طراحی و تحلیل الگوریتم ها

دکتر امیر لکی زاده

استادیار گروه مهندسی کامپیوتر دانشگاه قم

فصل 34 :

مسائل
- tractable (آسان و قابل حل در عمل) : الگوریتمی وجود دارد که آن را در زمان چند جمله ای حل می کند.
- intractable (سخت و غیر قابل حل در عمل) :

چرا اگر برای یک مسئله یک الگوریتم چند جمله ای از $O(n^{100})$ ارائه شود همچنان آن مسئله tractable نامیده می شود؟

۱- تعداد بسیار کمی مسئله دارای الگوریتمی با مرتبه $\Theta(n^{100})$ می باشد.

۲- در بیشتر مواقع وقتی یک الگوریتم چند جمله ای ارائه می شود بطور متوالی الگوریتم های بهتری ارائه می شود بطوریکه ممکن است به یکباره از $O(n^{100})$ کاهش قابل توجهی بوجود آید.

قابل تغییر

مسائل بهینه سازی: Optimization Problems : برای یک مسئله راه حل های ممکن زیادی وجود دارد متناظر با هر راه حل یک مقدار پست می آید و هدف پیدا کردن راه حل با بهینه ترین مقدار می باشد.

مسائل تصمیم گیری: Decision Problems : مسائل که برای آنها تنها جواب yes یا no وجود دارد.

الگوریتم حل سعی می کند که برقراری یا عدم برقراری شرایط خاص را بررسی کند.

مسائل بهینه سازی در مقابل مسائل تصمیم گیری :

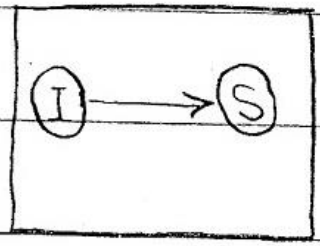
۱- هر مسأله بهینه سازی را می توان به یک مسأله تصمیم گیری تبدیل کرد (با گذاشتن حد بروی مقداری که باید بهینه شود)

مثال: کوتاهترین مسیر میان دو رأس u و v در گراف وزن دار G یک مسأله بهینه سازی می باشد.

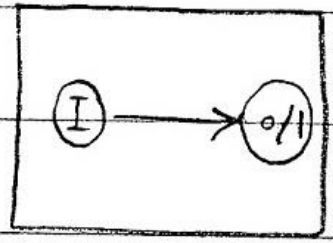
بررسی وجود یا عدم وجود مسیری با وزن کمتر K میان u و v در گراف G مسأله تصمیم گیری معادل می باشد

۲- هر مسأله تصمیم گیری از مسئله بهینه سازی معادل سخت تر نمی باشد (اگر بتوانیم مسأله بهینه سازی را حل کنیم می توان مسأله تصمیم گیری معادل را نیز حل کرد)

یک رابطه انتزاعی میان نمونه‌ی مسأله (Problem instances) و مجموعه جواب های مسأله (Problem solutions)



AStract optimization Problem



Astract Decision problem

تبدیل نمونه‌ی مسأله به کدی دودویی:

نمونه مسأله

$e(i) \longrightarrow 0/1$

مسأله تصمیم گیری

$Q(\langle i \rangle) \longrightarrow 0/1$

مسأله تصمیم گیری که ورودی‌ی آن کد شده است،

$$L = \{x \in \{0,1\}^* \mid Q(x) = 1\}$$

L: زبان شامل نمونه‌هایی که به ازای آنها الگوریتم برای مسأله تصمیم گیری جواب Yes (1) برمی‌گرداند.

زبان پذیرفته شده توسط الگوریتم A (language is accepted by A):

$$L = \{x \in \{0,1\}^* \mid A(x) = 1\}$$

زبان پذیرفته شده توسط الگوریتم A در زمان چند جمله‌ای:

زبان L را قابل پذیرش در زمان چندجمله‌ای توسط الگوریتم A می‌گوییم هرگاه یک ثابت مثل $k > 0$ وجود داشته باشد بطوریکه به ازای هر $x \in L$، الگوریتم A، x را در زمان $O(|x|^k)$ پذیرش کند. $x \in L, x \in \bar{L}$

L is accepted in polynomial time by A

زبان تصمیم پذیر توسط الگوریتم A در زمان چندجمله‌ای: L is deciable in polynomial time by A

زبان L را تصمیم پذیر در زمان چندجمله‌ای بوسیله الگوریتم A می‌گوییم هرگاه یک ثابت مثل $k > 0$ وجود داشته باشد بطوریکه به ازای هر $x \in \{0,1\}^*$، الگوریتم A، x را در زمان $O(|x|^k)$ تصمیم گیری کند. $x \in L, x \in \bar{L}$

