Technische Universität München (I21)
Wolfgang Nicka

Summer term 2016
June 14, 2016

Seminar Decision Procedures – Homework 1

Discussed on Wednesday, 28^{th} June, 2016.

Exercise 1.1 Reverse unit propagation

Prove:
Let F be a CNF formula and C a clause not in F.
$\varnothing \in BCP(F \cup \bar{C}) \iff$ C has AT with respect to F.

Exercise 1.2 Redundancy Properties

Let $F := (\bar{a} \vee b) \wedge (c \vee d) \wedge (a \vee \bar{d}) \wedge (\bar{b} \vee \bar{e})$. Determine whether the following lemmas have T, AT, RT and/or RAT with respect to F:

- $(a \vee \bar{a})$
- $(\bar{d} \vee b)$
- $(e \vee a)$

Exercise 1.3 Lookouts

It is the year 2058 and you find yourself in military command of a space station at the intersection of a few valleys on mars in war time. There are no enemy ground troops yet, but you know the enemy's army will be coming from one of them in the near future. You are planning to place lookouts on towers to see the army coming. Naturally you want all the valleys covered, but you only have k lookouts available and there are $n = k + 1$ towers, which also sometimes offer vision into the same valleys. Luckily, most of the towers offer vision into multiple valleys and your lookouts are capable enough to watch all valleys a tower offers vision into at the same time.
Despite your excellent planning capabilities you think that you do not have enough lookouts to supply towers in such a way that you will see the enemy army coming no matter which valley they choose to attack from. You decide to go to your superior and request more. You don't have time to try out every possibility or look for a solution by hand. How can you use a CNF SAT solver to prove to your supervisor that you need more lookouts?

There are v valleys, k lookouts and $n = k + 1$ towers. For every tower $t \in [n]$ you have a list of boolean values $v_{t,1}...v_{t,v}$ that are true if tower t gives vision on valley v, otherwise false. You want to prove that there is no way you can place your k lookouts on towers in $[n]$ in such a way that all valleys are covered using a CNF SAT solver.

You do not have to show that getting more lookouts actually solves the problem, only that you don't have enough.

Notes

Resolution
For $A = x \vee \bigvee a_i$ and $B = \bar{x} \vee \bigvee b_i$ in F, $C = \bigvee a_i \vee \bigvee b_i$ is called the resolvent of A and B.

Boolean constraint propagation
Repeat until fixpoint: If there is a unit clause (l) in F remove all clauses containing l from $F \setminus \{(l)\}$ and remove $\bar{l}$ from all clauses in F.
The resulting clause is written as BCP(F).

Asymmetric Literal Addition
ALA for a clause C in F performs the following until fixpoint:
If there exist literals l_i in C and there exists a clause $(\bigvee l_i \vee l)$ in F\{C} then let $C := C \cup \bar{l}$.
The resulting clause is referred to as ALA(F, C).

Asymmetric Tautology / Reverse Unit Propagation A clause C has asymmetric tautology (AT) / is RUP with respect to F iff ALA(F, C) has property T (tautology).

Resolution Tautology / Resolution Asymmetric Tautology
A clause C in a CNF formula F has the property $R\mathcal{P}$ for $\mathcal{P} \in \{T, AT\}$ if either C has $\mathcal{P}$ or there exists a literal $l \in C$ so that $\forall C' \in F$ with $\bar{l} \in C'$ the resolvent of C and C' has $\mathcal{P}$.

