

Logic, Proof and Trust

Arlette, Gwyneth, Matthijs, Melanie, Wietske and Wilco

March 20, 2007

Logic and Proof

Practical Session I

Trust

Practical Session II

Discussion

Logic and the semantic web

- ▶ Previous meetings about *knowledge representation*
- ▶ *Logic* is the foundation of knowledge representation

Advantages of logic

- ▶ High expressive power
- ▶ Well-understood formal semantics
- ▶ Precise knowledge of logical consequence
- ▶ Automatic derivation possible
- ▶ Semantic logical consequence coincides with syntactic derivation
- ▶ *Predicate logic* is unique in this sense
- ▶ Logic can provide explanations for answers

Subsets of predicate logic

- ▶ Trade-off between *expressive power* and *computational complexity*
- ▶ Different reasonable subsets of predicate logic
 - ▶ Description Logic (examples: OWL Lite, OWL DL)
 - ▶ Horn Logic (examples: RuleML, Prolog)
- ▶ Description Logic and Horn Logic no subset of each other:
 - ▶ Not in DL: 'cookies made of chocolate taste good'
 - ▶ Not in HL: 'american cookies either contain chocolate or peanuts'

Horn logic I

Database: disjunctions of literals, at most one positive literal
Different cases:

	zero positive literals	one positive literal
no negative literals	-	Fact
some negative literals	Goal	Rule

Horn logic II

- ▶ One positive literal, no negative literals: fact
madeOfChocolate(americancookie)
- ▶ One positive literal, some negative literals: rule
 $\neg \text{madeOfChocolate}(X) \vee \neg \text{cookie}(X) \vee \text{tastesGood}(X)$
logically equivalent to
 $\text{madeOfChocolate}(X) \wedge \text{cookie}(X) \rightarrow \text{tastesGood}(X)$
- ▶ Zero positive literals, some negative literals: goal
 $\neg \text{tastesGood}(X)$ (to find counterexample!)

Syntax

Consider the following program:

- ▶ $\text{madeOfChocolate}(\text{americancookie})$.
- ▶ $\text{cookie}(\text{americancookie})$.
- ▶ $\text{madeOfChocolate}(X) \wedge \text{cookie}(X) \rightarrow \text{tastesGood}(X)$.

Now we can conclude that american cookies taste good.

We can see the different building blocks of Horn Logic: *variables*, *constants*, *predicates* and *function symbols*.

RuleML

Rules can be expressed in XML: $q(b) \rightarrow p(a)$
can be represented as

```
<rule>
  <_head>
    <atom>
      <predicate>p</predicate>
      <term>
        <const>a</const>
      </term>
    </atom>
  </_head>
  <_body>
    <atom>
      <predicate>q</predicate>
      <term>
        <const>b</const>
      </term>
    </atom>
  </_body>
</rule>
```

Monotonicity

- ▶ Predicate logic is *monotonic*: if a conclusion can be drawn it remains valid even if new knowledge becomes available
- ▶ There also exist non-monotonic logics
- ▶ Example: chocolate cookies taste good; chocolate cookies which have reached their expiration date don't taste good
- ▶ What to select in case of conflicting rules?
 - ▶ Rule with most source with most authority
 - ▶ Most recent rule
 - ▶ Most specific rule

Nonmonotonic rules in XML

- ▶ Nonmonotonic rules can be represented in XML in the same way
- ▶ A few differences:
 - ▶ There are no function symbols
 - ▶ Negated atoms may occur in the head and the body of the rule
 - ▶ Each rule has a label
 - ▶ Apart from rules and facts, a program also contains priority statements
- ▶ Example of a priority statement:
`<stronger superior="r1" inferior="r2" />`

TRIPLE - an alternative approach (1)

- ▶ RDF-triples are represented as follows:
- ▶ `AlbertHeijn[sells->americancookies]`.
- ▶ TRIPLE: rule language especially designed for querying and transforming RDF models
- ▶ RDF models are explicitly available in TRIPLE:
- ▶ Statements that are true in a specific model are written as `@model1`
- ▶ `AlbertHeijn[sells->americancookies]@model1`.

