All written documents allowed. No Internet access, no cell phone. The different sections are not independent.

1 CNF transforms

A propositional formula $F$ is in clausal form if and only if it is a conjunction ($\land$) of clauses, where each clause is a disjunction ($\lor$) of literals, and literals are either propositional variables $x$ or their negations $\neg x$.

SAT is the problem, given a formula in clausal form $F$, to decide whether $F$ is satisfiable, and is a well-known NP-complete problem.

The usual translation from a formula $F$ to a logically equivalent clausal form is exponential in time and space, in general. That translation is an algorithm which we call CNF: it takes a propositional formula $F$ as input, pushes negations inwards, and distributes $\land$ over $\lor$ until a clausal form is obtained.

The purpose of this section is to explore a more clever translation, due to Tseitin (1957), and which preserves satisfiability, not logical equivalence.

Let $F$ be a propositional formula, built from variables, negation $\neg$, truth $\top$, falsity $\bot$, binary conjunctions and disjunctions, and also binary exclusive or ($\oplus$) and $\iff$. Tseitin’s algorithm works as follows. For each non-variable subformula $G$ of $F$, we create a fresh variable $y_G$; for each variable $x$ occurring in $F$, we consider that the notation $y_x$ denotes $x$ itself; and we create the following clauses:

- for each non-variable subformula $G$ of $F$, say $G = G_1 \ op G_2$ (where $\ op \in \{\land, \lor, \oplus, \iff\}$), we create CNF($y_G = y_{G_1} \ op y_{G_2}$);
- we do the same for the unary operator $\neg$ (if $G = \neg G_1$, then we generate CNF($y_G = \neg y_{G_1}$)) and for the nullary operators (if $G = \top$, then we generate CNF($y_G = \top$), and similarly for $\bot$);
- finally, the unit clause $y_F$.

Let us call TSEITIN($F$) the conjunction of all the clauses thus produced on the input formula $F$.

Let $x_1, \ldots, x_m$ be an enumeration of the variables that occur in $F$. If $\rho$ is an assignment that satisfies $F$, then the assignment $\rho'$ that extends $\rho$ and maps
each of the fresh variables $y_G$ to the value of $G$ under $\rho$ satisfies TSEITIN($F$). Conversely, if $\rho'$ satisfies TSEITIN($F$), then one can show by induction on the subformula $G$ of $F$ that the value of $G$ under $\rho'$ is equal to $\rho'(y_G)$; in particular, since $\rho'$ satisfies the last clause $y_F$, $\rho'$ satisfies $F$. Hence TSEITIN preserves satisfiability.

**Question 1** Why does TSEITIN work in polynomial time? You will concentrate on the complexity of the various calls to CNF.

**Question 2** A propositional formula $F$ is uniquely satisfiable if and only if there is exactly one assignment $\rho$ of truth values for each of the variables $x_1, \ldots, x_m$ that occur in $F$, such that $\rho$ satisfies $F$. Show that $F$ is uniquely satisfiable if and only if TSEITIN($F$) is uniquely satisfiable.

## 2 The class MA

Recall that MA is the class of languages $L$ such that, for every $\ell \geq 0$, there is a language $D \in \mathcal{P}$ such that, for every input $x$, of size $n$:

- if $x \in L$ then there is a $y$ of size $p(n)$ such that $Pr_r[(x, y, r) \in D] \geq 1 - 1/2^{n^\ell}$;
- if $x \not\in L$ then for every $y$ of size $p(n)$, $Pr_r[(x, y, r) \in D] \leq 1/2^{n^\ell}$;

where the probabilities are taken over all random tapes $r$ of size $q(n)$, and $p(n)$ and $q(n)$ are two polynomials (which may depend on $\ell$).

**Question 3** Show that we obtain the same class by requiring no error in the $x \in L$ case. In other words, let MA$_0$ be the class defined as above, except for the clause:

- if $x \in L$ then there is a $y$ of size $p(n)$ such that, for every $r$, $(x, y, r) \in D$.

You must show that MA = MA$_0$. As a hint, you may imitate the proof of the Sipser-Gács-Lautemann Theorem (Proposition 1.24 in the second set of lecture notes, pcp.pdf).

**Question 4** Deduce that MA $\subseteq \Sigma_2^p$.

## 3 The Zachos Lemma

Let us recall the BP· operator from the lectures: for any complexity class $\mathcal{C}$, $\text{BP} \cdot \mathcal{C}$ is the class of languages $L$ such that there is a randomized polynomial time Turing machine $\mathcal{A}'$ and a language $D' \in \mathcal{C}$ such that, on input $x$ (of size $n$):
where probabilities are taken on random strings $r$ of size $q(n)$, for some polynomial $q$ in $n$.

Recall that an oracle machine is a multi-tape machine with a specific query tape, three extra control states $Q$, YES and NO. Let $A$ be a language. The semantics of the machine with oracle $A$ is as usual, except that when the machine reaches state $Q$, it then proceeds to state YES if the contents of the query tape is in $A$, and to NO otherwise. The relativized classes $P^A$, $NP^A$, $BPP^A$, etc., are obtained from their classical counterpart by changing the underlying Turing machine model to the corresponding oracle machine, with oracle $A$.

