EXERCISE 4

Science of Computational Logic

Steffen Hölldobler, Marcos Cramer

International Masters Programme in Computational Logic — winter semester 2018/2019

17.01.2019

Problem 4.1

Consider the knowledge base

$$\mathcal{F} = \left\{ \begin{array}{rcl} \texttt{interesting-food} & \leftrightarrow & \texttt{dessert} \lor \texttt{spinach-pilaf} \\ \texttt{dessert} & \leftrightarrow & \texttt{magic-cookie-bars} \lor \texttt{banana-burrito} \end{array} \right\}$$

the set of abducibles

$$\mathcal{F}_A = \{ \texttt{spinach-pilaf}, \texttt{magic-cookie-bars} , \texttt{banana-burrito} \}.$$

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.

Problem 4.2

Specify an abductive framework $\langle \mathcal{F}, \mathcal{F}_A, \mathcal{I} \rangle$ 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.

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 terms 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 $\mathcal{F}_{Trans}$ 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'.