Craig's Interpolation Theorem and Proof Complexity

Mahesh Viswanathan

Fall 2018

Craig's Interpolation theorem is a classical result in logic that holds for many different logics. The theorem has been used in different contexts in the broad area of formal methods and verification — in hardware and software specification; reasoning about large knowledge bases; type inference; combination of theorem provers for different first order theories; model checking of finite and infinite state systems through the construction of abstractions. In this lecture we look at some of its connections to theoretical computer science and complexity theory in particular. We will introduce the theorem for propositional logic, and its connections with proofs for propositional logic formulas.

1 Craig's Interpolation Theorem

Before we state and prove the interpolation theorem, it will be convenient to introduce some notation. A list of propositions $p_1, p_2, \ldots p_n$ will be denoted by $\overrightarrow{p}$ when the actual number of propositions in the list is unimportant. A formula φ over propositions $p_1, p_2, \ldots p_n, q_1, q_2, \ldots q_m$ will be sometimes denoted as $\varphi(\overrightarrow{p}, \overrightarrow{q})$, making explicit the propositions that *may* syntactically appear in φ. Finally, for a truth valuation v and a list of propositions $\overrightarrow{p}$, $\mathsf{v} \restriction_{\overrightarrow{p}}$ will denote the restriction of v to the propositions $\overrightarrow{p}$; note that v has an infinite domain, the domain of $\mathsf{v} \restriction_{\overrightarrow{p}}$ is finite and restricted to $\overrightarrow{p}$.

Theorem 1 (Craig). *Suppose $\models \varphi(\overrightarrow{p}, \overrightarrow{q}) \rightarrow \psi(\overrightarrow{q}, \overrightarrow{r})$. Then there is a formula $\eta(\overrightarrow{q})$ such that $\models \varphi(\overrightarrow{p}, \overrightarrow{q}) \rightarrow \eta(\overrightarrow{q})$ and $\models \eta(\overrightarrow{q}) \rightarrow \psi(\overrightarrow{q}, \overrightarrow{r})$. η is said to the* interpolant *of φ and ψ.*

Before presenting the proof of Theorem 1, let us examine its statement. There are different equivalent ways of presenting this result. Recall that $\models \varphi \rightarrow \psi$ iff $\varphi \models \psi$. Therefore, we could say that if ψ is a logical consequence of φ then η is a formula that captures the reason why, using only the common propositions. In this case, since $\varphi \models \eta$, we could think of η as an "abstraction" of φ (as η "forgets" the constraints φ imposes on *lstp*) that is sufficient to ensure ψ. Another formulation of Craig's theorem is as follows. Suppose $\varphi(\overrightarrow{p}, \overrightarrow{q}) \wedge \psi(\overrightarrow{q}, \overrightarrow{r})$ (or equivalently, $\neg(\varphi \rightarrow \psi)$) is unsatisfiable, then there is a formula $\eta(\overrightarrow{q})$ over the common propositions such that $\varphi \models \eta$ and $\eta \wedge \psi$ is unsatisfiable. Informally, η is an abstraction of φ that captures why $\varphi \wedge \psi$ is unsatisfiable. We will consider this formulation of Craig's theorem in terms of unsatisfiability, when we revist resolution later in this lecture.

Proof of Theorem 1. The proof of Craig's Interpolation Theorem is quite simple in this context of propositional logic. Let us define

$$M = \{\mathsf{v} \restriction_{\overrightarrow{q}} \mid \mathsf{v}[\![\varphi]\!] = 1\}.$$

Observe that since the domain (and range) of all functions in M is finite, M is a finite set. Let us, without loss of generality, take $M = \{\mathsf{v}_1, \mathsf{v}_2, \ldots \mathsf{v}_n\}$. The interpolant η will essentially say that "one among the assingments in M hold". Formally,

$$\eta(\overrightarrow{q}) = \vee_{i=1}^{n}(q_1^{(i)} \wedge q_2^{(i)} \wedge \cdots \wedge q_k^{(i)})$$

where

$$q_j^{(i)} = \begin{cases} q_j & \text{if } \mathsf{v}_i(q_j) = 1 \\ \neg q_j & \text{otherwise} \end{cases} \quad \text{and} \quad \overrightarrow{q} \text{ is } q_1, \ldots q_k$$

Clearly from the definition of M and η, we have $\varphi(\overrightarrow{p}, \overrightarrow{q}) \models \eta(\overrightarrow{q})$. Let us now argue that $\eta(\overrightarrow{q}) \models \psi(\overrightarrow{q}, \overrightarrow{r})$. Consider an arbitrary valuation v such that $\mathsf{v}[\![\psi]\!] = 0$. Since we have $\varphi \models \psi$, it must be that $\mathsf{v}[\![\varphi]\!] = 0$. Consider any assignment v' such that $\mathsf{v} \restriction_{\overrightarrow{q},\overrightarrow{r}} = \mathsf{v}' \restriction_{\overrightarrow{q},\overrightarrow{r}}$. Since v and v' agree on the propositions appearing in ψ, we have $\mathsf{v}'[\![\psi]\!] = 0$. Again, since ψ is a logical consequence of φ, it must be that $\mathsf{v}'[\![\varphi]\!] = 0$. Thus, no matter how the assignment to propositions $\overrightarrow{p}$ is changed from v, we will not be able to satisfy φ. This means that $\mathsf{v} \restriction_{\overrightarrow{q}} \notin M$. Thus, by our construction of η, $\mathsf{v}[\![\eta]\!] = 0$. This establishes that $\eta \models \psi$. □

Example 2. Let us look at a simple example that illustrates the construction of the interpolant in the proof of Theorem 1. Consider $\varphi = (p \wedge (q_1 \vee q_2)$ and let $\psi = (q_1 \vee q_2 \vee r)$. It is easy to see that $\models \varphi \rightarrow \psi$. The set M constructed in the proof in this case would be $M = \{\mathsf{v}_{11}, \mathsf{v}_{10}, assgn_{01}\}$, where v_{ij} is the function

$$\mathsf{v}_{ij}(q_1) = i \qquad \text{and} \qquad \mathsf{v}_{ij}(q_2)j$$

Given M, the proof constructs the follow formula as interpolant.

$$\eta = (q_1 \wedge q_2) \vee (q_1 \wedge \neg q_2) \vee (\neg q_1 \wedge q_2).$$

2 Size of Interpolants

The interpolant η constructed in the proof of Theorem 1 is exponential in the number of common variables, and therefore, could be exponential in the size of the formulas φ and ψ. Can this be imporved? Small interpolants can have a big impact in the contexts where interpolants are used, like in formal verification. Or can we prove that, in the worst case, the interpolant needs to be exponential in the size of the input formulas? Unfortunately, like many questions in theoretical computer science, this remains open and unresolved — we cannot prove or disprove that the construction in the proof of Theorem 1 is the best. However, in this section, we will show that it is unlikely that we can construct polynomial sized interpolants for all formulas. Or more precisely, we will show that resolving whether there exist polynomial sized interpolants for all formulas, is closely related to other open questions in complexity theory.

