Foundations of Artificial Intelligence

M. Helmert
S. Eriksson
Spring Term 2022

University of Basel
Computer Science

Exercise Sheet 9

Due: May 1, 2022

Important: for submission, consult the rules at the end of the exercise. Non-adherence to the rules will lead to your submission not being corrected.

Exercise 9.1 (2 marks)

Compile the formula $\chi = \Big((A \vee \neg C) \rightarrow \big(\neg C \wedge (\neg A \vee B)\big)\Big)$ to CNF by applying the logical equivalences from slide 24 of chapter 29 in the print version of the lecture. You may additionally consider commutativity (1) and associativity (2):

$$\varphi \wedge \psi \equiv \psi \wedge \varphi \text{ and } \varphi \vee \psi \equiv \psi \vee \varphi \tag{1}$$

$$((\varphi \wedge \psi) \wedge \eta) \equiv (\varphi \wedge (\psi \wedge \eta)) \text{ and } ((\varphi \vee \psi) \vee \eta) \equiv (\varphi \vee (\psi \vee \eta)). \tag{2}$$

Provide all intermediate formulas that result from applying an equivalence transformation, as shown in the example below. If you use commutativity or associativity, you do not need to provide the corresponding steps.

$$\begin{aligned} \chi' &= (\neg(A \vee \neg B) \vee \neg(C \wedge D)) \\ &\equiv ((\neg A \wedge \neg\neg B) \vee (\neg C \vee \neg D)) && \text{(De Morgan)} \\ &\equiv ((\neg A \wedge B) \vee (\neg C \vee \neg D)) && \text{(double negation)} \\ &\dots \end{aligned}$$

Exercise 9.2 (3 marks)

In the lecture, we have shown that the resolution method can be used for reasoning by a reduction to testing unsatisfiability. In this way, use the resolution method to show that $\neg E \wedge (B \vee D)$ follows logically from $\{\{A, D, E\}, \{A, B, \neg E\}, \{\neg A, B\}, \{\neg B, C\}, \{\neg C\}\}$. Compare the number of required resolution steps with the number of entries that would have been necessary to show the same result with a truth table.

Exercise 9.3 (2+1 marks)

(a) Perfom DPLL on

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

choosing variables alphabetically (also for unit propagation if there are several choices) and always assigning **T** first for splitting.

(b) Would always assigning **F** first result in a smaller tree?

Exercise 9.4 (1+1 marks)

We investigate phase transitions for arbitrary CNF formulas with n variables, m clauses, and k variables in each clause.
If we vary m while fixing n and k, then large values for m represent an *over-constrained region* since all else being equal having more clauses means having more subformulas connected by a logical and, which constrains the formula more.
For the following variations, do large values for the non-fixed variable lead to an under-constrained or over-constrained region? Justify your answer.

(a) Varying k while fixing n and m.

(b) Varying n while fixing m and k.

Submission rules:

- Exercise sheets must be submitted in groups of two students. Please submit a single copy of the exercises per group (only one member of the group does the submission).
- Create a single PDF file (ending .pdf) for all non-programming exercises. Use a file name that does not contain any spaces or special characters other than the underscore "_". 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, but ensure at the same time that the PDF size is not astronomically large. 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, only create those code textfiles required by the exercise. 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.
- For the submission: if the exercise sheet does not include programming exercises, simply upload the single PDF. If the exercise sheet includes programming exercises, upload a ZIP file (ending .zip, .tar.gz or .tgz; *not* .rar or anything else) containing the single PDF and the code textfile(s) and nothing else. Do not use directories within the ZIP, i.e., zip the files directly.
- Do not upload several versions to ADAM, i.e., if you need to resubmit, use the same file name again so that the previous submission is overwritten.