Intelligent Systems on the World Wide Web

SiLRI – simple logic-based RDF interpreter

Lecture Slides
Steffen Staab
Institute for Applied Computer Science and Formal Description Methods (AIFB)
Karlsruhe University

Ziel: Verteile Wissen auf dem Web

⇒ Wie repräsentiere ich Wissen?

⇒ Logik! (im wesentlichen Wiederholung aus dem Grundstudium)

Quelle: Schöning “Logik für Informatiker”, BI Wissenschaftsverlag

Logik

Beispiele:

1. Definition einer Syntax — SokratesIstSterblich ∧ SokratesIstMensch; $A \wedge B$; Klaus ∧ Frieda
2. Definition einer Semantik — (I(A) = 1 und I(B) = 1) gdw. $I(A \wedge B)=1$
3. Modell für eine Formel — $A \wedge B$ ist erfüllbar, verheiratet ∧ ¬verheiratet ist nicht erfüllbar
4. Semantische Ableitbarkeit — $A \wedge B \models A$, da für alle I aus $I(A \wedge B)=1$ folgt $I(A)=1$; $A \leftarrow B \models \neg A \vee B$
5. Syntaktisches Kalkül — $(A \rightarrow B) \wedge A \vdash B$ Modus Ponens

Vom Zeichen zur Bedeutung

Caveat!
“verheiratet”, “Klaus”, “grün”, “Klaus ist grün”
sind zunächst nur bedeutungslose Zeichen

erst durch Einbettung in eine logisch formalisierte Ontologie erhalten die Zeichen eine Bedeutung

© S. Staab, 2001

AIFB

1. Syntax der Aussagenlogik

a.) Eine atomare Formel hat die Form A_i, $A_i \in \mathcal{A}$

b.) alle atomaren Formeln sind Formeln

c.) Für alle Formeln F und G gilt:
$\neg F$, $(F \wedge G)$ und $(F \vee G)$ sind Formeln

Beispiele: $A \wedge B$, $A \wedge (B \vee C)$
EsRegnet $\vee$ (Schlossgarten $\wedge$ Sonne)

© S. Staab, 2001

AIFB

2. Definition einer Semantik der Aussagenlogik (I)

{0, 1} heißen Wahrheitswerte

Eine Belegung ist eine Funktion: $I : E \rightarrow \{0, 1\}$,

E ist die Menge aller Formeln, für die gilt:

$$I(F \wedge G) = \begin{cases} 1 & \text{falls } I(F) = 1 \text{ und } I(G) = 1 \\ 0, & \text{sonst} \end{cases}$$

$$I(F \vee G) = \begin{cases} 1 & \text{falls } I(F) = 1 \text{ oder } I(G) = 1 \text{ (oder beide)} \\ 0, & \text{sonst} \end{cases}$$

$$I(\neg F) \quad = \begin{cases} 1 & \text{falls } I(F) = 0 \\ 0, & \text{sonst} \end{cases}$$

© S. Staab, 2001

AIFB

Definition einer Semantik der Aussagenlogik (II)

Beispiel:

Falls $I(A) = 1$ und $I(B) = 0$, dann

$I(A \vee (\neg A \wedge B)) = 1$

$I(\neg A \vee B) = 0$

© S. Staab, 2001

AIFB

3. Modell für eine Formel (I)

Definition:

Sei F eine Formel und I eine Belegung.
Falls $I(F) = 1$ gilt, so schreiben wir $I \models F$,
d.h. „I ist ein Modell für F".

Falls $I(F) = 0$ gilt, so schreiben wir $I \not\models F$,
d.h. „I ist kein Modell für F".

© S. Staab, 2001

AIFB

Modell für eine Formel (II)

Definition:

Eine Formel F heißt gültig (oder Tautologie), falls jede zu F passende Belegung ein Modell für F ist, d.h. „|= F“.

Tautologien	erfüllbare, aber nicht gültige Formeln		unerfüllbare Formeln
$\neg G$	F	$\neg F$	G

© S. Staab, 2001

AIFB

Modell für eine Formel (III)

Beispiel:

„Wenn ich 100 m in < 9 Sek. laufe, werde ich Weltmeister. Da ich die 100 m nicht unter 9 Sek. laufe, werde ich nicht Weltmeister.“

Lösung:

$F = \neg\, schnell \vee wm$ $\Rightarrow$ $I_1(schnell) = 1,\ I_1(wm) = 1,\ I_1 \models F$

$G = \neg schnell \wedge \neg wm$ $\Rightarrow$ $I_2(schnell) = 1,\ I_2(wm) = 0,\ I_2 \text{ nicht}\models F$

$I_3(schnell) = 0,\ I_3(wm) = 0,\ I_3 \models F$

$I_1(G) = 0$

$\Rightarrow$ d.h. Es gibt ein Modell, das F erfüllt aber nicht G.

