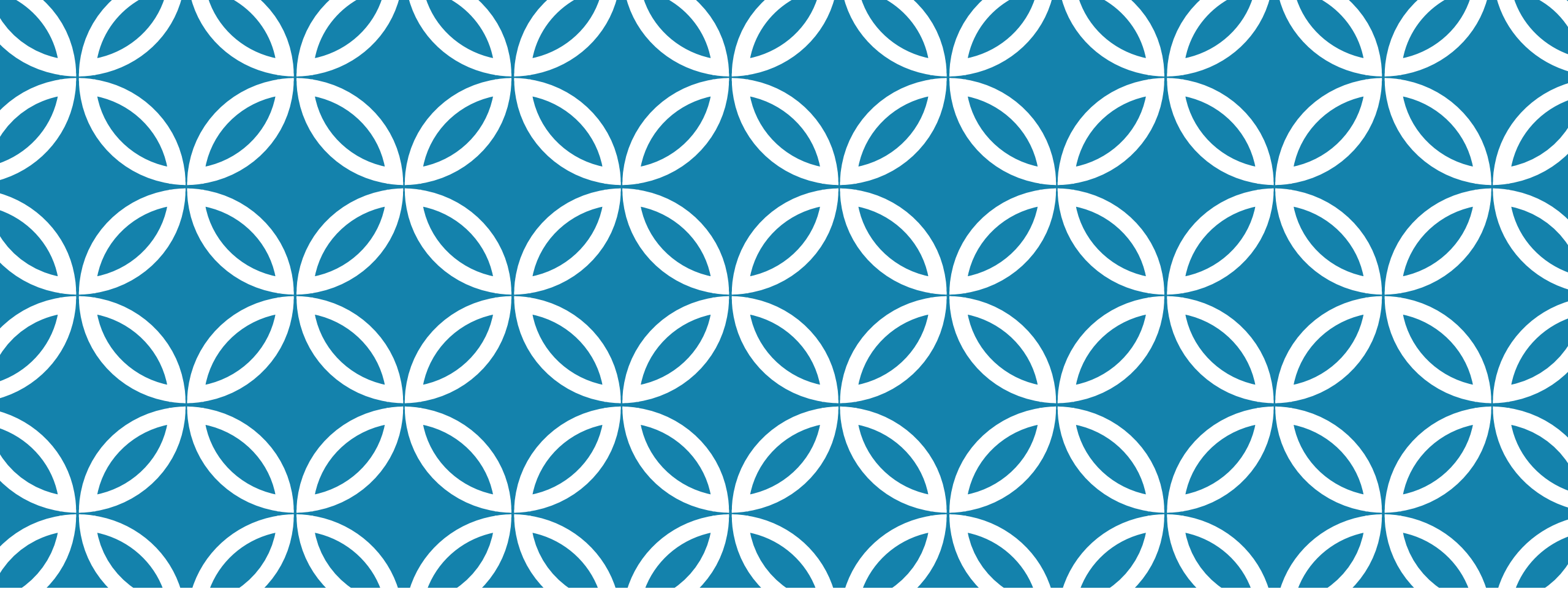




$P = ? NP$

محسن هوشمند

دانشکده علم رایانه و فناوری اطلاعات

دانشگاه تحصیلات تکمیلی علوم پایه زنجان

مباحثی که عرضه خواهد شد

مرور مباحث زبان‌ها و مدل‌های رایانشی

ماشین تورینگ و معادل‌های آن

مختصری رایانش‌پذیری

مختصری نظریه پیچیدگی

مباحث $P =? NP$

رایانه‌ها

دستگاه فاقد احساسات و مطیع قوانین منطق

بررسی آنها

- قادر به انجام چه کاری و اینکه چه کاری را انجام نمی‌دهند.
- همچنین مسائلی که در نظر دارای راه‌حل ولی عاجز از راه‌حل رضایت‌آمیز آنها
- در عمل نیاز به میزان عظیمی از زمان و فضا

رایانه‌ها

چند مسئله

- انجام کارهای نیازمند زمان طولانی در عرض چند ثانیه و حتی زیر ثانیه
- حذف ده‌هزار ایمیل دریافتی در آنی از ثانیه
- آن قدر سریع و عدم خطور انجام کاری به ذهن

جمع و ضرب

- جمع دو عدد n رقمی
- نیاز به n عملیات جمع
- ضرب دو عدد n رقمی
- n^2 ضرب به جز جمع‌ها
- برای رایانه زیاد نیست.
- دو عدد صد رقمی منجر به 10^4 عملیات ضرب
- در هر ثانیه اجرای صد هزار یا یک میلیون عمل ضرب منجر به اجرای کل ضرب در کمتر از نیم ثانیه

توجه به تعداد اعمال پایه‌ای به جای سرعت رایانه

رایانه‌ها

جستجو

- قفسه‌ای از پر کتاب
- جستجوی کامل از مرتبه زمانی n
- مرتب‌شده به ترتیب الفبا
- دارای روشی با زمان اجرای $\log_2 n$
- برای ۱۰۰۰ کتاب حدود ده جستجو
- وجود الگوریتم‌های متفاوت برای مسئله‌ای خاص
- کوشش برای یافتن الگوریتم با بیشترین کارایی

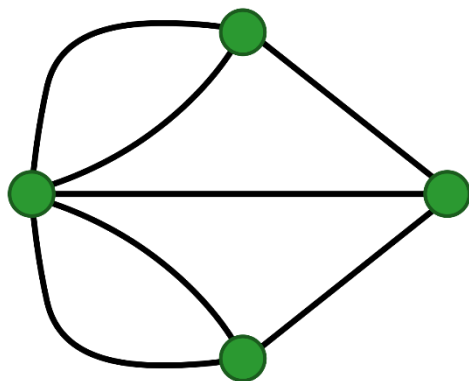
مرتب‌سازی

- الگوریتم مرتب‌سازی انتخابی از مرتبه n^2
- الگوریتم مرتب‌سازی ادغامی از مرتبه $n \log_2 n$
- ۱۰۰۰ عضو با روش اول یک میلیون عملیات با روش دوم ده‌هزار عملیات

رایانه‌ها

دور اویلری

- شروع از شهر کونیگسبرگ و هفت پلش
- گذر دقیقا یکبار از تمامی یال‌ها و برگشت به گره نخست
- ۱۷۳۶ اوئلر ۲۹ ساله
- مدل‌سازی گرافی
- درجه تمامی گره‌ها زوج باشد ممکن و گرنه ناممکن



رایانه‌ها

چند مسئله سخت

فروشنده دوره گرد

- گذر از تعدادی شهر و عبور از هر شهر دقیقا یکبار و برگشت به شهر نخست
- با شرط کمترین مسافت ممکن
- شش شهر حدود ۷۲۰ حالت

رایانه‌ها

چند مسئله سخت

فروشنده دوره گرد

- گذر از تعدادی شهر و عبور از هر شهر دقیقا یکبار و برگشت به شهر نخست
- با شرط کمترین مسافت ممکن
- شش شهر حدود ۷۲۰ حالت
- ۱۰۰ شهر

رایانه‌ها

چند مسئله سخت

فروشنده دوره گرد

- گذر از تعدادی شهر و عبور از هر شهر دقیقا یکبار و برگشت به شهر نخست
- با شرط کمترین مسافت ممکن
- شش شهر حدود ۷۲۰ حالت
- ۱۰۰ شهر
- ۱۰۰!

رایانه‌ها

چند مسئله سخت

فروشنده دوره گرد

- گذر از تعدادی شهر و عبور از هر شهر دقیقا یکبار و برگشت به شهر نخست
- با شرط کمترین مسافت ممکن
- شش شهر حدود ۷۲۰ حالت
- ۱۰۰ شهر
- $۱۰۰! > ۹ \times ۱۰^{۱۵۷}$ یا $۱۰۰!$

رایانه‌ها

چند مسئله سخت

فروشنده دوره گرد

- گذر از تعدادی شهر و عبور از هر شهر دقیقا یکبار و برگشت به شهر نخست
- با شرط کمترین مسافت ممکن
- شش شهر حدود ۷۲۰ حالت
- ۱۰۰ شهر
- $۱۰۰! > ۹ \times ۱۰^{۱۵۷} > ۱ \times ۱۰^{۱۰۰}$ یا $۱۰۰!$
- عدد گوگل

رایانه‌ها

چند مسئله سخت

فروشنده دوره گرد

- گذر از تعدادی شهر و عبور از هر شهر دقیقا یکبار و برگشت به شهر نخست
- با شرط کمترین مسافت ممکن
- شش شهر حدود ۷۲۰ حالت
- ۱۰۰ شهر

$$۱۰۰! > ۹ \times ۱۰^{۱۵۷} > ۱ \times ۱۰^{۱۰۰} \text{ یا } ۱۰۰!$$

▪ عدد گوگل

- فرض بررسی یک میلیون دور در ثانیه در رایانه

$$\frac{۹ \times ۱۰^{۱۵۷}}{۱۰^۶} = ۹ \times ۱۰^{۱۵۱}$$

رایانه‌ها

چند مسئله سخت

فروشنده دوره گرد

- گذر از تعدادی شهر و عبور از هر شهر دقیقاً یکبار و برگشت به شهر نخست
- با شرط کمترین مسافت ممکن
- شش شهر حدود ۷۲۰ حالت
- ۱۰۰ شهر

$$100! > 9 \times 10^{157} > 1 \times 10^{100} \text{ یا } 100! > 9 \times 10^{157} > 1 \times 10^{100}$$

▪ عدد گوگل

▪ فرض بررسی یک میلیون دور در ثانیه در رایانه

$$\frac{9 \times 10^{157}}{10^6} = 9 \times 10^{151}$$

▪ تعداد دقیقه لازم

$$\frac{9 \times 10^{151}}{60} = 15 \times 10^{149}$$

رایانه‌ها

چند مسئله سخت

فروشنده دوره گرد

- گذر از تعدادی شهر و عبور از هر شهر دقیقاً یکبار و برگشت به شهر نخست
- با شرط کمترین مسافت ممکن
- شش شهر حدود ۷۲۰ حالت
- ۱۰۰ شهر

$$۱۰۰! > ۹ \times ۱۰^{۱۵۷} > ۱ \times ۱۰^{۱۰۰} \text{ یا } ۱۰۰! > ۹ \times ۱۰^{۱۵۷} > ۱ \times ۱۰^{۱۰۰}$$

▪ عدد گوگل

▪ فرض بررسی یک میلیون دور در ثانیه در رایانه

$$\frac{۹ \times ۱۰^{۱۵۷}}{۱۰^۶} = ۹ \times ۱۰^{۱۵۱}$$

▪ تعداد دقیقه لازم

$$\frac{۹ \times ۱۰^{۱۵۱}}{۶۰} = ۱۵ \times ۱۰^{۱۴۹}$$

▪ تعداد ساعت لازم

$$\frac{۱۵ \times ۱۰^{۱۴۹}}{۶۰} = ۲۵ \times ۱۰^{۱۴۶}$$

رایانه‌ها

چند مسئله سخت

فروشنده دوره گرد

▪ گذر از تعدادی شهر و عبور از هر شهر دقیقا یکبار و برگشت به شهر نخست

▪ با شرط کمترین مسافت ممکن

▪ شش شهر حدود ۷۲۰ حالت

▪ ۱۰۰ شهر

▪ $100! > 9 \times 10^{157} > 1 \times 10^{100}$ یا $100!$

▪ عدد گوگل

▪ فرض بررسی یک میلیون دور در ثانیه در رایانه

$$\frac{9 \times 10^{157}}{10^6} = 9 \times 10^{151}$$

▪ تعداد دقیقه لازم

$$\frac{9 \times 10^{151}}{60} = 15 \times 10^{149}$$

▪ تعداد ساعت لازم

$$\frac{15 \times 10^{149}}{60} = 25 \times 10^{146}$$

▪ تعداد روز لازم

$$\frac{15 \times 10^{146}}{24} = 10^{146}$$

رایانه‌ها

چند مسئله سخت

فروشنده دوره گرد

▪ گذر از تعدادی شهر و عبور از هر شهر دقیقا یکبار و برگشت به شهر نخست

▪ با شرط کمترین مسافت ممکن

▪ شش شهر حدود ۷۲۰ حالت

▪ ۱۰۰ شهر

▪ $100! > 9 \times 10^{157} > 1 \times 10^{100}$ یا $100!$

▪ عدد گوگل

▪ فرض بررسی یک میلیون دور در ثانیه در رایانه

$$\frac{9 \times 10^{157}}{10^6} = 9 \times 10^{151}$$

▪ تعداد دقیقه لازم

$$\frac{9 \times 10^{151}}{60} = 15 \times 10^{149}$$

▪ تعداد ساعت لازم

$$\frac{15 \times 10^{149}}{60} = 25 \times 10^{146}$$

▪ تعداد روز لازم

$$\frac{15 \times 10^{146}}{24} = 10^{146}$$

▪ تعداد سال لازم

$$\frac{10^{146}}{365} = 3 \times 10^{143}$$

رایانه‌ها

چند مسئله سخت

فروشنده دوره گرد

▪ گذر از تعدادی شهر و عبور از هر شهر دقیقا یکبار و برگشت به شهر نخست

▪ با شرط کمترین مسافت ممکن

▪ شش شهر حدود ۷۲۰ حالت

▪ ۱۰۰ شهر

▪ $۱۰۰! > ۹ \times ۱۰^{۱۵۷} > ۱ \times ۱۰^{۱۰۰}$ یا $۱۰۰!$

▪ عدد گوگل

▪ فرض بررسی یک میلیون دور در ثانیه در رایانه

$$\frac{۹ \times ۱۰^{۱۵۷}}{۱۰^۶} = ۹ \times ۱۰^{۱۵۱}$$

▪ تعداد دقیقه لازم

$$\frac{۹ \times ۱۰^{۱۵۱}}{۶۰} = ۱۵ \times ۱۰^{۱۴۹}$$

▪ تعداد ساعت لازم

$$\frac{۱۵ \times ۱۰^{۱۴۹}}{۶۰} = ۲۵ \times ۱۰^{۱۴۶}$$

▪ تعداد روز لازم

$$\frac{۱۵ \times ۱۰^{۱۴۹}}{۲۴} = ۱۰^{۱۴۶}$$

▪ تعداد سال لازم

$$\frac{۱۰^{۱۴۶}}{۳۶۵} = ۳ \times ۱۰^{۱۴۳}$$

▪ تعداد قرن لازم

$$\frac{۳ \times ۱۰^{۱۴۳}}{۱۰۰} = ۳ \times ۱۰^{۱۴۱}$$

▪ !!

رایانه‌ها

$$(p \vee q) \rightarrow \neg(p \wedge q)$$

جدول ارزش 2^n

مسئله دیوفانتین

دور همیلتنی

شروع از گره آغاز و گذر دقیقاً یکبار از تمامی گره‌ها و برگشت به راس آغاز

شباهت دور اویلری و دور همیلتنی

مورد متقدم دارای ترفند

مورد اخیر صرفاً جستجوی کامل

افراز مجموعه

تقسیم به مجموعه‌ها با مقدار برابر

برای ۱۰۰ عضوی نیاز به $10^5 \times 4$ قرن

مسئله صدق‌پذیری

$\vee$ و $\neg$ و $\rightarrow$

نظریه رایانش

- نظریه رایانش (محاسبات) **computation theory**: توانایی‌ها و محدودیت‌های اصلی کامپیوتر
- خودکارها
 - رایانش پذیری (محاسبه پذیری) **computability**
 - پیچیدگی **complexity**

نظریه رایانش

نظریه رایانش (محاسبات) **computation theory**: توانایی‌ها و محدودیت‌های اصلی کامپیوتر

- خودکارها: مدل‌های ریاضی رایانش

- خودکاره متناهی خم FA و زبان‌های منظم

- پردازش متن

- مترجم‌ها

- طراحی سخت‌افزار

- خودکاره پشته‌ای و دستورهای مستقل از متن

- هوش مصنوعی

- مترجم‌ها

- پردازش زبان

- ماشین تورینگ

- رایانش پذیری (محاسبه پذیری) **computability**

- پیچیدگی **complexity**

نظریه رایانش

نظریه رایانش (محاسبات) **computation theory**: توانائی‌ها و محدودیت‌های اصلی کامپیوتر
▪ خودکارها

▪ رایانش پذیری (محاسبه پذیری) **computability**

▪ گودل، تورینگ، چرچ، پست کشف کردند که بعضی مسائل حل ناپذیر با رایانه

▪ چه مسئله‌ای حل پذیر است و چه مسئله‌ای حل پذیر نیست

▪ پیچیدگی **complexity**

نظریه رایانش

نظریه رایانش (محاسبات) **computation theory**: توانائی‌ها و محدودیت‌های اصلی کامپیوتر

- خودکارها
- رایانش پذیری (محاسبه پذیری) **computability**
- پیچیدگی **complexity**
- چه چیزی بعضی مسائل را سخت و دیگر مسائل را آسان می‌کند
- پاسخ؟ نمی‌دانیم!
- روشی برای طبقه‌بندی
- فهم اینکه چه جنبه‌ای از مسئله دارای رایانش (محاسبه) مشکل است
- پاسخ‌های با دقت کمتر (تقریب)
- بعضی مسائل در بدترین حالت مشکل هستند
- روش‌های دیگر حل مسئله
- مثال - رمزنگاری

چند مفهوم - رشته‌ها و زبان‌ها

رشته‌ها: سازنده‌های اصلی علم رایانه
▪ خود حاصل مجموعه‌ای از علامت‌ها به نام الفبا
▪ Σ, Γ

$$\Sigma = \{0,1\}$$
$$\Sigma = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z\}$$
$$\Sigma = \{\text{الف, ب, ت, ث, ج, چ, ح, خ, د, ذ, ر, ز, س, ش, ص, ض, ط, ظ, ع, غ, ف, ق, ک, گ, ل, م, ن, و, ه, ی, ء}\}$$
$$\Sigma = \{0,1, \text{الف, ب}\}$$

رشته روی الفبا: دنباله متناهی از حروف الفبا

طول رشته: تعداد علامت‌ها

$$|w|$$

$$\Sigma = \{0,1\}$$

$$w = 01100110$$

زبان زیرمجموعه‌ای از Σ^*

ماشین‌ها

خودکارهای که پاسخ آن صرفاً منحصر به بله یا خیر

▪ پذیرنده Acceptor

▪ پذیرنده پس از دریافت رشته ورودی یا آن را می‌پذیرد یا رد می‌کند.

▪ ماشین کلی‌تر که رشته‌هایی از نشانه‌های خروجی تولید می‌کند

مبدل

▪ تبدیل‌ساز رشته ورودی را به رشته خروجی

▪ transducer

تشخیص‌گر

▪ تشخیص‌پذیر

▪ بله/خیر/حلقه

▪ recognizer

پذیرنده

▪ تصمیم‌پذیر

▪ بله/خیر

▪ accepter

تعریف دستور

دستور به صورت چهارتایی $G = (V, \Sigma, S, R)$ نمایش داده می‌شود که

V مجموعه متناهی از اشیاء به نام متغیرهاست.

