Universität Zürich UZH

Introduction to Artificial Intelligence
Exercise Sheet 6

Laurin van den Bergh, 16-744-401

Yufeng Xiao, 19-763-663

Nora Beringer, 19-734-227

Universität Zürich

Institut für Informatik

Due: April 06, 2022

Exercise 6.1

a) We first brought φ into CNF in order to make e decision about the propositional formula in Task 6.1 a). Please consider the justification down below.

b)
$$\varphi = \neg(A \to (B \land \neg C)) \land ((A \land C) \lor \neg B)$$
$$\varphi \equiv \neg(\neg A \lor (B \land \neg C)) \land ((A \land C) \lor \neg B) \qquad \text{Implication eliminiation}$$
$$\varphi \equiv A \land \neg(B \land \neg C) \land ((A \land C) \lor \neg B) \qquad \text{DeMorgan Law, Double Negation eliminiation}$$
$$\varphi \equiv A \land \neg(B \land \neg C) \land (A \lor \neg B) \land (C \lor \neg B) \qquad \text{Distribution Law}$$
$$\varphi \equiv A \land (\neg B \lor C) \land (A \lor \neg B) \land (C \lor \neg B) \qquad \text{DeMorgan Law}$$
$$\varphi \equiv A \land (A \lor \neg B) \land (C \lor \neg B) \qquad \text{remove duplicate sentence}$$

Justification for a):

Set A to true and C to true then φ is satisfiable, hence it is not unsatisfiable.

Set A to false then φ is not valid.

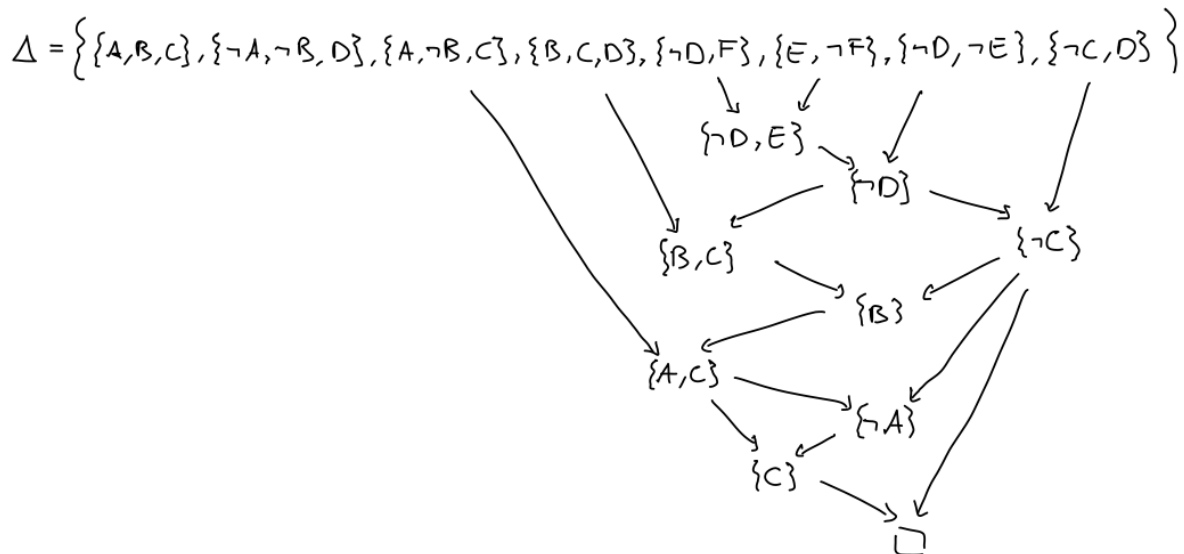Because the sentence φ is satisfiable and not valid it is falsifiable.

Exercise 6.2

a)

We want to find out if $\varphi \vDash \psi$ holds, where $\psi = C \land \neq D, \neq \psi \equiv \neq C \lor D$

To determine this, we can test if $\varphi \to \psi$ is a tautology,

equivalently: test if $\psi \cup \{\neq \psi\} =: \Delta$ is unsatisfiable, where Δ is a set of clauses.

Consider the Resolution steps down below:

We conclude that Δ is unsatisfiable and therefore $\varphi \vDash \psi$ holds.

   b) We used 9 steps. A truth table for the same sentence would have 2^6 rows as the number of variables (A, B, C, D, E, F) is equal to 6. Therefore we need less steps for the

Resolution in comparison tho building the truth table.

Exercise 6.3

a)

$\Delta = \{\{A,B\}, \{\neg A,B\}, \{\neg B,C\}, \{\neg C,\neg D\}, \{\neg C, D, E, \neg F\}, \{\neg B, \neg E, \neg F\}\}$
pure symbol heuristics : $F \rightarrow F$

$\{\{A,B\}, \{\neg A,B\}, \{\neg B,C\}, \{\neg C,\neg D\}\}$, splitting on variable A

1a) $A \rightarrow \top$
$\{\{B\}, \{\neg B,C\}, \{\neg C,\neg D\}\}$
unit clause heuristics : $B \rightarrow \top$
$\{\{C\}, \{\neg C,\neg D\}\}$
unit clause heuristics : $C \rightarrow \top$
$\{\{\neg D\}\}$
unit clause heuristics : $D \rightarrow F$
$\{\}$

1b) $A \rightarrow F$
$\{\{B\}, \{\neg B,C\}, \{\neg C,\neg D\}\}$
unit clause heuristics : $B \rightarrow \top$
$\{\{C\}, \{\neg C,\neg D\}\}$
unit clause heuristics : $C \rightarrow \top$
$\{\{\neg D\}\}$
unit clause heuristics : $D \rightarrow F$
$\{\}$

Exercise 6.4

a) As we didn't find a pure symbol or unit clause heuristic we choose to split on A, while assigning True first for splitting. We then break ties in alphabetical order.
Leading to the following recursive structure of 1a):

$\Delta = \{\{A, B\}, \{\neg A, B\}, \{\neg B, C\}, \{\neg C, \neg D\}, \{\neg C, D, E\}, \{\neg B, \neg E\}\}$

1a) splitting on variable A: A→T

$\{\{B\}, \{\neg B, C\}, \{\neg C, \neg D\}, \{\neg C, D, E\}, \{\neg B, \neg E\}\}$

unit clause heuristics : B→T

$\{\{C\}, \{\neg C, \neg D\}, \{\neg C, D, E\}, \{\neg E\}\}$

unit clause heuristics : C→T

$\{\{\neg D\}, \{D, E\}, \{\neg E\}\}$

unit clause heuristics : D→F

$\{\{E\}, \{\neg E\}\}$

unit clause heuristics : E→T

$\{\square\}$

1a) splitting on variable A: A→F

$\{\{B\}, \{\neg B, C\}, \{\neg C, \neg D\}, \{\neg C, D, E\}, \{\neg B, \neg E\}\}$

unit clause heuristics : B→T

$\{\{C\}, \{\neg C, \neg D\}, \{\neg C, D, E\}, \{\neg E\}\}$

unit clause heuristics : C→T

$\{\{\neg D\}, \{D, E\}, \{\neg E\}\}$

unit clause heuristics : D→F

$\{\{E\}, \{\neg E\}\}$

unit clause heuristics : E→F

$\{\square\}$

b) The variable order doesn't influence the DPLL execution in this example. Whichever variable we choose for splitting, while afterwards picking True first, will lead to the same amount of steps as demonstrated in 6.4 a)

c) When we take the variable order used in 6.4 a) and pick False first we still have the same amount of steps until the DPLL terminates. Hence the Termination will be reached at the same speed. This is because after the first Iteration where we split on Variable A, where A → F, we end up with a the unsatisfied clauses where we have to choose the unit clause heuristic for B next, where B → T. Afterwards the order will be the same as in a) if we follow the breaking of ties in alphabetical order. We have provided the recursive structure as well under 6.4a) 1b).