For a complexity class $C$, we write $NP^C$ for the union of the classes $NP^{L'}$, $L' \in C$.

It is clear that $C \subseteq C'$ implies $NP^C \subseteq NP^{C'}$.

**Question 5** Show that $NP^{BPP} \subseteq MA$.

**Question 6** Show the Zachos Lemma: if $NP \subseteq BPP$, then $PH \subseteq BPP$. Here is the proof, your task is to replace the “why?” questions by appropriate justifications. If $NP \subseteq BPP$, then:

\[
PH = \Sigma_2^P = NP^{NP} \subseteq NP^{BPP} \subseteq MA \subseteq AM \subseteq BP \cdot NP \subseteq BP \cdot BPP \subseteq BPP
\]

**Question 7** Let $F$ be a propositional formula in clausal form, built on propositional variables $x_1, \ldots, x_m$, say. Let $X$ be the set of environments (mappings from the propositional variables $x_1, \ldots, x_m$ to truth-values) $\rho$ that satisfy $F$ (in notation, $\rho \models F$). Let $m' \geq 2$ be a number such that $2^{m'-2} \leq |X| \leq 2^{m'-1}$,
where \(|X|\) is the cardinality of \(X\). Identify each environment \(\rho\) with the obvious vector in \(\Sigma^m\). Show that:

\[
Pr_{h,b}[\exists! \rho \in \Sigma^m \cdot \rho \models F \text{ and } h(\rho) = b] \geq \frac{1}{8}
\]

where \(h\) is drawn at random uniformly among all linear hash functions from \(\Sigma^m\) to \(\Sigma^{m'}\), and \(b\) is drawn at random uniformly, and independently, in \(\Sigma^{m'}\).

We write \(\exists!\) for “there exists a unique”. (Hint: given a fixed \(\rho\), find a lower bound for the probability of the event \(C_\rho(h,b)\), defined as holding whenever \(h(\rho) = b\) but \(h(\rho') \neq b\) for every \(\rho' \in X\) such that \(\rho' \neq \rho\).)

**Question 8** We take \(F\) and \(m\) as above, but we no longer assume that \(m'\) is known. Show that, if we draw \(m'\) at random uniformly among \(\{2, 3, \ldots, m+1\}\), and a linear hash function \(h : \Sigma^m \to \Sigma^{m'}\) and a vector \(b\) in \(\Sigma^{m'}\) at random as before, then:

- if \(F\) is satisfiable, then \(Pr_{m',h,b}[\exists! \rho \in \Sigma^m \cdot \rho \models F \text{ and } h(\rho) = b] \geq \frac{1}{8m}\).

**Question 9** Define a randomized polynomial time algorithm \(W\) that takes a propositional formula \(F\) in clausal form as input (on \(m\) variables \(x_1, \ldots, x_m\) as above) and returns a propositional formula \(F'\) in clausal form such that: (a) if \(F\) is satisfiable, then \(F'\) is uniquely satisfiable with probability at least \(1/(8m)\), and (b) if \(F\) is unsatisfiable, then \(F'\) is unsatisfiable.

**Question 10** On input \(F\) (a clausal form again), we now build \(k\) formulae \(F_1, \ldots, F_k\) in clausal form, by calling \(W\) \(k\) times, and where \(k\) is a parameter, depending polynomially on the size \(n\) of \(F\). Let \(\epsilon \in ]0, 1[\) be an arbitrary parameter (possibly depending on the size \(n\) of \(F\)). We wish to find \(k\) such that: (a) if \(F\) is satisfiable, then at least one of \(F_1, \ldots, F_k\) is uniquely satisfiable, with probability at least \(1 - \epsilon\), and (b) if \(F\) is unsatisfiable, then no formula \(F_i\) is uniquely satisfiable. Show that one can achieve this, by giving an explicit formula for \(k\) as a function of \(n\) and \(\epsilon\).

**Question 11** Deduce the Valiant-Vazirani theorem: if \(USAT \in \text{P}\), then \(\text{NP} = \text{RP}\).

Here \(USAT\) is the unique satisfiability problem: given a clausal form \(F\) (on variables \(x_1, \ldots, x_m\)), is there a unique \(\rho \in \Sigma^m\) that satisfies \(F\)?

We define another operator \(\oplus\) (“parity”) as follows: \(L \in \oplus \cdot \text{C}\) iff there is a language \(L'\) in \(\text{C}\), and a polynomial \(p(n)\), such that:

- \(x \in L\) iff the number of strings \(y\) of size \(p(n)\) such that \((x, y) \in L'\) is odd.

I.e., \(\oplus \cdot \text{P}\) is the class of languages decidable on a (balanced, i.e., binary branching and whose branches all have the same length) non-deterministic Turing machine by accepting iff the number of accepting branches is odd.

**Question 12** Using the same ideas as before, show that \(\text{NP} \subseteq \text{RP}^{\oplus \cdot \text{P}}\).