Σ مجموعه متناهی الفباست

$S \in V$ متغیر آغاز است.

R مجموعه متناهی از قوانین تولید (یا قوانین جانشینی)

مجموعه‌های V و Σ ناتهی و دارای اشتراک تهی

در حالت کلی تمامی قوانین دستور به صورت

$$x \rightarrow y$$

است که $x \in (V \cup \Sigma)^+$ و $y \in (V \cup \Sigma)^*$

اشتقاق: رشته قوانین جانشینی که منجر به رشته‌ای خواهد شد.

تعریف دستور منظم

○ فرض می‌کنیم که $A, B \in V$ و $x \in \Sigma^*$ باشد. دستور $G=(V, \Sigma, S, R)$ منظم است اگر قوانین آن به صورت راستگرد یا چپگرد باشد.



دستور راستگرد: اگر قوانین دستور منظم $G=(V, \Sigma, S, R)$ به صورت $A \rightarrow xB|x$ باشد.

دستور چپگرد: اگر قوانین دستور منظم $G=(V, \Sigma, S, R)$ به صورت $A \rightarrow Bx|x$ باشد.

دستور چپگرد یا راستگرد، دستور منظم است.


```

graph LR
    A[رشته ورودی] --> B[خَم]
    B --> C[پذیرش / بله]
    B --> D[رد / خیر]
  
```

رشته روی نوار w

نوار ورودی

نوک (فقط خواندن)

کنترل
متناهی

پذیرش رشته

■ پایان رشته همزمان با بودن در حالت پذیرش

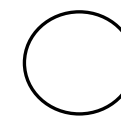
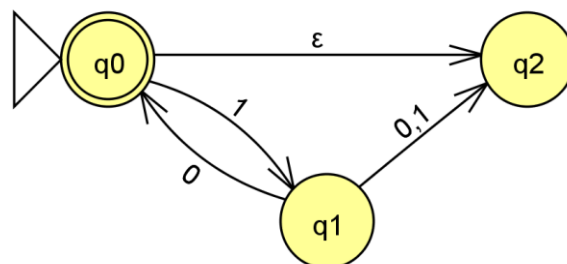
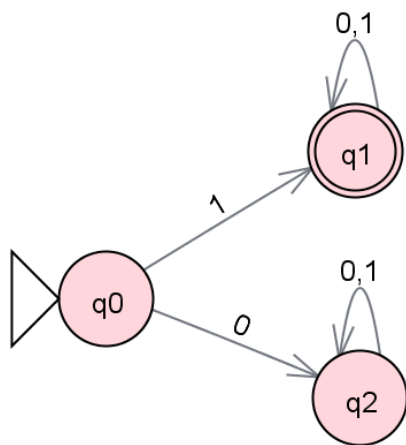
Finite Automata

نمایش خودکارۀ متناهی

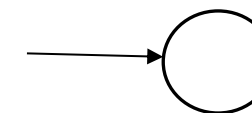
$$\Sigma = \{0, 1\}$$

$$L_1 = \{ \text{مجموعه تمام رشته‌هایی که با '1' شروع می‌شوند} \}$$

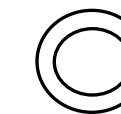
$$= \{1, 11, 10, 101, 111, \dots\}$$



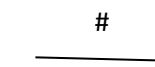
حالت



حالت آغاز



حالت پایان (پذیرش)



انتقال

قضیه کلین (نحوه تشخیص منظم بودن یک زبان)

زبانی منظم است اگر آن را بتوان با

- ختم نشان داد.
- ختم نشان داد.
- عبارت منظم نشان داد.
- دستور منظم نشان داد.

شناسائی زبان‌های نامنظم

نامتناهی بودن بعضی از زبان‌های نامنظم

اما وجود محدودیت‌هایی بر این گونه زبان‌ها به خاطر ماهیت حافظه محدود خودکاره (متناهی بودن حافظه)

زبانی زمانی منظم است که میزان اطلاعات لازم برای نگهداری در حافظه در حین پردازش هر رشته محدود باشد.

دستور مستقل از متن

تعریف- دستور $G = (V, \Sigma, S, R)$ مستقل از متن است اگر قوانین آن به صورت $A \rightarrow x$ باشد،
به طوری که $A \in V$ و $x \in (V \cup \Sigma)^*$

زبان مستقل از متن است اگر و فقط اگر دستور مستقل از متن G موجود باشد به طوری که $L=L(G)$

هر دستور منظم، دستور مستقل از متن است.

خودکاره پشته‌ای نامعین (خپن)

هفت تائی $(Q, \Sigma, \Gamma, \delta, q_0, z, F)$ است به طوری که

- Q مجموعه متناهی حالت‌ها

- دو الفباء

- Σ الفباء ورودی (متناهی)

- Γ الفباء پشته (متناهی)

- $\delta: Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow Q \times \Gamma^*$ تابع انتقال

- با هر زوج حالت فعلی و ورودی فعلی، حالت جدید

- $q_0 \in Q$ حالت آغاز

- $z \in \Gamma$ نشان ته پشته

- $F \subseteq Q$ مجموعه حالت‌های پذیرش (نهائی)

$$\delta(q_i, a, A) = (q_j, B)$$

دو خروجی

سه ورودی

q_i حالت فعلی خپ، نوک خواندن روی a ، A در بالای پشته

q_j حالت بعدی و A از پشته برداشته شود و B قرار گیرد

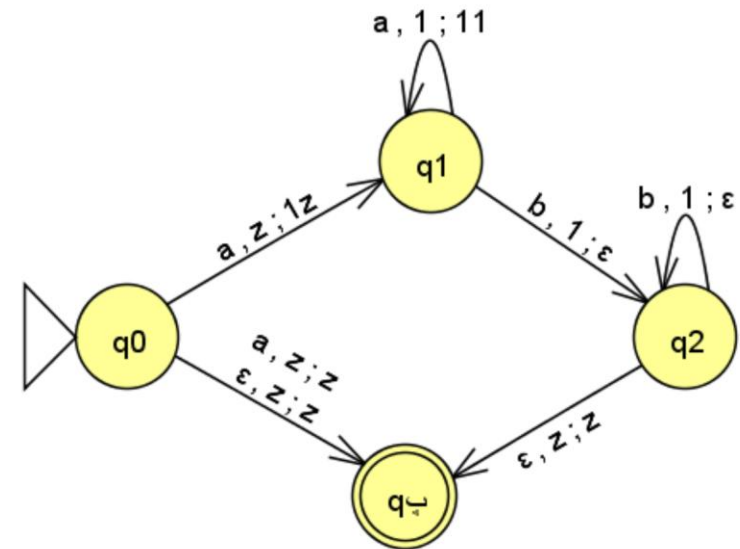
مثال - $L = \{a^n b^n, n \geq 1\}$

$S \rightarrow aSb|ab$

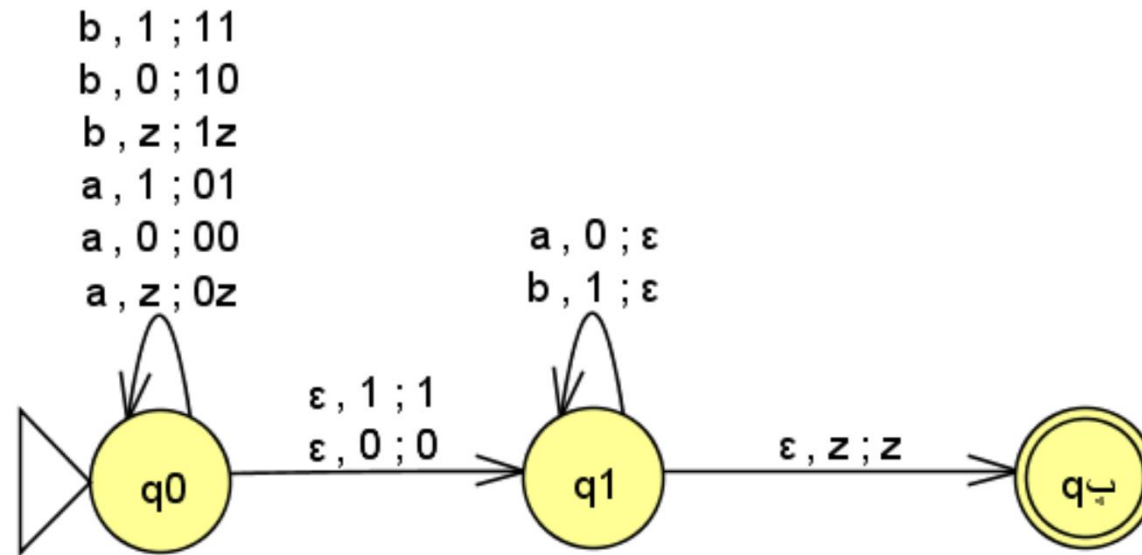
$S \Rightarrow ab$

$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaabbb$

$L = \{a^n b^n, n \geq 1\}$



مثال- $L = \{w^r w, w \in \Sigma^*\}$



$\delta(q_0, abba, z) \vdash \delta(q_0, bba, 0z) \vdash \delta(q_0, ba, 10z) \vdash \delta(q_1, ba, 10z) \vdash \delta(q_1, a, 0z) \vdash \delta(q_1, \epsilon, z) \vdash \delta(q_f, \epsilon, z)$

زبان‌های حساس به متن و خودکاره کراندار خطی

ماشین تورینگ نامعین که الفبای آن دارای علائم آغاز و پایان

- به طوری که حرکت با دیدن علامت آغاز صرفاً به سمت راست و با علامت پایان صرفاً به سمت چپ

- عدم نیاز محاسبات به فضائی خارج از ورودی اولیه

- مثال $a^n b^n c^n, n \geq 1$

قوی‌تر از خپ

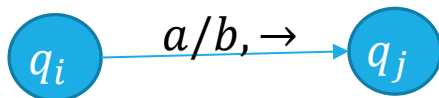
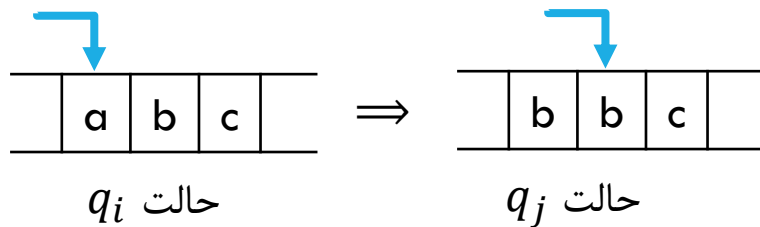
تعریف ماشین تورینگ

$$\delta(q_i, a) = (q_j, b, \leftarrow)$$

دو ورودی

ماشین در حالت q_i ، نوک با خواندن a ، به حالت q_j می‌رود و b را جانشین a می‌کند و به سمت چپ حرکت می‌کند.

$$\delta(q_i, a) = (q_j, b, \rightarrow)$$



چندتایی $M = (Q, \Sigma, \Gamma, \delta, q_0, q_n, q_p)$

Q : مجموعه متناهی حالت‌ها

Σ : مجموعه متناهی الفباء ورودی

Γ : مجموعه متناهی الفباء نوار شامل علامت تهی و

δ : تابع انتقال $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{\rightarrow, \leftarrow\}$

q_0 : حالت آغاز

q_n : همه حالات به جز حالت پذیرش

q_p : حالت [یا حالات] پذیرش $q_p \neq q_n$

نوک خواندن/نوشتن در هر حرکت یک علامت را می‌خواند

ماشین تورینگ استاندارد

توصیف آنی

$M = (Q, \Sigma, \Gamma, \delta, q_0, q_n, q_p)$ ماشین تورینگ است. هر رشته $a_1 a_2 \dots a_{k-1} q a_k a_{k+1} \dots a_n$ با q_1 $q \in Q$ و $a_i \in \Gamma$ توصیف آنی باشد، آن گاه

▪ حرکت $a_1 a_2 \dots a_{k-1} q a_k a_{k+1} \dots a_n \vdash a_1 a_2 \dots a_{k-1} b q_2 a_{k+1} \dots a_n$ ممکن است اگر و فقط اگر

$$\delta(q_1, a_k) = (q_2, b, \rightarrow)$$

▪ حرکت $a_1 a_2 \dots a_{k-1} q a_k a_{k+1} \dots a_n \vdash a_1 a_2 \dots q_2 a_{k-1} b a_{k+1} \dots a_n$ ممکن است اگر و فقط اگر

$$\delta(q_1, a_k) = (q_2, b, \leftarrow)$$

حالت توقف

▪ ماشین M با شروع از وضعیت اولیه $x_1 q_2 x_2 \xRightarrow{*} y_1 q_j y_2$ متوقف می‌شود، اگر به ازای هر q_j و a $\delta(q_j, a)$ تعریف نشده باشد.

▪ رایانش (محاسبه) **computation**: دنباله‌ای از پیکربندی‌ها که به یک حالت توقف منجر می‌شود.

▪ حلقه بی‌پایان $x_1 q_2 x_2 \xRightarrow{*} \infty$

تعریف ماشین تورینگ پذیرنده زبان

Acceptor

رشته روی نوار نوشته می شود و نوک خواندن -نوشتن روی سمت چپ ترین حرف w قرار می گیرد و با شروع از وضع q_0 اگر در حالت نهائی متوقف شود، w پذیرفته می شود.

$M = (Q, \Sigma, \Gamma, \delta, q_0, q_n, q_p)$ ماشین تورینگ است. آن گاه زبانی که M می پذیرد به صورت زیر است:

$$L(M) = \{w \in \Sigma^+ : q_0 w \stackrel{*}{\Rightarrow} x_1 q_p x_2; x_1, x_2 \in \Gamma^*\}$$

اگر w عضو $L(M)$ نباشد

۱- ماشین در حالتی غیرپایانی متوقف می شود

۲- ماشین در حلقه بی پایان می افتد و هرگز متوقف نمی شود.

ماشین تورینگ – مبدل TRANSDUCER

تاکنون تمرکز صرفاً بر پذیرندگی زبان

مدل انتزاعی کامپیوتر دیجیتال

▪ هدف کامپیوتر تبدیل ورودی به خروجی

ورودی رایانش

▪ نشان‌های روی نوار در زمان آغاز،

خروجی رایانش

▪ آنچه در انتها روی نوار است

زبان تورینگ – تصمیم پذیر DECIDABLE

در بعضی منابع زبان بازگشتی

تعریف تصمیم گیری: ماشین روی تمام ورودی ها چه در صورت پذیرش چه در صورت رد کردن، متوقف خواهد شد.

تعریف زبان تورینگ-تصمیم پذیر:

▪ زبانی که ماشین تورینگی آن را تصمیم بگیرد.

دو حالت در قبل رشته ورودی

▪ ورودی پذیرفته شود (ماشین در حالت پذیرش متوقف خواهد شد)

▪ ورودی پذیرفته نشود (ولی ماشین در حالت غیرپذیرش توقف کند)

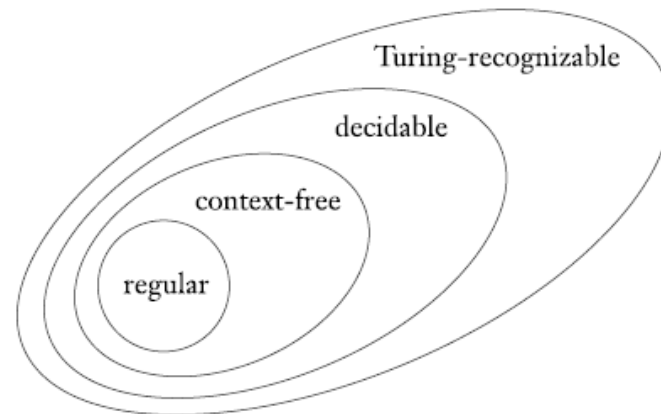
▪ در بعضی منابع افزودن حالت رد و در صورت پذیرفته نشدن ماشین در حالت رد

زبان تورینگ - تصمیم‌پذیر در مقابل تورینگ - تشخیص‌پذیر

بعضی زبان‌ها تشخیص‌پذیرند ولی تصمیم‌ناپذیرند

قضیه: مسئله اینکه زبان مستقل از متنی رشته‌ای را تولید می‌کند، تصمیم‌پذیر است

قضیه: مسئله اینکه ماشین تورینگ یک رشته ورودی را می‌پذیرد یا نه، تشخیص‌پذیر است.



انواع زبان - سلسله مراتب چامسکی - شوتزنبرگر

$$G = (V, \Sigma, S, R)$$

نوع ۳- منظم: قوانینی به صورت $A \rightarrow xB|x$ یا $A \rightarrow Bx|x$ دارد.

نوع ۲- مستقل از متن: در سمت چپ تمامی قوانین آن، صرفاً یک متغیر باشد.

نوع ۱- حساس به متن: قوانین آن $x \rightarrow y$ است که $x, y \in (V \cup \Sigma)^+$ و $|x| \leq |y|$.

نوع صفر- بدون محدودیت: هیچ محدودیتی روی قوانین تولید اعمال نشود مگر اینکه نباید ϵ در سمت چپ قوانین باشد.

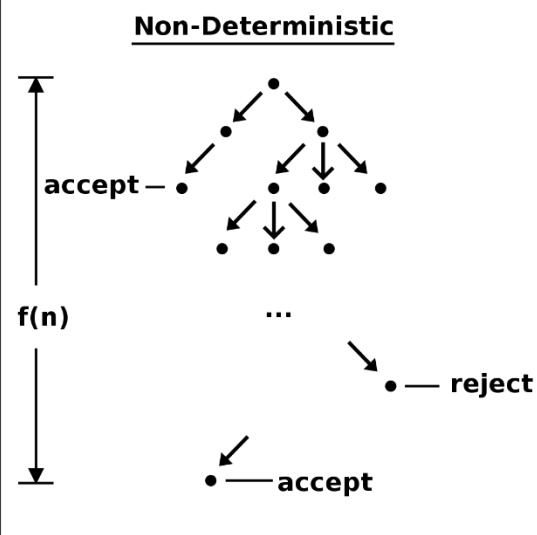
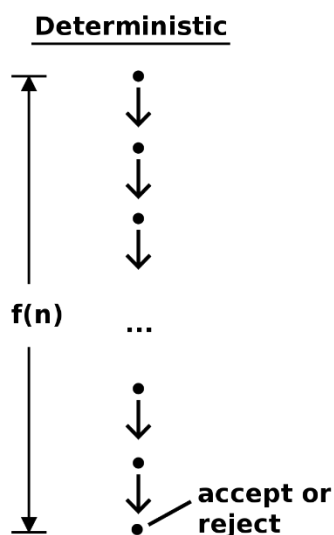
دستور	زبان	خودکاره متناظر	قواعد تولید	مثال
نوع-۳	منظم	خودکاره متناهی	$A \rightarrow xB x$	a^n
نوع-۲	مستقل-از متن	خودکاره پشته‌ای	$A \rightarrow (V \cup \Sigma)^+$	$a^n b^n$
نوع-۱	حساس به متن	خودکاره کراندار متناهی	$x \rightarrow y, x \leq y $	$a^n b^n c^n$
نوع-۰	بدون محدودیت	بدون محدودیت		$a^n b^n c^n$

انواع دیگر ماشین‌های تورینگ

صرفاً تعریف ماشین تورینگِ معینِ تک‌نواره نامحدود (از اینجا به اختصار ماشین تورینگ)

مدل‌های دیگر

- ماشین تورینگ با حالت ایست $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{\rightarrow, \leftarrow, \downarrow\}$
- ماشین تورینگ با نوار نیمه متناهی
- ماشین تورینگ چند نواره $\delta: Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{\rightarrow, \leftarrow\}^k$
- ماشین تورینگ نامعین $\delta: Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{\rightarrow, \leftarrow\}}$
- قضیه هر ماشین تورینگ نامعین، ماشین تورینگ معین متناظر دارد.