In order to present these results on the size of interpolants, we need to introduce new circuit complexity classes. Circuit complexity for a problem is based on a *non-uniform* model of computation. The idea is the following. Imagine you have the ability to choose a different algorithm for each input length; how much resource would you need? The "programs" for each input length are circuits, and different aspects of these circuits correspond to different computational resources one may care about. Running time in this context, roughly corresponds to circuit size. This leads us to analogs of P, NP, co-NP in the context of non-uniform complexity, which are defined next.

Definition 3 (Circuits)**.** A *Boolean Circuit* C is a sequence of assignments $A_1, A_2, \ldots A_n$, where each A_i is of one of the following forms.

$$\begin{array}{l} P_i = 0 \\ P_i = 1 \\ P_i = ? \\ P_i = P_j \wedge P_k,\ j,k < i \\ P_i = P_j \vee P_k,\ j,k < i \\ P_i = \neg P_j,\ j < i \end{array}$$

where each P_i is a variable that appears on the left-hand side in only A_i. The size of such a circuit, denoted $|C|$, is n.

The *input variables* are variables P_i such that the corresponding line A_i is $P_i = ?$; the input variables of C will be denoted by $I(C)$. Given an assignment $\mathsf{v} : I(C) \rightarrow \{0,1\}$, the value of C under v is the value assigned to the variable P_n in the last line. There is a natural order on the variables (based on which line they are assigned a value) which also imposes an order on the input variables. Thus, an assignment $I(C) \rightarrow \{0,1\}$ can be thought of a binary string x, where $x[i]$ is the value assigned to the ith input variable. In such a situation, the value of C under string x, will be denoted by $C(x)$.

Example 4. Circuits and formulas are two ways to represent Boolean functions. It is instructive to see how they differ by looking at an example.

Consider the boolean function parity$(x_1, x_2, \ldots x_n)$ which computes whether the number of propositions in $\{x_1, x_2, \ldots x_n\}$ set to 1 is odd or even. For example, we could write down the formula for some simple cases.

$$\begin{array}{l}\text{parity}(x_1,x_2) = (x_1 \vee x_2) \wedge \neg(x_1 \wedge x_2)\\ \text{parity}(x_1,x_2,x_3) = (((x_1 \vee x_2) \wedge \neg(x_1 \wedge x_2)) \vee x_3) \wedge \neg(((x_1 \vee x_2) \wedge \neg(x_1 \wedge x_2)) \wedge x_3)\end{array}$$

More generally, we can inductively write down the formula for parity$(x_1, x_2, \ldots x_n)$ by observing that the parity of $x_1, x_2, \ldots x_n$ is odd iff either (a) the parity of $x_1, \ldots x_{n-1}$ is even and x_n is 1, or (b) parity of $x_1, \ldots x_{n-1}$ is odd and x_n is 0. This results in the following definition.

$$\text{parity}(x_1, x_2, \ldots x_{n-1}, x_n) = (\text{parity}(x_1, x_2, \ldots x_{n-1}) \vee x_n) \wedge \neg(\text{parity}(x_1, x_2, \ldots x_{n-1}) \wedge x_n)$$

Observe that the formula we wrote down for parity(x_1, x_2, x_3) is based on exactly this definition. Observe that the size of the formula parity$(x_1, \ldots x_n)$ is double the size of parity$(x_1, \ldots x_{n-1})$. This means that the size of parity for n arguments is $O(2^n)$.

A circuit for the same formula does not grow as rapidly. This is because it can "reuse" previous computed answers. Let us look at the circuit corresponding to the formula parity(x_1, x_2, x_3) we wrote above.

$$\begin{array}{l} x_1 = ?\\ x_2 = ?\\ x_3 = ?\\ P_4 = x_1 \vee x_2\\ P_5 = x_1 \wedge x_2\\ P_6 = \neg P_5\\ P_7 = P_4 \wedge P_6\\ P_8 = P_7 \vee x_3\\ P_9 = P_7 \wedge x_3\\ P_{10} = \neg P_9\\ P_{11} = P_8 \wedge P_{10}\end{array}$$

Notice that variable P_7 stores parity(x_1, x_2) and it is reused without paying an extra cost. More generally, if C_{n-1} is the circuit computing parity$(x_1, \ldots x_{n-1})$ with last line $P_{k_{n-1}}$, the circuit C_n computing parity$(x_1, \ldots x_n)$ is given by

$$\begin{array}{l} C_{n-1}\\ x_n = ?\\ Q_1 = P_{k_{n-1}} \vee x_n\\ Q_2 = P_{k_{n-1}} \wedge x_n\\ Q_3 = \neg Q_2\\ P_{k_n} = Q_1 \wedge Q_3\end{array}$$

Now if $|C_n| = |C_{n-1}| + 5$. Therefore, in general, we get $|C_n| = O(n)$. Thus the circuit for the same syntactic formula can be exponentially smaller.

Definition 5. A language $A \subseteq \{0,1\}^*$ is said to be in *P/poly* iff there are constants $c, k \in \mathbb{N}$, and a (infinite) family of circuits $\{C_i\}_{i \in \mathbb{N}}$ such that (a) for every n, $|C_n| \leq cn^k$, and (b) for every x, $x \in A$ iff $C_{|x|}(x) = 1$.

A language $A \subseteq \{0,1\}^*$ is said to be in *NP/poly* (*co-NP/poly*) iff there are constants $c, k \in \mathbb{N}$, and a (infinite) family of circuits $\{C_i\}_{i \in \mathbb{N}}$ such that (a) for every n, $|C_n| \leq cn^k$, and (b) for every x, $x \in A$ iff for *some* (*all*) r, $C_{|x|}(x, r) = 1$.

