Human-Understandable Inference of Causal Relationships

Alva L. Couch – Tufts University
Mark Burgess – Oslo City University
Explaining relationships between entities

- A knowledge base describes relationships between entities.
- Humans often need to understand relationships between entities to troubleshoot a computer network.
- We describe how to create a “story” that concisely describes relationships between two chosen entities.
This talk in a nutshell

- Unrestricted logical abduction is **too much** explanation of a relationship between network entities to be useful. (“The porridge is too hot.”)

- Using links between items without use of any logic is **too little** explanation. (“The porridge is too cold.”)

- Our “stories” – based upon a very limited form of abduction – are **just right good enough.**
How this work came about

- Mark asked Alva to comment on Mark’s new topic map system for documenting Cfengine.
- Alva reported that it was frustrating; things he needed couldn’t be found quickly enough by browsing.
- Mark told Alva to fix it...
- Several weeks and attempts later, Alva did...!
The problem with browsing knowledge bases...

- ...is that one doesn’t have **time to browse**!
- One doesn’t approach network knowledge with an unfocused **desire to learn**.
- One browses with **Rome already burning**, and **no time to fiddle around**!
- How can we simplify finding exactly the knowledge we need in a knowledge base, when we need it?
Some failed approaches

- Using unrestricted computer logic is too time-consuming and difficult to explain to a user.
- Considering connections – without logic – leads to useless connections, e.g.,
  - Cfengine is written by Mark.
  - Mark wrote *Analytical System Administration*.
  - So Cfengine is somehow connected to the book *Analytical System Administration*???
- Conclusion: need a **limited form of logical reasoning that explains relationships of interest (ROIs)**.
This work is difficult to characterize...

- It is **not**
  - natural language processing... ... even though it outputs natural language explanations...
  - ontological reasoning... ... because it defines **relationship semantics** via interactions between relationships (rather than object semantics as interactions between objects)

- It is:
  - a form of **logical abduction**... ... but it does logic via **graph algorithms**...
  - a shorthand for
    - Making **new connections** between entities.
    - **Simplifying fact bases** via derived rules.
    - Explaining derived **connections** in terms of existing ones.
Four relationships of interest

- **X determines Y**: X controls Y’s behavior.
- **X influences Y**: X has partial control over Y.
- **X might determine Y**: in some cases, X controls Y’s behavior.
- **X might influence Y**: in some cases, X has partial control over Y.

These are the target relationships about which we want more information.

(Note: modal relationships are encapsulated inside formal symbols, e.g., might determine.)
Binary relationships

- Many (but not all) entity relationships are **binary**, e.g.,
  - The host muffin provides name service for the domain cs.tufts.edu.
  - The host houdini is part of the domain cs.tufts.edu.
  - Therefore, the host muffin provides name service for the host houdini.

This reasoning is a limited form of **logical abduction**, i.e., it explains the relationship between muffin and houdini in terms of their relationships to a third party eecs.tufts.edu.
Weak transitive laws

- The inference in the previous slide looks something like a transitive law:
  
  If X provides name service for Y, and Y contains Z, then X provides name service for Z.

- We call this kind of rule a “weak transitive law”.

- We notate it as

\[
<\text{provides name service for}, \\
\text{contains}, \\
\text{provides name service for}>
\]
Parsing statements into relationships

- Annotate the text with attributes:
  - (The) **host** muffin **provides name service for** (the) **domain** cs.tufts.edu.
  - (The) **domain** cs.tufts.edu **contains** (the) **host** houdini.
  - Therefore, (the) **host** muffin **provides name service for** (the) **host** houdini.

- We typeset **nouns** in fixed type, **qualifiers** in script, and **relationships** via underlining.
These sentences look like **topic map associations** as described by S. Pepper.

Consider “(The) **host** muffin **provides name service for** (the) **domain** cs.tufts.edu.”

**muffin** and **cs.tufts.edu** are **topics**, i.e., names about which knowledge is stored.

**host** and **domain** are **topic roles**, i.e., qualifiers that determine the **scope** of topic names **muffin** and **cs.tufts.edu**, respectively, in the context of the association.

**provides name service for** is an **association**, i.e., a relationship between topics.
Symbols and meanings

- As in topic maps, *muffin, provides name service for*, and *cs.tufts.edu* are **formal symbols**, devoid of meaning.

- As in topic maps, every association has an **inverse**, e.g.,
  - “(The) *host muffin provides name service for* (the) *domain cs.tufts.edu*.”

  has the inverse association:
  - “(The) *domain cs.tufts.edu uses name server host muffin.*”

- Inverses for relationships are **defined** (in English), and never inferred.

- Meanings are derived from **where symbols appear** in relationships and laws.

