# Questionnaire “Logic and Computability”

# Summer Term 2022

## Contents

# 4 SAT Solvers

## 4.1 Lecture

### 4.1.1 The DPLL-Algorithm

1. [Lecture] Use the DPLL algorithm (*without* BCP, PL and clause learning) to determine whether or not the set of clauses given is satisfiable. Decide variables in alphabetical order starting with the *positive* phase. If the set of clauses resulted in `SAT`, give a satisfying model.

   Clause 1: $(\neg a \vee b)$

   Clause 2: $(\neg b \vee c)$

   Clause 3: $(\neg c \vee d)$

   Clause 4: $(\neg d \vee e)$

   Clause 5: $(\neg e \vee \neg a)$

2. [Lecture] In the context of the DPLL algorithm, explain what a *Unit Clause* is. Give an example.

3. [Lecture] Use the DPLL algorithm with *Boolean Constrain Propagation* (*without* PL and clause learning) to determine whether or not the set of clauses given is satisfiable. Decide variables in alphabetical order starting with the *positive* phase. If the set of clauses resulted in `SAT`, give a satisfying model.

   Clause 1: $(\neg a \vee b)$

   Clause 2: $(\neg b \vee c)$

   Clause 3: $(\neg c \vee d)$

   Clause 4: $(\neg d \vee e)$

   Clause 5: $(\neg e \vee \neg a)$

4. [Lecture] In the context of the DPLL algorithm, explain what a *Pure Literal* is. Give an example.

5. [Lecture] In the context of the DPLL algorithm, explain why it is advantageous to apply *Boolean Constrain Propagation (BCP)* and *Pure Literals (PL)* before making a decision.

6. [Lecture] Use the DPLL algorithm with *Boolean Constrain Propagation* and *Pure Literals* (*without* clause learning) to determine whether or not the set of clauses given is satisfiable. Decide variables in alphabetical order starting with the *positive* phase. If the set of clauses resulted in `SAT`, give a satisfying model.

   Clause 1: $(\neg a \vee b)$

   Clause 2: $(\neg b \vee c)$

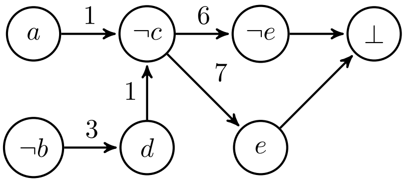
   Clause 3: $(\neg c \vee d)$

   Clause 4: $(\neg d \vee e)$

   Clause 5: $(\neg e \vee \neg a)$

7. [Lecture] In the context of the DPLL algorithm, explain what *Conflict-Driven Clause Learning* is and why most modern SAT solvers use this technique.

8. [Lecture] Consider the following conflict graph with the following set of clauses:



Clause 1: $\{\neg a, \neg c, \neg d\}$

Clause 2: $\{a, \neg d\}$

Clause 3: $\{b, d\}$

Clause 4: $\{\neg b, d, e\}$

Clause 5: $\{\neg b, \neg e\}$

Clause 6: $\{c, \neg e\}$

Clause 7: $\{c, e\}$

Give the resolution proof for the given conflict graph and clauses and state the clause to be learned from the conflict.

9. [Lecture] Use the DPLL algorithm with conflict-driven clause learning to determine whether or not the set of clauses given is satisfiable. Decide variables in alphabetical order starting with the *negative* phase. For conflicts, draw conflict graphs after the end of the table, and add the learned clause to the table.
If the set of clauses resulted in `SAT`, give a satisfying model. If the set of clauses resulted in `UNSAT`, give a resolution proof that shows that the conjunction of the clauses from the table is unsatisfiable.

Clause 1: $\{\neg a, \neg b\}$

Clause 2: $\{a, c\}$

Clause 3: $\{b, \neg c\}$

Clause 4: $\{\neg b, d\}$

Clause 5: $\{\neg c, \neg d\}$

Clause 6: $\{c, e\}$

Clause 7: $\{c, \neg e\}$

10. [Lecture] Use the DPLL algorithm with conflict-driven clause learning to determine whether or not the set of clauses given is satisfiable. Decide variables in alphabetical order starting with the *positive* phase. For conflicts, draw conflict graphs after the end of the table, and add the learned clause to the table.
If the set of clauses resulted in `SAT`, give a satisfying model. If the set of clauses resulted in `UNSAT`, give a resolution proof that shows that the conjunction of the clauses from the table is unsatisfiable.

