2. Datalog — Syntax und Semantik

Lexikalische Syntax (1)

Syntaktisches Material (Mathematisch):

- $\mathcal{CON}$: Menge der Konstanten
  $\mathcal{CON}$ sollte mindestens 2 Elemente enthalten.
- $\mathcal{PRED}$: Menge der Prädikate
- arity: $\mathcal{PRED} \mapsto \mathbb{N}$: Stelligkeit (Anzahl Argumente/Spalten).
- $\mathcal{VAR}$: Menge der Variablen ($\mathcal{VAR} \cap \mathcal{CON} = \emptyset$).
  Üblicherweise setzt man voraus, daß $\mathcal{VAR}$ unendlich ist.
- $\mathcal{LSYM}$: Logische Zeichen, $\mathcal{LSYM} := \{\leftarrow,\ \wedge,\ (,\ ),\ ,,\ .\}$
  Es muß gelten: $\mathcal{LSYM} \cap (\mathcal{CON} \cup \mathcal{PRED} \cup \mathcal{VAR}) = \emptyset$. Die Variablen werden häufig auch zu den logischen Zeichen gerechnet.

Lexikalische Syntax (2)

Atome in Prolog:

- Kleinbuchstabe (Buchstabe | Ziffer | _)*
  z.B.: diesIstEinAtom, x27, dies_ist_auch_erlaubt.

- ' (beliebige Zeichen)* '
  Falls in der Zeichenfolge ein ' vorkommt, ist es zu verdoppeln (bei manchen Systemen auch \'). \ ist ebenfalls zu verdoppeln.
  'a' und a sind dasselbe Atom.

- (#|$|&|*|+|-|.|/|:|<|=|>|?|@|\|^|~)+
  Aber: "." gefolgt von Leerzeichen markiert Ende der Klausel.
  Weitere Ausnahme: Mit "/*" beginnt ein Kommentar.

- Spezielle Atome: !, ;, [], {}.

Lexikalische Syntax (3)

Prädikate in Prolog:

- Atom/Stelligkeit, z.B. `p/2`.

Konstanten in Prolog:

- Atome
- Integer-Zahlen: z.B. `23`, `-765`, `16'1F` (=31), `0'a` (=97)
  Im Prolog-Standard beginnen statt dessen Hexadezimalzahlen mit `0x`, Oktalzahlen mit `0o` und Binärzahlen mit `0b`. Andere Basen sind nicht vorgesehen (wohl aber `0'` für den ASCII-Code).
- Float-Zahlen: z.B. `-1.23E5` (bei kommerziellen Systemen).
- Strings: z.B. `"abc"`.

Lexikalische Syntax (4)

Variablen in Prolog:

- (Großbuchstabe | _) (Buchstabe | Ziffer | _)*
  Viele Prolog-Systeme geben eine Warnung aus, wenn eine Variable, die nur ein einziges Mal in einer Regel auftritt, nicht mit einem _ beginnt (Schutz vor Tippfehlern).
- Ausnahme: „_“ (anonyme Variable): Bezeichnet bei jedem Auftreten eine neue systemgenerierte Variable.

Kommentare in Prolog:

- Von „%“ bis zum Zeilenende (wie in TEX).
- Von „/*“ bis „*/“ (wie in C).

Lexikalische Syntax (5)

Logische Symbole in Prolog:

- „`:-`“ für $\leftarrow$.
- „`,`“ für $\wedge$.

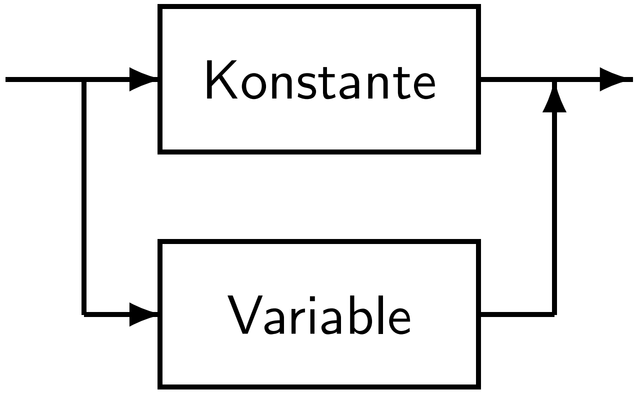
Aufgaben:

- Was ist der Unterschied zwischen `"abc"` und `'abc'`?
  Und was bedeutet das für deduktive Datenbanken (bei denen Fakten auf der Platte abgespeichert sind)?
- Informatieren Sie sich über die lexikalische Syntax eines Prolog-Systems / einer deduktiven DB (z.B. CORAL).

Syntax (1)

Term:

- Ein Term ist eine Konstante oder eine Variable:

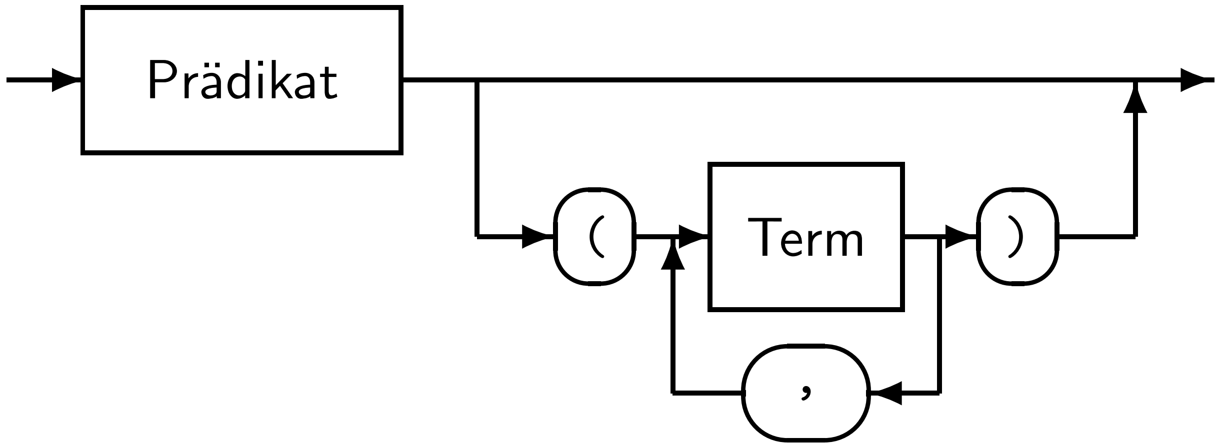


- Die meisten deduktiven DBen erlauben wie Prolog auch strukturierte Terme. Mit Funktionssymbolen, d.h. „Record-Konstruktoren“, z.B. `datum(27,1,97)`. Siehe Kapitel 6.

Syntax (2)

Literal:

- Ein Literal besteht aus einem Prädikat und n Termen, wobei n die Stelligkeit des Prädikates ist:



- Z.B. „es_regnet" (0-stelliges Prädikat) oder „elternteil(julia, X)".
- Viele Systeme lassen auch Infix-Schreibweise etc. zu, z.B. `X > 5`.

