Knowledge Representation and Reasoning

Exercise Sheet 10

Albert-Ludwigs-Universität Freiburg

Bernhard Nebel, Gregor Behnke, Thorsten Engesser, Rolf-David Bergdoll, Leonardo Mieschendahl, Johannes Herrmann

January 14th, 2022

Exercise 10.1

Answer Sets

Exercise 10.1 (Answer Sets, 4)

Find an answer set for the program Π_n consisting of the following n rules

$$p_i \leftarrow not\ p_{i+1}. \qquad (1 \le i < n)$$

where n is a natural number.

How we come up with an answer set X for Π_n:

- Π_n is stratified ($\Pi_n = \{p_{n-1} \leftarrow \text{not } p_n\} \cup \{p_{n-2} \leftarrow \text{not } p_{n-1}\} \cup \ldots$)
- p_n cannot be in X since it occurs only in the body of a rule.
- p_{n-1} must be in X because of $p_{n-1} \leftarrow \text{not } p_n$ and $p_n \notin X$.
- p_{n-2} cannot be in X because of $p_{n-2} \leftarrow \text{not } p_{n-1}$ and $p_{n-1} \in X$.
- ...

There is exactly one such answer set: $X = \begin{cases} \{p_1, p_3, \ldots, p_{n-1}\}: & n \text{ is even} \\ \{p_2, p_4, \ldots, p_{n-1}\}: & n \text{ is odd} \end{cases}$

Exercise 10.2

Answer Set Programming and Defaults

Exercise 10.2 (Answer Set Programming and Defaults, 2+2+2)

Reconsider the following knowledge base from exercise 9.2 (slightly condensed): *By default, students are not lazy. But computer science students are typically intelligent, and intelligent students are usually lazy. Anne and Bob study computer science.* Using Default Reasoning, the conclusion *Anne and Bob are lazy* follows credulously. Your task:

(a) Model the knowledge base as an Answer Set Program. Make use of the two versions of negation provided by the ASP language. Use clingo[1] to output all answer sets of your program.

(b) Choose one of the answer sets and check that it is indeed an answer set according to the definition from the lecture.

(c) Show that the answer sets correspond to the extensions of the default theoretical formalization.

Exercise 10.2 (a)

Answer Set Programming and Defaults – Modeling the Problem

The Description:

> *By default, students are not lazy. But computer science students are typically intelligent, and intelligent students are usually lazy. Anne and Bob study computer science.*

As Answer Set Program:

```
-lazy(X) :- stud(X), not lazy(X).
int(X) :- cs(X), not -int(X).
lazy(X) :- int(X), not -lazy(X).

stud(anne). stud(bob). cs(anne). cs(bob).
```

Exercise 10.2 (a)

Answer Set Programming and Defaults – Clingo Output

```
clingo version 5.4.0
Reading from students.asp
students.asp:2:22-29: info: atom does not occur in any rule head:
  (-int(X))

Solving...
Answer: 1
stud(anne) stud(bob) cs(anne) cs(bob) int(anne) int(bob) lazy(anne) lazy(bob)
Answer: 2
stud(anne) stud(bob) cs(anne) cs(bob) int(anne) int(bob) lazy(anne) -lazy(bob)
Answer: 3
stud(anne) stud(bob) cs(anne) cs(bob) int(anne) int(bob) -lazy(anne) lazy(bob)
Answer: 4
stud(anne) stud(bob) cs(anne) cs(bob) int(anne) int(bob) -lazy(anne) -lazy(bob)
SATISFIABLE

Models       : 4
Calls        : 1
Time         : 0.001s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.001s
```

Exercise 10.2 (b)

Answer Set Programming and Defaults – Checking the Answer Set

We verify the answer set $X = \{\mathit{stud}(\text{anne}), \mathit{stud}(\text{bob}), \mathit{cs}(\text{anne}), \mathit{cs}(\text{bob}), \mathit{int}(\text{anne}), \mathit{int}(\text{bob}), \mathit{lazy}(\text{anne}), \text{-}\mathit{lazy}(\text{bob})\}$:

$$
\begin{aligned}
P = \{\text{-}\mathit{lazy}(\text{anne}) &\leftarrow \mathit{stud}(\text{anne}), \text{not } \mathit{lazy}(\text{anne}) \\
\text{-}\mathit{lazy}(\text{bob}) &\leftarrow \mathit{stud}(\text{bob}), \text{not } \mathit{lazy}(\text{bob}) \\
\mathit{int}(\text{anne}) &\leftarrow \mathit{cs}(\text{anne}), \text{not } \text{-}\mathit{int}(\text{anne}) \\
\mathit{int}(\text{bob}) &\leftarrow \mathit{cs}(\text{bob}), \text{not } \text{-}\mathit{int}(\text{bob}) \\
\mathit{lazy}(\text{anne}) &\leftarrow \mathit{int}(\text{anne}), \text{not } \text{-}\mathit{lazy}(\text{anne}) \\
\mathit{lazy}(\text{bob}) &\leftarrow \mathit{int}(\text{bob}), \text{not } \text{-}\mathit{lazy}(\text{bob}) \\
&\leftarrow \mathit{int}(\text{anne}), \text{-}\mathit{int}(\text{anne}) \\
&\leftarrow \mathit{int}(\text{bob}), \text{-}\mathit{int}(\text{bob}) \\
&\leftarrow \mathit{lazy}(\text{anne}), \text{-}\mathit{lazy}(\text{anne}) \\
&\leftarrow \mathit{lazy}(\text{bob}), \text{-}\mathit{lazy}(\text{bob})\} \\
\cup\{\mathit{stud}(\text{anne}) \leftarrow, \mathit{stud}(\text{bob}) \leftarrow, &\mathit{cs}(\text{anne}) \leftarrow, \mathit{cs}(\text{bob}) \leftarrow\}
\end{aligned}
$$

Exercise 10.2 (b)

Answer Set Programming and Defaults – Checking the Answer Set

We verify the answer set $X = \{\mathit{stud}(\text{anne}), \mathit{stud}(\text{bob}), \mathit{cs}(\text{anne}), \mathit{cs}(\text{bob}), \mathit{int}(\text{anne}), \mathit{int}(\text{bob}), \mathit{lazy}(\text{anne}), -\mathit{lazy}(\text{bob})\}$:

$$\begin{aligned}
P = \{-\mathit{lazy}(\text{anne}) &\leftarrow \mathit{stud}(\text{anne}), \text{not } \mathit{lazy}(\text{anne})\\
-\mathit{lazy}(\text{bob}) &\leftarrow \mathit{stud}(\text{bob}), \text{not } \mathit{lazy}(\text{bob})\\
\mathit{int}(\text{anne}) &\leftarrow \mathit{cs}(\text{anne}), \text{not } -\mathit{int}(\text{anne})\\
\mathit{int}(\text{bob}) &\leftarrow \mathit{cs}(\text{bob}), \text{not } -\mathit{int}(\text{bob})\\
\mathit{lazy}(\text{anne}) &\leftarrow \mathit{int}(\text{anne}), \text{not } -\mathit{lazy}(\text{anne})\\
\mathit{lazy}(\text{bob}) &\leftarrow \mathit{int}(\text{bob}), \text{not } -\mathit{lazy}(\text{bob})\\
&\leftarrow \mathit{int}(\text{anne}), -\mathit{int}(\text{anne})\\
&\leftarrow \mathit{int}(\text{bob}), -\mathit{int}(\text{bob})\\
&\leftarrow \mathit{lazy}(\text{anne}), -\mathit{lazy}(\text{anne})\\
&\leftarrow \mathit{lazy}(\text{bob}), -\mathit{lazy}(\text{bob})\}\\
\cup\{\mathit{stud}(\text{anne}) \leftarrow, \mathit{stud}(\text{bob}) &\leftarrow, \mathit{cs}(\text{anne}) \leftarrow, \mathit{cs}(\text{bob}) \leftarrow\}
\end{aligned}$$

Exercise 10.2 (b)

Answer Set Programming and Defaults – Checking the Answer Set

We verify the answer set $X = \{stud(\text{anne}), stud(\text{bob}), cs(\text{anne}), cs(\text{bob}),$ $int(\text{anne}), int(\text{bob}), lazy(\text{anne}), -lazy(\text{bob})\}$:

$$\begin{aligned}
P^X = \{-lazy(\text{bob}) &\leftarrow stud(\text{bob}) \\
int(\text{anne}) &\leftarrow cs(\text{anne}) \\
int(\text{bob}) &\leftarrow cs(\text{bob}) \\
lazy(\text{anne}) &\leftarrow int(\text{anne}) \\
&\leftarrow int(\text{anne}), -int(\text{anne}) \\
&\leftarrow int(\text{bob}), -int(\text{bob}) \\
&\leftarrow lazy(\text{anne}), -lazy(\text{anne}) \\
&\leftarrow lazy(\text{bob}), -lazy(\text{bob})\} \\
\cup\{stud(\text{anne}) \leftarrow, stud(\text{bob}) &\leftarrow, cs(\text{anne}) \leftarrow, cs(\text{bob}) \leftarrow\}
\end{aligned}$$

$\Gamma^1 = \{stud(\text{anne}), stud(\text{bob}), cs(\text{anne}), cs(\text{bob})\}$

$\Gamma^2 = \{stud(\text{anne}), stud(\text{bob}), cs(\text{anne}), cs(\text{bob}), int(\text{anne}), int(\text{bob}), -lazy(\text{bob})\}$

