Informatics 2D. Tutorial 5

Generalised Modus Ponens, Resolution, and Situation Calculus

Week 6

1 Generalised Modus Ponens

Part 1: Convert the following sentences to first-order logic formulae suitable for use with Generalised Modus Ponens.

1. Horses, cows and pigs are mammals.
2. An offspring of a horse is a horse.
3. Bluebeard is a horse.
4. Bluebeard is Charlie's parent.
5. Offspring and parent are inverse relations.

Part 2: Use the sentences to answer a query using a backward-chaining algorithm.

- Draw the proof tree generated by an exhaustive backward-chaining algorithm for the query $Horse(h)$, where clauses are matched in the order given.
- How many solutions are a logical consequence of your knowledge base?
- How could we solve this problem?

Answer

Part 1:

1. $Horse(x) \Rightarrow Mammal(x)$
   $Cow(x) \Rightarrow Mammal(x)$
   $Pig(x) \Rightarrow Mammal(x)$
2. $Offspring(y, x) \wedge Horse(x) \Rightarrow Horse(y)$ (y is offspring of x)
3. $Horse(Bluebeard)$
4. $Parent(Bluebeard, Charlie)$ (x is parent of y)

5. $Offspring(x, y) \Rightarrow Parent(y, x)$
   $Parent(x, y) \Rightarrow Offspring(y, x)$

Part 2:

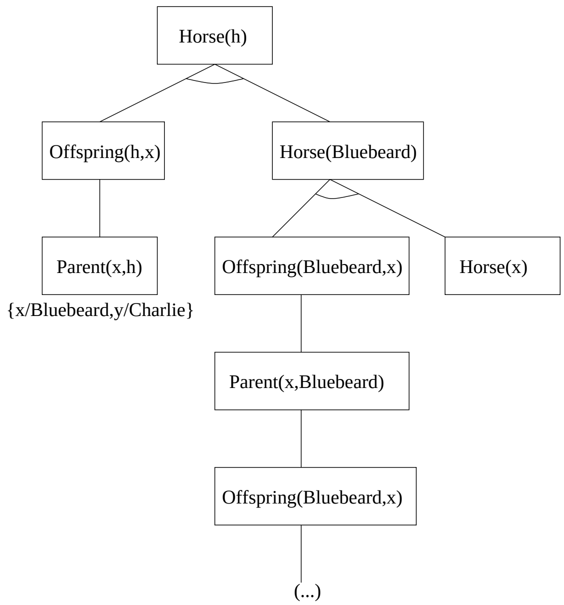

Figure 1: Solution to the Generalised Modus Ponens problem.

This question deals with the problem of looping in backward-chaining proofs. The proof tree is shown in Figure 1.

The branch with $OffSpring(Bluebeard, y)$ and $Parent(y, Bluebeard)$ repeats indefinitely, so the rest of the proof is never reached.

We get an infinite loop because of rule 2, $Offspring(x, y) \wedge Horse(y) \Rightarrow Horse(x)$.

The specific loop appearing in the figure arises because of the ordering of the clauses. We could be order $Horse(Bluebeard)$ before rule 2, which solve the problem of finding $Horse(Bluebeard)$ as a possible answer to the query.

2 Resolution

From "Horses are animals" it follows that "The head of a horse is the head of an animal". Demonstrate that this inference is valid by carrying out the following steps:

1. Translate the premise and the conclusion into the language of First-Order Logic. Use three predicates: $HeadOf(h, x)$ (meaning "h is the head of x"), $Horse(x)$, and $Animal(x)$.
2. Negate the conclusion, and convert the premise and the negated conclusion into Conjunctive Normal Form.
3. Use resolution to show that the conclusion follows from the premises.

Answer

(1):
$\forall x.Horse(x) \Rightarrow Animal(x)$
$\forall x, h.Horse(x) \wedge HeadOf(h, x) \Rightarrow \exists y.Animal(y) \wedge HeadOf(h, y)$

(2):
A: $\neg Horse(x) \vee Animal(x)$
B: $Horse(G)$
C: $HeadOf(H, G)$
D: $\neg Animal(y) \vee \neg HeadOf(H, y)$
Here A comes from the first sentence in (1). while the others come from the second. H and G are Skolem constants.

