Nonmonotonic Reasoning as a Temporal Activity

Daniel G. Schwartz

Florida State University
Tallahassee, Florida, USA

schwartz@cs.fsu.edu

Objective

To create a framework for modeling the reasoning processes of an artificial agent to the extent that those processes adhere to the principles of a well-defined logic.

Key Idea

Represent reasoning as an activity that takes place in time.

- Basic approach originally published in:

  Dynamic reasoning with qualified syllogisms, *Artificial Intellgence*, 93, 1-2 (1997) 103–167.

- An example illustrating the newer idea of a controller was published as:

  Formal specifications for a document management assistant, *International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CISSE 09)*, an e-conference based at University of Bridgeport, CT, December 4-12, 2009.

- A 65-page article, Dynamic reasoning systems, is currently in review with *ACM Transactions on Computational Logic*.

Outline

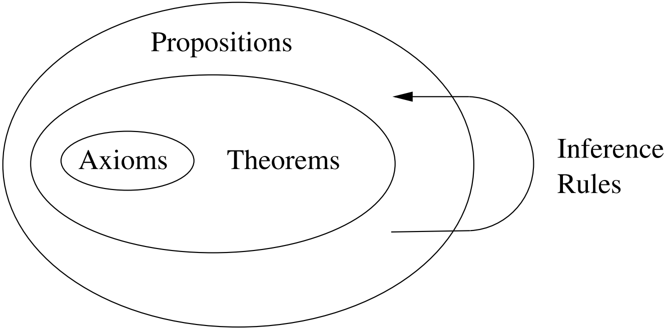
1. Classical Formal Logical Systems
2. Nonmonotonic Reasoning
3. Dynamic Reasoning Systems
4. Example: Document Management Assistant

1. Classical Formal Logical Systems

Consist of:

- A Language for expressing propositions
- Axioms
- Inference rules
- Theorems: All propositions that can be derived from the axioms by means of the rules

Classical System Structure



2. Nonmonotonic Reasoning

- Classical systems are monotonic:

  Adding new information (axioms) increases the set of theorems.

- Everyday reasoning often is nonmonotonic:

  Adding new information (axioms) can cause one to go back and retract old conclusions (theorems).

Examples

Opus the penguin (Touretsky, 1984)

- On Tuesday given: Opus is a bird.
  - Already know: (Typically) birds can fly.
  - Conclude: (Tentatively) Opus can fly.
- On Thursday given: Opus is a penguin.
  - Already know: Penguins cannot fly.
  - Go back and retract the above conclusion; opus can't fly.

Examples (Cont.)

Object classification

- Given: $A \subset B$
- Assume $B \cap C = \emptyset$
- Assign object x to category A
  - Conclude: $x \in B$
- Assign x to category C
  - Observe confilct
- Then
  - Remove assignment $x \in C$, or
  - Retract assumption $B \cap C = \emptyset$, or
  - Remove assignment $x \in A$ and retract $x \in B$.

Examples (Cont.)

Robot motion planning and learning

- Conjecture: There is a clear path from point A to point B.
- Begin moving from A toward B.
- Encounter an obstacle.
- Retract/revise conjecture.
- Try to circumnavigate obstacle.

The Problem

How to represent and manage this type of reasoning.

Well-known Early Approaches

- Truth (or Reason) Maintenance — J. Doyle, 1979, 1988
- Circumscription — J. McCarthy, 1980
- Default Logic — R. Reiter, 1980
- Nonmonotonic Logic — D. McDermott and J. Doyle, 1980

Later Approaches

- AGM Belief Revision – C.E. Alchourrón, P. Gärdenfors and D. Makinson, 1985
  - Subsequent work by S.O. Hansson, 1999, and many others
- Step/Active Logic – D. Perlis and J. Elgot-Drapkin, 1990; M. Miller, 1993
  - Represents reasoning as a temporal activity
- Answer Set Programming – Y. Dimopoulos, B. Nebel and J. Köhler, J., 1997, and many others (notably C. Baral, 2003)
  - An adaptation of the Prolog programming language

The Present Approach

Has elements in common with all of:

- Reason maintenance
- AGM belief revision
- Step/active logic

but differs from each of them in important respects.

3. Dynamic Reasoning Systems

Explicitly portrays reasoning as a temporal activity.

- Inference rule applications occur in discrete time steps.

Provides a computational reasoning framework.

- Replaces the infinite set of theorems with a finite derivation path.

Accomodates any well-defined and sound formal logic (classical, modal, multivalent, fuzzy, etc.)

For the present discussion assume classical first-order predicate logic (adapted from A.G. Hamilton 1988). This has:

A language for expressing relations and quantification ($\forall$) over individuals, using the logical connectives ($\neg, \vee, \wedge, \rightarrow, \leftrightarrow$) and the falsehood symbol ($\perp$).

Axiom schemas:

1. $(\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{A}))$
2. $((\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C}) \rightarrow ((\mathcal{A} \rightarrow \mathcal{B}) \rightarrow (\mathcal{A} \rightarrow \mathcal{C})))$
3. $((\neg\mathcal{A} \rightarrow \neg\mathcal{B}) \rightarrow (\mathcal{B} \rightarrow \mathcal{A}))$
4. $(\perp \leftrightarrow (\mathcal{A} \wedge \neg\mathcal{A}))$
5. $((\forall x)\mathcal{A} \rightarrow \mathcal{A})$
    - Has two instantiations; see the following.
6. $((\forall x)(\mathcal{A} \rightarrow \mathcal{B}) \rightarrow (\mathcal{A} \rightarrow (\forall x)\mathcal{B}))$

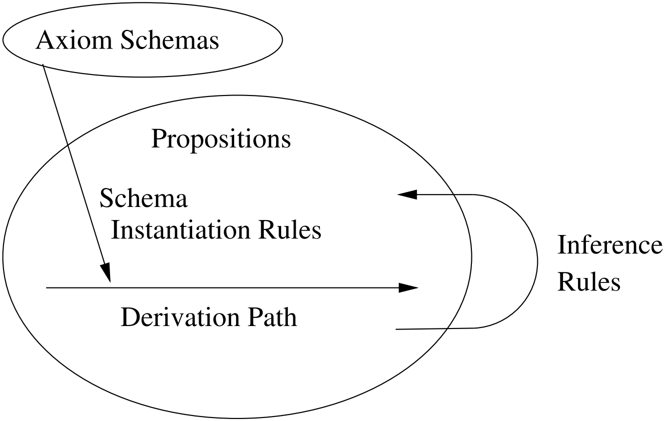
Inference rules:

1. *Modus Ponens*: From P and $(P \rightarrow Q)$ infer Q.
2. *Generalization*: From P infer $(\forall x)P$.

Our example will make use of some derived inference rules:

- *Hypothetical Syllogism*: From $(P \to Q)$ and $(Q \to R)$ infer $(P \to R)$.
- *Aristotelean Syllogism*: From $(\forall x)(P \to Q)$ and $P(a/x)$, infer $Q(a/x)$.
- *Subsumption*: From $(\forall x)(P \to Q)$ and $(\forall x)(Q \to R)$, infer $(\forall x)(P \to R)$.
- *Conflict Detection*: From $(\forall x)(\neg(P \land Q))$, $P(a/x)$, and $Q(a/x)$ infer $\perp$.

Dynamic Reasoning System Structure



A dynamic reasoning system (DRS) is comprised of:

1. a path logic
   - Logics are generic
2. a controller
   - Controllers are application domain specific

Path Logic

Propositions are entered into the derivation path either:

1. by instantiating an axiom schema
   - called logical axioms
   - expressing absolute knowledge
2. from an external source (human user, sensor, etc.), e.g., Bird(Opus)
   - called extralogical axioms
   - expressing domain-specific knowledge/beliefs
3. by applying an inference rule to some propositions appearing earlier in the path

Instantiating axiom schemas requires additional inference rules:

1. *Schema Instantiation 1*: Where S is one of axiom schemas (1) through (4), infer $\mathbf{S}(P_1, \ldots, P_n / \mathcal{A}_1, \ldots, \mathcal{A}_n)$, where $\mathcal{A}_1, \ldots, \mathcal{A}_n$ are all the distinct proposition meta symbols occurring in $\mathbf{S}$, and $P_1, \ldots, P_n$ are propositions.
2. *Schema Instantiation 2*: Where S is axiom schema (5), infer $((\forall x)P \rightarrow P)$, where P is any proposition, and x is any individual variable that does not occur free in P.
3. *Schema Instantiation 3*: Where S is axiom schema (5), infer $((\forall x)P \rightarrow P(a/x))$, where P is any proposition, a is any individual constant, and x is an individual variable that occurs free in P.
4. *Schema Instantiation 4*: Where S is axiom schema (6), infer $((\forall x)(P \rightarrow Q) \rightarrow (P \rightarrow (\forall x)Q))$, where P, Q are any propositions, and x is any individual variable that does not occur free in P.