Solutions

Reverse Unit Propagation:
C has AT $\Leftrightarrow$ ALA(F, C) = $(\bigvee l_i \vee l \vee \bar{l})$ for literals l_i, l.
First show: every literal in ALA(F, C) in each step corresponds to a negated unit clause in $BCP(F \cup \bar{C})$.
Let $ALA_j(F, C)$ be the clause C after computation step j of $ALA(F, C)$ and $BCP_j(F \cup \bar{C})$ the BCP formula after unit propagation on all units in $BCP_{j-1}(F \cup \bar{C})$ that also appear in negated form in $ALA_{j-1}(F, C)$.
Recursion:
Initialization: $ALA_0(F, C)$ is C. $\bar{C} = \overline{\bigvee l_i} = \bigwedge(\bar{l_i})$ are unit clauses in $BCP_0(F \cup \bar{C})$.
Step: For $ALA_j(F, C)$ let $C_{j-1} := ALA_{j-1}(F, C)$, $\bar{l}$ be the literal added to C_{j-1} in step j. Then there exists a clause D in F such that $D = (\bigvee a_i \vee l)$ such that all a_i are in C_{j-1}. Because all $\bar{a}_i$ are unit clauses in $BCP_{j-1}(F \cup \bar{C})$ (recursion step) unit propagation on $BCP_{j-1}(F \cup \bar{C})$ removes all a_i from D leaving only $D = (l)$, a unit clause in $BCP_j(F \cup \bar{C})$.
For $\varnothing \in BCP_n(F \cup \bar{C})$ there exist clauses D, E in $BCP_{n-1}(F \cup \bar{C})$ such that $D = (l)$ and $E = (\bigvee a_i \vee \bar{l})$ (removal of a clause never leaves the empty set) and all $(\bar{a}_i)$ are unit clauses. It follows that all a_i as well as l and $\bar{l}$ are in $ALA_{n-1}(F, C)$, which means it and all potential later versions of it have T, so ALA(F, C) has T and C has AT.
In the other direction if C has AT then ALA(F, C) has T, so l, $\bar{l}$ are in ALA(F, C) for some literal l, which means $(\bar{l})$ and (l) are unit clauses in some BCP step and unit propagation on either resolves in a conflict.

Redundancy properties:

- $(a \vee \bar{a})$ is a tautology (has T) and thus also has AT, RT and RAT.
- $(\bar{d} \vee b)$ doesn't have T. Unit propagation on $\overline{(\bar{d} \vee b)} = (d) \wedge (\bar{b})$ results in $BCP(F \wedge (d) \wedge (\bar{b})) = BCP((\bar{a}) \wedge (a) \wedge (d) \wedge (\bar{b})) = \varnothing \wedge (d) \wedge (\bar{b})$, a conflict on a, so the lemma is RUP / has AT, and thus also RAT. The only clause in F containing d is $(c \vee d)$ and does not resolve to a tautology with $(\bar{d} \vee b)$. The same is true for $\bar{b} \vee \bar{e}$, the only clause in F containing $\bar{b}$, so the lemma does not have RT.
- $(e \vee a)$ is not a tautology. $BCP(F \wedge \overline{(e \vee a)}) = (c) \wedge (\bar{d}) \wedge (\bar{e}) \wedge (\bar{a})$, so doesn't result in a conflict and doesn't have AT. The clauses in F containing $\bar{a}$ and $\bar{e}$ respectively are $(\bar{a} \vee b)$ and $(\bar{b} \vee \bar{e})$, which resolve to $(b \vee e)$ and $(a \vee b)$ respectively. Each resolvent is not a tautology, so the lemma doesn't have RT. Unit propagation on $\overline{(b \vee e)}$ doesn't result in a conflict, neither does unit propagataion on $(\bar{a}) \wedge (\bar{b})$, so the lemma doesn't have RAT.

Lookouts
Let T be the set of towers and let $T_1, ..., T_v \subseteq T$ be the set of indices of the towers that can see into valleys 1 to v, easily constructed from the adjacency matrix given by the $v_{j,i}$.
To solve this problem with SAT build a CNF formula F that is satisfiable if it is possible to find a subset of the towers, missing at least one, that covers each valley. First, construct clauses $C_1, ..., C_v$ by creating literals $t_1, ..., t_n$ for the towers that can see into valleys $1, ..., n$ using $T_1, ..., T_v$. Obviously, C_i is true exactly when one of the variables in C_i is true (when a lookout is placed on that tower), and $\bigwedge C_i$ is true exactly when every C_i is true, meaning every valley is covered by at least one lookout.
Finally, add a clause with the information that only $k = n - 1$ lookouts are available by constructing a clause C_+ that is true only if at least one tower is empty: $C_+ = (\bigvee_{i=1}^{n} \bar{t_i})$. Then $F := \bigwedge_{i=1}^{v} C_i \wedge C_+$ is a CNF formula that is satisfiable exactly when k lookouts can be placed on the n towers such that all valleys are covered.