Lecture #14: Introduction to the Polynomial Hierarchy
Lecture Presentation

Getting "Close" to Being Satisfied

Once again, consider a Boolean formula $\mathcal{F}$, defined over the set of Boolean variables $\mathcal{V} = \{x_0, x_1, x_2, \dots\}$, such that $\mathcal{F}$ is in ***conjunctive normal form***. Then

$$\mathcal{F} = (C_1 \wedge C_2 \wedge \cdots \wedge C_s)$$

for a positive integer s, where

$$C_i = (\ell_{i,1} \vee \ell_{i,2} \vee \cdots \vee \ell_{i,m_i})$$

where m_i is a positive integer, for every integer i such that $1 \le i \le s$ — and where $\ell_{i,j}$ is a ***literal*** (either x_h or $\neg x_h$, for some non-negative integer h) for all i and j such that $1 \le i \le s$ and $1 \le j \le m_i$.

In some situations, we might be interested in *satisfying as many clauses of a Boolean formula $\mathcal{F}$, in conjunctive normal form, as possible*. This may suggest the following decision problem.

Satisfying Clauses

Instance: A Boolean formula $\mathcal{F}$, in conjunctive normal form, with s clauses for some positive integer s, and a non-negative integer k

Question: Are the following conditions both satisfied?

(a) There exists a partial truth assignment $\varphi : \mathcal{V} \to \{\mathtt{T}, \mathtt{F}\}$ such that $\varphi(C_i)$ is defined, for every integer i such that $1 \le i \le s$, and such that at least k of $\varphi(C_1), \varphi(C_2), \dots, \varphi(C_k)$ are T.

(b) There *does not* exist a partial truth assignment $\psi : \mathcal{V} \to \{\mathtt{T}, \mathtt{F}\}$ such that $\psi(C_i)$ is defined, for every integer i such that $1 \le i \le s$, and such that at least $k+1$ of $\psi(C_1), \psi(C_2), \dots, \psi(C_k)$ are T.

Recall that Boolean formulas can be encoded as strings over the alphabet Σ_F, introduced in Lecture #11, in a straightforward way. Since Σ_F includes each of the symbols $0, 1, 2, \ldots, 9$, a non-negative integer k can be encoded, as a string in $\Sigma_F^\star$, using its unpadded decimal representation

Now let $\widehat{\Sigma}_F = \Sigma_F \cup \{ (\,,\,,\,) \}$. An instance of the above decision problem, including a Boolean formula $\mathcal{F}$ and a non-negative integer k, can therefore be encoded as a string in $\widehat{\Sigma}_F^\star$ consisting of the encoding of $\mathcal{F}$, and the decimal representation of k, enclosed by brackets and separated by a comma.

Let $L_{\text{Formula+Number}} \subseteq \widehat{\Sigma}^\star$ be the language of encodings of *instances* of the "Satisfying Clauses" problem, given above, and let $L_{\text{ClausesSatisfied}} \subseteq L_{\text{Formula+Number}}$ be the language of encodings of Yes-instances of this decision problem.

Proof That $L_{\text{Formula+Number}} \in \mathcal{P}$:

Proof That $L_{\textsf{ClausesSatisfied}} \in \Sigma_2\mathcal{P}$:

Proof That $L_{\text{ClausesSatisfied}} \in \Pi_2\mathcal{P}$:

Proof That $\Pi_i\mathcal{P} =$ co-$\Sigma_i\mathcal{P}$ for Every Positive Integer i: