

Имплементација  $DP$  процедуре за исказну  
логику

Ирена Васиљевић, 1018/2021

јун 2022.

## 1 Опис алгоритма

*DP* процедура је алгоритам за испитивање задовољивости формуле исказне логике, заснован на резолуцији. Процедура има својство заустављања, потпуности и сагласности. Суштински корак алгоритма је елиминација променљиве, који је заснован на правилу резолуције. Поред тога, поступак укључује и наредне операције:

- пропација јединичних клауза (*unit propagation*);
- елиминација чистих литерала (*pure literal elimination*).

Такође, важан део процедуре је корак елиминације таутологичних клауза, на почетку процедуре и након сваке итерације. Улаз у алгоритам је скуп клауза, а на излазу су могућа два сценарија:

- добија се празан скуп клауза (у том случају алгоритам пријављује задовољивост);
- добија се празна клауза (у том случају алгоритам пријављује незадовољивост).

## 2 Опис имплементације

Процедура је имплементирана у програмском језику *C++* и суштински прати наредни псеудокод:

**Улаз:** скуп клауза  $S$

**Излаз:** истинитосна вредност која одговара својству задовољивости

```
repeat :
  //unit propagation :
  while  $S$  садржи јединичну клаузу  $\{l\}$ :
    за сваку клаузу  $c$  у  $S$  која садржи  $l$ :
      обриши  $c$  из  $S$ 
    за сваку клаузу  $c$  у  $S$  која садржи  $\neg l$ :
      обриши  $\neg l$  из  $c$ 
  //tautology elimination :
  за сваку клаузу  $c$  у  $S$  која садржи литерал  $l$  и његову негацију  $\neg l$ :
    обриши  $c$  из  $S$ 
  //pure literal elimination :
  while  $S$  садржи чист литерал  $l$ :
    за сваку клаузу  $c$  у  $S$  која садржи  $l$ :
      обриши  $c$  из  $S$ 
  //stopping conditions :
  if  $S$  је празан:
    return true
  if  $S$  садржи празну клаузу:
    return false
  //variable elimination :
  изабери литерал  $l$  који се појављује у оба поларитета у  $S$ 
  за сваку клаузу  $c$  из  $S$  која садржи  $l$  и
  сваку клаузу  $d$  из  $S$  која садржи  $\neg l$ :
    //resolve :
    додај клаузу  $(c \setminus \{l\}) \cup (d \setminus \{\neg l\})$  у  $S$ 
  за сваку клаузу  $c$  из  $S$  која садржи  $l$  или  $\neg l$ :
    обриши  $c$  из  $S$ 
```

### 3 Превођење и покретање

Програм се преводи са: `g++ -o solve main.cpp solver.cpp`, а покреће се са: `./solve`. Програм на улазу очекује формулу у *DIMACS CNF* формату. На излазу програма биће исписано *SAT* уколико је формула задовољива, а *UNSAT* уколико није.

### 3.1 DIMACS CNF формат

*DIMACS CNF* формат је текстуална репрезентација формуле у конјуктивној нормалној форми. Линије које почињу са *c* се сматрају коментарима. Репрезентација формуле почиње линијом *p cnf broj\_promenljivih broj\_klauza*. Након тога, представљају се клаузе формуле. Свака клауза се задаје низом целих бројева, који представљају литерале, и завршава се са 0. На пример, формула  $(x \vee \neg y) \wedge (x \vee y \vee \neg z)$  би имала наредну репрезентацију:

```
p cnf 3 2
1 -2 0
1 2 -3 0
```