$\Gamma^3 = \{stud(\text{anne}), stud(\text{bob}), cs(\text{anne}), cs(\text{bob}), int(\text{anne}), int(\text{bob}), -lazy(\text{bob}), lazy(\text{anne})\}$

$\Gamma^4 = \Gamma^3 = X$

Exercise 10.2 (c)

Answer Set Programming and Defaults – Default Theoretical Formalization

In Exercise 9.2 (b), we had the following default theory:

$$D = \left\{ \frac{stud(x) : \neg lazy(x)}{\neg lazy(x)}, \frac{cs(x) : int(x)}{int(x)}, \frac{int(x) \wedge stud(x) : lazy(x)}{lazy(x)} \right\}$$

$$W = \{stud(\text{anne}), stud(\text{bob}), stud(\text{jim}), stud(\text{mary}), cs(\text{anne}), cs(\text{bob})\}$$

Ignoring Jim and Mary, we have:

$$W' = \{stud(\text{anne}), stud(\text{bob}), cs(\text{anne}), cs(\text{bob})\}$$

We obtain the following extensions (analogous to our solution for Ex. 9.2):

- $Th(W' \cup \{int(\text{anne}), int(\text{bob}), \neg lazy(\text{anne}), \neg lazy(\text{bob})\})$
- $Th(W' \cup \{int(\text{anne}), int(\text{bob}), lazy(\text{anne}), \neg lazy(\text{bob})\})$
- $Th(W' \cup \{int(\text{anne}), int(\text{bob}), \neg lazy(\text{anne}), lazy(\text{bob})\})$
- $Th(W' \cup \{int(\text{anne}), int(\text{bob}), lazy(\text{anne}), lazy(\text{bob})\})$

We can see that they correspond to the four answer sets found earlier.

Exercise 10.3

Alternative Characterization of Answer Sets

Exercise 10.3 (Alternative Characterization of Answer Sets, 2+2+2)

For the purpose of this exercise, we assume that the rules $a \leftarrow b_1, \dots, b_m, \text{not } c_1, \dots, \text{not } c_k$ of a normal logic program can also be written as propositional formulas $a \leftarrow b_1 \land \dots \land b_m \land \neg c_1 \land \dots \land \neg c_k$ and that a grounded normal logic program thus corresponds to a set of propositional formulas.

Let P be a grounded normal logic program and M be a (propositional) model of P. We say an atom $a \in \mathit{atoms}(P)$ is *supported* if P contains a rule $a \leftarrow \phi$ such that $M \models \phi$. Furthermore, we say that a is *founded* if it can be derived from the reduct P^M via a (possibly empty) sequence of rule applications (using modus ponens). One can show that answer sets are exact those models M of P such that all atoms in M are supported and founded (however, we won't do this in this exercise). Instead of using the fixpoint algorithm from the lecture, this allows us to find answer sets by reduction to propositional SAT. For this, we define the *completion* of a grounded normal logic program P as follows (we assume facts $a \leftarrow$ are always written as $a \leftarrow \top$ and constraints $\leftarrow \phi$ are always written as $\bot \leftarrow \phi$):

$$\mathit{Comp}(P) = \left\{ a \leftrightarrow \bigvee_{a \leftarrow \phi \;\in\; P} \phi \;\middle|\; a \in \mathit{atoms}(P) \cup \{\bot\} \right\}$$

(a) Show that M is a model for the completion $\mathit{Comp}(P)$ if and only if M is a model for P and all atoms $a \in M$ are supported.

(b) Use an example to show that models for $\mathit{Comp}(P)$ can still contain unfounded atoms.

(c) Look up *loop formulas*.[2] Show for your example that the set of formulas $\mathit{Comp}(P) \cup \mathit{LF}(P)$ which you obtain by applying the completion and adding all loop formulas has only models in which all atoms are supported and founded. Use the fixpoint algorithm from the lecture to show that these models are indeed answer sets.

Exercise 10.3 (a)

Alternative Characterization of Answer Sets

$$Comp(P) = \left\{ a \leftrightarrow \bigvee_{a \leftarrow \phi \, \in \, P} \phi \;\middle|\; a \in atoms(P) \cup \{\bot\} \right\}$$

Example:

$P = \{a \leftarrow \neg b, b \leftarrow \neg a\}$ has the propositional interpretations $\{a\}$, $\{b\}$, and $\{a, b\}$. Since a and b are not supported in $\{a, b\}$, it is not an answer set. $Comp(p) = \{a \leftrightarrow \neg b, b \leftrightarrow \neg a, \bot \leftrightarrow \bot\}$ has the interpretations $\{a\}$ and $\{b\}$, the atoms of which are supported and founded $\rightarrow$ they are answer sets.

