1 Notes on 23rd & 25th Mar.

1.1 Main Topic

In the previous class, we saw some complete sets for NC^1 and TC^0. Today, we will introduce brief proofs of completeness for standard classes.

Let C be our favorite class (with a corresponding type of machine), then the set $\{(i, x, 1^n) : M_i(x)$ accepts in resource n, where i is an indexing of Turing machines for $C\}$ is complete for C.

For example:

1. $C = NP$, $B = \{(i, x, 1^n) : M_i(x)$ accepts in time $n\}$ is complete for NP (under $\leq_m^{AC^0}$)
   (a) $B \in NP$
   (b) Let $A \in NTIME(n^k)$, s.t. $A = L(M_i)$
   $x \in A \Leftrightarrow (i, x, 1^{|x|^k}) \in B$
   (c) So, B is $NP \Leftrightarrow complete$
2. $C = NL$, $B = \{(i, x, 1^n) : M_i$ is an NTM with a binary alphabet, and M_i accepts x within space $\log n\}$ is complete for NL(under $\leq_m^{AC^0}$)
   (a) $B \in NL$
   (b) Let A be in NL,
   then there is some $NTM M_i$ accepting A within space $k \log n$ (for some k).
   Now x is in $A \Leftrightarrow (i, x, 1^{n^k}) \in B$.
   (c) B is $NL \Leftrightarrow complete$.

1.2 Easy proof that SAT is $NP - complete$

Let $A \in NP$
There is a set $B \in P$, s.t.
$x \in A \Leftrightarrow \exists y, |y| = |x|^k$ and $(x, y) \in B$

Recall that B has a $Dlogtime \Leftrightarrow uniform$ circuit family $\{C_n\}$:
$x \in A \Leftrightarrow \exists y$, C_{n+n^k} accepts on input x and y.

Thus, "Circuit-SAT" defined as below is $NP \Leftrightarrow complete$:
$\{C$: there is a setting to the inputs of C that makes C output 1$\}$.

Given a circuit C (without loss of generality, assume all its gates other than inputs and outputs are NOR gates), we can build a formula ϕ_c, s.t.
$C \in$ Circuit $SAT \Leftrightarrow \phi_c \in SAT$.

ϕ_c has variables for each NOR gate of C and clauses of the following form:
if gate h and k are gate g's two inputs, we'll have the clauses:
$(g \vee \overline{h} \vee \overline{k}) \wedge (\overline{g} \vee h \vee k) \wedge (\overline{g} \vee h \vee \overline{k}) \wedge (\overline{g} \vee \overline{h} \vee k)$.

1.3 PSPACE's complete set

A quantified Boolean formula is a Boolean formula with quantifiers:

$$\exists x \forall y \exists z[(x \vee \overline{y}) \wedge (y \vee z \vee x)] \Leftrightarrow \forall y \exists z[(1 \vee y) \wedge (y \vee z \vee 1)] \vee \forall y \exists z[(0 \vee \overline{y}) \wedge (y \vee z \vee 0)] \quad (1)$$

Fact 1 *: $TQBF = \{\phi : \phi$ is a QBF that is true$\}$ is compete for $PSPACE$.*

Proof:

Let $A \in PSAPCE$

Let $A = L(M)$ where M is a poly-time ATM that makes an alternation between existential and universal states in each step.

Clearly, $x \in L(M) \Leftrightarrow$
$\exists$ moves @ time 1 s.t.
$\forall$ moves @ time 2
$\exists \cdots$
$\cdots$
$\forall$ moves at time n^k
M accepts along this path (this predicate is in NP, so we can build a formula ϕ, s.t. $\phi \in SAT \Leftrightarrow$the predicate is true:
$\exists m_1 \forall m_2 \cdots \forall m_{n^k} \exists z_1 \exists z_2 \cdots \exists z_r \phi(m_1 m_2 \cdots m_{n^k}, z_1 z_2 \cdots z_r) \in TQBF \Leftrightarrow x \in A$

Fact 2 *Most natural problems are complete for some complexity class (under AC^0 reduc ibility).*

Theorem 1 (Ladner's thorem) *Assume $P \neq NP$, then there is a set A in $NP \Leftrightarrow P$, s.t. A is not $NP \Leftrightarrow$complete (under $\leq_T^P$).*

Proof:

Assume $P \neq NP$
for each finite set B, and for each poly-time machine M_i
$SAT \neq L(M_i^B)$, and $SAT \Leftrightarrow B \neq L(M_i)$