Syntax (3)

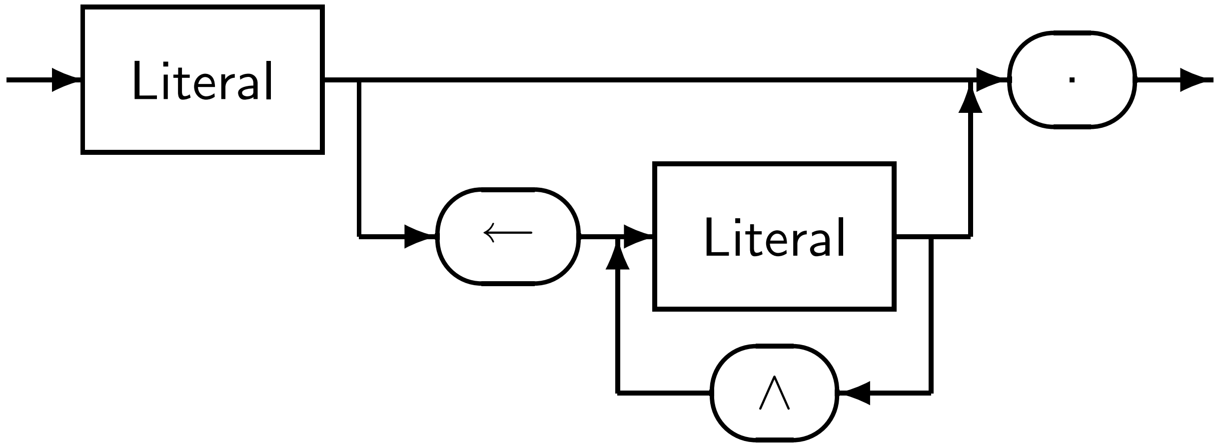
Bemerkungen:

- Die Terme heißen die Argumente des Literals.
- Das Prädikat heißt Prädikat des Literals.
- Ein variablenfreies Literal heißt Grundliteral (oder Fakt).
- Der Begriff „Literal“ wird hier nicht ganz korrekt verwendet.
  Besser: „atomare Formel“, ein Literal könnte auch negiert sein.
- Bisherige deduktive Datenbanken sind wie Prolog ungetypt.
  Bei zukünftigen Systemen ist zu erwarten, daß ein stärkeres Typsystem vorgesehen wird (wie auch in relationalen DBen).
- In Prolog läßt man die explizite Notation der Stelligkeit weg:
  Z.B. ist „`p/2`“ das Prädikat des Literals „`p(a,X)`“.

Syntax (4)

Regel:

- Eine Regel besteht aus einem Literal auf der linken Seite und 0 oder mehr Literalen auf der rechten Seite:



- Z.B.: großelternteil(X, Z) $\leftarrow$ elternteil(X, Y) $\wedge$ elternteil(Y, Z).

Syntax (5)

Begriffe:

- Das Literal auf der linken Seite (die Folgerung, Conclusion) heißt Kopfliteral dieser Regel.
- Die rechte Seite der Regel (die Voraussetzung, Prämisse) heißt Rumpf der Regel.
- Jedes Literal im Rumpf heißt ein Rumpfliteral der Regel.
- Wenn p das Prädikat des Kopfliterals ist, so heißt die Regel eine „Regel über p“ (weil sie zur Definition von p beiträgt).
- Eine Regel heißt Fakt, wenn ihr Rumpf leer ist und sie keine Variablen enthält, z.B. „eltern(emil, arno, birgit).“
  Siehe aber oben: Auch ein variablenfreies Literal heißt Fakt.

Syntax (6)

Mathematisch (abstrakte Syntax):

- $\mathcal{TERM} := \mathcal{CON} \cup \mathcal{VAR}$.
- $\mathcal{LIT} \subseteq \mathcal{PRED} \times \mathcal{TERM}^*$,

  $$\mathcal{LIT} := \big\{(\mathsf{p}, \mathsf{w}) \bigm| \mathsf{p} \in \mathcal{PRED},\ \mathsf{w} \in \mathcal{TERM}^*, \mathsf{arity}(\mathsf{p}) = \mathsf{length}(\mathsf{w})\big\}.$$
- $\mathsf{pred}\big((\mathsf{p}, \mathsf{w})\big) := \mathsf{p}$.

  $\mathsf{args}\big((\mathsf{p}, \mathsf{w})\big) := \mathsf{w}$.
- $\mathcal{RULE} := \mathcal{LIT} \times \mathcal{LIT}^*$.
- $\mathsf{head}\big((\mathsf{A}, \mathcal{B})\big) := \mathsf{A}$.

  $\mathsf{body}\big((\mathsf{A}, \mathcal{B})\big) := \mathcal{B}$.

Syntax (7)

Restriktion (Bereichsbeschränkung):

- Jede Variable des Kopfliterals muß auch im Rumpf vorkommen.
  Dadurch erreicht man, daß die Variablen an einen endlichen Bereich gebunden werden. Sonst könnten für die Variable einfach beliebige Zeichenketten und Zahlen eingesetzt werden.
- Diese Beschränkung ist nicht unbedingt notwendig (z.B. gilt sie nicht für Prolog und ältere Versionen von CORAL). Man kann in diesem Fall „Fakten mit Variablen“ herleiten. Das macht die Implementierung allerdings komplizierter und weniger effizient. Man hat auch nicht mehr die direkte Entsprechung von Faktenmengen zu Relationen. Spätestens mit eingebauten Prädikaten (wie „$<$“) ergeben sich weitere Komplikationen.

Syntax (8)

Logisches Programm:

- Ein logisches Programm ist eine Menge von Regeln.

Bestandteile des logischen Programms:

- Ein Fakt für jedes Tupel der Datenbank.
- Regeln über vordefinierte Sichten.
- Regeln über temporäre Prädikate (nur für diese Anfrage).
  Eines dieser Prädikate, das wir „answer“ nennen wollen, ist zu berechnen. Die Anfrage (Beweisziel) „$?\ \mathsf{A}_1 \wedge \cdots \wedge \mathsf{A}_n$“ stehe für die Regel „$\mathsf{answer}(\mathsf{X}_1, \ldots, \mathsf{X}_n) \leftarrow \mathsf{A}_1 \wedge \cdots \wedge \mathsf{A}_n$“, wobei $\mathsf{X}_1, \ldots, \mathsf{X}_n$ alle in $\mathsf{A}_1 \wedge \cdots \wedge \mathsf{A}_n$ vorkommenden Variablen sind.

Syntax (9)

Aufgaben:

- Skizzieren Sie Datenstrukturen in Pascal/C/C++/Java, um logische Programme zu repräsentieren.
- Datalog basiert auf dem Bereichskalkül. Wie könnte eine Sprache aussehen, die auf dem Tupelkalkül basiert? Oder wie könnte man sonst die nützlichen Attributnamen (statt Argumentpositionen) in Datalog verwenden?
- Was sind die Vor- und Nachteile von Typsystemen?
  Wäre ein Modul-Konzept vorteilhaft?
- Stellen Sie sich vor, Sie wollten eine deduktives DBS entwickeln. Wie würde seine Sprache aussehen?