© S. Staab, 2001

AIFB

Modell für eine Formel (IV)

Definition:

Eine Formel G heißt eine Folgerung der Formel F, falls jede Belegung, die F erfüllt, auch ein Modell für G ist, wir schreiben (vereinfachend) F |= G.

Verallgemeinerung:

$F_1, \ldots, F_n \models G_1, \ldots, G_m$

Problem:

Test auf semantische Folgerbarkeit ist extrem teuer!
(Exponentiell in der Länge/Anzahl der Formeln)

© S. Staab, 2001

AIFB

Äquivalenzen (I)

$F \wedge F \equiv F$
$F \vee F \equiv F$ Idempotenz

$F \wedge G \equiv G \wedge F$
$F \vee G \equiv G \vee F$ Kommutativität

$(F \wedge G) \wedge H \equiv F \wedge (G \wedge H)$
$(F \vee G) \vee H \equiv F \vee (G \vee H)$ Assoziativität

$F \wedge (G \vee H) \equiv ((F \wedge G) \vee (F \wedge H))$
$F \vee (G \wedge H) \equiv ((F \vee G) \wedge (F \vee H))$ Distributivität

$\neg\neg F \equiv F$

© S. Staab, 2001

Äquivalenzen (II)

$\neg(F \wedge G) \equiv (\neg F \vee \neg G)$
$\neg(F \vee G) \equiv (\neg F \wedge \neg G)$
de Morgan´ sche Regeln

$F \vee G \equiv F$,falls F Tautologie
$F \wedge G \equiv G$,falls F Tautologie

$F \vee G \equiv G$,falls F unerfüllbar
$F \wedge G \equiv F$,falls F unerfüllbar

Äquivalenzregeln erlauben die Umformung von Formeln so, dass die gültigen Belegungen unverändert bleiben.

© S. Staab, 2001

Normalformen (I)

Definition:

Ein Literal ist eine atomare Formel oder die Negation einer atomaren Formel. (Ersteres ist ein positives, zweiteres ein negatives Literal)

Definition:

Eine Formel ist in konjunktiver Normalform (KNF), falls sie eine Konjunktion von Disjunktionen von Literalen ist:

$$F = \left(\bigwedge_{i=1}^{n_i} \left(\bigvee_{j=1}^{m_j} L_{i,j} \right) \right)$$

$L_{i,j} \in \{A_1, A_2, ...\} \cup \{\neg A_1, \neg A_2, ...\}$

DNF : $\wedge$ mit $\vee$ vertauscht.

© S. Staab, 2001

Normalformen (II)

Satz:

Zu jeder Formel F gibt es eine äquivalente Formel G in KNF (bzw. auch eine in DNF).

© S. Staab, 2001

Hornlogik

Definition:

Eine Formel heißt Hornformel, falls F in KNF und jedes Disjunktionsglied in F höchstens ein positives Literal enthält.

Beispiel:

a.) $(A \vee \neg B) \wedge (\neg C \vee D \vee \neg E) \wedge F$ ist Horn

b.) $(A \vee \neg C) \wedge (D \vee E \vee \neg F) \wedge G$ ist nicht Horn

Äquivalente Schreibweisen zu a.):

$(A \leftarrow B) \wedge (D \leftarrow (C \wedge E)) \wedge F.$

$\{\{A, \neg B\},\{\neg C, D, \neg E\},\{F\}\}$

© S. Staab, 2001

AIFB

Resolution (I)

Resolution ist ein syntaktisches Verfahren, um die Erfüllbarkeit einer Formelmenge $\{F_1, ..., F_m\}$ zu überprüfen.

