## Principles of AI Planning

Prof. Dr. B. Nebel, Dr. R. Mattmüller
D. Speck, D. Drexler
Winter Semester 2018/2019

University of Freiburg
Department of Computer Science

# Exercise Sheet 14

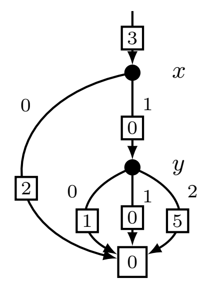
**Due: No submission – no grading – optional exercises**

**Exercise 14.1** (Planning literature)

As part of your work on exercise sheet 1, you read the paper titled "Everything You Always Wanted to Know About Planning (But Were Afraid to Ask)" by Jörg Hoffmann, published at the annual German Conference on Artificial Intelligence (KI) in 2011.
Back then, you were asked to write, as an answer to the exercise, two questions that came to your mind when reading the paper and that you would like to discuss in the exercise group.
Re-read the paper (however cursorily) and compare your impression of it you had three to four months ago and now. Does the paper give you a good overview of what material we discussed in class?
The paper can be found here: `http://fai.cs.uni-saarland.de/hoffmann/papers/ki11.pdf`

**Exercise 14.2** (Evaluating relaxed states with EVMDDs)

Consider a cost function represented by the EVMDD on the right.




(a) Let $s$ be a state with $s(x) = 1$ and $s(y) = 2$. To which value does the EVMDD evaluate for state $s$?
(b) Let $s^+$ be a relaxed state containing the facts $(x, 0)$ and $(x, 1)$ for $x$ and $(y, 0)$ and $(y, 2)$ for $y$. To which value does the EVMDD evaluate for $s^+$? Show how you computed your result by annotating the nodes of the EVMDD appropriately.

**Exercise 14.3** (EVMDDs)

An EVMDD is called *reduced* if it contains no two isomorphic subgraphs and if it contains no nodes where all outgoing edges lead to the same successor node and carry the same weight. It is *canonical* if for each node, the minimal outgoing edge weight is zero.

(a) Let $c_1 = xy + y$ for variables $x, y$ with $\mathcal{D}_x = \mathcal{D}_y = \{0, 1\}$. Draw the canonical reduced ordered EVMDDs for $c_1$ for both possible variable orders ($x, y$ and $y, x$). Compare their sizes.
(b) Let $c_2 = x \cdot (2 + y + z) - u^2 + 7$ for variables $x, y, z, u$ with $\mathcal{D}_x = \mathcal{D}_y = \mathcal{D}_z = \{0, 1\}$ and $\mathcal{D}_u = \{0, 1, 2\}$. Draw the canonical reduced ordered EVMDD for $c_2$ and variable order $x, y, z, u$.

**Exercise 14.4** (Strong stubborn sets)

Consider the SAS$^+$ planning task $\Pi$ with variables $V = \{\mathit{pos}, \mathit{left}, \mathit{right}, \mathit{hat}\}$, $\mathcal{D}_{\mathit{pos}} = \{\mathit{home}, \mathit{uni}\}$ and $\mathcal{D}_{\mathit{left}} = \mathcal{D}_{\mathit{right}} = \mathcal{D}_{\mathit{hat}} = \{t, f\}$. The initial state is $I = \{\mathit{pos} \mapsto \mathit{home}, \mathit{left} \mapsto f, \mathit{right} \mapsto f, \mathit{hat} \mapsto f\}$ and the goal specification is $\gamma = \{\mathit{pos} \mapsto \mathit{uni}\}$. There are four operators in $O$, namely

$$\begin{aligned}
\mathit{wear\text{-}left\text{-}shoe} &= \langle \mathit{pos} = \mathit{home} \land \mathit{left} = f, \mathit{left} := t \rangle, \\
\mathit{wear\text{-}right\text{-}shoe} &= \langle \mathit{pos} = \mathit{home} \land \mathit{right} = f, \mathit{right} := t \rangle, \\
\mathit{wear\text{-}hat} &= \langle \mathit{pos} = \mathit{home} \land \mathit{hat} = f, \mathit{hat} := t \rangle, \text{ and} \\
\mathit{go\text{-}to\text{-}university} &= \langle \mathit{pos} = \mathit{home} \land \mathit{left} = t \land \mathit{right} = t, \mathit{pos} := \mathit{uni} \rangle.
\end{aligned}$$

(a) Draw the breadth-first search graph (with duplicate detection) for planning task $\Pi$ without any form of partial-order reduction.

(b) Draw the breadth-first search graph (with duplicate detection) for planning task $\Pi$ using strong stubborn set pruning. For each expansion of a node for a state $s$, specify in detail how $T_s$ (and thus $T_{app(s)}$) are computed, i.e., explain how the initial disjunctive action landmark is chosen and how operators are iteratively added to $T_s$ as part of necessary enabling sets or interfering operators, respectively. Break ties in favor of *wear-left-shoe* over *wear-right-shoe*.

How many node expansion do you save with strong stubborn sets compared to plain breadth-first search? What about the lengths of the resulting solutions?

You may abbreviate the operator names, state descriptions etc. appropriately.