

IA085: Satisfiability and Automated Reasoning

Seminar 2

Exercise 1 Finish the exercises from the previous seminar.

Exercise 2 Using the resolution method, prove that the following sets of clauses are not satisfiable.

1. $\{\{A, B\}, \{\neg A, B, C\}, \{C, \neg D, \neg E\}, \{A, D\}, \{\neg B, D\}, \{\neg C, D\}, \{\neg D, E\}, \{\neg D, \neg E\}\}$
2. $\{\{\neg B, \neg C, D\}, \{\neg B, C, D\}, \{B, E\}, \{B, \neg E\}, \{\neg A, C, \neg D\}, \{\neg C, \neg D\}, \{A, \neg D\}\}$

Exercise 3 Using the resolution method, prove that the following propositional entailments hold.

1. $(A \rightarrow C) \wedge (B \rightarrow C) \wedge (A \vee B) \models C$
2. $(A \rightarrow B) \wedge (A \rightarrow C) \wedge A \models B \wedge C$
3. $(A \rightarrow (B \vee C)) \wedge \neg B \models A \rightarrow C$

Exercise 4 Consider a CDCL-based SAT solver in the following state:

- trail: $[A, \neg B, C, D, E, \neg F, \neg H, I, J]$
- decisions: $[A, C, E]$
- antecedent clauses:
 - $\text{reason}[\neg B] = \{\neg A, \neg B\}$
 - $\text{reason}[D] = \{B, \neg C, D\}$
 - $\text{reason}[\neg F] = \{\neg D, \neg E, \neg F\}$
 - $\text{reason}[\neg H] = \{F, \neg H\}$
 - $\text{reason}[I] = \{\neg A, H, I\}$
 - $\text{reason}[J] = \{B, H, J\}$

In this state, the clause $\{\neg A, F, H, \neg J\}$ was found to be conflicting. Perform conflict analysis and compute the learnt conflict clause using the first UIP criterion. To what decision level should the solver backjump after learning the clause?

Exercise 5 Consider the following list of requirements between packages:

- package X requires packages Y and Z1
- package Y requires package A or package B or package Z2
- package Z1 requires package A
- package A requires packages C and D

- *package C requires package E or package F*
- *package D requires package F or package Z1*
- *packages Z1 and Z2 are not compatible*

Use PySAT and write these constraints on installed packages as a formula. Write a script that finds a model of the formula in which the package X is installed.

Finally, implement a script that computes a model of the formula in which the package X is installed and that is minimal with respect to inclusion of installed packages (i.e., a model from which no package can be removed).

BONUS: Find a model in which the package X is installed and that has the minimal number of installed packages.

Hint: Call the solver multiple times with added constraints. Try to find a smaller model than the current one.