One can think of P/poly (or NP/poly or co-NP/poly) as the collection of problems that be solved in polynomial time (or nondeterministic polynomial time or co-nondeterministic polynomial time) *given* a polynomially long *advice string*, namely, the description of the appropriate circuit. Just like in the case of (uniform/regular) complexity classes where it is open whether $\mathrm{P} \overset{?}{=} \mathrm{NP} \cap \text{co-NP}$, the same question is open in the non-uniform context as well. That is, a long standing open problem is whether $\mathrm{P/poly} \overset{?}{=} \mathrm{NP/poly} \cap \text{co-NP/poly}$.

Mundici's theorem says that establishing the existence of interpolants with polynomial circuit representation for all pairs of formulas, is equivalent to resolving the P/poly versus NP/poly ∩ co-NP/poly question. Thus, it is likely to be difficult to establish.

Theorem 6 (Mundici)**.** *If for any φ and ψ such that $\varphi \models \psi$ there is an interpolant whose circuit size is polynomial in φ and ψ then*

$$P/poly = NP/poly \cap co\text{-}NP/poly$$

Proof. Consider a problem $L \in \mathrm{NP/poly} \cap \text{co-NP/poly}$. Thus, there are families of circuits $\{A_n\}_{n\in\mathbb{N}}$ and $\{B_n\}_{n\in\mathbb{N}}$ such that for each n, A_n and B_n have size bounded by a polynomial function of n and for any binary string w, $w \in L$ iff $\exists p.\ A_{|w|}(p, w) = 1$ iff $\forall r.\ B_{|w|}(w, r) = 1$. These observations are just a consequence of $L \in \mathrm{NP/poly}$ and $L \in \text{co-NP/poly}$.

Based on the previous paragraph, we have, for any n, if for some assignment of values to $\overrightarrow{q}$, if $\exists \overrightarrow{p}.\ A_n(\overrightarrow{p}, \overrightarrow{q})$ holds then it also the case that $\forall \overrightarrow{r}.\ B_n(\overrightarrow{q}, \overrightarrow{r})$ holds. Therefore, we have $A_n(\overrightarrow{p}, \overrightarrow{q}) \models B_n(\overrightarrow{q}, \overrightarrow{r})$. By our assumption on interpolants, we have an interpolant (as a circuit) $C_n(\overrightarrow{q})$ such that $|C_n|$ is bounded by a polynomial in $|A_n|$ and $|B_n|$. Since $|A_n|$ and $|B_n|$ are bounded by polynomials in n, we have $|C_n|$ is bounded by a polynomial in n. Further, since C_n is an interpolant, we have if $\exists \overrightarrow{p}.\ A_n(\overrightarrow{p}, \overrightarrow{q})$ holds then $C_n(\overrightarrow{q})$ holds, and if $C_n(\overrightarrow{q})$ holds then $\forall \overrightarrow{r}.\ B_n(\overrightarrow{q}, \overrightarrow{r})$ holds. Thus, $\{C_n\}$ is a family of polynomially sized circuits deciding L, and thereby demonstrating that $L \in \mathrm{P/poly}$. □

3 Interpolants from Refutations

Constructing small interpolants for all formulas is likely to be difficult. However, it turns out that, if a formula $A(\overrightarrow{p}, \overrightarrow{q}) \to B(\overrightarrow{q}, \overrightarrow{r})$ has a short proof in some proof systems, then the proof can be used to construct an interpolant, whose size is propositional to the original proof. One such proof system that admits such a result is resolution.

Theorem 7. *Let the collection of clauses $\Gamma = \{A_i(\overrightarrow{p}, \overrightarrow{q})\}_{i=1}^{k} \cup \{B_j(\overrightarrow{q}, \overrightarrow{r})\}_{j=1}^{\ell}$ have a resolution refutation of length n. Then there is a circuit $C(\overrightarrow{q})$ such that*

$$\bigwedge_i A_i(\overrightarrow{p}, \overrightarrow{q}) \models C(\overrightarrow{q}) \text{ and } C(\overrightarrow{q}) \cap \bigwedge_j B_j(\overrightarrow{q}, \overrightarrow{r}) \text{ is unsatisfiable.}$$

Further $|C|$ is $O(n)$.

Proof. Since $\Gamma = \{A_i(\overrightarrow{p}, \overrightarrow{q})\}_i \cup \{B_j(\overrightarrow{q}, \overrightarrow{r})\}_j$ is unsatisfiable, every truth valuation v fails to satisfy at least one of the $A_i(\overrightarrow{p}, \overrightarrow{q})$ or $B_j(\overrightarrow{q}, \overrightarrow{r})$. We can also restate the properties of an interpolant C as

$$\begin{array}{l} \models \neg C(\overrightarrow{q}) \to \neg \bigwedge_i A_i(\overrightarrow{p}, \overrightarrow{q}) \text{ and} \\ \models C(\overrightarrow{q}) \to \neg \bigwedge_j B_j(\overrightarrow{q}, \overrightarrow{r}) \end{array}$$

Thus, $C(\overrightarrow{q})$ can be thought of as a way of labelling truth valuations: those labelled 0 will not satisfy some clause $A_i(\overrightarrow{p}, \overrightarrow{q})$ and those labelled 1 will not satisfy some $B_j(\overrightarrow{q}, \overrightarrow{r})$.

The structure of the proof will be as follows. Let $\psi_1, \psi_2, \ldots \psi_n$ be a resolution refutation of Γ. Let the set of common variables $\overrightarrow{q} = \{q_1, \ldots q_m\}$. Our circuit for the interpolant will be a sequence of the form $q_1 = ?, q_2 = ?, \ldots q_m = ?, P_1 = E_1, P_2 = E_2, \ldots P_n = E_n$ — the first n lines asserting that q_i's are variables, and then having one line $P_i = E_i$ corresponding to each line ψ_i of our refutation. It will be convenient to consider expressions E_i on the right hand side that have more than one logical connective. We will find it convenient to talk about the "circuit corresponding to a line ψ_i in the refutation". What we mean by this is look at the sequence of assignments upto (and including) line $P_i = E_i$; we will denote this as circuit C_i. The value of C_i (with respect to a truth assignment) will simply be the value variable P_i gets when we compute this circuit.

