Preprocessing in Incremental SAT
A proof supplement

(Technical report: IE/IS-2012-03)

Alexander Nadel [1] Vadim Rivchyn[1,2] Ofer Strichman[2]
alexander.nadel@intel.com rvadim@tx.technion.ac.il
ofers@ie.technion.ac.il

[1] Design Technology Solutions Group, Intel Corporation, Haifa, Israel
[2] Information Systems Engineering, IE, Technion, Haifa, Israel

In this note we prove the correctness of Preprocess-inc, which is presented here together with its descendants in Alg. 1. The presentation here of the algorithm is identical to [1], except that ReEliminate-Or-ReIntroduce has been abstracted: it uses a nondeterministic choice between reintroduction and reelimination (a conservative abstraction of the algorithm), and ignores the fact that assumptions must be reintroduced.

The notation used in this note is mostly consistent with [1], but some additional symbols are required. Specifically we use subscripts to denote restrictions of a formula to a subset of its clauses[1], e.g., for a CNF formula θ,

$$\begin{aligned}
\theta_j \ &\doteq \{c \mid c \in \theta, v_j \in c\}, \\
\theta_{\neq j} &\doteq \theta \setminus \theta_j, \text{and} \\
\theta_{>j} &\doteq \{c \mid c \in \theta, v_t \in c \Longrightarrow t > j\}.
\end{aligned}$$

Unless otherwise stated variable symbols, such as v_j in the definitions above, refer to variables, not literals. As in [1], we will use ψ to denote formulas without preprocessing at all, and φ to denote formulas in different stages of the preprocessing algorithm. Specifically, the first formula to be decided is ψ^0, and ψ^i is the $i+1$-th such formula. Similarly, φ^i is the $i+1$-th formula that is actually solved by the SAT solver. We refer to each iteration of the external loop of the incremental solver (from the time Δ^i, the i-th increment to the formula, is added, until the preprecessed formula is submitted to the SAT solver), as a *layer*. The i-th layer begins by applying our algorithm to $\varphi^{i-1} \wedge \Delta^i$. We use a 2-dimension superscript to denote the position in the algorithm, e.g., $\theta^{i,j}$ denotes the value of θ in *layer* i, at the j-th iteration of the (first) internal loop of Preprocess-Inc. Additional notation is summarized in Table 1.
Our goal is to prove:

Proposition 1. *Algorithm* PreProcess-Inc *is correct, i.e.,*

$$\forall i.\ \psi^i \textit{ is equisatisfiable with } \varphi^i\ .$$

[1] We adopt the traditional dual view of CNF-s as either formulas or sets of clauses, depending on the context.

Algorithm 1 Preprocessing in an incremental setting. This algorithm is identical to the one appeared in [1], except the abstraction in ReEliminate-Or-ReIntroduce.

function ReIntroduceVar(var v, int loc, int i)
  $\varphi^i \mathrel{+}= S_v \cup S_{\bar{v}}$;
  erase $ElimVarQ[loc]$; ▷ v is not eliminated, hence 0 resolvents

function ReEliminateVar(clauseset Res, var v, int loc, int i)
  $S_v = S_v \cup \varphi_v^i; S_{\bar{v}} = S_{\bar{v}} \cup \varphi_{\bar{v}}^i$;
  $ElimVarQ[loc].resolvents \mathrel{+}= |Res|$;
  $\varphi^i = (\varphi^i \cup Res) \setminus (\varphi_v^i \cup \varphi_{\bar{v}}^i)$;
  ClearDataStructures (v);
  $TouchedVars = TouchedVars \cup vars(Res)$;

function ReEliminate-Or-ReIntroduce(int loc, int i)
  clauseset$Res =$ $Res_i(\varphi_v^i, \varphi_{\bar{v}}^i) \cup$
  $Res_i(\varphi_v^i, S_{\bar{v}}) \cup Res_i(S_v, \varphi_{\bar{v}}^i)$;
  if * **then** ReIntroduceVar(v, loc, i);
  else ReEliminateVar (Res, v, loc, i);

function Preprocess-inc(int i) ▷ preprocessing of φ^i
  $SubsumptionQ = \{c \mid \exists v.\ v \in c \wedge v \in vars(\Delta^i)\}$;
  RemoveSubsumptions ();
  for ($j = 0 \ldots |ElimVarQ| - 1$) **do** ▷ scanning eliminated vars *in order*
    $v = ElimVarQ[j].v$;
    if $|\varphi_v^i| = |\varphi_{\bar{v}}^i| = 0$ **then continue**;
    ReEliminate-Or-ReIntroduce (j, i);
  while $SubsumptionQ \neq \emptyset$ **do**
    for each non-assumption variable $v \notin ElimVarQ$ **do** ▷ scanning the rest
      $SubsumptionQ =$ EliminateVar-inc (v, i);
      RemoveSubsumptions ();
    $SubsumptionQ = \{c \mid vars(c) \cap TouchedVars \neq \emptyset\}$;
    $TouchedVars = \emptyset$;

ψ^0	Initial formula.
Δ^i	The i-th increment.
ψ^i	$= \psi^0 \wedge \bigwedge_{k=1}^{i} \Delta^k$.
φ^i	The formula solved by the SAT solver instead of ψ^i.
$cl(\varphi)$	$\doteq \{c \mid \varphi \implies c, vars(c) \subseteq vars(\varphi)\}$. In words, $cl(\varphi)$ is the logical closure of a CNF φ. We use it to refer to a formula together with clauses learnt from it.
$Res_j(c_1, c_2)$	Binary resolution of clauses c_1, c_2 where v_j is the pivot.
$Res_j(cls)$	For a set of clauses cls such that $cls = cl_1 \cup cl_2$ and cl_1, cl_2 contain respectably v_j and $\bar{v_j}$, $Res_j(cls) = \bigwedge_{c_1 \in cl_1, c_2 \in cl_2} Res_j(c_1, c_2)$.
j	A variable index in $ElimVarQ$.
S_j	$\doteq S_{v_j} \cup S_{\bar{v_j}}$. Recall, for a variable v, S_v denotes the set of clauses containing v that were resolved away because v is eliminated.
E	An ordered set of variables, equivalent to the projection of $ElimVarQ$ to its variables.
$E^{i,j}$	The set E at the i-th layer, after j iterations of the loop in Alg. 2, restricted to the variables $v_0 \ldots v_j$.
$\exists[varset]$.	An existential quantification operation over the set of variables in $varset$.
$sz[i]$	$\doteq \lvert E^{i,0} \rvert - 1$. This is the number of elements in $ElimVarQ$, with which $\varphi^{i-1} \wedge \Delta^i$ is being preprocessed. The shortened notation below is used instead.
sz	Same as $sz[i]$, when i is known from the context (simplifies notation).

Table 1. Notation used in this note.

Note that the loop in lines 8 – 13 of Preprocess-inc is standard (non-incremental) preprocessing. Hence it is sufficient to prove that the formula entering this loop for the first time is equisatisfiable to ψ^i. From thereon equisatisfiability is maintained by the standard properties of preprocessing. Hence, we focus on lines 4 – 7 of that algorithm, as shown in Alg. 2. ReEliminate-Or-ReIntroduce of line 7 in Preprocess-inc and its descendants have been inlined. Some of the statements were rewritten so they use the notation that we will use in the proof, and is explained in Table 1.[2] Note that:

- Line 2 is equivalent to line 6 of Preprocess-inc, only that it uses our convention by which subscripts denote restrictions to *variables* rather than literals;
- Line 4 is equivalent to line 4 of ReEliminateVar;
- Line 5 is equivalent to line 2 in ReIntroduceVar.

For simplicity Alg. 2 and the proof that follows ignores the issue of assumption variables. It also mostly ignores the function RemoveSubsumptions since it only applies subsumption and self-subsumption, both of which maintain logical equivalence.

Algorithm 2 An abstract inlined version of the loop in lines 4–7 of Preprocess-inc.

1: **for** $j = 0 \ldots sz$ **do**
2:     **if** $|\varphi^i_j| = 0$ **then continue**;
3:     **if** * **then**
4:         $\varphi^{i,j} = (\varphi^{i,j-1}_{\neq j}) \wedge Res_j(\varphi^{i,j-1}_j \cup S_j)$ ▷ reelimination
5:     **else** $\varphi^{i,j} = \varphi^{i,j-1} \wedge S_j$ ▷ reintroduction

Whereas it is sufficient, by the arguments given above, to prove that Alg. 2 maintains satisfiability in order to establish Proposition 1, this claim is not strong enough to be inductive. We therefore prove a stronger proposition as stated in Proposition 2 below.

Without loss of generality, assume that $E = v_0, v_1, \ldots,$ in that order, and that variables not in $ElimVarQ$ have higher indices. We can always rename the variables to force this property at the beginning of each layer: since variables that are eliminated in line 10 of Preprocess-inc are pushed to the end of ElimVarQ, such renaming will not change the order within this list. We now claim that:

Proposition 2.

$$\forall i\ .\ \ \psi^i \Longrightarrow \varphi^{i,sz} \tag{1}$$

$$\forall i\ .\ \ \varphi^{i,sz} \Longrightarrow \exists[E^{i,sz}].\ \psi^i\ . \tag{2}$$

[2] The difference in the notation is related to the fact that in the algorithm variables are updated several times in the loop, whereas in the proof we need each symbol to refer to a single value at a specific location.

Recall that sz is a shortened version for $sz[i]$, and hence $\varphi^{i,sz}$ is simply the formula at the end of Alg. 2. Clearly Proposition 2 implies that the formulas at the beginning and end of Alg. 2 are equisatisfiable, which in itself implies Prop 1.

Proof. (Proposition 2) The proof is by induction on i.

Base case. For $i = 0$ $ElimVarQ = \emptyset$ at the beginning of Alg. 2, and hence $sz = 0$ and $E^{0,sz} = \emptyset$. Alg. 2 is therefore skipped, $\psi^0 \Leftrightarrow \varphi^{0,0}$ and both (1) + (2) hold trivially.

Step. We assume Proposition 2 holds up to $i - 1$ — hereafter referred to as hypothesis **H1** — and prove the proposition for i. We define:

$$Removed_{\geq j} \doteq \bigwedge_{k=j}^{sz} \bigwedge_{cls \in S_k} cls \ .$$

In words, $Removed_{\geq j}$ for $j \in [0..sz]$ is a conjunction of all the eliminated clauses, such that if v_t led to the elimination of the clause, then $t \geq j$. We claim that:

Lemma 1. *For $j \in [0..sz]$:*

$$\psi^i \Longrightarrow \varphi^{i,j} \wedge Removed_{\geq j+1} \tag{3}$$

$$\varphi^{i,j} \wedge Removed_{\geq j+1} \Longrightarrow \exists[E^{i,j}].\ \psi^i \ . \tag{4}$$

Note that when $j = sz$ this lemma is equivalent to Proposition 2 for a given i.

Proof. The proof is by induction on j:

Base case. The first iteration begins with $\varphi^{i-1} \wedge \Delta^i$ and ends with either

$$\text{Reelimination:} \quad \varphi^{i,0} = \varphi^{i-1} \wedge \Delta^i_{>0} \wedge Res_0(S_0 \cup \Delta^i_0) \ \text{, or} \tag{5}$$

$$\text{Reintroduction:} \quad \varphi^{i,0} = \varphi^{i-1} \wedge \Delta^i \wedge S_0 \tag{6}$$

Note that (6) implies (5), because

$$\Delta^i \Longrightarrow \Delta^i_{>0}, \text{and}$$
$$\Delta^i \Longrightarrow \Delta^i_0 \text{ and } S_0 \wedge \Delta^i_0 \Longrightarrow Res_0(S_0 \cup \Delta^i_0) \ .$$

Hence, from the four combinations of $\{(3),(4)\} \times \{(5),(6)\}$ we only need to prove ((3),(6)) and ((4),(5)):

