Witnessing and conditional independence results (Chapter 24 of FRVPC, §24.1–§24.3)

Y. F.

August 29, 2013

1 The theories involved

This chapter discusses relations between the theories PV, S^1, T^1 and computational complexity theory. The theories mentioned are:

- PV, which is the theory of polynomial time computations. It has symbols for all polynomial time functions, and axioms expressing relations among them. It also has open induction, which is induction in which the inductive predicate is only allowed to have sharply bounded quantifiers (quantifiers in which the variables are bounded by polynomials in the lengths of other variables). PV is a universal theory, that is it is axiomatized by axioms using only universal quantifiers.
- T^1, confusingly denoted in the book as T^1_2, adds to PV the induction axiom for Σ^b_1 predicates, which consist of a string of bounded existential quantifiers followed by a sharply bounded quantified formula.
- S^1, confusingly denoted in the book as S^1_2, adds to PV instead the *polynomial induction scheme* (or equivalently the *length induction scheme*) for Σ^b_1 formulas. These schemes, which are weaker than usual induction, are:
  - Polynomial induction: given $A(0)$ and $\forall x A(\lfloor x/2 \rfloor) \to A(x)$, deduce $\forall x A(x)$.
  - Length induction: given $A(0)$ and $\forall x A(x) \to A(x+1)$, deduce $\forall x A(|x|)$.

2 The weak pigeonhole principle and S^1

The first application is from Krajíček and Pudlák's *Some consequences of cryptographical conjectures for S^1_2 and EF*, and it concerns the weak pigeonhole principle for polynomial time functions. This principle states that if $f\colon \{0,1\}^{N+1} \to \{0,1\}^N$ is given by a circuit, then there are two inputs x, y of length N such that $f(x) = f(y)$. Technically, the parameter N needs to be the length of some variable. The goal is to show that S^1 cannot prove this principle, assuming that RSA is secure.

RSA is a cryptosystem in which a message m is encrypted by calculating $c = m^e \pmod{n}$, where $(e, \varphi(n)) = 1$. Here c, n, e are public, and the goal is to recover m. The message can be recovered if we could determine $d = e^{-1} \pmod{\varphi(n)}$, since $m = c^d \pmod{n}$. We will assume that RSA is secure in the following sense: given c, n, e, it is difficult to determine the LSB of m (this is a hard-core bit).

The idea is as follows: using the weak pigeonhole principle, we will find an exponent r such that $c^r = 1 \pmod{n}$. Let $r' = \mathrm{ord}_n c$ be the order of c modulo n, which divides both r and $\varphi(n)$. Since $(e, \varphi(n)) = 1$, we have $(e, r') = 1$ and so $r' | r/(e, r)$. Let $s = r/(e, r)$. Then $(s, e) = 1$ and $c^s = 1 \pmod{n}$. Since $(e, s) = 1$, we can find d satisfying $ed = 1 \pmod{s}$ using the extended GCD algorithm. Given d, we can recover m: $c^d = m^{ed} = m \pmod{n}$.

The original proof proceeds as follows. The function $t \mapsto c^t \pmod{n}$ is polynomial time, and so if the weak pigeonhole principle were true in S^1, then S^1 would prove the existence of $x \neq y$ of length $|n| + 1$ such that $c^x = c^y \pmod{n}$. Buss's witnessing theorem implies that we can find x, y in polynomial time. Given x, y, notice that $c^{x-y} = 1 \pmod{n}$, and so we can recover m.

We can mimic this proof in the new framework. Let F be the set of polynomial time functions computable given the inputs c, n. Since PV is a universal theory, $K(F)$ satisfies PV. If S^1 proved the weak pigeonhole principle then PV would also, since S^1 is $\forall\exists$-conservative over PV. In that case, the principle would be valid in $K(F)$, which we assume for the rest of the proof. Let $\alpha \in F$ be the function which computes a circuit $C\colon \{0,1\}^{|n|+1} \to \{0,1\}$ implementing the function $t \mapsto c^t \pmod{n}$ considered above. The weak pigeonhole principle implies that

$$[\![\exists\beta,\gamma.c^\beta = c^\gamma \pmod{n}]\!] = 1.$$

Closure properties of F (see below) guarantee that for every $\epsilon > 0$ there exist β, γ such that

$$\mu([\![c^\beta = c^\gamma \pmod{n}]\!]) > 1 - \epsilon$$

and so, by the definition of μ, for every $\epsilon > 0$ there exist β, γ such that

$$\Pr[c^{\beta(c,n)} = c^{\gamma(c,n)} \pmod{n}] > 1 - \epsilon.$$

In other words, there are algorithm enabling us to find x, y satisfying $c^x = c^y \pmod{n}$ with probability $1 - \epsilon$. Therefore we can break a random RSA pair c, n with probability $1 - \epsilon$ for every $\epsilon > 0$.

There is a small deterioration in the result going from the original proof to the new one: the original proof relied only on the worst-case hardness of RSA, whereas the new proof relies on average-case hardness. The same deterioration also occurs in the other examples in this chapter.

We needed a certain closure property of the family F, its being *closed under definition by cases by open formulas*: if $\alpha, \beta \in F$ and B is an open formula, then we need the following function to be in F:

$$\delta(\omega) = \begin{cases} \alpha(\omega) & \text{if } B(\alpha(\omega)), \\ \beta(\omega) & \text{otherwise.} \end{cases}$$

This is clearly the case for our family F, since polynomial time algorithms can implement conditionals. Given this, let us follow the argument. We started by assuming (after substituting a universal quantifier)

$$[\![\exists\beta,\gamma.c^\beta = c^\gamma \pmod{n}]\!] = 1.$$

Therefore there is a sequence (β_i, γ_i) satisfying

$$\bigvee_i [\![c^{\beta_i} = c^{\gamma_i} \pmod{n}]\!] = 1.$$

For every i, define

$$\delta_i, \epsilon_i = \begin{cases} \beta_0, \gamma_0 & \text{if } c^{\beta_0} = c^{\gamma_0} \pmod{n}, \\ \beta_1, \gamma_1 & \text{if } c^{\beta_0} \neq c^{\gamma_0} \pmod{n}, c^{\beta_1} = c^{\gamma_1} \pmod{n}, \\ \cdots & \cdots \\ \beta_i, \gamma_i & \text{otherwise.} \end{cases}$$

Note that $\delta_i, \epsilon_i \in F$ since F is closed under definition by cases by open formulas. Also,

$$[\![c^{\delta_i} = c^{\epsilon_i} \pmod{n}]\!] = \bigvee_{j \le i} [\![c^{\beta_j} = c^{\gamma_j} \pmod{n}]\!].$$

The σ-additivity of μ then implies that

$$\mu([\![c^{\delta_i} = c^{\epsilon_i} \pmod{n}]\!]) \to 1,$$

and so, since $\Pr[B]$ is the standard part of $\mu(B)$,

$$\Pr[c^{\delta_i} = c^{\epsilon_i} \pmod{n}] \to 1.$$

3 Restricted oracle classes and S^1 vs. T^1

The second application, from Krajíček's *Fragments of bounded arithmetic and bounded query classes* (see also Red Book §6.3,10.3) aims at a complexity-theoretic condition for separating S^1 and T^1.

A predicate is Σ_2^b-definable in T^1 if and only if it is in P^{NP}. What about the weaker theory S^1? It turns out that a predicate is Σ_2^b-definable in S^1 if and only if it is in the restricted oracle class $P^{NP}[O(\log n)]$, which consists of those predicates computable using $O(\log n)$ queries to an NP-complete oracle, queries which in addition to a Boolean answer also return a witness in the positive case. (The class is also equal to L^{NP}.) It follows that $S^1 \neq T^1$ given $P^{NP} \neq P^{NP}[O(\log n)]$. One can come up with an oracle relative to which the latter statement holds, and so $S^1(\alpha) \neq T^1(\alpha)$ (theories including an additional uninterpreted predicate α which is subject to the induction axioms), that is there is a relativized separation between S^1 and T^1.

It is straightforward to see that all predicates in $P^{NP}[O(\log n)]$ are Σ_2^b-definable in S^1. The difficult part is showing the converse. The original proof uses cut elimination followed by case-by-case analysis of all the derivation rules, in the spirit of Buss's witnessing theorem and other witnessing arguments.

The new proof considers the set F of all functions in $P^{NP}[O(\log n)]$, where n is a fixed non-standard integer. That is, each function in F depends on n inputs ω and is computed by a $P^{NP}[c \log n]$ machine for some standard $c > 0$. As in the preceding section, PV is automatically valid in $K(F)$. To show that moreover S^1 is valid in F, we prove the *bounded function minimization scheme*, which is equivalent to polynomial induction.

The bounded function minimization scheme states that every function computed by some circuit C on $|a|$ inputs and $\log |a|$ outputs attains a minimal value at some point u. The minimal value itself can be found using binary search, which only requires $\log |a| = O(\log n)$ oracle queries (since $|a|$ is polynomial in n), and the point u can be recovered from the corresponding witness. This construction shows that $K(F)$ is a model of S^1.