As mentioned above, we will construct the circuit line by line, corresponding to the refutation. With each line ψ_i in the refutation, we can associate a set of truth assignments, namely those that *do not* satisfy ψ_i, i.e., $M_i = \{\mathsf{v} \mid \mathsf{v}[\![\psi_i]\!]\} = 0$. The invariant we will maintain, as we build the circuit line by line, is that C_i "correctly labels" assignments belonging to M_i. That is, for $\mathsf{v} \in M_i$, if $C_i(\mathsf{v}) = 0$ then v does not satisfy some clause $A_i(\overrightarrow{p}, \overrightarrow{q})$ and if $C_i(\mathsf{v}) = 1$ then v does not satisfy some clause $B_j(\overrightarrow{q}, \overrightarrow{r})$. Notice that the last line $\psi_n =$, and so M_n is the set of all truth assignments. Thus, if the invariant is maintained, C_n will indeed be an interpolant because it will "correctly" label all assignments.

Let us now describe how we construct the circuit. For each line ψ_e, we will add a line $P_e = E_e$. What E_e is will depend on the justification for the line ψ_e in the refutation. Let us begin with the cases when ψ_e is a clause belonging to Γ. There are two cases to consider.

- Suppose $\psi_e \in \{A_i(lstp, \overrightarrow{q})\}_{i=1}^{k}$. Observe that any $\mathsf{v} \in M_e$ clearly does not satisfy ψ_e and could be safely labeled 0. Thus, the line corresponding to ψ_e will be $P_e = 0$.

- If $\psi_e \in \{B_j(\overrightarrow{q}, \overrightarrow{r})\}_{j=1}^{\ell}$, then each $\mathsf{v}inM_e$ could be labeled 1 because it does not satisfy $\psi_e \in \{B_j(\overrightarrow{q}, \overrightarrow{r})\}_{j=1}^{\ell}$. Thus, we have $P_e = 1$.

The next cases to consider is when ψ_e is a resolvent of two clauses. So let $\psi_e = \rho_1 \cup \rho_2$ and let it be the resolvent of clauses ψ_a and ψ_b in the refutation. We need to consider different cases based on what proposition the resolution is happening with respect to. Let us begin by considering the case when the proposition being resolved is one of the common variables. Without loss of generality, let us take $\psi_a = \rho_1 \cup \{q\}$ and $\psi_b = \rho_2 \cup \{\neg q\}$. Consider an arbitrary assignment $\mathsf{v} \in M_e$, i.e., $\mathsf{v}[\![\psi_e]\!] = 0$. From the soundness of the resolution proof rule, we know that either $\mathsf{v} \in M_a$ or $\mathsf{v} \in M_b$. If $\mathsf{v}(q) = 0$ then $\mathsf{v} \in M_a$; it may also, in addition, be the case that $\mathsf{v} \in M_b$, but that is unimportant. We could label such assignments in the same manner as the labeling corresponding to line ψ_a, i.e., as per the value of variable P_a. Similarly, if $\mathsf{v}(q) = 1$ then $\mathsf{v} \in M_b$ and so it can be labeled in the same manner as P_b. This gives us that the line corresponding to ψ_e in this case should be $P_e = (\neg q \wedge P_a) \vee (q \wedge P_b)$.

Let us now consider the case when ψ_e is a resolvent of ψ_a and ψ_b, but the resolution step is taken with respect to a proposition (say s) that is either in $\overrightarrow{p}$ or in $\overrightarrow{r}$. Once again any $\mathsf{v} \in M_e$ must also belong to $M_a \cup M_b$, but now since s is not in $\overrightarrow{q}$ it cannot be explicitly mentioned in the line $P_e = E_e$, as we did in the previous case. Again, without loss of generality, let us assume that $\psi_e = \rho_1 \cup \rho_2$, $\psi_a = \rho_1 \cup \{s\}$, $\psi_b = \rho_2 \cup \{\neg s\}$. Let v' be the assignment that is identical to v, except that it flips the assignment to s. Without loss of generality, we can assume that $\mathsf{v}(s) = 0$ and $\mathsf{v}'(s) = 1$, and for all other propositions t, $\mathsf{v}(t) = \mathsf{v}'(t)$. Observe that $\mathsf{v} \in M_a$ and $\mathsf{v}' \in M_b$. Further, $C_a(\mathsf{v}) = C_a(\mathsf{v}')$ and $C_b(\mathsf{v}) = C_b(\mathsf{v}')$; this is because C_a and C_b do not mention s. Let us consider two cases based on whether $s \in \overrightarrow{p}$ or $s \in \overrightarrow{r}$.

- Suppose $s \in \overrightarrow{p}$. Observe that in this case, for any j, $\mathsf{v}[\![B_j]\!] = \mathsf{v}'[\![B_j]\!]$, because s does not appear in B_j. Now, since C_a and C_b satisfy our invariant, we have, if $C_a(\mathsf{v}) = 1 (= C_a(\mathsf{v}'))$ then for some j, $\mathsf{v}[\![B_j]\!] = 0 (= \mathsf{v}'[\![B_j]\!])$. Similarly, if $C_b(\mathsf{v}') = 1 (= C_b(\mathsf{v}))$ then for some j, $\mathsf{v}'[\![B_j]\!] = 0 (= \mathsf{v}[\![B_j]\!])$. Thus, if either C_a or C_b label v, v' by 1, then C_e must do the same. Otherwise, both C_a and C_b label v and v' as 0, and this common label is correct as per out invariant. Therefore, we have $P_e = P_a \vee P_b$ in this case.

- Now let us consider $s \in \overrightarrow{q}$. In this case, for any i, $\mathsf{v}[\![A_i]\!] = \mathsf{v}'[\![A_i]\!] = 0$. Using an argument similar to the previous case, we can argue that if either C_a or C_b label v, v' by 0 then C_e must do the same. Otherwise, C_a and C_b agree on the label, and that is indeed the correct label. Therefore, dually, in this case we have, $P_e = P_a \wedge P_b$.

The proof that the invariant is maintained follows inductively, from the arguments we have made for each case. Thus, C_n is indeed the interpolant. It is worth noting that in C_n we have a sequence of initial assignments $q_1 = ?, q_2 = ?, \ldots q_m = ?$ that assert that $\overrightarrow{q}$ are input variables. This seems to suggest that the size of C_n also depends on $|\overrightarrow{q}|$ and hence on Γ. However, instead of asserting that all variables in $\overrightarrow{q}$ are input variables, we could assert only those variables in $\overrightarrow{q}$ that appear in the refutation. Thus, $|C_n|$ is indeed linear in the size of the refutation. $\square$

Example 8. Let us look at an example to see how an interpolant can be constructed from a refutation. Consider the set of clauses

$$\Gamma = \{\overbrace{\{p, q\}, \{\neg p, r\}}^{A}, \overbrace{\{\neg q, r\}, \{\neg r\}}^{B}\}.$$

Here p is a proposition that only appears in the A-clauses, and q and r are propositions that appear in both A and B clauses. Γ is unsatisfiable, and we are looking for an interpolant that only mentions the common variables q, r. Below we have the refutation (on the left) alongside the circuit for the interpolant (on the right) as per the proof of Theorem 7.

$$\begin{array}{ll}
 & q = ? \\
 & r = ? \\
\{\neg p, r\} & P_1 = 0 \\
\{\neg r\} & P_2 = 1 \\
\{\neg p\} & P_3 = (\neg r \wedge P_1) \vee (r \wedge P_2) \\
\{\neg q, r\} & P_4 = 1 \\
\{\neg q\} & P_5 = (\neg r \wedge P_4) \vee (r \wedge P_2) \\
\{p, q\} & P_6 = 0 \\
\{q\} & P_7 = P_3 \vee P_6 \\
\{\} & P_8 = (\neg q \wedge P_7) \vee (q \wedge P_5)
\end{array}$$

Observations like Theorem 7 have also been established for other proof systems of propositional logic. That is, in these proof systems, a short proof for a fact can be converted into a construction of a small interpolant. Theorem 7 can be strengthened for certain special sets of clauses — one can show that in certain special cases, not only is the interpolant small, but it is also *monotonic*. We conclude this section by presenting this result.

Definition 9 (Monotonic Circuits)**.** A *monotone circuit C* is one where there are no assignments of the form $P_i = \neg P_j$.

A monotonic circuit has the following monotonic property. Let us say $\mathsf{v}_1 \leq \mathsf{v}_2$ if for all proposition p, if $\mathsf{v}_1(p) = 1$ then $\mathsf{v}_2(p) = 1$, i.e., v_2 sets at least as many propositions to 1 as v. The value of a monotnoic circuit with respect to this ordering on assignments is monotonic. In other words, if C is monotonic, then for any $\mathsf{v}_1, \mathsf{v}_2$ such that $\mathsf{v}_1 \leq \mathsf{v}_2$, we have $C(\mathsf{v}_1) = 1$ implies $C(\mathsf{v}_2) = 1$.

Theorem 10. *Let the collection of clauses $\Gamma = \{A_i(\overrightarrow{p}, \overrightarrow{q})\}_{i=1}^{k} \cup \{B_j(\overrightarrow{q}, \overrightarrow{r})\}_{j=1}^{\ell}$ have a resolution refutation of length n. Further assume that either $\overrightarrow{q}$ occur only positively in A_is or $\overrightarrow{q}$ occur only negatively in B_js. Then there is a* monotone *circuit $C(\overrightarrow{q})$ such that*

$$\textstyle\bigwedge_i A_i(\overrightarrow{p}, \overrightarrow{q}) \models C(\overrightarrow{q}) \text{ and } C(\overrightarrow{q}) \cap \bigwedge_j B_j(\overrightarrow{q}, \overrightarrow{r}) \text{ is unsatisfiable.}$$

In addition, $|C|$ is $O(n)$.

Proof. The construction of the interpolant is identical to that in the proof of Theorem 7. All cases in that proof go through except for the case when we consider a line $\psi_e = \rho_1 \cup \rho_2$ in the refutation that is the resolvent of lines $\psi_a = \rho_1 \cup \{q\}$ and $\psi_b = \rho_2 \cup \{\neg q\}$ with respect to a proposition in $\overrightarrow{q}$, i.e., the common variables. In this case, in the proof of Theorem 7, the circuit was $P_e = (\neg q \wedge P_a) \vee (q \wedge P_b)$, which is not monotonic because if the use of $\neg q$. To prove our result, we need to change the circuit in this case. We will change it by forcing it to be monotonic in the most naïve way — we will remove the offending $\neg q$ and write $P_e = P_a \vee (q \wedge P_b)$.

The resulting construction is correct, but then the inductive argument using the invariant from Theorem 7 does not go through! Let us see what the problem is. Recall, that for any line ψ_e, we defined the set $M_e = \{\mathsf{v} \mid \mathsf{v}[\![\psi_e]\!] = 0\}$. The invariant we proved in Theorem 7 was for any valuation $\mathsf{v} \in M_e$, we have

- If $\mathsf{v}[\![C_e]\!] = 0$ then $\mathsf{v}[\![A_i]\!] = 0$ for some i, and
- If $\mathsf{v}[\![C_e]\!] = 1$ then $\mathsf{v}[\![B_j]\!] = 0$ for some j.

Let us try to prove this invariant by induction as in the proof of Theorem 7. Consider the case that we just changed, i.e., of a resolvent with respect to a common variable. So $\psi_e = \rho_1 \cup \rho_2$ is the resolvent of lines $\psi_a = \rho_1 \cup \{q\}$ and $\psi_b = \rho_2 \cup \{\neg q\}$. And we have, $P_e = P_a \vee (q \wedge P_b)$. Consider a valuation $\mathsf{v} \in M_e$. The problem with carrying out this inductive proof occurs when $\mathsf{v}(q) = 1$; the other case of $\mathsf{v}(q) = 0$ goes through rather simply. Then $\mathsf{v} \in M_b$. Now if $C_a(\mathsf{v}) = 0$ or $C_b(\mathsf{v}) = 1$ then $C_e(\mathsf{v}) = C_b(\mathsf{v})$ and correctness follows from the inductive assumptions on C_b. The problem occurs when $C_a(\mathsf{v}) = 1$ and $C_b(\mathsf{v}) = 0$.

The way to fix the problem is to prove a *stronger* invariant. In our old invariant, we proved that our circuit for line e was correct on the truth assignments in $M_e = \{\mathsf{v} \mid \mathsf{v}[\![\psi_e]\!] = 0\}$. Our strengthening will show that the circuit for line e is correct on a larger set of truth assignments. Depending on whether we consider the case when $\overrightarrow{q}$ appears only positively in $\{A_i(\overrightarrow{p}, \overrightarrow{q})\}_i$ or the case when $\overrightarrow{q}$ occurs only negatively in $\{B_j(\overrightarrow{q}, \overrightarrow{r})\}_j$, the invariant (and the proof) is slightly different. We will present the proof only for the case when $\overrightarrow{q}$ occurs negatively in $\{B_j(lstq, \overrightarrow{r})\}_j$. We will state the modified invariant for the other case, but leave the details to be filled out by the reader.