Verfahren:

$\{A_1, \neg A_2, \neg A_3\}$ $\{A_3, \neg A_2, \neg A_4\}$ $\{A_1, A_4\}$

$\{A_1, \neg A_2, \neg A_4\}$

$\{A_1, \neg A_2\}$

$\{A_1\}$ $\{\neg A_1\}$

$\square$

© S. Staab, 2001

AIFB

Resolution (II)

Definition:

$Res(F) = F \cup \{R |$ R ist Resolvent zweier Klauseln in F$\}$

$Res^0(F) = F$

$Res^{n+1}(F) = Res(Res^n(F))$ für $n \geq 0$

$Res^*(F) = \bigcup_{n \geq 0} Res^n(F)$

Resolutionssatz der Aussagenlogik:

Eine Klauselmenge F ist unerfüllbar gdw.

$\in Res^*(F)$

© S. Staab, 2001

AIFB

Resolution (III)

Test, ob G von F impliziert wird:

Ja, falls $Res^*(F \wedge \neg G)$ enthält

Nein, sonst.

Resolution i.A.:

Exponentieller Zeitaufwand

(Potenzmenge der Klauseln in F!)

© S. Staab, 2001

AIFB

Resolution (IV)

Resolution von Hornformeln:

Satz:

Resolution von Hornklauseln lässt sich auf die Fälle beschränken, in denen mindestens einer der beiden Resolventen die Länge 1 besitzt, dennoch bleibt das Resolutionsverfahren vollständig (i.Ggs. zum allg. Fall)

Jeder Resolutionsschritt führt also nur zu kürzeren Klauseln; Man benötigt nur O(n) Resolutionsschritte, um Erfüllbarkeit zu entscheiden.

Resolution - Anfragen

Regeln:

Regen :- TiefÜberBaden ∧ Warm.
Schnee :- TiefÜberBaden ∧ Kalt.
Lieblingskneipe :- Regen ∧ Feierabend.

Fakten:

Warm.
TiefÜberBaden.

Anfrage:

:- Lieblingskneipe.

"Regen", "Schnee", "Lieblingskneipe" etc. haben zunächst keine Bedeutung.

Aber, wenn ich mich darauf einige, daß aus "Regen" und "Feierabend" die "Lieblingskneipe" folgt (und auf die anderen Regeln), dann folgt aus den Fakten unweigerlich die "Lieblingskneipe".

"Übersetzung in KNF"

Regen :- TiefÜberBaden ∧ Warm.	{{Regen, ¬TiefÜberBaden, ¬ Warm},
Schnee :- TiefÜberBaden ∧ Kalt.	{Schnee, ¬ TiefÜberBaden, ¬Kalt},
Lieblingskneipe :- Regen ∧ Feierabend.	{Lieblingskneipe, ¬Regen, ¬Feierabend},
Warm.	{Warm},
TiefÜberBaden.	{TiefÜberBaden},
:- Lieblingskneipe.	{¬Lieblingskneipe}}

Resolution

{Regen, ¬TiefÜberBaden, ¬ Warm},

{Schnee, ¬ TiefÜberBaden, ¬Kalt},

{Lieblingskneipe, ¬Regen, ¬Feierabend},

{Warm},
{TiefÜberBaden},
{Feierabend},

{¬Lieblingskneipe}

{¬Regen, ¬Feierabend}

{¬Regen}

{¬TiefÜberBaden, ¬Warm}

{ ¬Warm}

□

Yes, heute in meine Lieblingskneipe!

Prädikatenlogik

Was ist mit

"Bill Clinton kennt Gorbatschow."
"Bill ist verheiratet mit Hillary."
"Bill ist Präsident."

Wie beschreiben wir

Relationen zwischen Objekten?

(NB: "TiefÜberBaden" ist geschummelt)

© S. Staab, 2001

AIFB

Syntax der Prädikatenlogik

Eine Variable hat die Form x_i mit i=1,2,3...

Ein Prädikatensymbol/Funktionssymbol hat die Form P^k_i / f^k_i.
Hierbei heißt i der Unterscheidungsindex und k die Stelligkeit.

A. Jede Variable ist ein Term.

B. Falls f ein Funktionssymbol der Stelligkeit k ist, und falls $t_1,\ldots,t_k$ Terme sind, dann ist $f(t_1,\ldots,t_k)$ ein Term (k=0 erlaubt!).

