Algorithms for Computational Logic

Summary

Contents

1 SAT and Modeling with SAT **2**
1.1 Cardinality Constraints 2
1.1.1 AtMost1 2

2 Optimization problems and SAT-Based Problem Solving **4**

3 Satisfiability Modulo Theories **5**

4 Answer Set Programming **6**

Chapter 1

SAT and Modeling with SAT

1.1 Cardinality Constraints

In order to handle cardinality constraints we have two options: encode the cardinality constraints to CNF and use a SAT solver, or use a pseudo boolean (PB) solver.

1.1.1 AtMost1

- $\sum_{j=1}^{n} x_j = 1$ can be encoded with $\left(\sum_{j=1}^{n} x_j \leq 1\right) \wedge \left(\sum_{j=1}^{n} x_j \geq 1\right)$
- $\sum_{j=1}^{n} x_j \geq 1$ can be encoded with $(x_1 \vee x_2 \vee ... \vee x_n)$
- $\sum_{j=1}^{n} x_j \leq 1$ can be encoded with:
  - Pairwise encoding
  - Sequential counter encoding
  - Bitwise encoding

Sequential Counter

In order to realize this encoding, we need to add new variables s_i for the fact "there is a 1 on some position $1..i$":

$$s_i \text{ is true if } \sum_{j=1}^{i} x_j \geq 1$$

Encoding $\sum_{j=1}^{n} x_j \leq 1$ with sequential counter:

$$\begin{aligned}
&(\neg x_1 \vee s_1) \wedge \\
&(\neg x_i \vee s_i), i \in 2..n-1 \wedge \\
&(\neg s_{i-1} \vee s_i), i \in 2..n-1 \wedge \\
&(\neg x_i \vee \neg s_{i-1}), i \in 2..n
\end{aligned}$$

If $x_j = 1$, then all s_i variables are assigned and all other x variables must take value 0. There are $\mathcal{O}(n)$ clauses and $\mathcal{O}(n)$ auxiliary variables.

Bitwise Encoding

In bitwise encoding, we represent the constraint $\sum_{j=1}^{n} x_j \leq 1$ by encoding the index of the potential true variable in binary. For this, we add new auxiliary variables:

$$v_0, ...v_r - 1;\ r = \lceil \log n \rceil (\text{with } n > 1)$$

Each variable x_j is assigned a unique binary number that represents its index. Then, for each variable x_j with binary index representation i, we create clauses that enforce the condition: if $x_j = 1$ then the auxiliary bits $b_1, b_2, ..., b_k$ must match the binary encoding of i. For a variable x_j with binary index i:

$$x_j \to (b_1^{(i)} \wedge b_2^{(i)} \wedge ... \wedge b_k^{(i)})$$

Where each $b_l^{(i)}$ is either b_l or $\neg b_l$ depending on weather the l-th bit of i is 1 or 0. For example:

$$\begin{aligned}
x_1 &\to (\neg b_1 \wedge \neg b_2) \\
x_2 &\to (\neg b_1 \wedge b_2) \\
x_3 &\to (b_1 \wedge \neg b_2) \\
x_4 &\to (b_1 \wedge b_2)
\end{aligned}$$

There are $\mathcal{O}(n \log n)$ clauses and $\mathcal{O}(\log n)$ auxiliary variables.

Chapter 2

Optimization problems and SAT-Based Problem Solving

Chapter 3

Satisfiability Modulo Theories

Chapter 4

Answer Set Programming