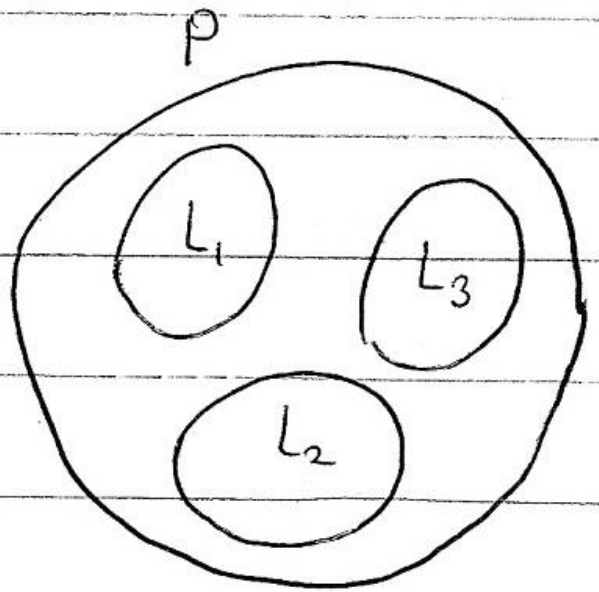
مسائل تصمیم پذیر در زمان چند جمله ای (مجموعه زبان های تصمیم پذیر در زمان چندجمله ای):

Polynoimial Decidable Problems (P)

$$P = \{ L \subseteq \{0,1\}^* \mid L \text{ is decidable in Polynomial time by an Alg A} \}$$

L_1: مسأله مرتب سازی

L_2: مسأله جستجو



الگوریتم تشخیص (verification Alg) با دو پارامتر x و y:

زبان قابل تشخیص در زمان چند جمله‌ای: L is verified by verification Alg A وجود داشته باشد $y \in \{0,1\}^*$

می‌گوییم زبان L توسط الگوریتم تشخیص A، در زمان چند جمله‌ای تشخیص داده می‌شود هرگاه $\forall x \in \{0,1\}^*$

که الگوریتم A در زمان $O(|x|^k)$ (k ثابت) بتواند $A(x,y) = 1$

x : کدینگ گراف هامیلتونی

y : کدینگ دور (کدینگ یک جایگشت از رئوس)

$$HAM\_CYCLE = \{ \langle G \rangle \mid G \text{ is a Hamilton Graph} \}$$

(NP class) مجموعه مسائل قابل تشخیص در زمان چندجمله‌ای (مجموعه زبان‌های قابل تشخیص در زمان چندجمله‌ای)

$L \in NP \iff L = \{ x \in \{0,1\}^* \mid$ there exist a polynomial verification ALG A and there exist a certificate y ; $|y| = O(|x|^c)$ such that $A(x,y)=1 \}$

مسأله ↑ — یک رشته خاص

L is polynomial time verifiable by A

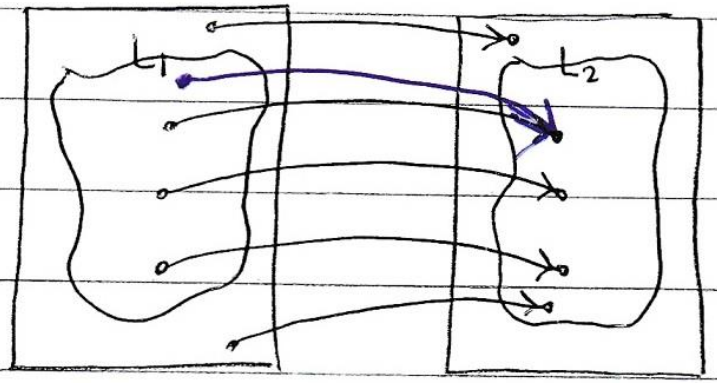
HAM-CYCLE $\in$ NP

$L \in P \Rightarrow L \in NP \Rightarrow P \subseteq NP$

Reduction تبدیلات:

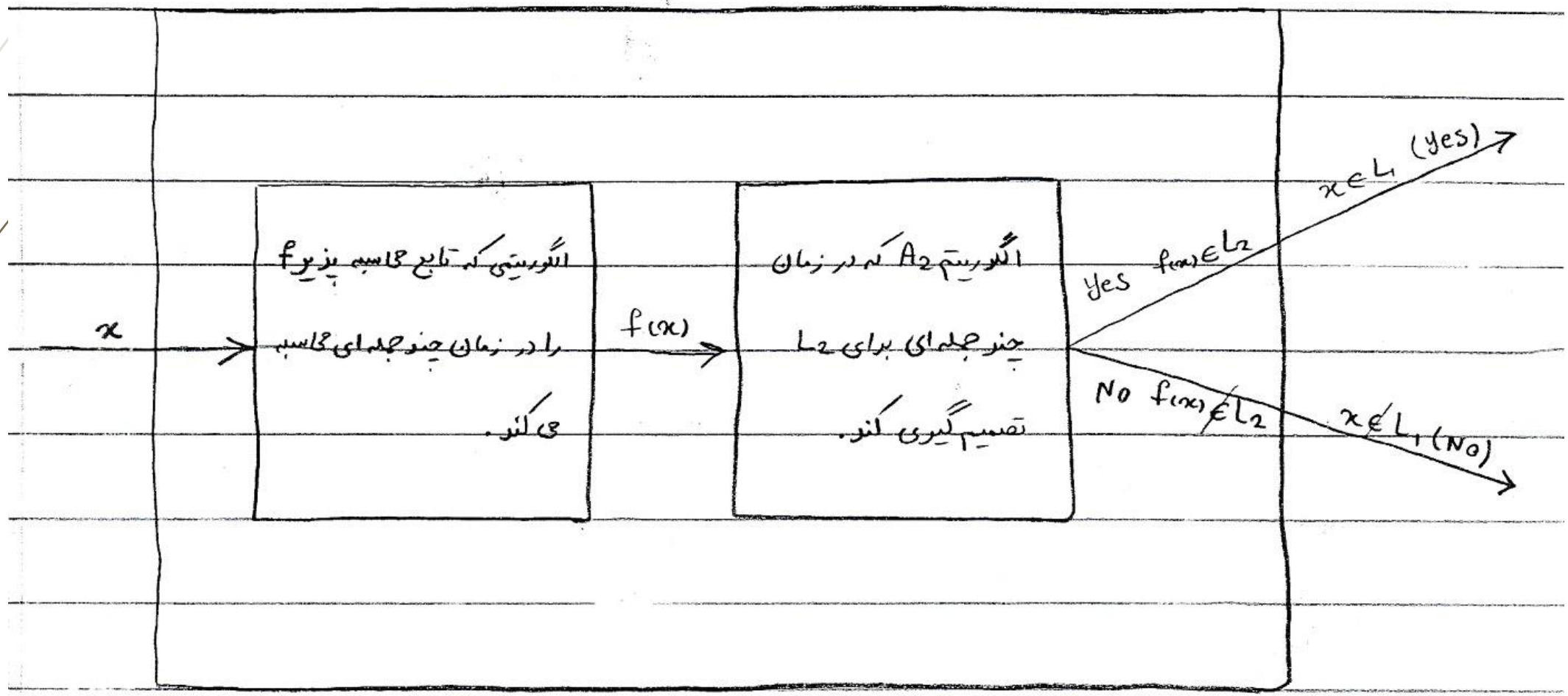
می‌گوییم زبان L_1 در زمان چندجمله‌ای قابل تبدیل (قابل کاهش Reducable) به زبان L_2 می‌باشد و می‌نویسیم $L_1 \leq_P L_2$ هرگاه یک تابع حاسب‌پذیر در زمان چندجمله‌ای مثل f وجود داشته باشد بطوریکه $(f : (0,1)^* \longrightarrow (0,1)^*)$

$\forall x \in (0,1)^* \quad x \in L_1 \iff f(x_1) \in L_2$



قضیه:

$L_1, L_2 \in \{0,1\}^*$, $L_1 \leq_P L_2$: $L_2 \in P \Longrightarrow L_1 \in P$



NP_complete (NP_C)

