Representations and Reasoning

Michael Rovatsos, University of Edinburgh
mrovatso@inf.ed.ac.uk

Representations & reasoning

- Any computation is a manipulation of symbols, in AI and related disciplines more ambitious goals
- Multiagent systems research uses an eclectic mix of representations and reasoning approaches
- What is the right formalism for my research? How do different ones relate to each other? Should I combine several of them?

Ad hoc formalisms

- Common approach in early days of agents research: invent your own formalism
- Imagine I want to reason about trust and reputation in a multiagent system
  - e.g. *trust(A,B,X)* – agent A trusts agent B with degree X
- These often look like logic but they are not, computationally just like a database (ground facts + negation)
- Central issue: Do you generate *new* (i.e. not explicitly stated) knowledge from existing knowledge?

Logic

- Next step: we can enable inferences in our formalism
  - *e.g.* $trust(A,B,X) \wedge trust(B,C,Y) \rightarrow trust(A,C,min\{X,Y\})$
- With proper semantics, we can develop sound and complete axiomatisations (doesn’t mean logic is useful)
- As long as there are only finitely many instances of this, no decidability problems
- But this means we would have to restrict ourselves to (exponentially large) propositional models

Logic

- Rule of thumb: if you want variables, you should stick to Horn logic to be safe
- Sometimes also fragments of harder languages tractable but then things get complicated (no simple answer)
- Hierarchical approches and abstraction useful in practice, but not in theory
- Modal logic: making states of a system (broadly speaking) implicit (but still –usually- propositional!)
  - □*trust(a,b,5)* translates to *trust(a,b,5,t)* and quantification over all pairs *(s,t)* in a relation over *S x S*

Uncertainty

- Add probabilities to propositional logic, and you get Bayesian networks (every formula now has a probability)
  - *trust(A,B,5)* => *trust(A,C,3)* has probability 0.43
- Description length exponentially larger, and you need all those numbers; first-order case pretty hopeless
- But think twice before you do uncertainty without probabilities, or using ad hoc formalisms
  - most principled approach to uncertainty, grounded in statistics and fundamental axioms, easily combined with learning

Preferences

- Capturing objective(s) of agent(s) (without preferences actually not much need for "agents")
  - *trust(A,B,5)* $>_A$ *trust(A,B,0)*
- Like probabilities, they impose another layer of meta-information over descriptions of states and their properties
- MDPs and POMDPs are like Bayesian nets with utilities, or probabilistic FSMs (cf. probabilistic model checking)

Preferences

- Add several agents, and you obtain normal-form (stochastic) games (and weird complexity issues)
- Remove probabilities and numerical utilities, and you get (multiagent) CP-nets
- Expressive languages for describing games much studied, usually “around” level of propositional logic
- Usual tradeoff: length of description versus complexity of queries (not much known about tradeoff in practice)

Generic problem solvers

- Many tasks motivated by agent issues often not distinct from other problems in other areas of AI
- Sometimes worrying: ultimately, when you dig down to the computational problem, you find it already exists
- We shouldn't reinvent the wheel, and also produce efficient implementations
- Specifically: off-the-shelf CSP solvers, planners, SAT solvers, POMDP solvers, theorem provers, model checkers, game solvers, machine learning algorithms

Summary

- How to choose right framework for representation and reasoning?
- There aren't that many choices really: state-transition systems, computational logic and its fragments, probability theory and utility theory

  ...and they're actually all kind of connected
- There is a big difference between theoretical and practical (not necessarily applied!) mode
  - do you ever want to *compute* non-trivial examples?
- Reuse, reuse, reuse

Personal view

Some of the things I’ve tried in the past:

- using neural networks to learn utility functions in large games, and genetic algorithms to evolve best-response strategies
- hierarchical reinforcement learning to optimise strategies in agent communication and argumentation
- doing social reasoning in a BDI-style architecture using on belief revision and model checking

Personal view

- using standard AI planners for argument generation in argumentation systems
- using planning technology to compute equilibrium plans among self-interested agents
- ...and plenty of ad hoc formalisms for opponent modelling, multiagent reinforcement learning, etc

Nice and fun to be eclectic, but not always rewarding.

Also, don’t forget you have to explain people this stuff!