

Transforming DPLL to resolution

Citation for published version (APA):

Tveretina, O., & Zantema, H. (2002). *Transforming DPLL to resolution*. (Computer science reports; Vol. 0207). Technische Universiteit Eindhoven.

Document status and date:

Published: 01/01/2002

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Transforming DPLL to Resolution

Olga Tveretina, Hans Zantema

Technical University of Eindhoven

o.tveretina@tue.nl, h.zantema@tue.nl

Abstract

Standard techniques for proving unsatisfiability of propositional formulas include resolution and DPLL. It is well-known that a DPLL refutation straightforwardly transforms to a resolution refutation of similar length. We give a transformation of a DPLL refutation to a resolution refutation of a number of steps which is essentially less than the number of unit resolution steps applied in the DPLL refutation. We prove that our bounds are tight: for some CNFs we prove that no shorter resolution refutation exists than we give by our transformation.

Keywords: propositional logic, satisfiability, DPLL, resolution

1 Introduction

The satisfiability problem (SAT) is to determine for a given propositional formula whether there exists a satisfying assignment, i.e., whether it is not equivalent to false. In our paper we consider propositional formulas in *conjunctive normal form (CNF)*.

One of the simplest and most widely investigated method is resolution [ROB65]. It is used for refuting unsatisfiable CNF formulas. It consists of the single rule $V \vee p, W \vee \neg p \vdash V \vee W$.

A related method is the DPLL procedure [DLL62]. It consists of a combination of unit resolution (the special case of resolution where V or W is empty) and doing case analysis upon p and $\neg p$ and going on recursively. It is well-known ([GOL79],[PUD98]) that a proof that a CNF is unsatisfiable by DPLL can be transformed to a resolution refutation of linear length in the size of the DPLL proof. More precisely, the length of the resulting resolution refutation does not exceed the total number of unit resolution steps executed in the DPLL procedure.

In this paper we give a formal description of resolution and analyze the above relation between DPLL and resolution in more detail, and improve the results. Our improvement has no concern with the order of magnitude. In particular, we prove that if in the DPLL procedure s unit resolution steps are executed and r recursive calls are done, a resolution refutation exists of length at most $s - r$, being essentially better than s . Crucial for this result is the observation that if l is the only unit clause in a CNF V admitting a unit

resolution refutation of n steps, then $\neg l$ can be derived from $V \setminus \{l\}$ in at most $n - 2$ resolution steps. This result is constructive; apart from the proof as is given in this paper we also implemented the procedure transforming a DPLL proof to a resolution refutation of the given length.

We even prove for suitable formulas that no shorter resolution refutation exists than we give by our transformation, so our bounds are tight.

The paper is organized as follows. In section 2 we give some definitions, preliminary lemmas and the basic theorem. In section 3 we give the definition of a DPLL tree such that the nodes correspond to recursive calls in the DPLL procedure. Section 4 contains the main results about the length of the resolution refutation which are presented in two theorems. In section 5 we show that the upper bound is tight. Section 6 contains an overview of related work and section 7 contains some conclusions.

2 Basic definitions and Preliminaries

This section contains notations and definitions used throughout the paper.

Let A be a set of atoms. The symbols p , q and r are reserved to denote atoms.

A *literal* l is an atom or a negated atom, as $\neg p$.

A *clause* C is a finite, possibly empty, set of literals. The number of literals in the clause is the *length* of this clause. Clauses of length one are called *unit clauses*. The empty clause is denoted by \perp .

A *CNF* is a finite, possibly empty, set of clauses. Letters V, W are reserved to denote CNFs.

Suppose V_1, V_2 are CNFs; $C \cup \{p\}, D \cup \{\neg p\} \in V_1$, where $p \in A$. Then the transition from V_1 to V_2 , where $V_2 = V_1 \cup \{C \cup D\}$ is called a resolution step and denoted as $V_1 \xrightarrow{CDp} V_2$. C, D, p can be omitted in the context where they are not relevant.

If $C = \emptyset$ or $D = \emptyset$ then the resolution step is called a unit resolution step. We use the notation $V_1 \xrightarrow{l} V_2$ for the unit resolution step, where l is either p if $C = \emptyset$ or $\neg p$ if $D = \emptyset$. This unit resolution step is called an *l -step*.

If $V_0 \mapsto \dots \mapsto V_n$ then V_0, \dots, V_n is called a resolution sequence of *length* n . If $V_0 \xrightarrow{l_1} \dots \xrightarrow{l_n} V_n$ then V_0, \dots, V_n is called a unit resolution sequence.

Suppose V_0 is a CNF and C is a clause. We say that C is *derived* from V_0 in $n > 0$ resolution steps if there is a resolution sequence V_0, \dots, V_n such that $V_n = V_{n-1} \cup \{C\}$. And we say that C is derived from V_0 in 0 resolution steps if $C \in V_0$.

An assignment is a mapping that assigns *false* or *true* to atoms. If the atom p is assigned to true then $\neg p$ is assigned to false and vice versa. An assignment *satisfies* a clause if it maps at least one of its literals to true. An assignment satisfies a CNF V if and only if it *satisfies* each of its clauses, and V is called *satisfiable*. If there is no assignment that maps a CNF V to true then V is called *unsatisfiable*.

A resolution sequence V_0, \dots, V_n such that $\perp \in V_n$ is called a *resolution refutation* and n is called a *length of the resolution refutation*.

In the following we use the well-known fact that the *CNF* V is unsatisfiable if there is a resolution refutation starting from V .

We introduce some notation.

Let V be a *CNF* and l be a *literal*.

Then $\mathcal{V}(V, l) = \{W : \forall C \in V \exists D \in W : D = C \vee D = C \cup \{l\}\}$.

At first we present the lemmas we use in the following.

The main observation of the lemmas that if the empty clause can be derived from some *CNF* $V \cup \{\{l\}\}$ in n resolution steps then under some conditions $\neg l$ can be derived from V in less number of steps.

Lemma 1 *Suppose $V_0 \mapsto V_1$ and $W_0 \in \mathcal{V}(V_0, l)$. Then $\exists W_1 \in \mathcal{V}(V_1, l)$ such that $W_0 \mapsto W_1$.*

Proof. From $W_0 \in \mathcal{V}(V_0, l)$ it follows that $\forall C \in V_0 \exists D \in W_0$ such that $D = C$ or $D = C \cup \{l\}$.

As $V_0 \mapsto V_1$ then $\exists C_1 \cup \{p\}, C_2 \cup \{\neg p\} \in V_0$ such that $V_1 = V_0 \cup \{C_1 \cup C_2\}$.

By definition either $C_1 \cup \{p\} \in W_0$ or $C_1 \cup \{p\} \cup \{l\} \in W_0$, and either $C_2 \cup \{\neg p\} \in W_0$ or $C_2 \cup \{\neg p\} \cup \{l\} \in W_0$.

1. Let $C_1 \cup \{p\}, C_2 \cup \{\neg p\} \in W_0$.

Let us choose $W_1 = W_0 \cup \{C_1 \cup C_2\}$. Clearly $W_0 \mapsto W_1$.

We check that $W_1 \in \mathcal{V}(V_1, l)$.

Let $C \in V_1$.

If $C = C_1 \cup C_2$ then there exists $D \in W_1$ such that $D = C$.

If $C \in V_0$ then by the lemma conditions there exists $D \in W_0$ such that $D = C$ or $D = C \cup \{l\}$. As $W_0 \subseteq W_1$ then there exists $D \in W_1$ such that $D = C$ or $D = C \cup \{l\}$.

2. Let us in all other cases choose $W_1 = W_0 \cup \{C_1 \cup C_2 \cup \{l\}\}$.

We check that $W_1 \in \mathcal{V}(V_1, l)$.

Let $C \in V_1$.

If $C = C_1 \cup C_2$ then there exists $D \in W_1$ such that $D = C \cup \{l\}$.

If $C \in V_0$ then by the lemma conditions there exists $D \in W_0$ such that $D = C$ or $D = C \cup \{l\}$. As $W_0 \subseteq W_1$ then there exists $D \in W_1$ such that $D = C$ or $D = C \cup \{l\}$.

□

Lemma 2 *Suppose $V_0 \mapsto \dots \mapsto V_n$ and $W_0 \in \mathcal{V}(V_0, l)$. Then there exists $W_0 \mapsto \dots \mapsto W_n$ such that $\forall i \in \{1, \dots, n\} W_i \in \mathcal{V}(V_i, l)$.*

Proof. Induction on n .

Base case. The lemma holds for $n = 1$ by Lemma 1.

Inductive step. Let the lemma hold for $n - 1$.

Then $W_{n-1} \in \mathcal{V}(V_{n-1}, l)$. By Lemma 1 there exists $W_n \in \mathcal{V}(V_n, l)$ such that $W_{n-1} \mapsto W_n$. \square

Lemma 3 *Suppose $V \cup \{\{l\}\} = V_0 \xrightarrow{l} V_1 \mapsto \dots \mapsto V_n$. Then there exists $V \mapsto W_1 \mapsto \dots \mapsto W_{n-1}$ such that $\forall i \in \{1, \dots, n-1\} W_i \in \mathcal{V}(V_{i+1}, \neg l)$.*

Proof. The lemma holds by Lemma 2.

Lemma 4 *Suppose $V \cup \{\{l\}\} = V_0 \mapsto V_1 \mapsto \dots \mapsto V_n$ and $\exists \{i_1, \dots, i_m\} \subseteq \{0, \dots, n-1\}$ s.t. $\forall j \in \{i_1, \dots, i_m\} V_j \xrightarrow{l} V_{j+1}$. Then there exists $V \mapsto W_1 \mapsto \dots \mapsto W_{n-m}$ such that $W_{n-m} \in \mathcal{V}(V_n, \neg l)$.*

Proof. Induction on m .

Base case. $m = 1$. The lemma holds by Lemma 3.

Inductive step. Let the lemma hold for $m - 1$. Then the lemma holds for m by Lemma 2 and Lemma 3. \square

Lemma 4 is needed for proving Lemma 5.

Lemma 5 *If the empty clause can be derived from $V \cup \{\{l\}\}$ in m resolution steps then either the empty clause can be derived from V in at most m resolution steps or $\neg l$ can be derived from V in at most $m - 1$ resolution steps.*

Proof. Let $V_0 = V \cup \{\{l\}\}$. By the lemma assumption there exists $V_0 \mapsto \dots \mapsto V_m$ such that $\perp \in V_m$.

Then one of the following holds.

1. There exists $V \mapsto V_1 \mapsto \dots \mapsto V_m$ such that $\perp \in V_m$. Then \perp can be derived in m resolution steps from V .
2. For $k > 0 \exists 0 \leq i_1 < \dots < i_k < m$ such that $\forall j \in \{i_1, \dots, i_k\} V_j \xrightarrow{l} V_{j+1}$. Then by Lemma 4 there exists $V \mapsto W_1 \mapsto \dots \mapsto W_{m-k}$ such that $W_{m-k} \in \mathcal{V}(V_m, \neg l)$. As $\perp \in V_m$ then by Lemma 4 either $\neg l \in W_{m-k}$ or $\perp \in W_{m-k}$. And either $\neg l$ or \perp can be derived from V in $m - k \leq m - 1$ resolution steps.

\square

Now we prove the basic theorem we use in the following section.

Theorem 6 *If the empty clause can be derived from $V \cup \{\{l\}\}$ in $m > 0$ resolution steps, and the empty clause can be derived from $V \cup \{\{\neg l\}\}$ in $n > 0$ resolution steps then the empty clause can be derived from V in at most $m + n - 1$ resolution steps.*

Proof. By Lemma 5 the empty clause can be derived from V either in $\min(m, n)$ or in $m - 1 + n - 1 + 1 = m + n - 1$ resolution steps. In both cases we have the number of resolution steps no more than $m + n - 1$. \square

Example 7 We consider a CNF V .

$$V = \{\{p_1, p_2, p_3\}, \{\neg p_2, p_3\}, \{p_1, \neg p_3\}, \{\neg p_1, p_4\}, \{\neg p_4, p_5\}, \{\neg p_4, \neg p_5\}\}.$$

There exists $V \cup \{\{\neg p\}\} \vdash V_1 \vdash V_2 \vdash V_3 \vdash V_4$, where

$$V_1 = V \cup \{\{\neg p\}\} \cup \{\{p_2, p_3\}\}, V_2 = V_1 \cup \{\{p_3\}\},$$

$$V_3 = V_2 \cup \{\{\neg p_3\}\}, V_4 = V_3 \cup \{\perp\}.$$

There exists $V \cup \{\{p\}\} \vdash V_1 \vdash V_2 \vdash V_3 \vdash V_4$, where

$$V_1 = V \cup \{\{p\}\} \cup \{\{p_4\}\}, V_2 = V_1 \cup \{\{p_5\}\},$$

$$V_3 = V_2 \cup \{\{\neg p_5\}\}, V_4 = V_3 \cup \{\perp\}.$$

By Theorem 6 there exists a resolution derivation of length at most 7.

There exists $V \vdash V_1 \vdash V_2 \vdash V_3 \vdash V_4 \vdash V_5 \vdash V_6$, where

$$V_1 = V \cup \{\{p_1, p_3\}\}, V_2 = V_1 \cup \{\{p_1\}\}, V_3 = V_2 \cup \{\{\neg p_1, p_5\}\},$$

$$V_4 = V_3 \cup \{\{\neg p_1, \neg p_5\}\}, V_5 = V_4 \cup \{\{\neg p_1\}\}, V_6 = V_5 \cup \{\perp\}.$$

3 A DPLL tree

Given a CNF V and a literal l . Let $V|l$ denotes the formula obtained from V by removing all the clauses that contain l and deleting all $\neg l$ from all clauses that contain $\neg l$.

A literal l in the CNF V is called monotone if $\neg l$ does not appear in V .

The procedure of deleting monotone literals from V is denoted as $mon_lit(V)$.

A DPLL tree T on V is a binary tree, where every node is labelled with a unit resolution sequence. The root is labelled with a unit resolution sequence starting from V . If a node is labelled with the unit resolution sequence V_1, \dots, V_n then the left child is labelled with a unit resolution sequence starting from $mon_lit(V_n|p)$ and the right child is labelled with a unit resolution sequence starting from $mon_lit(V_n|\neg p)$. A node is a leaf if either $V_n = \emptyset$ or $\perp \in V_n$.

A DPLL tree is nothing but a static representation of the recursive calls in the executing of the usual DPLL procedure.

If T is a DPLL tree on V then V is satisfiable if and only if there exists a leaf in T labelled with a unit resolution sequence V_1, \dots, V_n such that $V_n = \emptyset$.

Hence building a DPLL tree implies decision procedure for satisfiability, therefore we will speak about DPLL proof rather than DPLL tree.

A DPLL tree on unsatisfiable formula is called a DPLL refutation.

Suppose a node is labelled with $mon_lit(V|l) \vdash V_1 \vdash \dots \vdash V_n$. Then the number of unit resolution steps corresponding to the node is defined to include the length of the unit resolution sequence n and the number of $\neg l$ in V .

The total number of unit resolution steps for the DPLL tree is called the length of the DPLL proof.

4 Upper bounds on resolution refutation length

In this section we give two upper bounds on resolution refutation length measured in the length of the DPLL refutation and the number of its nodes. The first one is a direct analysis of a DPLL refutation. The second one has an extra restriction on a resolution sequence used in the first result. The second bound is stronger, we even show that it will be tight.

Theorem 8 *Suppose V is an unsatisfiable CNF; a DPLL refutation on V has length s and the number of its nodes is r . Then there exists a resolution refutation on V of length less or equal $s - (r - 1)/2$.*

Proof. Induction on r .

Base case. Let $r = 1$. Then $s - (1 - 1)/2 = s$. The lemma holds.

Inductive step. Assume that the Lemma holds for $r - 2$. By induction hypothesis the lemma holds for the subtrees rooted at children nodes of the root.

Let one subtree have a DPLL refutation of length s_1 and the number of its nodes be r_1 . Let another subtree have a DPLL refutation of length s_2 and the number of its nodes be r_2 . And s_0 be a number of unit resolution steps corresponding to the root.

Then by Theorem 6 the length of a resolution refutation on V is $s_0 + ((s_1 - (r_1 - 1)/2) + (s_2 - (r_2 - 1)/2) - 1) = s - (r - 1)/2$, where $s = s_0 + s_1 + s_2$, $r = r_1 + r_2 + 1$. \square

Example 9 *We consider the pigeonhole formula P_n for $n = 2$.*

$$P_2 = \{\{p_{11}, p_{12}\}, \{p_{21}, p_{22}\}, \{p_{31}, p_{32}\}, \\ \{\neg p_{11}, \neg p_{21}\}, \{\neg p_{12}, \neg p_{22}\}, \{\neg p_{11}, \neg p_{31}\}, \\ \{\neg p_{12}, \neg p_{32}\}, \{\neg p_{21}, \neg p_{31}\}, \{\neg p_{22}, \neg p_{32}\}\}.$$

A DPLL refutation for P_2 is depicted in Figure 1.

The node 1 is labelled with P_2 .

The node 2 is labelled with $P_2^2 \mapsto P_2^2 \cup \{\{p_{22}\}\}$, where $P_2^2 = \{\{p_{21}, p_{22}\}, \{p_{31}, p_{32}\}, \{\neg p_{21}\}, \{\neg p_{12}, \neg p_{22}\}, \{\neg p_{31}\}, \{\neg p_{12}, \neg p_{32}\}, \{\neg p_{21}, \neg p_{31}\}, \{\neg p_{22}, \neg p_{32}\}\}$.

The node 3 is labelled with $P_2^3 \mapsto P_2^3 \cup \{\{p_{32}\}\} \mapsto P_2^3 \cup \{\{p_{32}\}\} \cup \{\perp\}$, where $P_2^3 = \{\{p_{31}, p_{32}\}, \{\neg p_{21}\}, \{\neg p_{12}\}, \{\neg p_{31}, \{\neg p_{12}, \neg p_{32}\}, \{\neg p_{21}, \neg p_{31}\}, \{\neg p_{32}\}\}$.

The node 4 is labelled with P_2^4 , where $P_2^4 = \{\{p_{21}\}, \{p_{31}, p_{32}\}, \{\neg p_{21}\}, \{\neg p_{31}\}, \perp\}$.

The node 5 is labelled with $P_2^5 \mapsto P_2^5 \cup \{\{\neg p_{32}\}\}$, where $P_2^5 = \{\{p_{12}\}, \{p_{21}, p_{22}\}, \{p_{31}, p_{32}\}, \{\neg p_{12}, \neg p_{22}\}, \{\neg p_{12}, \neg p_{32}\}, \{\neg p_{21}, \neg p_{31}\}, \{\neg p_{22}, \neg p_{32}\}\}$.

The node 6 is labelled with P_2^6 , where

$$P_2^6 = \{\{p_{12}\}, \{p_{31}, p_{32}\}, \{\neg p_{12}\}, \{\neg p_{12}, \neg p_{32}\}, \{\neg p_{21}, \neg p_{31}\}, \{\neg p_{32}\}\}.$$

The node 7 is labelled with P_2^7 , where $P_2^7 = \{\{p_{12}\}\{\neg p_{12}\}, \{\neg p_{21}, \neg p_{31}\}, \perp\}$.

The node 8 is labelled with $P_2^8 \mapsto P_2^8 \cup \perp$, where $P_2^8 = \{\{p_{12}\}, \{p_{31}\}, \{\neg p_{12}\}, \{\neg p_{21}, \neg p_{31}\}\}$.

The node 9 is labelled with $P_2^9 \mapsto P_2^9 \cup \{\{p_{31}\}\} \mapsto P_2^9 \cup \{\{p_{31}\}\} \cup \{\{\neg p_{21}\}\}$

$\mapsto P_2^9 \cup \{\{p_{31}\}\} \cup \{\{\neg p_{21}\}\} \cup \{\perp\}$,
 where $P_2^9 = \{\{p_{12}\}, \{p_{21}\}, \{p_{31}, p_{32}\}, \{\neg p_{12}, \neg p_{32}\}, \{\neg p_{21}, \neg p_{31}\}, \{\neg p_{32}\}\}$.

The DPLL refutation on P_2 has $s = 20$ unit resolution steps and $r = 9$ nodes. Then by Theorem 8 there exists the resolution refutation on P_2 of length at most 16.

Figure 1: A DPLL refutation on P_2

We can improve the upper bound by making a restriction on the unit resolution sequences associated with nodes of the DPLL refutation.

We say that a unit resolution sequence is *complete* if no unit resolution steps can be applied at the last CNF of the sequence.

For the proof of the theorem we use the following lemmas.

Lemma 10 *Let V be a CNF such that it contains no monotone literals, and no unit resolution can be applied. Then V contains no unit clauses.*

Proof. By contradiction.

Let $V = \{\{l\}\} \cup V'$, where V' is a CNF. By the lemma assumption V contains no monotone literals so $\exists C \in V' : \neg l \in C$. And at least one unit resolution step can be applied. We have a contradiction. And V contains no unit clauses. \square

Now we arrive at the crucial observation mentioned in the introduction.

Lemma 11 *If \perp can be derived from $V \cup \{\{l\}\}$ in $n \geq 2$ unit resolution steps, and V contains no unit clauses then either $\neg l$ or \perp can be derived from V in at most $n - 2$ resolution steps.*

Proof. Induction on n .

Base case. $n = 2$.

As V does not contain monotone literals so \perp cannot be derived from V in two unit resolution steps. And the lemma holds for $n = 2$.

Inductive step. Let the lemma hold for $n - 1$.

Suppose \perp can be derived from $V \cup \{\{l\}\}$ in $n \geq 2$ unit resolution steps.

If the unit resolution sequence contains more than one l -step then the lemma holds by Lemma 4.

As V contains no unit clauses the first unit resolution step is an l -step. So the remaining case is if this first step is the only l -step.

Then $V \cup \{\{l\}\} \xrightarrow{l} V \cup \{\{l\}\} \cup \{\{l'\}\}$. And \perp can be derived from $V \cup \{\{l\}\} \xrightarrow{u} V \cup \{\{l\}\} \cup \{\{l'\}\}$ in $n - 1$ unit resolution steps.

As the resolution sequence contains only one l -step then \perp can be derived from $V \cup \{\{l'\}\}$ in $n - 1$ unit resolution steps.

By induction hypothesis either $\neg l'$ or \perp can be derived from $V \cup \{\{l'\}\}$ in no more than $n - 3$ resolution steps. As $\{\neg l, l'\} \in V$ then we need one extra resolution step in case if we derived $\neg l'$. And either $\neg l$ or \perp can be derived from V in $n - 2$ resolution steps. \square

Example 12 Let $V = \{\{\neg p_1, p_2\}, \{\neg p_2, p_3\}, \{\neg p_2, p_4\}, \{\neg p_3, \neg p_4\}\}$ and $l = p_1$.

There exists the derivation of the empty clause from $V \cup \{\{p_1\}\}$ in 5 unit resolution steps $V \cup \{\{p_1\}\} \mapsto V_1 \mapsto V_2 \mapsto V_3 \mapsto V_4 \mapsto V_5$, where

$$V_1 = V \cup \{\{p_2\}\}, V_2 = V_1 \cup \{\{p_3\}\}, V_3 = V_2 \cup \{\{p_4\}\}, V_4 = V_3 \cup \{\{\neg p_4\}\}, V_5 = V_4 \cup \{\perp\}.$$

There exists the derivation of $\{\neg p_1\}$ from V in 3 resolution steps $V \mapsto V_1 \mapsto V_2 \mapsto V_3$, where

$$V_1 = V \cup \{\{\neg p_2, \neg p_4\}\}, V_2 = V_1 \cup \{\{\neg p_2\}\}, V_3 = V_2 \cup \{\{\neg p_1\}\}.$$

Just like Theorem 6 was needed to prove Theorem 8 we now state Theorem 13 to use it for proving Theorem 14.

Theorem 13 Suppose V contains no unit clauses. If \perp can be derived from $V \cup \{\{l\}\}$ in $m > 2$ unit resolution steps, and \perp can be derived from $V \cup \{\{\neg l\}\}$ in $n > 2$ unit resolution steps then \perp can be derived from V in $m + n - 3$ resolution steps.

Proof. If \perp can be derived from $V \cup \{\{l\}\}$ in $m > 2$ resolution steps then by Lemma 11 one of the following holds

1. \perp can be derived from V in at most $m - 2$ resolution steps.
2. $\neg l$ can be derived from V in at most $m - 2$ resolution steps.

If \perp can be derived from $V \cup \{\{\neg l\}\}$ in $n > 2$ resolution steps then by Lemma 11 one of the following holds

3. \perp can be derived from V in at most $n - 2$ resolution steps.
4. l can be derived from V in at most $m - 2$ resolution steps.

In cases 1 or 3 \perp can be derived from V in $\min(m - 2, n - 2)$ resolution steps. For $m > 2$ and $n > 2$ $\min(m - 2, n - 2) \leq m + n - 3$. In case of combination of cases 2 and 4 we need extra resolution step to get \perp . And it can be derived in $m - 2 + n - 2 + 1 = m + n - 3$ resolution steps. \square

Theorem 14 *Suppose V is an unsatisfiable CNF; a DPLL refutation on V has length s , the number of its nodes is $r \geq 3$ and every unit resolution sequence associated with a node is complete. Then there exists a resolution refutation on V of length less or equal $s - r$.*

Proof. Induction on r .

Base case. Let $r = 3$. Then the Lemma holds by Lemma 10 and Theorem 13.

Inductive step. Assume that the Lemma holds for $r - 2$. By induction hypothesis the lemma holds for the subtrees rooted at children nodes of the root.

Let one subtree have a DPLL refutation of length s_1 and the number of its nodes be r_1 . Let another subtree have a DPLL refutation of length s_2 and the number of its nodes be r_2 . And s_0 be a number of unit resolution steps corresponding to the root.

Then by Theorem 6 the length of a resolution refutation on V is $s_0 + ((s_1 - r_1) + (s_2 - r_2) - 1) = s - r$, where $s = s_0 + s_1 + s_2$, $r = r_1 + r_2 + 1$. \square

Using the result a DPLL refutation can be transformed to a resolution refutation of shorter length.

Example 15 *We consider the pigeonhole formula P_2 from Example 9. The DPLL refutation on P_2 is depicted on Figure 2. It satisfies Theorem 14 conditions.*

A DPLL refutation on P_2 is depicted in Figure 2.

Figure 2: A DPLL refutation on P_2

The node 1 is labelled with P_2 .

The node 2 is labelled with $P_2^2 \mapsto P_2^2 \cup \{\{p_{22}\}\} \mapsto P_2^2 \cup \{\{p_{22}\}\} \cup \{\{\neg p_{12}\}\} \mapsto P_2^2 \cup \{\{p_{22}\}\} \cup \{\{\neg p_{12}\}\} \cup \{\{p_{32}\}\} \mapsto P_2^2 \cup \{\{p_{22}\}\} \cup \{\{\neg p_{12}\}\} \cup \{\{p_{32}\}\} \cup \{\{p_{31}\}\} \mapsto P_2^2 \cup \{\{p_{22}\}\} \cup \{\{\neg p_{12}\}\} \cup \{\{p_{32}\}\} \cup \{\{p_{31}\}\} \cup \{\{\perp\}\}$, where $P_2^2 = \{\{p_{21}, p_{22}\}, \{p_{31}, p_{32}\}, \{\neg p_{21}\}, \{\neg p_{12}, \neg p_{22}\}, \{\neg p_{31}\}, \{\neg p_{12}, \neg p_{32}\}, \{\neg p_{21}, \neg p_{31}\}, \{\neg p_{22}, \neg p_{32}\}\}$.

The node 3 is labelled with $P_2^3 \mapsto P_2^3 \cup \{\{p_{22}\}\} \mapsto P_2^3 \cup \{\{\neg p_{22}\}\} \cup \{\{\neg p_{32}\}\} \mapsto P_2^3 \cup \{\{\neg p_{22}\}\} \cup \{\{\neg p_{32}\}\} \cup \{\{p_{21}\}\} \mapsto P_2^3 \cup \{\{\neg p_{22}\}\} \cup \{\{\neg p_{32}\}\} \cup \{\{p_{21}\}\} \cup \{\{\neg p_{22}\}\} \cup \{\{\neg p_{32}\}\} \cup \{\{p_{21}\}\} \cup \{\{p_{31}\}\} \mapsto P_2^3 \cup \{\{\neg p_{22}\}\} \cup \{\{\neg p_{32}\}\} \cup \{\{p_{21}\}\} \cup \{\{p_{31}\}\} \cup \{\{\neg p_{31}\}\} \cup \{\{\perp\}\}$, where $P_2^3 = \{\{p_{12}\}, \{p_{21}, p_{22}\}, \{p_{31}, p_{32}\}, \{\neg p_{12}, \neg p_{22}\}, \{\neg p_{12}, \neg p_{32}\}, \{\neg p_{21}, \neg p_{31}\}, \{\neg p_{22}, \neg p_{32}\}\}$.

Example 16 Let $V = \{\{p, q, r\}, \{p, q, \neg r\}, \{p, \neg q, r\}, \{p, \neg q, \neg r\}, \{\neg p, q, r\}, \{\neg p, q, \neg r\}, \{\neg p, \neg q, r\}, \{\neg p, \neg q, \neg r\}\}$.

The DPLL refutation on V is depicted on Figure 3

The root of the DPPL refutation is labelled with V .

The node 2 is labelled with $V_n|p = \{q, r\}, \{q, \neg r\}, \{\neg q, r\}, \{\neg q, \neg r\}$.

The node 3 is labelled with $V_n|\neg p = \{q, r\}, \{q, \neg r\}, \{\neg q, r\}, \{\neg q, \neg r\}$.

The nodes 4, 5, 6, 7 are labelled with a resolution sequence $\{\{r\}, \{\neg r\}\} \mapsto \{\{r\}, \{\neg r\}, \perp\}$.

By definition the number of unit resolution steps for the DPLL refutation on V is 20, and it has 7 nodes.

By Theorem 14 there exists a resolution refutation of length no more than 13.

There exists the resolution refutation $V \mapsto V_1 \mapsto V_2 \mapsto V_3 \mapsto V_4 \mapsto V_5 \mapsto V_6 \mapsto V_7$, where

$$V_1 = V \cup \{\{p, q\}\}, V_2 = V_1 \cup \{\{p, \neg q\}\},$$

$$V_3 = V_2 \cup \{\{\neg p, q\}\}, V_4 = V_3 \cup \{\{\neg p, \neg q\}\}, V_5 = V_4 \cup \{\{p\}\}$$

$$V_6 = V_5 \cup \{\{\neg p\}\}, V_7 = V_6 \cup \{\perp\}.$$

Figure 3: A DPLL refutation on V

5 Tightness of the upper bound

One can wonder whether Theorem 14 can be improved further. In this section by analyzing of some CNFs class we show that it cannot be done.

A CNF is *minimally unsatisfiable* if it is unsatisfiable and each of its subsets is satisfiable.

Let V_0, \dots, V_n be a resolution sequence, where $V_n = \{C_1, \dots, C_m\}$. Then a directed graph $G(V, E)$, where V is a set of vertices and E is a set of edges is called a *resolution graph* if $V = \{C_1, \dots, C_m\}$ and $E = \{(C_i, C_j) : \exists r \in \{0, \dots, n-1\}, l \in L, C_k \in V_r \text{ such that } V_{r+1} = V_r \cup C_j, \text{ where } C_j = (C_i \cup C_k) \setminus \{l, \neg l\}\}$.

Lemma 17 *Suppose minimally unsatisfiable CNF V contains n clauses. Then there is no resolution refutation on V with length less than $n - 1$.*

Proof. By contradiction.

Let $V \mapsto V_1 \mapsto \dots \mapsto V_r$ be a resolution refutation on V , and $G(V^*, E^*)$ be a resolution graph on $V \mapsto V_1 \mapsto \dots \mapsto V_r$.

By construction, $|V^*| = n + r$, $|E^*| = 2r$.

Let $r < n - 1$. Then $|E^*| < |V^*| - 1$ and G is a disconnected graph.

If G is a disconnected graph then by its construction $\exists C \in V$ such that there is no path from C to \perp on G . And \perp can be derived from $V \setminus C$. We have a contradiction. \square

We introduce some class of CNFs to prove that the upper bound presented by Theorem 14 is tight.

Let $n \geq 1$. We define $V_n = \{\{\neg p_1, q_1\}, \{\neg p_2, \neg q_1, q_2\}, \dots, \{\neg p_n, \neg q_{n-1}, q_n\}, \{\neg q_n, q_{n+1}\}, \{\neg q_n, q_{n+2}\}, \{\neg q_{n+1}, \neg q_{n+2}\}, \{p_1, r_{11}\}, \{\neg r_{11}, r_{21}\}, \{\neg r_{11}, r_{31}\}, \{\neg r_{31}, \neg r_{21}\}, \dots, \{p_n, r_{1n}\}, \{\neg r_{1n}, r_{2n}\}, \{\neg r_{1n}, r_{3n}\}, \{\neg r_{2n}, \neg r_{3n}\}\}$.

Lemma 18 *For $n \geq 1$ V_n is minimally unsatisfiable.*

Proof. We prove that V_n is minimally unsatisfiable by giving a satisfying assignment for W obtained by deleting a clause from V .

1. $W = V \setminus \{\neg p_1, q_1\}$
 $\forall i \in \{1, \dots, n\} p_i = 1, r_{1i} = 0, r_{2i} = 0, r_{3i} = 0; \forall i \in \{1, \dots, n+2\} q_i = 0.$
2. For $k \in \{2, \dots, n\} W = V \setminus \{\neg p_k, \neg q_{k-1}, q_k\}$
 $\forall i \in \{1, \dots, n\} p_i = 1, r_{1i} = 0, r_{2i} = 0, r_{3i} = 0; \forall i \in \{1, \dots, n+2\} \setminus \{k\} q_i = 0; q_k = 1.$
3. $W = V \setminus \{\neg q_n, q_{n+1}\}$
 $\forall i \in \{1, \dots, n\} p_i = 1, r_{1i} = 0, r_{2i} = 0, r_{3i} = 0; \forall i \in \{1, \dots, n+2\} \setminus \{n+1\} q_i = 1; q_{n+1} = 0.$
4. $W = V \setminus \{\neg q_n, q_{n+2}\}$
 $\forall i \in \{1, \dots, n\} p_i = 1, r_{1i} = 0, r_{2i} = 0, r_{3i} = 0; \forall i \in \{1, \dots, n+1\} q_i = 1; q_{n+2} = 0.$
5. $W = V \setminus \{\neg q_{n+1}, \neg q_{n+2}\}$
 $\forall i \in \{1, \dots, n\} p_i = 1, r_{1i} = 0, r_{2i} = 0, r_{3i} = 0; \forall i \in \{1, \dots, n+2\} q_i = 1.$
6. For $k \in \{1, \dots, n\} W = V \setminus \{p_k, r_{1k}\}$
 $\forall i \in \{1, \dots, n\} \setminus \{k\} p_i = 1; p_k = 0; \forall i \in \{1, \dots, n\} r_{1i} = 0, r_{2i} = 0, r_{3i} = 0; \forall q_i \in \{0, \dots, k-1\} q_i = 1; \forall q_i \in \{k, \dots, n+2\} q_i = 0;$
7. For $k \in \{1, \dots, n\} W = V \setminus \{\neg r_{1k}, r_{2k}\}$
 $\forall i \in \{1, \dots, n\} \setminus \{k\} p_i = 1; p_k = 0; \forall i \in \{1, \dots, n\} \setminus \{k\} r_{1i} = 0; r_{1k} = 1; \forall i \in \{1, \dots, n\} r_{2i} = 0; \forall i \in \{1, \dots, n\} \setminus \{k\} r_{3i} = 0; r_{3k} = 1; \forall q_i \in \{0, \dots, k-1\} q_i = 1; \forall q_i \in \{k, \dots, n+2\} q_i = 0.$

$$8. W = V \setminus \{\neg r_{1k}, r_{3k}\}$$

$$\forall i \in \{1, \dots, n\} \setminus \{k\} p_i = 1; p_k = 0; \forall i \in \{1, \dots, n\} \setminus \{k\} r_{1i} = 0; r_{1k} = 1; \forall i \in \{1, \dots, n\} \setminus \{k\} r_{2i} = 0; r_{2k} = 1; \forall i \in \{1, \dots, n\} r_{3i} = 0; \forall q_i \in \{0, \dots, k-1\} q_i = 1; \forall q_i \in \{k, \dots, n+2\} q_i = 0.$$

$$9. W = V \setminus \{\neg r_{2k}, \neg r_{3k}\}$$

$$\forall i \in \{1, \dots, n\} \setminus \{k\} p_i = 1; p_k = 0; \forall i \in \{1, \dots, n\} r_{1i} = 0; \forall i \in \{1, \dots, n\} \setminus \{k\} r_{2i} = 0; r_{2k} = 1; \forall i \in \{1, \dots, n\} \setminus \{k\} r_{3i} = 0; r_{3k} = 1; \forall q_i \in \{0, \dots, k-1\} q_i = 1; \forall q_i \in \{k, \dots, n+2\} q_i = 0.$$

□

Lemma 19 For $n \geq 1$ V_n has a DPLL refutation of length $7n + 3$ and the number of its nodes is $2n + 1$.

Proof. In Figure 4 the DPLL refutation for V_n is depicted.

Figure 4: A DPLL refutation on V_n

The node 1 is labelled with original CNF V_n .

For $i = 1, \dots, n-1$ the node $2i+1$ is labelled with the resolution sequence

$$W_{2i+1} \mapsto W_{2i+1} \cup \{r_{1i}\} \mapsto W_{2i+1} \cup \{r_{1i}\} \cup \{r_{2i}\}$$

$$\mapsto W_{2i+1} \cup \{r_{1i}\} \cup \{r_{2i}\} \cup \{r_{3i}\}$$

$$\mapsto W_{2i+1} \cup \{r_{1i}\} \cup \{r_{2i}\} \cup \{r_{3i}\} \cup \{\neg r_{3i}\}$$

$$\mapsto W_{2i+1} \cup \{r_{1i}\} \cup \{r_{2i}\} \cup \{r_{3i}\} \cup \{\neg r_{3i}\} \cup \perp,$$

where $W_{2i+1} = \{\{\neg p_i\}, \{\neg p_{i+2}, \neg q_{i+1}, q_{i+2}\}, \dots, \{\neg p_n, \neg q_{n-1}, q_n\}, \{\neg q_n, q_{n+1}\}, \{\neg q_n, q_{n+2}\}, \{\neg q_{n+1}, \neg q_{n+2}\}, \{p_1, r_{11}\}, \{\neg r_{11}, r_{21}\}, \{\neg r_{11}, r_{31}\}, \{\neg r_{31}, \neg r_{21}\}, \dots, \{p_n, r_{1n}\}, \{\neg r_{1n}, r_{2n}\}, \{\neg r_{1n}, r_{3n}\}, \{\neg r_{2n}, \neg r_{3n}\}\}$.

The node $2n+1$ is labelled with the resolution sequence

$$W_{2n+1} \mapsto W_{2n+1} \cup \{r_{1n}\} \mapsto W_{2n+1} \cup \{r_{1n}\} \cup \{r_{2n}\}$$

$$\mapsto W_{2n+1} \cup \{r_{1n}\} \cup \{r_{2n}\} \cup \{r_{3n}\}$$

$$\mapsto W_{2n+1} \cup \{r_{1n}\} \cup \{r_{2n}\} \cup \{r_{3n}\} \cup \{\neg r_{3n}\}$$

$\mapsto W_{2n+1} \cup \{r_{1n}\} \cup \{r_{2n}\} \cup \{r_{3n}\} \cup \{\neg r_{3n}\} \cup \perp$,
 where $W_{2n+1} = \{\{\neg p_n\}, \{\neg q_{n+1}, \neg q_{n+2}\}, \{p_1, r_{11}\}, \{\neg r_{11}, r_{21}\}, \{\neg r_{11}, r_{31}\},$
 $\{\neg r_{31}, \neg r_{21}\}, \dots, \{p_n, r_{1n}\}, \{\neg r_{1n}, r_{2n}\}, \{\neg r_{1n}, r_{3n}\}, \{\neg r_{2n}, \neg r_{3n}\}\}$.

For $i = 1, \dots, n - 1$ the node $2i$ is labelled with the CNF

$W_{2i} = \{\{\neg p_{2i+1}, q_{2i+1}\}, \{\neg p_{2i+2}, \neg q_{2i+1}, q_{2i+2}\}, \dots, \{\neg p_n, \neg q_{n-1}, q_n\} \{\neg q_n, q_{n+1}\},$
 $\{\neg q_n, q_{n+2}\}, \{\neg q_{n+1}, \neg q_{n+2}\}, \{p_1, r_{11}\}, \{\neg r_{11}, r_{21}\}, \{\neg r_{11}, r_{31}\}, \{\neg r_{31}, \neg r_{21}\}, \dots,$
 $\{p_n, r_{1n}\}, \{\neg r_{1n}, r_{2n}\}, \{\neg r_{1n}, r_{3n}\}, \{\neg r_{2n}, \neg r_{3n}\}\}$.

The node $2n$ is labelled with the resolution sequence

$W_{2n} \mapsto W_{2n} \cup \{\neg q_{n+2}\} \mapsto W_{2n} \cup \{\neg q_{n+2}\} \cup \perp$,
 where $W_{2n} = \{q_{n+1}\}, \{q_{n+2}\}, \{\neg q_{n+1}, \neg q_{n+2}\}, \{p_1, r_{11}\}, \{\neg r_{11}, r_{21}\}, \{\neg r_{11}, r_{31}\},$
 $\{\neg r_{31}, \neg r_{21}\}, \dots, \{p_n, r_{1n}\}, \{\neg r_{1n}, r_{2n}\}, \{\neg r_{1n}, r_{3n}\}, \{\neg r_{2n}, \neg r_{3n}\}$.

By definition of a DPLL refutation of a node is labelled with $mon_lit(V|l) \mapsto V_1 \mapsto \dots \mapsto V_n$. Then the number of unit resolution steps corresponding to the node includes the length of the unit resolution sequence n and the number of $\neg l$ in V .

And $\forall n \geq 1$ V_n has a DPLL refutation of length $7n + 3$ and the number of its nodes is $2n + 1$. \square

Now we are ready to prove tightness of our main result.

Theorem 20 *For $n \geq 1$ V_n has a DPLL refutation such that it has length s , the number of its nodes is r and there is no resolution refutation on V_n of length less than $s - r$.*

Proof. By Lemma 19 $\forall n \geq 1$ V_n has a DPLL refutation of length $7n + 3$ and the number of its nodes is $2n + 1$. By Theorem 14 there exists a resolution refutation on V of length less or equal $5n + 2$.

For $n \geq 1$ V_n has $5n + 3$ clauses. By Lemma 17 and Lemma 18 there is no resolution refutation on V of length less than $5n + 2$. \square

6 Related work

In the last decade we can observe significant progress in solving SAT problems. Many current propositional provers ([ZHG97],[SIS99]) are based on this procedure.

In [RIS00] Rish and Dechter use directional resolution (a form of ordered resolution) to search for SAT problems. Directional resolution identifies pairs of resolvable clauses quickly. A heuristic for choosing a good ordering was. They also showed that two hybrid algorithms combining resolution and search performed more efficiently than pure search .

Van Gelder [GEL95] replaces the unit propagation and pure literal operations of DPLL with a number of resolution and subsumption operations. It is pointed out that efficient data structures are extremely important in practical hybrid algorithms. They show that their hybrid algorithm performs better than DPLL, and the performance improvement is greater for more difficult problems.

The use of trees (discrimination trees) is considered in [ZHA97, MEG93]. De Kleer and H. Zhang used trees to represent propositional clauses for efficient subsumption. One of the major motivations for developing these approaches was to solve open problems in algebra concerning the existence of quasigroups satisfying certain constraints.

DDPP (Discrimination-tree-based Davis-Putman prover) is a straightforward implementation of the Davis-Putman method based on tree-merge operation [MEG93]. It performs the operation nondestructively, and the result shares (nearly) maximal structure with the original tree to minimize the memory allocation.

We can see in [GOM98] that randomization and restart strategies are very effective for backtrack search. They eliminate this phenomenon. The randomization is introduced into a backtrack algorithm bringing in certain randomness into branching heuristic, that has affect the selection of variables and their values for branching and backtrack in the search space. The randomization defines a cutoff value in some backtracks and repeatedly starts a randomized complete search procedure at the root of the search tree. These two techniques are very useful for solving some hard combinatorial problems, combining these technique eg. with lemma learning helps to solve hard real world satisfiably problems [BAP98].

In [ZHA00] a new random jump strategy which never cause any repetition of search performed by original search algorithm. It does not demand any change to branching heuristic. The space explored by backtrack search procedure can be represented by a tree where an internal node represents a backtrack point and a leaf node represents either a solution or no solution. At a check point author looks at the path from the current node to the root of the tree to see how many branches have been closed and how many branches are still open to estimate percentage of the remaining space. If the remaining space is sufficiently large they may jump up along the path skipping some open branches along the way. The checkpoints can be decided as the cutoff points in the restart strategy. If a restart strategy explores only one leaf node , then this special case of restart strategy is a variant of the limited discrepancy search proposed by Harvey and Ginsberg [HAR95a,HAR95b], and modified by Korf [KOR96] and Walsh [WAL97].

There has been interest in using resolution in combination with DPLL search, to reduce the amount of search required to solve SAT problems [GEL95,RIS00]. Resolution can shorten the size of the search tree, but it is only useful if the time spent on resolution is less than the time gained by reducing the search space.

In [DRA02] it is presented two techniques that combine resolution and DPLL search, and describe the relationship between them. The first technique , neighbour resolution, uses a restricted form of resolution during search, while the second uses a single level of binary resolution as a preprocessing step. It is shown that the preprocessing technique can cut the search tree as much as neighbour resolution during search.

7 Conclusions

We proved that if in the DPLL procedure s unit resolution steps are executed and r recursive calls are done, a resolution refutation can be constructed of length at most $s - r$.

We implemented this construction, and it turned out that for many formulas, including pigeon hole formulas, the constructed resolution refutation had a length that was much less than $s - r$. On the other hand we gave a class of formulas for which this bound $s - r$ is tight: we proved that no shorter resolution refutation exists. Since at every node DPLL allows freedom of how to choose the next atom it is difficult to draw general conclusions from experiments based on one particular choice.

References

[BAP98] Baptista, L., and Margues-Silva, J.P., Using Randomization and Learning to solve hard Real -World Instances of Satisfiability, in Proceeding of the 6th International Conference on principles and Practice of Constraint Programming (CP), September 2000.

[COM97] S. A. Cook and D. G. Mitchell. Finding hard instances of the satisfiability problem: a survey. Satisfiability Problem: Theory and Applications. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Volume 35, 1997.

[DAV60] Davis, M. and Putman, H. A computing procedure for quantification theory, Journal of the Association for Computing Machinery 7(3) (July 1960), 201-215.

[DLL62] Davis, M., Logemann, G., and Loveland, D. A machine program for theorem proving, Communications of the Association for Computing Machinery 5(7) (July 1962), 394-397.

[DRA02] Lyndon Drake, Alan Frisch and Toby Walsh, Adding resolution to the DPLL procedure for Boolean satisfiability. USA.2002.

[GOL79] A. Goldberg. Average Case Complexity of the Satisfiability Problem. In Proc. 4th Workshop on Automated Deduction, pp. 1-6, Austin Texas, 1979.

[GOM00] Gomes,C.P., Selman, B., Crato, N., and Kautz, H., Heavy-Tailed Phenomena in Satisfiability and Constraint Satisfaction Problem. J. of Automated Reasoning, vol. 24 (1/2), pages 67-100, 2000.

[HAK85] A.Haken. The intractability of resolution. Theoretical Computer Science, 39: 297-308, 1985.

[HAR95a] Harvey, W.D. and Ginsberg, M.L. Limited discrepancy search. In Proceedings of 14th IJCAI, 1995.

[HAR95b] Harvey, W.D. Nonsystematic backtracking search. PhD thesis, Stanford University, 1995.

[JO99] Joo P. Marques Silva and Karem A.Sakallah. GRASP : A search algorithm for propositional satisfiability. IEEE Transactions on Computers,48 (5), May 1999.

[KLE92] De Kleer, J, An improved incremental algorithm for generating prime implicants , in Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92) , San Jose, California, July 1992, pp.780-785.

[KOR96] Korf,R. Improved limited discrepancy search. In Proceedings of 13th AAAI, 1996.

[MEG93] Meglicki, G. Stickel's Davis-Putman engineered reversely, available by anonymous FTP from arp.anu.edu.au, Automated Reasoning Project, Australian National University, Canberra, Australia,1993.

[MOS01] M. Moskevycz, C. Madigan, Y. Zhao, L. Zhang, and S. Malik. Chaff: Engineering an efficient sat solver. In 39th design Automation Conferebce, Las Vegas, June 2001.

[RIS00] Irina Rish and Rina Dechter. Resolution versus search: Two strategies for SAT. In Ian Gent, Hans van Maaren, and Toby Walsh, editors, SAT2000: Highlights of Satisfiability Research in the 2000, volume 63 of Frontiers in Artificial Intelligence and Applications, pages 215-259, IOS press,2000.

[ROB65] J.A. Robinson. A machine-oriented logic based on resolution principle. Journal of the Association for Computing Machinery, 12(1): 23-41, 1965.

[WAL97] Walsh, T. Depth-bounded disperancy search. Proceedings of 15th IJCAI, 1997.

[YUG95] N. Yugami. Theoretical analysis of Davis-Putnam procedure and propositional satisfiability. In Proceedings of IJCAI-95. 282-288, 1995.

[ZHA97] Zhang, H. SATO: An efficient prepositional prover, in Proceedings of International Conference on Automated Deduction (CADE-97), Lecture Notes in Artificial Intelligence 1104, Spriger-Verlag, 1997, pp 308-312.

[ZHA00] Zhang, H., A Random Jump Strategy for Combinatorial Search. In

[ZHA00] Hantao Zhang and Mark E. Stickel. Implementing the Davis-Putman method. In Ian Gent, Hans van Maaren, and Toby Walsh, editors, SAT2000: Highlights of Satisfiability Research in the 2000, volume 63 of Frontiers in Artificial Intelligence and Applications, pages 309-326,IOS press,2000.