COMS E6998: Proof Complexity and Applications (Spring'25) February 20, 2025

Lecture 4: Frege and Extended Frege

Instructor: *Toniann Pitassi* Scribes: *Adejuwon Fasanya, Jonah Stockwell*

1 Frege Systems

1.1 Definitions

We begin by defining some basic notation on logical consequences. Generally speaking, we use $\vdash$ to represent syntactical consequences and $\vDash$ to represent semantic consequences. In particular, we define the following notations:

$$\begin{aligned} \overset{\mathcal{P}}{\vdash} B &: \text{There is a } \mathcal{P}\text{-proof of B} \\ A_1, \dots, A_n \overset{\mathcal{P}}{\vdash} B &: \text{There is a } \mathcal{P}\text{-derivation of B from } A_1, \dots, A_n \\ \vDash B &: B \text{ is a tautology} \\ A_1, \dots, A_n \vDash B &: A_1 \wedge \dots \wedge A_n \text{ logically implies } B \end{aligned}$$

We omit the letter $\mathcal{P}$ above $\vdash$ if it is clear from the context.

Recall that a proof system $\mathcal{P}$ is sound and complete if $\overset{\mathcal{P}}{\vdash} A \iff \models A$ for all Boolean formulas A. We further define the implicational completeness of the system.

Definition 1. *A Frege system $\mathcal{P}$ is* implicationally complete *if $A_1, \dots, A_n \overset{\mathcal{P}}{\vdash} B$ whenever $A_1, \dots, A_n \vDash B$. That is, there is a $\mathcal{P}$-proof of B from $A_1, \dots, A_n$ whenever $A_1, \dots, A_n$ logically implies B.*

We now introduce the general notion of Frege systems, formally defined by Cook and Reckhow [CR79]. Frege systems are a family of proof systems sharing similar structures. Loosely speaking, a Frege proof consists of a sequence of Boolean formulas. Each system defines a *finite* set of axioms and (implicationally sound) inference rules which are used to deduce a new Boolean formula from existing ones. More formally, we have the following definition:

Definition 2 (Frege Systems)**.** *A* Frege system *over a complete Boolean basis is defined by a finite set of Frege rules. Each Frege rule has the form $\dfrac{C_1 \;\; C_2 \;\; \dots \;\; C_k}{D}$ where $C_1, \dots, C_k \models D$. In particular, an axiom is a rule with $k = 0$ (i.e., $\dfrac{}{D}$). Note that k, the maximum number of antecedent formulas in each rule, is a fixed constant that depends on the particular Frege system.*

Let $A_1, \dots, A_n, B$ be propositional formulas. A Frege derivation of B from a sequence of formulas $A_1, \dots, A_n$ is a finite sequence of formulas with the last formula being B, where each formula is either A_i

for some $i \in [n]$ or derived from previous formulas by a substitution instance of one of the Frege rules. [1] *A Frege proof of B is a derivation of B (that is, a derivation of B where $n = 0$).*

Any such proof system is trivially sound since each rule $\frac{C_1 \ C_2 \ \ldots \ C_k}{D}$ *satisfies* $C_1, \ldots, C_k \models D$ *by definition. A proof system is called a* Frege system *if it follows this syntax and is* implicationally complete*.*

Cook and Reckhow [CR79] showed that any two Frege systems are p-equivalent. Hence, to study the proof complexity of Frege systems, one may choose an arbitrary Frege system.

We now describe 2 typical types of syntax for Frege systems:

Hilbert style: where each line in the proof is a formula. These lines should be semantically viewed as tautologies when proving $\vdash B$.

Gentzen style: where each line is a *sequent*, which has the form $\Gamma \longrightarrow \Delta$, where Γ and Δ are sets of propositional formulas. Note that the symbol $\longrightarrow$ is *not* a logical connective in the language. It is part of the syntax of a Gentzen style proof system. [2] Semantically, a sequent $A_1, \ldots, A_n \longrightarrow B_1, \ldots, B_n$ is equivalant to the propositional formula: $(A_1 \wedge \cdots \vee A_n) \rightarrow (B_1 \vee \cdots \vee B_n)$. In other words, the conjunction of the A_i's implies the disjunction of the B_j's.

Note that technically speaking a Gentzen style proof system operates with sequents as the basic lines in a proof which doesn't fit our general definition where the basic lines are single propositional formulas. To remedy this situation, we can simply replace each sequent by its equivalent formula in order to view a Gentzen proof as a "Frege" proof.

1.2 Examples of Frege systems

We now give some examples of Frege systems. We begin by a typical Hilbert-style proof system, sometimes called Shoenfield's proof system, which generalizes Resolution.

Example 3. *The following axioms and deduction rules form an implicationally complete and sound Hilbert style system over the basis $\{\neg, \vee\}$:*

Axiom: $\frac{}{A \vee \neg A}$

Rules: $\frac{A}{A \vee B}$ $\quad \frac{A \vee A}{A}$ $\quad \frac{A \vee (B \vee C)}{(A \vee B) \vee C}$ $\quad \frac{A \vee B \quad \neg A \vee C}{B \vee C}$

Note that as a Hilbert-style system, each line of the proof is a propostional formula. One may see that if we restrict all lines in the proof to clauses (i.e., ORs of literals), the system degenerates to Resolution. So the completeness of this system follows from the completeness of Resolution. Since it is believed that Resolution does not p-simulate Frege, this illustrates a direct example of how restricting the lines affects the power of a proof system.

We now introduce one of the most widely-used Gentzen style proof systems, known as the sequent calculus system PK.

[1] Note that this is different from adding $A_1, \ldots, A_n$ as axiom schemas. In the definition of derivations, the hypotheses $A_1, \ldots, A_n$ can only be used without substitutions.

[2] In this note, we use the long right arrow $\longrightarrow$ to separate the antecedents and succedents of a sequent and the normal right arrow $\rightarrow$ as the logical connective.

Example 4 (PK sequent calculus (Gentzen style)). *In* PK*, the* Γ *and* Δ *of each sequent* $\Gamma \longrightarrow \Delta$ *are sets, so they can be reordering and contraction can be applied arbitrarily.*

The axioms and rules of PK*are the following (below we treat* Γ *and* Δ *as an arbitrary set of formulas):*

Axiom:
$$\frac{}{A \longrightarrow A}$$

Weakening rule:
$$\frac{\Gamma \longrightarrow \Delta}{A, \Gamma \longrightarrow \Delta} \text{ (left weakening)} \qquad \frac{\Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, A} \text{ (right weakening)}$$

Logical rules:
$$\frac{\Gamma \longrightarrow \Delta, A \quad \Gamma \longrightarrow \Delta, B}{\Gamma \longrightarrow \Delta, A \wedge B} \text{ ($\wedge$-right)} \qquad \frac{A, B, \Gamma \longrightarrow \Delta}{A \wedge B, \Gamma \longrightarrow \Delta} \text{ ($\wedge$-left)}$$
$$\frac{\Gamma \longrightarrow \Delta, A, B}{\Gamma \longrightarrow \Delta, A \vee B} \text{ ($\vee$-right)} \qquad \frac{A, \Gamma \longrightarrow \Delta \quad B, \Gamma \longrightarrow \Delta}{A \vee B, \Gamma \longrightarrow \Delta} \text{ ($\vee$-left)}$$
$$\frac{A, \Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, \neg A} \text{ ($\neg$-right)} \qquad \frac{\Gamma \longrightarrow \Delta, A}{\neg A, \Gamma \longrightarrow \Delta} \text{ ($\neg$-left)}$$

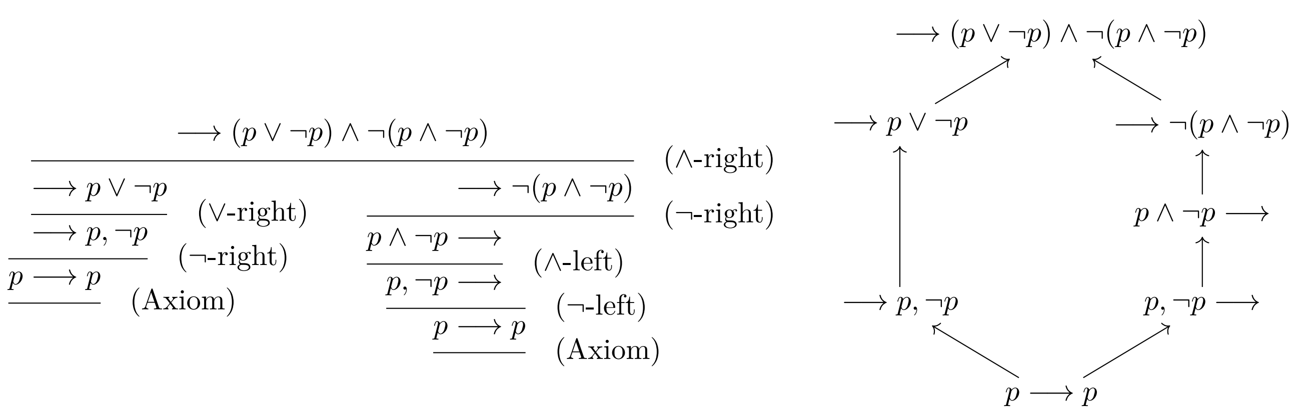
Cut rule:
$$\frac{\Gamma \longrightarrow \Delta, A \quad A, \Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta} \text{ (cut rule)}$$

It is worth noting that removing the cut rule from the system does not affect its soundness or completeness. This is usually called *cut elimination*. A proof is called *cut-free* if it does not make use of the cut rule. However, proofs for the same sequent can be significantly shorter when using the cut rule than in a cut-free proof.

It is easy to see that *cut-free* PK proofs have the *subformula property*: every formula appearing in the proof of a sequent $\Gamma \longrightarrow \Delta$ is a subformula of one of the formulas in $\Gamma \cup \Delta$. This is a nice property that can be useful when proving the completeness of PK.

Note that the logical rules are called, e.g., $\wedge$-right, because it introduces an $\wedge$ connective to the right of the sequent. So if a formula sequent $\Gamma \longrightarrow \Delta, A \wedge B$ appears in a cut-free proof, then unless the formula $A \wedge B$ is from axioms or weakening, the $\wedge$ connective must be introduced because of an application of the $\wedge$-right rule.

Figure 1 illustrates an example of proving the sequent $\longrightarrow (p \vee \neg p) \wedge \neg(p \wedge \neg p)$ in PK. By convention, the PK proof trees are usually written from bottom to top as shown in Figure 1a. Similarly to Resolution, we can also write the proof as a DAG as shown in Figure 1b.



(a) A proof tree in PK

(b) A corresponding proof DAG

Figure 1: Proof of $\longrightarrow (p \vee \neg p) \wedge \neg(p \wedge \neg p)$ in PK.

There are also some Frege systems that are neither Hilbert-style nor Gentzen-style. The Tait calculus is one example. It is similar in spirit to the sequent calculus PK, but works on sets of formulas instead of sequents. We can think of the Tait calculus as a version of the Sequent Calculus where every sequent in the proof has no formulas on the left-side of the sequent.

Example 5 (Tait calculus)**.** *Each line in Tait calculus is a set of propositional formulas* $\{B_1, \ldots, B_n\}$*. Informally speaking, one may interpret each line semantically as* $B_1 \vee B_2 \vee \cdots \vee B_n$*. The axioms and rules are the following (below we treat* Γ *as an arbitrary set of formulas):*

Axiom: $\dfrac{}{A, \neg A}$

Rules: $\dfrac{\Gamma}{\Gamma, A}$ (weakening)

$\dfrac{\Gamma, A \qquad \Gamma, B}{\Gamma, A \wedge B}$ ($\wedge$-intro) $\qquad \dfrac{\Gamma, A, B}{\Gamma, A \vee B}$ ($\vee$-intro)

$\dfrac{\Gamma, A \qquad \Gamma, \neg A}{\Gamma}$ (cut rule)

Similarly to PK, the system is also complete and sound even without the cut rule.

1.3 Soundness and Completeness of PK

In this section, we show that the sequent calculus PK is sound and complete. In fact, we will show that even cut-free PK is complete. This gives a proof of the cut elimination mentioned in Example 4. The implicational completeness of PK (with the cut rule) follows easily from the completeness.[3] Hence, PK is indeed a Frege system.

We start from the soundness of PK, which is the easier direction.

Theorem 6. PK *is* ***sound****.*

Proof. It is easily to check the implicational soundness of all cut-free rules and axioms: if the top sequents are valid, so is the bottom sequent. The soundness then follows by induction. □

We now sketch the proof that cut-free PK is complete.

Theorem 7. *Cut-free* PK *is* ***complete****.*

Proof Idea: Let f be a tautology. We will construct the proof backwards, starting with the sequent $\longrightarrow f$ We start with the sequent $\longrightarrow f$ at the root, and then working backwards, we will show that we will eventually construct a proof tree with the leaves labelled with sequents that are instances of the axiom. Starting at the root sequent $\longrightarrow f$ we will repeatedly apply a logical rule in *reverse* to an outermost connective of a current sequent. We repeatedly apply this process to each leaf sequent of the current partial proof tree derived so far, until eventually all leaf sequents are *atomic sequents*; that is, sequents of the form $X_{i_1}, \ldots, X_{i_c} \longrightarrow X_{j_1}, , \ldots, X_{j_d}$ where $X_{i_k}, ...X_{j_l}$ are all variables. We prove in the claim below, that if f is a tautology, then each atomic sequent eventually derived in the proof tree has the property that there exists some variable that occurs on both the right and the left side of the atomic sequent; that is, each atomic sequent is a weakening of an axiom instance. Assuming for now that this claim is true,

[3]Note that cut-free PK is not implicationally complete. For example, $\longrightarrow p \to q, \longrightarrow q \to r \models \longrightarrow p \to r$, but it is easy to see from the subformula property that this is not provable by cut-free PK.

the final last step of the construction is to apply weakening (in reverse) to each atomic sequent in order to derive an axiom instance at each leaf of the proof tree.

For example, below is a proof when $f = \neg(\neg P \wedge \neg Q) \vee \neg(P \vee Q)$, where P, Q are variables. As described above, we first label the root (written at the top of the proof tree) with the sequent $\rightarrow f$ that we want to derive. Then we grow the proof tree (from the root down to the leaves) by repeatedly applying a logical rule in reverse. In the example below, the leaves are all labelled with *atomic sequents*. Furthermore, all leaves satisfy the property in the claim (they are weakenings of an axiom instance); therefore we can complete the final last step (not shown) which is to derive an axiom instance at each leaf by applying the appropriate weakening rule in reverse.

$$\longrightarrow \neg(\neg P \wedge \neg Q) \vee \neg(P \vee Q)$$
$$\text{(}\vee\text{-right)}$$
$$\longrightarrow \neg(\neg P \wedge \neg Q), \neg(P \vee Q)$$
$$\text{(}\neg\text{-right)}$$
$$\neg P \wedge \neg Q \longrightarrow \neg(P \vee Q)$$
$$\text{(}\neg\text{-right)}$$
$$P \vee Q, \neg P \wedge \neg Q \longrightarrow$$
$$\text{(}\wedge\text{-left)}$$
$$P \vee Q, \neg P, \neg Q \longrightarrow$$
$$\text{(}\vee\text{-left)}$$
$$P, \neg P, \neg Q \longrightarrow \qquad\qquad Q, \neg P, \neg Q \longrightarrow$$
$$\text{(}\neg\text{-left)} \qquad\qquad \text{(}\neg\text{-left)}$$
$$P, \neg Q \longrightarrow P \qquad\qquad Q, \neg P \longrightarrow Q$$
$$\text{(Axiom)} \qquad\qquad \text{(Axiom)}$$

Proof of Theorem 7. We want to show that the procedure described above always succeeds in producing a legal Sequent Calculus proof of f, whenever f is a tautology. Note that in each step of the construction of the proof tree where we apply a logical rule in reverse, the number of sequents grows, but on the other hand, the number of *connectives* strictly decreases by at least one. Therefore, eventually we will obtain atomic sequents at the leaves of the partial proof tree. To complete the proof of completeness, it therefore suffices to prove the following claim, stating that when f is a tautology, every path in the constructed proof tree will end in a leaf that is a weakening of the axiom.

Claim 8. *If f is a tautology, along every path in this constructed tree, we will eventually reach a leaf where some formula appears on both the left and right side of the sequent (that is, a weakening of the axiom).*

Proof of Claim 8. Assume for sake of contradiction that f is a tautology, and there is some "bad" path p in the proof tree such that the leaf of p is labelled with the atomic sequent:

$$X_{i_1}, \ldots, X_{i_c} \longrightarrow X_{j_1}, \ldots, X_{j_d} \tag{1}$$

where the variables on the left and right of the arrow are disjoint. Then we show how to construct an assignment α to the underlying variables of f that falsifies f, this contradicting our assumption that f is a tautology. We construct α by setting all variables that occur on the left side of sequent (1) to true, and setting all variables on the right side of (1) to false. (That is, we set $X_{i_1}, \ldots, X_{i_c}$ to 1, and we set $X_{j_1}, \ldots, X_{j_d}$ to 0.) If any variable does not occur on either the left or the right side of this sequent then we set it to 0. Note that the assignment α is well-defined since no variable in f occurs on both the left and right side of (1) by our assumption that p is a bad path. Also α clearly falsifies the leaf sequent (1). Next we notice that the logical rules satisfy the **inversion property**: if there exists an assignment

α falsifying the lower sequent of a rule, then α also falsifies all upper sequents of the rule. Thus by the inversion property we have that *all* sequents on the path p from (1) to the root are falsified by α. Then since the root of p is labelled by the sequent/formula f, this implies that f is falsified by α, which contradicts our assumption that f is a tautology. □

This completes the proof of completeness. □

2 Restricted Frege Systems

In this section, we introduce Frege systems for smaller circuit classes such as low-depth circuits. We begin by the general definition of $\mathcal{C}$-Frege, then discuss how to define such systems for unbounded fan-in circuit classes. For simplicity, we use variants of PK to define these restricted systems.

2.1 $\mathcal{C}$-Frege

Let $\mathcal{C}$ be any circuit class. The $\mathcal{C}$-Frege system is exactly the same as PK except that for each application of the cut rule, the cut formula A must be in the class $\mathcal{C}$.

It is worth noting that if we restrict our cut formula to be in $\mathcal{C}$, and if the tautology we are proving is also in $\mathcal{C}$ (e.g., a DNF), then any formula in the proof is also in $\mathcal{C}$ by the subformula property.

2.2 Unbounded Fan-In Frege and AC^0-Frege

We will mostly talk about $\mathcal{C}$-Frege for constant-depth unbounded fan-in circuits. For such systems, we need to additionally include logical rules for unbounded fan-in $\wedge$ and $\vee$:

$$\frac{\Gamma \longrightarrow \Delta, A_1 \qquad \Gamma \longrightarrow \Delta, \wedge(A_2, \ldots, A_n)}{\Gamma \longrightarrow \Delta, \wedge(A_1, A_2, \ldots, A_n)} \ (\wedge\text{-right}) \qquad \frac{A_1, \wedge(A_2, \ldots, A_n), \Gamma \longrightarrow \Delta}{\wedge(A_1, A_2, \ldots, A_n), \Gamma \longrightarrow \Delta} \ (\wedge\text{-left})$$

$$\frac{\Gamma \longrightarrow \Delta, A_1, \vee(A_2, \ldots, A_n)}{\Gamma \longrightarrow \Delta, \vee(A_1, A_2, \ldots, A_n)} \ (\vee\text{-right}) \qquad \frac{A_1, \Gamma \longrightarrow \Delta \qquad \vee(A_2, \ldots, A_n), \Gamma \longrightarrow \Delta}{\vee(A_1, A_2, \ldots, A_n), \Gamma \longrightarrow \Delta} \ (\vee\text{-left})$$

In particular, AC^0-Frege restricts all cut formulas in the proof to have constant depth.

2.3 $\mathsf{AC}^0[2]$-Frege

To define $\mathsf{AC}^0[2]$-Frege, we can use the same unbounded $\vee$ and $\wedge$ logical rules as AC^0-Frege and add rules for unbounded parity gates rules. For $b \in \{0, 1\}$, let $\oplus_b(x_1, \ldots, x_n)$ be the connective indicating $\bigoplus x_i = b$. The logical rules for $\oplus_b$ is the following:

$$\frac{A_1, \Gamma \longrightarrow \Delta, \oplus_{1-b}(A_2, \ldots, A_n) \qquad \Gamma \longrightarrow \Delta, A_1, \oplus_b(A_2, \ldots, A_n)}{\Gamma \longrightarrow \Delta, \oplus_b(A_1, \ldots, A_n)} \ (\oplus\text{-right})$$

$$\frac{A_1, \oplus_{1-b}(A_2, \ldots, A_n), \Gamma \longrightarrow \Delta \qquad \oplus_b(A_2, \ldots, A_n), \Gamma \longrightarrow \Delta, A_1}{\oplus_b(A_1, \ldots, A_n), \Gamma \longrightarrow \Delta} \ (\oplus\text{-left})$$

3 Properties of Frege Systems

3.1 Robustness of Frege Systems

Cook and Reckhow [CR79] proved that all Frege systems are p-equivalent to each other. This even applies to Frege systems with different complete constant-fanin bases. The proofs involved many manipulations which showed how one could efficiently translate proofs in one Frege system to another Frege system. There are many other systems that are neither Hilbert-like or Gentzen-like that are also p-equivalent to Frege systems, such as natural deduction and Tait calculus (see Example 5).

Note that this does not include $\mathcal{C}$-Frege systems for restricted circuit classes $\mathcal{C}$. Indeed, the pigeon-hole principle has a polynomial-size proof in Frege, but we will show in Lecture 6 that it requires exponential-size AC^0-Frege proofs.

3.2 Normal Form for Frege Systems

We now introduce the notion of a Frege proof in Normal Form and a theorem regarding the existence and size of such proofs.

Definition 9. *Let π be a Frege proof of some formula F consisting of s sequents. We say that π is balanced if the depth of π is at most $O(\log s)$.*

Theorem 10 (Frege Normal Form)**.** *Let π be a Frege proof of F. Then there exists another Frege Proof π' of f such that: (i) π' is tree-like and balanced, and (ii) $|\pi'| \leq \mathsf{poly}(|\pi|)$.*

We give a sketch of the main ideas in the proof of the above theorem. Let π be a Frege proof of F.

1. We first convert π to a tree-like proof of size polynomial in the size of π, via the following steps:
   (a) We can first place each line of the proof f_i with $f_1 \wedge f_2 \wedge \cdots \wedge f_i$, where our lines are indexed by a topological ordering of the proof DAG.
   (b) Give a tree-like proof of $\bigwedge_{j=1}^{i+1} f_j$ from $\bigwedge_{j=1}^{i} f_j$
2. The second step is to convert the tree-like proof from the previous step into a balanced-tree-like proof.
   (a) Find line/vertex ℓ^* in the proof tree π such that the size of the subtree of π rooted at ℓ^* is roughly half of the size of the tree π. That is, $\frac{1}{3}|Tree(f)| \leq |Tree(\ell^*)| \leq \frac{2}{3}|Tree(f)|$, where $|Tree(f)|$ is the size of the subtree of π with root labelled by f.
   (b) Inductively give balanced derivations of $\ell^* \to f$ and $\to \ell^*$.
   (c) Then use apply the cut rule on $\ell^* \to f$ and $\to \ell^*$ to derive a balanced tree-like proof of f.

3.3 Extended Frege systems and its robustness

The *Extended Frege* (EF) system is any Frege system plus an *extension rule*

$$y \leftrightarrow A(x)$$

where y is a new variable that does not appear in earlier lines of the proof. This can be equivalently viewed as a generalization of the standard Frege system where each line is a Boolean *circuit* instead of a propositional formula.

Similar to the earlier results proved by Cook and Reckhow [CR79], different extended Frege systems are p-equivalent to each other. There are also other types of generalization to the Frege system that turn out to be p-equivalent to Extended Frege. For example, it is known that Substitution Frege,Renaming Frege, 0/1 Substitution Frege (SF) and Hajos Calculus are all p-equivalent to Extended Frege. (See [Dow85; PU92; Bus95] for these equivalences.)

Open Problem: There is one variant of the Frege system that is not known to be p-equivalent to either Frege or Extended Frege: Permuataion Frege. It is known that it p-simulates Frege and can be p-simulated by Extended Frege, giving

$$\text{Frege} \leq \text{Permutation Frege} \leq \text{Extended Frege},$$

but it is open whether either of the inequalities above are proper. (See [Bus95] for details.)

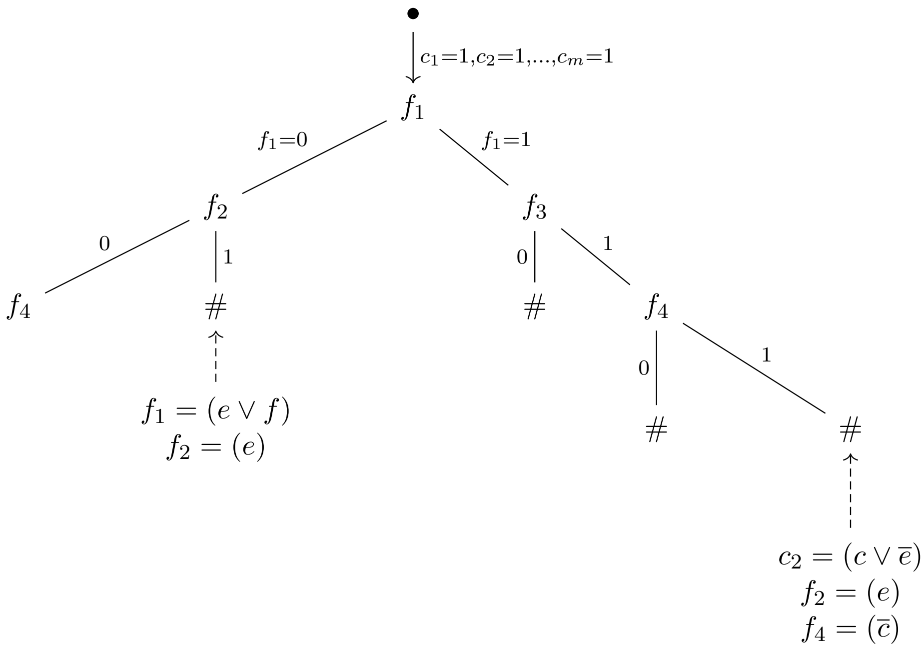
3.4 Equivalence with Prover-Delayer Games

Similar to Resolution, Frege systems also have an equivalent formulation as Prover-Delayer games. A tautology f has a polynomial-size Frege proof if and only if there is a polynomial-size Frege Prover-Delayer game for $\neg f$.

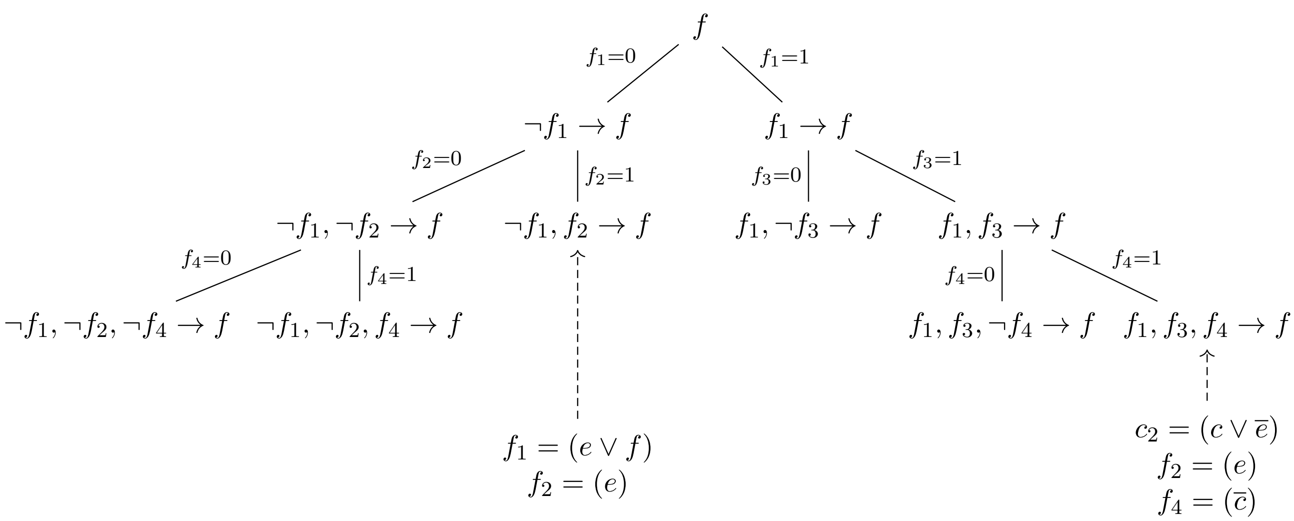
Definition 11 (Frege Prover-Delayer game [PB95])**.** *Let $f = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ be an unsatisfiable CNF. The Prover-Delayer game for a Frege system is a combinatorial game played between two parties called the "Prover" and the "Delayer". The Prover tries to show that the formula f is indeed unsatisfiable.*

The Delayer first claims they have a assignment α that satisfies f. In each round, the Prover tries to challenge the Delayer by querying the truth values of an arbitrary formula f_q and the Delayer responds by providing the truth value under their supposed assignment. This forms a decision tree where nodes are formulas queried by the Prover and branches are truth values provided by the Delayer. The game ends when every path on the tree ends in a "truth table" contradiction, in the sense that the truth value of the formulas on the path semantically contradicts with f. (See Figure 2 for an example).

Then, a proof consists of a tree of cuts, with simple derivations at leaves. (See [PB95] for details.) Note that this implicitly gives another normal form for Frege proofs.



(a) A high-level example



(b) An expanded example.

Figure 2: Example of the Frege Prover-Delayer game

3.5 "Complete" Axiom for Frege and extended Frege

Definition 12. *Let* $\mathsf{Soundness}_{\text{Frege}}(s)$ *be the propositional formula:*

$$\mathsf{Proof}_{\text{Frege}}(f,\pi) \rightarrow \mathsf{SAT}(f,\alpha)$$

where $\mathsf{Proof}_{\text{Frege}}(f,\pi)$ *states that* π *is a size* s *Frege proof of* f *and* $\mathsf{SAT}(f,\alpha)$ *states that* α *is a satisfying assignment for f. In a bit more detail, there are three vectors of variables that underlie* $\mathsf{Soundness}_{\text{Frege}}(s)$*: The* x*-variables describe the Boolean formula* f*; the* y*-variables describe an alleged Frege proof* π *of size* s*, and the* z*-variables describe a truth assignment to the underlying variables of* f*. The constraints state that if* π *is a legal Frege proof of* f*, then every assignment* α *to the variables of* f *satisfies* f*.*

Theorem 13. *(1)* $\mathsf{Soundness}_{\text{Frege}}(s)$ *has a polynomial-size Frege proof.*

(2) For any proof system $\mathcal{P}$ *that p-simulates depth-2 Frege,* $\mathcal{P} + \mathsf{Soundness}_{\text{Frege}}(s)$*, which is* $\mathcal{P}$ *extended with the axiom* $\mathsf{Soundness}_{\text{Frege}}(s)$*, p-simulates Frege.*

A similar theorem also holds for Extended Frege. Therefore, Frege + $\mathsf{Soundness}_{\text{EF}}$ is p-equivalent to Extended Frege, which means that Frege p-simulates EF *if and only if* $\mathsf{Soundness}_{\text{EF}}$ has a polynomial-size proof in Frege. This suggests potential ways of proving the equivalence or separation between Frege and extended Frege. However, $\mathsf{Soundness}_{\text{EF}}$ is an artificial formula that is hard to argue with directly. Luckily, Avigad [Avi97] gave a natural *combinatorial* principle that is p-equivalent to $\mathsf{Soundness}_{\text{EF}}$. Hence, proving the equivalence between Frege and Extended Frege boils down to showing a polynomial-size Frege proof for the combinatorial principle.

References

[Avi97] Jeremy Avigad. “Plausibly hard combinatorial tautologies”. In: *Proof complexity and feasible arithmetics* (1997), pp. 1–12 (cit. on p. 10).

[Bus95] S.R. Buss. “Some remarks on lengths of propositional proofs”. In: *Arch Math Logic* 34 (1995), pp. 377–394 (cit. on p. 8).

[CR79] Stephen A. Cook and Robert A. Reckhow. “The relative efficiency of propositional proof systems”. In: *Journal of Symbolic Logic* 44.1 (1979), pp. 36–50. DOI: 10.2307/2273702 (cit. on pp. 1, 2, 7, 8).

[Dow85] Martin Dowd. “Model-theoretic aspects of P= NP”. In: *Unpublished manuscript* (1985) (cit. on p. 8).

[PU92] T. Pitassi and A. Urquhart. “The complexity of the Hajos calculus”. In: *Proceedings., 33rd Annual Symposium on Foundations of Computer Science*. 1992, pp. 187–196. DOI: 10.1109/SFCS.1992.267773 (cit. on p. 8).

[PB95] Pavel Pudlák and Samuel R Buss. “How to lie without being (easily) convicted and the lengths of proofs in propositional calculus”. In: *Computer Science Logic: 8th Workshop, CSL’94 Kazimierz, Poland, September 25–30, 1994 Selected Papers 8*. Springer. 1995, pp. 151–162 (cit. on p. 8).