$L \in NP\_C \iff$ iff
1. $L \in NP$
2. $\forall L' \in NP \quad L' \leq_P L$

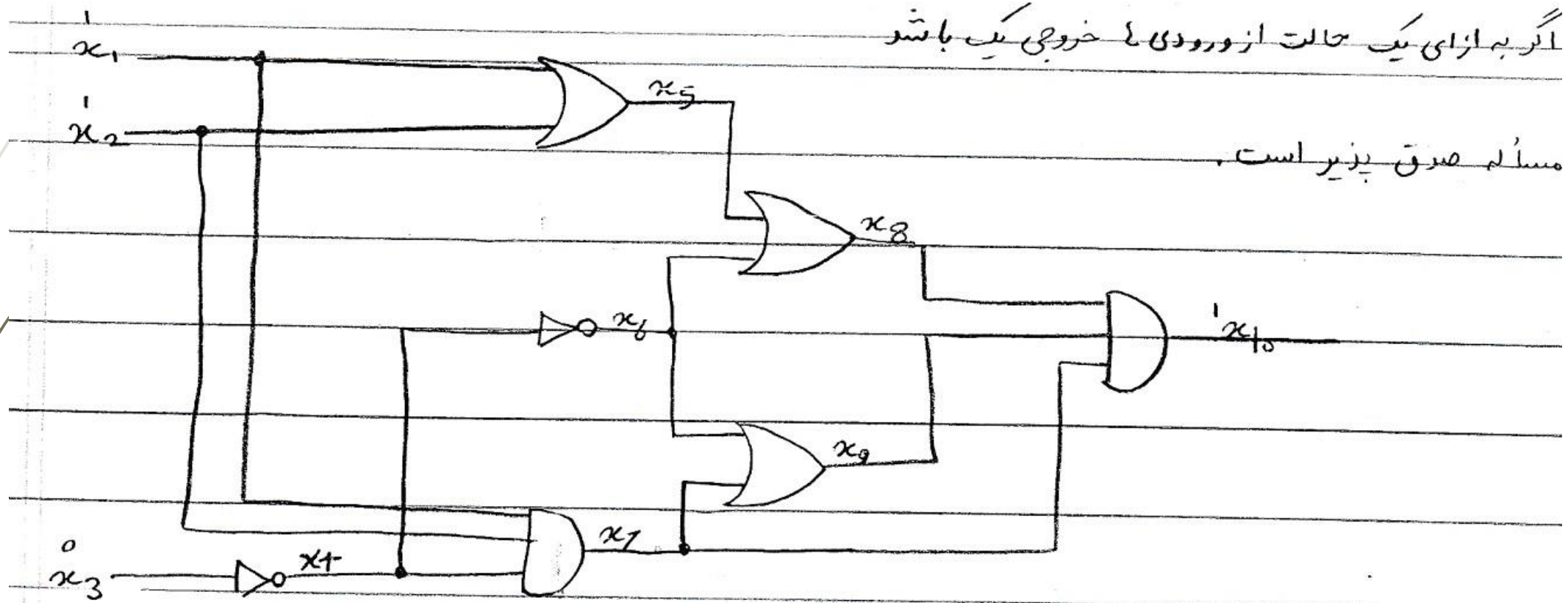
$L \in NP\_hard \iff$ iff $\forall L' \in NP \quad L' \leq_P L$

$L \in NP\_C \iff$ iff
1. $L \in NP$
2. $\exists L' \in NP\_C \quad L' \leq_P L$

اولین مسأله NP_C : مسأله صدق پذیری مداری الکتریکی : circuit satisfiability problem

اگر به ازای یک حالت از ورودی ها خروجی یک باشد

مسأله صدق پذیر است.



$$CIRCUIT = \{ \langle c \rangle \mid c \text{ is a satisfiable combinational cricuit} \}$$

مجموعه کلیه مدارات صدق پذیر در کلاس NP-C قرار دارند.

$CIRCUIT \in NP\_C$

$CIRCUIT \in NP\_C$ ⟸ 1. $CIRCUIT \in NP$ بدیهی

2. $\forall L' \in NP \quad L' \leq_P CIRCUIT$

Satisfiability Problem (SAT) : صدق پذیری یک فرمول منطقی ، NP-C مسأله دومین

$$SAT \{ \langle \Phi \rangle \mid \Phi \text{ is a satisfiable formula} \}$$

$$\Phi = ((x_1 \rightarrow x_2) \vee ((\sim x_1 \leftrightarrow x_3) \vee x_4)) \wedge \sim x_2 \quad CNF$$

$SAT \in NP\_C$ ⟸ $SAT \in NP$ بدیهی CNF $(\vee\vee)\wedge(\vee\vee)$ کافی است یکی از $(\vee\vee)$ صفر شود

$CIRCUIT \leq_P SAT$ (خروجی هر گیت را یک متغیر درنظر می‌گیریم) شکل بالای صفحه

$$\Phi = x_{10} \wedge (x_3 \leftrightarrow \sim x_4) \wedge (x_5 \leftrightarrow x_1 \vee x_2) \wedge (x_6 \leftrightarrow \sim x_4) \wedge (x_7 \leftrightarrow x_1 \wedge x_2 \wedge x_4)$$
$$\wedge (x_8 \leftrightarrow x_5 \vee x_6) \wedge (x_9 \leftrightarrow x_6 \vee x_7) \wedge (x_{10} \leftrightarrow x_7 \wedge x_8 \wedge x_9)$$

اندازه Φ چند جمله‌ای است چون اگر در بدترین حالت اندازه یک clause به اندازه کل متغیرها باشد باز Φ در مجموع در زمان چند جمله ای انجام می‌شود.