TRIPLE - an alternative approach (2)

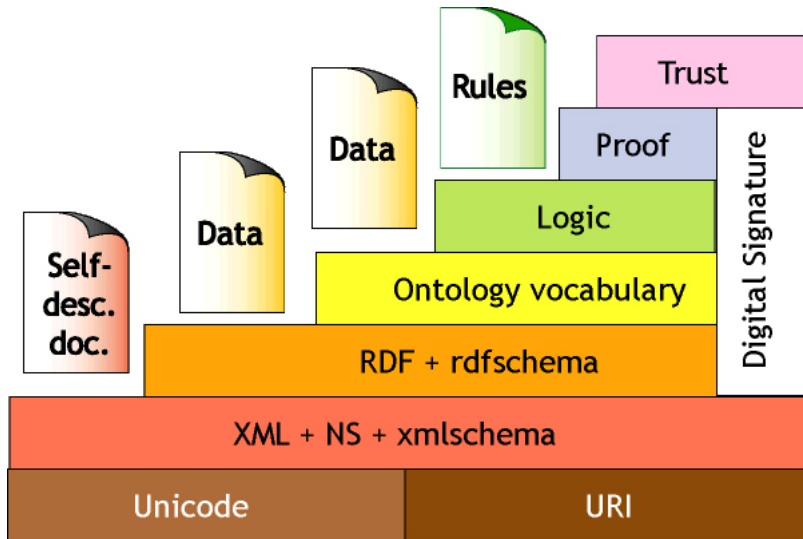
- ▶ The symbols AND, OR, NOT, FORALL, EXISTS, \leftarrow and \rightarrow can be used

Example: `FORALL X X[tastes \rightarrow good]@model1 \leftarrow
X[isA \rightarrow cookie]@model1 AND
X[madeOf \rightarrow chocolate]@model1.`

- ▶ A query showing all true statements

Example: `FORALL X,Y,Z \leftarrow X[Y \rightarrow Z]@model1.`

Practical Session: Clips



Definitions of trust

- ▶ Trust is the belief in the good character of one party, presumed to seek to fulfill policies, ethical codes, law and their previous promises. (*Wikipedia*)
- ▶ The degree to which an agent considers an assertion to be true for a given context. (*in 'Finding Bacon's Key'*)
- ▶ The willingness to be vulnerable based on positive expectations about the actions of others. (*in 'Towards a Model to Understand the Influence of Trust in KSD'*)

Why do we need trust on the Semantic Web?

Content filtering

Make statements about statements

Define security at the semantic level

Knowledge sharing decisions

Applications

Using trust to personalize content.

- ▶ FilmTrust
- ▶ FOAF
- ▶ Trust ontology

```
<owl:DatatypeProperty rdf:ID="trustValue">  
<rdfs:label>Trust Value</rdfs:label>  
  <rdfs:domain rdf:resource="#TopicalTrust"/>  
</owl:DatatypeProperty>  
  
<owl:ObjectProperty rdf:ID="trustSubject">  
<rdfs:label>Trust Subject</rdfs:label>  
  <rdfs:domain rdf:resource="#TopicalTrust"/>  
</owl:ObjectProperty>  
  
<owl:ObjectProperty rdf:ID="trustedAgent">  
<rdfs:label>Trusted Agent</rdfs:label>  
  <rdfs:domain rdf:resource="#TopicalTrust"/> |  
  <rdfs:domain rdf:resource="http://xmlns.com/foaf/0.1/Agent"/>  
</owl:ObjectProperty>
```

A more formal definition of trust

	Direct exchange	Indirect exchange
$Pass(x, i, y, C)$	$B_x i \wedge B_x competent(x, i)$ $\wedge B_x r(i, y) \wedge B_x m(i, y)$ $\wedge B_x integrity(y, i)$ $\wedge B_x integrity(y, x)$	$B_x i \wedge B_x competent(x, i) \wedge$ $B_x (\exists y \in C : r(i, y) \wedge m(i, y))$ $\wedge B_x integrity(C, i)$ $\wedge B_x integrity(C, x)$
$Accept(x, i, y, C)$	$B_x r(i, x) \wedge B_x m(i, x)$ $\wedge B_x B_y i$ $\wedge B_x competent(y, i)$ $\wedge B_x integrity(y, i)$ $\wedge B_x integrity(y, x)$	$B_x r(i, x) \wedge B_x m(i, x) \wedge$ $B_x (\exists y \in C : B_y i \wedge competent(y, i))$ $\wedge B_x competent(C, i)$ $\wedge B_x integrity(C, i)$ $\wedge B_x integrity(C, x)$

Table 1. Trust-related conditions for knowledge sharing

The social aspect: sharing knowledge I

Knowledge owners prefer to share it within a controllable, trusted group under conditions negotiated for the specific situation and partners.