- (3),(6): We need to prove

$$\psi^i \Longrightarrow \underbrace{\varphi^{i-1} \wedge \Delta^i \wedge S_0}_{\varphi^{i,0}(6)} \wedge Removed_{\geq 1} \ .$$

- $\psi^i \implies \varphi^{i-1}$, since $\psi^i \implies \psi^{i-1}$ and by hypothesis H1, $\psi^{i-1} \implies \varphi^{i-1}$;
- $\psi^i \implies \Delta^i$, since $\psi^i = \psi^{i-1} \wedge \Delta^i$; and
- $\psi^i \implies S_0 \wedge \mathit{Removed}_{\geq 1}$ because the RHS is contained in $cl(\psi^i)$.

– (4),(5): We need to prove

$$\underbrace{\varphi^{i-1} \wedge \Delta^i_{>0} \wedge \mathit{Res}_0(S_0 \cup \Delta^i_0)}_{\varphi^{i,0}(5)} \wedge \mathit{Removed}_{\geq 1} \implies \exists v_0.\ \psi^i\ .$$

The RHS is equivalent to $\exists v_0.\ \psi^{i-1} \wedge \Delta^i$, or to:

$$\psi^{i-1}_{>0} \wedge \Delta^i_{>0} \wedge \exists v_0.\ \psi^{i-1}_0 \wedge \Delta^i_0\ ,$$

and is implied by the LHS:

- $\varphi^{i-1} \wedge \mathit{Removed}_{\geq 1} \implies \psi^{i-1}_{>0}$, because $\psi^{i-1}_{>0} \subseteq (\varphi^{i-1} \cup \mathit{Removed}_{\geq 1})$ (to be more precise it is $\psi^{i-1}_{>0}$ after subsumption that is contained in the RHS, but as mentioned earlier we ignore subsumption because it maintains logical equivalence). To see why, observe that:
  - $\psi^{i-1} \subseteq (\varphi^{i-1} \cup \mathit{Removed}_{\geq 0})$;
  - Equivalently, $\psi^{i-1}_0 \wedge \psi^{i-1}_{>0} \subseteq (\varphi^{i-1} \cup S_0 \cup \mathit{Removed}_{\geq 1})$;
  - ψ^{i-1}_0 and $S0$ only contain clauses with v_0, and they are the only components in the above that contain such clauses. Hence we can remove them from both sides, which gives us $\psi^{i-1}_{>0} \subseteq (\varphi^{i-1} \cup \mathit{Removed}_{\geq 1})$.
- $\Delta^i_{>0}$ is on the LHS; and
- $\varphi^{i-1} \wedge \mathit{Res}_0(S_0 \cup \Delta^i_0) \implies \exists v_0.\ \psi^{i-1}_0 \wedge \Delta^i_0$, because

  $$\mathit{Res}_0(\psi^{i-1}_0 \cup \Delta^i_0) \implies \exists v_0.\ \psi^{i-1}_0 \wedge \Delta^i_0$$

  by definition of resolution, and

  $$\varphi^{i-1} \wedge \mathit{Res}_0(S_0 \cup \Delta^i_0) \implies \mathit{Res}_0(\psi^{i-1}_0 \cup \Delta^i_0)\ . \tag{7}$$

  Justifying (7) requires a split:
  - First, assume there is no subsumption. In that case, $\psi^{i-1}_0 \subseteq S_0$ (S_0 can contain more clauses than ψ^{i-1}_0 because of conflict clauses), which means that $\mathit{Res}_0(S_0 \cup \Delta^i_0)$ alone implies the RHS.
  - Now assume there is subsumption. In that case a clause $c \in \psi^{i-1}_0$ may be subsumed by a clause c' which is not in ψ^{i-1}_0. On the other hand c' (or yet another clause that subsumed c', and so on) is still in φ^{i-1}. Any resolvent of c on the RHS must be subsumed by c' as well. The reason is that the pivot is v_0 and v_0 is not in c'. Hence $c' \subset (c \setminus v_0)$ and therefore must be part of the resolvent.

Step (Lemma 1). Assume (3) and (4) hold for $j-1$ — hereafter referred to as hypothesis **H2** — and prove the proposition for j. The $j-1$-th iteration begins with $\varphi^{i,j-1}$ and ends with

$$\text{Reelimination: } \varphi^{i,j} = (\varphi^{i,j-1}_{\neq j}) \wedge \mathit{Res}_j(\varphi^{i,j-1}_j \cup S_j) \tag{8}$$

$$\text{Reintroduction: } \varphi^{i,j} = \varphi^{i,j-1} \wedge S_j\ . \tag{9}$$

Similarly to the base case, we rely on the relation between the two possible outcomes to reduce the number of cases that need to be checked. In particular, (9) implies (8) because:

$$\begin{aligned}\varphi^{i,j-1} &\implies (\varphi^{i,j-1}_{\neq j}),\ \text{and}\\ \varphi^{i,j-1}_{j} \wedge S_j &\implies Res_j(\varphi^{i,j-1}_{j} \cup S_j), \text{and } \varphi^{i,j-1}_{j} \subset \varphi^{i,j-1}\ .\end{aligned}$$

Hence from the four combinations of $\{(3),(4)\} \times \{(8),(9)\}$ we only need to prove ((3),(9)) and ((4),(8)):

– (3),(9): We need to prove:

$$\psi^i \implies \underbrace{\varphi^{i,j-1} \wedge S_j}_{\varphi^{i,j}(9)} \wedge Removed_{\geq j+1}\ ,$$

or equivalently

$$\psi^i \implies \varphi^{i,j-1} \wedge Removed_{\geq j}\ ,$$

- $\psi^i \implies \varphi^{i,j-1}$ by **H2**;
- $cl(\psi^i) \supseteq Removed_{\geq j}$.

– (4),(8): We need to prove:

$$\underbrace{(\varphi^{i,j-1}_{\neq j} \wedge Res_j(\varphi^{i,j-1}_{j} \cup S_j)}_{\varphi^{i,j}(8)} \wedge Removed_{\geq j+1} \implies \exists [E^{i,j}].\ \psi^i\ . \tag{10}$$

Falsely assume that (10) does not hold, which implies that there exists an assignment α such that

$$\alpha \models (\varphi^{i,j-1}_{\neq j}) \wedge Res_j(\varphi^{i,j-1}_{j} \cup S_j) \wedge Removed_{\geq j+1} \wedge \forall [E^{i,j}].\ \neg\psi^i\ . \tag{11}$$

Note that v_j is not free in (11), and hence α does not have to include a value for it. We will prove that from α we can derive an assignment that contradicts the hypothesis **H2**, and therefore α cannot exist.
By **H2**:

$$\varphi^{i,j-1} \wedge Removed_{\geq j} \implies \exists [E^{i,j-1}].\ \psi^i\ ,$$

which is the same as

$$(\varphi^{i,j-1}_{\neq j}) \wedge \varphi^{i,j-1}_{j} \wedge S_j \wedge Removed_{\geq j+1} \implies \exists [E^{i,j-1}].\ \psi^i\ . \tag{12}$$

We will show an assignment that contradicts this implication.
Standard resolution implies that

$$Res_j(\varphi^{i,j-1}_{j} \cup S_j) \implies \exists v_j.\ (\varphi^{i,j-1}_{j} \wedge S_j)\ , \tag{13}$$

and hence (11) implies:

$$\alpha \models (\varphi^{i,j-1}_{\neq j}) \wedge (\exists v_j.\ (\varphi^{i,j-1}_{j} \wedge S_j)) \wedge Removed_{\geq j+1}\ . \tag{14}$$

Denote by α' the assignment α augmented with a value of v_j chosen to instantiate the existential quantifier in (14). Since v_j only appears in $(\varphi_j^{i,j-1} \wedge S_j)$[3] and α does not have a value for v_j, then

$$\alpha' \models (\varphi_{\neq j}^{i,j-1}) \wedge (\varphi_j^{i,j-1} \wedge S_j) \wedge \mathit{Removed}_{\geq j+1} ,$$

i.e., α' satisfies the LHS of (12). It is left to show that α' also satisfies the negation of the RHS of (12), i.e.,

$$\alpha' \models \neg\exists[E^{i,j-1}].\ \psi^i ,$$

or

$$\alpha' \models \forall[E^{i,j-1}].\ \neg\psi^i . \tag{15}$$

By (11) we know that

$$\alpha \models \forall[E^{i,j}].\ \neg\psi^i . \tag{16}$$

This is equivalent to

$$\alpha \models \forall v_j \forall[E^{i,j-1}].\ \neg\psi^i .$$

Note that (15) can be seen as an instantiation of the left universal quantifier to the value given to v_j by α', and hence is implied by the above.
Hence, we have shown an assignment α' that contradicts (12) and therefore the assumption **H2**. From this we conclude that our assumed α that contradicts (10) cannot exist. Hence (10) holds.

This concludes the proof of Lemma 1. $\square$

For $j = sz$ Lemma 1 gives us

$$\begin{aligned} \psi^i &\implies \varphi^{i,sz} \\ \varphi^{i,sz} &\implies \exists[E^{i,sz}].\ \psi^i . \end{aligned}$$

Proposition 2 has now been proven, and consequently also Proposition 1. $\square$

The importance of the elimination order Recall that Alg. 1 and 2 require that the order of elimination is consistent for different values of i. We begin by showing an example where without this requirement equisatisfiability is not maintained. We will then show where the proof would fail had the order was not maintained.

Example 1. Consider the satisfiable formula:

$$\psi^0 = (v_1 \bar{v_2} v_3)(\bar{v_1} \bar{v_2} \bar{v_3})(v_1 \bar{v_3})(v_2) .$$

[3] This argument is only true because of the consistent order in which variables are reeliminated. We discuss this further in the end of this note.

Elimination of v_3 yields

$$(v_1\bar{v_2})(v_2)\ ,$$

and then eliminating v_2 yields

$$(v_1)\ .$$

So far we have

$$S_3 = (v_1\bar{v_2}v_3)(\bar{v_1}\bar{v_2}\bar{v_3})(v_1\bar{v_3})$$
$$S_2 = (v_1\bar{v_2})(v_2)$$

We now add the formula

$$\Delta_1^v = (\bar{v_1}v_3)$$

which is unsatisfiable if conjoined with the original formula ψ^0. This brings us to

$$(v_1)(\bar{v_1}v_3)$$

We now reeliminate v_2, and then reintroduce v_3 (note the inconsistent order). The former has no impact, since the formula does not contain the variable v_2. On the other hand reintroducing v_3 adds S_3, which leaves us with:

$$(v_1)(\bar{v_1}v_3)(v_1\bar{v_2}v_3)(\bar{v_1}\bar{v_2}\bar{v_3})(v_1\bar{v_3})$$

This formula is satisfied by the assignment $(v_1\bar{v_2}v_3)$. Hence equisatisfiability is not maintained. $\square$

The proof relies on the consistent order in several places. For example, after (14) we relied on an assumption that $\mathit{Removed}_{\geq j+1}$ does not include v_j when reeliminating v_j, which is only guaranteed to be true because of the consistent order. The following example demonstrates this fact. In layer $i-1$ v_j is reintroduced, which brings back a clause $c = (v_j, v_{j+1})$. Then v_{j+1} is reeliminated, and now S_{j+1} contains c. In layer i, v_j is eliminated again (in line 10) and hence pushed to ElimVarQ after $j+1$. But suppose that we do not maintain this order, and now try to reeliminate v_j before v_{j+1}. Now we are in a state that we reeliminate v_j whereas v_j is in S_{j+1} and hence in $\mathit{Removed}_{\geq j+1}$. Recall our argument right after (14): "Since v_j only appears in $(\varphi_j^{i,j-1} \wedge S_j)$ ..."; this is no longer true when reeliminaing v_j, since $\mathit{Removed}_{\geq j+1}$ contains j.

References

1. Alexander Nadel, Vadim Ryvchin, and Ofer Strichman. Preprocessing in incremental SAT. In *SAT'12*, 2012.