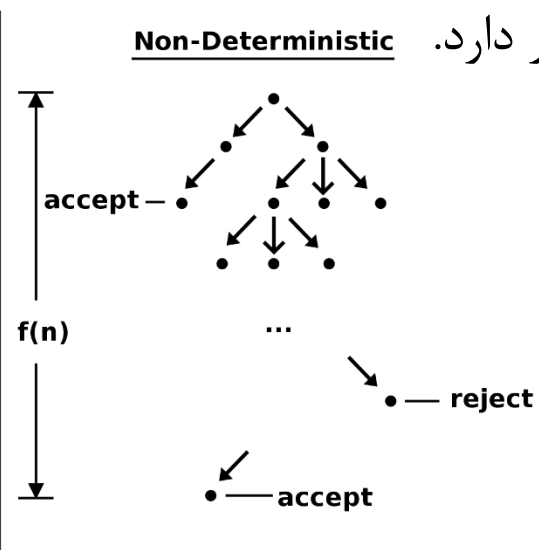
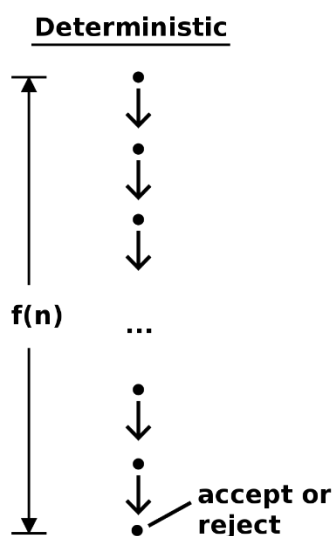
همگی دارای قدرت یکسان با ماشین تورینگ ماشین تورینگ معرفی شده

انواع دیگر ماشین های تورینگ

▪ ماشین تورینگ نامعین $\delta: Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{\rightarrow, \leftarrow\}}$

قضیه هر ماشین تورینگ نامعین، ماشین تورینگ معین متناظر دارد.

-ایده اثبات با استفاده از جستجو سطح اول



الگوریتم

تعریف نادقیق

- مجموعه دستورالعمل‌هایی آسان برای اجرای وظیفه‌ای
- رویه، دستور، روش
- دارای نقش مهم در ریاضی
- از باستان یافتن اعداد اول و کم‌م
- معاصر الگوریتم در مرزهایی فراتر
- با وجود ریشه در ریاضی، عدم تعریف دقیق تا صد سال اخیر
- صرفاً شهودی
- اما ناکافی برای فهم دقیق‌تر الگوریتم‌ها
- مثال - مسئله هیلبرت

مسئله‌های هیلبرت

۱۹۰۰

سخنرانی در کنگره بین‌المللی ریاضی‌دانان در پاریس

معرفی ۲۳ مسئله به مثابه چالش در قرن بعد

مسئله دهم مرتبط با الگوریتم

نیاز به مقدماتی

▪ چندجمله‌ای

- جمع ترم‌هایی که هر ترم حاصلضرب متغیرهایی و ثابتی (ضریبی) است $6x^3y^5z^9$
- ریشه چندجمله‌ای: یافتن مقداری برای متغیرها که چندجمله‌ای برابر صفر شود.
- ریشه صحیح: تمامی مقداری متغیرها جهت یافتن ریشه برابر عدد صحیح باشد

مسئله دهم هیلبرت

چندجمله‌ای

- جمع ترم‌هایی که هر ترم حاصلضرب متغیرهایی و ثابتی (ضریبی) است $6x^3y^5z^9$
- ریشه چندجمله‌ای: یافتن مقداری برای متغیرها که چندجمله‌ای برابر صفر شود.
- ریشه صحیح: تمامی مقداری متغیرها جهت یافتن ریشه برابر عدد صحیح باشد

مسئله دهم هیلبرت به دنبال الگوریتمی جهت آزمایش اینکه چندجمله‌ای ریشه صحیح دارد یا نه

- عدم استفاده از اصطلاح «الگوریتم»
- «فرایندی یا رویه‌ای کشف کرد که با آن بتوان با تعداد اجرای متناهی از عملیات [دارا بودن ریشه متناهی] مشخص شود»
- استفاده از «کشف» به معنای فرض وجود و صرفاً تلاش برای یافتن آن

چنین الگوریتمی وجود ندارد

- «الگوریتم‌ها حل‌ناپذیر»
- شهود شاید برای اینکه الگوریتمی وجود دارد کافی باشد ولی ناکافی و بی‌فایده برای اینکه الگوریتمی وجود ندارد
- اثبات اینکه الگوریتمی وجود ندارد نیاز به تعریف دقیق الگوریتم
- پیشبرد مسئله دهم بسته به تعریف

مسئله دهم هیلبرت

- برآمدن تعریف در ۱۹۳۶
- مقاله‌های الززو چرچ و الن تورینگ
- چرچ: جبر لاند
- تورینگ: ماشین‌ها
- تناظر دو تعریف
- تز چرچ-تورینگ - اتصال خجسته بین تعریف نادقیق و تعریف دقیق
- فراهم‌کننده تعریف الگوریتم لازم برای حل مسئله دهم هیلبرت
- ۱۹۴۰ یوری ماتياسویچ بر اساس کار دیویس و پاتنم و رایینسن اثبات کرد که الگوریتمی وجود ندارد که بتواند ریشه صحیح داشتن چندجمله‌ای را آزمایش کند.

تعریف الگوریتم

تابع $f: D \rightarrow R$ ، ماشین تورینگ M است که با داشتن ورودی $d \in D$ روی نوار ماشین مذکور، در نهایت با پاسخ صحیح $f(d) \in R$ روی نوار متوقف شود. به دیگر سخن

$$\forall d \in D: q_0 d \vdash_M^* q_p f(d), q_p \in F$$

تز چرچ-تورینگ-

▪ فهم شهودی الگوریتم‌ها متناظر با الگوریتم‌های ماشین تورینگ

تز چرچ-تورینگ

از زبان چرچ:

▪ «هیچ رویهٔ رایانش‌پذیری (محاسباتی) را الگوریتم در نظر نمی‌توان گرفت مگر اینکه با ماشین تورینگ نمایش داده شود.»

از زبان تورینگ

▪ «تابعی به طور کارآمدی رایانش‌پذیر است اگر مقادیر آن با فرایندی کاملاً مکانیکی اجرا شود.»

هر نوع رایانش که مکانیکی انجام‌پذیر باشد، با ماشین تورینگ نیز انجام‌پذیر است.

قضیه

مجموعه ماشین‌های تورینگ شمارای نامتناهی است.

نتیجه- زبان‌هایی وجود دارد که نمی‌توان آنها را نمایش داد.

نتیجه- بعضی مسائل الگوریتم ندارند.

پیچیدگی رایانشی

عملی بودن مسائل قابل حل با وجود تصمیم‌پذیری
نادیده‌گیری جزئیات سخت‌افزاری و ساختار داده‌ای و پیاده‌سازی
اندازه‌گیری کارایی با زمان و مکان (فضا)

پیچیدگی زمانی

- معیار مدت زمان مورد نیاز رایانش خاص
- وابسته به اندازه ورودی
- مثال مرتب‌سازی اعداد صحیح
- چشم‌پوشی از جزئیات و معیارهای دیگر
- فرض استفاده از ماشین تورینگ
- هر حرکت در واحد زمانی
- زمان محاسبه برابر با تعداد حرکت‌های انجام شده
- در اینجا صرفاً بدترین حالت

پیچیدگی زمانی

تعریف پیچیدگی زمانی (زمان اجرا)

- ماشین تورینگ معین که روی تمام ورودی‌ها توقف کند را فرض می‌گیریم. پیچیدگی زمانی تابع $f: N \rightarrow N$ است که $f(n)$ بیشترین تعداد گام‌ها و حرکتهایی است که ماشین روی ورودی به طول n صرف می‌کند.
- می‌گوییم ماشین در زمان $f(n)$ اجرا می‌شود
- ماشین تورینگ زمان $f(n)$

پیچیدگی زمانی

تأثیر مدل ماشین تورینگ بر پیچیدگی زمانی
 $a^n b^n$ ▪

▪ ماشین تورینگ تک‌نواره $O(n^2)$

▪ ماشین تورینگ دونواره $O(n)$

▪ تمایز بین رایانش پذیری و پیچیدگی

▪ تز چرچ-تورینگ تمامی مدل‌های رایانش هم‌ارزند.