Wir definieren nun:

1. Falls P ein Prädikatensymbol der Stelligkeit k ist, und falls $t_1,\ldots,t_k$ Terme sind, dann ist $P(t_1,\ldots,t_k)$ eine Formel.
2. Für jede Formel F ist auch $\neg F$ eine Formel
3. Für alle Formeln F und G sind auch $(F \wedge G)$ und $(F \vee G)$ Formeln
4. Falls x eine Variable ist und F eine Formel, dann sind auch $\forall xF$ und $\exists xF$ Formeln

© S. Staab, 2001

AIFB

Bezeichnungen

Atomare Formeln sind gemäß 1. aufgebaut

Falls F eine Formel ist und F als Teil einer Formel G auftritt, heißt F Teilformel von G.

Variablen in Formeln sind frei oder gebunden. Das Vorkommen einer Variablen x in F heißt gebunden, wenn x nur in Teilformeln von F mit der Form $\forall xG$ oder $\exists xG$, vorkommt:

$\forall$x Mensch(x) -> sterblich(x).	x ist gebunden
Mensch(x) -> $\forall$y sterblich(x).	x ist frei.
Mensch(x) -> $\forall$x sterblich(x).	x ist einmal gebunden und einmal frei

© S. Staab, 2001

AIFB

Beispiel

$\forall$x Mensch(x) -> sterblich(x).

Mensch(Bill).

Kind(Ehe(Bill,Hillary),Chelsea).

Variablen: x; gebundene Variablen: x

Funktionssymbole: Bill, Hillary, Chelsea, Ehe

Terme: x, Bill, Hillary, Chelsea, Ehe(Bill, Hillary)

Prädikatssymbole: Mensch, sterblich, Kind

© S. Staab, 2001

AIFB

Semantik der Prädikatenlogik

Eine Struktur ist ein Paar $A=(U_A,I_A)$, wobei U_A eine beliebige nichtleere Menge ist (das Universum von A). I_A ist eine Abbildung (Interpretation), die

- jedem k-stelligen Prädikatensymbol P ein k-stelliges Prädikat über U_A zuordnet,
- jedem k-stelligen Funktionssymbol f eine k-stellige Funktion über U_A zuordnet,
- jeder Variablen x ein Element des Universums zuordnet.

© S. Staab, 2001

AIFB

Beispiel

∀x Mensch(x) -> sterblich(x).
Mensch(Bill).
Kind(Ehe(Bill,Hillary),Chelsea).

Mögliche Interpretation:

A=({„bill“, „hillary“, „chelsea“, „ehe1“,...},I).

Mensch^A={„bill“, „hillary“, „chelsea“, „Helmut Kohl“,...}
sterblich^A={„bill“, „hillary“, „chelsea“, „Helmut Kohl“,...}
Kind^A={(„ehe1“, „chelsea“),(„ehe2“, „Helmut Kohl“)}

Ehe^A={(„bill“, „hillary“, „ehe1“), („...“, „...“, „ehe2“),...}
Bill^A=„bill“, Hillary^A=„hillary“, Chelsea^A=„chelsea“

© S. Staab, 2001

AIFB

Beispiel

∀x Mensch(x) -> sterblich(x).
Mensch(Bill).
Kind(Ehe(Bill,Hillary),Chelsea).

Andere logisch mögliche Interpretation:

A=({„queen elizabeth“, „prinz philipp“, „prinz charles“, „ehe1“,...},I).

Mensch^A={„queen elizabeth“, „prinz philipp“, „prinz charles“}
sterblich^A={„queen elizabeth“, „prinz philipp“, „prinz charles“, „Helmut Kohl“,...}
Kind^A={(„ehe1“, „queen elizabeth“),(„ehe2“, „prinz philipp“)}

Ehe^A={(„Helmut Kohl“, „prinz charles“, „ehe1“), („...“, „...“, „ehe2“),...}
Bill^A=„Helmut Kohl“, Hillary^A=„prinz charles“, Chelsea^A=„queen elizabeth“

© S. Staab, 2001

AIFB

Semantik der Prädikatenlogik (Fortsetzung)

Sei F eine Formel und A=(U,I) eine passende Struktur, dann gilt für den Wert von t in der Struktur A (d.h. A(t)):

a. Falls t eine Variable x, dann ist A(t)=A(x)
b. Falls t die Form $f(t_1,\ldots,t_k)$ hat, dann ist $A(t)=A(f)(A(t_1),\ldots,A(t_k))$

