DD3501 Current Research in Proof Complexity	Jan 18, 2011
Lecture 13	
Lecturer: Jakob Nordström	*Scribe: Léo Perrin*

1 Quick Recap of Last Lecture

We started the last lecture by asking three questions:

1. If a k-CNF formula F is easy with respect to length, is it then also easy with respect to clause space?

2. If F is refutable in length L and clause space s, can it be refuted in length $\mathrm{O}(L)$ and clause space $\mathrm{O}(s)$ simultaneously?

3. If F is refutable in length L and we do not care about clause space as long as it is not worse than the worst-case upper bound $\mathrm{O}(S(F))$ (i.e., linear in the formula size), can the formula be refuted in simultaneous length $\mathrm{O}(L)$ and clause space $\mathrm{O}(S(F))$?

We then revealed (finally, since these questions were raised for the first time already during the autumn term) the answer to all three questions is "no," and set about giving the formal proofs of the answers to the first two questions.

Our starting point for the proof was pebbling formulas, which have interesting trade-offs between length and *variable* space. However, these formulas have resolution refutations in simultaneous minimal length (linear) and *clause* space (constant), and so will not give us the results we are looking for.

We then discussed the idea of replacing variables x in a CNF formula F by Boolean functions $f(x_1, \ldots, x_d)$ and expanding this to a new CNF formula $F[f]$. We saw that if we refute $F[f]$ in the naive way in resolution by mimicking a resolution refutation of the original formula F, then there is a space blow-up in that the variable space of the original refutation of F will be a lower bound on the clause space of the simulating refutation of $F[f]$. We claimed that such a space blow-up in fact turns out to be unavoidable if we pick the right substitution function f.

In order to prove this formally, we wanted to show that we can "project" any refutation of $F[f]$ to a refutation of the original formula F. We gave the following definition of what it means to "project" a configuration $\mathbb{D}$ derived from $F[f]$ to a configuration $\mathbb{C}$ derived from F. Recall that for sets of clauses $\mathbb{C}$ and $\mathbb{D}$ we define $\mathbb{C} \vee \mathbb{D} = \{C \vee D \mid C \in \mathbb{C}, D \in \mathbb{D}\}$ and for a set of variables $V = \{x, y, z, \ldots\}$, we let

$$\mathit{Vars}^d(V) = \{x_1, x_2, \ldots, x_d, y_1, y_2, \ldots, y_d, z_1, z_2, \ldots, z_d, \ldots\} \tag{1.1}$$

denote the variables after substitution (which we assume are disjoint from the variables in V).

Definition 1.1 (Projection). Assume that $f : \{0,1\}^d \mapsto \{0,1\}$ is a fixed Boolean function and let $\mathcal{P}$ be any sequential proof system. Let $\mathbb{D}$ denote an arbitrary set of Boolean functions over $\mathit{Vars}^d(V)$ of the form specified by $\mathcal{P}$. Also, let $\mathbb{C}$ denote an arbitrary set of disjunctive clauses over V. Then the function proj_f mapping sets of Boolean functions $\mathbb{D}$ to clauses $\mathbb{C}$ is an *f-projection* if it is:

Complete: If $\mathbb{D} \vDash C[f]$, then there is a $C' \subseteq C$ such that $C' \in \mathit{proj}_f(\mathbb{D})$ (i.e., the clause C either is in $\mathit{proj}_f(\mathbb{D})$ or is derivable from $\mathit{proj}_f(\mathbb{D})$ by weakening).

Nontrivial: If $\mathbb{D} = \emptyset$, then $\mathit{proj}_f(\mathbb{D}) = \emptyset$.

Monotone: If $\mathbb{D}' \vDash \mathbb{D}$ and $C \in \mathit{proj}_f(\mathbb{D})$, then $C \supseteq C' \in \mathit{proj}_f(\mathbb{D}')$.

Incrementally sound: Let A be a clause over V and let L_A be the encoding of some clause in $A[f]$ as a Boolean function of the type prescribed by $\mathcal{P}$. Then if $C \in \mathit{proj}_f(\mathbb{D} \cup \{L_A\})$, it holds for all literals $a \in \mathit{Lit}(A) \setminus \mathit{Lit}(C)$ that $\overline{a} \vee C \supseteq C_a \in \mathit{proj}_f(\mathbb{D})$.

Note that Definition 1.1 is stated for projections from any sequential proof system $\mathcal{P}$ to resolution, not just for projections from resolution to resolution. The next lemma says that we can always extract resolution refutations from $\mathcal{P}$-refutations using projections as defined in Definition 1.1.

Lemma 1.2. *Let $\mathcal{P}$ be a sequential proof system and $f : \{0,1\}^d \mapsto \{0,1\}$ a Boolean function, and suppose that proj_f is an f-projection. Then for any CNF formula F it holds that if $\pi_f = \{\mathbb{D}_0, \mathbb{D}_1, \ldots, \mathbb{D}_\tau\}$ is a $\mathcal{P}$-refutation of the substitution formula $F[f]$, the sequence of sets of projected clauses $\{\mathit{proj}_f(\mathbb{D}_0), \mathit{proj}_f(\mathbb{D}_1), \ldots, \mathit{proj}_f(\mathbb{D}_\tau)\}$ forms the "backbone" of a resolution refutation π of F in the following sense:*

1. *$\mathit{proj}_f(\mathbb{D}_0) = \emptyset$.*
2. *$\bot \in \mathit{proj}_f(\mathbb{D}_\tau)$.*
3. *All transitions from $\mathit{proj}_f(\mathbb{D}_{t-1})$ to $\mathit{proj}_f(\mathbb{D}_t)$ for $t \in [\tau]$ can be accomplished in resolution in such a fashion that $\mathit{VarSp}(\pi) = \mathrm{O}\big(\max_{\mathbb{D} \in \pi_f}\{\mathit{VarSp}(\mathit{proj}_f(\mathbb{D}))\}\big)$.*
4. *The length of π is upper-bounded by π_f in the sense that the only time π does a download of $C \in F$ is when π_f downloads some axiom $L_C \in C[f]$ from $F[f]$.*

Intuitively speaking, Lemma 1.2 says that we can consider the projected clause configurations $\mathit{proj}_f(\mathbb{D}_i)$ as "snapshots" of a resolution refutation π of F. These snapshots capture the "interesting part" of the refutation π in that "nothing important" happens in between the snapshots, in particular with regards to variable space.

2 How to Use Projections to Prove Space Lower Bounds

We ended last time after having stated Lemma 1.2. One of the items on today's agenda is to prove this lemma. Before doing so, however, we want to discuss how it can be used.

Note that Lemma 1.2 as stated holds for any sequential proof system $\mathcal{P}$. At some later point we will have to restrict our attention to $\mathcal{P}$ being resolution for the proofs to work, but for as long as it is possible we will try to keep the discussion general.

In order for Lemma 1.2 to be really useful, there is one more component that we need. Namely, we would like the projection to somehow preserve space when it is projecting derivations in the proof system $\mathcal{P}$ to resolution derivations. These considerations lead us to the concept of *space-faithfulness* as defined next.

Definition 2.1 (Space-faithful projection). Consider a sequential proof system $\mathcal{P}$ with space measure $\mathit{Sp}_\mathcal{P}(\cdot)$. Let $f : \{0,1\}^d \mapsto \{0,1\}$ be a fixed Boolean function, and suppose that proj_f is an f-projection. Then we say that proj_f is *space-faithful of degree K with respect to $\mathcal{P}$* if there is a polynomial Q of degree at most K such that for any set of Boolean functions $\mathbb{D}$ over $\mathit{Vars}^d(V)$ on the form prescribed by $\mathcal{P}$, it holds that $Q(\mathit{Sp}_\mathcal{P}(\mathbb{D})) \geq \big|\mathit{Vars}(\mathit{proj}_f(\mathbb{D}))\big|$. We say that proj_f is *linearly space-faithful* if Q has degree 1, and that proj_f is *exactly space-faithful* if we can choose $Q(n) = n$.

The way to read Definition 2.1 is that the smaller the degree K is, the tighter the reduction between the proof system $\mathcal{P}$ and resolution will be with respect to space.

Before proving Lemma 1.2, let us see how it can be used to prove trade-offs provided that we can construct space-faithful projections. As stated in the introduction, proving such trade-offs is the very reason we set up all this mathematical machinery.

Theorem 2.2. *Let $\mathcal{P}$ be a sequential proof system with space measure $Sp_{\mathcal{P}}(\cdot)$. Suppose that $f : \{0,1\}^d \mapsto \{0,1\}$ is a Boolean function such that there exists an f-projection which is space-faithful of degree K with respect to $\mathcal{P}$. Then if F is any unsatisfiable CNF formula and π_f is any $\mathcal{P}$-refutation of the substitution formula $F[f]$, there is a resolution refutation π of F such that:*

- *The length of π is upper-bounded by π_f in the sense that π makes at most as many axiom downloads as π_f.*
- *The space of π is upper-bounded by π_f in the sense that $VarSp(\pi) = \mathrm{O}\big(Sp_{\mathcal{P}}(\pi_f)^K\big)$.*

In particular, if there is no resolution refutation of F in variable space $\mathrm{O}(s)$ that does $\mathrm{O}(L)$ axioms downloads, then there is no $\mathcal{P}$-refutation of $F[f]$ in simultaneous length $\mathrm{O}(L)$ and $\mathcal{P}$-space $\mathrm{O}\big(\sqrt[K]{s}\big)$.

Proof. First, note that the first bullet in the theorem is a direct consequence of Lemma 1.2 (namely part 4 of this lemma) . Thus, we only need to prove the second bullet here.

Let π_f be a $\mathcal{P}$-refutation of $F[f]$, and let π be the resolution refutation we obtain by applying the projection $proj_f$ on π_f as in Lemma 1.2. By part 3 of the lemma we have

$$VarSp(\pi) = \mathrm{O}\left(\max_{\mathbb{D}\in\pi_f}\{\, VarSp(proj_f(\mathbb{D}))\}\right) \quad . \tag{2.1}$$

Fix some $\mathcal{P}$-configuration $\mathbb{D}$ maximizing the right-hand side of (2.1). For this $\mathbb{D} \in \pi$ we have

$$VarSp(proj_f(\mathbb{D})) = \mathrm{O}\big(Sp_{\mathcal{P}}(\mathbb{D})^K\big) \tag{2.2}$$

according to Definition 2.1, and since obviously $Sp(\mathbb{D}) \leq \max_{\mathbb{D}\in\pi_f}\{Sp(\mathbb{D})\} = Sp\big(\pi_f\big)$ we get by combining (2.1) and (2.2) that

$$VarSp(\pi) \leq \mathrm{O}\left(Sp\big(\pi_f\big)^K\right) \tag{2.3}$$

and the theorem follows. □

We now turn to the proof of Lemma 1.2. We will give a fairly detailed outline of the proof, but for the complete details we refer to Appendix A.

Proof sketch for Lemma 1.2. Suppose that $\pi_f = \{\mathbb{D}_0, \mathbb{D}_1, \ldots, \mathbb{D}_\tau\}$ is a $\mathcal{P}$-refutation of $F[f]$. In this proof, for the sake of conciseness let us write $\mathbb{C}_t$ to denote $proj_f(\mathbb{D}_t)$. We prove the parts of the lemma one by one.

