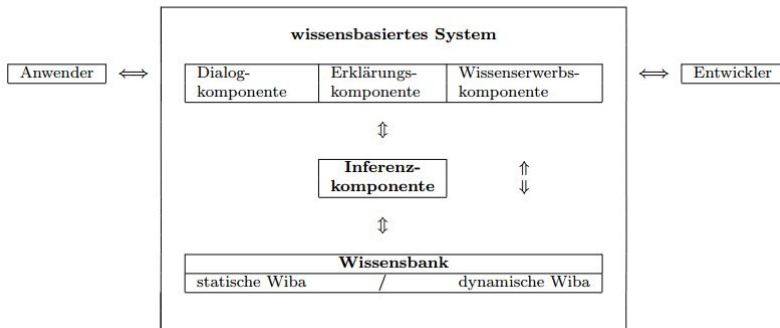


WISSENBASIS RELOADED

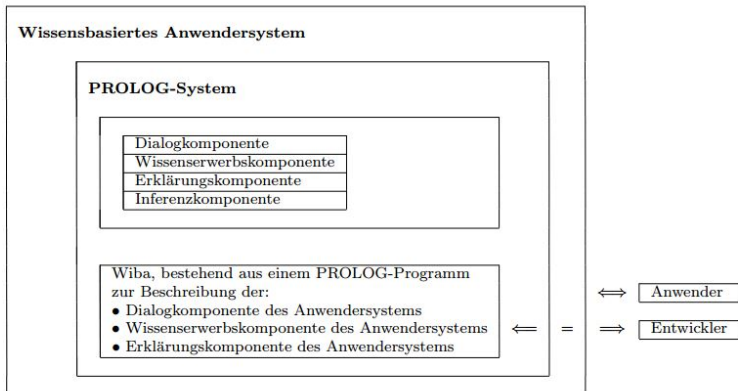
W
i
s
s
e
n
s
b
a
n
k

statische Wissensbasis	dynamische Wissensbasis
Fakten: <div>dic(ha,kö)</div> <div>dic(ha,fu)</div> <div>dic(kö,ma)</div> <div>dic(fu,mü)</div>	mögliche ableitbare Fakten: <div>ic(ha,kö)</div> <div>ic(kö,ma)</div> <div>ic(ha,ma)</div> <div>ic(ha,fu)</div> <div>ic(fu,mü)</div> <div>ic(ha,mü)</div>
Regeln zur Bestimmung einer IC-Verbindung aus den Direktverbindungen	

WISSENSBASIERTE SYSTEME



PROLOG SYSTEME ALS WISSENSBASIERTE SYSTEME



ERWEITERUNG MIT REGELN

Regelkopf

Regelrumpf

$\text{zwischen}(\text{Von}, \text{Nach}): - \text{dic}(\text{Von}, \text{Z}) \text{ , } \text{dic}(\text{Z}, \text{Nach}).$
--

$\text{dic}(\text{ha}, \text{kö}).$

$\text{dic}(\text{ha}, \text{fu}).$

$\text{dic}(\text{kö}, \text{ma}).$

$\text{dic}(\text{fu}, \text{mü}).$

$\text{zwischen}(\text{Von}, \text{Nach}) :- \text{dic}(\text{Von}, \text{Z}), \text{dic}(\text{Z}, \text{Nach}).$



WIEDERHOLUNG: PROLOG FUNKTIONSWEISE

Instanziierung und Unifizierung

Um ein Goal auf seine Ableitbarkeit hin zu überprüfen, sucht die Inferenzkomponente - beginnend mit der 1. Klausel der Wiba - nach dem ersten Klauselkopf, der denselben Prädikatsnamen und die gleiche Anzahl an Argumenten wie das Goal besitzt.



INSTANZIERUNG UND UNIFIZIERUNG

Goal: zwischen(ha,mü).

Fakten:

dic(ha,kö).
dic(ha,fu).
dic(kö,ma).
dic(fu,mü).

Regel:

zwischen(Von,Nach) :- dic(Von,Z),dic(Z,Nach).

INSTANZIERUNG UND UNIFIZIERUNG

- Im Hinblick auf das von uns angegebene Goal scheiden die Fakten hierbei aus.
- Es bleibt allein der Regelkopf zwischen (Von, Nach), der genau wie das Goal den Prädikatsnamen zwischen mit 2 Argumenten besitzt.
- Eine vollständige Übereinstimmung von Goal und Regelkopf ist dann gegeben, wenn die Variable Von durch die Konstante ha und die Variable Nach durch die Konstante mü ersetzt wird.
- Eine derartige Ersetzung wird Instanzierung (engl.: instantiation) genannt.
- Sie kennzeichnet den Vorgang, dass eine Variable mit einem Wert belegt und dadurch an einen Wert gebunden wird.



INSTANZIERUNG UND UNIFIZIERUNG

- Die Durchführung derartiger Instanzierungen von Variablen im Hinblick darauf, dass ein Prädikat - als Zeichenmuster - vollständig mit einem Klauselkopf übereinstimmt, wird **Unifizierung (engl.: unification)** genannt.
- Beim Unifizieren wird für die Prädikatsnamen und deren Argumente ein reiner Abgleich von Zeichenmustern (**pattern matching**) durchgeführt, indem - bei beiden Prädikaten - Zeichen für Zeichen miteinander verglichen wird.
- Ein Goal und ein Klauselkopf (ein Fakt oder ein Regelkopf) lassen sich dann unifizieren (**matchen, treffen, gleichmachen, in Übereinstimmung bringen**), wenn:



INSTANZIERUNG UND UNIFIZIERUNG

- die Prädikatsnamen des Goals und des Klauselkopfs identisch sind, und die Anzahl der Argumente in beiden Prädikaten gleich ist, und
 - falls in den Argumenten des Regelkopfs bzw. des Goals Variable auftreten, so müssen die Variablen im Regelkopf und im Goal so instanziiert werden können, dass die korrespondierenden Argumente in beiden Prädikaten in ihren Zeichenmustern übereinstimmen, und
 - falls in den Argumenten des Regelkopfs bzw. des Goals Konstante vorkommen, so müssen diejenigen Konstanten, die bzgl. ihrer Argumentpositionen miteinander korrespondieren, identisch sein.



ABLEITBARKEIT

Beim Versuch, ein Goal abzuleiten, durchsucht die Inferenzkomponente das PROLOG-Programm von oben nach unten, bis eine (erste) Unifizierung des Goals mit einem Fakt oder einem Regelkopf möglich ist.
Läßt sich keine Unifizierung durchführen, so ist das Goal nicht ableitbar.



ABLEITBARKEITMECHANISMUS

In unserem Fall wird eine Unifizierung des Goals “zwischen(ha,mü)” mit dem Klauselkopf “zwischen(Von,Nach)” versucht:

Goal:	zwischen(ha,mü).	
Regel:	zwischen(Von,Nach):- dic(Von,Z) , dic(Z,Nach).	
	↑	↑
	1. Subgoal	2. Subgoal

Wie oben beschrieben, lassen sich das Goal “zwischen(ha,mü)” und der Regelkopf “zwischen(Von,Nach)” dadurch unifizieren, daß die Variablen “Von” und “Nach” mit den Werten “ha” bzw. “mü” instanziiert werden.

ABLEITBARKEITMECHANISMUS

Das Goal zwischen (ha, mü) ist - gemäß der Regel - somit dann aus der Wiba ableitbar, wenn der Regelrumpf des unfizierten Regelkopfs ableitbar ist:

Goal:	zwischen(ha,mü).	
Regel:	zwischen(Von,Nach):-	dic(Von,Z) , dic(Z,Nach).
Instanzierung:	zwischen(ha,mü):-	dic(ha,Z) , dic(Z,mü).
	(Von:=ha,Nach:=mü)	↑ ↑
		1. Subgoal 2. Subgoal

Dieser Regelrumpf — als *logische UND-Verbindung* zweier Prädikate — ist dann ableitbar, wenn sich *sowohl* das 1. Prädikat “dic(ha,Z)” *als auch* das 2. Prädikat “dic(Z,mü)” — für eine geeignete Instanzierung von “Z” — ableiten lassen. Um dies überprüfen zu können, werden die beiden Prädikate “dic(ha,Z)” und “dic(Z,mü)” als neue Goals — zur Unterscheidung nennen wir sie *Subgoals* (*Teilziele*) — aufgefaßt. In dieser Situation wird das ursprüngliche Goal, das mit dem Klauselkopf unfiziert wurde, als *Parent-Goal* (*Eltern-Ziel*) der jetzigen Subgoals bezeichnet.

ABLEITBARKEITMECHANISMUS

Zunächst wird versucht, das 1. Subgoal “dic(ha,Z)” innerhalb der Wiba zu unifizieren. Gelingt dies, so ist der Wert, mit dem die Variable “Z” bei dieser Unifizierung instanziiert wird, für die Variable “Z” im 2. Subgoal “dic(Z,mü)” einzusetzen. Sofern das 2. Subgoal mit dieser Instanzierung von “Z” anschließend ebenfalls innerhalb der Wiba unifiziert werden kann, sind sowohl die beiden Subgoals und — wegen der logischen UND-Verbindung — folglich auch der Regelkopf und somit auch das Goal ableitbar.