1. Falls F die Form hat $P(t_1,\ldots,t_k)$, dann ist A(F)=1, falls $(A(t_1),\ldots,A(t_k)) \in A(P)$; 0 sonst
2. Falls F die Formen $\neg G$, $(H \wedge G)$ oder $(H \vee G)$ hat, dann wie bei Aussagenlogik
3. Falls F die Form $\forall x G$ hat, so ist A(F)=1, falls für alle $d \in U$ gilt: $A_{[x/d]}(G)=1$; 0 sonst
4. Falls F die Form $\exists\, xG$ hat, so ist A(F)=1, falls ein $d \in U$ existiert, so daß $A_{[x/d]}(G)=1$; 0 sonst

$A_{[x/d]}$ ist eine zu A mit einer Ausname identische Struktur, für die aber gilt A(x)=d

© S. Staab, 2001

AIFB

Bezeichnungen

Falls A(F)=1, dann A|=F, F gilt in A, A ist ein Modell für F

Falls jede zu F passende Struktur ein Modell für F ist, so ist F allgemeingültig, eine Tautologie, |=F.

Falls es ein Modell für F gibt, so ist F erfüllbar, andernfalls ist F unerfüllbar.

© S. Staab, 2001

Pränexform

Eine Formel heißt pränex, falls sie die Form

$$Q_1 x_1 \dots Q_n x_n \, F$$

hat (Q_i sind Quantoren, x_i sind Variablen, F ist quantorenfrei).

Eine Formel heißt bereinigt, falls keine Variable frei und gebunden vorkommt und jeder Quantor eine andere Variable benutzt.

Satz: Für jede Formel F gibt es eine äquivalente bereinigte und pränexe Formel G.

© S. Staab, 2001

Äquivalenzumformungen

Die Äquivalenzumformungen der Aussagenlogik bleiben valide!
Hinzu kommen Regeln für das Handhaben der Quantoren:

1. $\neg\forall x F$ ist äquivalent zu $\exists x \neg F$
   $\neg\exists x F$ ist äquivalent zu $\forall x \neg F$
2. Falls x in G nicht frei vorkommt, gilt
   $Qx\,(F \wedge G)$ ist äquivalent zu $QxF \wedge QxG$, Q ist ein Quantor
   $Qx\,(F \vee G)$ ist äquivalent zu $QxF \vee QxG$, Q ist ein Quantor
3. $(\forall xF \wedge \forall xG)$ ist äquivalent zu $\forall x(F \wedge G)$
   $(\exists xF \vee \exists xG)$ ist äquivalent zu $\exists x(F \vee G)$
4. $\forall y \forall x F$ ist äquivalent zu $\forall x \forall y F$
   $\exists y \exists x F$ ist äquivalent $\exists x \exists y F$

© S. Staab, 2001

Geschlossene Formeln

Eine Formel heißt geschlossen, falls jede Variable durch einen Quantor gebunden ist.

Satz: Falls F eine Formel mit ungebundenen Variablen $x_1 \dots x_n$ ist, dann ist $G = \exists x_1 \dots \exists x_n \, F$ erfüllbarkeitsäquivalent zu F.

© S. Staab, 2001

Skolemform

Für jede Formel F in BPF wird die Skolemform definiert durch:

while F enthält Existenzquantor do
  begin
    $F = \forall y_1 \dots \forall y_n \exists z\, G$ (G in BPF und $n \geq 0$)
    sei f ein bisher unbenutztes Funktionssymbol;
    $F := \forall y_1 \dots \forall y_n \, G[z/f(y_1 \dots y_n)]$
  end

Satz:
Für jede Formel F in BPF gilt: F ist erfüllbar genau dann, wenn die Skolemform von F erfüllbar ist.

© S. Staab, 2001

Skolemform: Beispiel

$\forall x\ \exists y \forall z\ (\neg \forall w\ (P(a,w) \vee Q(f(x),y))) \Leftrightarrow$	Äquivalenzumformungen
$\forall x\ \exists y \forall z\ \exists w\ \neg\ (P(a,w) \vee Q(f(x),y)$	BPF
KEINE Äquivalenzumformung	
$\forall x\ \forall z \exists w\ \neg(P(a,w) \vee Q(f(x),g(x)))$	
$\forall x\ \forall z\ \neg\ (P(a,h(x,z)) \vee Q(f(x),g(x)))$	Skolemform
$\forall x\ \forall z\ \neg P(a,h(x,z)) \wedge \neg Q(f(x),g(x))$	KNF
$\{ \{\neg P(a,h(x,z))\}, \{\neg Q(f(x),g(x))\} \}$	Klauselschreibweise (implizit Allquantifizierung der Variablen)