- (Note: roles are part of the association: might write the above as *cs.tufts.edu domain uses name server host muffin.*)
Basis for our troubleshooting logic

- A set of **architectural facts**, about how neighboring entities relate to one another.
- A set of **logical rules** that allow one to infer how non-neighboring entities relate to one another.
Our rules

- There are only two kinds of rules, with different purposes: for relationships $r, s, t$ and entities $X, Y, Z$:
  - An **implication** $r \rightarrow s$ means “If $XrY$ then $XsY$”. These rules **raise the level of abstraction** at which reasoning occurs.
  - A **weak transitive law** $<r,s,t>$ means “If $XrY$ and $YsZ$ then $XtZ$”. These rules **make new connections** between unconnected entities.
Layers of abstraction

- **X provides DNS**: a low-level statement, concrete.
  
  \[ \Downarrow \]

- **X determines DNS**: a higher level of abstraction.
  
  \[ \Downarrow \]

- **X influences DNS**: an even higher level of abstraction.
  
  DNS **is used by** Y: a concrete statement.
  
  \[ \Downarrow \]

- DNS **influences** Y: an abstract statement.

- Then, using \(<\text{influences, influences, influences}>\), we infer **X influences Y**, which can be explained as

- **X provides DNS is used by Y**: a story of X influences Y.

- Pattern: **reason at a high level, explain at a concrete level.**
A simple example

host01
provides DNS for influences

host02
provides file service for influences

host03
is used by influences

user01
Lifting by implication
Inferences under the hood: Transitive closure under <influences,influences,influences>
Are transitive laws enough?

- Many inferences are only weakly transitive:
  - <determines, is a part of, influences>
  - <is a part of, determines, determines>
  - <influences, is a part of, influences>
  - <is a part of, influences, influences>
  - <influences, is an instance of, might influence>
  - <is an instance of, influences, influences>
- These rules might be considered a definition of influences.
A less trivial example

host01

is an instance of

DHCP server

feeds data to

influences

DNS server

has instance

host02

Inferences under the hood:

<is an instance of, influences, influences>

<influences, has instance, influences>

Story seen by user
Computing stories

- Relationships are sets.
- Semantic networks are graphs.
- Distance is # of weak transitive laws required to link two topics.
- Computation uses variants of shortest-path algorithms in graphs.
Logic and sets

- We can think of relationships as sets, e.g.,
  \[ \text{provides name service for} = \{ (X, Y) \mid X \text{ provides name service for } Y \} \]
- An implication \( r \rightarrow s \) raises the level of abstraction of a statement from specific to more general, e.g.,
  \[ \text{provides name service for} \rightarrow \text{influences as relationships means that} \]
  \[ \text{provides name service for} \subseteq \text{influences as sets} \]
- The rule \( r \rightarrow s \) is equivalent with the assertion \( r \subseteq s \)
Weak transitive laws and sets

- \(<r,s,t>\) is also equivalent to a subset assertion:
- \(<r,s,t>\) means “If XrY and YsZ then XsZ.”
- If we let \((r \otimes s) = \{ (X,Z) | XrY and YsZ \} \)
- Then the rule \(<r,s,t>\) is equivalent to the assertion \((r \otimes s) \subseteq t\).
Summary of set relationships

\[ r' \cap s' \cap \cap \cap r \otimes s \subseteq t \Rightarrow r' \otimes s' \subseteq t' \cap \cap t' \]
Or, using our rule notation

\[
\begin{align*}
\langle r', s', t' \rangle & \Rightarrow \langle r', s', t' \rangle
\end{align*}
\]
Why the set-theoretic formulation is important

- The rules do not backtrack, so it is never necessary to use backward chaining or logic programming.
- One can add information without restarting computation.
- One can formulate computation in terms of graph algorithms, rather than in terms of logic!
How the algorithm works

- Complete the facts by adding explicit inverses.
- Complete the rules by adding implied rules.
- Apply implied rules to completed facts.
- Compute minimum-distance facts by variant of all-pairs shortest-path.
- (For the relationships of interest.)
Why the set-theoretic characterization is important

- Can restart a partial calculation.
- Can add new facts or rules without starting over.
- Can implement the algorithm in a Map/Reduce environment.
Some counter-intuitive aspects of the logical calculus

- Modal relationships, e.g., might influence, are just formal symbols like any other relationship.
- Modal relationships are defined by means of weak transitive laws.
- The purpose of the laws is not to define logic, but rather, to define terms in a language via their logical inter-relationships.
- Thus this is not a calculus of logic, but rather, a calculus of language and meaning.
Practical Applications

- Abducting the relationship between two elements X and Y: this is a minimum-distance story of the relationship between X and Y.
- Finding the most likely causes of a set of symptoms: input is symptoms, output is the set of things that influence them, in order of distance.
Some subtleties

- There is no way to retract a fact or rule.
- Rather we version the entities and relationships as necessary to change their definitions.
  - New facts correspond to a new entity.
  - New rules correspond to new relationships.
Conclusions

- What we have here is not really computer logic.
- It is instead a clever way to manipulate natural language to explain relationships.
- It looks like abduction on the surface, but its laws choose convenient explanations rather than inferring previously unknown truth.
- Next steps: Map/Reduce implementation, user testing.
Human-Understandable Inference of Causal Relationships

Alva L. Couch – Tufts University
Mark Burgess – Oslo City University