1. The fact that $\mathbb{C}_0 = \emptyset$ is a direct consequence of the nontriviality of the projection.
2. $\bot \in \mathbb{C}_\tau$ is also implied immediately by the definition of a projection, namely by the completeness condition.
3. So far, not too much interesting stuff happened in the proof, but this part is more interesting. We need to show that if we can derive $\mathbb{D}_t$ from $\mathbb{D}_{t-1}$ using our favorite proof system $\mathcal{P}$, then we can derive $\mathbb{C}_t$ from $\mathbb{C}_{t-1}$ in resolution. Note that we only need to worry about new clauses in $\mathbb{C}_t \setminus \mathbb{C}_{t-1}$ here—clauses in $\mathbb{C}_{t-1} \setminus \mathbb{C}_t$ that disappear at time t can just be erased and will not cause us any problems. We have three cases for the derivation step at time t:

Inference: Any $C \in \mathbb{C}_t \setminus \mathbb{C}_{t-1}$ can be derived from $\mathbb{C}_{t-1}$ by weakening as $proj_f$ is monotone.

Erasure: Again, monotonicity allows us to derive any new clauses in $\mathbb{C}_t$ from $\mathbb{C}_{t-1}$ by weakening.

Axiom download: Let C be a clause in $\mathbb{C}_t \setminus \mathbb{C}_{t-1}$ for $\mathbb{C}_t = proj_f(\mathbb{D}_{t-1} \cup \{L_A\})$, where L_A is an encoding in $\mathcal{P}$ of some clause in $A[f]$ downloaded at this time step. The incremental soundness of projections tells us that for all literals $a \in Lit(A) \setminus Lit(C)$ we have some $C_a \in \mathbb{C}_{t-1}$ such that $C_a \subseteq \overline{a} \vee C$. This means that we can derive $\overline{a} \vee C$ by weakening from $\mathbb{C}_{t-1}$ for all $a \in Lit(A) \setminus Lit(C)$. Then we download the axiom clause A in our resolution derivation and resolve with all these clauses one by one to derive C. □

This proof is only a sketch because some things were swept under the rug. For instance, we did not look at the variable space. While we know its values for the projected clause configurations $\mathbb{C}_t$, what happens in between these configurations? Who knows, maybe the space treacherously increases a lot after $\mathbb{C}_t$ before decreasing again to reach the known value at $\mathbb{C}_{t+1}$, pretending nothing happened? Actually, as shown in Appendix A, this cannot happen, but the proof sketch provided above does not argue why this is so.

3 Designing Projections for Resolution

What Theorem 2.2 says is the following: suppose we have a family of CNF formulas with lower bounds for refutation variable space in resolution, or with trade-offs between refutation length (or, more precisely, number of axiom downloads) and refutation variable space (such as for instance pebbling contradictions over suitable graphs, as we discussed last lecture). Then we can amplify these lower bounds or trade-offs to stronger proof complexity measures in a potentially stronger proof system $\mathcal{P}$, provided that we can find a Boolean function f and an f-projection $proj_f$ that is space-faithful with respect to $\mathcal{P}$.

Thus, at this point we can in principle forget everything about proof complexity. If we want to prove space lower bounds or time-space trade-offs for some proof system $\mathcal{P}$, we can focus on studying Boolean functions of the form used by $\mathcal{P}$ and trying to devise space-faithful projections for such functions. For the rest of this lecture, we will let $\mathcal{P}$ be resolution. What will be said below works in slightly greater generality, however, namely for the stronger family of k-DNF resolution proof systems. Although we did not have time to cover these results in class, we discuss them briefly in Appendix B.

To describe the projections we will study, we need a definition.

Definition 3.1 (Precise implication [BN11]). Let f be a Boolean function of arity d, let $\mathbb{D}$ be a set of Boolean functions over $Vars^d(V)$, and let C be a disjunctive clause over V. If

$$\mathbb{D} \models C[f] \tag{3.1a}$$

but for all strict subclauses $C' \subsetneq C$ it holds that

$$\mathbb{D} \not\models C'[f] \ , \tag{3.1b}$$

we say that the clause set $\mathbb{D}$ implies $C[f]$ *precisely* and write

$$\mathbb{D} \rhd C[f] \ . \tag{3.2}$$

Informally speaking, a precise implication of some Boolean function means that precisely this function is implied but nothing stronger. We want to project clauses that correspond to such precise implications as described next.

Definition 3.2 (Resolution projection [BN11]). Let f denote a Boolean function of arity d and let $\mathbb{D}$ be any set of Boolean functions over $Vars^d(V)$. Then we define the set of clauses

$$Rproj_f(\mathbb{D}) = \{C \mid \mathbb{D} \rhd C[f]\} \tag{3.3}$$

to be the *resolution projection* of $\mathbb{D}$.

What we want to prove now is that for $\mathcal{P}$ being resolution, and for the right choice of Boolean function f, $Rproj_f$ is a linearly space-faithful projection. Then we can apply Theorem 2.2 to the CNF formula families we discussed at the beginning of last lecture, and this will give us the results we are after. Let us start by showing that $Rproj_f$ is a projection.

Lemma 3.3. *The mapping $Rproj_f$ is an f-projection (for any sequential proof system $\mathcal{P}$).*

Proof. The proof of this lemma is very straightforward. All we have to do is to check each condition $Rproj_f$ must satisfy in order to be an f-projection.

Nontriviality is obvious. An empty $\mathcal{P}$-configuration evaluates to true under all assignments, and so cannot project any clauses according to Definition 3.2. Thus, $Rproj_f(\emptyset) = \emptyset$.

Suppose $\mathbb{D} \rhd C[f]$ and $\mathbb{D}' \vDash \mathbb{D}$. Then we can remove literals from C one by one until we get a new clause C' which is minimal with respect to the property that $\mathbb{D}' \vDash C'[f]$. By definition, we then have that $\mathbb{D}' \rhd C'[f]$, i.e., $C' \in Rproj_f(\mathbb{D}')$. This shows that $Rproj_f$ satisfies both *completeness* and *monotonicity*.