▪ در نظریه پیچیدگی تأثیر انتخاب مدل بر پیچیدگی زمانی زبان‌ها

▪ نظریه پیچیدگی: رده‌بندی زبان‌ها با توجه به پیچیدگی زمانی آن‌ها

▪ اما هر زبان امکان دارا بودن زمان اجرای متفاوت روی مدل‌های رایانش متفاوت

▪ عدم تأثیر آنچنان مدل‌های معین معمول روی زمان اجرا

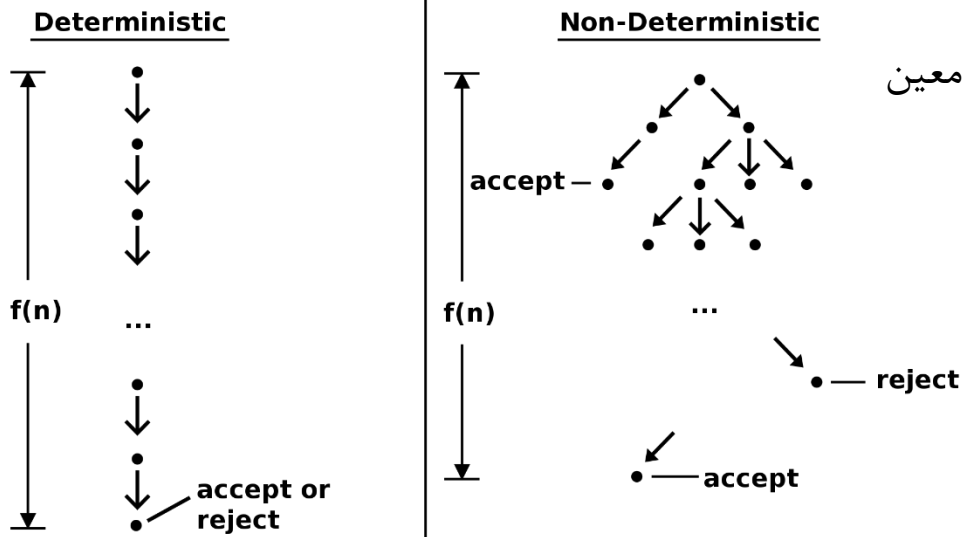
▪ اما ماجرادر نامعینی جداست!

پیچیدگی زمانی

قضیه- اگر ماشین تورینگ دو نواره رایانشی را در n حرکت انجام دهد، آن گاه رایانش مذکور در ماشین تورینگ استاندارد در $O(n^2)$ حرکت انجام خواهد شد.

قضیه- اگر ماشین تورینگ نامعین رایانشی را در n حرکت انجام دهد، آن گاه رایانش مذکور در ماشین تورینگ استاندارد در $O(k^{an})$ حرکت انجام خواهد شد.

▪ اثبات بر اساس شبیه‌سازی خام و ابتدایی، و امید به بهبود زمان روی ماشین معین



پیچیدگی زمانی

مسئله صدق پذیری SAT

▪ متغیر بولی (منطقی)

$$x_i = \begin{cases} 0 \\ 1 \end{cases} \text{ n متغیر بولی}$$

$$s_i = \begin{cases} x_i \\ \neg x_i \end{cases}$$

$$t_j = s_1 \vee s_2 \vee \dots \vee s_p$$

$$e = t_1 \wedge t_2 \wedge \dots \wedge t_k$$

▪ معروف به صورت نرمال ترکیب عطفی CNF

▪ تشخیص صدق پذیری

▪ روش معین: نیاز به بررسی 2^n پیچیدگی زمانی نمایی

▪ روش نامعین: نیاز به بررسی $O(n)$ پیچیدگی زمانی خطی

▪ اهمیت معین یا نامعین بودن مدل و نوع ماشین تورینگ

رده‌های پیچیدگی P و NP

تعریف P یا «چ»

$$P = \bigcup_{i \geq 1} DTIME(n^i)$$

رده P شامل همه زبان‌هایی که در ماشین تورینگ معین در زمان چندجمله‌ای تصمیم‌پذیر باشد.

▪ شامل همه مدل‌های رایانشی چندجمله‌ای هم‌ارز ماشین تورینگ تک‌نواره

▪ تقریباً معادل تمامی روش‌های حل‌پذیر در عالم واقع!

تعریف NP یا «چن»

$$NP = \bigcup_{i \geq 1} NTIME(n^i)$$

رده NP شامل همه زبان‌هایی که در ماشین تورینگ نامعین در زمان چندجمله‌ای تصمیم‌پذیر باشد.

رده‌های پیچیدگی P و NP

تعریف P یا «چ»

$$P = \bigcup_{i \geq 1} DTIME(n^i)$$

- رده P شامل همهٔ زبان‌هایی که در ماشین تورینگ معین در زمان چندجمله‌ای تصمیم‌پذیر باشد.
- شامل همه مدل‌های رایانشی چندجمله‌ای هم‌ارز ماشین تورینگ نک‌نواره
 - تقریباً معادل تمامی روش‌های حل‌پذیر در عالم واقع!

تعریف NP یا «چن»

$$NP = \bigcup_{i \geq 1} NTIME(n^i)$$

ردهٔ NP شامل همهٔ زبان‌هایی که در ماشین تورینگ نامعین در زمان چندجمله‌ای تصمیم‌پذیر باشد.

$$P \subseteq NP$$

- پرسش: آیا محض است؟ نمی‌دانیم.

رده‌های پیچیدگی P و NP

مثال‌هایی از رده P یا «چ»

- یافتن مسیر در گراف جهت‌دار
- اول نسبی بودن دو عدد طبیعی
- عضویت در زبان مستقل از متن

مثال‌هایی از رده NP یا «چن»

- دور همیلتنی
- هرچند دارای ویژگی اعتبارسنجی پاسخ چندجمله‌ای
- Polynomial verifiability
- تأیید درستی وجود دور همیلتنی از مشخص کردن وجودش سخت‌تر

رده پیچیدگی NP و اعتبارسنجی

اعتبارسنجی زبان

- الگوریتمی است که بتواند در زمان چندجمله‌ای بررسی درستی راه‌حل موجودی از نمونه مسئله را انجام دهد.

تعریف دوم NP

- طبقه از زبان‌هایی که دارای اعتبارسنجی چندجمله‌ای هستند.

قضیه- زبانی NP است اگر و فقط اگر با ماشین تورینگ نامعین در زمان چندجمله‌ای تصمیم گرفته شود.

نتایج- زبان‌هایی وجود دارند که دارای اعتبارسنجی زمان چندجمله‌ای نیستند.

- مشهور به coNP

- مثال- عدم وجود دور همیلتنی در گراف

$$P \subseteq coNP$$

پرسش بعدی $coNP = ? NP$

- نمی‌دانیم

رده‌های پیچیدگی NP و coNP

مثال‌هایی از رده NP یا «چن»

- کلیک در گراف بی جهت
- زیرگرافی کامل
- قضیه - $NP \in \text{کلیک-}k$.
- قضیه - $NP \in \text{جمع زیرمجموعه}$.

مثال‌هایی از رده coNP

- عدم وجود کلیک
- عدم وجود جمع زیرمجموعه
- تأیید عدم حاضر و موجود بودن بسیار سخت‌تر

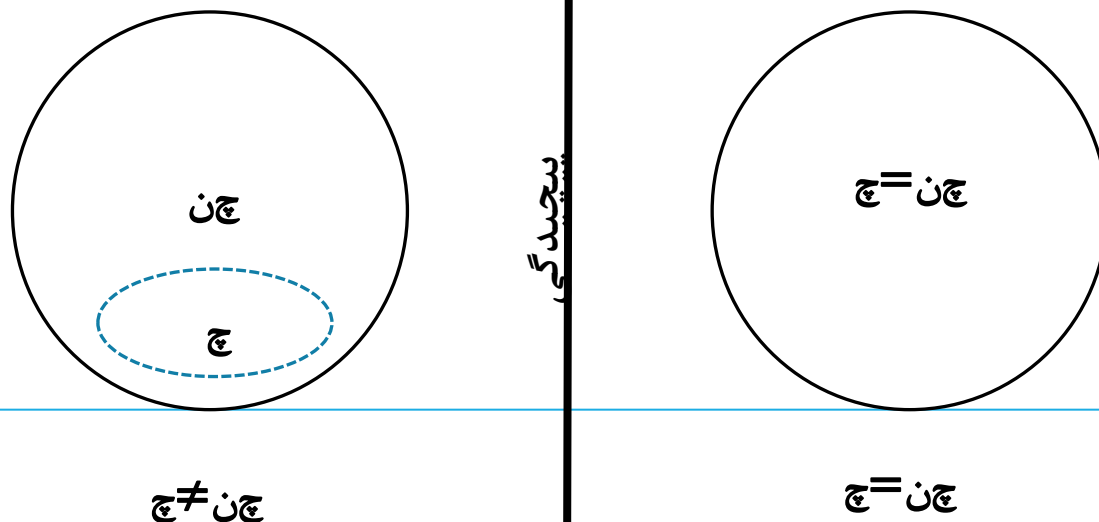
پرسش رده‌های پیچیدگی P و NP

چن یا NP

- رده زبان‌های حل‌پذیر زمان چندجمله‌ای روی ماشین تورینگ نامعین
- هم‌ارز: رده زبان‌های دارای تایید عضویت در زبان با زمان چندجمله‌ای

چن: رده زبان‌هایی که عضویت به سرعت تأیید گرفته می‌شود.

چ: رده زبان‌های که عضویت به سرعت تصمیم گرفته می‌شود.