Three kinds of trust (sociological research):

1. dispositional
2. interpersonal
3. institutional

The social aspect: sharing knowledge II

Different types of trust apply to different exchange situations. Knowledge sharing is the result of two different socio-cognitive decisions:

- ▶ knowledge owning: decision to pass or not pass knowledge
- ▶ knowledge needing: decision to accept or reject knowledge

Security in open environments

Goals: to support autonomous systems (without direct user intervention) and generate automate security response.

Security Infrastructure:

- ▶ only *appropriate* interactions between parties
- ▶ parties should verify whether security requirements and capabilities can be met by the other parties
- ▶ address both needs of users and of system administrators

Security in open environments

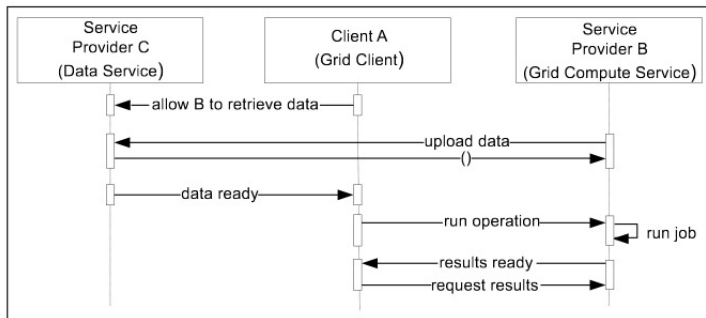
Requirements that should be met by security infrastructure:

1. description capabilities
2. reasoning capabilities
3. infrastructure capabilities

Semantic firewalls

Service operating alongside the traditional firewall to be able to describe and reason about security requirements at the semantic level.

Aims to provide dynamic, adaptive network security.



Message exchange with semantic firewall

Description capabilities

Describing what are appropriate process definitions or workflows, annotated with the related security requirements.

This could be done by Process Model ontology of OWL-S, but also by using a conversational policy.

Needed: knowledge of types or roles of interacting parties.

Message exchange with semantic firewall

Reasoning capabilities

Incorporate an appropriate description logics reasoning engine.

Determine the frequency with which the Semantic Firewall will need to perform reasoning for each suggested workflow.

Provide possibility for protected parties to address the conflicts.

Infrastructure capabilities

Implications to the more conventional Web infrastructure.

Key Free Trust

Making statements about statements.

Cryptography (signatures, keys) need not be the only basis by which we evaluate trust: *fingerprint, links to sites.*

Hypothesis (Joseph M. Reagle Jr.):

“The pervasive use of digest values to identify the statements in the Semantic Web will engender a preponderance of evidence for trust without cryptography.”

Key Free Trust

Consequences:

- ▶ complex public key infrastructure may not be necessary
- ▶ cryptographic signatures might not be necessary to make trust evaluation about a statement

Inferring trust in Social Networks

How can we infer trust on the basis of existing trust relations?

Example: TidalTrust

- ▶ For a fixed trust rating, shorter paths have a lower error
- ▶ For a fixed path length, higher trust ratings have a lower error
- ▶ Shortest path connects source and sink
- ▶ Minimum trust threshold

Practical Session: Real life trust experiment

- ▶ FOL (First Order Logic) is too strict to model human logic, so we shouldn't use it.
- ▶ In the article by Golbeck et. al., a part is about the influence of "nearness" on trust. If A and B are very close to each other, they might share some beliefs and are likely to be willing to invest a similar amount of trust in C. It's argued that this might not be true for humans, but only for agents. Why does it work for agents then? And isn't the theory originally derived from social networks?

- ▶ Does the Semantic Firewall really make things secure? Isn't trust a big issue here too? How does one know that an agent keeps his promise concerning for example destroying private data?
- ▶ The Friend Of A Friend principle is never going to work, due to a lack of (freely accessible, public) information sources and lack of information about people in general.
- ▶ Inference is dangerous. Information can be inferred that was never meant to be, with huge consequences especially when trust and security are involved. How can we control this?