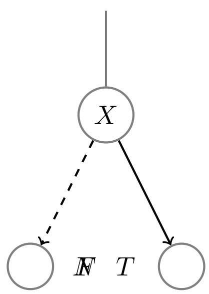
Maintaining GAC via UP on CNF encodings from BDDs and sDNNFs

Jean Christoph Jung   Valentin Mayer-Eichberger

November 23, 2011

1 Definitions

Definition 1 We are working with the following structures:



- Syntax:
  - BDD node N is tuple: $\langle N, X, T, F\rangle$
  - Set of nodes $\mathcal{N}$
  - Set of variables $\mathcal{V}$ with total order $\prec$ and successor operations (s(X) = X+1)
- Rules:

  SINK Let $M = max(\mathcal{V}) + 1$, there are to distinct unique sinks $0, 1 \in \mathcal{N}$ with:

  $$\langle 0, M, 1, 0\rangle$$
  $$\langle 1, M, 1, 0\rangle$$

  EXIST $\forall N \in \mathcal{N}$

  $$\exists T, F \in \mathcal{N} \exists X \in \mathcal{V}. \langle N, X, T, F\rangle$$

  UNIQUE $\forall T, F \in \mathcal{N}\ \forall X \in \mathcal{V} \exists N_1, N_2 \in \mathcal{N}$

  $$\langle N_1, X, T, F\rangle \wedge \langle N_2, X, T, F\rangle \text{ then } N_1 = N_2$$

  REDUNDANT $\forall H \in \mathcal{N}\ \forall X \in \mathcal{V}$

  $$\neg\exists N \in \mathcal{N} \text{ s.t.} \langle N, X, H, H\rangle$$

  ORDER $\forall N, T, F \in \mathcal{N}\ \forall X, Y_1, Y_2 \in \mathcal{V}$

  $$(\langle N, X, T, F\rangle \wedge \langle T, Y_T, *, *\rangle \wedge \langle F, Y_F, *, *\rangle) \rightarrow (X \prec Y_T \wedge X \prec Y_F)$$

- Semantics:

  FORMULA

  $$N \rightarrow ((X \rightarrow T) \wedge (\neg X \rightarrow F))$$ □

Definition 2 The tuple $(\mathcal{V}, \mathcal{N}, R)$ is

BDD if it obeys SINK and EXIST

ROBDD if it is a BDD and obeys UNIQUE,REDUNDANT and ORDER □

Definition 3 $[N_1, ..., N_n]$ is a path iff $\forall N_i$ it holds $N_i \neq 0$ and $((\neg X \wedge \langle N_i, X, *, N_{i+1}\rangle) \vee (X \wedge \langle N_i, X, N_{i+1,*}\rangle)$ □

NOTE IN THIS DEFINITION X IS FREE, if X has domain 0,1 we could use both descendants

Definition 4

- N is reachable $P(N)$, iff there exists a path $[R, ..., N]$
- N is consistent $C(N)$, iff there exists a path $[N, ..., 1]$ □

Definition 5 BDD encoding is GAC wrt scope $(X_1, X_2 ... X_n)$ and their current domains IFF $\exists N_1, N_2, \ldots N_l \in \mathcal{N}$ such that $\forall X_1, X_2, \ldots X_n$ it holds $\bigwedge_{i=1}^{l}(C(N_i) \wedge P(N_i))$. □

2 BDD GAC Encoding

Definition 6 Let $(\mathcal{V}, \mathcal{N})$ be a ROBDD (possible shared). The set $\mathcal{N}$ contains the additional variables to describe a path $N_1, N_2, \ldots N_n$ through the BDD. To construct a path we have several options to prepare it for the SAT solver, with different strength:

ROOT Root is reachable, the 1 sink is true and the 0 sink is false:

$$R \wedge 1 \wedge \neg 0$$

BDD1 N is reachable and $X = 1$ ($X = 0$) then T (F) is reachable

$$(N \wedge X) \rightarrow T$$
$$(N \wedge \neg X) \rightarrow F$$

BDD2 1) N is reachable then one of the descendants is reachable. 2) If both descendants are not reachable, then N is not reachable

$$N \rightarrow (F \vee T)$$
$$(\neg F \wedge \neg T) \rightarrow \neg N$$

BDD3

$$(N \wedge T) \rightarrow X$$
$$(N \wedge F) \rightarrow \neg X$$

PARENT Let $P_1, P_2 \dots P_k$ be all parents of node N. If non of parents is reachable then N not reachable. Let $P_1, P_2 \dots P_k$ with $(\langle P_i, *, N, * \rangle \vee \langle P_i, *, *, N \rangle)$ then

$$\bigwedge_{i=1}^{k} \neg P_i \rightarrow \neg N$$

TOPDOWN Let $N_1, N_2 \dots N_k$ with $\langle N_i, X, T_i, F_i \rangle$ and let $Q_1, Q_2 \dots Q_l$ s.t. $\exists Y, Z \in \mathcal{V}$ with $Y \prec X \prec Z$ s.t. $\langle Q_i, Z, *, * \rangle$ and $\exists P \in \mathcal{N}$ s.t. $\langle P, Y, Q_i, * \rangle$ or $\langle P, Y, *, Q_i \rangle$ then

$$\bigwedge_{i=1}^{l} \neg Q_i \rightarrow \left(\left(\bigwedge_{i=1}^{k} \neg F_i \rightarrow X \right) \wedge \left(\bigwedge_{i=1}^{k} \neg T_i \rightarrow \neg X \right) \right)$$

MAX Max one is reachable in each level: Let $N_1, N_2 \dots N_k$ with $\langle N_i, X, *, * \rangle$

$$\bigwedge_{i \prec j} \neg N_i \vee \neg N_j$$

□

3 LOCAL KNOWLEDGE rules

I guess we can change this section to work only with sDNNF, then ROBDD are a special case of it. so the rules have to be rewritten (possible simpler). (rule for local knowledge, rule for OR node, rule for AND node, rule for leaf)

This is a more general approach (could work for sDNNF as well). We model intersection and union with additional local knowledge variables. Node variables mean again *being on the final path*.

Definition 7 We define the occurring literals of a sDNNF subtree with root N as follows

$$lit(N) = \begin{cases} \{L\} & \text{if } N \text{ is leaf and } L \text{ is its literal} \\ \bigcup_i lit(F_i) & F_1 \dots F_n \text{ children of } N \end{cases}$$

□

Now for each node N we introduce variables N_Y for each $Y \in lit(N)$. They correspond to: N_Y means that literal Y is known in node N.

GLOBAL Local knowledge at the root is global knowledge.

$$\forall Y \in lit(R) \quad R_Y \rightarrow Y$$

INTERSECTION Let N be an OR node. Implications are propagated up, if they are known in all descendants $F_1 \dots F_n$. $\forall Y \in lit(N)$

$$\left(\bigwedge_{i=1}^{n} F_{i,Y} \right) \rightarrow N_Y$$

UNION Let N be an AND node. Implications are propagated up, if they are known in one of the descendants $F_1 \dots F_n$. $\forall Y \in lit(N)$

$$\left(\bigvee_{i=1}^{n} F_{i,Y} \right) \rightarrow N_Y$$

QUESTION FOR JEAN: WHICH SDDNDDDSFS BLABLABLA excludes this ?

Example 1 does it work for:

$$(X \vee Y) \wedge (\neg X \vee Y)$$

□

4 SPECIAL RULES

This section contains rules for BDD encodings representing special classes of boolean functions:

PBC WE NEED EXTRA SYMBOLS AND BOTTOM UP RULES FOR CONSISTENCY The BDD represents a PBC. Then let $N_1, N_2 \dots N_k$ with $\langle N_i, X, *, * \rangle$ and $C(N_i)$ implies $C(N_{i+1})$ (follows from being PBCs, and level property corresponding logical function), then

$$\bigwedge_{i=1}^{n-1} C(N_i) \rightarrow C(N_{i+1})$$

5 UPBDD Encoding Rules

This section contains rules for the UPBDD theories. No change in strenth of UP but faster and smaller encoding.

6 Conclusion

We showed that the two standard SAT encodings of PBC do not maintain GAC and proposed an encoding that does.