Observation:

Obviously, $Comp(P)$ is equivalent to

$$\left\{ a \leftarrow \bigvee_{a \leftarrow \phi \, \in \, P} \phi \;\middle|\; a \in atoms(P) \cup \{\bot\} \right\} \cup \left\{ a \rightarrow \bigvee_{a \leftarrow \phi \, \in \, P} \phi \;\middle|\; a \in atoms(P) \cup \{\bot\} \right\},$$

which is equivalent to $P \cup \left\{ a \rightarrow \bigvee_{a \leftarrow \phi \, \in \, P} \phi \;\middle|\; a \in atoms(P) \right\}$.

Exercise 10.3 (a)

Alternative Characterization of Answer Sets

$$Comp'(P) = P \cup \left\{ a \rightarrow \bigvee_{a \leftarrow \phi \,\in\, P} \phi \;\middle|\; a \in atoms(P) \right\},$$

$M \models Comp'(P) \Rightarrow M \models P$ and all $a \in M$ are supported

- Let $M \models Comp'(P)$.
- Since $P \subseteq Comp'(P)$, it follows that $M \models P$.

- We now show by contradiction that all $a \in M$ are supported:
- Let $a \in M$ be unsupported, i.e., there is no $a \leftarrow \phi \in P$ such that $M \models \phi$.
- Then $M \not\models \bigvee_{a \leftarrow \phi \,\in\, P} \phi$, and with $a \in M$, we have $M \not\models a \rightarrow \bigvee_{a \leftarrow \phi \,\in\, P} \phi$.
- This contradicts our assumption that $M \models Comp'(P)$. $\square$

Exercise 10.3 (a)

Alternative Characterization of Answer Sets

$$Comp'(P) = P \cup \left\{ a \rightarrow \bigvee_{a \leftarrow \phi \,\in\, P} \phi \;\middle|\; a \in atoms(P) \right\},$$

$M \models Comp'(P) \Leftarrow M \models P$ and all $a \in M$ are supported

- Let all $a \in M$ be supported, i.e., there is an $a \leftarrow \phi \in P$ such that $M \models \phi$.
- Then it also holds that $M \models \left\{ a \rightarrow \bigvee_{a \leftarrow \phi \,\in\, P} \phi \;\middle|\; a \in atoms(P) \right\}$.
- With $M \models P$, we have $M \models Comp'(P)$. □

Exercise 10.3 (b)

Alternative Characterization of Answer Sets

$$Comp(P) = \left\{ a \leftrightarrow \bigvee_{a \leftarrow \phi \,\in\, P} \phi \;\middle|\; a \in atoms(P) \cup \{\bot\} \right\}$$

Example:

$P = \{a \leftarrow b, a \leftarrow c, b \leftarrow a\}$ has the propositional models $\{a,b,c\}$, $\{a,b\}$ and $\emptyset$. Atom c is unsupported for $\{a,b,c\}$. Atoms a and b are supported for $\{a,b\}$.

$Comp(P) = \{a \leftrightarrow b \vee c, b \leftrightarrow a, c \leftrightarrow \bot, \bot \leftrightarrow \bot\}$ has the propositional models $\{a,b\}$ and $\emptyset$. There are no more models with unsupported atoms.

However, while both atoms in $\{a,b\}$ are supported, they are not founded.

We can verify that $X = \emptyset$ is the only answer set for P with $P^X = P$, $\Gamma^0 = \emptyset$ and $\Gamma^1 = \emptyset = \Gamma^0 = X$.

Exercise 10.3 (c)

Alternative Characterization of Answer Sets

Example with loop formulas:

We looked at the program $P = \{a \leftarrow b, a \leftarrow c, b \leftarrow a\}$ and its completion $Comp(P) = \{a \leftrightarrow b \vee c, b \leftrightarrow a, c \leftrightarrow \bot, \bot \leftrightarrow \bot\}$

The program P contains only the loop $L = \{a, b\}$.

The external supports are $ES_P(L) = \{a \leftarrow c\}$.

We thus get the loop formula $LF_P(L) = (a \vee b) \rightarrow c \equiv \neg c \rightarrow (\neg a \wedge \neg b)$.

And: $Comp(P) \cup LF(P) \equiv \{a \leftrightarrow b \vee c, b \leftrightarrow a, c \leftrightarrow \bot, \neg c \rightarrow (\neg a \wedge \neg b)\}$

The only remaining model for $Comp(P) \cup LF(P)$ is the answer set $\emptyset$.

Even simpler example:

For $P' = \{a \leftarrow b, b \leftarrow a\}$ we would get the loop formula $\neg\bot \rightarrow (\neg a \wedge \neg b)$.

Again, the only model for $Comp(P') \cup LF(P')$ is the answer set $\emptyset$.