Interpretationen und Modelle (1)

Bemerkung (Signatur):

- Manchmal sind $\mathcal{CON}$ und $\mathcal{PRED}$ nur Vorräte von Symbolen.
- Die Anwendung verwendet dann nur eine Teilmenge, und nur dieser Teilmenge kann eine Bedeutung zugeordnet werden.
  Z.B. ich kenne keine Person julia. Diese Konstante hat daher keine konkrete Bedeutung. Aber nächstes Jahr wird eine Julia geboren.
- Eine Signatur $\mathsf{SIG} = (\mathsf{CON}, \mathsf{PRED})$ mit $\mathsf{CON} \subseteq \mathcal{CON}$ und $\mathsf{PRED} \subseteq \mathcal{PRED}$ definiert die tatsächlich in der Anwendung verwendeten Symbole.
  Im Relationenmodell redet man nur über Zeichenketten, Zahlen, etc. Dann hat man dieses Problem nicht (feste Signatur). Aber im ER-Modell redet man direkt über Objekte (z.B. Personen).

Interpretationen und Modelle (2)

Allgemeine Interpretation:

Eine SIG-Interpretation $\mathcal{I} = (\mathcal{I}_{\mathsf{dom}}, \mathcal{I}_{\mathsf{con}}, \mathcal{I}_{\mathsf{pred}})$ definiert:

- Eine nichtleere Menge $\mathcal{I}_{\mathsf{dom}}$
  (Objekte des betrachteten Weltausschnitts),
  $\mathcal{I}_{\mathsf{dom}}$ heißt „Universum“, „Domäne“ oder „Individuenbereich“.
- ein Element $\mathcal{I}_{\mathsf{con}}(\mathsf{c}) \in \mathcal{I}_{\mathsf{dom}}$ für jede Konstante $\mathsf{c} \in \mathsf{CON}$,
- eine Relation $\mathcal{I}_{\mathsf{pred}}(\mathsf{p}) \subseteq \mathcal{I}^{n}_{\mathsf{dom}}$ für jedes Prädikat $\mathsf{p} \in \mathsf{PRED}$,
  dabei sei $n := \mathsf{arity}(\mathsf{p})$.
  Die Relation $\mathcal{I}_{\mathsf{pred}}(\mathsf{p})$ heißt auch die Extension von p in $\mathcal{I}$.

Interpretationen und Modelle (3)

Variablenbelegung:

Eine Variablenbelegung zu $\mathcal{I}$ ist eine Abbildung $\alpha\colon \mathcal{VAR} \rightarrow \mathcal{I}_{\mathsf{dom}}$.

Auswertung von Termen:

Sei $\mathcal{I}$ eine Interpretation und α eine Variablenbelegung zu $\mathcal{I}$. Dann schreiben wir eval$(\mathcal{I}, \alpha, \mathsf{t})$ für den Wert eines Terms t:

- eval$(\mathcal{I}, \alpha, \mathsf{t}) := \mathcal{I}_{\mathsf{con}}(\mathsf{t})$, falls $\mathsf{t} \in$ CON (eine Konstante),
- eval$(\mathcal{I}, \alpha, \mathsf{t}) := \alpha(\mathsf{t})$, falls $\mathsf{t} \in \mathcal{VAR}$ (eine Variable).

Interpretationen und Modelle (4)

Wahrheitswert eines Literals:

Sei $\mathcal{I}$ eine Interpretation und α eine Variablenbelegung zu $\mathcal{I}$. Dann schreiben wir truth$(\mathcal{I}, \alpha, \mathsf{A})$ für den Wahrheitswert eines Literals A:

- Sei p := pred(A) und $\mathsf{t}_1 \ldots \mathsf{t}_n$:= args(A).
  Dann sei truth$(\mathcal{I}, \alpha, \mathsf{A})$:= T gdw.

  $$\big(\mathsf{eval}(\mathcal{I}, \alpha, \mathsf{t}_1)\ , \ldots ,\ \mathsf{eval}(\mathcal{I}, \alpha, \mathsf{t}_n)\big) \in \mathcal{I}_{\mathsf{pred}}(\mathsf{p}).$$

  Ansonsten sei truth$(\mathcal{I}, \alpha, \mathsf{A})$:= F.

Interpretationen und Modelle (5)

Wahrheitswert einer Regel:

Außerdem schreiben wir truth$(\mathcal{I}, \alpha, \mathcal{R})$ für den Wahrheitswert einer Regel $\mathcal{R}$:

- Sei A := head$(\mathcal{R})$ und $\mathsf{B}_1 \ldots \mathsf{B}_m$:= body$(\mathcal{R})$, d.h. $\mathcal{R}$ habe die Form $\mathsf{A} \leftarrow \mathsf{B}_1 \wedge \cdots \wedge \mathsf{B}_m$.
  - Falls truth$(\mathcal{I}, \alpha, \mathsf{B}_i) = \mathsf{T}$ für $i = 1, \ldots, m$ und truth$(\mathcal{I}, \alpha, \mathsf{A}) = \mathsf{F}$, so ist truth$(\mathcal{I}, \alpha, \mathcal{R}) := \mathsf{F}$.
  - In allen anderen Fällen ist truth$(\mathcal{I}, \alpha, \mathcal{R}) := \mathsf{T}$.

Ist truth$(\mathcal{I}, \alpha, \mathcal{R}) := \mathsf{T}$, so schreibt man auch $(\mathcal{I}, \alpha) \models \mathcal{R}$ und sagt „$\mathcal{R}$ gilt in $\mathcal{I}$ unter α“. Andernfalls schreibt man $(\mathcal{I}, \alpha) \not\models \mathcal{R}$.

Interpretationen und Modelle (6)

Modell einer Regel:

Ist truth$(\mathcal{I}, \alpha, \mathcal{R}) =$ T für alle Variablenbelegungen α zu $\mathcal{I}$, so sagt man „$\mathcal{I}$ ist Modell von $\mathcal{R}$“ und schreibt $\mathcal{I} \models \mathcal{R}$.

Modell eines logischen Programms:

Sei P ein logisches Programm (in dem nur Symbole aus SIG vorkommen). Gilt $\mathcal{I} \models \mathcal{R}$ für alle Regeln $\mathcal{R} \in$ P, so sagt man „$\mathcal{I}$ ist Modell von P und schreibt $\mathcal{I} \models$ P.

Interpretationen und Modelle (7)

Aufgabe:

Geben Sie zwei möglichst verschiedene Modelle des folgenden logischen Programms P an:

$$\mathsf{p(a).}$$

$$\mathsf{p(b).}$$

$$\mathsf{q(a, b).}$$

$$\mathsf{r(X) \leftarrow p(X) \land q(X, Y).}$$

Es sei dabei SIG $=$ (CON, PRED) mit CON $= \{a, b\}$ und PRED $= \{$p, q, r$\}$. Geben Sie außerdem eine Interpretation an, die nicht Modell von P ist.

Exkurs (1)

Allgemeine Formeln:

- Jedes Literal ist eine Formel (atomare Formel).
- Ist φ eine Formel, so ist auch $(\neg\varphi)$ eine Formel (Negation, logisches „nicht“).
- Sind φ_1 und φ_2 Formeln, so sind auch Formeln:
  - $(\varphi_1 \wedge \varphi_2)$ (Konjunktion, logisches „und“),
  - $(\varphi_1 \vee \varphi_2)$ (Disjunktion, logisches „oder“),
  - $(\varphi_1 \leftarrow \varphi_2)$, $(\varphi_1 \rightarrow \varphi_2)$, $(\varphi_1 \leftrightarrow \varphi_2)$.
- Ist φ eine Formel und X eine Variable, so sind auch Formeln: $(\forall X\colon \varphi)$ („für alle X gilt φ“), $(\exists X\colon \varphi)$ („es gibt ein X mit φ“).

