COMP9414: Artificial Intelligence

Lecture 8b: Logic Applications

Wayne Wobcke

e-mail:w.wobcke@unsw.edu.au

This Lecture

- Ontologies
  - Taxonomies
  - Roles and Groups
- Reasoning About Action
  - Situation Calculus
  - Domain Constraints
  - Actions and the Frame Problem

Ontology

- General concepts and relationships between concepts
  - Subclass relationships
  - Part-whole relationships
  - Role relationships
- Knowledge base of facts about objects
  - Class membership
  - Equality of objects
  - Part-whole relationships
  - Integrity constraints

Example Ontology

AfPak Ontology

- Ashraf Ghani is President Ghani – equality
- Ashraf Ghani is the President of Afghanistan – role
- Ashraf Ghani is in the government – member of
- Nangarhar is a province – a kind of
- Nangarhar is in Afghanistan – part of
- Bombing implies Attacking – linguistic meaning/semantics

Taxonomies

- Type hierarchies *p is-a q*
  - $\forall x\,(p(x) \rightarrow q(x))$
  - $\forall x\,(hospital(x) \rightarrow building(x))$
- Transitivity
  - $\{\forall x\,(p(x) \rightarrow q(x)), \forall x\,(q(x) \rightarrow r(x))\} \vdash \forall x\,(p(x) \rightarrow r(x))$
- Part-whole relations *x part-of y*
  - $\forall x \forall y\,(location(e,\ x) \wedge part\text{-}of(x,\ y) \rightarrow location(e,\ y))$
- Transitivity
  - $\forall x \forall y \forall z\,(part\text{-}of(x,\ y) \wedge part\text{-}of(y,\ z) \rightarrow part\text{-}of(x,\ z))$

Roles and Groups

- Every country has one (and only one) President
  - $\forall c\,(country(c) \rightarrow \exists p\, president(c, p))$
  - $\forall c \forall p \forall q\,(president(c,\ p) \wedge president(c,\ q) \rightarrow p = q)$
  - $president(Afghanistan,\ Ashraf\ Ghani)$
- Quetta Shura is a subgroup of Taliban
  - $\forall x\,(member(x,\ Quetta\ Shura) \rightarrow member(x,\ Taliban))$
- Hibatullah is a member of Taliban
  - $member(Hibatullah, Taliban)$
- If the target is Hibatullah then the target is Taliban
  - $\forall x \forall y\,(target(e,\ x) \wedge member(x,\ y) \rightarrow target(e,\ y))$

Planning Agent

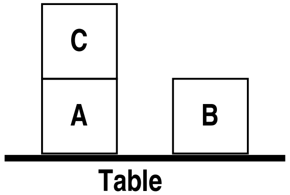
- Environment changes due to the performance of actions
- Planning scenario
  - Agent can control its environment
  - Only atomic actions, not processes with duration
  - Only single agent in the environment (no interference)
  - Only changes due to agent executing actions (no evolution)
- More complex examples
  - Robocup dog
  - Delivery robot
  - Self-driving car

Reasoning About Action

- Semantics: Divide the world into a sequence of (notional) time points
  - Situation is a (complete) state of world at a time point
  - Action is a transition between situations
  - Nothing (of relevance) happens between situations
- Planner: Maintain an incomplete description of situations
  - Confusingly, also called a state of the world
  - Search for path from initial state to a goal state
  - State transitions correspond to actions
  - Major problem is to specify actions

The Blocks World

- Blocks can be placed on the table and can be stacked on one another
- All blocks the same size and table large enough to hold all blocks



State: $on(C, A)$, $on(A, Table)$, $on(B, Table)$, $clear(B)$, $clear(C)$

Specifying Actions (STRIPS)

- Action Description – name of action
- Preconditions – action can be performed in a situation only if precondition holds in situation prior to action being performed
- Delete List – literals to be deleted from the state (description) after action is performed
- Add List – literals to be added to the state (description) after action is performed
- STRIPS Assumption – any literals in the state (description) not contained in the delete list remain the same after the action is performed (c.f. frame problem)

Assumes actions are executed perfectly (reasonable for planning?)

Blocks World Actions (STRIPS)

- Action Description: $move(x, y, z)$ $(x \neq y \neq z?)$
- Preconditions: $on(x, y)$, $clear(x)$, $clear(z)$
- Delete List: $clear(z)$, $on(x, y)$
- Add List: $on(x, z)$, $clear(y)$, $clear(Table)$
  - Add $clear(Table)$ to ensure table is always clear

Problems in Reasoning About Action

- Frame Problem
  - How to characterize what in the state does not change by performing an action
    - Problem is there are a lot of such facts
    - Both "epistemological" and "computational" problem
- Ramification Problem
  - What are the direct and indirect effects of performing an action?
    - Problem is that indirect effects depend on initial situation
- Qualification Problem
  - What preconditions are required in a specification of an action?
    - Problem is that qualifications depend on context

Situation Calculus

- First-order logic formalism for describing states and change
  - Situation is a (complete) state of world at a time point
- Reify situations: terms denote situations, e.g. $S_0, S_1, \cdots$
- Actions are arguments of a special "do" function
  - Situation $do(A, S)$ denotes the result of doing A in situation S
  - Assumes world is deterministic (but agent knowledge incomplete)
- Propositions (assertion that some fact holds in a situation)
  - Add situation argument to each predicate, or
  - Special "holds" predicate (predicates become functions)
- Axioms for domain constraints and performing actions

Blocks World States

- State – Here a description of what holds in a situation
  - More like the "state" of a planning agent
- Method 1: Add situation argument to each predicate

  $on(C, A, S_1), on(A, Table, S_1), on(B, Table, S_1)$
  $clear(B, S_1), clear(C, S_1)$

- Method 2: Special "holds" predicate (predicates become functions)

  $holds(on(C, A), S_1), holds(on(A, Table), S_1), holds(on(B, Table), S_1)$
  $holds(clear(B), S_1), holds(clear(C), S_1)$

Domain Constraints

- Also known as state constraints
- True at all (legal) states, though involve state-dependent relations
- Examples
  - x is on the table iff it is not on top of another block
    - $\forall x \forall s\,(on(x, Table, s) \leftrightarrow \neg\exists y\,(on(x, y, s) \wedge (y \neq Table)))$
  - x is clear iff there is no block on top of it
    - $\forall x \forall s\,(clear(x, s) \leftrightarrow \neg\exists y\, on(y, x, s))$
  - if y is a block and there is another block on it, then y is not clear
    - $\forall x \forall y \forall s\,(on(x, y, s) \wedge (y \neq Table) \rightarrow \neg clear(y, s))$

Actions

$do(A, S)$ – "Result of doing action A in situation S"

- Example: "move block x from y to z" and "clear y"
  - $\forall x \forall y \forall z \forall s\,(on(x, y, s) \wedge clear(x, s) \wedge clear(z, s) \wedge (x \neq z) \rightarrow on(x, z, do(move(x, y, z), s)))$
  - $\forall x \forall y \forall z \forall s\,(on(x, y, s) \wedge clear(x, s) \wedge clear(z, s) \wedge (x \neq z) \rightarrow \neg on(x, y, do(move(x, y, z), s)))$
  - $\forall x \forall y \forall z \forall s\,(on(x, y, s) \wedge clear(x, s) \wedge clear(z, s) \wedge (x \neq z) \wedge (y \neq z) \rightarrow clear(y, do(move(x, y, z), s)))$
  - $\forall x \forall y \forall z \forall s\,(on(x, y, s) \wedge clear(x, s) \wedge clear(z, s) \wedge (x \neq z) \wedge (z \neq Table) \rightarrow \neg clear(z, do(move(x, y, z), s)))$

The Frame Problem

- Previous action descriptions are not "complete"
  - They describe what changes
  - They do **not** specify what stays the same
- Proposal: Frame Axioms

$$on(x,\ y,\ s) \wedge (x \neq u) \rightarrow on(x,\ y,\ do(move(u,\ v,\ z),\ s))$$

$$\neg on(x,\ y,\ s) \wedge ((x \neq u) \vee (y \neq z)) \rightarrow \neg on(x,\ y,\ do(move(u,\ v,\ z),\ s))$$

$$clear(u,\ s) \wedge (u \neq z) \rightarrow clear(u,\ do(move(x,\ y,\ z),\ s))$$

$$\neg clear(u,\ s) \wedge (u \neq y) \rightarrow \neg clear(u,\ do(move(x,\ y,\ z),\ s))$$

Planning with the Situation Calculus

- To determine a plan to achieve goal $\Gamma(s)$, prove $\exists s\, \Gamma(s)$ using action theory and initial state
- For example, $\exists s\, on(B,\ Table,\ s)$ using

  $on(B,\ A,\ S_0)$
  $on(A,\ C,\ S_0)$
  $on(C,\ Table,\ S_0)$
  $clear(B,\ S_0)$
  $clear(Table,\ S_0)$
  $on(x,\ y,\ s) \wedge clear(x,\ s) \wedge clear(z,\ s) \wedge (x \neq z) \rightarrow on(x,\ z,\ do(move(x,\ y,\ z),\ s))$
- Also require equality axioms: $A = A$, $\neg(A = B)$, etc.
- Obtain $s = do(move(B,\ A,\ Table),\ S_0)$

Conclusion

- Ontologies
  - Description Logic more restrictive and efficient
  - Standard ontologies in various domains
  - Tools for ontology construction and maintenance
- Reasoning About Action
  - Interesting from philosophical point of view
  - Lack concise solution to the frame problem
  - Event calculus is another formalism more like Prolog

