CS 6910-ACIS - Project 7
THE SEMANTIC WEB
Web of Trust
Terry Goodman
Preferential Reasoning?

- WWW is currently a vast collection of knowledge.
- This knowledge contradicts itself.
  - Site (a) gives its “opinion” of truth.
  - Site (b) gives its “opinion” of truth.
  - Sites (a and b) do not necessarily agree!
- Who can we trust?
Preferential Reasoning?

- Humans can “innately” decide for themselves what resource they “trust” or “believe” in more.
  - They still may be incorrect themselves, but least they made a decision!
- Software Entities lack this “innately” human quality.
  - Deciding between good and bad data is not as obvious to an Artificial Intelligence.
Preferential Reasoning and Answer Set Programming

• Based heavily on the concept of “Web of Trust”.
  – We seen in the previous presentation, the idea of “Locality” of trust.
  – Agent (a) “computed” its respective trust of some Agent (c) based upon its “locally trusted” Agent (b).
  – i.e ( Summation of Aggregations )

• Answer Set Programming
  – Uses Stable Model Semantics with the underlying concept of negation as representing “failure”.
Answer Set Programming (ASP)

• Logic Program
  – Represents knowledge one wishes to represent.
  – An encoding to a particular problem
    • i.e ( A Planning Problem )

• Answer Sets
  – The results of the Logic Program
  – Represents the Logic Programs “Intentional Knowledge”
  – Represents “Solutions” to the particular problem.
Answer Set Programming
Logic Programs

– Traditional Logic Programs
  • Limited view on world
    – Limited to their own knowledge (no external sources)

– Answer Set Programming Logic Programs
  • Semantic Web Agents
    – Conceptual “call literals”
      » View as function calls
    – Normal Literals
      » View as program variables
Answer Set Programming
Logic Programs

- Call Literals Representative of “Knowledge Sources”
  - OWL DL
  - SQL Database
  - RDF Data
- Call Literals associated to some “Decision Problem”
  - \( \neg \text{train} \leftarrow \text{geo1.300km}(\text{brussels}, \text{madrid}) \)
  - In English: (if brussels is more than 300 KM from madrid, one should NOT take a train)
- This “fact” of distance may come from one of many data sources
  - A SQL query on a geographic database for example
- Logic Programs that wish to use these data sources only need to know there public interfaces (how to call them)
Answer Set Programming
Logic Programs

• Since there may exist call literals (or rules) that are contradicting .... We must be able to “decide” which to take
  - \( \neg \text{train} \leftarrow \text{geo1}.300\text{km}(\text{brussels}, \text{madrid}) \)
  - \( \text{train} \leftarrow \text{geo2}.\text{notDividedByWater}(\text{brussels}, \text{madrid}) \)

• One rule states we shouldn’t take the train, the other might state we should.

• How can we decide ?
Semantic Decideability

- It's not feasible to allow a Software Agent to remain “undecisive”
- Fuzzy logic doesn’t work for the Semantic Web.
- New Knowledge can’t be “discovered” as long as there are conflicting decision rules.
- Knowledge sources must be “weighted” as to some level of preference, over others.
‘Web of Trust’
Semantic Defeat

- Extended Answer Set Semantics contains the notion of “defeat”
  - A call literal (a) is allowed to be “unsatisfied” under the condition
    - A Call literal (b) contains the “opposite head”
      \{ ¬\text{train} \} to \{ \text{train} \}
- So one way to look at Semantic Defeat is to say as long as someone returns “true” then its ok.
“Semantic Preference”

• As in the real world, just because someone says you should do something, doesn’t always mean you really should.
  – Here lies the “Preference” of ones decision over anothers.

• How is software any different?
  – Extended Answer Set Semantics provides for the notation of Preference of Rules (Call Literals)
Preference Of Rules

• Semantic Preference Notation
  – Tr1 < Tr2
    • States a preference of satisfaction of Rule Tr1 Over Rule Tr2
    • This preference is easily translated to a preference of each rules underlying Answer Sets.
      – { ¬train } < { train }
      – This line of reasons goes hand in hand with OPPNets with regards to preference to device discovery.
        • Preferring longer range mobile devices over smaller range devices
Preference Of Rules

– Without Extended Answer Set Semantics
  • Software agents would typically assume its own knowledge to correct.
  • Only in External knowledge sources are there sources of conflict.

– We need some concept of
  • Authority
  • Reliability
Preferential Reasoning and the “Web of Trust”

• Using the locality principle of trust
  – We tend to trust our neighbors more than strangers.
• We can “Compute” a over-all sense of “trust” and believe using the same graph style approach we seen before
• This computed graph of trust can be utilized to make rule preference decisions from external knowledge sources.
Preferred Answer Set
Programming Using Call Literals

• Our Logic Program first begins with some known “facts”
  – Movies(kine)
  – Rest(pizzi)
  – Rest(ilpast)
  – Time(8:00pm)
  – Time (10:00pm)

• We would like to have a Semantic Rule that would produce a “plan” for an evening that includes dinner and a movie at the times 8:00 and 10:00
Preferred Answer Set
Programming Using Call Literals

• Our Rule
  – P : plan( Rest : rest, Time1 : Time, Movies : movie, Time2: Time ) <- Rest.res(Time1), geo.near(Rest, Movies), Time1 != Time2, not otherpl(Rest, Time1, Movies, Time2)

• The Interpretation !
  – We wish to go to some movie and dinner, we wish for the theater, and restaurant to be close to each other, and dinner time cant be equal to the movie time … else we couldn’t do both. Also I only want one answer in the set, since I wont be to decide otherwise !
A good example of preferential reasons in the content of Extended Answer Set Semantics is the purchasing of stock.
- \( \text{Stock(lmby)} \leftarrow \text{buy(s)} \leftarrow \text{ft.buy(s)}, \text{nyt.buy(s)} \)
- \( \text{Stock(wtww)} \leftarrow \neg \text{buy(s)} \leftarrow \neg \text{not pdh.buy(s)} \)

Here are stating the call literals Answer Sets to be retrieved from the external knowledge sources of the financial times (ft) the New York Times (nyt) and a Personal friend (pdh)
Based Upon the call literal evaluations of:
  – Eval(ft.buy(lmby)) = false
  – Eval(ft.buy(wtww)) = true
  – Eval(nyt.buy(wtww)) = true
  – Eval(nyt.buy(wtww)) = true
  – Eval(pdh.buy(wtww)) = false
  – Eval(pdh.buy(wtww)) = false

We need to make a decision on whos “knowledge” we trust more than the others, as we some serious conflict here on the Answer Sets we Received.
As seen previously we can denote “preferential reasoning” in the Extended Answer Set Semantics via the ‘<‘ logic operator.

\[ \{ \text{not pdh.buy(lmby), not pdh.buy(wtww)} \} < \{ \text{ft.buy(lmby), ft.buy(wtww), nyt.buy(lmby), nyt.buy(wtww)} \} \]

In this notation we are stating we “believe” the knowledge of our stock investor friend (pdh) than the opinions of the Financial Times or New York Times.

We are saying the there is more “Credibility” to our friends knowledge than the others.

This decision on credibility could be computed via of Web of Trust we seen previously.
Conclusions

• Extending the concept of logic programming with Extended Answer Set Semantics provides for an intuitive extension of trust into automated decision making.

• Existing Knowledge sources need not be updated, logic programs only need to know how to interface to them. (all the more reason for an adoption of RDF Data Document standard)

• Semantic Rules and Decision making are reduced logically based on Extended Set Semantics (See Semantic Defeat)
  – This logically deduction can then be ordered on preference determined from the web of trust and locality of trust principles.
Conclusions for OPPNets

- Preferential Reasoning has multiple uses within the realm of Opportunistic Networks.
  - Help Node Discovery, and underlying preference of one potential helper to another.
  - Web of Trust concepts of determining helper node “trustability”
    - i.e (if one agent doesn’t trust another, but other helper nodes do trust this agent, how can we decide is we SHOULD trust the new agent)
  - Increased Trust within OPPNets offloads risk of Privacy and Security.
    - Not completely
    - But its a lot easier to have Security and Privacy when you only deal with those you “trust”
References

- Stijn Heymans, Davy Van Nieuwenborgh, Dirk Vermeir: Preferential Reasoning On a Web Of Trust. Dept. of Computer Science Vrije Universiteit Brussel, VUB