It remains to consider the *incremental soundness*.. Suppose, using the notation introduced above, that $\mathbb{D} \cup \{L_A\} \vDash C[f]$. Consider a truth value assignment α such that $\alpha(\mathbb{D}) = 1$ and $\alpha(C[f]) = 0$. If no such α exists, then simply by parsing what semantic implication $\vDash$ means we get that $\mathbb{D} \vDash C[f]$ and we have already seen how C can be derived from $Rproj_f(\mathbb{D})$ by weakening in this case. It would be great if this were always the case, because if so we would now be done with the proof, but as conscientious mathematicians we must also consider the other case.

Thus, suppose there exists such an α. This α must falsify $L_A \in A[f]$ or we immediately get a contradiction. If we write $A = \bigvee_{i=1}^{k} a_i$, we have that $A[f] = a_1[f] \vee a_2[f] \vee \cdots \vee a_k[f]$ and α must falsify every $a_i[f]$ in order to falsify L_A. Turning the tables, this means that $\alpha(\overline{a}_i[f]) = 1$ for all $i = 1, \ldots, k$. From this we can deduce that if α satisfies $\mathbb{D}$, then this assignment must also satisfy $\overline{a_i}[f] \vee C[f]$ for all $i = 1, \ldots, k$. The incremental soundness follows. □

In order to get a space-faithful projection $Rproj_f$, we pick a Boolean function f as in the next definition.

Definition 3.4 (Non-authoritarian function [BN11]). We say that a Boolean function $f(x_1, \ldots, x_d)$ is *k-non-authoritarian*[1] if no restriction to $\{x_1, \ldots, x_d\}$ of size k can fix the value of f. In other words, for every restriction ρ to $\{x_1, \ldots, x_d\}$ with $|\rho| \leq k$ there exists two assignments $\alpha_0, \alpha_1 \supset \rho$ such that $f(\alpha_0) = 0$ and $f(\alpha_1) = 1$. If this does not hold, f is *k-authoritarian*. A 1-(non-)authoritarian function is called just *(non-)authoritarian*.

Observe that a function on d variables can be k-non-authoritarian only if $k < d$. Perhaps the most natural example of a k-non-authoritarian function is exclusive or $\oplus_{k+1}$ of $k+1$ variables. Indeed, even if you fix any k variables in this function to any values, you can still flip the function to both true and false by choosing the right value of the final $(k+1)$st variable. One could say that in such a function, every variable has its say in deciding the final result. That is, as Boolean functions go, it seems to be somewhat democratic, or at least not authoritarian. Hence the terminology in Definition 3.4.

Now we are finally ready to state the main technical result, which will allow us to prove the space lower bounds and trade-offs that we are after: picking f non-authoritarian is enough to guarantee that $Rproj_f$ is linearly space-faithful for resolution.

[1] Such functions have also been referred to as $(k+1)$-*robust* in [ABRW02].

Theorem 3.5 ([BN11]). *If f is a non-authoritarian Boolean function, then the projection $Rproj_f$ is linearly space-faithful with respect to the resolution proof system.*

Proof. In the case of resolution, the set of Boolean functions $\mathbb{D}$ just consists of disjunctive clauses over $Vars^d(V)$. Fix an arbitrary such clause set $\mathbb{D}$ and let $V^* = Vars\big(Rproj_f(\mathbb{D})\big)$. What we want to prove is that $|\mathbb{D}| \geq |V^*|$. We assume without loss of generality that $\mathbb{D}$ is a minimal clause set such that $Vars\big(Rproj_f(\mathbb{D})\big) = V^*$, since otherwise we can just remove clauses from $\mathbb{D}$ until we get such a set.

Consider a bipartite graph with the vertices on the left labelled by clauses $D \in \mathbb{D}$ and the vertices on the right labelled by variables $x \in V^*$. We draw an edge between D and x if some variable x_i from $Vars^d(x)$ appears in D. Let $N(\mathbb{D}')$ denote the neighbours on the right of a clause set $\mathbb{D}'$. We claim without proof that $N(\mathbb{D}) = V^*$, i.e., that all $x \in V^*$ have incoming edges from $\mathbb{D}$ (establishing this claim is left as a useful but not too hard exercise in juggling with the definitions in this lecture).

Fix some $\mathbb{D}_1 \subseteq \mathbb{D}$ of maximal size with neighbour set $V_1^* = N(\mathbb{D}_1)$ such that $|\mathbb{D}_1| \geq |V_1^*|$ (and note that such a set always exist, since $\mathbb{D}_1 = \emptyset$ is allowed). If $\mathbb{D}_1 = \mathbb{D}$ we are done, since if so $|\mathbb{D}| \geq |V^*|$ which is exactly what we want to prove. Suppose therefore that $\mathbb{D}_1 \neq \mathbb{D}$ and let us argue by contradiction. Let $\mathbb{D}_2 = \mathbb{D} \setminus \mathbb{D}_1 \neq \emptyset$ and $V_2^* = V^* \setminus V_1^*$. For all $\mathbb{D}' \subseteq \mathbb{D}_2$ we must have $|\mathbb{D}'| \leq |N(\mathbb{D}') \setminus V_1^*| = |N(\mathbb{D}') \cap V_2^*|$, since otherwise we could have added $\mathbb{D}'$ to $\mathbb{D}_1$ and so this latter set would not have been chosen of maximal size. This in turns implies by Hall's theorem that there is a matching M from $\mathbb{D}_2$ into V_2^*.

Consider some clause $C \in Rproj_f(\mathbb{D}) \setminus Rproj_f(\mathbb{D}_1)$ such that $\mathbb{D}_1$ is "too weak" to project C (such a clause exists by the minimality of $\mathbb{D}$). Let C_i be the part of C that mentions variables from V_i^* for $i = 1, 2$. Then by Definitions 3.1 and 3.2 it holds that $\mathbb{D}_1 \cup \mathbb{D}_2 \vDash C_1[f] \vee C_2[f]$ but $\mathbb{D}_1 \nvDash C_1[f]$. This means that there is a truth value assignment α_1 to $Vars^d(V_1^*)$ satisfying $\mathbb{D}_1$ but falsifying $C_1[f]$. Observe that $Vars(\mathbb{D}_1) \subseteq Vars^d(V_1^*)$ by construction and that $Vars(\mathbb{D}_2) \cap Vars^d(V_1^*) = \emptyset$. Thus, we can choose α_1 to be a a partial truth value assignment that satisfies all clauses of $\mathbb{D}$ that contain variables that α_1 are assigning values to.

Using the matching M, we can find another partial truth value assignment α_2 to $Vars^d(V_2^*)$ that satisfies all clauses $D \in \mathbb{D}_2$ by setting at most one variable x_i for every $x \in V_2^*$. In particular, this means that α_2 does not determine the truth value of $C_2[f]$ since f is non-authoritarian, and this in turn means that we can extend α_2 to a full assignment over $Vars^d(V_2^*)$ such that $C_2[f]$ is falsified. But then $\alpha_1 \cup \alpha_2$ is an assignment[2] that satisfies $\mathbb{D}_1 \cup \mathbb{D}_2$ but falsifies $C_1[f] \vee C_2[f]$, which is a contradiction. □

4 Space Lower Bounds and Length-Space Trade-offs for Resolution

Now the three theorems we discussed at the beginning of lecture 12 follow with clause space instead of variable space in the bounds, just as we wanted. Let us conclude by stating these theorems (where for concreteness we choose the substitution function to be binary exclusive or, which after substitution gives us 6-CNF formulas).

Theorem 4.1 ([BN08]). *There is a family of explicitly constructible 6-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ which have resolution refutation length $L_{\mathcal{R}}(F_n \vdash \bot) = \mathrm{O}(n)$ but require clause space $Sp_{\mathcal{R}}(F_n \vdash \bot) = \Omega\big(\frac{n}{\log n}\big)$.*

That is, Theorem 4.1 tells us that although small space complexity implies that a formula is easy also with respect to length in resolution, the opposite does not hold: a formula can be maximally easy with respect to length but still have essentially worst-case clause space

[2] Note that $\alpha_1 \cup \alpha_2$ is a legal truth value assignment since α_1 and α_2 are assigning two disjoint sets of variables, namely $Vars^d(V_1^*)$ and $Vars^d(V_2^*)$. Therefore, they cannot "overlap" in such a way that there is some variable assigned to 1 by one assignment and to 0 by the other.

complexity. (In fact, although we did not have time to discuss this in class, by combining [ET01] and [HPV77] it is straightforward to show that if a formula is refutable in length $O(n)$, then it is also refutable in clause space $O(n/\log n)$, so the bound in Theorem 4.1 is optimal.)

The next two theorems give examples of trade-offs between proof length and proof space. These are just two particular instances of theorems that can be proven with this machinery—we refer to [BN11] for more examples.

Theorem 4.2 ([BN11, Nor12]). *Let $g : \mathbb{N}^+ \mapsto \mathbb{N}^+$ be any non-constant monotone function with $\omega(1) = g(n) = O\big(n^{1/7}\big)$ and fix any $\epsilon > 0$. Then there is a family of explicitly constructible 6-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ such that:*

- *The total space of refuting F_n is $TotSp_{\mathcal{R}}(F_n \vdash \bot) = O\big(g(n)\big)$ (i.e., very small space).*
- *There are resolution refutations $\pi_n : F_n \vdash \bot$ with length $L(\pi_n) = O(n)$ and total space $TotSp(\pi_n) = O\left(\left(n/(g(n))^2\right)^{1/3}\right)$ (i.e., very short refutations, but in substantially larger space).*
- *Any refutation $\pi_n : F_n \vdash \bot$ in clause space $Sp(\pi_n) = O\left(\left(n/(g(n))^2\right)^{1/3-\epsilon}\right)$ has superpolynomial length $L(\pi_n) = n^{\omega(1)}$ (i.e., decreasing the clause space significantly compared to the short refutations above leads to a superpolynomial blow-up in length).*

Theorem 4.3 ([BN11]). *Let κ be a sufficiently large constant. Then there is a family of explicitly constructible 6-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ and a constant $\kappa' \ll \kappa$ such that:*

- *The total space of refuting F_n is bounded by $TotSp_{\mathcal{R}}(F_n \vdash \bot) \leq \kappa' \frac{n}{\log n}$.*
- *There are resolution refutations $\pi_n : F_n \vdash \bot$ with length $L(\pi_n) = O(n)$ and linear total space $TotSp(\pi_n) = O(n)$.*
- *Any refutation $\pi_n : F_n \vdash \bot$ in clause space $Sp(\pi_n) \leq \kappa \frac{n}{\log n}$ must have exponential length $L(\pi_n) = \exp\left(n^{\epsilon}\right)$ for some $\epsilon > 0$.*

A very interesting question is whether results similar to the ones above could be proven for other strictly stronger proof systems such as cutting planes or polynomial calculus resolution, or even for polynomial calculus (which is formally speaking incomparable to resolution). For these systems no trade-offs are known (as far as the lecturer is aware) except for the very recent results in [HN12], and these are probably not trade-offs in a strict sense.

It can be shown that the projection in Definition 3.2 will not work for these proof systems (i.e., it is not space-faithful for CP, PC or PCR), and although we did not have time to cover this in class we discuss this briefly in Appendix B. But maybe there are other ideas that could work.

Open Problem 1. *Is it possible to prove space lower bounds and/or trade-offs between proof length/size and space for cutting planes, polynomial calculus, or PCR by designing smarter projections than in Definition 3.2 that are space-faithful for these proof systems?*

A A Full Proof of Lemma 1.2

In this appendix, we present a complete proof of Lemma 1.2 including the details that we glossed over in class.

Fix any sequential proof system $\mathcal{P}$, any f-projection $proj_f$ (for some Boolean function f), and any CNF formula F. Recall that we want to show that if $\pi_f = \{\mathbb{D}_0, \mathbb{D}_1, \ldots, \mathbb{D}_\tau\}$ is a $\mathcal{P}$-refutation of the substitution formula $F[f]$, then the sequence of projected clause sets

$\{proj_f(\mathbb{D}_0), proj_f(\mathbb{D}_1), \ldots, proj_f(\mathbb{D}_\tau)\}$ is essentially a resolution refutation π except for some details that we might have to fill in when going from $proj_f(\mathbb{D}_{t-1})$ to $proj_f(\mathbb{D}_t)$ in the derivation.

As we already mentioned above, parts 1 and 2 of Lemma 1.2 are immediate from Definition 1.1, since we have $proj_f(\mathbb{D}_0) = proj_f(\emptyset) = \emptyset$ by nontriviality and $\bot \in proj_f(\mathbb{D}_\tau)$ by completeness (note that $\mathbb{D}_\tau \models \bot = \bot[f]$ and the empty clause clearly cannot be derived by weakening).

We want to show that a resolution refutation of F can get from $proj_f(\mathbb{D}_{t-1})$ to $proj_f(\mathbb{D}_t)$ as claimed in part 3 of the lemma. For brevity, let us write $\mathbb{C}_i = proj_f(\mathbb{D}_i)$ for all i, and consider the possible derivation steps at time t.

Inference Suppose $\mathbb{D}_t = \mathbb{D}_{t-1} \cup \{L_t\}$ for some L_t inferred from $\mathbb{D}_{t-1}$. Since $\mathcal{P}$ is sound we have $\mathbb{D}_{t-1} \models \mathbb{D}_t$, and since the projection is monotone by definition we can conclude that all clauses in $\mathbb{C}_t \setminus \mathbb{C}_{t-1}$ are derivable from $\mathbb{C}_{t-1}$ by weakening. We go from $\mathbb{C}_{t-1}$ to $\mathbb{C}_t$ in three steps. First, we erase all clauses $C \in \mathbb{C}_{t-1}$ for which there are no clauses $C' \in \mathbb{C}_t$ such that $C \subseteq C'$. Then, we derive all clauses in $\mathbb{C}_t \setminus \mathbb{C}_{t-1}$ by weakening, noting that all clauses needed for weakening steps are still in the configuration. Finally, we erase the rest of $\mathbb{C}_t \setminus \mathbb{C}_{t-1}$. At all times during this transition from $\mathbb{C}_{t-1}$ to $\mathbb{C}_{t-1}$, the variable space of the intermediate clause configurations is upper-bounded by $\max\{\mathit{VarSp}(\mathbb{C}_{t-1}), \mathit{VarSp}(\mathbb{C}_t)\}$.

Erasure Suppose $\mathbb{D}_t = \mathbb{D}_{t-1} \setminus \{L_{t-1}\}$ for some $L_{t-1} \in \mathbb{D}_{t-1}$. Again we have that $\mathbb{D}_{t-1} \models \mathbb{D}_t$, and we can appeal to the monotonicity of the projection and proceed exactly as in the case of an inference above.

Axiom download So far, the only derivation rules used in the resolution refutation π that we are constructing are weakening and erasure, which clearly does not help π to make much progress towards proving a contradiction. Also, the only properties of the f-projection that we have used are completeness, nontriviality, and monotonicity. Note, however, that a "projection" that sends $\emptyset$ to $\emptyset$ and all other configurations to $\{\bot\}$ also satisfies these conditions. Hence, the axiom downloads are where we must expect the action to take place, and we can also expect that we will have to make crucial use of the incremental soundness of the projection.

Assume that $\mathbb{D}_t = \mathbb{D}_{t-1} \cup \{L_A\}$ for a function L_A encoding some clause from the substitution clause set $A[f]$ corresponding to an axiom clause $A \in F$. We want to show that all clauses in $\mathbb{C}_t \setminus \mathbb{C}_{t-1}$ can be derived in π by downloading A, resolving (and possibly weakening) clauses, and then perhaps erasing A, and that all this can be done without the variable space exceeding $\mathit{VarSp}(\mathbb{C}_{t-1} \cup \mathbb{C}_t) \leq \mathit{VarSp}(\mathbb{C}_{t-1}) + \mathit{VarSp}(\mathbb{C}_t)$.

We already know how to derive clauses by weakening, so consider a clause $C \in \mathbb{C}_t \setminus \mathbb{C}_{t-1}$ that cannot be derived by weakening from $\mathbb{C}_{t-1}$. By the incremental soundness of the projection, it holds for all literals $a \in \mathit{Lit}(A) \setminus \mathit{Lit}(C)$ that the clauses $\overline{a} \vee C$ can be derived from $\mathbb{C}_{t-1}$ by weakening. Once we have these clauses, we can resolve them one by one with A to derive C.

Some care is needed, though, to argue that we can stay within the variable space bound $\mathit{VarSp}(\mathbb{C}_{t-1}) + \mathit{VarSp}(\mathbb{C}_t)$ while performing these derivation steps. Observe that what was just said implies that for all $a \in \mathit{Lit}(A) \setminus \mathit{Lit}(C)$ there are clauses $\overline{a} \vee C_a \in \mathbb{C}_{t-1}$ with $C_a \subseteq C$. In particular, we have $\overline{a} \in \mathit{Lit}(\mathbb{C}_{t-1})$ for all $a \in \mathit{Lit}(A) \setminus \mathit{Lit}(C)$. This is so since by the incremental soundness there must exist some clause $C' \in \mathbb{C}_{t-1}$ such that $\overline{a} \vee C$ is derivable by weakening from C', and if $\overline{a} \notin \mathit{Lit}(C')$ we would have that C is derivable by weakening from C' as well, contrary to our assumption above.

If it happens that all clauses in $\mathbb{C}_t \setminus \mathbb{C}_{t-1}$ can be derived by weakening, we act as in the cases of inference and erasure above. Otherwise, to make the transition from $\mathbb{C}_{t-1}$ to $\mathbb{C}_t$ in a space-efficient fashion we proceed as follows.

1. Erase all clauses in $\mathbb{C}_{t-1} \setminus \mathbb{C}_t$ not used in any of the steps below.

2. Infer all clauses in $\mathbb{C}_t \setminus \mathbb{C}_{t-1}$ that can be derived by weakening from $\mathbb{C}_{t-1}$.

3. Erase all clauses in $\mathbb{C}_{t-1} \setminus \mathbb{C}_t$ used in these weakening moves but not used in any further steps below.

4. Download the axiom clause A, and derive any clauses $C \in \mathbb{C}_t \setminus \mathbb{C}_{t-1}$ such that $A \subseteq C$ by weakening.

5. For all remaining clauses $C \in \mathbb{C}_t \setminus \mathbb{C}_{t-1}$ that have not yet been derived, derive $\overline{a} \vee C$ for all literals $a \in \mathit{Lit}(A) \setminus \mathit{Lit}(C)$ and resolve these clauses with A to obtain C.

6. Erase all remaining clauses in the current configuration that are not present in $\mathbb{C}_t$, possibly including A.

Clearly, step 1 can only decrease the variable space, and steps 2 and 3 do not increase it. Step 4 can increase the space, but as was argued above we have $\mathit{Vars}(A) \subseteq \mathit{Vars}(\mathbb{C}_{t-1}) \cup \mathit{Vars}(C) \subseteq \mathit{Vars}(\mathbb{C}t-1) \cup \mathit{Vars}(\mathbb{C}_t)$ for every new clause C derived with the help of A. Step 5 does not change the variable space, and step 6 can only decrease it. It follows that the set of variables mentioned during these intermediate steps is contained in $\mathit{Vars}(\mathbb{C}_{t-1} \cup \mathbb{C}_t)$.

Wrapping up the proof, we have shown that no matter what $\mathcal{P}$-derivation step is made in the transition $\mathbb{D}_{t-1} \rightsquigarrow \mathbb{D}_t$, we can perform the corresponding transition $\mathbb{C}_{t-1} \rightsquigarrow \mathbb{C}_t$ for our projected clause sets in resolution without the variable space going above $\mathit{VarSp}(\mathbb{C}_{t-1}) + \mathit{VarSp}(\mathbb{C}_t)$. Also, the only time we need to download an axiom $A \in F$ in our projected refutation π of F is when π_f downloads some axiom from $A[f_d]$. The lemma follows.

B A Look at Stronger Proof Systems

It can be shown that the projection in Definition 3.2 also works for the k-DNF resolution proof systems, although we will not prove it in these notes. Unfortunately, there is quite a substantial loss in the parameters of the reduction here, as can be seen in the next theorem.

Theorem B.1 ([BN11]). *If f is a $(k+1)$-non-authoritarian Boolean function (for some fixed k), then the projection $\mathit{Rproj}_f(\mathbb{D})$ is space-faithful of degree $k+1$ with respect to k-DNF resolution.*

Proof sketch for Theorem B.1. Let us restrict our attention to 2-DNF resolution, since this already captures the hardness of the general case. Also, we sweep quite a few technical details under the rug to focus on the main idea of the proof.

Suppose that we have a set of 2-DNF formulas $\mathbb{D}$ of size $|\mathbb{D}| = m$ such that the set of projected variables $V^* = \mathit{Vars}(\mathit{Rproj}_f(\mathbb{D}))$ has size $|V^*| \geq K \cdot m^3$ for some suitably large constant K of our choice. We want to derive a contradiction.

As a first preprocessing step, let us prune all formulas $D \in \mathbb{D}$ one by one by shrinking any 2-term $a \wedge b$ in D to just a or just b, i.e., making D weaker, as long as this does not change the projection $\mathit{Rproj}_f(\mathbb{D})$. This pruning step does not decrease the size (i.e., the number of formulas) of $\mathbb{D}$.

By counting, there must exist some formula $D \in \mathbb{D}$ containing literals belonging to at least $K \cdot m^2$ different variables in V^*. Consider some clause $C \in \mathit{Rproj}_f(\mathbb{D})$ such that $\mathbb{D} \setminus \{D\}$ is too weak to project it. This means that there is an assignment α such that $\alpha(\mathbb{D} \setminus \{D\}) = 1$ but $\alpha\big(C[f]\big) \neq 1$, i.e., α either fixes $\alpha\big(C[f]\big)$ to false or leaves it undetermined. Let us pick such an α assigning values to the minimal amount of variables. It is clear that the domain size of α will then be at most $2(m-1)$ since the assignment needs to fix only one 2-term for every formula in $\mathbb{D} \setminus \{D\}$. But this means that the formula D contains a huge number of unset variables. We would like to argue that somewhere in D there is a 2-term that can be set to true without satisfying $C[f]$, which would lead to a contradiction.