Clause 1: $(\neg a \vee d)$

Clause 2: $(\neg d \vee c)$

Clause 3: $(\neg b \vee e)$

Clause 4: $(\neg b \vee \neg e)$

Clause 5: $(b \vee f)$

Clause 6: $(b \vee \neg f)$

11. [Lecture] Use the DPLL algorithm with conflict-driven clause learning to determine whether or not the set of clauses given is satisfiable. Decide variables in alphabetical order starting with the *negative* phase. For conflicts, draw conflict graphs after the end of the table, and add the learned clause to the table.
If the set of clauses resulted in `SAT`, give a satisfying model. If the set of clauses resulted in `UNSAT`, give a resolution proof that shows that the conjunction of the clauses from the table is unsatisfiable.

    Clause 1: $(\neg a \lor \neg c)$

    Clause 2: $(b \lor c)$

    Clause 3: $(\neg b \lor \neg d)$

    Clause 4: $(\neg d \lor e)$

    Clause 5: $(d \lor e)$

    Clause 6: $(a \lor \neg c \lor \neg e)$

    Clause 7: $(\neg b \lor c \lor d)$

## 4.2 Practicals

For the following exercises use the DPLL algorithm (including Boolean Constraint Propagation (BCP), pure literals, and conflict-driven clause learning) to check on paper, if the following CNF formulas are satisfiable.

If the formula is satisfiable, give a satisfying model, else show a complete resolution proof for the formula's unsatisfiability.

- Write down all the steps of the DPLL algorithm,
- draw the conflict graphs,
- and state the resolution proofs for all learned clauses

**Rules:**

- When resolving a conflict, only undo the last decision.
- Choose variables for decisions, BCP and pure literals in alphabetical order, starting with the *negative* phase ($\neg a > a > \neg b > b ...$).
- Always try to perform BCP first, before checking for pure literals, before making a decision.

1. [Practicals] [**2 Points**]

   Clause 1: $\{a, b, c\}$

   Clause 2: $\{\neg a, \neg b, \neg c\}$

   Clause 3: $\{a, c, \neg e\}$

   Clause 4: $\{\neg b, \neg c, e\}$

   Clause 5: $\{b, e\}$

   Clause 6: $\{b, \neg d\}$

   Clause 7: $\{\neg c, d\}$

   Clause 8: $\{\neg c, e\}$

2. [Practicals] [**2.5 Points**]

   Clause 1: $\{\neg a, c\}$

   Clause 2: $\{\neg a, b, \neg c\}$

   Clause 3: $\{\neg b, e\}$

   Clause 4: $\{a, d\}$

   Clause 5: $\{a, \neg c\}$

   Clause 6: $\{\neg a, \neg e\}$

   Clause 7: $\{a, \neg b\}$

   Clause 8: $\{b, \neg d\}$

3. [Practicals] [**2.5 Points**]

   Clause 1: $\{a, \neg b, c\}$

   Clause 2: $\{b, \neg c, d\}$

   Clause 3: $\{a, \neg b\}$

   Clause 4: $\{a, c\}$

   Clause 5: $\{\neg c, \neg d\}$

4. [Practicals] [**3 Points**]

   Clause 1: $\{a, \neg b\}$

   Clause 2: $\{a, c\}$

   Clause 3: $\{\neg a, e\}$

   Clause 4: $\{b, c\}$

   Clause 5: $\{b, d\}$

   Clause 6: $\{b, \neg e\}$

   Clause 7: $\{\neg d, e\}$

5. [Practicals] [**3 Points**]

   Clause 1: $\{a, b, c\}$

   Clause 2: $\{\neg a, b\}$

   Clause 3: $\{\neg b, c\}$

   Clause 4: $\{\neg c, d\}$

   Clause 5: $\{\neg c, e\}$

   Clause 6: $\{\neg d, \neg e\}$

6. [Practicals] [**4 Points**]

   Clause 1: $\{a, \neg c, \neg e\}$

   Clause 2: $\{\neg a, \neg e\}$

   Clause 3: $\{b, e\}$

   Clause 4: $\{\neg b, d, e\}$

   Clause 5: $\{\neg b, \neg d\}$

   Clause 6: $\{c, \neg d\}$

   Clause 7: $\{c, d\}$

7. [Practicals] [**3 Points**]

   You are about to plan a train journey in Europe, but you are not yet sure, where to go. You have a few cities in mind, but there are a few restrictions due to a pandemic:

   Your biggest wish is to go to Paris, you are definitely going there. After visiting Paris you are either going to London, or to Madrid, but not both. There is no direct train from your home to Paris, therefore you can take a train either via Berlin or via Zurich. On your way back you can choose between Amsterdam or Zurich. As you want to visit as many cities as possible, you do not want to go trough Zurich twice, therefore you have to go at least through once trough Amsterdam or Berlin. As traveling is currently restricted due to a pandemic, you may not visit Madrid after you visited Berlin and vice versa. You may also not visit London after you went to Amsterdam and vice versa.

   Create a CNF from this description. You can use the following rule to make the formula shorter:

   $$(\neg s \wedge t) \vee (s \wedge \neg t) \ \vdash \ \neg s \vee \neg t$$

   Then use the DPLL algorithm to figure out which which cities would be theoretically possible to visit during the vacation. Formulate your answer as a sentence in English.

## 4.3 Self-Assessment

### 4.3.1 The SAT-Problem

12. [Self-Assessment] Define the *Boolean Satisfiability Problem*?

13. [Self-Assessment] What is the complexity of the SAT-Problem? What does its complexity imply?

### 4.3.2 The DPLL-Algorithm

14. [Self-Assessment] Explain the basic *DPLL algorithm* for checking satisfiability of propositional formulas in *Conjunctive Normal Form (CNF)*. Give a pseudo-code implementation to illustrate your explanations. For simplicity, you can skip all advanced concepts such as Boolean Constraint Propagation, Pure Literals, and Clause Learning.

15. [Self-Assessment] *SAT solvers* make choices based on *heuristics* on which variable and value to pick for the next decision. (a) Why is the variable order for decisions important for the performance of SAT solvers? (b) Explain a commonly used decision heuristics.

16. [Self-Assessment] Given a formula $\varphi$ in CNF representation. (a) What is a *partial assignment* of variables? (b) What is a *total assignment* of variables? (c) What does it mean that a clause in *conflicting* with an assignment? (d) What does it mean that a clause in *satisfied* by an assignment?

17. [Self-Assessment] Given an formula $\varphi$ in CNF representation and an assignment $A$. Tick the following statements if they are true.

    □ A clause is *satisfied* by $A$, if $A$ makes a clause true.

    □ If a clause is *conflicting* with an assignment $A$, if the assignment makes the clause false.

    □ If a clause is *conflicting* with an assignment $A$, all variables in the clause are given the opposite value in A.

    □ A expression $\varphi[A]$ means that all variables within $\varphi$ are assigned according to its truth values in $A$.

18. [Self-Assessment] Within the context of DPLL, explain the terms *decision* and *decision level.*

19. [Self-Assessment] Given the set of clauses $C_\varphi = \{\{a, \neg b\}, \{\neg a, c\}, \{b, \neg c\}, \{\neg a, \neg c\}\}$ and the assignment $A = \{\neg a\}$. Tick the correct statements.

    □ $\varphi[A] = \{\{a, \neg b\}, \{\neg a, c\}, \{\neg a, \neg c\}\}$

    □ $\varphi[A] = \{\{c\}, \{b, \neg c\}, \{\neg c\}\}$

    □ $\varphi[A] = \{\{\neg b\}, \{b, \neg c\}\}$

    □ $\varphi[A] = \{\{\neg b\}, \{c\}, \{b, \neg c\}, \{\neg c\}\}$

20. [Self-Assessment] In the context of the DPLL algorithm, what does a conflict that arises at decision level 0 imply about the satisfiability or unsatisfiability of a formula? Explain your answer.

21. [Self-Assessment] Use the DPLL algorithm (*without* BCP, PL and clause learning) to determine whether or not the set of clauses given is satisfiable. Decide variables in alphabetical order starting with the *positive* phase. If the set of clauses resulted in `SAT`, give a satisfying model.

    Clause 1: $(\neg a \vee b \vee \neg c)$

Clause 2: $(a \vee \neg b \vee c)$

Clause 3: $(\neg a \vee \neg b \vee c)$

Clause 4: $(a \vee b \vee \neg c)$

22. [Self-Assessment] Consider the formula $\varphi$ that consists of the conjunction of the following clauses:

    Clause 1: $(\neg a \vee b)$

    Clause 2: $(\neg a \vee \neg d)$

    Clause 3: $(c \vee \neg b)$

    Clause 4: $(\neg c \vee d)$

    Use the DPLL algorithm (*without* BCP, PL and clause learning) to determine whether or not the set of clauses given is satisfiable. If the set of clauses resulted in `SAT`, give a satisfying model.

    (a) Decide variables in alphabetical order starting with the *positive* phase.

    (b) Decide variables in alphabetical order starting with the *negative* phase.

    (c) What differences can you see between 22a and 22b? Explain in your own words, why for the DPLL algorithm making good decisions is very important.

23. [Self-Assessment] In the context of the DPLL algorithm, explain what *Boolean Constraint Propagation* is. Give an example.

24. [Self-Assessment] Use the DPLL algorithm with *Boolean Constrain Propagation* (*without* PL and clause learning) to determine whether or not the set of clauses given is satisfiable. Decide variables in alphabetical order starting with the *positive* phase. If the set of clauses resulted in `SAT`, give a satisfying model.

    Clause 1: $(\neg d \vee \neg b \vee \neg a)$

    Clause 2: $(\neg e \vee a \vee \neg f)$

    Clause 3: $(\neg a \vee c \vee b)$

    Clause 4: $(f \vee a \vee e)$

    Clause 5: $(d \vee \neg a \vee \neg b)$

    Clause 6: $(\neg a \vee \neg c \vee b)$

25. [Self-Assessment] Why does the DPLL algorithm check for *Boolean Constraint Propagations (BCP)* and *Pure Literals (PL)* before making a decision?

26. [Self-Assessment] Why is the decision level in the DPLL algorithm only incremented after a decision was made but not when the *Pure Literal Rule* or the *Boolean Constrain Propagation Rule* was applied?

27. [Self-Assessment] Use the DPLL algorithm with *Boolean Constrain Propagation* and *Pure Literals* (*without* clause learning) to determine whether or not the set of clauses given is satisfiable. Decide variables in alphabetical order starting with the *positive* phase. If the set of clauses resulted in `SAT`, give a satisfying model.

    Clause 1: $(\neg c \vee d)$

    Clause 2: $(a \vee \neg d \vee \neg e)$

    Clause 3: $(b \vee \neg c)$

    Clause 4: $(c \vee e)$

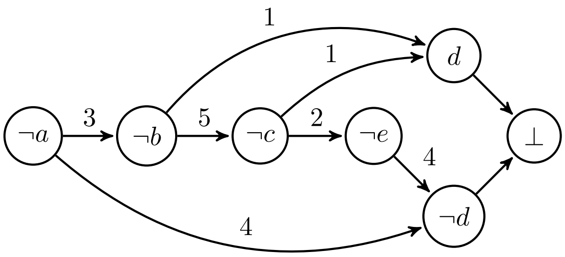
Clause 5: $(\neg b \lor \neg c)$

Clause 6: $(a \lor b)$

28. [Self-Assessment] Explain conflict driven clause learning (CDCL). How do learned clauses prevent the DPLL algorithm of running into already observed conflicts multiple times?

29. [Self-Assessment] In the context of DPLL, give the definition of the *resolution rule* used to construct a resolution proof. Show how the resolution rule derives from the basic natural deduction rules by providing a natural deduction proof.

30. [Self-Assessment] Consider the following conflict graph with the following set of clauses:



Clause 1: $\{b, c, d\}$

Clause 2: $\{c, \neg e\}$

Clause 3: $\{a, \neg b\}$

Clause 4: $\{a, \neg d, e\}$

Clause 5: $\{b, \neg c\}$

State the learned clause by making a resolution proof according to the given conflict graph and given clauses.

31. [Self-Assessment] Consider the formula $\varphi$ that consists of the conjunction of the following clauses:

Clause 1: $(a \lor b)$

Clause 2: $(\neg b \lor c)$

Clause 3: $(\neg a \lor \neg c)$

Clause 4: $(b \lor c)$

Clause 5: $(a \lor \neg b)$

(a) Use DPLL with learning to show that $\varphi$ is unsatisfiable. Decide variables in *alphabetic order* and starting with the *positive* phase.

(b) State and briefly explain the *resolution rule.*

(c) Using your results from 31a, give a resolution proof of the unsatisfiability of $\varphi$.

32. [Self-Assessment] Use the DPLL algorithm with conflict-driven clause learning to determine whether or not the set of clauses given is satisfiable. Decide variables in alphabetical order starting with the *negative* phase. For conflicts, draw conflict graphs after the end of the table, and add the learned clause to the table.
If the set of clauses resulted in `SAT`, give a satisfying model. If the set of clauses resulted in `UNSAT`, give a resolution proof that shows that the conjunction of the clauses from the table is unsatisfiable.

Clause 1: $\{a, b, \neg c\}$

Clause 2: $\{\neg b, c, d\}$

Clause 3: $\{c, d, \neg e\}$

Clause 4: $\{\neg a, d, \neg e\}$

Clause 5: $\{a, b, \neg d\}$

Clause 6: $\{c, \neg d, e\}$

33. [Self-Assessment] Use the DPLL algorithm with conflict-driven clause learning to determine whether or not the set of clauses given is satisfiable. Decide variables in alphabetical order starting with the *negative* phase. For conflicts, draw conflict graphs after the end of the table, and add the learned clause to the table.
If the set of clauses resulted in `SAT`, give a satisfying model. If the set of clauses resulted in `UNSAT`, give a resolution proof that shows that the conjunction of the clauses from the table is unsatisfiable.

    Clause 1: $\{\neg a, \neg b\}$

    Clause 2: $\{a, c, e\}$

    Clause 3: $\{b, \neg d\}$

    Clause 4: $\{\neg c, d, e\}$

    Clause 5: $\{\neg d, e\}$

    Clause 6: $\{\neg a, b\}$

    Clause 7: $\{a, d, \neg e\}$

34. [Self-Assessment] Use the DPLL algorithm with conflict-driven clause learning to determine whether or not the set of clauses given is satisfiable. Decide variables in alphabetical order starting with the *negative* phase. For conflicts, draw conflict graphs after the end of the table, and add the learned clause to the table.
If the set of clauses resulted in `SAT`, give a satisfying model. If the set of clauses resulted in `UNSAT`, give a resolution proof that shows that the conjunction of the clauses from the table is unsatisfiable.

    Clause 1: $\{a, \neg c\}$

    Clause 2: $\{b, c, e\}$

    Clause 3: $\{b, \neg e\}$

    Clause 4: $\{\neg a, c\}$

    Clause 5: $\{d, e\}$

    Clause 6: $\{b, \neg d\}$

    Clause 7: $\{\neg d, \neg e\}$

    Clause 8: $\{a, c\}$

35. [Self-Assessment] Use the DPLL algorithm with conflict-driven clause learning to determine whether or not the set of clauses given is satisfiable. Decide variables in alphabetical order starting with the *negative* phase. For conflicts, draw conflict graphs after the end of the table, and add the learned clause to the table.
If the set of clauses resulted in `SAT`, give a satisfying model. If the set of clauses resulted in `UNSAT`, give a resolution proof that shows that the conjunction of the clauses from the table is unsatisfiable.

    Clause 1: $\{a, b\}$

    Clause 2: $\{\neg a, c\}$

Clause 3: $\{a, \neg d\}$

Clause 4: $\{\neg b, c\}$

Clause 5: $\{\neg c, d\}$

Clause 6: $\{\neg c, e\}$

Clause 7: $\{d, \neg e\}$

36. [Self-Assessment] Use the DPLL algorithm with conflict-driven clause learning to determine whether or not the set of clauses given is satisfiable. Decide variables in alphabetical order starting with the *negative* phase. For conflicts, draw conflict graphs after the end of the table, and add the learned clause to the table.
If the set of clauses resulted in `SAT`, give a satisfying model. If the set of clauses resulted in `UNSAT`, give a resolution proof that shows that the conjunction of the clauses from the table is unsatisfiable.

    Clause 1: $\{\neg a, \neg b\}$

    Clause 2: $\{a, d, e\}$

    Clause 3: $\{b, \neg c\}$

    Clause 4: $\{c, \neg d, e\}$

    Clause 5: $\{\neg c, e\}$

    Clause 6: $\{\neg a, b\}$

    Clause 7: $\{a, c, \neg e\}$

37. [Self-Assessment] Use the DPLL algorithm with conflict-driven clause learning to determine whether or not the set of clauses given is satisfiable. Decide variables in alphabetical order starting with the *negative* phase. For conflicts, draw conflict graphs after the end of the table, and add the learned clause to the table.
If the set of clauses resulted in `SAT`, give a satisfying model. If the set of clauses resulted in `UNSAT`, give a resolution proof that shows that the conjunction of the clauses from the table is unsatisfiable.

    Clause 1: $\{\neg b, c, d\}$

    Clause 2: $\{\neg b, \neg d\}$

    Clause 3: $\{a, \neg c\}$

    Clause 4: $\{\neg c, e\}$

    Clause 5: $\{b, c\}$

    Clause 6: $\{\neg a, \neg e\}$

38. [Self-Assessment] Use the DPLL algorithm with conflict-driven clause learning to determine whether or not the set of clauses given is satisfiable. Decide variables in alphabetical order starting with the *negative* phase. For conflicts, draw conflict graphs after the end of the table, and add the learned clause to the table.
If the set of clauses resulted in `SAT`, give a satisfying model. If the set of clauses resulted in `UNSAT`, give a resolution proof that shows that the conjunction of the clauses from the table is unsatisfiable.

    Clause 1: $\{b, d\}$

    Clause 2: $\{b, c\}$

    Clause 3: $\{\neg b, \neg e\}$

    Clause 4: $\{\neg a, \neg c\}$

Clause 5: $\{\neg c, \neg d\}$

Clause 6: $\{\neg b, c\}$

Clause 7: $\{a, b\}$

Clause 8: $\{\neg b, d, e\}$

39. [Self-Assessment] Use the DPLL algorithm with conflict-driven clause learning to determine whether or not the set of clauses given is satisfiable. Decide variables in alphabetical order starting with the *negative* phase. For conflicts, draw conflict graphs after the end of the table, and add the learned clause to the table.
If the set of clauses resulted in `SAT`, give a satisfying model. If the set of clauses resulted in `UNSAT`, give a resolution proof that shows that the conjunction of the clauses from the table is unsatisfiable.

    Clause 1: $\{\neg b, d, e\}$

    Clause 2: $\{b, e\}$

    Clause 3: $\{c, d\}$

    Clause 4: $\{\neg a, \neg e\}$

    Clause 5: $\{a, \neg c, \neg e\}$

    Clause 6: $\{c, \neg d\}$

    Clause 7: $\{\neg b, \neg d\}$

40. [Self-Assessment] Use the DPLL algorithm with conflict-driven clause learning to determine whether or not the set of clauses given is satisfiable. Decide variables in alphabetical order starting with the *negative* phase. For conflicts, draw conflict graphs after the end of the table, and add the learned clause to the table.
If the set of clauses resulted in `SAT`, give a satisfying model. If the set of clauses resulted in `UNSAT`, give a resolution proof that shows that the conjunction of the clauses from the table is unsatisfiable.

    Clause 1: $(a \vee b \vee c)$

    Clause 2: $(\neg a \vee b)$

    Clause 3: $(\neg b \vee c)$

    Clause 4: $(\neg c \vee d)$

    Clause 5: $(\neg c \vee e)$

    Clause 6: $(\neg d \vee \neg e)$

41. [Self-Assessment] It is Sunday and your fridge is almost empty. You think that you can probably prepare a decent pizza with the little ingredients you have.

    You do have dough. The dough is absolutely necessary for your pizza. You also have arugula, bell pepper and eggplant. You want to put at least one of those three ingredients as toppings on your pizza. Cheese is necessary for the pizza too. You have cheddar and feta. You can use one or both kinds of cheese. You don't like the combination of feta and bell pepper, so you can put at most one of those two ingredients on your pizza. Furthermore you need to save some veggies for dinner, so you can only use either the bell pepper or the eggplant for your pizza.

    Create a CNF from this description. You can use the following rule to make the formula shorter:

$$(\neg s \wedge t) \vee (s \wedge \neg t) \ \vdash \ \neg s \vee \neg t$$

    Then use a DPLL to figure out which ingredients you should use for your pizza and which ingredients you shouldn't use. Formulate your answer as a sentence.

42. [Self-Assessment] Your little cousin needs help to plan her birthday party. There are five kids she thinks about inviting, but not all of them get along. Here is what she tells you:

    My very best friend is Anthony, I have to invite him! I'm also good friends with Daisy and Connie, I want at least one of them to come. But Daisy does not like Benjamin, I can't invite them both! But I do like Benjamin, and I also like Emily. I'd want one of them to be there, or both of them. But Emily is always fighting with Daisy, so only one of them can come.

    Create a CNF from this description. You can use the following rule to make the formula shorter:

$$(\neg s \wedge t) \vee (s \wedge \neg t) \ \vdash \ \neg s \vee \neg t$$

    Then use the DPLL algorithm to figure out which kids your cousin should invite to her birthday party, which kids she should not invite and which kids she can invite without upsetting any other invited guests. Formulate your answer as a sentence.