Multi-Agent Systems

B. Nebel, F. Lindner, T. Engesser
Summer Semester 2017

University of Freiburg
Department of Computer Science

Exercise Sheet 8

Due: July 10th, 2017, 10:00

Exercise 8.1 (Distributed Constraint Satisfaction, 2+4)
Consider the following distributed constraint satisfaction problem $\mathcal{P} = (A, X, C, D)$ with

$$\begin{aligned} A &= (0, 1, 2, 3) \\ X &= (x_0, x_1, x_2, x_3) \\ D &= (\{1, 2, 3\}, \{1, 3\}, \{2, 4\}, \{3, 4\}) \\ C &= (x_0 > x_1, x_0 \neq x_2, x_0 \neq x_3, x_2 \neq x_3). \end{aligned}$$

Remember that A is the list of agents, X is the list of variables s.th. X_i is the variable of agent A_i, D is the list of domains s.th. D_i is the domain of variable X_i, and C is the list of constraints.

(a) Build a communication graph for the agents that is appropriate for the application within the asynchronous backtracking search.

(b) Solve the CSP using asynchronous backtracking. Your submission should be a chronological log containing all sent and received messages for all of the agents as well as all corresponding changes of the agent's variables `current_value`, `agent_view`, and `constraint_list`. Your solution should contain nogood messages for at least two of the agents.

Exercise 8.2 (Admissible Discussions, 2+1+2)
We want to write a program that reads a single argumentation framework from a JSON specification file and decides for one given argument whether or not it is part of the **in**-set of some preferred labeling. The JSON object with which we represent an argumentation framework is a single dictionary where the keys are exactly the (names of the) arguments in the framework. The value assigned to each key is a list of exactly the attacked arguments. Both the filename of the JSON specification file and the name of the argument a for which the admissible discussions is to be performed should be passed to your program as command line parameters. The program should then write the following onto the standard output:

(a) all possible admissible discussions starting with `in(`a`)`, each on its own line,

(b) the winner of each discussion in brackets, at the end of the respective line, as well as

(c) one final line stating whether a `is in for (some|no) preferred labeling`.

Consider the following example, where the discussion framework specified in `df.json` is the JSON object `{"a": ["b"], "b": ["c", "d"], "c": ["d", "e"], "d": ["c", "e"], "e": []}` and the argument of interest is `d`. A call of `python3 discuss.py df.json d` could yield the following output:

```
in(d), out(b), in(a), out(c), in(b) [S]
in(d), out(b), in(a), out(c), in(d) [M]
in(d), out(c), in(b), out(b) [S]
in(d), out(c), in(b), out(a) [S]
in(d), out(c), in(d), out(b), in(a) [M]
d is in for some preferred labeling
```

Exercise 8.3 (CSPs and Admissible Discussions, 2+2)

(a) Generate a JSON specification file for the argumentation framework representing the constraint satisfaction problem from Exercise 8.1.

(b) Use your implementation and identify an admissible discussion in which M wins and which contains an assignment for each of the variables. *Hint: You might have to try different initial in-arguments.*