To describe the invariant in the case when $\overrightarrow{q}$ occurs negatively in $\{B_j(\overrightarrow{q}, \overrightarrow{r})\}_j$, we need to introduce some notation. For a clause ψ, $\psi \upharpoonright_{\overrightarrow{p}, \overrightarrow{q}}$ be the clause obtained by removing all literals involving propositions in $\overrightarrow{r}$. On the other hand, $\psi \upharpoonright_{-\overrightarrow{q}, \overrightarrow{r}}$ is the clause obtained from ψ by removing all literals of proposition $\overrightarrow{p}$ as well as all positive literals of $\overrightarrow{q}$. Our stronger invariant for circuit C_e will be for every valuation v

- if $\mathsf{v}[\![\psi \upharpoonright_{\overrightarrow{p}, \overrightarrow{q}}]\!] = 0$ and $C_e(\mathsf{v}) = 0$ then $\mathsf{v}[\![A_i]\!] = 0$ for some i, and
- if $\mathsf{v}[\![\psi \upharpoonright_{-\overrightarrow{q}, \overrightarrow{r}}]\!] = 0$ and $C_e(\mathsf{v}) = 1$ then $\mathsf{v}[\![B_j]\!] = 0$ for some j.

Notice that since $\{\} \upharpoonright_{\overrightarrow{p}, \overrightarrow{q}} = \{\} \upharpoonright_{-\overrightarrow{q}, \overrightarrow{r}} = \{\}$, proving this new invariant guarantees that C_n (where n is length of the resolution refutation) is an interpolant.

We now argue that the stronger invariant holds for the new construction. We consider each case in order.

$\psi_e \in \{A_i\}_i$: In this case, we have $\psi_e \upharpoonright_{\overrightarrow{p}, \overrightarrow{q}} = \psi_e$ and $P_e = 0$. The invariant, therefore, holds.

$\psi_e \in \{B_j\}_j$: Again we have $\psi_e \upharpoonright_{-\overrightarrow{q}, \overrightarrow{r}} = \psi_e$. Since $P_e = 1$, the invarint holds.

Reseolvent w.r.t $\overrightarrow{q}$: Let $\psi_e = \rho_1 \cup \rho_2$ be the resolvent of lines $\psi_a = \rho_1 \cup \{q\}$ and $\psi_b = \rho_2 \cup \{\neg q\}$. Recall we have $P_E = P_a \vee (q \wedge P_b)$.

1. Consider v such that $\mathsf{v}[\![\psi_e \upharpoonright_{\overrightarrow{p}, \overrightarrow{q}}]\!] = 0$ and $C_e(\mathsf{v}) = 0$. In this case, if $\mathsf{v}(q) = 1$ then $\mathsf{v}[\![\psi_b \upharpoonright_{\overrightarrow{p}, \overrightarrow{q}}]\!] = \mathsf{v}[\![\psi_b]\!] = 0$. Also, since $C_e(\mathsf{v}) = 0$, it must be that $C_b(\mathsf{v}) = 0$, and so correctness follows by induction. On the other hand, if $\mathsf{v}(q) = 0$ then $\mathsf{v}[\![\psi_a \upharpoonright_{\overrightarrow{p}, \overrightarrow{q}}]\!] = \mathsf{v}[\![\psi_a]\!] = 0$. Also, $C_a(\mathsf{v}) = 0$ and correctness follows by induction.
2. Consider v such that $v[\![\psi_e \upharpoonright_{-\overrightarrow{q}, \overrightarrow{r}}]\!] = 0$ and $C_e(\mathsf{v}) = 1$. If $C_a(\mathsf{v}) = 1$ then since $\mathsf{v}[\![\psi_a \upharpoonright_{-\overrightarrow{q}, \overrightarrow{r}}]\!] = \mathsf{v}[\![\rho_1 \upharpoonright_{-\overrightarrow{q}, \overrightarrow{r}}]\!] = 0$, the invariant follows by induction. Notice, how the stronger invariant helped the proof go through in this case which was problematic before. On the other hand, if $\mathsf{v}[\![q \wedge C_b]\!] = 1$ then $\mathsf{v}(q) = 1$. So $\mathsf{v}[\![\psi_b \upharpoonright_{-\overrightarrow{q}, \overrightarrow{r}}]\!] = 0$ and then invariant holds by induction.

Resolvent w.r.t. $\overrightarrow{p}$: Let $\psi_e = \rho_1 \cup \rho_2$ be the resolvent of lines $\psi_a = \rho_1 \cup \{p\}$ and $\psi_b = \rho_2 \cup \{\neg q\}$. Recall $P_e = P_a \vee P_b$.

1. Suppose $C_e(\mathsf{v}) = 0$ and $\mathsf{v}[\![\psi_e \upharpoonright_{\overrightarrow{p}, \overrightarrow{q}}]\!] = 0$. Then we know $C_a(\mathsf{v}) = C_b(\mathsf{v}) = 0$. Further either $\mathsf{v}[\![\psi_a]\!] = 0$ or $\mathsf{v}[\![\psi_b\}]\!] = 0$. Thus, correctness of construction by induction.
2. Suppose $\mathsf{v}[\![\psi_e \upharpoonright_{-\overrightarrow{q}, \overrightarrow{r}}]\!] = 0$ and $C_e(\mathsf{v}) = 1$. Now $\psi_e \upharpoonright_{-\overrightarrow{q}, \overrightarrow{r}} = \psi_a \upharpoonright_{-\overrightarrow{q}, \overrightarrow{r}} \cup \psi_b \upharpoonright_{-\overrightarrow{q}, \overrightarrow{r}}$, and so $\mathsf{v}[\![\psi_a \upharpoonright_{-\overrightarrow{q}, \overrightarrow{r}}]\!] = \mathsf{v}[\![\psi_b \upharpoonright_{-\overrightarrow{q}, \overrightarrow{r}}]\!] = 0$. Further since $C_e(\mathsf{v}) = 1$, either $C_a(\mathsf{v}) = 1$ or $C_b(\mathsf{v}) = 1$. So correctness follows by induction.

Resolvent w.r.t. $\overrightarrow{r}$: Proof similar to previous case.

The proof of correctness when $\overrightarrow{q}$ appears positively in $\{A_i(\overrightarrow{p}, \overrightarrow{q})\}_i$ is similar, though the invariant is slightly different. For a clause ψ, take $\psi \restriction_{\overrightarrow{p},+\overrightarrow{q}}$ to be the clause obtained by removing literals of $\overrightarrow{r}$ and negative literals of $\overrightarrow{q}$. In addition, $\psi \restriction_{\overrightarrow{q},\overrightarrow{r}}$ is the clause obtained by removing literals of $\overrightarrow{p}$. The invariant we will prove about the construction is, for every v,

- if $\mathsf{v}[\![\psi \restriction_{\overrightarrow{p},+\overrightarrow{q}}]\!] = 0$ and $C_e(\mathsf{v}) = 0$ then $\mathsf{v}[\![A_i]\!] = 0$ for some i, and
- if $\mathsf{v}[\![\psi \restriction_{\overrightarrow{q},\overrightarrow{r}}]\!] = 0$ and $C_e(\mathsf{v}) = 1$ then $\mathsf{v}[\![B_j]\!] = 0$ for some j.

The proof is similar and skipped. $\square$

Example 11. Let us consider the set of clauses Γ from Example 8.

$$\Gamma = \{\overbrace{\{p, q\}, \{\neg p, r\}}^{A}, \overbrace{\{\neg q, r\}, \{\neg r\}}^{B}\}.$$

Notice that the common propositions, q and r, appear only positively in A. The refutation alongside the interpolant construction is as follows.

$$\begin{array}{ll}
 & q = ? \\
 & r = ? \\
\{\neg p, r\} & P_1 = 0 \\
\{\neg r\} & P_2 = 1 \\
\{\neg p\} & P_3 = P_1 \vee (r \wedge P_2) \\
\{\neg q, r\} & P_4 = 1 \\
\{\neg q\} & P_5 = P_4 \vee (r \wedge P_2) \\
\{p, q\} & P_6 = 0 \\
\{q\} & P_7 = P_3 \vee P_6 \\
\{\} & P_8 = P_7 \vee (q \wedge P_5)
\end{array}$$

4 Lower bounds on Resolution Refutations

The results in Section 3 connecting resolution refutation lengths and size of interpolants, allows one to extract lower bounds on the length of resolution refutations for formulas. In particular we can show that there are sets of clauses Γ for which the shortest resolution refutations are exponential in the size of Γ. Thus, not every unsatisfiable formula has a short proof in resolution. The specific example we consider relate to cliques in graphs and their coloring. Let us recall these classical problems.

Recall that a graph $G = (V, E)$ is said to be k-colorable, if there exists a function $c : V \to \{1, 2, \ldots k\}$ such that if $(u, v) \in E$ then $c(u) \neq c(v)$. Observe that a graph $G = (V, E)$ can be represented by an assignment to propositions $\{q_{uv} \mid u, v \in \{1, 2, \ldots n\}\}$, with the interpretation that $(u, v) \in E$ iff q_{uv} is set to 1. We established the following proposition connecting colorability and satisfiability, that we recall here.

Proposition 12. *For any n, k, there is a set of $O(n^2k^2)$ clauses $color_{n,k}(\overrightarrow{q}, \overrightarrow{r})$ such that $\mathsf{v} \models color_{n,k}(\overrightarrow{q}, \overrightarrow{r})$ iff the graph represented by $\mathsf{v} \restriction_{\overrightarrow{q}}$ has a k-coloring.*

Proof. Let us recall the proof of this observations. The propositions r_{ui} in dictate that "vertex u has color i"; the propositions $\overrightarrow{r}$ are used to encode a color of the graph. Then $\text{color}_{n,k}$ is the following set of clauses.

- For each $u \in V$, the clause $r_{u1} \vee r_{u2} \vee \cdots \vee r_{uk}$. Intuitively these clauses capture the constraint that every vertex gets at least one of the k colors.
- For each $u \in V$ and $1 \leq i, j \leq k$ with $i \neq j$, the clause $\neg r_{ui} \vee \neg r_{uj}$. These clauses capture the constraint that a vertex does not get two different colors.

- For each edge $(u, v) \in E$ and color $1 \leq i \leq k$, the clause $\neg p_{ui} \vee \neg p_{vi}$. These clauses ensure that adjacent vertices do not get the same color.

The proof that these clauses satisfy the proposition is left as exercise. $\square$

A related property to coloring is that of *cliques* in a graph.

Definition 13. A k-clique in a graph $G = (V, E)$ is a subset $U \subseteq V$ such that $|U| = k$ and for every $u, v \in U$, with $u \neq v$, we have $(u, v) \in E$.

The problem of checking if a graph has a k-clique is known to be NP-complete.

Theorem 14. *The problem CLIQUE* $= \{\langle G, k\rangle \mid G$ *has a* k*-clique*$\}$ *is NP-complete.*

Like graph coloring, the CLIQUE problem is also closely connected with satisfiability of propositional logic.

Proposition 15. *For any* n, k*, there is a set of* $O(n^2k^2)$ *clauses* $clique_{n,k}(\overrightarrow{p}, \overrightarrow{q})$ *such that* $\mathsf{v} \models clique_{n,k}(\overrightarrow{p}, \overrightarrow{q})$ *iff the graph represented by* $\mathsf{v} \upharpoonright_{\overrightarrow{q}}$ *has a* k*-clique.*

Proof. The proof of this observation is similar to that of Proposition 12. We will introduce propositions that encode the k-clique, and clauses will specify constraints that characterize properties of a k-clique. Let proposition p_{iu}, for $i \in \{1, \ldots k\}$ and $u \in \{1, \ldots n\}$, denote that "the ith vertex in clique is u". Then $\text{clique}_{n,k}(\overrightarrow{p}, \overrightarrow{q})$ is the following set of clauses.

- For each $1 \leq i \leq k$, the clause $p_{i1} \vee p_{i2} \vee \cdots \vee p_{in}$. These clauses capture the constraint that the ith vertex of the clique must be among $\{1, \ldots n\}$.
- For each $1 \leq i \leq k$, and $1 \leq u, v \leq n$ such that $u \neq v$, the clause $\neg p_{iu} \vee \neg p_{iv}$. Intuitively, this says that the ith vertex of the clique can be at most one vertex.
- For each $1 \leq i, j \leq k$ and $1 \leq u, v \leq n$ with $u \neq v$, we have the clause $\neg p_{iu} \vee \neg p_{jv} \vee q_{uv}$. These clauses together say that if u, v are vertices in the clique then they have an edge between them.