Every function in P^{NP} is Σ_2^b-definable in T^1. If T^1 were valid in $K(F)$ then every function $f \in P^{NP}$ would be Σ_2^b-definable in $K(F)$, that is for some Σ_2^b-formula A_f, $\forall x \exists y A_f(x,y)$ would be valid, where $A_f(x,y) = \exists z \forall t A(x,y,z,t)$ formalizes $f(x) = y$. A witnessing argument, relying on the same property of F as in the previous section but more complicated, shows that for every $\epsilon > 0$ we can find β, γ such that

$$\mu([\![\forall t A(\omega, \beta, \gamma, t)]\!]) \geq 1 - \epsilon.$$

(Recall ω is the input to each function in F.) A simple argument now allows us to deduce that for every $\epsilon > 0$ we can find β, γ such that

$$\Pr[\forall t A(\omega, \beta(\omega), \gamma(\omega), t)] \geq 1 - \epsilon.$$

The function $\beta(\omega) \in P^{NP}(O(\log n))$ thus computes f with probability $1 - \epsilon$, implying that every function in P^{NP} can be calculated (with arbitrarily small error) in $P^{NP}[O(\log n)]$. (Notice that again the new framework produced a weaker result.)

4 Polynomial size circuits for SAT and S^1 vs. PV

The final application, taken from Krajíček, Pudlák and Takeuti's *Bounded arithmetic and the polynomial hierarchy* (see also Red Book §10.2), concerns a condition for separating PV from S^1. This paper proved the celebrated KPT theorem, which was uses to show that if SAT has no polynomial size circuits then $PV \neq S^1$.

The idea of the original proof is as follows. Suppose that $S^1 = PV$. Let $R(\langle v_1, \ldots, v_r \rangle, \langle w_1, \ldots, w_s \rangle)$, where $v_1, \ldots, v_r$ are formulas, $w_1, \ldots, w_s$ are truth assignments, and $r \leq s$, be the polynomial time predicate stating that w_i is a satisfying assignment for v_i for all $1 \leq i \leq s$. Since $R(\langle v_1, \ldots, v_r \rangle, \langle \rangle)$ holds, S^1 proves that there is a maximal s such that $R(\langle v_1, \ldots, v_r \rangle, \langle w_1, \ldots, w_s \rangle)$ for some $w_1, \ldots, w_s$; we call such $w_1, \ldots, w_s$ a maximal satisfying assignment for $v_1, \ldots, v_r$. If $S^1 = PV$ then PV also proves this, and we can apply the KPT theorem to conclude that there are functions $f_1, \ldots, f_k \in FP$ always outputting satisfying assignments for their inputs such that one of the following holds:

- $f_1(v_1, \ldots, v_r)$ is a maximal satisfying assignment for $v_1, \ldots, v_r$.

- $f_2(v_1, \dots, v_r; b_1)$ is a maximal satisfying assignment for $v_1, \dots, v_r$, where b_1 is a counterexample to $f_1(v_1, \dots, v_r)$ being maximal.
- $f_3(v_1, \dots, v_r; b_1, b_2)$ is a maximal satisfying assignment for $v_1, \dots, v_r$, where b_1 is the counterexample for f_1, and b_2 is a counterexample for f_2.
- ...
- $f_k(v_1, \dots, v_r; b_1, \dots, b_{k-1})$ is a maximal satisfying assignment for $v_1, \dots, v_r$.

Let V_1 be the set of satisfiable formulas of size n and X the set of satisfying assignments, where $v \in V_1$ has a satisfying assignment $w(v) \in X$. We define a function $F \colon V_1^k \to [k] \times X$ as follows: given k satisfiable formulas $v_1, \dots, v_k \in V_1$, let ℓ be the minimal index such that $|f_\ell(v_1, \dots, v_k; w(v_1), \dots, w(v_{\ell-1}))| \geq \ell$, and let w be the satisfying assignment produced by the algorithm f_ℓ for v_ℓ. Then $F(v_1, \dots, v_k) = (\ell, w)$.

For $Q \in \binom{Q_1}{k-1}$ and $v \in V_1 \backslash Q$, we say that Q helps v if for some order of $Q \cup \{v\}$, $F(Q \cup \{v\}) = (\ell, w)$, where ℓ is the index of v. On the one hand, the number of pairs (Q, v) such that Q helps v is $\binom{|V_1|}{k}$, and on the other hand there are only $\binom{|V_1|}{k-1}$ different subsets Q. Therefore some Q_1 helps at least $\binom{|V_1|}{k} / \binom{|V_1|}{k-1} = (|V_1| - k + 1)/k$ different $v \in V_1$. Remove those to form a new set V_2, and repeat the construction to obtain Q_2 which helps at least $(|V_2| - k + 1)/k$ different $v \in V_2$. Remove those to form a new set V_3, and repeat. It is not difficult to calculate that $|V_t| < (1 - 1/k)^t |V_1| + k$, and so $|V_t| \leq k$ for $t = O(n)$.

For a given input length n, let $a(n)$ consist of the satisfying assignments for all formulas in $Q_1, \dots, Q_t, V_t$ for $t = O(n)$ considered above. Given an arbitrary $v \in Q_1$, we can find a satisfying assignment for it given $a(n)$ by first checking whether v has a satisfying assignment in $a(n)$, and otherwise trying all possible Q_i (and all possible permutations) in order to see if any of them helps v. Therefore we can determine whether an arbitrary v has a satisfying assignment in polynomial time given the polynomial advice $a(n)$. In other words, $NP \subseteq P/\text{poly}$.

The proof in the new framework is similar. Fix a non-standard integer n. The sample space Ω of the new structure consists of n-tuples $\omega = (\omega_1, \dots, \omega_n)$ of different satisfiable formulas of length n. The collection F is the smallest collection of functions defined on Ω which is closed under composition and contains PV and the function $h(\psi_1, \dots, \psi_n)$ defined on n-tuples of satisfying assignments as follows: if $\psi_1, \dots, \psi_{i-1}$ are satisfying assignments for $\omega_1, \dots, \omega_{i-1}$ but ψ_i is not a satisfying assignment for ω_n then $h(\psi_1, \dots, \psi_n) = \langle \psi_1, \dots, \psi_{i-1}, \psi, \psi_{i+1}, \dots, \psi_n \rangle$, where ψ is some fixed satisfying assignment for ω_i (depending only on the formula ω_i; this corresponds to the function w above).

Let $f(\langle \omega_1, \dots, \omega_n \rangle, \langle \psi_1, \dots, \psi_n \rangle)$ denote the maximal i such that $\psi_1, \dots, \psi_i$ are satisfying assignments for $\omega_1, \dots, \omega_i$. The theory S^1 proves that f attains a maximum. If $S^1 = PV$ then the same would hold in $K(F)$, which as in the previous cases is a model of PV. Hence

$$[\![\exists \psi_1, \dots, \psi_n \forall \chi_1, \dots, \chi_n f(\langle \omega_1, \dots, \omega_n \rangle, \langle \psi_1, \dots, \psi_n \rangle) \geq f(\langle \omega_1, \dots, \omega_n \rangle, \langle \chi_1, \dots, \chi_n \rangle)]\!] = 1.$$

As before, for every $\epsilon > 0$ there exist $\psi_1, \dots, \psi_n$ such that

$$\mu([\![\forall \chi_1, \dots, \chi_n f(\langle \omega_1, \dots, \omega_n \rangle, \langle \psi_1, \dots, \psi_n \rangle) \geq f(\langle \omega_1, \dots, \omega_n \rangle, \langle \chi_1, \dots, \chi_n \rangle)]\!]) \geq 1 - \epsilon.$$

In particular,

$$\mu([\![f(\langle \omega_1, \dots, \omega_n \rangle, \langle \psi_1, \dots, \psi_n \rangle \geq f(\langle \omega_1, \dots, \omega_n \rangle, h(\langle \psi_1, \dots, \psi_n \rangle))]\!]) \geq 1 - \epsilon.$$

Therefore for every $\epsilon > 0$, we can find $\psi_1, \dots, \psi_n$ such that with probability at least $1 - \epsilon$, $\psi_1, \dots, \psi_n$ are satisfying assignments for $\omega_1, \dots, \omega_n$.

Suppose that the computation of $\psi_1, \dots, \psi_n$ includes $k - 1$ calls to the function h. Marginalizing shows that we can fix $\omega_{k+1}, \dots, \omega_n$ to some values so that $\psi_1, \dots, \psi_k$ are satisfying assignments for $\omega_1, \dots, \omega_k$ with probability at least $1 - \epsilon$. Now we can run an argument similar to the one before (but accommodating the fact that there is an error probability of ϵ) to come up with a P/poly algorithm solving SAT on $1 - \epsilon$ of the instances.