Exkurs (2)

Wahrheitswert allgemeiner Formeln:

- Ist φ ein Literal, so wurde truth$(\mathcal{I}, \alpha, \varphi)$ schon oben definiert.
- Hat φ die Form $(\neg\varphi_1)$, $(\varphi_1 \wedge \varphi_2)$, $(\varphi_1 \vee \varphi_2)$, $(\varphi_1 \leftarrow \varphi_2)$, $(\varphi_1 \rightarrow \varphi_2)$, oder $(\varphi_1 \leftrightarrow \varphi_2)$, so berechnet sich truth$(\mathcal{I}, \alpha, \varphi)$ folgendermaßen aus truth$(\mathcal{I}, \alpha, \varphi_1)$ und truth$(\mathcal{I}, \alpha, \varphi_2)$:

φ_1	φ_2	$(\neg\varphi_1)$	$\wedge$	$\vee$	$\leftarrow$	$\rightarrow$	$\leftrightarrow$
F	F	T	F	F	T	T	T
F	T	T	F	T	F	T	F
T	F	F	F	T	T	F	F
T	T	F	T	T	T	T	T

Exkurs (3)

Wahrheitswert allgemeiner Formeln, Forts.:

- Hat φ die Form $(\forall X: \varphi_1)$, so ist truth$(\mathcal{I}, \alpha, \varphi) :=$ T gdw. truth$(\mathcal{I}, \alpha', \varphi_1) =$ T für alle Variablenbelegungen α', die sich höchstens an der Stelle X von α unterscheiden.

- Hat φ die Form $(\exists X: \varphi_1)$, so ist truth$(\mathcal{I}, \alpha, \varphi) :=$ T gdw. truth$(\mathcal{I}, \alpha', \varphi_1) =$ T für mindestens eine Variablenbelegung α', die sich höchstens an der Stelle X von α unterscheidet.

Exkurs (4)

Weitere Definitionen:

- $\mathcal{I}$ heißt Modell einer Formel φ (geschrieben $\mathcal{I} \models \varphi$) gdw. $\text{truth}(\mathcal{I}, \alpha, \varphi) = \text{T}$ für alle Variablenbelegungen α zu $\mathcal{I}$.
- $\mathcal{I}$ heißt Modell einer Formelmenge Φ (geschrieben $\mathcal{I} \models \Phi$) gdw. $\mathcal{I}$ Modell jeder Formel $\varphi \in \Phi$ ist.
- Eine Formel φ heißt logische Folgerung aus Φ ($\Phi \models \varphi$) gdw. jedes Modell von Φ auch Modell von φ ist.
- Eine Formelmenge Φ heißt konsistent gdw. Φ ein Modell hat.
- Zwei Formeln φ_1 und φ_2 heißen äquivalent ($\varphi_1 \equiv \varphi_2$) gdw. für alle $\mathcal{I}$ und α gilt: $\text{truth}(\mathcal{I}, \alpha, \varphi_1) = \text{truth}(\mathcal{I}, \alpha, \varphi_2)$.

Exkurs (5)

Bemerkung:

Es genügt, Formeln mit den Verknüpfungen (Junktoren) $\neg$ und $\vee$ und dem Quantor $\exists$ einzuführen, und dann folgende Abkürzungen zuzulassen:

- $\varphi_1 \wedge \varphi_2$ für $\neg\big((\neg\varphi_1) \vee (\neg\varphi_2)\big)$,
- $\varphi_1 \leftarrow \varphi_2$ für $\varphi_1 \vee (\neg\varphi_2)$,
- $\varphi_1 \rightarrow \varphi_2$ für $(\neg\varphi_1) \vee \varphi_2$,
- $\varphi_1 \leftrightarrow \varphi_2$ für $\big((\varphi_1 \leftarrow \varphi_2) \wedge (\varphi_1 \rightarrow \varphi_2)\big)$,
- $\forall \mathsf{X}\colon \varphi_1$ für $\neg\big(\exists \mathsf{X}\colon (\neg\varphi_1)\big)$.

Exkurs (6)

Weitere Äquivalenzen (Auswahl):

- $\neg\neg\varphi \;\equiv\; \varphi$
- $\varphi_1 \wedge \varphi_2 \;\equiv\; \varphi_2 \wedge \varphi_1$ (entsprechend für $\vee$)
- $(\varphi_1 \wedge \varphi_2) \wedge \varphi_3 \;\equiv\; \varphi_1 \wedge (\varphi_2 \wedge \varphi_3)$ (entsprechend für $\vee$)
- $\varphi_1 \wedge (\varphi_2 \vee \varphi_3) \;\equiv\; (\varphi_1 \wedge \varphi_2) \vee (\varphi_1 \wedge \varphi_3)$
  (auch mit $\wedge$, $\vee$ vertauscht)
- $\neg(\varphi_1 \wedge \varphi_2) \;\equiv\; (\neg\varphi_1) \vee (\neg\varphi_2)$ (auch mit $\wedge$, $\vee$ vertauscht)
- $\neg(\forall \mathsf{X}\colon \varphi) \;\equiv\; \exists \mathsf{X}\colon (\neg\varphi)$ (auch mit $\forall$, $\exists$ vertauscht)
- $\forall \mathsf{X}\colon (\forall \mathsf{Y}\colon \varphi) \;\equiv\; \forall \mathsf{Y}\colon (\forall \mathsf{X}\colon \varphi)$ (entsprechend für $\exists$)

Exkurs (7)

Beispiel:

Die Regel

$$\text{großvater}(X, Z) \leftarrow \text{vater}(X, Y) \wedge \text{vater}(Y, Z)$$

ist äquivalent zu:

- $\forall\, X, Y, Z\colon\ \text{großvater}(X, Z) \leftarrow \text{vater}(X, Y) \wedge \text{vater}(Y, Z)$
  Alle Variablen sind implizit allquantifiziert.
- $\forall\, X, Y, Z\colon\ \text{großvater}(X, Z) \vee \neg\big(\text{vater}(X, Y) \wedge \text{vater}(Y, Z)\big)$
- $\forall\, X, Y, Z\colon\ \text{großvater}(X, Z) \vee \neg\text{vater}(X, Y) \vee \neg\text{vater}(Y, Z)$
  Diese Form heißt auch Klausel-Form, speziell Hornklausel.
- $\forall\, X, Z\colon\ \text{großvater}(X, Z) \leftarrow \exists Y\colon\ \big(\text{vater}(X, Y) \wedge \text{vater}(Y, Z)\big)$

Herbrand-Interpretationen (1)

Bemerkung:

- Es ist sehr unübersichtlich, beliebige Interpretationen betrachten zu müssen.
- Viel einfacher sind die sogenannten Herbrand-Interpretationen, bei denen $\mathcal{I}_{\mathsf{dom}}$ und $\mathcal{I}_{\mathsf{con}}$ festgelegt sind, und nur noch $\mathcal{I}_{\mathsf{pred}}$ gewählt werden kann.
- Es gilt: Hat eine Menge von Allaussagen (z.B. Formeln ohne Quantoren) überhaupt ein Modell, so hat sie auch ein Herbrand-Modell.
  Für allgemeine Formeln gilt das nicht.

Herbrand-Interpretationen (2)

Definitionen:

Sei ein Program P zu einer Signatur $\mathsf{SIG} = (\mathsf{CON}, \mathsf{PRED})$ gegeben. Falls SIG nicht explizit festgelegt ist, so bestehe sie aus den in P vorkommenden Konstanten und Prädikaten.

- Das Herbrand-Universum $\mathcal{U}_{\mathsf{SIG}}$ sei die Menge CON bzw. $\{\mathsf{a}\}$ falls $\mathsf{CON} = \emptyset$ (mit einer beliebigen Konstante $\mathsf{a} \in \mathcal{CON}$).
- Eine Interpretation $\mathcal{I}$ heißt Herbrand-Interpretation gdw.
  - $\mathcal{I}_{\mathsf{dom}} = \mathcal{U}_{\mathsf{SIG}}$ und
  - $\mathcal{I}_{\mathsf{con}}(\mathsf{c}) = \mathsf{c}$ für alle $\mathsf{c} \in \mathsf{CON}$.
- Die Herbrand-Basis $\mathcal{H}_{\mathsf{SIG}}$ zu SIG ist die Menge aller Grundliterale über SIG.

Herbrand-Interpretationen (3)

Bemerkung:

- Eine Herbrand-Interpretation $\mathcal{I}$ kann mit der Teilmenge

$$\mathsf{H} := \{\mathsf{A} \in \mathcal{H}_{\mathsf{SIG}} \mid \mathcal{I} \models \mathsf{A}\}$$

  der Herbrand-Basis $\mathcal{H}_{\mathsf{SIG}}$ identifiziert werden.

- Ist umgekehrt $\mathsf{H} \subseteq \mathcal{H}_{\mathsf{SIG}}$ gegeben, so sei

$$\mathcal{I}_{\mathsf{pred}}(\mathsf{p}) := \Big\{(\mathsf{c}_1, \ldots, \mathsf{c}_n) \in \mathsf{CON}^n \;\Big|\; \mathsf{p}(\mathsf{c}_1, \ldots, \mathsf{c}_n) \in \mathsf{H}\Big\}$$

  (wobei $n = \mathsf{arity}(\mathsf{p})$).

Herbrand-Interpretationen (4)

Aufgabe:

Geben Sie zwei möglichst verschiedene Herbrand-Modelle des obigen logischen Programms P an:

$$p(a).$$

$$p(b).$$

$$q(a, b).$$

$$r(X) \leftarrow p(X) \wedge q(X, Y).$$

Geben Sie außerdem eine Herbrand-Interpretation an, die nicht Modell von P ist.

Minimales Herbrand-Modell (1)

Bemerkung:

- Ein Modell einer Regelmenge kann „zu groß sein“ (überflüssige Grundliterale enthalten).
- Die Regeln besagen nur: „Wenn der Rumpf wahr ist, muß auch der Kopf wahr sein.“
  Falls der Rumpf nicht wahr ist, ist die Regel automatisch erfüllt. Über den Kopf wird dann keine Aussage gemacht. Das ist wichtig, weil es mehrere Regeln über ein Prädikat geben kann.
- Wir möchten natürlich ein Modell, das nur die Grundliterale enthält, die aufgrund der Regeln gelten müssen.
- Dies ist das minimale Herbrand-Modell.

Minimales Herbrand-Modell (2)

Minimales Herbrand-Modell:

Eine Herbrand-Interpretation H heißt minimales (Herbrand-)Modell eines Programms P gdw.

- H ist Modell von P ($H \models P$),
- es gibt kein kleineres Modell (H' mit $H' \models P$ und $H' \subset H$).

Satz:

- Jedes Programm hat ein minimales Herbrand-Modell.
- Das minimale Herbrand-Modell eines Programms P ist eindeutig bestimmt.

Minimales Herbrand-Modell (3)

Beziehung zu relationalen DBen:

- Der Zustand einer relationalen Datenbank ist eine Interpretation (wie oben definiert).

  Individuenbereich und Interpretation der Konstanten sind dabei vom System vorgegeben (i.w. wie Herbrand-Interpretation).

- Durch das minimale Herbrandmodell ist auch jedem logischen Programmen eine solche Interpretationen zugeordnet.

- Die Extensionen der EDB-Prädikate stammen dabei mit den ursprünglich gegebenen Relationen überein.

- An das minimale Modell könnte man also z.B. auch mit SQL Anfragen stellen.

Minimales Herbrand-Modell (4)

Aufgabe:

- Was ist das minimale Modell $\mathcal{I}_0$ des folgenden Programms?

$$\mathrm{mutter}(\mathrm{arno}, \mathrm{birgit}).$$

$$\mathrm{vater}(\mathrm{birgit}, \mathrm{chris}).$$

$$\mathrm{elternteil}(X, Y) \leftarrow \mathrm{vater}(X, Y).$$

$$\mathrm{elternteil}(X, Y) \leftarrow \mathrm{mutter}(X, Y).$$

$$\mathrm{vorfahr}(X, Y) \leftarrow \mathrm{elternteil}(X, Y).$$

$$\mathrm{vorfahr}(X, Z) \leftarrow \mathrm{elternteil}(X, Y) \wedge \mathrm{vorfahr}(Y, Z).$$

- Geben Sie auch ein nichtminimales Modell an.

Berechnung des minimalen Modells (1)

Bemerkung:

- Man kann das minimale Modell berechnen, indem man iterativ bekannte Fakten für die Rumpfliterale einsetzt, und das entsprechende Fakt für das Kopfliteral herleitet.
- Beispiel:

$$\mathsf{p(a,b)}$$

$$\uparrow$$

$$\boxed{\mathsf{p(a,X)} \leftarrow \mathsf{q(X,Y)} \wedge \mathsf{r(Y,a)}.}$$

$$\uparrow \qquad \uparrow$$

$$\mathsf{q(b,c)} \qquad \mathsf{r(c,a)}$$

Berechnung des minimalen Modells (2)

Substitution:

- Eine Substitution ist eine Abbildung $\theta\colon \mathcal{VAR} \to \mathcal{TERM}$, wobei die Menge $\{\mathsf{X} \in \mathcal{VAR} \mid \theta(\mathsf{X}) \neq \mathsf{X}\}$ endlich ist.
- Eine Substitution wird häufig in der Form $\{\mathsf{X}/\mathsf{a},\ \mathsf{Y}/\mathsf{Z}\}$ notiert (d.h. $\theta(\mathsf{X}) = \mathsf{a}$, $\theta(\mathsf{Y}) = \mathsf{Z}$ und $\theta(\mathsf{V}) = \mathsf{V}$ sonst).
- Eine Substitution heißt Grundsubstitution für eine Regel, wenn sie alle in der Regel vorkommenden Variablen durch Konstanten ersetzt.
- Bei Herbrand-Interpretationen entsprechen Grundsubstitutionen im wesentlichen Variablenbelegungen.

Berechnung des minimalen Modells (3)

Anwendung einer Substitution:

- Man kann den Definitionsbereich von Substitutionen sukzessive auf Terme, Literale und Regeln erweitern, indem man die vorkommenden Variablen ersetzt und den Rest unverändert läßt.
- Z.B. würde die Substitution $\theta = \{\mathsf{X}/\mathsf{a},\ \mathsf{Y}/\mathsf{Z}\}$ angewendet auf das Literal $\mathsf{p}(\mathsf{X}, \mathsf{Y}, \mathsf{V}, \mathsf{b})$ das Literal $\mathsf{p}(\mathsf{a}, \mathsf{Z}, \mathsf{V}, \mathsf{b})$ ergeben.
- Zur Anwendung einer Substitution wird häufig die Postfix-Notation verwendet, z.B. $\mathsf{A}\theta$ statt $\theta(\mathsf{A})$.
- Jede Variable wird nur einmal ersetzt. Z.B. würde bei $\theta = \{\mathsf{X}/\mathsf{Y},\ \mathsf{Y}/\mathsf{Z}\}$ die Variable X durch Y ersetzt, nicht durch Z.

Berechnung des minimalen Modells (4)

Instanzen und Grundinstanzen:

- Eine Regel $\mathcal{R}_1$ heißt Instanz einer Regel $\mathcal{R}_2$ gdw. es eine Substitution θ mit $\mathcal{R}_1 = \mathcal{R}_2\theta$ gibt (Literale entsprechend).
- $\mathcal{R}_1$ heißt Grundinstanz von $\mathcal{R}_2$ gdw. $\mathcal{R}_1$ Instanz von $\mathcal{R}_2$ ist und keine Variablen enthält (d.h. das Ergebnis der Anwendung einer Grundsubstitution ist).
- Statt Grundinstanz sagt man auch „Grundbeispiel". Indem man für die Variablen konkrete Werte einsetzt, bekommt man ein Beispiel der Regel. Zum Beispiel ist

$$\mathsf{elternteil}(\mathsf{arno}, \mathsf{birgit}) \leftarrow \mathsf{mutter}(\mathsf{arno}, \mathsf{birgit})$$

eine Grundinstanz von: $\mathsf{elternteil}(\mathsf{X}, \mathsf{Y}) \leftarrow \mathsf{mutter}(\mathsf{X}, \mathsf{Y})$.

Berechnung des minimalen Modells (5)

Aufgabe:

Gegeben sei die Regel:

$$p(a, X) \leftarrow q(X, Y) \wedge r(Y, a).$$

Welche der folgenden Regeln sind Grundinstanzen dieser Regel?

- $p(a, a) \leftarrow q(a, a) \wedge r(a, a).$
- $p(a, b) \leftarrow q(a, b) \wedge r(b, a).$
- $p(a, b) \leftarrow q(b, c) \wedge r(c, a).$
- $p(b, a) \leftarrow q(a, a) \wedge r(a, a).$
- $p(a, b) \leftarrow q(b, Y) \wedge r(Y, a).$

Berechnung des minimalen Modells (6)

Direkte Konsequenzen ($\mathsf{T_P}$-Operator):

Zu jedem Programm P definieren wir einen Operator $\mathsf{T_P}$, der Herbrand-Interpretationen in Herbrand-Interpretationen abbildet:

$$\mathsf{T_P}(\mathcal{I}) := \Big\{\mathsf{F} \in \mathcal{H}_{\mathcal{SIG}} \;\Big|\; \text{Es gibt eine Regel } \mathsf{A} \leftarrow \mathsf{B}_1 \wedge \cdots \wedge \mathsf{B}_m \text{ in P und eine Grundsubstitution } \theta\text{, so daß}$$

- $\mathsf{B}_i\theta \in \mathcal{I}$ für $i = 1, \ldots, m$, und
- $\mathsf{F} = \mathsf{A}\theta\Big\}$.

$\mathsf{T_P}(\mathcal{I})$ enthält die mittels Regeln aus P und Fakten aus $\mathcal{I}$ direkt herleitbaren Fakten.

Fakten in P werden dabei wie Regeln mit leerem Rumpf behandelt: $\mathsf{A} \leftarrow \mathsf{true}$.

Berechnung des minimalen Modells (7)

Aufgaben:

- Vereinfachen Sie die Definition des $\mathsf{T_P}$-Operators durch Verwendung des Begriffes „Grundinstanz einer Regel“.
- Gegeben sei folgendes Programm P:

$$\begin{aligned}
&\mathsf{p}(\mathsf{a}, \mathsf{b}).\\
&\mathsf{p}(\mathsf{c}, \mathsf{c}).\\
&\mathsf{q}(\mathsf{X}, \mathsf{Y}) \leftarrow \mathsf{p}(\mathsf{X}, \mathsf{Y}).\\
&\mathsf{q}(\mathsf{Y}, \mathsf{X}) \leftarrow \mathsf{p}(\mathsf{X}, \mathsf{Y}).
\end{aligned}$$

  Es sei $\mathcal{I}_0 := \emptyset$. Berechnen Sie: $\mathcal{I}_1 := \mathsf{T_P}(\mathcal{I}_0)$, $\mathcal{I}_2 := \mathsf{T_P}(\mathcal{I}_1)$ und $\mathcal{I}_3 := \mathsf{T_P}(\mathcal{I}_2)$.

Berechnung des minimalen Modells (8)

Satz:

Sei P ein beliebiges logisches Programm und sei

- $\mathcal{I}_0 := \emptyset$,
- $\mathcal{I}_{i+1} := \mathsf{T}_\mathsf{P}(\mathcal{I}_i)$ für $i = 0, 1, \ldots$,
- $n \in \mathbb{N}_0$ mit $\mathcal{I}_{n+1} = \mathcal{I}_n$.
  (Falls P endlich ist, gibt es tatsächlich so ein n.)

Dann ist $\mathcal{I}_n$ das minimale Herbrand-Modell von P.

Berechnung des minimalen Modells (9)

Aufgaben:

- Berechnen Sie das minimale Modell des folgenden logischen Programms durch iterative Anwendung des T_P-Operators:

$$\mathrm{mutter}(\mathrm{arno}, \mathrm{birgit}).$$

$$\mathrm{vater}(\mathrm{birgit}, \mathrm{chris}).$$

$$\mathrm{elternteil}(X, Y) \leftarrow \mathrm{vater}(X, Y).$$

$$\mathrm{elternteil}(X, Y) \leftarrow \mathrm{mutter}(X, Y).$$

$$\mathrm{vorfahr}(X, Y) \leftarrow \mathrm{elternteil}(X, Y).$$

$$\mathrm{vorfahr}(X, Z) \leftarrow \mathrm{elternteil}(X, Y) \wedge \mathrm{vorfahr}(Y, Z).$$

- Gilt $\mathcal{I} \subseteq T_P(\mathcal{I})$ für beliebige $\mathcal{I}$?

Exkurs: Etwas Verbandstheorie (1)

Partielle Ordnung:

Gegeben sei eine Menge $\mathcal{M}$ mit einer Relation $\preceq\ \subseteq\ \mathcal{M} \times \mathcal{M}$.

- $\preceq$ heißt partielle Ordnung gdw. für alle $\mathcal{I}, \mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3 \in \mathcal{M}$:
  - reflexiv: $\mathcal{I} \preceq \mathcal{I}$
  - transitiv: $\mathcal{I}_1 \preceq \mathcal{I}_2,\ \mathcal{I}_2 \preceq \mathcal{I}_3 \implies \mathcal{I}_1 \preceq \mathcal{I}_3$
  - antisymmetrisch: $\mathcal{I}_1 \preceq \mathcal{I}_2,\ \mathcal{I}_2 \preceq \mathcal{I}_1 \implies \mathcal{I}_1 = \mathcal{I}_2$.
- $(\mathcal{M}, \preceq)$ heißt dann partiell geordnete Menge.