سومین مسأله NP-C : فرم نرمال عطفی Conjunctive Normal Form (CNF)

$$\phi = \bigwedge_{i=1}^{k} C_i \quad , \quad C_i = \bigvee_{j=i}^{m_i} L_j$$

clause ↗ ، ↘ literal

$$\phi = \underbrace{(x_1)}_{C_1} \wedge \underbrace{(x_1 \vee \sim x_2)}_{C_2} \wedge \underbrace{(\sim x_1 \vee x_2 \vee \sim x_3)}_{C_3}$$

3 CNF

$$\phi = \bigwedge_{i=1}^{k} C_i \quad , \quad C_i = \bigvee_{j=i}^{3} L_j$$

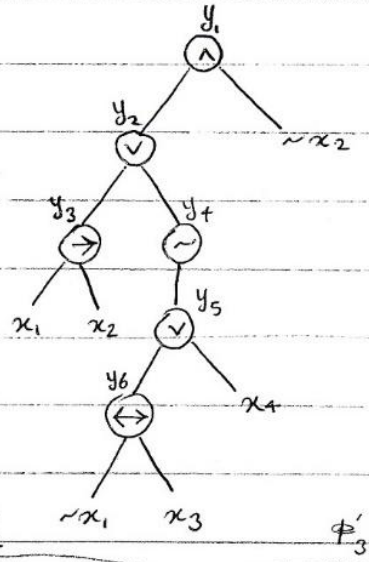
3_CNF

$$\Phi = \bigwedge_{i=1}^{k} C_i \quad , C_i = \bigvee_{j=i}^{3} L_j$$

$3\text{-CNF} \in NP\text{-C}$ {
1) $3\text{-CNF} \in NP$ ؛ clause تعدادی or که در بدترین حالت باید کل
بررسی کرده و هنگی یک باشند
2) $SAT \leq_P 3\text{-CNP}$

ایده: تبدیل فرمول SAT به یک درخت دودویی و سپس نظیر کردن یک متغیر به خروجی هر گره داخل
(عملگرها در گره‌های میانی و لیترال‌ها در برگ‌های درخت قرار می‌گیرند).

$$\Phi = \Big((x_1 \rightarrow x_2) \vee \big((\sim x_1 \leftrightarrow x_3) \vee x_4\big)\Big) \wedge \sim x_2$$



Φ'_1 Φ'_2 Φ'_3

$$\Phi' = y_1 \wedge (y_1 \leftrightarrow y_2 \wedge \sim x_2) \wedge (y_2 \leftrightarrow y_3 \vee y_4) \wedge (y_3 \leftrightarrow x_1 \rightarrow x_2)$$
$$\wedge (y_4 \leftrightarrow \sim y_5) \wedge (y_5 \leftrightarrow y_6 \vee x_4) \wedge (y_6 \leftrightarrow \sim x_1 \leftrightarrow x_3)$$

$$\Phi' = \Phi'_1 \wedge \Phi'_2 \wedge \Phi'_3 \wedge \dots$$

در ادامه برای هر Φ'_i جدول درستی را رسم می‌کنیم (Φ'_2)

y_1	y_2	x_2	$y_1 \leftrightarrow y_2 \wedge \sim x_2$
1	1	1	0
1	1	0	1
1	0	1	0
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	1
0	0	0	1

در ادامه $\sim \Phi'_i$ را بصورت DNF می‌نویسیم:

$$\sim \Phi'_2 = (y_1 \wedge y_2 \wedge x_2) \vee (y_1 \wedge \sim y_2 \wedge x_2) \vee (y_1 \wedge \sim y_2 \wedge \sim x_2) \vee (\sim y_1 \wedge y_2 \wedge \sim x_2)$$

در ادامه قرار می‌دهیم

$$\Phi''_i = \sim(\sim \Phi'_i) = \Phi'_i$$

حال Φ''_i به فرم CNF می‌باشد.

$$\Phi' = \Phi'_1 \wedge \Phi'_2 \wedge \Phi'_3 \wedge \dots$$

$$\Phi'' = \Phi''_1 \wedge \Phi''_2 \wedge \Phi''_3$$

(CNF) (CNF) (CNF) (CNF)

تمرین

- Click Decision Problem(CDP)
- Node Cover Decision Problem(NCDP)
- …

موفق باشید و سربلند

طراحی و تحلیل الگوریتم ها

- دکتر امیر لکی زاده
- استادیار گروه مهندسی کامپیوتر دانشگاه قم