We note first that if D contains $2m$ 2-terms $x_i^b \wedge y_j^c$ with all literals[3] in these terms belonging to pairwise disjoint variable sets for distinct terms (but where we can have $x = y$), we immediately get a contradiction. Namely, if this is the case we can find at least one 2-term $x_i^b \wedge y_j^c$ such that α does not assign values to any variables $x_{i'}$ or $y_{j'}$. We can satisfy this 2-term, and hence all of $\mathbb{D}$, without satisfying $C[f]$ since by assumption f is 3-non-authoritarian (so any assignments to x_i and y_j can be repaired by setting other variables $x_{i'}, y_{j'}$ to appropriate values).

But if D does *not* contain $2m$ such 2-terms over disjoint variables, then by counting (and adjusting our constant K) there must exist some literal a that occurs in D in at least $2m$ terms $a \wedge x_i^b$ with the x_i belonging to different variables. Moreover, these 2-terms were not pruned in our preprocessing step, so they must all be necessary. Because of this, one can argue that there must exist some other assignment α' such that $\alpha'(\mathbb{D} \setminus \{D\}) = 1$, $\alpha\big(C[f]\big) \neq 1$, and $\alpha'(a) = 1$. Now at least one of the $2m$ companion variables of a is untouched by α' and can be set to true without satisfying $C[f]$. This is the contradiction we needed to finish this proof. □

Although we will not go into any details here, we want to mention that Theorem B.1 can be used to obtain the following results:

1. The k-DNF resolution proof systems form a strict hiearchy with respect to space. That is, for every $k \geq 1$ there is a family of formulas which requires non-constant formula space (the generalization of clause space) in k-DNF resolution but can be refuted in constant formula space in $(k+1)$-DNF resolution.

2. There are length space trade-offs for k-DNF resolution qualitatively similar to those in Theorems 4.1, 4.2, and 4.3, albeit with slightly worse parameters.

Comparing Theorem 3.5 to Theorem B.1, we can see that for $k = 1$ (i.e., standard resolution) the latter theorem is off by one in the exponent. A natural question is whether the exponent in Theorem B.1 can be improved from $k+1$ to k, or whether perhaps the projection in Definition 3.2 is even linearly space-faithful for k-DNF resolution for any k. The answer is that the loss in the parameters in Theorem B.1 as compared to Theorem 3.5 is necessary, except perhaps for an additive constant 1 in the degree (which does not rule out, however, that stronger trade-offs for k-DNF resolution, matching those for resolution, could be proven by other means).

Theorem B.2 ([NR11]). *Let f denote the exclusive or of $k+1$ variables. Then the projection $Rproj_f$ cannot be space-faithful with respect to k-DNF resolution for any degree $K < k$.*

To see why we cannot hope to use $Rproj_f$ to prove lower bounds for cutting planes, polynomial calculus, or PCR, consider the following examples.

Example B.3. If we have variables $x[1], x[2], x[3], \ldots$ and make substitutions using binary exclusive or $\oplus_2$ to get new variables $x[1]_1, x[1]_2, x[2]_1, x[2]_2, x[3]_1, x[3]_2, \ldots$, then the example

$$\sum_{i=1}^{k} (x[i]_1 - x[i]_2) \geq k \tag{B.1}$$

shows that just a single CP-inequality can project an arbitrarily large conjunction $x[1] \wedge x[2] \wedge \cdots \wedge x[k]$. Thus, here we have $|\mathbb{D}| = 1$ while $VarSp(Rproj_{\oplus_2}(\mathbb{D}))$ goes to infinity.

Example B.4. Again using substitutions with $\oplus_2$, for polynomial calculus and PCR we have the example

$$-1 + \prod_{i=1}^{k} x[i]_1 x[i]_2 \tag{B.2}$$

[3] Although it is not important for this discussion, let us mention that when we use the notation x^b in the context of (k-DNF) resolution, this denotes the positive literal x if $b = 1$ and the negative literal $\overline{x}$ if $b = 0$.

showing that just two monomials can project the arbitrarily large conjunction $\overline{x[1]} \wedge \overline{x[2]} \wedge \cdots \wedge \overline{x[k]}$ if we use the projection in Definition 3.2.

As already mentioned in Open Problem 1, it would be very interesting to see whether other projections could be constructed that preserve space for CP, PC and/or PCR, or whether there are some fundamental obstacles here explaining why such an approach cannot work.

References

[ABRW02] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Space complexity in propositional calculus. *SIAM Journal on Computing*, 31(4):1184–1211, 2002. Preliminary version appeared in *STOC '00*.

[BN08] Eli Ben-Sasson and Jakob Nordström. Short proofs may be spacious: An optimal separation of space and length in resolution. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS '08)*, pages 709–718, October 2008.

[BN11] Eli Ben-Sasson and Jakob Nordström. Understanding space in proof complexity: Separations and trade-offs via substitutions. In *Proceedings of the 2nd Symposium on Innovations in Computer Science (ICS '11)*, pages 401–416, January 2011. Full-length version available at http://eccc.hpi-web.de/report/2010/125/.

[ET01] Juan Luis Esteban and Jacobo Torán. Space bounds for resolution. *Information and Computation*, 171(1):84–97, 2001. Preliminary versions of these results appeared in *STACS '99* and *CSL '99*.

[HN12] Trinh Huynh and Jakob Nordström. On the virtue of succinct proofs: Amplifying communication complexity hardness to time-space trade-offs in proof complexity. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC '12)*, May 2012. To appear.

[HPV77] John Hopcroft, Wolfgang Paul, and Leslie Valiant. On time versus space. *Journal of the ACM*, 24(2):332–337, April 1977.

[Nor12] Jakob Nordström. On the relative strength of pebbling and resolution. *ACM Transactions on Computational Logic*, 13(2), 2012. To appear. Preliminary version appeared in *CCC '10*.

[NR11] Jakob Nordström and Alexander Razborov. On minimal unsatisfiability and time-space trade-offs for k-DNF resolution. In *Proceedings of the 38th International Colloquium on Automata, Languages and Programming (ICALP '11)*, pages 642–653, July 2011.