Propositional Proof Systems (p. 247-257)

Petri Savola

Laboratory for Theoretical Computer Science, TKK

8.10.2007

5.1 Introduction

Motivation: Relation to boolean circuits and open problems in complexity theory (e.g. co-NP ?= NP).

Definition: Tautology is a propositional formula which is true in every truth assignment. If $\emptyset \models T$, then T is a tautology.

Tautologies can be proved with different proof systems. The length (or complexity) of the proof depends on axioms and rules of interference of the system.

There are many different proof systems, including:

- Gentzen propositional sequent calculus (LK)
- resolution (R)
- Nullstellensatz systems (NS)
- polynomial calculus (PC)
- cutting planes (CP)
- propositional treshold calculus (PTK)
- Frege systems (F)
- extended Frege systems (EF)

Example: A Frege system using only connectives $\neg$ and $\rightarrow$.

Axioms:

1. $F \rightarrow (G \rightarrow F)$
2. $(F \rightarrow (G \rightarrow H)) \rightarrow ((F \rightarrow G) \rightarrow (F \rightarrow H))$
3. $(\neg F \rightarrow \neg G) \rightarrow ((\neg F \rightarrow \neg G) \rightarrow F)$

The only rule of inference, $modus\ ponens$:

$\frac{p, p \rightarrow q}{q}$

One can prove every propositional tautology using these axioms and modus ponens.

Some notations

If formula F can be derived from T (set of formulas), we denote $T \vdash F$. This means there is a sequence $P = (F_1, ..., F_n)$ such that $F_n = F$, and for each $1 \leq i \leq n$, F_i either belongs to set T or is derived from the previous formulas F_j, where $i > j$.

If $\emptyset \vdash F$ (or $\vdash F$) then F is a theorem and derivation P is the proof of F. Proof system $\mathcal{P}$, which was used, is indicated by notation $T \vdash_{\mathcal{P}} F$

More definitions

Proof system $\mathcal{P}$ is sound if every theorem F of $\mathcal{P}$ is valid ($\models F$).
Moreover, $\mathcal{P}$ is implicationally complete if for any propositional formulas $F_1, ..., F_k, G$ it is the case that $F_1, ..., F_k \models G$ implies $F_1, ..., F_k \vdash_{\mathcal{P}} G$.

Length of proof $P = (F_1, ..., F_n)$ is n (the number of inferences or steps).

The size of proof P is $\sum_{i=1}^{n} |F_i|$, where $|F_i|$ is the number of symbols in F_i.

5.2 Complexity of Proofs

Generalization of a proof system

Let Σ_1 and Σ_2 be finite alphabets such that their cardinality is two or greater and let $L \subseteq \Sigma_2^*$. Propositional proof system for L is a polynomial time computable surjection $f : \Sigma_1^* \to L$.

Note that typically L is collection $TAUT$ and for example in the De Morgan basis $\Sigma_1 = \{0, 1, \neg, \wedge, x,' (',')'\}$. Different variables can be represented as string xb where b is a binary number.

Proof system $f : \Sigma^* \to L$ is polynomially bounded if there is a polynomial p such that

$$(\forall x \in L)(\exists y \in \Sigma^*)(f(y) = x \wedge |y| \leq p(|x|)) \qquad (1)$$

Theorem 5.2.1

NP = co-NP $\Leftrightarrow$ There is a polynomially bounded propositional proof system for $TAUT$.

$Proof.$ $x \in TAUT \Leftrightarrow \neg x \in UNSAT$.
Thus, $\neg x \notin TAUT \Leftrightarrow x \notin UNSAT \Leftrightarrow x \in SAT$.
Because SAT is NP-complete, $TAUT$ must be co-NP-complete.

$'' \Rightarrow''$: Let $\Sigma = \{0, 1, \neg, \wedge, x,' (',')'\}$. $TAUT \in$ co-NP, so $\overline{TAUT} \in$ NP.
Hence there is a polynomial p and a polynomial time computable relation R such that $\forall x : x \in TAUT \Leftrightarrow (\exists y \in \Sigma^*)(R(x, y) \wedge |y| \leq p(|x|))$.

This makes sense, because a nondeterministic Turing machine on input x can guess y and verify that y is correct by $R(x, y)$.

Define propositional proof system $f : (\Sigma \cup \{'<',',' >'\})^* \rightarrow TAUT$ by $f(w) = x$, if $\exists y : R(x, y) \wedge w =< x, y >$ and $f(w) = p \vee \neg p$ otherwise.
Now f is polynomially bounded.

$'' \Leftarrow''$: Let $f : \Gamma^* \rightarrow TAUT$ be a polynomially bounded propositional proof system for $TAUT$. Let p satisfy the corresponding definition: $\forall x : x \in TAUT \Leftrightarrow (\exists y \in \Gamma^*)(f(y) = x \wedge |y| \leq p(|x|))$. From this definition we obtain that $TAUT \in$ NP. Assume $R \in$ co-NP. Because $TAUT$ is co-NP-complete, R is polynomially reducible to $TAUT$. Because $TAUT \in$ NP so is R. Thus, co-NP = NP. $\square$

Note that Theorem 5.2.1 holds for any finite, adequate set of connectives.

Definition

A propositional proof system T is automatizable if there is an algorithm A_T, which given any propositional formula A yields a proof in T of A in time polynomial in size of A, provided that such exists.

New definitions

- Propositional connective is a function symbol of given arity
- Formula in the set κ of connectives is a finite, rooted, ordered, labeled tree, which is either a single node labeled by a variable or whose root is labeled by a connective of arity n from κ, and whose children $F_1, ..., F_n$ are formulas
- The size of formula F, denoted by $|F|$, is the total number of symbols in F
- The formula size $(f(F))$ is the total number of connectives in F
- The circuit size $(c(F))$ is the number of distinct subformulas in F
- The leaf size $(||F||)$ is the number of occurrences of variables in F
- The root is called principal connective

Example: Frege system $\Sigma = \{x, 0, 1, \neg, \rightarrow\}$.

$|x_i| = 1 + |i|$
$|\neg F| = 1 + |F|$
$|F \rightarrow G| = 1 + |F| + |G|$
$||x_i|| = 1$
$||\neg F|| = ||F||$
$||F \rightarrow G|| = ||F|| + ||G||$
$f(F) =$ "number of gates in the formula tree"
$c(F) =$ "minimum number of gates in a circuit which represents F"

Assume that all connectives of formula F have arity at most k and there are never two successive occurances of a unary connective and variables appearing in F are $x_1, ..., x_m$, where $m = ||F||$.

Then $f(F) + ||F||$ is the number of nodes in the formula tree. Clearly $||F|| \leq |F| = \mathcal{O}(||F|| log_2 ||F||)$.

Definitions

For proof system F and tautology T, $size_F(T)$ is the minimum size of proof P of T in system F. Relations between different types of size can be easily found (e.g. $c(F) \leq f(F)$).

Total truth assginment is a mapping $\sigma : \{x_1, ..., x_n\} \rightarrow \{0, 1\}$. A boolean function $f \in \mathcal{B}_n$ is represented by formula F if $f(\sigma) = F \restriction_\sigma$ for all total truth assignments in $\{0, 1\}^n$.

A set κ of connectives is adequate if every boolean function can be represented by a formula in κ. A tautology $T \in TAUT_\kappa$ is a tautology in the connective set κ. Similarly, $Form_\kappa$ is the set of formulas in connective set κ. Let $Form$ denote the set of formulas over the De Morgan set $\{0, 1, \neg, \vee, \wedge\}$ of connectives.

Theorem 5.2.2

There is a polynomial time computable translation $tr : Form_\kappa \rightarrow Form$ satisfying $tr(F) \equiv F$ for all $F \in Form_\kappa$, and which is surjective in the sense that for every $G \in Form$ there exists $F \in Form_\kappa$ such that $tr(F) \equiv G$.

$Proof$. Left for an optional home excercise.

Note that now Theorem 5.2.1 holds for $TAUT_\kappa$ in place of $TAUT$. Theorem 5.2.1 also implies that if no propositional proof system is polynomially bounded for $TAUT$ then $P \neq NP$.

Pigeonhole principle

If $n+1$ pigeons occupy n pigeonholes at least one hole must be occupied by at least two pigeons. This example demonstrates how this can be written as a propositional logic formula. Let m be the number of pigeons and $m > n$, and let $p_{i,j}$ be a propositional variable, whose interpretation is that the pigeon i sits in hole j.

$$\neg \bigwedge_{i=1}^{m} \bigvee_{j=1}^{n} p_{i,j} \vee \bigvee_{1 \leq i < i' \leq m} \bigvee_{j=1}^{n} (p_{i,j} \wedge p_{i',j}) \tag{2}$$

This formula is naturally a tautology with $\mathcal{O}(m^2 n)$ symbols. The formula expresses that there is no injective relation from set of size m into a set of size n.

Last definitions before the next chapter

Let f, g be proof systems such that $f : \Sigma_1^* \rightarrow TAUT$ and $g : \Sigma_2^* \rightarrow TAUT$. Then g p-simulates f if there is a polynomial time computable function $h : \Sigma_1^* \rightarrow \Sigma_2^*$ such that $g(h(x)) = f(x)$ for all $x \in \Sigma_1^*$.

Alternatively if h is polynomially bounded, but not necessarily polynomial time computable, let $\mathcal{P}_1$ and $\mathcal{P}_2$ be arbitary proof systems for propositional logic. System $\mathcal{P}_1$ simulates $\mathcal{P}_2$ if and only if there is a polynomial $p(x)$ such that for any proof Q of formula A in $\mathcal{P}_2$ there is a proof P of A in $\mathcal{P}_1$ and $size(P) \geq p(size(Q))$.

If $\mathcal{P}_1$ and $\mathcal{P}_2$ have the same language then the simulation is said to be strong.

5.3 Gentzen Sequent Calculus

- Connectives: $\neg, \vee, \wedge$
- Cedent is a finite set of propositional formulas, typically denoted with large Greek letters
- $\Gamma \mapsto \Delta$ is a sequent if Γ and Δ are cedents.
- Γ is antcendent and Δ is succedent
- Γ, Δ is an abbreviation of $\Gamma \cup \Delta$

Notice similarity between $\mapsto$ and $\vdash$. If one wants to think in such a way, the meaning of $\Gamma \mapsto \Delta$ is $\bigwedge \Gamma \rightarrow \bigvee \Delta$.

Rules of inference

$\neg - left : \frac{\Gamma \mapsto \Phi, \Delta}{\neg\Phi, \Gamma \mapsto \Delta}$ $\quad$ $\neg - right : \frac{\Phi, \Gamma \mapsto \Delta}{\Gamma \mapsto \neg\Phi, \Delta}$

$\vee - left : \frac{\Phi, \Gamma \mapsto \Delta \qquad \Psi, \Gamma \mapsto \Delta}{\Phi \vee \Psi, \Gamma \mapsto \Delta}$

$\vee - right : \frac{\Gamma \mapsto \Phi, \Delta}{\Gamma \mapsto \Phi \vee \Psi, \Delta}$ $\quad$ $\vee - right : \frac{\Gamma \mapsto \Phi, \Delta}{\Gamma \mapsto \Psi \vee \Phi, \Delta}$

$\wedge - left : \frac{\Phi, \Gamma \mapsto \Delta}{\Phi \wedge \Psi, \Gamma \mapsto \Delta}$ $\quad$ $\wedge - left : \frac{\Phi, \Gamma \mapsto \Delta}{\Psi \wedge \Phi, \Gamma \mapsto \Delta}$

$\wedge - right : \frac{\Gamma \mapsto \Phi, \Delta \qquad \Gamma \mapsto \Psi, \Delta}{\Gamma \mapsto \Phi \wedge \Psi, \Delta}$

$cut : \frac{\Gamma \mapsto \Phi, \Delta \qquad \Phi, \Gamma \mapsto \Delta}{\Gamma \mapsto \Delta}$

$structural : \frac{\Gamma \mapsto \Delta}{\Gamma' \mapsto \Delta'}$ $(\Gamma \subseteq \Gamma', \Delta \subseteq \Delta')$

The only axioms are of the form $p \mapsto p$, where p is a propositional variable.

A proof of $\Gamma \mapsto \Delta$ is a sequence P of sequents $S_1, ..., S_n$ such that S_n is the end sequent of $\Gamma \mapsto \Delta$.

A proof is tree-like if each sequent is used at most once as the hypothesis of a rule. A tree-like proof $\Gamma \mapsto \Delta$ is thus a tree, satisfying:

- $\Gamma \mapsto \Delta$ is the root
- Leaves are exioms
- Every node other than the root is an upper sequent of a rule
- Every node other than a leaf is a lower sequent of a rule

A proof without the cut rule is called cut-free.

The size $S(\Pi)$ of derivation $\Pi = (\Phi_1, ..., \Phi_n)$ is the total number of symbols in Π. The length $L(\Pi)$ is n. If Φ is a tautology then $S(\Phi)$ $(S_T(\Phi))$ is $S(\Pi)$, where Π is the smallest proof (tree-like proof) of Φ. Similar statement holds for length L.

Example derivation

Problem. Derive $T = A \vee \neg A$ using LK. Clearly T is a tautology.

$\dfrac{}{A \mapsto A} \Rightarrow (\neg R)$

$\dfrac{A \mapsto A}{\mapsto \neg A, A} \Rightarrow (\vee R)$

$\dfrac{\mapsto \neg A, A}{\mapsto A \vee \neg A, A} \Rightarrow$

$\dfrac{\mapsto A \vee \neg A, A}{\mapsto A, A \vee \neg A} \Rightarrow (\vee R)$

$\dfrac{\mapsto A, A \vee \neg A}{\mapsto A \vee \neg A, A, \vee \neg A} \Rightarrow (cut)$

$\dfrac{\mapsto A \vee \neg A, A, \vee \neg A}{\mapsto A \vee \neg A}$

Summary

- Example of Frege system
- Polynomial bounding for proof systems
- A propositional formula can be represented as a tree
- Combinatorial statements can be formalized into logical form
- Basics of Gentzen Sequent Calculus
- (Never come to TB353 at the wrong time!)