DD3501 Current Research in Proof Complexity Oct 27, 2011

Lecture 2

Lecturer: Jakob Nordström *Scribe: Joel Mickelin*

The objective of today's lecture is to introduce the concept of *proof width*. Width is an important concept for resolution for several reasons, but the main reason for us to want to focus on it now is that it turns out to be very helpful when we want to understand the fundamental measures of length and space in resolution.

1 Terminology and Notation Used Throughout These Notes

Let us start by recalling some of the terminology and notation introduced in the previous lecture:

- A *literal* a is a variable x or its negation $\overline{x}$.
- A *clause* $C = a_1 \lor \ldots \lor a_k$ is a set of literals joined by the connective $\lor$. A clause with at most k literals is known as a k*-clause.*
- A *CNF formula* $F = C_1 \land \ldots \land C_m$ is a set of clauses joined by the connective $\land$. A k*-CNF formula* is a CNF formula consisting of k-clauses.
- $\mathit{Vars}(\cdot)$ denotes the set of variables in a formula. $\mathit{Lit}(\cdot)$ denotes the set of literals in a formula.
- We write $F \vDash D$ to denote semantic implication. What this means is that for all truth value assignments α, if $\alpha(F)$ is true then $\alpha(D)$ is true.

2 Recap of Resolution and a Slightly Modified Definition

As described in the last lecture, a resolution derivation from a CNF formula F can be defined as a sequence of sets of clauses, or clause configurations, $\{\mathbb{D}_0, \mathbb{D}_1, \ldots, \mathbb{D}_\tau\}$. The derivation steps to go from a configuration $\mathbb{D}_i$ to the next configuration $\mathbb{D}_{i+1}$ are as follows:

Axiom Download $\mathbb{D}_t = \mathbb{D}_{t-1} \cup \{C\}$ where C is some clause in F.

Inference $\mathbb{D}_t = \mathbb{D}_{t-1} \cup \{D\}$ where D is a clause obtained by resolving two clauses in $\mathbb{D}_{t-1}$.

Erasure $\mathbb{D}_t = \mathbb{D}_{t-1} \setminus \{D\}$ where D is a clause in $\mathbb{D}_{t-1}$.

We can observe that the erasure rule is only important when we care about measuring space. If we are only interested in length, as will be the case in this and the next lecture, then we can give a slightly simplified description of what a resolution derivation is as follows.

Definition 2.1. We write $\pi : F \vdash A$, and say that π is a *resolution derivation* of A from F, if π is a sequence of clauses $\pi = \{D_1, \ldots, D_s\}$, where $D_s = A$ and each D_i is either a clause $C \in F$ (an *axiom*) or a *resolvent* derived from clauses $D_j, D_k \in \pi$ with $j, k < i$ by the *resolution rule.* Recall that the resolution rule

$$\frac{B \lor x \quad C \lor \overline{x}}{B \lor C} \tag{2.1}$$

says that we can resolve two clauses $B \lor x$ and $C \lor \overline{x}$ over the variable x to derive $B \lor C$.

A *resolution refutation* of the CNF formula F is a derivation $\pi : F \vdash \bot$, where $\bot$ denotes the empty clause containing no literals (which is the same thing as contradiction, since the empty clause has no literals that can satisfy it and therefore is false by definition).

$$
\begin{aligned}
F = \quad & (x \lor z) \land (\overline{z} \lor y) \land (x \lor \overline{y} \lor u) \land (\overline{y} \lor \overline{u}) \\
& \land\, (u \lor v) \land (\overline{x} \lor \overline{v}) \land (\overline{u} \lor w) \land (\overline{x} \lor \overline{u} \lor \overline{w})
\end{aligned}
$$

(a) CNF formula F.

1.	$x \lor z$	Axiom	9.	$x \lor y$	Res$(1, 2)$
2.	$\overline{z} \lor y$	Axiom	10.	$x \lor \overline{y}$	Res$(3, 4)$
3.	$x \lor \overline{y} \lor u$	Axiom	11.	$\overline{x} \lor u$	Res$(5, 6)$
4.	$\overline{y} \lor \overline{u}$	Axiom	12.	$\overline{x} \lor \overline{u}$	Res$(7, 8)$
5.	$u \lor v$	Axiom	13.	x	Res$(9, 10)$
6.	$\overline{x} \lor \overline{v}$	Axiom	14.	$\overline{x}$	Res$(11, 12)$
7.	$\overline{u} \lor w$	Axiom	15.	$\bot$	Res$(13, 14)$
8.	$\overline{x} \lor \overline{u} \lor \overline{w}$	Axiom			

(b) Resolution refutation of F.

Figure 1: Example resolution refutation.

An example of a resolution refutation is given in Figure 1.

Two important concepts for proof systems (in our case proof systems for refuting unsatisfiable CNF formulas) are *soundness* and *completeness*.

Definition 2.2. If F is a CNF formula and A is a clause, we say that a proof system is *sound* if $\pi : F \vdash A$ implies that $F \vDash A$.

In other words, if we can derive a clause A from a formula F, then it must also hold that F semantically implies A. We can derive no invalid conclusions in a sound system.

Definition 2.3. If F is a CNF formula and A is a clause, we say that a proof system is *implicationally complete* if $F \vDash A$ implies that there is a resolution derivation $\pi : F \vdash A'$ for some $A' \subseteq A$.

In other words, a proof system with the completeness property is constructed in such a way that all true implications can be proven.

Lemma 2.4. *Resolution is sound and implicationally complete.*

Proof sketch. We will not argue soundness in any detail, but this follows from the fact that the resolution rule described above is sound.

For completeness, let us give a proof by example. We begin by defining what we mean by a *decision tree*. For a CNF formula F, a decision tree T_F is a binary tree with leaves labelled by clauses in F, internal vertices labelled by variables x, and two edges from each internal vertex labelled 0 and 1. Suppose that we have a truth value assignment α to the variables in F. This assignment defines a path through T_F by starting from the root and following from each internal node x the edge label agreeing with the value assigned to x by α. Such a path ends in some leaf C, which is the answer of T_F on α.

The *search problem* for F, given α, is to find a clause C in F which is falsified by α. The tree T_F *solves* the search problem for F if on any α the answer C is a clause falsified by α. In Figure 2, we see a decision tree which solves the search problem for the formula:

$$F = (x \lor z) \land (\overline{z} \lor y) \land (x \lor \overline{y} \lor u) \land (\overline{y} \lor \overline{u}) \land (u \lor v) \land (\overline{x} \lor \overline{v}) \land (\overline{u} \lor w) \land (\overline{x} \lor \overline{u} \lor \overline{w}) \tag{2.2}$$

It should be clear that if F is unsatisfiable, then one can always construct a decision tree solving the search problem for F. It can be proven (which we leave as an exercise) that such a

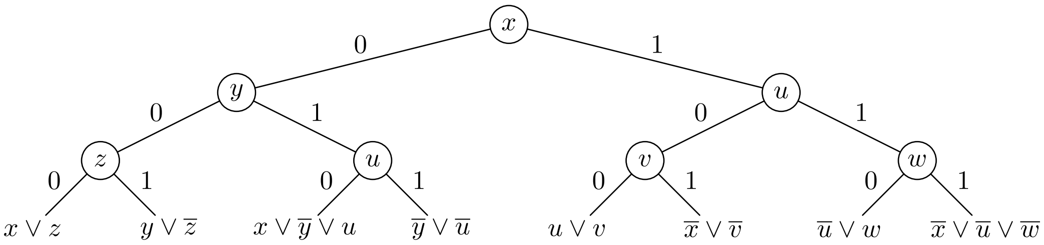
x
0
1
y
u
0
1
0
1
z
u
v
w
0
1
0
1
0
1
0
1
$x \lor z$
$y \lor \overline{z}$
$x \lor \overline{y} \lor u$
$\overline{y} \lor \overline{u}$
$u \lor v$
$\overline{x} \lor \overline{v}$
$\overline{u} \lor w$
$\overline{x} \lor \overline{u} \lor \overline{w}$

Figure 2: Decision tree for F.

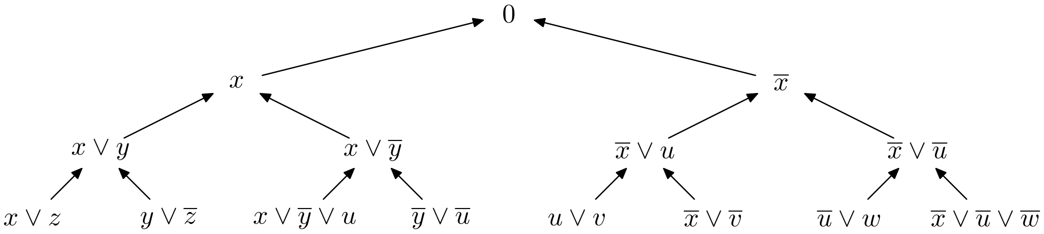
0
x
$\overline{x}$
$x \lor y$
$x \lor \overline{y}$
$\overline{x} \lor u$
$\overline{x} \lor \overline{u}$
$x \lor z$
$y \lor \overline{z}$
$x \lor \overline{y} \lor u$
$\overline{y} \lor \overline{u}$
$u \lor v$
$\overline{x} \lor \overline{v}$
$\overline{u} \lor w$
$\overline{x} \lor \overline{u} \lor \overline{w}$

Figure 3: Resolution refutation of F corresponding to the decision tree in Figure 2.

decision tree can be transformed into a resolution refutation of F by starting with the clauses in the leaves and resolving in a bottom-up fashion over the variables labelling the internal vertices. We show in Figure 3 how this transformation works for the decision tree in Figure 2. □

We want to make the concept of visualizing resolution derivations as graphs for formal. For this purpose, let us define *derivation graphs*.

Definition 2.5. The *derivation graph* G_π corresponding to a resolution derivation π is a *directed acyclic graph* (commonly abbreviated as DAG). The vertices of this graph are the clauses of the derivation. The edges of this graph go from $B \lor x$ and $B \lor \overline{x}$ to $B \lor C$ for each application of the resolution rule, and from B to $B \lor C$ if $B \lor C$ is derived from B by weakening.

Definition 2.6. A resolution derivation π is *tree-like* if G_π is a tree. (We may make copies of the axiom clauses to make G_π into a tree.)

For example, the derivation in Figure 3 is tree-like. Let us next recall what the measure *length* in resolution is.

Definition 2.7. The *length* $L(\pi)$ of a resolution derivation $\pi : F \vdash A$ is the number of clauses in π, counting repetitions. The *length of deriving* A from F in resolution is

$$L_{\mathcal{R}}(F \vdash A) = \min_{\pi : F \vdash A} \{L(\pi)\} \tag{2.3}$$

where the minimum is taken over all resolution derivations of A from F. The *length of deriving* A from F in *tree-like resolution* is

$$L_{\mathcal{T}}(F \vdash A) = \min_{\pi : F \vdash A} \{L(\pi)\} \tag{2.4}$$

where the minimum is taken over tree-like derivations π. The *length of refuting* F is the minimum length of deriving the empty clause $\perp$ from F, denoted $L_{\mathcal{R}}(F \vdash \perp)$ for general resolution and $L_{\mathcal{T}}(F \vdash \perp)$ for tree-like resolution.

As an example, the resolution refutation in Figure 3 has length 15.

When the flavour of resolution used is clear from context, we will drop the index. In particular, unless stated otherwise $L(F \vdash \bot)$ is understood to denote minimum refutation length in general resolution.

The overall goal of this and the next lecture is to prove the following exponential lower bound.

Theorem 2.8 ([Hak85]). *There is a family of unsatisfiable CNF formulas* $\{F_n\}_{n=1}^{\infty}$ *of size* $\Theta(n^3)$ *such that* $L_{\mathfrak{R}}(F_n \vdash \bot) = \exp(\Omega(n))$.

We mention that it has been shown that general resolution is exponentially stronger than tree-like resolution [BEGJ00, BIW04].

Although there are exponential lower bounds known for resolution, this proof system is widely used in practice anyway. This is due to the fact that is it is a nice system in which to construct proof search algorithms (though, recalling the definition from the first lecture, resolution is probably not *automatizable* as shown in [AR01]).

From a theoretical point of view, there are (at least) two reasons why we want to understand the resolution proof system. Firstly, it helps us to understand SAT solvers using resolution, and these SAT solvers are the best solvers known today. Secondly, by studying a simple proof system such as resolution, we can hope to develop insights and techniques which may then help us when we want to attack more powerful proof systems.

3 Weakening and Restrictions

For technical reasons, it turns out to be convenient to add an extra derivation rule to the resolution proof system. This rule, known as *weakening*, intuitively says that we are allowed to "throw away" information in a resolution refutation by going from stronger statements to weaker statements. Formally, the weakening rule is defined as follows.

Definition 3.1. The *weakening* rule

$$\frac{B}{B \vee C} \tag{3.1}$$

says that from a clause B we can derive the weaker clause $B \vee C$ for an arbitrary C. We say that $B \vee C$ is a *weakening* of B.

We can add this extra rule without loss of generality, since any applications of weakening in a resolution refutation can always be eliminated.

Proposition 3.2. *Any resolution refutation* $\pi : F \vdash \bot$ *using weakening can be transformed into a refutation* $\pi' : F \vdash \bot$ *without weakening in at most the same length.*

Proof. The proof follows easily by induction over the resolution refutation and is left as an exercise. (In fact, it is easy to show that the weakening rule does not affect any of the proof complexity measures we are interested in.) □

An important tool in proof complexity is that of *restrictions*, which are partial truth value assignments.

Definition 3.3. A *restriction* ρ is a partial truth value assignment. We represent a restriction by the set of literals $\rho = \{a_1, \ldots, a_m\}$ set to true by ρ.

Let us now explain how restrictions operate on clauses, formulas and derivations. In words, if a clause does not contain any variable set by a restriction, then it is not affected at all. Otherwise, if a clause is satisfied by a restriction, it can be ignored and is removed, and if it is not satisfied then we shrink it by removing falsified literals (which cannot help to satisfy the clause since they have been set to false).

$\pi =$

1.	$x \lor z$	Axiom in F
2.	$\overline{z} \lor y$	Axiom in F
3.	$x \lor \overline{y} \lor u$	Axiom in F
4.	$\overline{y} \lor \overline{u}$	Axiom in F
5.	$u \lor v$	Axiom in F
6.	$\overline{x} \lor \overline{v}$	Axiom in F
7.	$\overline{u} \lor w$	Axiom in F
8.	$\overline{x} \lor \overline{u} \lor \overline{w}$	Axiom in F
9.	$x \lor y$	$\mathsf{Res}(1, 2)$
10.	$x \lor \overline{y}$	$\mathsf{Res}(3, 4)$
11.	$\overline{x} \lor u$	$\mathsf{Res}(5, 6)$
12.	$\overline{x} \lor \overline{u}$	$\mathsf{Res}(7, 8)$
13.	x	$\mathsf{Res}(9, 10)$
14.	$\overline{x}$	$\mathsf{Res}(11, 12)$
15.	$\bot$	$\mathsf{Res}(13, 14)$

(a) Resolution refutation π.

$\pi{\restriction}_x =$

1.	1	
2.	$\overline{z} \lor y$	Axiom in $F{\restriction}_x$
3.	1	
4.	$\overline{y} \lor \overline{u}$	Axiom in $F{\restriction}_x$
5.	$u \lor v$	Axiom in $F{\restriction}_x$
6.	$\overline{v}$	Axiom in $F{\restriction}_x$
7.	$\overline{u} \lor w$	Axiom in $F{\restriction}_x$
8.	$\overline{u} \lor \overline{w}$	Axiom in $F{\restriction}_x$
9.	1	
10.	1	
11.	u	$\mathsf{Res}(5, 6)$
12.	$\overline{u}$	$\mathsf{Res}(7, 8)$
13.	1	
14.	$\bot$	$\mathsf{Res}(11, 12)$
15.	$\bot$	

(b) Restriction $\pi{\restriction}_x$ setting x to true.

Figure 4: Proof by example that restrictions preserve resolution refutations.

Definition 3.4. For a clause C, the *ρ-restriction* of C is

$$C{\restriction}_\rho = \begin{cases} 1 & \text{if } \rho \cap \mathit{Lit}(C) \neq \emptyset \\ C \setminus \{\overline{a} \mid a \in \rho\} & \text{otherwise} \end{cases} \tag{3.2}$$

where 1 denotes the trivially true clause. For a formula F, the *ρ-restriction* of F is $F{\restriction}_\rho = \bigwedge_{C \in F} C{\restriction}_\rho$. For a derivation $\pi = \{D_1, \ldots, D_s\}$, the *$\rho$-restriction* of π is $\pi{\restriction}_\rho = \{D_1{\restriction}_\rho, \ldots, D_s{\restriction}_\rho\}$.

A restriction of our example refutation π in Figure 1 can be seen in Figure 4. A closer look at Figure 4 reveals that $\pi{\restriction}_x$ is in fact a resolution refutation of $F{\restriction}_x$. This is true in general, as stated next.

Proposition 3.5. *If $\pi : F \vdash A$ is a resolution derivation and ρ is a restriction on $Vars(F)$, then $\pi{\restriction}_\rho$ is a derivation of $A{\restriction}_\rho$ from $F{\restriction}_\rho$, possibly using weakening.*

Proof. This is again an easy proof by induction over the resolution derivation, which we again leave as an exercise. □

Using Proposition 3.2, we can conclude that if $\pi : F \vdash \bot$, then $\pi{\restriction}_\rho$ can be transformed into a resolution refutation of $F{\restriction}_\rho$ *without weakening* in at most the same length as π.

4 The Concept of Width

We define the most important concept discussed in today's lecture, related to *length*, namely the concept of *width*. We define the concept of width for clauses, formulas and derivations.

Definition 4.1. The width of a clause C is written $W(C)$ and is defined as the number of literals in C. The width of a formula F is written $W(F)$ and is defined as $W(F) = \max_{C \in F}\{W(C)\}$, and the width of a derivation π is $W(F) = \max_{C \in \pi}\{W(C)\}$. The width of deriving A from F, denoted $W(F \vdash A)$, is defined to be $W(F \vdash A) = \min_{\pi : F \vdash A}\{W(\pi)\}$, where the minimum is taken over all resolution derivations of A from F. The width of refuting F is $W(F \vdash \bot)$.

As an example, the resolution from Figure 3 has width 3. We remark that for the width measure, there is no need to distinguish between general resolution width and tree-like resolution width. Any derivation in general resolution can be transformed into a tree-like derivation of the same width just by repeating all clauses enough times.

Clearly, the width of refuting a formula can never be larger than the total number of variables in it.

Observation 4.2. *For a formula F, we always have that $W(F \vdash \bot) \leq \big|Vars(F)\big|$.*

Also, it is easy to see that a *narrow* proof in general resolution is *short* by necessity.

Observation 4.3. *If $\pi : F \vdash \bot$ is a general resolution refutation in width $W(\pi) = w$, then $L(\pi) \leq \big(2 \cdot |Vars(F)|\big)^w$.*

Proof. For a proof in width w, $\big(2 \cdot |Vars(F)|\big)^w$ is an upper bound on the total number of possible distinct clauses. If π is a derivation in general resolution, there is no need to mention any clause more than once.[1] □

In an influential paper titled “Short proofs are narrow—resolution made simple”, Ben-Sasson and Wigderson proved that there is a kind of converse to Observation 4.3. As many research articles in theoretical computer science, this paper was first published in a conference version [BW99] and later appeared as a full-length journal version [BW01] (which is the reference we will use in what follows). We first describe the result of [BW01] very informally.

Theorem 4.4 (Very informal [BW01]). *If there is a short resolution refutation of F, then there is a resolution refutation in small width as well.*

We will spend the rest of today’s lecture formalizing and proving this theorem. We will also return to the title of the paper and discuss some questions that it raises.

5 Two Technical Lemmas

In order to make Theorem 4.4 more precise, we start by establishing two fairly simple but important technical lemmas.

Lemma 5.1. *If $W(F\!\restriction_x \vdash A) \leq w$ then $W(F \vdash A \vee \overline{x}) \leq \max\{w+1, W(F)\}$ (where the resolution derivations can make use of the weakening rule).*

Proof. Suppose $\pi = \{D_1, \ldots, D_s\}$ derives A from $F\!\restriction_x$ in width $W(\pi) \leq w$. We want to construct a derivation π' of $A \vee \overline{x}$ from F.

We start π' by simply listing all clauses in F. We then continue π' by listing all clauses in π' but with the literal $\overline{x}$ added to all clauses. We claim that this makes π' into a legal derivation of $A \vee \overline{x}$ from F (possibly with weakening). Clearly the last line in π' is $A \vee \overline{x}$, and the width is at most $\max\{w+1, W(F)\}$ so if we can just prove this claim that π' is a correct derivation, then we are done.

To prove the claim, we need to show that each clause in π' can be derived from previous clauses by resolution or weakening. The first half of π' just lists axioms from F. Thus, we only need to consider the second half, containing clauses of the form $D_i \vee \overline{x}$ for $D_i \in \pi$ with $\overline{x}$ added. Let $F_{\overline{x}} = \{C \in F \mid \overline{x} \in Lit(C)\}$ be the set of all clauses of F containing $\overline{x}$. We have three cases:

1. $D_i \in F_{\overline{x}}\!\restriction_x$: This means that $D_i \vee \overline{x} \in F$.

2. $D_i \in F\!\restriction_x \setminus F_{\overline{x}}\!\restriction_x$: This means that $D_i \in F$, so $D_i \vee \overline{x}$ can be derived by weakening D_i.

[1] But note that, as was pointed out in class, this argument does *not* work for tree-like resolution.

3. D_i is not an axiom in $F\restriction_x$: Thus, D_i is derived from $D_j, D_k \in \pi$ by resolution. We know then that we can derive $D_j \vee \overline{x}$ and $D_k \vee \overline{x} \in \pi'$ by weakening D_j and D_k. We can then resolve $D_j \vee \overline{x}$ and $D_k \vee \overline{x}$ to get $D_i \vee \overline{x}$.

Thus, we have shown that π' is a legal derivation of $A \vee \overline{x}$ from F, and the lemma follows. □

The second lemma will be our key lemma in what follows, and is a straightforward consequence of Lemma 5.1.

Lemma 5.2. *If $W(F\restriction_x \vdash \bot) \leq w - 1$ and $W(F\restriction_{\overline{x}} \vdash \bot) \leq w$ then it holds that $W(F \vdash \bot) \leq \max\{w, W(F)\}$.*

Proof. We begin by deriving $\overline{x}$ in width at most w, as we know we can do by Lemma 5.1. We then resolve $\overline{x}$ with all clauses $C \in F$ containing the literal x. This has the effect of removing all literals x, that is, the same effect as restricting F by $\overline{x}$. Thus, after this step we have derived all clauses in $F\restriction_{\overline{x}}$ in width at most $W(F)$. Finally, we derive $\bot$ from $F\restriction_{\overline{x}}$ in width at most w, using the derivation assumed to exist in the statement of the lemma. This gives the desired result. □

6 Length Versus Width in Resolution

For tree-like resolution, Theorem 4.4 can now be formalized as stated in the next theorem.

Theorem 6.1 ([BW01])**.** *For tree-like resolution, the width of refuting a CNF formula F is bounded from above by*

$$W(F \vdash \bot) \leq W(F) + \log_2 L_{\mathcal{T}}(F \vdash \bot).$$

From Theorem 6.1 we get the following immediate corollary, which can be used to obtain lower bounds on length from lower bounds on width.

Corollary 6.2 ([BW01])**.** *For tree-like resolution, the length of refuting a CNF formula F is bounded from below by*

$$L_{\mathcal{T}}(F \vdash \bot) \geq 2^{(W(F \vdash \bot) - W(F))}.$$

Proof of Theorem 6.1. We prove by induction that if $L_{\mathcal{T}}(F \vdash \bot) \leq 2^b$ then $W(F \vdash \bot) \leq W(F) + b$. The induction is over b and over the number of variables of F.

Base cases: If $b = 0$, there exists a proof of length 1. The only way this can happen is that F already contains the empty clause, which is a correct proof (if a rather uninteresting one) in width 0.

If the formula contains only one variable, say x, the formula must be $x \wedge \overline{x}$. By resolving these two clauses, we get the empty clause $\bot$. This proof has width 1.

Induction step: Suppose for a formula F with n variables that π is a tree-like refutation of F having length $\leq 2^b$. The last step of any given refutation $\pi : F \vdash \bot$ is $\frac{x \quad \overline{x}}{\bot}$ for some x. Let π_x and $\pi_{\overline{x}}$ be the tree-like subderivations of x and $\overline{x}$, respectively, as depicted in Figure 5. Since $L(\pi) = L(\pi_x) + L(\pi_{\overline{x}}) + 1 \leq 2^b$ (this is true since π is *tree-like*), one of π_x and $\pi_{\overline{x}}$ has length $\leq 2^{b-1}$. Suppose without loss of generality that $L(\pi_{\overline{x}}) \leq 2^{b-1}$.

1. $\pi_{\overline{x}}\restriction_x$ is a refutation of $F\restriction_x$ in length at most 2^{b-1}. Hence, by the induction hypothesis we have that $W(F\restriction_x \vdash \bot) \leq W\big(F\restriction_x\big) + b - 1 \leq W(F) + b - 1$.

2. $\pi_x\restriction_{\overline{x}}$ is a refutation in length at most 2^b of $F\restriction_{\overline{x}}$ where $F\restriction_{\overline{x}}$ has at most $n-1$ variables (since the restriction eliminates x). This means that we can again use the induction hypothesis to conclude that $W(F\restriction_{\overline{x}} \vdash \bot) \leq W\big(F\restriction_{\overline{x}}\big) + b \leq W(F) + b$.

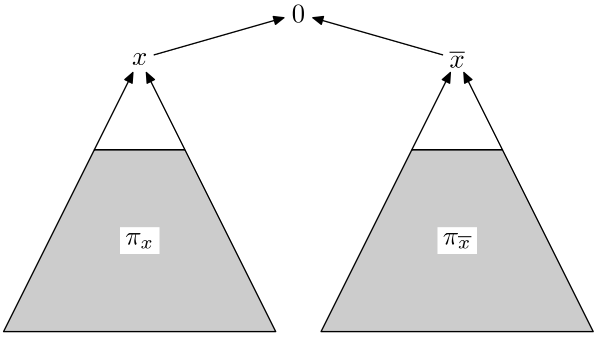

Figure 5: Depiction of the subderivations π_x and $\pi_{\overline{x}}$.

We have now shown that $W(F{\restriction_x} \vdash \bot) \leq W(F) + b - 1$ and $W(F{\restriction_{\overline{x}}} \vdash \bot) \leq W(F) + b$. But this means that we can apply Lemma 5.2 to derive that $W(F \vdash \bot) \leq W(F) + b$. The theorem now follows by the induction principle. □

In the light of the above proof of Theorem 6.1, is it true that "short proofs are narrow"? We start with a short proof which need not be narrow, and end up with a narrow proof. However, if we consider the details of the proof, we see that each inductive step may make the proof length several times larger. This because whenever we use Lemma 5.2, in step 2 in the construction of the proof of this lemma we have to repeat the whole derivation of $\overline{x}$ every time we need it in order to maintain tree-likeness. This can potentially blow up length exponentially, and such a blow-up cannot be avoided in a worst case scenario as was proven in [Ben02] (which later appeared as the journal version [Ben09]). Thus, if we want to nitpick it is in fact not true, at least not for tree-like resolution, that "short proofs are narrow." (But of course it is a great title for a paper anyway, and above all a great paper.)

Let us now focus on general, unrestricted resolution. In this case, we cannot get quite as strong a bound as for the tree-like case.

Theorem 6.3 ([BW01]). *The width of refuting a CNF formula F over n variables in general resolution is bounded from above by*

$$W(F \vdash \bot) \leq W(F) + \mathrm{O}\left(\sqrt{n \log L(F \vdash \bot)}\right) .$$

For tree-like resolution, we obtained an upper bound for width proportional to the logarithm of the length of a shortest proof. For general resolution, one way of viewing Theorem 6.3 is that since $\exp(\mathrm{O}(n))$ is the maximal possible proof length, we can write the bound above as

$$W(F \vdash \bot) \lessapprox W(F) + \sqrt{\log(\text{worst case proof length}) \cdot \log L(F \vdash \bot)} \ , \tag{6.1}$$

that is, as being proportional to the *geometric mean* of the logarithm of the length of a shortest proof and the logarithm of the worst case.

Just as in the tree-like case, we can rewrite Theorem 6.3 so that it yields lower bounds on length from lower bounds on width as follows.

Corollary 6.4 ([BW01]). *For general resolution, the length of refuting a CNF formula F over n variables is bounded from below by*

$$L(F \vdash \bot) \geq \exp\left(\Omega\left(\frac{(W(F \vdash \bot) - W(F))^2}{n}\right)\right) .$$

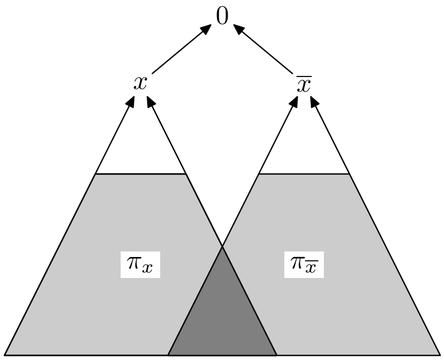

Figure 6: Depiction of the subderivations π_x and $\pi_{\overline{x}}$ for the general case.

As we noted in class, the bounds in Theorem 6.3 and Corollary 6.4 look perhaps a little bit ugly, and it is natural to ask whether one could improve them to something nicer and stronger, along the lines of the bounds that were proved for tree-like resolution above. This is a question that we will return to in a future lecture.

But be the bounds ugly or pretty, Corollary 6.4 has turned out to be terrifically useful in that it has been used to simplify many length lower bound proofs in resolution, as well as to prove number of new ones.

Let us discuss what is needed to prove superpolynomial lower bounds on length with the help of Corollary 6.4. Clearly, we will want $W(F)$ to be small, and in particular significantly smaller than $W(F \vdash \perp)$. Because if it is not, then the difference $W(F \vdash \perp) - W(F)$ will never be large and so we cannot get any interesting lower bounds. In the cases where we will apply Corollary 6.4, we will make sure that $W(F)$ is some constant. So how large will we then want the difference $W(F \vdash \perp) - W(F)$ to be? A moment of thought reveals that we will need $W(F \vdash \perp) - W(F) = \omega\bigl(\sqrt{n \log n}\bigr)$ in order to get superpolynomial lower bounds. Because if $W(F \vdash \perp) - W(F) = \mathrm{O}\bigl(\sqrt{n \log n}\bigr)$, we get the length bound $L(F \vdash \perp) \geq \exp\bigl(\mathrm{O}(\log n)\bigr)$ which is just a polynomial.

At the very end of the lecture, we briefly discussed what the proof of Theorem 6.3 (and hence Corollary 6.4) looks like. One can see that the type of proof used for tree-like resolution breaks down in the general case. It is not true that $L(\pi) = L(\pi_x) + L(\pi_{\overline{x}}) + 1$, since subderivations π_x and $\pi_{\overline{x}}$ may share clauses, as depicted in Figure 6. Therefore, making an inductive argument by restricting on the very last variable x resolved over in the proof is no longer necessarily a good idea.

The overall structure of the proof for the general case is the same, but now one looks at very wide clauses, and then constructs restrictions that eliminate many of these clauses by setting commonly occuring literals to true. This leads to a more complicated inductive argument. In this case, too, one gets an exponential blow-up in proof length, but for general resolution it is *not* known whether such a blow-up is necessary or not.

Open Problem 1. *Can the exponential length blow-up, which is unavoidable in tree-like resolution according to [Ben09], be avoided for general resolution? I.e., given short resolution refutation, can we find a refutation that is both narrow and short? (With at most polynomial blow-up, say.) Or is there a length-width trade-off, so that decreasing width must always increase length in worst case?*

A Proof of Theorem 6.3 (Not Covered in the Lecture)

Let us finally, with joint efforts by the scribe and the lecturer, add the missing proof of Theorem 6.3 that we did not have time to do in class.

Recall that Lemma 5.2 says that if $W(F{\restriction_x} \vdash \bot) \leq w - 1$ and $W(F{\restriction_{\overline{x}}} \vdash \bot) \leq w$, then $W(F \vdash \bot) \leq \max\{w, W(F)\}$. The high-level argument will again be to find some good variable to restrict, use the induction hypothesis to get small-width derivations of the formulas obtained by restricting this variable to true and false, and then apply Lemma 5.2 to stitch this together to a small-width refutation of the unrestricted formula.

Let F be a k-CNF formula over n variables. Suppose $W(F \vdash \bot) \leq k + \sqrt{8n \ln L}$, where ln is the natural logarithm. Fix a minimum length refutation $\pi : F \vdash \bot$ of length $L(\pi) = L$. Set

$$d = \sqrt{2n \ln L} \tag{A.1}$$

and

$$a = \left(1 - \frac{d}{2n}\right)^{-1} . \tag{A.2}$$

Since $L \leq 2^{n+1} \lll e^{2n}$ we have $d < 2n$ and hence $a > 1$.

Let us say that a clause D is *fat* if $W(D) \geq d$. We write $\mathit{fat}(\pi)$ to denote the number of fat clauses in a derivation π. Then Lemma 5.2 says that follows if we can prove the next lemma.

Lemma A.1. *Let G be any* k*-CNF over $m \leq n$ variables and suppose there exists a refutation π' of G such that $\mathit{fat}(\pi') < a^b$ for a as in* (A.2)*. Then*

$$W(G \vdash \bot) \leq k + d + b - 1 \tag{A.3}$$

Proof of Theorem 6.3 assuming Lemma A.1. To see that Lemma A.1 yields that

$$W(F \vdash \bot) \leq W(F) + \sqrt{8n \ln L} \ , \tag{A.4}$$

observe first that $k = W(F)$ and $d = \sqrt{2n \ln L}$. Thus, we only need to work on b.

Clearly, not all clauses in the resolution refutation can be fat, so we have $\mathit{fat}(\pi) < L \leq a^{\lceil \log_a L \rceil} \leq a^{\log_a L + 1}$. Thus, it is sufficient to set $b = \log_a L + 1 = 1 + \frac{\ln L}{\ln a}$ (where we use that $\log_a x = \frac{\ln x}{\ln a}$). Furthermore, we see that $\ln a = \ln\bigl((1 - \frac{d}{2n})^{-1}\bigr) = -\ln\bigl(1 - \frac{d}{2n}\bigr) \geq \frac{d}{2n}$, using the inequality $\ln(1 + x) \leq x$ which is valid for $x > -1$. By (A.1) we get $\frac{d}{2n} = \sqrt{\frac{\ln L}{2n}}$. Hence, $W(F \vdash \bot) \leq k + d + b - 1 \leq k + 2\sqrt{2n \ln L}$. □

It remains to prove Lemma A.1. Again, we do the proof by nested induction over b and the number of variables m in G (which is at most the number of variables n in the original formula F).

Base cases: We have the following base cases for our inductive proof:

1. $m = 1$ or $k = 1$ or $L \leq 3$: This means that G contains $x \wedge \overline{x}$. If so, $W(G \vdash \bot) = 1$ and we are done.

2. $m \leq k$ (for $m, k \geq 2$, $L \geq 4$): Since $L > 3$ we have $d > 1$. but the refutation width is at most the number of variables, i.e., at most $m \leq k \leq k + \underbrace{d + b - 1}_{\geq 0}$.

3. $b = 0$: If $b = 0$, this means that there are no fat clauses in the resolution refutation. Thus, the refutation width is at most $d \leq d + \underbrace{k + b - 1}_{\geq 0}$.

Induction step: Suppose the induction hypothesis (A.3) holds for:

- all k-CNF formulas over strictly less than m variables,

- all k-CNF formulas over m variables with proofs π' such that $\mathit{fat}(\pi') < a^{b-1}$.

Consider $\pi : G \vdash \bot$ with $\mathit{fat}(\pi) < a^b$. G has $2m$ distinct literals all in all. There are at least $d \cdot \mathit{fat}(\pi)$ literals in fat clauses. So, by counting there is some literal appearing in at least $\frac{d}{2m}\mathit{fat}(\pi) \geq \frac{d}{2n}\mathit{fat}(\pi)$ fat clauses.

Suppose without loss of generality that x is such a literal that appears in at least $\frac{d}{2n}\mathit{fat}(\pi)$ number of fat clauses. Consider $\pi\!\upharpoonright_x : G\!\upharpoonright_x \vdash \bot$. All the fat clauses containing the literal x are satisfied by the restriction and disappear, so $\pi\!\upharpoonright_x$ has less than $\left(1 - \frac{d}{2n}\right)a^b \leq a^{b-1}$ fat clauses. By the induction hypothesis, we deduce that

$$W(G\!\upharpoonright_x \vdash \bot) \leq k + d + b - 2 \ . \tag{A.5}$$

Now consider $\pi\!\upharpoonright_{\overline{x}} : G\!\upharpoonright_{\overline{x}} \vdash \bot$. The refutation $\pi\!\upharpoonright_{\overline{x}}$ has less than a^b fat clauses and $G\!\upharpoonright_{\overline{x}}$ has less than m variables. This means that we can again apply the induction hypothesis to derive

$$W(G\!\upharpoonright_{\overline{x}} \vdash \bot) \leq k + d + b - 1 \ . \tag{A.6}$$

But now we can use Lemma 5.2 on the inequalities (A.5) and (A.6). This gives us that $W(G \vdash \bot) \leq \max\{k, k + d + b - 1\} = k + d + b - 1$, and Lemma A.1 follows.

References

[AR01] Michael Alekhnovich and Alexander A. Razborov. Resolution is not automatizable unless W[P] is tractable. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS '01)*, pages 210–219, October 2001.

[BEGJ00] Maria Luisa Bonet, Juan Luis Esteban, Nicola Galesi, and Jan Johannsen. On the relative complexity of resolution refinements and cutting planes proof systems. *SIAM Journal on Computing*, 30(5):1462–1484, 2000. Preliminary version appeared in *FOCS '98*.

[Ben02] Eli Ben-Sasson. Size space tradeoffs for resolution. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC '02)*, pages 457–464, May 2002.

[Ben09] Eli Ben-Sasson. Size space tradeoffs for resolution. *SIAM Journal on Computing*, 38(6):2511–2525, May 2009. Preliminary version appeared in *STOC '02*.

[BIW04] Eli Ben-Sasson, Russell Impagliazzo, and Avi Wigderson. Near optimal separation of treelike and general resolution. *Combinatorica*, 24(4):585–603, September 2004.

[BW99] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC '99)*, pages 517–526, May 1999.

[BW01] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. *Journal of the ACM*, 48(2):149–169, March 2001. Preliminary version appeared in *STOC '99*.

[Hak85] Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39(2-3):297–308, August 1985.