Automated Planning (TDDD48)

Jendrik Seipp
Mika Skjelnes

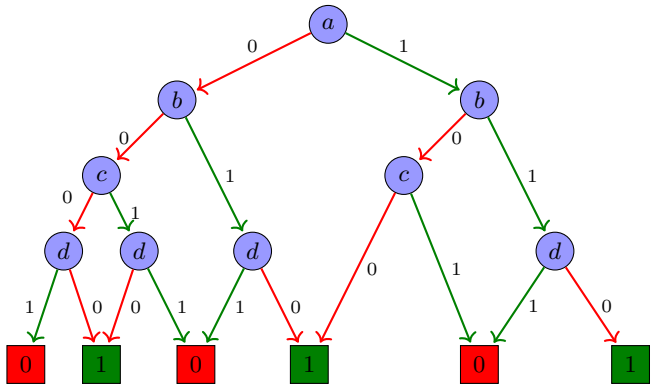
Linköping University

Lab 3

Important: For submission, consult the rules at the end of this document. Non-adherence to these rules might lead to a penalty in the form of a deduction of points. Some points are *bonus points*. These can help you reach the point quota per lab (4/12 points) and the overall point quota ($50\% \cdot 7 \cdot 12 = 42$ points).

Bonus Exercise 3.1 (2+1=3 bonus points)
For both parts, you do not need to show intermediate results, but partial points may be awarded for wrong results with correct intermediate steps.

(a) Consider the following ordered BDD:



Provide the equivalent reduced ordered BDD.

(b) Provide a reduced ordered BDD for the formula

$$\varphi = (a \vee b) \rightarrow (d \vee ((c \vee d) \rightarrow (\neg a \wedge b)))$$

with order $a \prec b \prec c \prec d$.

Exercise 3.2 (1+1+2+2 points)
Let $\Pi = \langle V, I, O, \gamma \rangle$ be a propositional planning task with the following components.

- $V = \{v_1, v_2, v_3\}$
- $I(v_1) = 1$, $I(v_2) = 0$ and $I(v_3) = 0$
- $O = \{o_1, o_2\}$
  - $o_1 = \langle v_1, v_2 \wedge \neg v_3 \rangle$
  - $o_2 = \langle v_2, \neg v_2 \wedge v_3 \rangle$
- $\gamma = v_3$

Furthermore, we assume a variable order $v_1 \prec v_1' \prec v_2 \prec v_2' \prec v_3 \prec v_3'$. Your task is to execute symbolic forward breadth-first search on this planning task by hand.

(a) Draw the reduced ordered BDD for γ, i.e., bdd-formula(γ).

(b) Draw the reduced ordered BDD for I, i.e., bdd-singleton(I).

(c) Draw the reduced ordered BDDs for o_1 and o_2, i.e., bdd-formula($\tau_V(o_1)$) and bdd-formula($\tau_V(o_2)$).

(d) Describe the steps that symbolic forward breadth-first search will execute for the symbolic representation of planning task Π created in steps (a)-(c). In your description, specify the *formulas* (not the BDDs themselves) represented by the BDDs $reached_0$, $reached_1$ and $reached_2$.

Exercise 3.3 (4.5+1.5 points)

Update the course repository (`/vagrant/tddd48` in your course VM) with `git pull`. Navigate to the new directory `lab3` which contains the files required for this exercise.

Pyperplan (`https://github.com/aibasel/pyperplan`) is a lightweight STRIPS planner written in Python. While it is not as fast as Fast Downward, it is very easy to extend and modify.

(a) In the file `pyperplan/src/search/bdd_bfs.py` you find an incomplete implementation of a BDD-based breadth-first search. Complete it by using the utility methods in the file `pyperplan/src/search/bdd.py`. Do not modify anything else than the file `pyperplan/src/search/bdd_bfs.py` (and do not modify the constructor of `BDDSearch` yet, this is for part (b)). Test your search on the tasks in the directory `blocks` and make sure that it can find valid plans.

You can run the code with the command
`./pyperplan/src/pyperplan.py -s bdd blocks/domain.pddl blocks/p1.pddl`

(b) The constructor of `BDDSearch` contains a commented out alternative variable order for the variables within the BDD. Change the order by commenting out the old order and including the new order instead. Print the number of total BDD nodes after adding each operator and after each expansion step (use the provided method `print_bdd_nodes()`). Compare the two variable orders on a small task and discuss the results.

Submission rules:

- Lab sheets must be submitted in groups of 2–3 students. Clone the labs repo (`https://github.com/mrlab-ai/tddd48-labs`) and push it to a repo at the University GitLab instance `https://gitlab.liu.se`. Make sure the repo is **private** and give read access to Mika Skjelnes (`mika.skjelnes@liu.se`).
- For non-programming exercises, create a single PDF file at the location `labX/solution.pdf`. If you want to submit handwritten solutions, include their scans in the single PDF. Make sure it is in a reasonable resolution so that it is readable. Put the names of all group members on top of the first page. Either use page numbers on all pages or put your names on each page. Make sure your PDF has size A4 (fits the page size if printed on A4).
- For programming exercises, directly edit the code in the cloned repository and only create those code text file(s) required by the lab. Put your names in a comment on top of each file. Make sure your code compiles and test it. Code that does not compile or which we cannot successfully execute will not be graded.