Bei der Überprüfung der Ableitbarkeit des 1. Subgoals haben wir die folgende Ausgangssituation:

CALL: dic(ha,Z)		
dic(ha,kö).		
dic(ha,fu).		
dic(kö,ma).		
dic(fu,mü).		

ABLEITBARKEITMECHANISMUS

- Mit diesem Schema beschreiben wir im folgenden den jeweiligen Stand der Ableitbarkeits-Prüfung.
- Wie die jeweilige Untersuchung der (in der ersten Spalte untereinander eingetragenen) Fakten der Wiba verläuft, kennzeichnen wir innerhalb der 3. Spalte durch die Schlüsselwörter **EXIT**, **REDO** und **FAIL**.
- **EXIT**: erfolgreiche Unifizierung
- **REDO**: erneut erforderliche Ableitbarkeits-Prüfung und
- **FAIL**: das endgültige Scheitern der Ableitbarkeits-Prüfung des aktuellen Subgoals.
- Sofern eine erfolgreiche, durch **EXIT** gekennzeichnete Unifizierung gelungen ist, werden wir die zugehörige Instanzierung in der 2. Spalte des Schemas vermerken.
- **TRACE**



TRACE

→

CALL: dic(ha,Z)		
dic(ha,kö).	Z:=kö	EXIT: *dic(ha,kö)
dic(ha,fu).		
dic(kö,ma).		
dic(fu,mü).		

Für die Variable “Z” müssen wir jetzt überall, wo diese Variable im Regelrumpf auftritt, den Wert “kö” einsetzen. Damit stellt sich das 2. Subgoal in der Form “dic(kö,mü)” dar.

TRACE

- Bevor die PROLOG-Inferenzkomponente versucht, dieses 2. Subgoal abzuleiten, markiert sie - intern - die Klausel $\text{dic}(\text{ha}, \text{kö})$. als **BacktrackingKlausel (engl.: Choicepoint)**.
- Zu dieser Backtracking-Klausel wird dann zurückgekehrt, wenn der Versuch, das 2. Subgoal aus der Wiba abzuleiten, fehlschlägt.
- Go To nächste Klausel als weitere Alternative für eine mögliche Unifizierung des 1. Subgoals untersucht.
- Die Überprüfung, ob das 2. Subgoal $\text{dic}(\text{kö}, \text{mü})$ (Z ist mit kö instanziiert) aus der Wiba abgeleitet werden kann, ist von der zuvor durchgeführten Ableitbarkeits-Prüfung des 1. Subgoals unabhängig, da das aktuelle 2. Subgoal eine weitere Komponente der UND-Verbindung darstellt.



TRACE

Für die Ableitbarkeits-Prüfung des 2. Subgoals betrachten wir ein weiteres Exemplar der Wiba:

CALL: $\text{dic}(\text{kö}, \text{mü})$		
$\text{dic}(\text{ha}, \text{kö}).$		
$\text{dic}(\text{ha}, \text{fu}).$		
$\text{dic}(\text{kö}, \text{ma}).$		
$\text{dic}(\text{fu}, \text{mü}).$		

TRACE

Jetzt wird dieses neue Exemplar der Wiba wiederum - von oben nach unten - danach untersucht, ob eine Unifizierung mit dem jetzt aktuellen Subgoal `dic(kö,mü)` möglich ist. Die Ableitung des aktuellen Subgoals gelingt nicht, da es in unserem PROLOG-Programm keinen Fakt gibt, der sich mit `dic(kö,mü)` unifizieren lässt.

→

CALL: dic(ha,Z)		
dic(ha,kö).	Z:=kö	EXIT: *dic(ha,kö)
dic(ha,fu).		
dic(kö,ma).		
dic(fu,mü).		

→

CALL: dic(kö,mü)		
dic(ha,kö).		REDO: dic(kö,mü)
dic(ha,fu).		REDO: dic(kö,mü)
dic(kö,ma).		REDO: dic(kö,mü)
dic(fu,mü).		FAIL: dic(kö,mü)

BACKTRACKING

- Die Unifizierung des 2. Subgoals ist fehlgeschlagen, d.h. die Instanziierung der Variablen z bei der Unifizierung des 1. Subgoals führt nicht dazu, dass das 1. Subgoal und anschließend das 2. Subgoal abgeleitet werden können.
- Suche eine alternative Unifizierung des 1. Subgoals.
- Um eine Alternative zu finden, kehrt die PROLOG-Inferenzkomponente zur (letzten) Backtracking-Klausel zurück, d.h. derjenigen Klausel, für welche die Unifizierung des 1. Subgoals (unmittelbar) zuvor gelungen war.
- Die Instanziierung $z := k\ddot{o}$ wird jetzt aufgehoben \Rightarrow die Variable z ist wieder ungebunden.
- Es wird also die Unifizierung des 1. Subgoals $dic(ha, z)$ ab derjenigen Klausel versucht, die unmittelbar hinter (unter) der Backtracking-Klausel im PROLOG-Programm enthalten ist.