Whenever a new proposition is entered into the derivation path, it is assigned a label consisting of:

1. Time stamp: an index indicating the proposition's position in the path
2. From-list: indicating how the proposition came to be entered into the path
   - If from an external source, the identity of the source
   - If derived, what inference rule was applied and the indexes (time-stamps) of the premises employed
3. To-list: indexes of all derived propositions for which the given proposition was used as a premise
4. Status indicator: whether the proposition is currently believed (*bel*) or disbelieved (*disbel*)
5. Epistemic entrenchment factor: the strength with which the proposition is held (can be modified over time)
6. Knowledge category: *a priori*, *a posteriori*, *analytic*, or *synthetic* (after Kant), affecting entrenchment/retraction

Thus the derivation path comprises the agent's current belief set (or knowledge base).

- Represents the agent's "conscious awareness"
- Grows and evolves over time
  - New beliefs are added
  - Old beliefs may be retracted
- May harbor inconsistencies
  - The agent doesn't notice this until a contradictory proposition is derived and entered into the derivation path.

But consistency should be strived for to whatever extent this is possible (the role of the controller; see the following).

Controller

Implements the agent's "purpose"

- Defines goal-seeking behavior

Every DRS has an algorithm for dialectical belief revision (after Hegel's dialectic "negating the negation")

- Invoked when a contradiction is introduced into the derivation path
- Uses from-lists to backtrack to "culprit" propositions that gave rise to the contradiction
- Retracts culprits that are least entrenched (changing their belief status to *disbel*)
    - May be automatic or human-user assisted
- Uses to-lists to forward chain and retract all propositions whose derivations employed the retracted culprit

Whenever a new proposition is entered into the derivation path, a reasoning algorithm is applied.

- Which algorithm to apply depends on the triggering event (the form of the proposition, its source, etc.)
- These reasoning algorithms are domain dependent

In general, the aims are:

1. To derive all salient consequences relevant for its domain
2. To ensure consistency for the belief set

Achieving these aims is made possible by restricting:

1. The kinds of formulas that can be input
2. The inference rules that can be applied

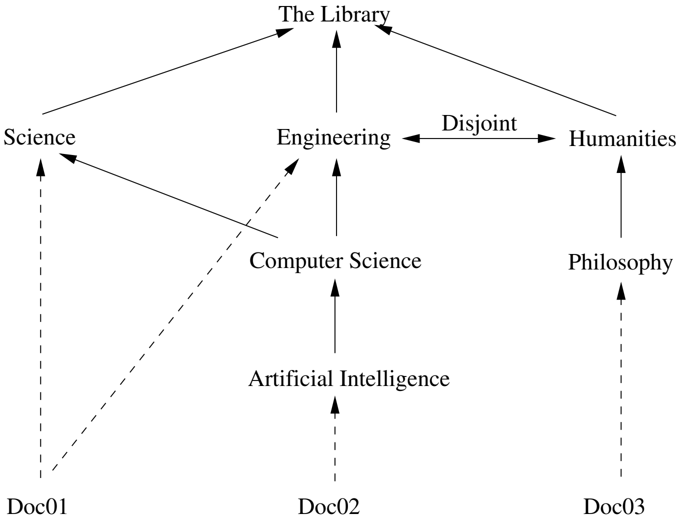
4. A Document Management Assistant

Propositions express document classifications and interrelations among classification categories.

- $\texttt{ComputerScience}(\texttt{Doc1})$
  - Doc1 is a Computer Science document.
- $(\forall x)(\texttt{ComputerScience}(x) \rightarrow \texttt{Engineering}(x))$
  - Computer Science is a subset of Engineering.
- $(\forall x)\neg(\texttt{Engineering}(x) \wedge \texttt{Humanities}(x))$
  - Engineering and Humanities are disjoint.

Only propositions of these forms are allowed to be input (entered into the derivation path) by an external agent.

Example DMA Classification Taxonomy



The figure would be generated by inputting the following propositions:

1. $(\forall x)(\texttt{Science}(x) \rightarrow \texttt{TheLibrary}(x))$
2. $(\forall x)(\texttt{Engineering}(x) \rightarrow \texttt{TheLibrary}(x))$
3. $(\forall x)(\texttt{Humanities}(x) \rightarrow \texttt{TheLibrary}(x))$
4. $(\forall x)(\texttt{ComputerScience}(x) \rightarrow \texttt{Science}(x))$
5. $(\forall x)(\texttt{ComputerScience}(x) \rightarrow \texttt{Engineering}(x))$
6. $(\forall x)(\texttt{Philosophy}(x) \rightarrow \texttt{Humanities}(x))$
7. $(\forall x)(\texttt{ArtificialIntelligence}(x) \rightarrow \texttt{ComputerScience}(x))$
8. $(\forall x)(\neg(\texttt{Engineering}(x) \wedge \texttt{Humanities}(x)))$
9. $\texttt{Science}(\texttt{Doc1})$
10. $\texttt{Engineering}(\texttt{Doc1})$
11. $\texttt{ArtificialIntelligence}(\texttt{Doc2})$
12. $\texttt{Philosophy}(\texttt{Doc3})$

DMA Controller

Objectives for this application:

1. Saliency: Derive all implicit document classifications and enter them explicitly into the derivation path.
   - I.e., derive all propositions of the form `Category`(`Doc`).
2. Ensure consistency of the belief set.

Method:

- Whenever a proposition is entered into the path, a reasoning algorithm is triggered, depending on
  1. the form of the proposition, and
  2. its source, i.e., whether it is input by an external agent (human or artificial) or derived internally by means of an inference rule application.
- Another algorithm may be triggered when a process of the above kind terminates.
- There are seven such triggering events.

DMA Controller Algorithms

1. A proposition of the form $P(a/x)$ is entered into the path by an external agent.
   a. A node representing document a and an is-an-element-of link connecting this node to the node representing P is added to the graph.
   b. The current path is searched for any propositions having the form $(\forall x)(P \rightarrow Q)$, and, for each such proposition, Aristotelian Syllogism is applied to enter $Q(a/x)$ into the path.

2. The process initiated by an event of type 1 terminates.
   a. For each proposition of the form $Q(a/x)$ that was entered into the path during the preceding process, the current derivation path is searched for propositions having the forms $(\forall x)(\neg(Q \wedge R))$ and $R(a/x)$.
   b. If found, the Conflict Detection rule is applied to enter the formula $\perp$ into the path.
      - An event of type 3.

3. The proposition $\perp$ is entered into the path as the result of an application of the Conflict Detection rule.
   a. A dialectical belief revision process is applied to remove the detected conflict.
   b. Removing the conflict may entail retracting some document classifications or retracting the associated disjointedness constraint.
      - Retractions may be automated, using epistemic entrenchment values, or user assisted.

4. A proposition of the form $(\forall x)(P \to Q)$ is entered into the path by an external agent (P and Q are distinct).
   a. Some corresponding nodes and is-a-subset-of link are entered into the graph.
   b. The current belief set is searched for any propositions having the form $(\forall x)(Q \to R)$, and, for each such proposition, the Subsumption rule is applied to enter $(\forall x)(P \to R)$ into the belief set.
      - An event of type 5.

5. A proposition of the form $(\forall x)(P \to Q)$ is entered into the path as the result of a rule application.
   a. If P and Q are distinct, the current belief set is searched for any propositions of the form $(\forall x)(Q \to R)$, and, for each such proposition, the Subsumption rule is applied to enter $(\forall x)(P \to R)$ into the belief set.
      - An event of type 5.
   b. If P and Q are identical, this indicates a loop in the subsumption hierarchy. In this case, a backtracking process is carried out utilizing from-lists to obtain a list of all extralogical axioms involved in deriving this formula. The status of one or more such axioms must be changed from from *bel* to *disbel* (user assisted or based on epistemic entrenchment), after which the controller then starts with these chosen axioms and, using to-lists, forward chains through the indicated derivations and changes the status from *bel* to *disbel* for all propositions whose derivations depended on the chosen axioms.

6. The process initiated by an event of type 4 terminates.
   a. The belief set is searched for any proposition having the form $(\forall x)(P \to Q)$, and, for each such proposition, Aristotelian Syllogism is applied to enter $Q(a/x)$ into the path.
   b. Do the same as for event type 2.

7. A proposition of the form $(\forall x)(\neg(P \wedge Q))$ is entered into the path by an external agent.
   a. An is-disjoint link between the nodes representng P and Q is entered into the graph.
   b. The belief set is searched for pairs of propositions having the forms $P(a/x)$ and $Q(a/x)$, and, if found, the Conflict Detection rule is applied to enter the proposition $\perp$ into the path.
      - An event of type 3.

Results

The given controller achieves the two desired objectives (saliency and consistency).

Consistency preservation for the algorithms is proven by a semantic argument employing the soundness of the given axiomatization of first-order predicate logic.

Conclusion

Further examples involving subsumption and property inheritance with exceptions appear in the paper in review by ACM Transactions on Computational Logic (which can be viewed at arXiv CORR).

Other applications to be pursued include:

- Frame-based expert systems
- Description logics
    - Adding a temporal component
- Robot motion planning and learning
- Multi-agent systems
    - Agent-based modeling
    - Interactions involving belief and trust