

1 Implementierung des SAT-Solvers

In diesem Kapitel wird die Implementierung des SAT-Solvers diskutiert. Hierbei werden zunächst die notwendigen Datenstrukturen beschrieben und danach werden die Algorithmen und der Aufbau des Programmes dargestellt.

1.1 Datenstrukturen

In der Implementierung werden mehrere

1.2 Algorithmen

1.2.1 Unitpropagation

Die Unitpropagation ist einer der wichtigsten Algorithmen für moderne SAT-Solver. In der Unitpropagation werden Klauseln, die nur ein unbelegtes Literal besitzen und noch nicht als SAT evaluiert sind, so belegt, dass Sie wahr werden (siehe Kapitel 4.1).

Algorithmus 1.1 Unitpropagation.

```

unitPropagation :: ClauseList -> TupleClauseList -> Level -> MappedTupleList ->
(ClauseList, TupleClauseList, MappedTupleList)
unitPropagation clauseL tupleCl lvl mapTl
| null clauseL || null unitCl = (clauseL, tupleCl, mapTl)
| otherwise = unitPropagation resolutionC (tlist ++ [(calcTuple, ogClause)]) lvl upda-
tedMap
where
    unitCl = getUnitClause clist
    calcTuple = setVariable unitCl
    ogClause = Reason (getOGFromReducedClauseAndOGClause unitClause)
    updatedMap = pushToMappedTupleList mapped lvl calcTuple ogClause
    subsumptionC = unitSubsumption clist calcTuple
    resolutionC = unitResolution subsumptionC calcTuple

```

Listing 1.1: listing

```

1  unitPropagation :: ClauseList -> TupleClauseList -> Level ->
    MappedTupleList -> TriTuple
2  unitPropagation clist tlist lvl mapped
3
4      — Case: no UnitClause found or no more clauses in
        ClauseList
5      | null clist || null fstElem = (clist, tlist, mapped)
6      | otherwise = unitPropagation resolutionC (tlist ++ [(
        calcTuple, ogClause)]) lvl updatedMap
7  where
8      unitClause = getUnitClause clist
9      fstElem = getClauseFromReducedClauseAndOGClause
        unitClause
10     calcTuple = setVariable fstElem
11     ogClause = Reason (getOGFromReducedClauseAndOGClause
        unitClause)
12     updatedMap = pushToMappedTupleList mapped lvl
        calcTuple ogClause
13     subsumptionC = unitSubsumption clist calcTuple
14     resolutionC = unitResolution subsumptionC calcTuple

```

1 Implementierung des SAT-Solvers

Zunächst wird überprüft, ob die ClauseList Elemente besitzt. Wenn dies der Fall ist, wird nach einer Klausel gesucht, die nur ein unbelegtes Literal besitzt. Wenn jedoch keine Elemente mehr vorhanden sind oder keine Unit-Clause gefunden wird, werden die übergebenen Daten zurückgegeben und der Algorithmus wird beendet. Ansonsten setzt der Algorithmus das Literal so, dass die Klausel wahr wird und wendet Subsumption auf die ClauseList an. Die neue ClauseList wird dann mit Resolution verändert. Gleichzeitig wird der Grund für die Entscheidung des Algorithmus in die MappedTupleList aufgenommen.. Solange der erste Fall nicht eintritt, arbeitet der Algorithmus rekursiv mit den bearbeiteten Daten weiter.