Lecture Summary: Satisfying Assignments and CNF/DNF

Generated by LectureMate

February 22, 2025

1 Introduction

In the previous lecture, we discussed converting logical formulas into their normal forms, specifically Disjunctive Normal Form (DNF) and Conjunctive Normal Form (CNF). We explored methods such as Karnaugh maps and De Morgan's laws to achieve these conversions. The focus of today's lecture is to understand the significance of CNF and DNF and how they can be utilized to find satisfying assignments for logical formulas.

2 Satisfying Assignments

2.1 Definition

A satisfying assignment is a specific assignment of truth values to variables in a logical formula that makes the formula true. For example, given a formula involving variables a, b, c, etc., we can ask whether there exists an assignment of true or false values that satisfies the formula.

2.2 Importance

Finding satisfying assignments is crucial in various applications, such as:

- Timetabling problems, where we need to schedule exams without conflicts.
- Many computational problems can be transformed into the problem of finding satisfying assignments.

It is noteworthy that any reasonable problem can be expressed as a satisfiability problem, although some problems are inherently difficult.

3 Davis-Putnam-Logemann-Loveland (DPLL) Algorithm

The DPLL algorithm is a well-known method for determining satisfying assignments. It is named after its creators: Martin Davis, Hilary Putnam, George Logemann, and Donald Loveland. The algorithm is celebrated for its simplicity and effectiveness in solving satisfiability problems.

3.1 Background of the Creators

- Martin Davis was a set theorist, not primarily a computer scientist.
- Hilary Putnam was known for his philosophical contributions to logic.
- Logemann and Loveland were computer scientists who contributed to the development of the algorithm.

4 Formal Definitions

4.1 CNF and DNF

- CNF (Conjunctive Normal Form): A formula that is a conjunction of clauses, where each clause is a disjunction of literals.
- DNF (Disjunctive Normal Form): A formula that is a disjunction of conjunctions of literals.

4.2 Haskell Representation

We can represent Boolean propositions in Haskell using data types. For example:

```
data Literal = Pos Atom | Neg Atom
data Clause = Or [Literal]
data Formula = And [Clause]
```

Here, `Literal` can be either a positive or negative atom, `Clause` represents a disjunction of literals, and `Formula` represents a conjunction of clauses.

5 Evaluation of Formulas

To evaluate a formula, we need a valuation that assigns truth values to the propositional variables. A valuation can be represented as a list of literals that are considered true:

```
data Valuation = Val [Literal]
```

5.1 Evaluation Function

We can define a function to evaluate a formula given a valuation:

```
evaluate :: Valuation -> Formula -> Bool
evaluate (Val literals) (And clauses) = all (evaluateClause literals) clauses
```

This function checks if all clauses in the formula evaluate to true based on the provided valuation.

6 Finding Satisfying Assignments

To find a satisfying assignment, we can generate all possible valuations and check if any of them satisfy the formula. The number of possible valuations for n variables is 2^n, which can become computationally expensive.

6.1 Complexity Considerations

The problem of finding a satisfying assignment is related to the P vs NP question in computer science. While we can generate all possible valuations, it is not efficient for large n. However, many practical problems can be solved efficiently despite the theoretical complexity.

7 Conclusion

In this lecture, we explored the concepts of CNF, DNF, and satisfying assignments. We discussed the DPLL algorithm and its significance in finding satisfying assignments for logical formulas. Understanding these concepts is crucial for tackling various computational problems effectively.