© S. Staab, 2001

Resolution in der Prädikatenlogik

Ziel: zeige Unerfüllbarkeit

Beobachtung:

Falls Literale bis auf Negation übereinstimmen funktioniert die Resolution in der Prädikatenlogik genauso wie in der Aussagenlogik,z.B.:

$\{\{\neg P(a,h(x,z))\}, \{\neg P(a,h(x,z))\}\}$ ist unerfüllbar

© S. Staab, 2001

Resolution in der Prädikatenlogik (II)

Ziel: zeige Unerfüllbarkeit

Beobachtung:

Falls Literale nicht übereinstimmen, aber „in Übereinstimmung gebracht werden können", dann können Literale resolviert werden, da die resolvierten Klauseln aufgrund der Allquantifikation abgeleitet werden können, z.B.

$\{\{\neg P(a,h(x,z))\}, \{P(a,y)\}\}$

kann resolviert werden, da die Variable y durch die implizite Allquantifikation auch den Wert von h(x,z) annehmen können muß.

Damit kann □ abgeleitet werden.

© S. Staab, 2001

Unifikation

Der Unifikationsalgorithmus ersetzt Variablen durch Funktionen (bzw. Konstanten).

"In Übereinstimmung bringen" erfolgt durch den Unifikationsalgorithmus (oder ausprobieren).

$\{P(y)\}$	$\{\neg P(g(b,x)), \neg Q(b)\}$
$[y/g(b,a)]$	$[x/a]$
$\{P(g(b,a))\}$	$\{\neg P(g(b,a)), \neg Q(b)\}$

$\{\neg Q(b)\}$

Zusammenfassung: Resolution in PL

Input: F

1. Sei F_1 die zu F äquivalente Formel in BPF
2. Sei F_2 die erfüllbarkeitsäquivalente, geschlossene Formel zu F_1
3. Sei F_3 die (erfüllbarkeitsäquivalente) Skolemform von F_2
4. Sei F_4 die KNF von F_3, schreibe F_4 in Klauselform
5. Resolviere (unter Berücksichtigung der Unifikation), falls □ abgeleitet werden kann, ist F unerfüllbar.

Satz: Falls F unerfüllbar ist, terminiert die Resolution in endlich vielen Schritten.

Satz: Die Resolution für Hornformeln ohne Funktionssymbole terminiert in linearer Zeit.

Anfragen

Bill is president and man:	President(bill) ∧ Man(bill).
All presidents are persons:	∀x President(x) → Person(x).
Who is a person?	∀x ¬Person(x) Zielklausel

{{President(bill)},{Man(bill)},{¬President(x),Person(x)},{¬Person(x)}}

[] [x/bill] [x/bill]

{President(bill)} {¬President(bill),Person(bill)}

{¬Person(bill)}

{Person(bill)}

□

Wenn □ resolviert werden kann, ergeben die Substitutionen für die Variablen der Zielklausel, die Antworten (hier: [x/bill])!

Reifikation

President(bill) ∧ Man(bill).
∀x President(x) → Person(x).
∀x ¬Person(x)
Wie schreibe ich: Gib mir alle Unterkonzepte von Person???

Antwort a: Logik höherer Ordnung (Berechnungsprobleme)
Antwort b: Reifikation (Prädikate werden zu Funktionen/Konstanten)

True(instOf,bill,Person) ∧ True(instOf,bill,President) ∧ True(instOf,bill,Man).
True(isa,President,Person).
∀x ∀c ∀d ((True(instOf,x,c) ∧ True(isa,c,d)) → True(instOf,x,d)).
∀x ¬True(instOf,x,Person).

Frame-Logic

AIFB Inferenzmaschine:

Objektorientierte Frame-Logic Notation wird in reifizierte Horn-Klauseln transformiert.

