بسم الله الرحمن الرحیم

برنامه‌ریزی نیمه‌معین برای طراحی الگوریتم‌های تقریبی

جلسه بیستم: CSP

Project: People

مسئله SAT و چرا مهم است؟

مسئله صدق‌پذیری (SAT)

- ورودی
  - تعدادی متغیر بولی (صحیح/غلط)
  - یک فرمول به صورت
  - «و»ی یک سری «یا»ی متغیر یا نقیض متغیر
- خروجی
  - آیا می‌توان یک مقداردهی به متغیرها پیدا کرد که کل عبارت صحیح شود؟

$$(a \vee \neg b \vee c) \wedge (\neg a \vee c \vee d \vee \neg e) \wedge (b \vee \neg d \vee e)$$

مسئله صدق‌پذیری (SAT)

- ورودی
  - تعدادی متغیر بولی (صحیح/غلط)
  - یک فرمول به صورت
  - «و»ی یک سری «یا»ی متغیر یا نقیض متغیر
- خروجی
  - آیا می‌توان یک مقداردهی به متغیرها پیدا کرد که کل عبارت صحیح شود؟

تعاریف

- متغیر
- لیترال: متغیر یا نقیض متغیر
- بند: یای چند لیترال
- عبارت: «و»ی چند بند

$$(a \vee \neg b \vee c) \wedge (\neg a \vee c \vee d \vee \neg e) \wedge (b \vee \neg d \vee e)$$

- مسئله m-SAT:
  - بندها شامل m لیترال

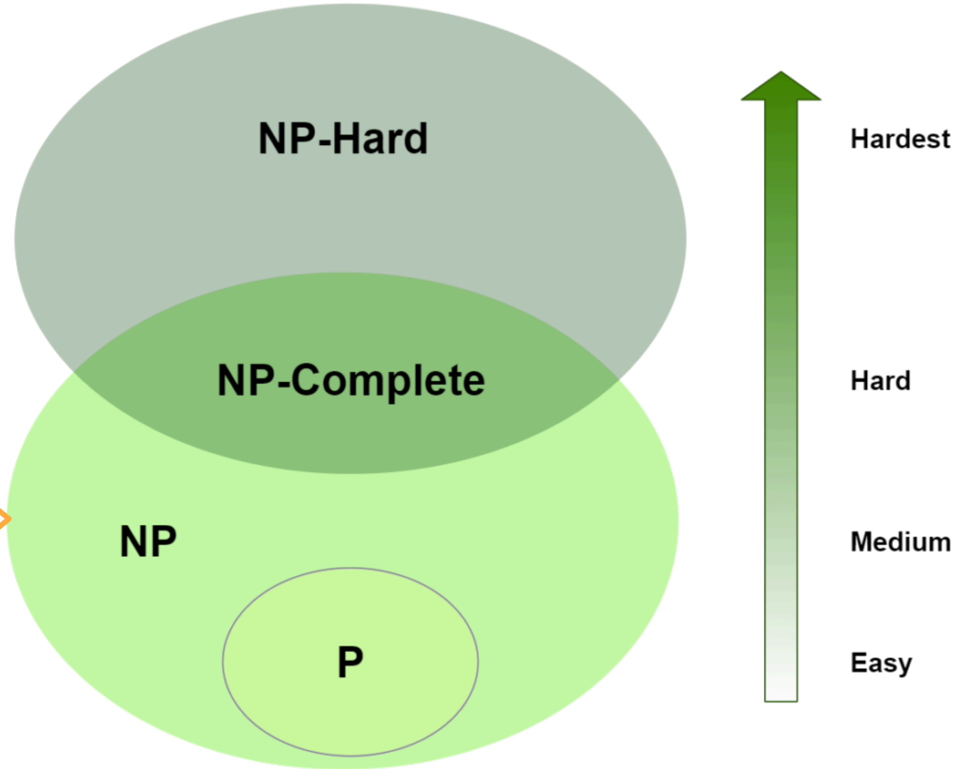
- مسئله m-SAT:
  - بندها شامل m لیترال

مثالی از یک عبارت ۳-SAT:

$$(x_1 \vee x_2 \vee \overline{x}_3) \wedge (x_1 \vee x_3 \vee x_5) \wedge (\overline{x}_2 \vee \overline{x}_4 \vee x_5) \wedge (x_2 \vee \overline{x}_3 \vee x_5)$$

مسئله SAT:

مسئله‌هایی که می‌توان با **ماشین تورینگ غیرقطعی** در زمان **چندجمله‌ای** حل کرد



ماشین تورینگ

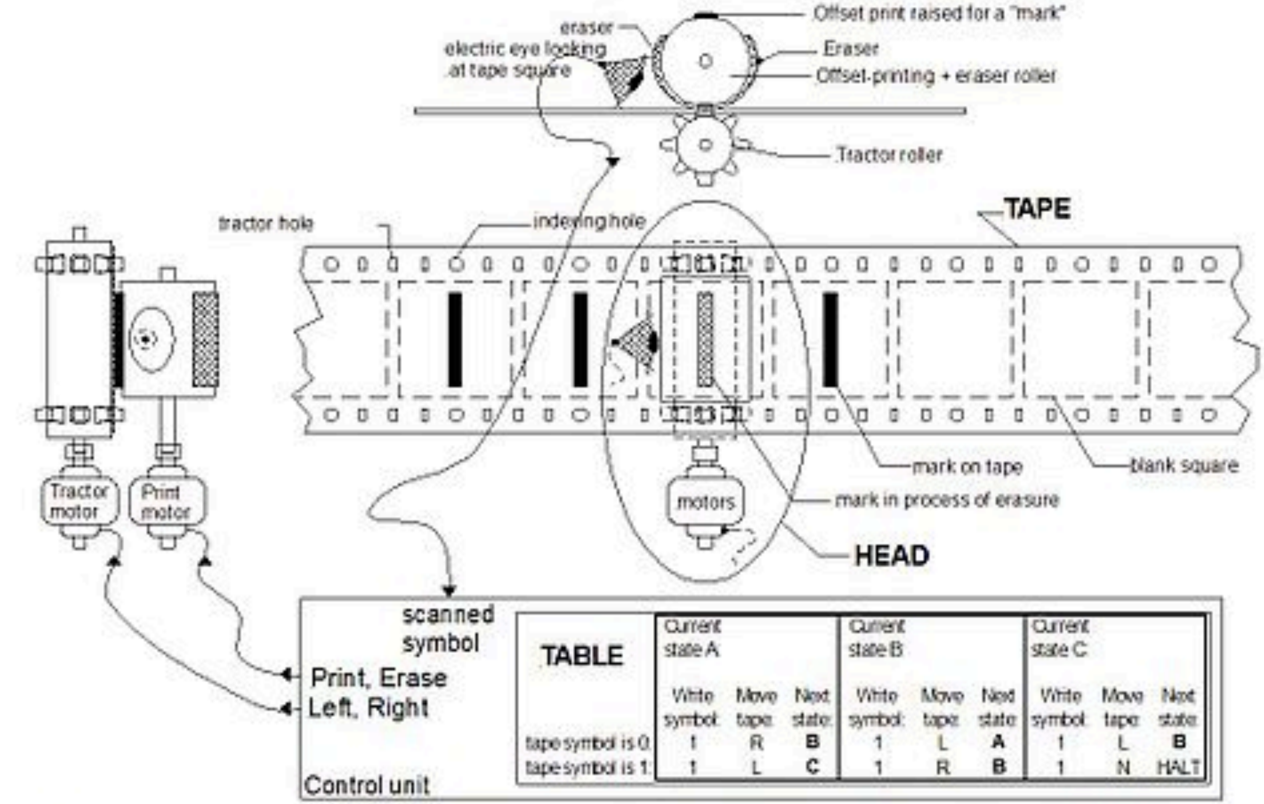
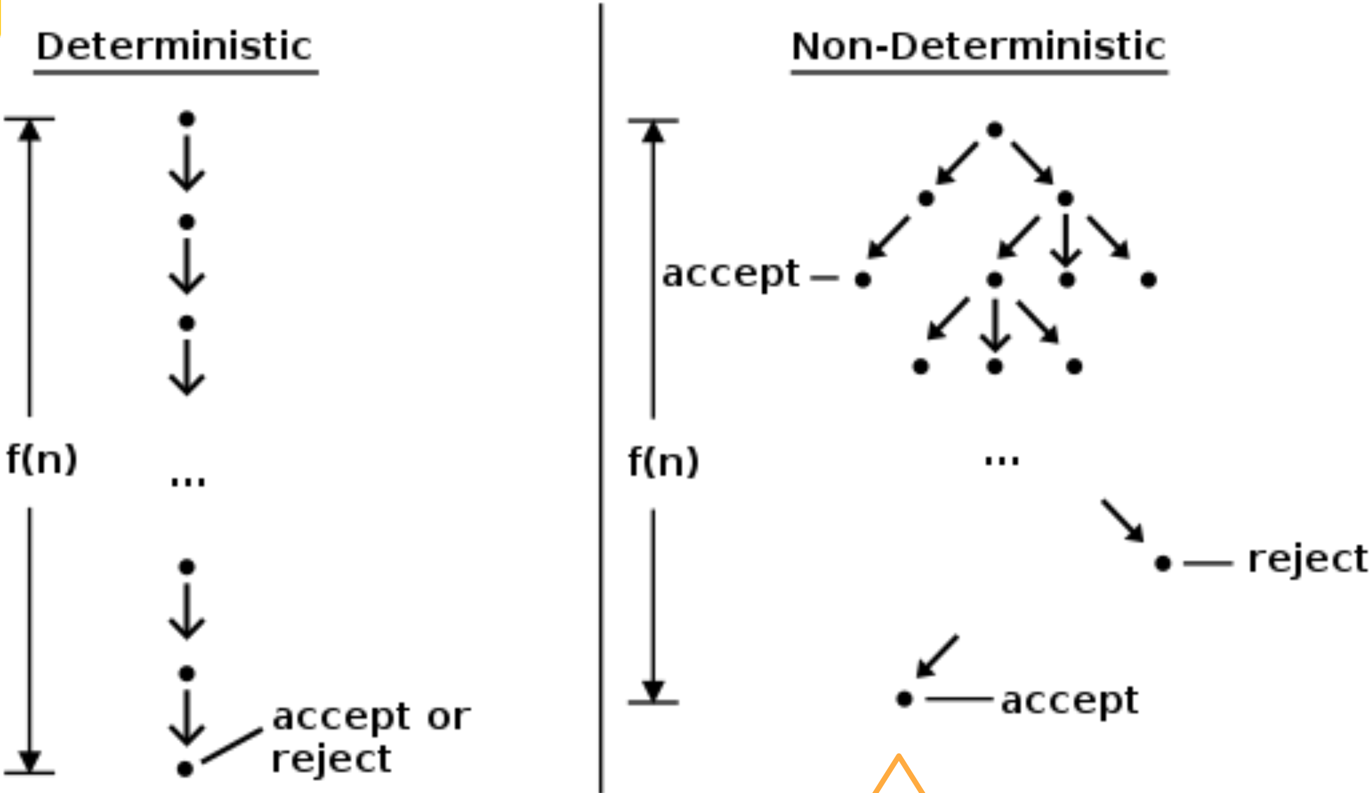


TABLE	Current state A			Current state B			Current state C		
	Write symbol	Move tape	Next state	Write symbol	Move tape	Next state	Write symbol	Move tape	Next state
tape symbol is 0	1	R	**B**	1	L	**A**	1	L	**B**
tape symbol is 1	1	L	**C**	1	R	**B**	1	N	HALT

A fanciful mechanical Turing machine's TAPE and HEAD. The TABLE instructions might be on another "read only" tape, or perhaps on punch-cards. Usually a "finite state machine" is the model for the TABLE.

ماشین تورینگ غیرقطعی



- پذیرش = وقتی تصمیم‌هایی باشد که به yes برسد

قضیه: مسئله SAT در رده NP است.

قضیه: مسئله SAT در رده NP است.

- اثبات: یک ماشین تورینگ غیر قطعی

قضیه: مسئله SAT در رده NP است.

- اثبات: یک ماشین تورینگ غیر قطعی
  - ۱) به صورت غیر قطعی متغیرها را مقداردهی می‌کند

قضیه: مسئله SAT در رده NP است.

- اثبات: یک ماشین تورینگ غیر قطعی
  - ۱) به صورت غیر قطعی متغیرها را مقداردهی می‌کند
  - ۲) چک می‌کند که آیا عبارت صحیح می‌شود؟ اگر شد پاسخ «بله» برمی‌گرداند

قضیه: مسئله SAT در رده NP است.

- اثبات: یک ماشین تورینگ غیر قطعی
  - ۱) به صورت غیر قطعی متغیرها را مقداردهی می‌کند
  - ۲) چک می‌کند که آیا عبارت صحیح می‌شود؟ اگر شد پاسخ «بله» برمی‌گرداند

- اگر صدق‌پذیر باشد => ماشین می‌پذیرد

قضیه: مسئله SAT در رده NP است.

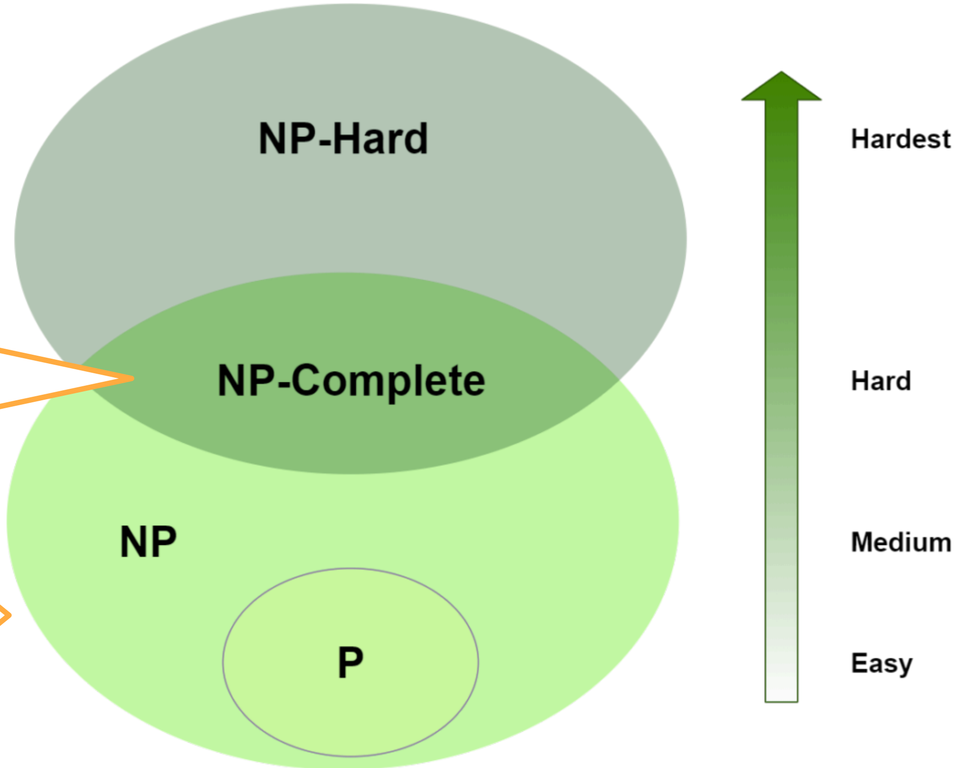
- اثبات: یک ماشین تورینگ غیر قطعی
  - ۱) به صورت غیر قطعی متغیرها را مقداردهی می‌کند
  - ۲) چک می‌کند که آیا عبارت صحیح می‌شود؟ اگر شد پاسخ «بله» برمی‌گرداند

- اگر صدق‌پذیر باشد => ماشین می‌پذیرد
  - و اگر نباشد، ماشین نمی‌پذیرد

مسئله SAT:

مسئله‌ای هست که اگر راه حل چندجمله‌ای داشته باشد، همه مسئله‌های NP راه حل چندجمله‌ای دارند.

مسئله‌هایی که می‌توان با **ماشین تورینگ غیرقطعی** در زمان **چندجمله‌ای** حل کرد

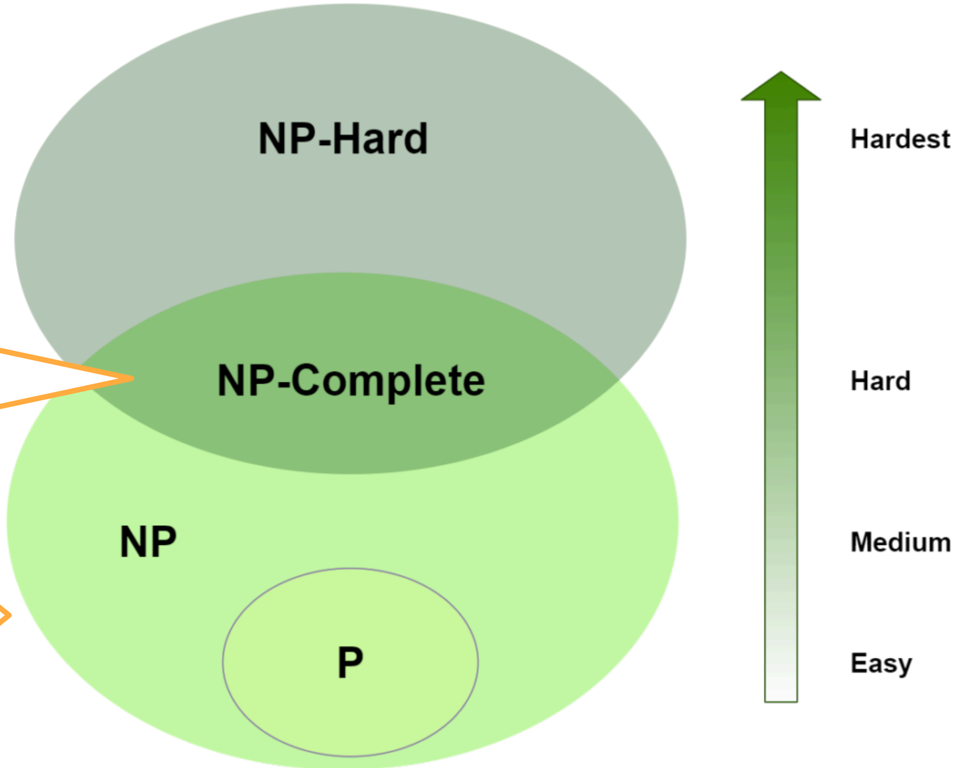


مسئله SAT:

مسئله SAT:

مسئله‌ای هست که اگر راه حل چندجمله‌ای داشته باشد، همه مسئله‌های NP راه حل چندجمله‌ای دارند.

مسئله‌هایی که می‌توان با **ماشین تورینگ غیرقطعی** در زمان **چندجمله‌ای** حل کرد



قضیه Cook-Levin: مسئله SAT، در رده NP ـ تمام است.

قضیه Cook-Levin: مسئله SAT، در رده NP_ تمام است.

- قضیه == اگر الگوریتمی چندجمله‌ای برای SAT داشته باشیم، هر مسئله NP را می‌توان در زمان چندجمله‌ای حل کرد.

قضیه Cook-Levin: مسئله SAT، در رده NP_تمام است.

- قضیه == اگر الگوریتمی چندجمله‌ای برای SAT داشته باشیم، هر مسئله NP را می‌توان در زمان چندجمله‌ای حل کرد.

- ایده اثبات:

قضیه Cook-Levin: مسئله SAT، در رده NP_ تمام است.

- قضیه == اگر الگوریتمی چندجمله‌ای برای SAT داشته باشیم، هر مسئله NP را می‌توان در زمان چندجمله‌ای حل کرد.

- ایده اثبات:
  - A: الگوریتم مسئله SAT

قضیه Cook-Levin: مسئله SAT، در رده NP_تمام است.

- قضیه == اگر الگوریتمی چندجمله‌ای برای SAT داشته باشیم، هر مسئله NP را می‌توان در زمان چندجمله‌ای حل کرد.

- ایده اثبات:
  - A: الگوریتم مسئله SAT
  - یک مسئله NP، با ماشین غیرقطعی T با زمان p

قضیه Cook-Levin: مسئله SAT، در رده NP_تمام است.

- قضیه == اگر الگوریتمی چندجمله‌ای برای SAT داشته باشیم، هر مسئله NP را می‌توان در زمان چندجمله‌ای حل کرد.

- ایده اثبات:
  - A: الگوریتم مسئله SAT
  - یک مسئله NP، با ماشین غیرقطعی T با زمان p
  - نمایش حالت ماشین با متغیرهای ۰ و ۱ (حداکثر O(p) متغیر)

قضیه Cook-Levin: مسئله SAT، در رده NP_تمام است.

- قضیه == اگر الگوریتمی چندجمله‌ای برای SAT داشته باشیم، هر مسئله NP را می‌توان در زمان چندجمله‌ای حل کرد.

- ایده اثبات:
  - A: الگوریتم مسئله SAT
  - یک مسئله NP، با ماشین غیرقطعی T با زمان p
  - نمایش حالت ماشین با متغیرهای ۰ و ۱ (حداکثر O(p) متغیر)
  - نمایش همه حالت‌های ماشین در همه مراحل با متغیرهای ۰ و ۱ (حداکثر $O(p^۲)$ متغیر)

قضیه Cook-Levin: مسئله SAT، در رده NP_تمام است.

- قضیه == اگر الگوریتمی چندجمله‌ای برای SAT داشته باشیم، هر مسئله NP را می‌توان در زمان چندجمله‌ای حل کرد.

- ایده اثبات:
  - A: الگوریتم مسئله SAT
  - یک مسئله NP، با ماشین غیرقطعی T با زمان p
  - نمایش حالت ماشین با متغیرهای ۰ و ۱ (حداکثر O(p) متغیر)
  - نمایش همه حالت‌های ماشین در همه مراحل با متغیرهای ۰ و ۱ (حداکثر O(p۲) متغیر)
  - صحیح بودن گام‌های متوالی == یک عبارت منطقی

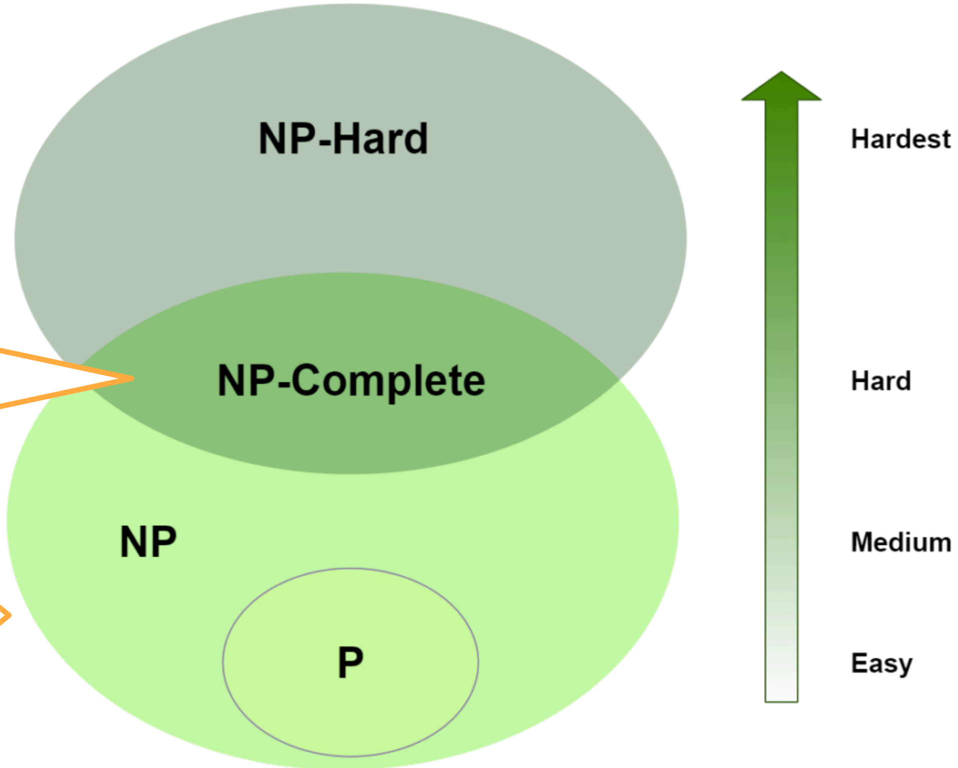
مسئله SAT:

مسئله 3-SAT:

مسئله SAT:

مسئله‌ای هست که اگر راه حل چندجمله‌ای داشته باشد، همه مسئله‌های NP راه حل چندجمله‌ای دارند.

مسئله‌هایی که می‌توان با **ماشین تورینگ غیرقطعی** در زمان **چندجمله‌ای** حل کرد



برخی مسئله‌ها در رده NP ـ تمام

- مسئله SAT، 3-SAT، m-SAT (برای m>=3)، ...

برخی مسئله‌ها در رده NP ـ تمام

- مسئله SAT، 3-SAT، m-SAT (برای m>=3)، ...
- اما 2-SAT راه حل چندجمله‌ای دارد

مسئله حداکثر صدق‌پذیری (MAX-SAT)

- ورودی
  - تعدادی متغیر بولی (صحیح/غلط)
  - یک فرمول به صورت
  - «و»ی یک سری «یا»ی متغیر یا نقیض متغیر
- خروجی
  - مقداردهی که بیشترین تعداد بندها را صحیح کند.

مسئله حداکثر صدق‌پذیری (MAX-SAT)

- ورودی
  - تعدادی متغیر بولی (صحیح/غلط)
  - یک فرمول به صورت
  - «و»ی یک سری «یا»ی متغیر یا نقیض متغیر
- خروجی
  - مقداردهی که بیشترین تعداد بندها را صحیح کند.

- مشابها: مسئله MAX-3-SAT

رده پیچیدگی MAX-3-SAT؟

- سوال؟
  - NP؟
  - NP ـ تمام؟
  - NP ـ سخت؟

-

الگوریتم تقریبی برای MAX-3-SAT؟

- رده پیچیدگی الگوریتم تقریبی برای MAX-3-SAT؟
  - اگر P != NP، هیچ الگوریتم تقریبی با ضریب بهتر از ۷/۸ وجود ندارد.

الگوریتم ۷/۸ ـ تقریب برای MAX-3-SAT

الگوریتم ۷/۸ــ تقریب برای MAX-3-SAT

- همه متغیرها برنولی با احتمال ۱/۲

الگوریتم ۷/۸ـ تقریب برای MAX-3-SAT

- همه متغیرها برنولی با احتمال ۱/۲
- احتمال صحیح شدن هر بند

$$x_1 \vee \overline{x}_2 \vee x_3$$

الگوریتم ۷/۸ـ تقریب برای MAX-3-SAT

- همه متغیرها برنولی با احتمال ۱/۲

- احتمال صحیح شدن هر بند

$$x_1 \vee \overline{x}_2 \vee x_3$$

- =۷/۸

الگوریتم ۷/۸ـ تقریب برای MAX-3-SAT

- همه متغیرها برنولی با احتمال ۱/۲

- احتمال صحیح شدن هر بند

$$x_1 \vee \overline{x}_2 \vee x_3$$

- =۷/۸

- امید تعداد بندهای صحیح شده = 7m/8

الگوریتم ۷/۸ـ تقریب برای MAX-3-SAT

- همه متغیرها برنولی با احتمال ۱/۲

- احتمال صحیح شدن هر بند

$$x_1 \vee \overline{x}_2 \vee x_3$$

  - ۷/۸=
- امید تعداد بندهای صحیح شده = 7m/8
- الگوریتم: ...

الگوریتم ۷/۸ـ تقریب برای MAX-3-SAT

- همه متغیرها برنولی با احتمال ۱/۲
- احتمال صحیح شدن هر بند

$$x_1 \vee \overline{x}_2 \vee x_3$$

بندهای با تعداد کمتر از ۳ لیترال؟

  - ۷/۸=
- امید تعداد بندهای صحیح شده = 7m/8
- الگوریتم: ...

مسئله MAX-2-SAT

- مسئله 2-SAT الگوریتم چندجمله‌ای
- مسئله MAX-2-SAT الگوریتم چندجمله‌ای ندارد (با فرض‌های مناسب)
- تقریب با ضریب بهتر از 0.954 ندارد

مسئله CSP و چرا مهم است؟

تعمیمی از SAT؟

- بندها شامل عبارت‌های منطقی دیگر

تعمیمی از SAT؟

- بندها شامل عبارت‌های منطقی دیگر

Boolean Gate	*Formula*
AND gate	$x_o = x_1.x_2$
NAND gate	$x_o = \overline{x_1.x_2}$
OR gate	$x_o = x_1 + x_2$
NOR gate	$x_o = \overline{x_1 + x_2}$
NOT gate	$x_o = \overline{x_1}$
XOR gate	$x_o = x_1 \oplus x_2$

تعمیمی از SAT؟

- بندها شامل عبارت‌های منطقی دیگر

Boolean Gate	*Formula*	*SAT formulation*
AND gate	$x_o = x_1.x_2$	$(\overline{x_1} \vee \overline{x_2} \vee x_o) \wedge (x_1 \vee \overline{x_o}) \wedge (x_2 \vee \overline{x_o})$
NAND gate	$x_o = \overline{x_1.x_2}$	$(\overline{x_1} \vee \overline{x_2} \vee \overline{x_o}) \wedge (x_1 \vee x_o) \wedge (x_2 \vee x_o)$
OR gate	$x_o = x_1 + x_2$	$(x_1 \vee x_2 \vee \overline{x_o}) \wedge (\overline{x_1} \vee x_o) \wedge (\overline{x_2} \vee x_o)$
NOR gate	$x_o = \overline{x_1 + x_2}$	$(x_1 \vee x_2 \vee x_o) \wedge (\overline{x_1} \vee \overline{x_o}) \wedge (\overline{x_2} \vee \overline{x_o})$
NOT gate	$x_o = \overline{x_1}$	$(\overline{x_1} \vee \overline{x_o}) \wedge (x_1 \vee x_o)$
XOR gate	$x_o = x_1 \oplus x_2$	$(\overline{x_1} \vee \overline{x_2} \vee \overline{x_o}) \wedge (x_1 \vee x_2 \vee \overline{x_o}) \wedge (x_1$ $\vee \overline{x_2} \vee x_o) \wedge (\overline{x_1} \vee x_2 \vee x_o)$

تعمیمی از SAT؟

- بندها شامل عبارت‌های منطقی دیگر

Boolean Gate	*Formula*	*SAT formulation*
AND gate	$x_o = x_1.x_2$	$(\overline{x_1} \vee \overline{x_2} \vee x_o) \wedge (x_1 \vee \overline{x_o}) \wedge (x_2 \vee \overline{x_o})$
NAND gate	$x_o = \overline{x_1.x_2}$	$(\overline{x_1} \vee \overline{x_2} \vee \overline{x_o}) \wedge (x_1 \vee x_o) \wedge (x_2 \vee x_o)$
OR gate	$x_o = x_1 + x_2$	$(x_1 \vee x_2 \vee \overline{x_o}) \wedge (\overline{x_1} \vee x_o) \wedge (\overline{x_2} \vee x_o)$
NOR gate	$x_o = \overline{x_1 + x_2}$	$(x_1 \vee x_2 \vee x_o) \wedge (\overline{x_1} \vee \overline{x_o}) \wedge (\overline{x_2} \vee \overline{x_o})$
NOT gate	$x_o = \overline{x_1}$	$(\overline{x_1} \vee \overline{x_o}) \wedge (x_1 \vee x_o)$
XOR gate	$x_o = x_1 \oplus x_2$	$(\overline{x_1} \vee \overline{x_2} \vee \overline{x_o}) \wedge (x_1 \vee x_2 \vee \overline{x_o}) \wedge (x_1 \vee \overline{x_2} \vee x_o) \wedge (\overline{x_1} \vee x_2 \vee x_o)$

همه عبارت‌ها را می‌توان به صورت SAT نوشت

مسئله k-CSP

- عبارت شامل تعدادی بند
- هر بند یک عبارت منطقی برروی k متغیر
  - عبارت منطقی = $P\colon D^k \to \{\mathsf{False}, \mathsf{True}\}$
  - مجموعه توابع مورد قبول: $\mathscr{P}$
    - مثال: یای چند لیترال
- تعمیم: متغیرها بتوانند مقدار از مجموعه D بگیرند
  - مثال: $D = \{\mathsf{False}, \mathsf{True}\}$

مسئله k-CSP

- عبارت شامل تعدادی بند
- هر بند یک عبارت منطقی برروی k متغیر
  - عبارت منطقی = $P: D^k \to \{\mathsf{False}, \mathsf{True}\}$
  - مجموعه توابع مورد قبول: $\mathscr{P}$
    - مثال: یای چند لیترال
- تعمیم: متغیرها بتوانند مقدار از مجموعه D بگیرند
  - مثال: $D = \{\mathsf{False}, \mathsf{True}\}$

مثال خاص: 3-SAT

عبارت‌های
قابل قبول

متغیرهای
هر بند

$$k\text{-CSP}[\mathcal{P}]$$

عبارت‌های قابل قبول

متغیرهای هر بند

$$k\text{-CSP}[\mathcal{P}]$$

مثال:

$$P_1(x_1, x_2, x_3) \wedge P_2(x_2, x_1, x_4) \wedge \cdots \wedge P_m(x_3, x_5, x_1)$$

A *class of constraint satisfaction problems* k-CSP[$\mathcal{P}$] is specified by

- A finite domain D
- A natural number k (the arity)
- A set $\mathcal{P}$ of k-ary predicates over D

k and $|D|$ are usually treated as *constants*. An *instance* of k-CSP[$\mathcal{P}$] is

$$P_1(x_{i_{11}}, x_{i_{12}}, \ldots, x_{i_{1k}}) \wedge P_2(x_{i_{21}}, x_{i_{22}}, \ldots, x_{i_{2k}}) \wedge \cdots$$

$$\wedge P_m(x_{i_{m1}}, x_{i_{m2}}, \ldots, x_{i_{mk}}),$$

where $P_1, \ldots, P_m \in \mathcal{P}$ and $i_{11}, i_{12}, \ldots, i_{mk} \in \{1, 2, \ldots, n\}$. An *assignment* for this instance is an n-tuple $(a_1, \ldots, a_n) \in D^n$, and the generalized clause $P_\ell(x_{i_{\ell 1}}, \ldots, x_{i_{\ell k}})$ is *satisfied* by that assignment if $P_\ell(a_{i_{\ell 1}}, \ldots, a_{i_{\ell k}}) = \mathsf{True}$.

برخی حالت‌های خاص

- مسئله MaxCut؟
  - بند: x!=y
- D = {True, False}

۲

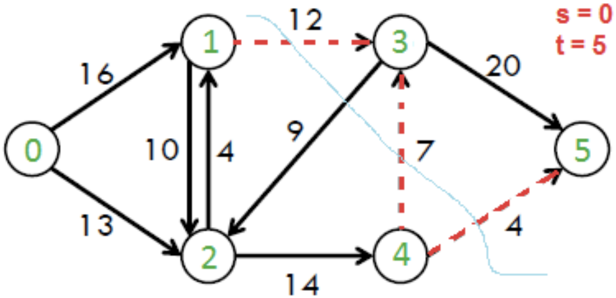
عبارت‌های

x != y

$$k\text{-CSP}[\mathcal{P}]$$

برخی حالت‌های خاص

- مسئله MaxDiCut ؟
  - بند
- D = {True, False}



عبارت‌های “$x \wedge \overline{y}$,”

۲

k-CSP[$\mathcal{P}$]

برخی حالت‌های خاص

- مسئله Max-2-SAT ؟
  - بند:
- D = {True, False}

۲

$$k\text{-CSP}[\mathcal{P}]$$

برخی حالت‌های خاص

- مسئله Max-2-SAT ؟
  - بند:
- D = {True, False}

۲

عبارت‌های
"$x \wedge y$",
"$\overline{x} \wedge y$",
"$x \wedge \overline{y}$",
"$\overline{x} \wedge \overline{y}$"

$$k\text{-CSP}[\mathcal{P}]$$

برخی حالت‌های خاص

- مسئله ؟؟؟؟ ؟
  - بند:
- D = {True, False}

۲

عبارت‌های “$x = y$.”

$$k\text{-CSP}[\mathcal{P}]$$

برخی حالت‌های خاص

- مسئله ؟؟؟؟ ؟
  - بند:
- $D = \{1, 2, 3\}$

۲

عبارت‌های “$x \neq y.$”

$$k\text{-CSP}[\mathcal{P}]$$

برخی حالت‌های خاص

- مسئله ؟؟؟؟ ؟
  - بند:
- D = {1, 2, 3}

۳ـ رنگ‌پذیری

عبارت‌های
“$x \neq y$.”

۲

k-CSP[$\mathcal{P}$]

حدس

- حدس: نسخه تصمیم مسئله یا P است یا NP_ تمام

$$k\text{-CSP}[\mathcal{P}]$$

آرام‌سازی 2-CSPها

نمایش حقیقی متغیرهای منطقی

$$y_i = 1 \quad x_i = \text{True}$$

$$y_i = 0 \quad x_i = \text{False}$$

نمایش حقیقی متغیرهای منطقی

$$y_i = 1 \quad x_i = \text{True}$$

$$y_i = 0 \quad x_i = \text{False}$$

$$y_i \in \{0, 1\}$$

قید xi = True یا False

نمایش حقیقی متغیرهای منطقی

$$y_i = 1 \quad x_i = \text{True}$$

$$y_i = 0 \quad x_i = \text{False}$$

$$y_i \in \{0, 1\}$$

قید xi = True یا False

$$y_i(1 - y_i) = 0$$

نمایش حقیقی متغیرهای منطقی

$$y_i = 1 \qquad x_i = \text{True}$$

$$y_i = 0 \qquad x_i = \text{False}$$

$$y_i \in \{0, 1\}$$

قید xi = True یا False

$$y_i(1 - y_i) = 0$$

تابع هدف:

$$\sum_{\ell=1}^{m} f_\ell(y_1, \ldots, y_n)$$

؟

تعريف $f_\ell(y_1, \ldots, y_n)$

$$x_1 \vee x_2$$

$$f_1(y_1, \ldots, y_n) := y_1 y_2 + y_1(1 - y_2) + (1 - y_1) y_2$$

تعريف $f_\ell(y_1, \ldots, y_n)$

$$x_1 \vee x_2$$

$$f_1(y_1, \ldots, y_n) := y_1 y_2 + y_1(1 - y_2) + (1 - y_1) y_2$$

$$y_1 + y_2 - y_1 y_2$$

تعریف $f_\ell(y_1, \ldots, y_n)$

$$x_1 \vee x_2$$

$$f_1(y_1, \ldots, y_n) := y_1 y_2 + y_1(1 - y_2) + (1 - y_1) y_2$$

$$y_1 + y_2 - y_1 y_2$$

دقیقا یکی از موارد زیر ۱ است

$y_1 y_2 \qquad (1 - y_1) y_2 \qquad y_1(1 - y_2) \qquad (1 - y_1)(1 - y_2)$

نمایش حقیقی متغیرهای منطقی

بیشینه کن: $\sum_{\ell=1}^{m} f_\ell(y_1, \ldots, y_n)$

$$y_i(1 - y_i) = 0$$

$$f_1(y_1, \ldots, y_n) := y_1 y_2 + y_1(1 - y_2) + (1 - y_1) y_2$$

نمایش حقیقی متغیرهای منطقی

بیشینه کن: $\sum_{\ell=1}^{m} f_\ell(y_1, \ldots, y_n)$

$$y_i(1 - y_i) = 0$$

$$f_1(y_1, \ldots, y_n) := y_1 y_2 + y_1(1 - y_2) + (1 - y_1)y_2$$

مسئله 2-CSP ==> برنامه‌نویسی درجه ۲

آرام‌سازی پایه‌ای

$$y_i \qquad\qquad \mathbf{t}_i$$

$$y_i(1 - y_i) = 0 \qquad\qquad \mathbf{t}_i^T(\mathbf{e} - \mathbf{t}_i) = 0$$

به جای ۱

آرام‌سازی پایه‌ای، مثال

The basic semidefinite relaxation of Max-2-Sat shown for the formula $(x_1 \vee x_2) \wedge (\overline{x}_2 \vee x_4) \wedge (x_1 \vee \overline{x}_3)$

$$\begin{array}{ll} \text{Maximize} & \mathbf{t}_1^T \mathbf{t}_2 + \mathbf{t}_1^T(\mathbf{e} - \mathbf{t}_2) + (\mathbf{e} - \mathbf{t}_1)^T \mathbf{t}_2 \\ & + (\mathbf{e} - \mathbf{t}_2)^T \mathbf{t}_4 + (\mathbf{e} - \mathbf{t}_2)^T (\mathbf{e} - \mathbf{t}_4) + \mathbf{t}_2^T \mathbf{t}_4 \\ & + \mathbf{t}_1^T (\mathbf{e} - \mathbf{t}_3) + \mathbf{t}_1^T \mathbf{t}_3 + (\mathbf{e} - \mathbf{t}_1)^T (\mathbf{e} - \mathbf{t}_3) \\ \text{subject to} & \mathbf{e}^T \mathbf{e} = 1 \\ & \mathbf{t}_i^T (\mathbf{e} - \mathbf{t}_i) = 0 \text{ for all } i. \end{array}$$

روش دیگر مدل‌سازی

$$y_i = +1 \quad x_i = \text{True}$$

$$y_i = -1 \quad x_i = \text{False}$$

روش دیگر مدل‌سازی

$$y_i = +1 \quad x_i = \text{True}$$
$$y_i = -1 \quad x_i = \text{False}$$

$$y_i \in \{0, 1\}$$

قید True = xi یا False

روش دیگر مدل‌سازی

$$y_i = +1 \quad x_i = \text{True}$$
$$y_i = -1 \quad x_i = \text{False}$$

$$y_i \in \{0, 1\}$$

قید True = xi یا False

$$y_i^2 = 1$$

روش دیگر مدل‌سازی

$$\mathbf{v}_0^T \mathbf{v}_i = +1$$

$$y_i = +1 \quad x_i = \text{True}$$

$$y_i = -1 \quad x_i = \text{False}$$

$$y_i \in \{0, 1\}$$

قید xi = True یا False

$$y_i^2 = 1$$

روش دیگر مدل‌سازی

$$\mathbf{v}_0^T \mathbf{v}_i = +1 \qquad y_i = +1 \quad x_i = \text{True}$$

$$\mathbf{v}_0^T \mathbf{v}_i = -1 \qquad y_i = -1 \quad x_i = \text{False}$$

$$y_i \in \{0, 1\}$$

قید xi = True یا False

$$y_i^2 = 1$$

روش دیگر مدل‌سازی

$$\mathbf{v}_0^T \mathbf{v}_i = +1 \qquad y_i = +1 \quad x_i = \text{True}$$

$$\mathbf{v}_0^T \mathbf{v}_i = -1 \qquad y_i = -1 \quad x_i = \text{False}$$

$$y_i \in \{0, 1\}$$

قید True = xi یا False

$$\|\mathbf{v}_i\|^2 = 1 \qquad y_i^2 = 1$$

الگوریتم α_{GW} ـ تقریب

- همان الگوریتم GW: