Recitation 8

John Chilton

August 1, 2007

MODEXP (1/2)

$$MODEXP = \{a, b, c, p \mid a^b \mod p = c \mod p\}$$

Show $MODEXP \in P$

MODEXP (2/2)

$$a^b \mod p = \begin{cases} 1 & b = 0 \\ ((a^{\frac{b}{2}}) \mod p)^2 \mod p & \text{b is even} \\ (a \mod p)(a^{b-1} \mod p) \mod p & \text{b is odd} \end{cases}$$

MODEXP (2/2)

$$a^b \mod p = \begin{cases} 1 & b = 0 \\ ((a^{\frac{b}{2}}) \mod p)^2 \mod p & \text{b is even} \\ (a \mod p)(a^{b-1} \mod p) \mod p & \text{b is odd} \end{cases}$$

```
(define (expmod a b p)
  (cond
    ((= b 0) 1)
    ((even?  b)
     (mod (square (expmod a (/ b 2) p)) p))
    (else
     (mod (* (mod a p) (expmod a (- b 1) p)) p))))
```

MODEXP (2/2)

$$a^b \mod p = \begin{cases} 1 & b = 0 \\ ((a^{\frac{b}{2}}) \mod p)^2 \mod p & \text{b is even} \\ (a \mod p)(a^{b-1} \mod p) \mod p & \text{b is odd} \end{cases}$$

```
(define (expmod a b p)
  (cond
    ((= b 0) 1)
    ((even? b)
     (mod (square (expmod a (/ b 2) p)) p))
    (else
     (mod (* (mod a p) (expmod a (- b 1) p)) p))))

(define (in-mod-exp? a b c p)
  (= (expmod a b p) (mod c p)))
```

Problem 7.14

B_i indicates whether $w_{1 \dots i}$ be decomposed so that $w_{1 \dots i} \in A^*$.

Problem 7.14

B_i indicates whether $w_{1...i}$ be decomposed so that $w_{1...i} \in A^*$.

i	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	...	w_{n-1}	w_n
B_i	0	1	1	0	1	0	?		...		

Problem 7.14

B_i indicates whether $w_{1...i}$ be decomposed so that $w_{1...i} \in A^*$.

i	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	$\dots$	w_{n-1}	w_n
B_i	0	1	1	0	1	0	?		$\dots$		

Define *computeB*(i)

If $w_1 \dots w_i \in A$, return 1

For k from 1 to $n-1$

If B_k and $w_{k+1} \dots w_i \in A$, return 1

return 0

B_i indicates whether $w_{1\ldots i}$ be decomposed so that $w_{1\ldots i} \in A^*$.

i	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	$\ldots$	w_{n-1}	w_n
B_i	0	1	1	0	1	0	?		$\ldots$		

If $w = \epsilon$, *accept*.
For i from 1 to n
    $B_i := computeB(i)$
Accept if $B_n = 1$, else *reject*.

7.16

Show *UNARYSSUM* is in P.

7.17

$A \neq \{\}$ and $A \neq \Sigma^*$. So for any such A there exists strings x, y such that $x \in A$ and $y \notin A$.

7.17

$A \neq \{\}$ and $A \neq \Sigma^*$. So for any such A there exists strings x, y such that $x \in A$ and $y \notin A$.

$$S = \begin{array}{l} \text{On input } < \phi > \\ \text{1. Determine if } \phi \in SAT \\ \text{2. If yes, output } x. \text{ If no, output } y. \end{array}$$

7.17

$A \neq \{\}$ and $A \neq \Sigma^*$. So for any such A there exists strings x, y such that $x \in A$ and $y \notin A$.

$S =$ On input $< \phi >$
1. Determine if $\phi \in SAT$
2. If yes, output x. If no, output y.

Hence $SAT \leq_p A$, and $A \in P \subseteq NP$, hence A is NP-complete.

7.21

$$
\begin{aligned}
S = \quad & \text{On input } < \phi > \\
& \text{1. Let } y \text{ be variable not in } \phi \\
& \text{2. Output } \phi \wedge (y \vee \overline{y})
\end{aligned}
$$

Argue that it works.

Hence $SAT \leq_p DOUBLESAT$.

7.23 - 2-CNF

Show 2CNF is in P.

7.23 - 3-CNF

$$(x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee x_1 \vee \overline{x2}) \wedge (x_1 \vee \overline{x_2} \vee x_3)$$

7.23 - 3-CNF

$$(x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee x_1 \vee \overline{x2}) \wedge (x_1 \vee \overline{x_2} \vee x_3)$$

Replace each x_1 with another new variable.

