Problem 4.1
Consider the knowledge base

\[ F = \{ \text{interesting-food } \leftrightarrow \text{dessert } \lor \text{spinach-pilaf} \}
\]
\[ \text{dessert } \leftrightarrow \text{magic-cookie-bars } \lor \text{banana-burrito} \}

the set of abducibles

\[ F_A = \{ \text{spinach-pilaf} , \text{magic-cookie-bars} , \text{banana-burrito} \}. \]

and an empty set of integrity constraints. Compute the set of possible explanations for the observation "interesting-food"

- by using SLD–resolution, and
- by model generation.

Solution

- by SLD-resolution:

\[ F_L = \{ 1) \text{interesting-food } \leftarrow \text{dessert}, 
2) \text{interesting-food } \leftarrow \text{spinach-pilaf}, 
3) \text{dessert } \leftarrow \text{magic-cookie-bars}, 
4) \text{dessert } \leftarrow \text{banana-burrito} \} \]

SLD search space:

Negation of leafs yields possible explanations:

\{\text{magic-cookie-bars}\}
\{\text{banana-burrito}\}
\{\text{spinach-pilaf}\}
• by model generation:

\[ \mathcal{F} \rightarrow \cup \{ \text{interesting-food} \} = \{
\begin{align*}
1) & \quad \text{interesting-food} \rightarrow \text{dessert} \lor \text{spinach-pilaf}, \\
2) & \quad \text{dessert} \rightarrow \text{magic-cookie-bars} \lor \text{banana-burrito}, \\
3) & \quad \text{interesting-food}
\end{align*}\]

The minimal models of this set are:

\{\text{interesting-food, spinach-pilaf}\}
\{\text{interesting-food, dessert, magic-cookie-bars}\}
\{\text{interesting-food, dessert, banana-burrito}\}

Restricting these models to abducible predicates yields the following explanations:

\{\text{spinach-pilaf}\}
\{\text{magic-cookie-bars}\}
\{\text{banana-burrito}\}

Problem 4.2

Specify an abductive framework \( \langle \mathcal{F}, \mathcal{F}_A, I \rangle \) and an observation \( G \), such that the observation can be explained according to the satisfiability view in a way that is not available by the theoremhood view.

Solution

For example:

\[ \langle \mathcal{F}, \mathcal{F}_A, I \rangle \quad \text{with} \]
\[ \mathcal{F} = \{ b \leftarrow a \lor c \} \]
\[ \mathcal{F}_A = \{ a, c \} \]
\[ I = \{ a \rightarrow c \} \]
\[ G = b \]

• In the satisfiability view, \( \{a\} \) is an explanation, because

\[ \mathcal{F} \cup \{a\} \models b \]

and \( \mathcal{F} \cup \{a\} \cup I \) is satisfiable

(by the interpretation \( \{a, b, c\} \))

• In the theoremhood view, \( \{a\} \) is not an explanation, because

\[ \mathcal{F} \cup \{a\} \not\models I \]

\( (\{a, b\} \) is a model for \( \mathcal{F} \cup \{a\} \), but no model for \( I \))
Problem 4.3
Assume that you have the data structure char of ASCII characters available.

1. Define the data structure string according to the following specification:
   A string may be empty or may be obtained by adding an ASCII character to the end of a string.
   Which are the constructors? Which are the selectors?

2. Express explicitly the following conditions that the data structure string should satisfy:
   (a) Different constructors produce different objects;
   (b) Constructors of arity > 0 induce injective mappings on the set of constructor ground terms;
   (c) Each constructor ground term can be represented as an application of some constructor to the results of application of selectors, if any applicable selectors exists;
   (d) Each selector is ‘inverse’ to the constructor it belongs to;

3. Write a program \( \mathcal{F}_{\text{Trans}} \) that defines the function Trans over non-empty strings, which transforms any string into a string of the same length containing only the character ‘a’.

Solution

1. \( \Lambda : \text{string} \)
   \( \circ : \text{string} \times \text{char} \rightarrow \text{string} \) ] constructors
   \( h : \text{string} \rightarrow \text{string} \) ] selectors (= Initial part)
   \( t : \text{string} \rightarrow \text{char} \) (= tail element)
   ( \( \circ \) will be written infix)

2. (a) (\( \forall S : \text{string}, C : \text{char} \) (\( \Lambda \neq S \circ C \))

   (b) (\( \forall S, S' : \text{string}, C, C' : \text{char} \) (\( S \circ C \approx S' \circ C' \rightarrow (S \approx S' \land C \approx C') \))

   (c) (\( \forall S : \text{string} \) (\( S \approx \Lambda \lor S \approx h(S) \circ t(S) \))

   (d) (\( \forall S : \text{string}, C : \text{char} \) (\( h(S \circ C) \approx S \))

   (\( \forall S : \text{string}, C : \text{char} \) (\( t(S \circ C) \approx C \))

3. \( \mathcal{F}_{\text{Trans}} = \{ (\forall X : \text{string}) (\text{Trans}(X) \approx \Lambda \leftarrow X \approx \Lambda), (\forall X : \text{string}) (\text{Trans}(X) \approx \text{Trans}(h(X)) \circ 'a' \leftarrow X \neq \Lambda) \} \)

   (\( \text{Trans}(\Lambda) \approx \Lambda \) is ok, too)