Exkurs: Etwas Verbandstheorie (2)

Kleinste obere Schranke:

Sei $(\mathcal{M}, \preceq)$ partiell geordnete Menge.

- Ein Element $\mathcal{I} \in \mathcal{M}$ heißt obere Schranke von $\mathcal{N} \subseteq \mathcal{M}$ gdw.
  - $\mathcal{J} \preceq \mathcal{I}$ für alle $\mathcal{J} \in \mathcal{N}$.
- Ein Element $\mathcal{I} \in \mathcal{M}$ heißt kleinste obere Schranke („least upper bound“) von $\mathcal{N} \subseteq \mathcal{M}$ gdw.
  - $\mathcal{I}$ ist obere Schranke von $\mathcal{N}$,
  - $\mathcal{I} \preceq \mathcal{I}'$ für alle $\mathcal{I}' \in \mathcal{M}$, die auch obere Schranke von $\mathcal{N}$ sind.
- Entsprechend: untere Schranke, größte untere Schranke.

Exkurs: Etwas Verbandstheorie (3)

Lemma:

Falls die kleinste obere Schranke existiert, so ist sie eindeutig bestimmt. (Entsprechend: größte untere Schranke.)

Vollständiger Verband:

Eine partiell geordnete Menge $(\mathcal{M}, \preceq)$ heißt vollständiger Verband gdw.

- für alle $\mathcal{N} \subseteq \mathcal{M}$ existieren kleinste obere und größte untere Schranke.

Man schreibt dann lub($\mathcal{N}$) für die kleinste obere Schranke von $\mathcal{N}$ und glb($\mathcal{N}$) für die größte untere Schranke.

Exkurs: Etwas Verbandstheorie (4)

Kleinstes Element:

In einem vollständigen Verband gibt es immer ein kleinstes Element $\bot := \text{glb}(\mathcal{M})$ und ein größtes Element $\top := \text{lub}(\mathcal{M})$.

Beispiel:

- Die Menge aller Herbrand-Interpretationen (über einer festen Signatur) zusammen mit $\subseteq$ ist ein vollständiger Verband.

  Teilmengen-Verband: Menge aller Teilmengen von $\mathcal{H}_{\mathcal{SIG}}$.

- Es gilt dann:

$$\text{lub}(\mathcal{N}) = \bigcup_{\mathcal{I} \in \mathcal{N}} \mathcal{I}, \quad \text{glb}(\mathcal{N}) = \bigcap_{\mathcal{I} \in \mathcal{N}} \mathcal{I}, \quad \bot = \emptyset, \quad \top = \mathcal{H}_{\mathcal{SIG}}.$$

Exkurs: Etwas Verbandstheorie (5)

Eigenschaften von Abbildungen $\mathsf{T}\colon \mathcal{M} \to \mathcal{M}$:

- T heißt monoton gdw. $\mathsf{T}(\mathcal{I}_1) \preceq \mathsf{T}(\mathcal{I}_2)$ für alle $\mathcal{I}_1 \preceq \mathcal{I}_2$.
- T heißt stetig gdw. $\mathsf{T}\bigl(\mathsf{lub}(\mathcal{N})\bigr) = \mathsf{lub}\bigl(\mathsf{T}(\mathcal{N})\bigr)$ für alle $\mathcal{N} \subseteq \mathcal{M}$ derart, daß alle endlichen Teilmengen von $\mathcal{N}$ eine obere Schranke in $\mathcal{N}$ haben.
  $\mathsf{T}(\mathcal{N})$ bezeichne dabei $\bigl(\{\mathsf{T}(\mathcal{I}) \mid \mathcal{I} \in \mathcal{N}\}\bigr)$.

Lemma:

Ist T stetig, so ist T auch monoton.

Exkurs: Etwas Verbandstheorie (6)

Fixpunkte:

- $\mathcal{I} \in \mathcal{M}$ heißt Fixpunkt von T gdw. $\mathsf{T}(\mathcal{I}) = \mathcal{I}$.
- $\mathcal{I} \in \mathcal{M}$ heißt kleinster Fixpunkt von T gdw. $\mathsf{T}(\mathcal{I}) = \mathcal{I}$ und $\mathcal{I} \preceq \mathcal{J}$ für alle $\mathcal{J} \in \mathcal{M}$ mit $\mathsf{T}(\mathcal{J}) = \mathcal{J}$.

Satz:

Eine monotone Abbildung T in einem vollständigen Verband hat einen kleinsten Fixpunkt, und zwar:

$$\mathsf{glb}\big(\{\mathcal{I} \in \mathcal{M} \mid \mathsf{T}(\mathcal{I}) \preceq \mathcal{I}\}\big).$$

Für den kleinsten Fixpunkt von T schreiben wir lfp(T).

Exkurs: Etwas Verbandstheorie (7)

Lemma:

- Der Operator $\mathsf{T_P}$ ist monoton und sogar stetig.
- $\mathcal{I}$ ist Modell von P gdw. $\mathsf{T_P}(\mathcal{I}) \subseteq \mathcal{I}$.

Satz:

Der kleinste Fixpunkt von $\mathsf{T_P}$ ist das minimale Modell von P.

Exkurs: Etwas Verbandstheorie (8)

Potenzen einer Abbildung:

- $\mathsf{T} \uparrow 0 := \bot$.
- $\mathsf{T} \uparrow (n+1) := \mathsf{T}\big(\mathsf{T} \uparrow n\big)$.
- $\mathsf{T} \uparrow \omega := \mathsf{lub}\big(\{\mathsf{T} \uparrow n \mid n \in \mathbb{N}_0\}\big)$.

Bemerkungen:

- Falls es ein $n \in \mathbb{N}$ mit $\mathsf{T} \uparrow (n+1) = \mathsf{T} \uparrow n$ gibt, gilt natürlich $\mathsf{T} \uparrow m = \mathsf{T} \uparrow n$ für alle $m \geq n$ und damit auch $\mathsf{T} \uparrow \omega = \mathsf{T} \uparrow n$.
- Man kann $\mathsf{T} \uparrow \gamma$ allgemein für beliebige Ordinalzahlen definieren: $\mathsf{T} \uparrow \gamma := \mathsf{T}(\mathsf{T} \uparrow \gamma - 1)$ falls falls γ Nachfolger von $\gamma - 1$ und $\mathsf{T} \uparrow \gamma := \mathsf{lub}\big(\{\mathsf{T} \uparrow \beta \mid \beta < \gamma\}\big)$ sonst.

Exkurs: Etwas Verbandstheorie (9)

Lemma:

Ist T monoton, so gilt für alle $i \in \mathbb{N}_0$: $\mathrm{T} \uparrow i \ \preceq \ \mathrm{T} \uparrow (i+1)$.

Satz:

Ist T stetig, so gilt: $\mathrm{lfp}(\mathrm{T}) = \mathrm{T} \uparrow \omega$.

Bemerkungen:

- Daraus folgt der Satz auf Folie 2-45 (Berechnung des minimalen Modells).
- Ist T nur monoton, so gibt es immerhin eine Ordinalzahl γ mit $\mathrm{lfp}(\mathrm{T}) = \mathrm{T} \uparrow \gamma$.

Kausale Modelle, Vervollständigung (1)

Kausales Modell:

Ein Herbrandmodell $\mathcal{I}$ eines logischen Programms P heißt kausal gdw. $\mathsf{T}_\mathsf{P}(\mathcal{I}) = \mathcal{I}$ (d.h. es ist ein Fixpunkt von T_P).

Bemerkung:

Für jedes in einem kausalen Modell $\mathcal{I}$ gültige Fakt A gibt es also einen Grund in Gestalt einer Regel-Grundinstanz

$$\mathsf{A} \leftarrow \mathsf{B}_1 \wedge \cdots \wedge \mathsf{B}_m,$$

die A herzuleiten gestattet (weil die B_i auch in $\mathcal{I}$ enthalten sind).

Kausale Modelle, Vervollständigung (2)

Korollar:

Das minimale Modell ist kausal.

Bemerkungen:

- Die Umkehrung gilt nicht. Zum Beispiel ist $\mathcal{I} = \{\mathsf{p}\}$ ein kausales Modell von $\mathsf{P} = \{\mathsf{p} \leftarrow \mathsf{p}\}$, aber nicht das minimale Modell ($\mathcal{I}_0 = \emptyset$).
  Praktisches Beispiel: $\mathsf{verheiratet}(\mathsf{X}, \mathsf{Y}) \leftarrow \mathsf{verheiratet}(\mathsf{Y}, \mathsf{X})$.
- Für nicht-rekursive Programme (s.u.) gilt aber auch die Umkehrung: Sie haben nur ein kausales Modell, nämlich das minimale Modell.

Kausale Modelle, Vervollständigung (3)

Clark's Vervollständigung:

- Kausale Modelle kann man syntaktisch charakterisieren: Clark's Vervollständigung comp(P) ist eine Formelmenge, deren Modelle genau die kausalen Modelle von P sind.
- comp(P) hat allerdings nicht nur Herbrand-Modelle. Die „freie“ Interpretation von Konstanten (und Funktionssymbolen) wird durch Axiome über die Gleichheit erzwungen. Der Individuenbereich kann aber unbenannte Objekte enthalten.
- Grundlegende Idee: Fasse „$\leftarrow$“ als „$\leftrightarrow$“ auf!

Kausale Modelle, Vervollständigung (4)

Clark's Vervollständigung, Beispiele:

- $\mathsf{elternteil}(X, Y) \leftarrow \mathsf{vater}(X, Y).$
  $\mathsf{elternteil}(X, Y) \leftarrow \mathsf{mutter}(X, Y).$

  $\mathsf{elternteil}(X, Y) \leftrightarrow \mathsf{vater}(X, Y) \vee \mathsf{mutter}(X, Y).$

- $\mathsf{großvater}(X, Z) \leftarrow \mathsf{elternteil}(X, Y) \wedge \mathsf{vater}(Y, Z).$

  $\mathsf{großvater}(X, Z) \leftrightarrow \exists Y\colon \big(\mathsf{elternteil}(X, Y) \wedge \mathsf{vater}(Y, Z)\big).$

- $\mathsf{verheiratet}(X, Y) \leftarrow \mathsf{ehepaar}(X, Y).$
  $\mathsf{verheiratet}(X, Y) \leftarrow \mathsf{verheiratet}(Y, X).$

  $\mathsf{verheiratet}(X, Y) \leftrightarrow \mathsf{ehepaar}(X, Y) \vee \mathsf{verheiratet}(Y, X).$

Kausale Modelle, Vervollständigung (5)

Clark's Vervollständigung, Beispiele (Forts.):

- $\mathrm{ehepaar}(\mathrm{arno}, \mathrm{birgit}).$

  $\mathrm{ehepaar}(\mathrm{chris}, \mathrm{doris}).$

  $$\begin{aligned}\mathrm{ehepaar}(X, Y) \;\leftrightarrow\; & \;(X = \mathrm{arno} \land Y = \mathrm{birgit}) \\ & \lor (X = \mathrm{chris} \land Y = \mathrm{doris}).\end{aligned}$$

- $\mathrm{vorfahr}(X, Y) \leftarrow \mathrm{elternteil}(X, Y).$

  $\mathrm{vorfahr}(X, Z) \leftarrow \mathrm{elternteil}(X, Y) \land \mathrm{vorfahr}(Y, Z).$

  (zur Übung)

  Es gibt keine Formel der Logik erster Stufe, die die transitive Hülle für beliebige endliche Relation definiert. Finden Sie ein Beispiel, bei dem die Vervollständigung nicht funktioniert.

Closed World Assumption (1)

Satz:

Im minimalen Herbrand-Modell von P sind genau die Fakten enthalten, die logische Folgerungen aus P sind.

Closed World Assumption:

$\mathrm{CWA}(\mathsf{P}) = \mathsf{P} \cup \{\neg \mathsf{F} \mid \mathsf{F} \in \mathcal{H}_{\mathcal{SIG}},\ \mathsf{P} \not\models \mathsf{F}\}$.

Bemerkungen:

- Die CWA ist also praktisch äquivalent zur Annahme, daß das minimale Herbrand-Modell gemeint ist.
- Aber es werden nicht unbedingt nur Herbrand-Interpretationen betrachtet, z.B. $\mathsf{p}(\mathsf{X}) \leftrightarrow \mathsf{X} = \mathsf{a}$ folgt nicht aus $\mathrm{CWA}\big(\{\mathsf{p}(a)\}\big)$.

Closed World Assumption (2)

Gegenteil (Open World Assumption):

- Man gibt Formeln an, die in der zu beschreibenden Mini-Welt gelten. Wenn einem nichts mehr einfällt, hört man auf.
- Die Behandlung solch unvollständigen Wissens ist ein ziemlich schwieriges Problem für die künstliche Intelligenz.

Hier dagegen:

- Das spezifizierte Wissen ist vollständig.
- Alles, was aus den Formeln nicht folgt, ist falsch.
- Einfacher: Nur positives Wissen muß formuliert werden.

Annahmen über die Gleichheit (1)

Domain Closure Axiom (DCA):

- Das Universum enthält nur die über Konstanten bezeichneten Objekte.
- $\forall X\colon (X = c_1 \vee X = c_2 \vee \cdots \vee X = c_n)$.
- Praktisch sinnvoll:
  Antworten müssen ausgedruckt werden.
- Allerdings kann es z.B. vorkommen, daß offene EA-Ströme zwar an Variablen gebunden werden können, aber nicht durch konstante Terme bezeichnet werden können (Verletzung des DCA).
- Theoretisch schwierig:
  Geht nur bei endlich vielen Konstanten.

Annahmen über die Gleichheit (2)

Unique Names Axiom (UNA):

- Verschiedene Konstanten bezeichnen verschiedene Objekte.
- $\mathsf{c}_i \neq \mathsf{c}_j$ für $i \neq j$.
- Wird in der Praxis verletzt, z.B. $01 = +1 = 1$.
  Aber der Benutzer kennt diese Äquivalenzen. Es reicht deswegen, in der Antwortmenge einen Repräsentanten aus jeder Äquivalenzklasse auszudrucken.
- Praktisch nützlich: Ein zielgerichteter Beweiser würde nicht $\mathsf{p}(a) \leftarrow \ldots$ verwenden, um $\mathsf{p}(\mathsf{b})$ herzuleiten.
- Nullwerte (unbekannte Werte) könnte man als Konstanten behandeln, die das UNA verletzen. (Ebenso: Skolem-Konstanten.)