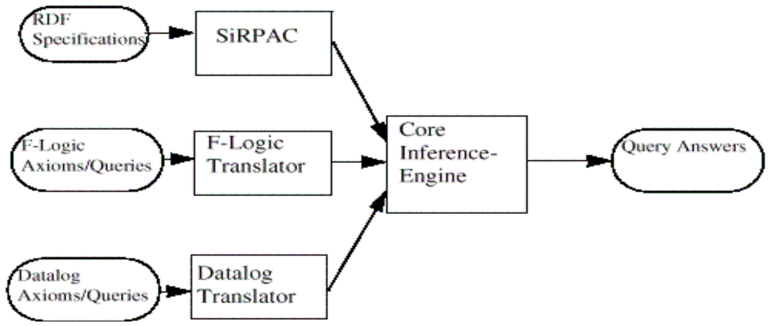
Die Beantwortung der Anfragen erfolgt durch Anwendung eines syntaktischen Schlußfolgerungsmechanismus. Dieser beinhaltet Unifikation und eine Strategie, die man sich - extrem vereinfacht - als Resolution vorstellen kann.

Konzepte und Relationen sind wie “gewöhnliche” Objekte in F-Logic, können also für Regeln verwendet und abgefragt werden.

Frame-Logic - einfachste Notationsprimitive

1. Isa / subconceptOf	C :: D.	President :: Person.
2. instanceOf	a : C.	bill : President.
3. Deklaration	C[R=>>D].	Person[kennt=>>Person].
4. Attribuierung	a[R->>b].	bill[kennt->>hillary].
5. Constraint	$\forall x_1,\ldots,x_m\ P_0$ <- P_1 AND ... P_n. Dabei $P_0 \ldots P_n$ aus 1-4.	FORALL X,Y Y[kennt->>X] <- X:Person[kennt->>Y]
6. Anfragen	$\forall x_1,\ldots,x_m$ <- P_1 AND ... P_n. Dabei $P_1 \ldots P_n$ aus 1-4.	FORALL X <- X:Person.

Architecture of SiLRI



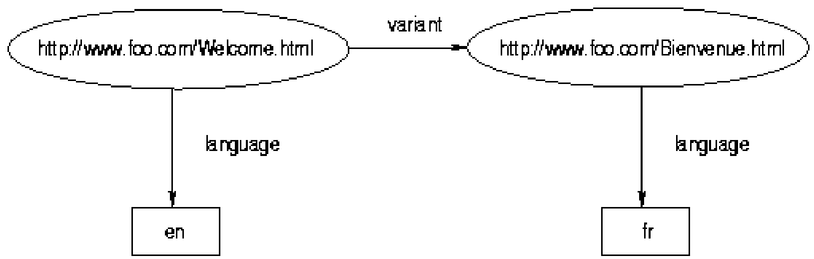
Example

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/TR/WD-rdf-syntax#"
    xmlns:a="http://www.schema.org/usefulpredicates/">

<rdf:Description about="http://www.foo.com/Welcome.html>
    <a:language>en</a:language>
    <a:variant rdf:resource="http://www.foo.com/Bienvenue.html" />
</rdf:Description>

<rdf:Description about="http://www.foo.com/Bienvenue.html">
    <a:language>fr</a:language>
</rdf:Description>
</rdf:RDF>
```


Datamodel



© S. Staab, 2001

Translation into F-Logic

„http://www.foo.com/Welcome.html“[„language“->“en“].

„http://www.foo.com/Welcome.html“[„variant“
->“http://www.foo.com/Bienvenue.html“].

->“http://www.foo.com/Bienvenue.html“[„language“->“fr“].

© S. Staab, 2001

RDFS-Example

```
<rdf:RDF>
   <rdfs:Class rdfs:ID="Employee">
        <rdfs:subClassOf
                rdf:resource="http://ontology.org/human-ontology#Person"/>
   </rdfs:Class>
   <rdfs:Class rdfs:ID="Researcher">
        <rdfs:subClassOf rdf:resource="#Employee"/>
   </rdfs:Class>
   <rdf:Property ID="cooperatesWith">
        <rdfs:domain rdf:resource="#Researcher"/>
        <rdfs:range rdf:resource="#Researcher"/>
   </rdf:Property>
</rdf:RDF>
```

Representation in F-logic:
Employee :: Person.
Researcher :: Employee[cooperatesWith=>>Researcher].

© S. Staab, 2001

SiLRI- Combine RDF mapping with rules

Combine RDF mapping with rules, e.g.

Representation in F-logic:
Employee :: Person.
Researcher :: Employee[cooperatesWith=>>Researcher].
FORALL X Y[cooperatesWith->X] <- X[cooperatesWith->Y].
Sst[cooperatesWith->rvo].

Query:
FORALL X,Y <- X[cooperatesWith->Y].

Result:
(sst,rvo)
(rvo,sst)