Thus the following function is recursive:
$r(i, n) = 1$
$r(2i, n) =$ the least m, s.t
for all $B \subseteq \Sigma^{\leq n}$
there is a $y, n < |y| < |y|^i \leq m$, s.t.
$M_i(y) = 1 \Leftrightarrow y \notin SAT \Leftrightarrow B$
$r(2i+1, n) =$ the least m s.t. for all $B \leq \Sigma^{\leq n}$
$\exists y$, $n < |y| < |y|^i \leq m$, s.t.
$M_i^B(y) = 1 \Leftrightarrow y \notin SAT$

Let $s(i)$ be time-constructible and $s(i) \geq r(i, s(i \Leftrightarrow 1))$
thus the set G is in P
where $G = \{x : \exists$ an even i, s.t. $s(i) \leq |x| \leq s(i+1)\}$

Define $A = SAT \cap G$
claim:
$A \in NP$ (trivial conclusion)
$A \notin P$
A not $NP \Leftrightarrow$*complete*

$\Rightarrow A \notin P$
Assume $A \in P, A = L(M_i)$
Let $B = A \cap \Sigma^{\leq s(2i-1)}$
Thus, $\exists y, s(2i \Leftrightarrow 1) \leq |y|^i \leq r(2i, s(2i \Leftrightarrow 1)) \leq s(2i)$
s.t. $M_i(y) = 1 \Leftrightarrow y \notin SAT \Leftrightarrow B$
$\Leftrightarrow y \notin (SAT \cap G)$(since $y \notin B$)
$\Leftrightarrow y \notin A$(since $y \notin G$)
There is a contradiction, so $A \notin P$.
$\Rightarrow A$ is not $NP \Leftrightarrow$*complete*
Assume $SAT = M_i^A$ (if A is $NP \Leftrightarrow$*complete*, this is true)
Let $B = A \cap \Sigma^{\leq s(2i)}$
There is a string y, s.t.
$s(2i) \leq |y|^i \leq r(2i+1, s(2i)) \leq s(2i+1)$ s.t.
$M_i^B = 1 \Leftrightarrow y \notin SAT$
Note that: M^i runs in n^i time,
M_i^B does not ask any queries about strings longer than $|y|^i \leq s(2i+1)$
Thus, $M_i^B(y) = M_i^A(y)$ (A has no strings in the gap)
Thus, $M_i^A(y) = 1 \Leftrightarrow y \in SAT$
There is a contradiction, so A is not $NP \Leftrightarrow$*complete*.

End of Proof of Ladner's theorem.

Note: How to compute $s(i)$

```
on input i
start a clock
if i = 1
  output one constant k
else
    compute s(i)
    compute r(i, s(i))
    output the current clock value
end
```

1.4 New Concept:

Factoring $= \{(x, i, b)$:the i^{th} bit of the prime factorization of x is $b\}$, where prime factorization of x is the string:
$x = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ where $p_1 < p_2 \cdots < p_k$ and $e_i > 0$.

Fact 3 *1. Factoring* $\in NP$

2. *Factoring* $\in UP = \{A \in NP : A = L(M_i)\ s.t. M_i(x)\ has \leq 1\ accepting\ path\ \}$

3. *Factoring* $\in CoNP$, *use the opposite (if b is correct, reject).*

Fact 4 *If Factoring is* $NP \Leftrightarrow$*complete, then* $NP = CoNP$.

Since it seems that $NP \neq CoNP$, we guess that Factoring is not in $NP \Leftrightarrow$ *complete*.

Question 1 *Is* $NP \subseteq P/Poly$? *(Here,* $P/Poly = \{A : A$ *has circuit family of poly s ize* $\}$*)*

If this is true, there is a circuit of poly size that can provide anwsers for input of size n.

Fact 5
$$\begin{aligned} NP \subseteq P/Poly &\Rightarrow NP^{NP} = CoNP^{NP} \\ &\Leftrightarrow \Sigma_2^P = \prod_2^P \end{aligned}$$

Fact 6 *Let n be given, most of the functions* $\{0,1\}^n \to \{0,1\}$ *require circuit of size* $\geq 2^{\sqrt{n}}$

Proof:

Note that a circuit of size s can be described by a string of size $\leq 15s \log s$, the number of functions having a small circuits $\leq$ the number of strings describing circuits $\leq 2^{15\sqrt{n}2^{\sqrt{n}}}$, which is $\ll 2^{2^n}$.