BACKTRACKING

- Der Vorgang, nach einem fehlgeschlagenen Unifizierungsversuch einen alternativen Ansatz zur Ableitung eines Goals bzw. Subgoals zu finden, wird **Backtracking** genannt.
- Kann also ein Subgoal nicht unifiziert werden, so werden beim Backtracking alle Instanzierungen von Variablen gelöst, die bei der Unifizierung des - innerhalb desselben Regelrumpfs - unmittelbar zuvor unifizierten (vorausgehenden, *links stehenden*) Subgoals vorgenommen wurden.
- Anschließend wird eine erneute Unifizierung des vorausgehenden Subgoals ab derjenigen Klausel versucht, die der BacktrackingKlausel, d.h. der zuletzt unifizierten Klausel, folgt.
- Schlägt dieser erneute Versuch wiederum fehl, so erfolgt wiederum ein Backtracking zur Backtracking-Klausel des dem aktuellen Subgoal vorausgehenden Subgoals.



BACKTRACKING

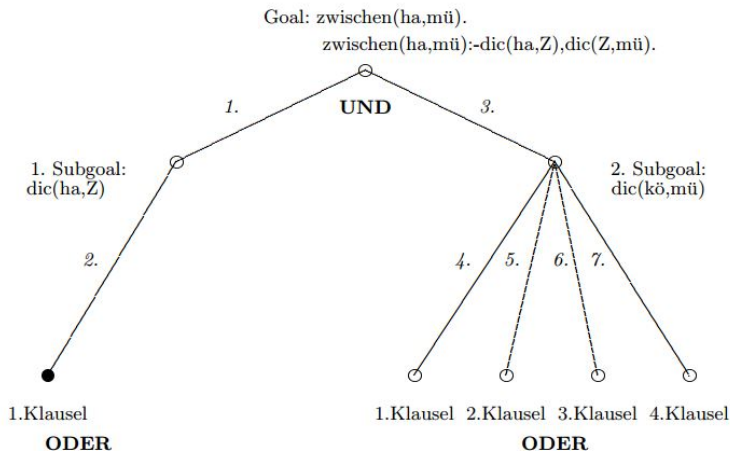
→

REDO: dic(ha,Z)		
dic(ha,kö).		
dic(ha,fu).	Z:=fu	EXIT: *dic(ha,fu)
dic(kö,ma).		
dic(fu,mü).		

→

CALL: dic(fu,mü)		
dic(ha,kö).		REDO: dic(fu,mü)
dic(ha,fu).		REDO: dic(fu,mü)
dic(kö,ma).		REDO: dic(fu,mü)
dic(fu,mü).		EXIT: dic(fu,mü)

ABLEITUNGSBAUM

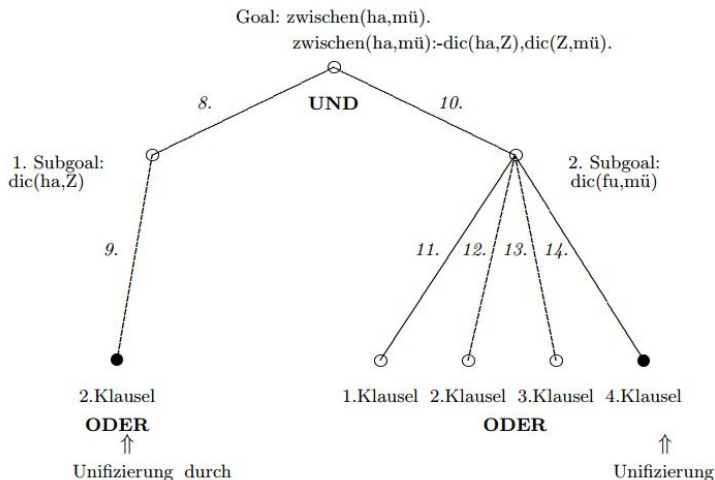


Unifizierung durch

Instanzierung:

Z:=kö

ABLEITUNGSBAUM



mit:

Instanzierung:
Z:=fu

dic(fu,mü).



REGELN MIT ODER VERBINDEN

$\text{ic}(\text{Von}, \text{Nach}) \text{ :- } \text{dic}(\text{Von}, \text{Nach}); \text{dic}(\text{Von}, \text{Z}), \text{dic}(\text{Z}, \text{Nach}).$

$\text{dic}(\text{ha}, \text{kö}).$

$\text{dic}(\text{ha}, \text{fu}).$

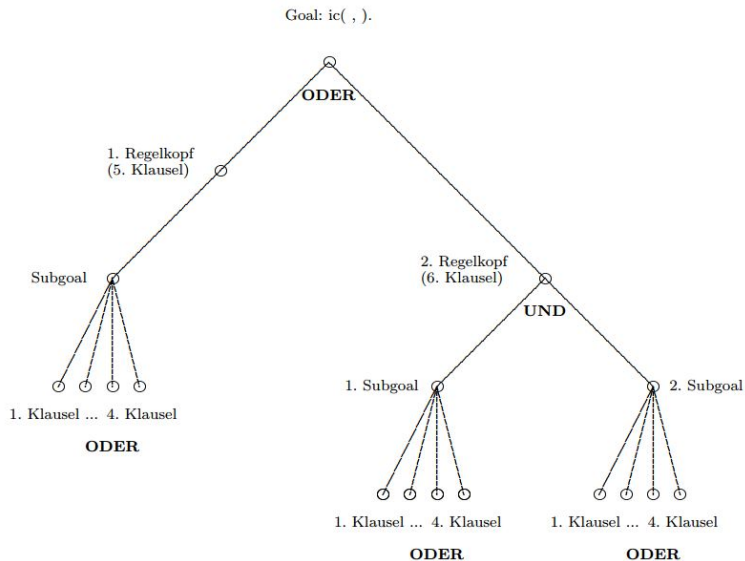
$\text{dic}(\text{kö}, \text{ma}).$

$\text{dic}(\text{fu}, \text{mü}).$

$\text{ic}(\text{Von}, \text{Nach}) \text{ :- } \text{dic}(\text{Von}, \text{Nach}).$

$\text{ic}(\text{Von}, \text{Nach}) \text{ :- } \text{dic}(\text{Von}, \text{Z}), \text{dic}(\text{Z}, \text{Nach}).$

REGELN MIT ODER VERBINDEN



REGELN MIT ODER VERBINDEN

- Ein Goal ist ableitbar falls der 1. Regelkopf oder der 2. Regelkopf - nach Instanziierung der Variablen `Von` und `Nach` - ableitbar ist.
- Der 1. Regelkopf ist ableitbar, wenn der 1. Regelrumpf - als Subgoal - mit einem der Fakten unifizierbar ist.
- JA \Rightarrow das Goal ist ableitbar (d.h. die 2. Regel braucht nicht mehr untersucht zu werden).
- NEIN: Durchführe Backtracking und versuche eine Unifizierung des 2. Regelkopfs durch Instanziierung der Variablen `Von` und `Nach` versucht.
- In diesem Fall wird das 1. Prädikat im 2. Regelrumpf zum 1. Subgoal und das 2. Prädikat zum 2. Subgoal.



REGELN MIT ODER VERBINDEN

- Um das 1. Subgoal zu unifizieren, ist eine geeignete Instanziierung der Variablen z vorzunehmen. Gelingt diese Unifizierung, so ist das 2. Subgoal mit dieser Instanziierung von z auf eine mögliche Unifizierung mit einem der Fakten zu überprüfen.
- Gelingt diese Unifizierung nicht, so ist - wegen der logischen UND-Verbindung von 1. Subgoal und 2. Subgoal - nach dem Backtracking im Rumpf der 2. Regel (verbunden mit der Aufhebung der Instanziierung von z) ein erneuter Versuch einer Unifizierung des 1. Subgoals zu versuchen.
- Schlägt dieser Versuch fehl, so ist das 1. Subgoal, somit - wegen der logischen UND-Verbindung von 1. Subgoal und 2. Subgoal - der 2. Regelrumpf, damit der 2. Regelkopf (als Parent-Goal) und folglich auch das Goal nicht aus der Wiba ableitbar.



REGELN MIT ODER VERBINDEN

Ist der nach dem Backtracking unternommene alternative Unifizierungsversuch jedoch erfolgreich, so ist das 2. Subgoal, damit der 2. Regelrumpf, somit der zugehörige unifizierte Regelkopf (als Parent-Goal) und folglich auch das Goal aus der Wiba ableitbar.