7.23 - 3-CNF

$$(x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee x_1 \vee \overline{x2}) \wedge (x_1 \vee \overline{x_2} \vee x_3)$$

Replace each x_1 with another new variable.

$$(x_{11} \vee x_{12} \vee x_2) \wedge (\overline{x_{13}} \vee x_{14} \vee \overline{x_2}) \wedge (x_{15} \vee \overline{x_2} \vee x_3)$$

7.23 - 3-CNF

$$(x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee x_1 \vee \overline{x2}) \wedge (x_1 \vee \overline{x_2} \vee x_3)$$

Replace each x_1 with another new variable.

$$(x_{11} \vee x_{12} \vee x_2) \wedge (\overline{x_{13}} \vee x_{14} \vee \overline{x_2}) \wedge (x_{15} \vee \overline{x_2} \vee x_3)$$

Need to ensure $x_{11}, \ldots, x_{15}$ have same truth value.

7.23 - 3-CNF

$$(x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee x_1 \vee \overline{x2}) \wedge (x_1 \vee \overline{x_2} \vee x_3)$$

Replace each x_1 with another new variable.

$$(x_{11} \vee x_{12} \vee x_2) \wedge (\overline{x_{13}} \vee x_{14} \vee \overline{x_2}) \wedge (x_{15} \vee \overline{x_2} \vee x_3)$$

Need to ensure $x_{11}, \ldots, x_{15}$ have same truth value.

$$(x_{11} \rightarrow x_{12}) \wedge (x_{12} \rightarrow x_{13}) \wedge (x_{13} \rightarrow x_{14}) \wedge (x_{14} \rightarrow x_{15}) \wedge (x_{15} \rightarrow x_{11})$$

7.23 - 3-CNF

$$(x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee x_1 \vee \overline{x2}) \wedge (x_1 \vee \overline{x_2} \vee x_3)$$

Replace each x_1 with another new variable.

$$(x_{11} \vee x_{12} \vee x_2) \wedge (\overline{x_{13}} \vee x_{14} \vee \overline{x_2}) \wedge (x_{15} \vee \overline{x_2} \vee x_3)$$

Need to ensure $x_{11}, \ldots, x_{15}$ have same truth value.

$$(x_{11} \rightarrow x_{12}) \wedge (x_{12} \rightarrow x_{13}) \wedge (x_{13} \rightarrow x_{14}) \wedge (x_{14} \rightarrow x_{15}) \wedge (x_{15} \rightarrow x_{11})$$

Which is equivelent to:

7.23 - 3-CNF

$$(x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee x_1 \vee \overline{x2}) \wedge (x_1 \vee \overline{x_2} \vee x_3)$$

Replace each x_1 with another new variable.

$$(x_{11} \vee x_{12} \vee x_2) \wedge (\overline{x_{13}} \vee x_{14} \vee \overline{x_2}) \wedge (x_{15} \vee \overline{x_2} \vee x_3)$$

Need to ensure $x_{11}, \ldots, x_{15}$ have same truth value.

$$(x_{11} \rightarrow x_{12}) \wedge (x_{12} \rightarrow x_{13}) \wedge (x_{13} \rightarrow x_{14}) \wedge (x_{14} \rightarrow x_{15}) \wedge (x_{15} \rightarrow x_{11})$$

Which is equivelent to:

$$(\overline{x_{11}} \vee x_{12}) \wedge (\overline{x_{12}} \vee x_{13}) \wedge (\overline{x_{13}} \vee x_{14}) \wedge (\overline{x_{14}} \vee x_{15}) \wedge (\overline{x_{15}} \vee x_{11})$$

7.24

$$(y_1 \vee y_2 \vee y_3)$$

becomes

$$(y_1 \vee y_2 \vee z_i) \wedge (\overline{z_i} \vee y_3 \vee b)$$

Argue the original formula is satisfiable iff reduced formula is $\neq$ satisfiable.

7.27

Show 3$COLOR$ is NP-complete.

7.36

If $P = NP$ show how to find a satisfying assignment for ϕ in polynomial time.

7.36

If $P = NP$ show how to find a satisfying assignment for ϕ in polynomial time.

BIG IDEA: If ϕ is satisfiable and $\phi \wedge x$ is satisfiable, then x can be true in a satisfying assignment.

If $P = NP$ show how to find a satisfying assignment for ϕ in polynomial time.

BIG IDEA: If ϕ is satisfiable and $\phi \wedge x$ is satisfiable, then x can be true in a satisfying assignment.

$S =$ On input $< \phi >$
1. If ϕ is not satisfiable, *reject*.
2. For each x_i in ϕ:
3. Determine x_i can be true in a satisifing assignment.
4. If yes, $\phi := \phi \wedge x_i$ and set x_i as true in assignment
5. If no, $\phi := \phi \wedge \overline{x_i}$ and set x_i as false in assignment