(3):
Resolve D and C to yield $\neg Animal(G)$. Resolve this with A (in (2) above to give $\neg Horse(G)$. Resolve this with B to obtain a contradiction.

3 Situation Calculus

Last week you learnt about the frame problem and you were shown how it can be fixed by adding frame axioms.

Consider the following predicates and functions:

1. $At(sq, s)$ means that the agent is at square sq in situation s.
2. $Heading(dir, s)$ means that the agent is facing in direction dir in situation s.
3. $Next(sq1, dir, sq2)$ means that square $sq2$ is adjacent to square $sq1$ in direction dir.
4. $Result(act, s)$ is the situation resulting from executing the action act in situation s.
5. $Turn(x)$ is the action of turning x where $x \in \{left, right\}$.
6. $Shoot$ is the action of shooting once forward.

7. $Newdir(dir1, x, dir2)$ means that $dir2$ is the new direction the agent will face if it is facing in direction $dir1$ and turns $x \in \{left, right\}$

8. $Wumpus(sq, s)$ means that that the Wumpus is in square sq in situation s.

In the following we assume that the action $Shoot$ only has an effect in directly adjacent squares.

a) Formalise a precondition and an effect axiom for the Wumpus World that best describes the action $Turn(x)$.

b) Formalise a precondition and an effect axiom that best describes the $Shoot$ action in the Wumpus World.

c) Formalise a frame axiom that best describes the $Shoot$ action in the Wumpus World. You only need to do this for the $Wumpus$ fluent.

Answer

Note that, the formalisation of actions using precondition and effect axioms differs from the lecture notes, where only one axiom is used. Alternative effect axioms, in the form used in the lecture notes, are given below.

- Preconditions decribe the fluents that must hold for an action to be possible.
- Effects describe the fluents that will hold as a result of taking the action.
- Frame axioms state what doesn't change as a result of taking an action

In the following axioms universal quantifiers (whose scope is the entire sentence) are omitted.

a) If the agent is heading in direction $dir1$ and the result of turning x is $dir2$ then the agent is heading in direction $dir2$ in the situation following after turning x.

- Precondition: $Heading(dir1, s) \wedge Newdir(dir1, x, dir2) \Rightarrow Poss(Turn(x), s)$
- Effect: $Poss(Turn(x), s) \Rightarrow Heading(dir2, Result(Turn(x), s))$

Alternatively:

$$Heading(dir1, s) \wedge Newdir(dir1, x, dir2) \Rightarrow Heading(dir2, Result(Turn(x), s))$$

b) If the agent is at square $sq1$ and heading in direction dir and the next square in direction dir is $sq2$ then the result of shooting will be that the wumpus is not in square $sq2$ (if it was there then it's dead).

- Precondition: $At(sq1, s) \wedge Heading(dir, s) \wedge Next(sq1, dir, sq2) \Rightarrow Poss(Shoot, s)$

- Effect: $Poss(Shoot, s) \Rightarrow \neg Wumpus(sq2, Result(Shoot, s))$

Alternatively:

$$At(sq1, s) \wedge Heading(dir, s) \wedge Next(sq1, dir, sq2) \Rightarrow \neg Wumpus(sq2, Result(Shoot, s))$$

c) If the agent is at square $sq1$ and heading in direction dir and the next square in direction dir is $sq2$ and the wumpus is in square $sq3$ and square $sq2$ doesn't equal $sq3$ then the Wumpus is still in square $sq3$ after shooting.

$$At(sq1, s) \wedge Heading(dir, s) \wedge Next(sq1, dir, sq2) \wedge Wumpus(sq3, s) \wedge sq2 \neq sq3 \Rightarrow Wumpus(sq3, result(Shoot, s))$$