Once again we leave the proof that these clauses satisfy the proposition to the reader. $\square$

Observe that if a graph G has a k-clique then it cannot be colored using $k-1$ colors. This is because each of the vertices in the clique must get different colors. Thus a graph with a k-clique needs at least k colors. This leads us to the following observation.

Proposition 16. *For any* n, k*,* $clique_{n,k}(\overrightarrow{p}, \overrightarrow{q}) \cup color_{n,k-1}(\overrightarrow{q}, \overrightarrow{r})$ *is unsatisfiable.*

Proof. A satisfying assignment for $\text{clique}_{n,k}(\overrightarrow{p}, \overrightarrow{q}) \cup \text{color}_{n,k-1}(\overrightarrow{q}, \overrightarrow{r})$ is a graph encoded by $\overrightarrow{q}$ that has k-clique (identified by $\overrightarrow{p}$) and can be colored using $k-1$ colors (with the coloring encoded by $\overrightarrow{r}$). This is clearly impossible. $\square$

Since $\text{clique}_{n,k}(\overrightarrow{p}, \overrightarrow{q}) \cup \text{color}_{n,k-1}(\overrightarrow{q}, \overrightarrow{r})$ is unsatisfiable, it must have a resolution refutation. How long is its refutation? Our goal will be to prove that this is exponential in n. Since the size of $\text{clique}_{n,k}(\overrightarrow{p}, \overrightarrow{q}) \cup \text{color}_{n,k-1}(\overrightarrow{q}, \overrightarrow{r})$ itself is polynomial in n, we have an example that has a "long" proof. In order to establish this result, we present a celebrated result in circuit complexity whose proof is beyond the scope of this course.

Theorem 17 (Razborov, Alon-Bopanna)**.** *Any monotone circuit that evaluates to* 1 *on* n*-vertex graphs containing a* k*-clique and evaluates to* 0 *on* n*-vertex graphs that are* $k-1$ *colorable must have size at least* $2^{\Omega(\sqrt{k})}$*, when* $k \leq n^{\frac{1}{4}}$*.*

Theorem 17 combined with Theorem 10 gives us the desired lower bound on proof lengths.

Theorem 18. *Any resolution refutation of* $clique_{n,k}(\overrightarrow{p}, \overrightarrow{q}) \cup color_{n,k-1}(\overrightarrow{q}, \overrightarrow{r})$ *must have length at least* $2^{\Omega(\sqrt{k})}$*, when* $k \leq n^{\frac{1}{4}}$*.*

Proof. Let there be a resolution refutation of length ℓ. Observe that $\overrightarrow{q}$ appears only positively in $\text{clique}_{n,k}(\overrightarrow{p}, \overrightarrow{q})$ and only negatively in $\text{color}_{n,k}(\overrightarrow{q}, \overrightarrow{r})$. Thus, $\text{clique}_{n,k}(\overrightarrow{p}, \overrightarrow{q}) \cup \text{color}_{n,k-1}(\overrightarrow{q}, \overrightarrow{r})$ satisfies the conditions of Theorem 10, and so there is a monotone interpolant of size $O(\ell)$. The interpolant is a monotone circuit satisfying conditions of Theorem 17. Thus, ℓ must be at least $2^{\Omega(\sqrt{k})}$. □

Theorem 18 establishes that resolution proofs can be long as a function of the size of the input clauses. Historically, the above theorem was not the first example or proof that some formulas may need long resolution proofs. The first such result was established by Haken. He showed that the *pigeon hole principle* has long resolution proofs. Recall that the pigeon hole principle says that if there are n holes and $n+1$ pigeons then some hole may contain more than one pigeon. Let pigeonhole_n be the propositional logic formula that says that every pigeon goes to a hole and no hole contains more than one pigeon. Then pigeonhole_n is unsatisfiable, and Haken showed that any resolution refutation of this fact must be exponential in n. The broad principle of using interpolation and lower bounds from monotonic circuit complexity have been used to establish that other proof systems can also have long proofs.

Understanding how long resolution proofs can be, and when they can be long, helps our theoretical understanding of the limits of sat solvers — what examples they may work well and when they can take long. But a probably more important reason for studying proof lengths in some proof system is because of its connections to some fundamental question in complexity theory. Essentially, the goal in proof complexity is to understand if there is a proof system for propositional logic that has the property that all facts have short proofs. Investigating whether this is true or not has important implications in complexity theory. Let us see why.

Definition 19. A proof system Π is *super* if every tautology φ has a proof in Π such that length of the proof is bounded by a polynomial functio on $|\varphi|$.

Now Theorem 18 says that resolution is not a super proof system. But are there other proof systems that are super? This is intimately tied to another open question in complexity theory.

Theorem 20 (Cook-Reckhow)**.** *Propositional logic has a super proof systems if and only if NP = co-NP.*

Proof. There are two directions to this proof. Assume that there is a super proof system. Then TAUT, the problem to determine if a given formula is a tautology, is in NP — the NP algorithm simply guesses the proof and checks that it is a proof in our super proof system. Since TAUT is co-NP-complete, it follows that co-NP = NP; the NP algorithm for an arbitrary problem $A \in$ co-NP is simple to reduce it to TAUT and then use the NP algorithm for TAUT.

On the other hand, suppose NP = co-NP then there is nondeterministic Turing machine N, running in polynomial time, that recognizes TAUT. Proofs for a formula φ in our new "super" proof system will simply be the nondeterministic choices that cause N to accept φ; notice, that these proofs will be polynomially long because N only has computations that are polynomially long. □

In the light of Theorem 20, to resolve the NP versus co-NP question, we need to prove that there are no super proof systems for proposition logic. Cook proposed an approach to tackling this problem. Consider concrete natural proof systems for propositional logic, one by one, and show that they are not super. Then use the intuition developed in this process to generalize and prove the absence of any super proof system. Resolution was the first proof system shown to be not super. Since then other proof systems have also been proved to be not super. However, there are still natural proof systems for which we have not been able to prove exponential lower bounds for proof lengths. One such proof system is the Frege proof system we introduced. It is still open whether there are tautologies for which proofs in the Frege proof system will be exponentially long.