پرسش رده‌های پیچیدگی P و NP

بهترین روش معین جهت تصمیم‌گیری NP از مرتبه زمان نمایی، یا

$$NP \subseteq EXPTIME = \bigcup_i DTIME(2^{n^i})$$

اما عدم اطلاع از زمان معین کوچکتر برای چن

منجر به یافتن سخت‌ترین مسائل در چن

▪ در صورت یافتن پاسخی چندجمله‌ای معین برای یکی از آنها

▪ تحویل کل رده «چن» به «چ»

▪ معروف به مسائل NP-کامل NP-complete

کاهش پذیری

اگر $B \leq A$ چ آن گاه A به B کاهش می یابد

- اگر بتوان راه حلی برای مسئله B را برای حل مسئله A به کار برد.
- کاهش پذیری چند جمله ای تبدیل را بتوان در زمان چند جمله ای داشت.

کاهش پذیری

قضیه صدق-۳ به کلیک در زمان چند جمله‌ای کاهش پذیر است.

یال‌ها

▪ هر جفت راس را به جز دو نوع جفت را به هم وصل می‌شوند.

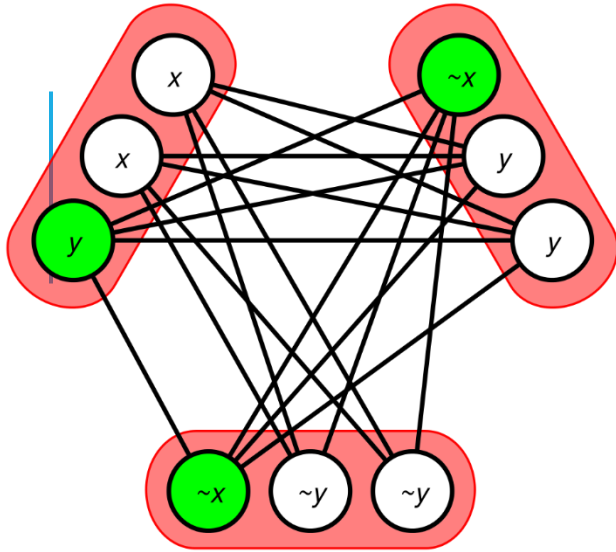
▪ الف- هیچ یالی بین گره‌های هر سه تایی برقرار نیست.

▪ ب- هیچ یالی بین دو گره متمم x و $\bar{x}$ برقرار نیست.

نتیجه: مسئله کلیک در زمان چند جمله‌ای حل پذیر باشد، صدق-۳ نیز حل پذیر خواهد بود.

تحویل پذیری چند جمله‌ای پیچیدگی دو مسئله ظاهراً بی‌ربط را به یکدیگر وصل می‌کند

NP-کامل NP-COMplete



تعریف- مسئله‌ای NP-کامل است اگر

- عضوی از چن باشد
- NP-hard «چن-سخت» باشد.

مسئله A از مسئله B سخت‌تر است، اگر حل آسان A منجر به حل آسان B شود.

تحویل یا کاهش‌پذیری زمان-چندجمله‌ای: $A \leq_{\text{چ}} B$ اگر بتوان راه‌حل مسئله B را برای حل مسئله A به کار برد.

قضیه- اگر: $A \leq_{\text{چ}} B$ و $B \in P$ ، آن‌گاه $A \in P$

▪ قضیه $3SAT \leq_{\text{چ}} \text{Clique}$

$$(x \vee x \vee y) \wedge (\neg x \vee \neg y \vee \neg y) \wedge (\neg x \vee y \vee y)$$

▪ چن-سخت به معنای تحویل چندجمله‌ای تمامی مسائل چن به مسئله مدنظر

NP-کامل NP-COMplete

تعریف- مسئله‌ای NP-کامل است اگر

▪ عضوی از چن باشد

▪ NP-hard «چن-سخت» باشد، یا $\forall L \in NP$ تحویل‌پذیر زمان-چندجمله‌ای به آ» یا $B \leq_{\text{چ}} L$ $\forall L \in NP$

NP-کامل NP-COMPLETE

تعریف- مسئله‌ای NP-کامل است اگر

▪ عضوی از چن باشد

▪ NP-hard «چن-سخت» باشد، یا $\forall L \in NP$ تحویل‌پذیر زمان-چندجمله‌ای به آ» یا $B \leq_{\text{چ}} L$ $\forall L \in NP$

پرسش: اولین مسئله NP-کامل؟

NP-کامل NP-COMplete

تعریف- مسئله‌ای NP-کامل است اگر

- عضوی از چن باشد

- NP-hard «چن-سخت» باشد، یا $\forall L \in NP$ تحویل‌پذیر زمان-چندجمله‌ای به آ» یا $B \leq_{\text{چ}} L$ $\forall L \in NP$

پرسش: اولین مسئله NP-کامل

- صدق‌پذیری SAT

NP-کامل NP-COMPLETE

تعریف - مسئله‌ای NP-کامل است اگر

- عضوی از چن باشد

- NP-hard «چن-سخت» باشد، یا $\forall L \in NP$ تحویل‌پذیر زمان-چندجمله‌ای به آ «یا $B \leq L$ چ $\forall L \in NP$ »

پرسش: اولین مسئله NP-کامل

- صدق‌پذیری SAT

قضیه کوک-لوین - $NP-C \in$ صدق‌پذیری.

NP-کامل NP-COMplete

تعریف - مسئله‌ای NP-کامل است اگر

▪ عضوی از چن باشد

▪ NP-hard «چن-سخت» باشد، یا $\forall L \in NP$ تحویل‌پذیر زمان-چندجمله‌ای به آ «یا B چ $L \leq$ $\forall L \in NP$ »

پرسش: اولین مسئله NP-کامل

▪ صدق‌پذیری SAT

قضیه کوک-لوین - $NP-C \in$ صدق‌پذیری.

قضیه - $P \in$ صدق‌پذیری اگر و فقط اگر $P = NP$.

NP-COMPLETE کامل

قضیه کوک-لویین - $NP-C \in$ صدق پذیری.

NP-کامل NP-COMPLETE

قضیه کوک-لوین - $NP-C \in$ صدق پذیری.

نتیجه - $3SAT \in NP-C$.

NP-کامل NP-COMPLETE

قضیه کوک-لوین - $NP-C \in$ صدق پذیری.

نتیجه - $SAT \in NP-C$.

نتیجه - $Clique \in NP-C$.

NP-کامل NP-COMPLETE

قضیه کوک-لوین - $NP-C \in$ صدق پذیری.

نتیجه - $3SAT \in NP-C$.

نتیجه - $Clique \in NP-C$.

قضیه - همیلتنی $چ \leq 3SAT$.

NP-کامل NP-COMPLETE

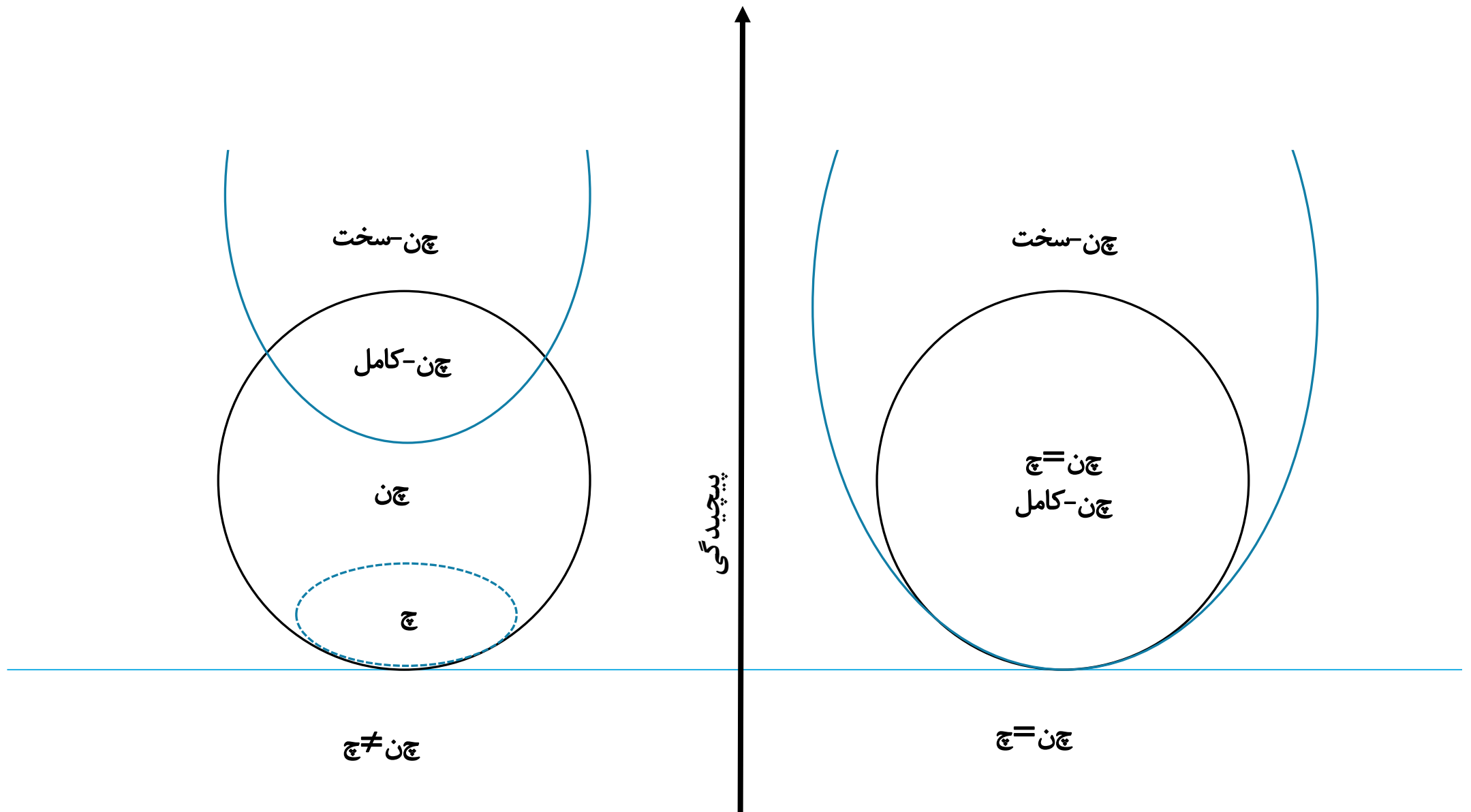
قضیه کوک-لوین - $NP-C \in$ صدق پذیری.

نتیجه - $3SAT \in NP-C$.

نتیجه - $Clique \in NP-C$.

قضیه - همیلتنی $\leq 3SAT$.

▪ نتیجه - $NP-C \in$ همیلتنی.



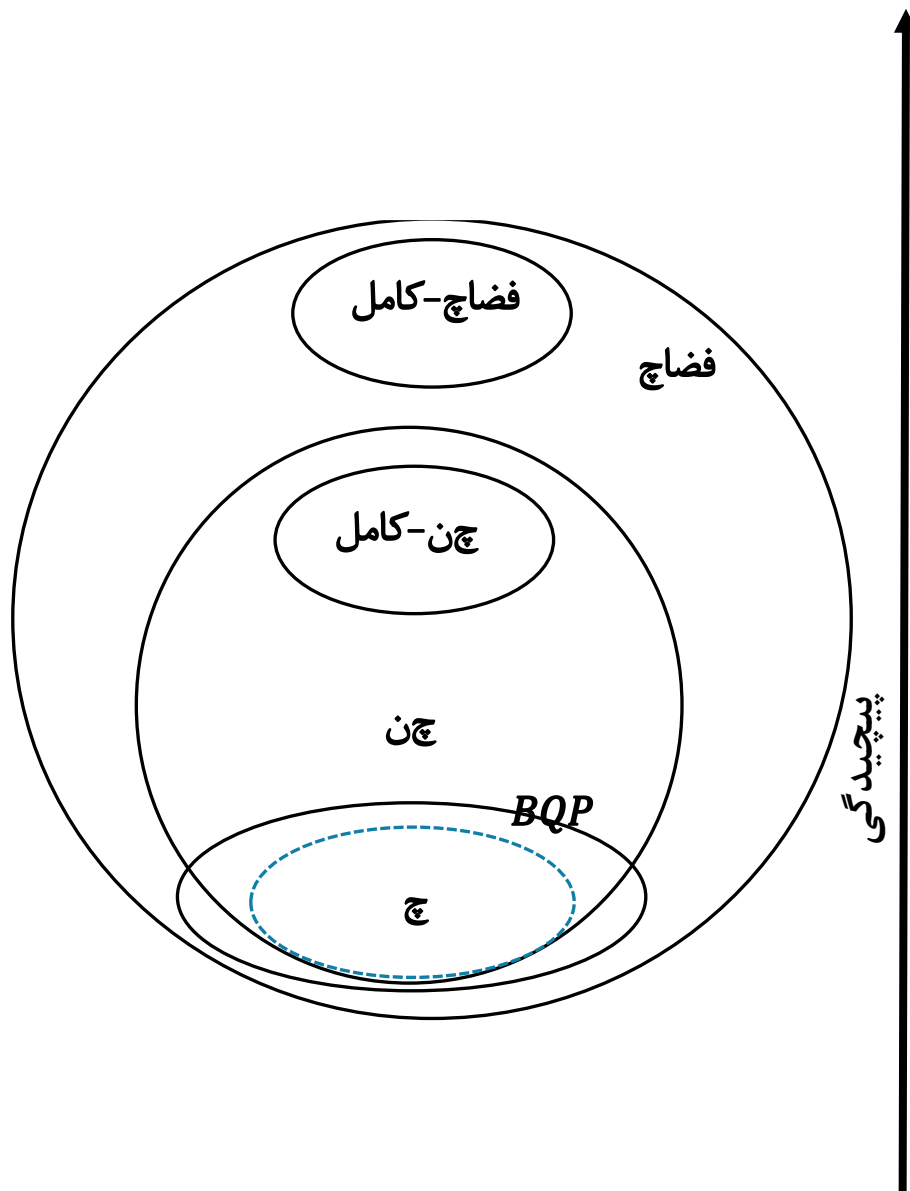
فضاچ PSPACE

قضیه - $PSPACE = NPSPACE$.

فضاچ-کامل

▪ معادله بولی تسویردار

▪ مشاعره



جمع بندی

آنچه می دانیم

$$L \subseteq NL \subseteq P \subseteq BPP \subset NP \subseteq PSPACE = NPSPACE$$

$$P \subseteq coNP$$
$$L \neq PSPACE$$

چند مورد حدسیات

$$P \subset NP$$
$$P = BPP$$
$$NP \neq coNP$$

جمع‌بندی

▪ BQP

▪ رده مسائل تصمیم‌حل‌پذیر با رایانه کوانتومی در زمان چندجمله‌ای با احتمال خطای $\frac{1}{3}$ برای تمامی نمونه‌ها

▪ PH

▪ رده پیچیدگی تمامی رده‌های پیچیدگی در سلسله مراتب چندجمله‌ای از NP, P ، $coNP$

▪ QMA

▪ وجود برهان اندازه-چندجمله‌ای متقاعدکننده اعتبارسنج کوانتومی با زمان چندجمله‌ای با احتمال بالا

▪ PP

▪ رده پیچیدگی تمامی مسائل تصمیم‌پذیر حل‌پذیر با ماشین تورینگ احتمالی در زمان چندجمله‌ای با احتمال خطای کمتر از $\frac{1}{3}$ برای تمامی نمونه‌ها.

آنچه می‌دانیم

$$L \subseteq NL \subseteq P \subseteq NP \subseteq PH \subseteq PP \subseteq PSPACE = NPSPACE$$

$$P \subseteq coNP$$

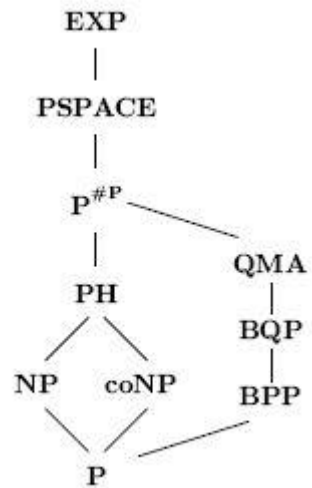
$$P \subseteq BPP \subseteq BQP \subseteq QMA \subseteq PP$$

$$L \neq PSPACE$$

$$P \subset NP$$

$$P = BPP$$

$$NP \neq coNP$$



چند مورد «حدسیات»

مراجع

[سیپسر] M. Sipser, “Introduction to the theory of computation,” Cengage Learning, 3rd ed, 2013

[ارورا] M. Arora, “Introduction to the theory of computation,” Cengage Learning, 3rd ed, 2013

[لینز] P. Linz, “An Introduction to Formal languages and automata,” 5th ed, 2012

[روزن] K Rosen, “Discrete Mathematics and its Applications,” McGraw Hill, 7th ed, 2012

[دیویس] M. Davis, “Computability and Unsolvability,” McGraw Hill, 1958

[مانوچپی]

[آرنسن]

تعریف خودکاره متناهی

پنج تائی $(Q, \Sigma, \delta, q_0, F)$ است به طوری که

- Q مجموعه متناهی حالت‌ها
- Σ الفباء ورودی (متناهی)
- $\delta: Q \times \Sigma \rightarrow Q$ تابع انتقال
- با هر زوج حالت فعلی و ورودی فعلی، حالت جدید
- $q_0 \in Q$ حالت آغاز
- $F \subseteq Q$ مجموعه حالت‌های پذیرش (نهائی)

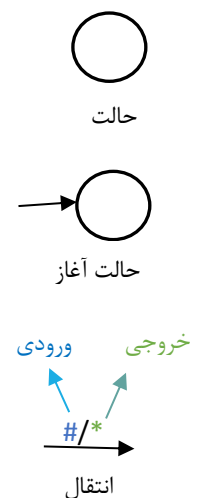
تابع انتقال جهت تعریف قواعد حرکت یا انتقال بین حالت‌ها

اگر در حالت الف باشیم و فلشی از حالت الف به حالت ب با مقدار یک روی فلش باشد. یعنی با دریافت ورودی یک حالت از الف به ب منتقل می‌شود.

$$\delta(\text{الف}, 1) = \text{ب}$$

نمودار انتقال نمایش هم‌ارز آن

زبان منظم: زبانی که ختم رشته‌های آن را بپذیرد



لم تزریق - قضیه

اگر L زبانی منظم باشد، آن گاه عدد طبیعی p وجود خواهد داشت که برای هر رشته s از L با طول حداقل p

$$s \geq |p|$$

▪ شامل سه زیررشته x و y و z

▪ $s = xyz$ با ویژگی‌های زیر

▪ الف) $\forall i \geq 0, i \in \mathbb{Z}: xy^i z \in L$

▪ ب) $|y| > 0$

▪ ج) $|xy| \leq p$

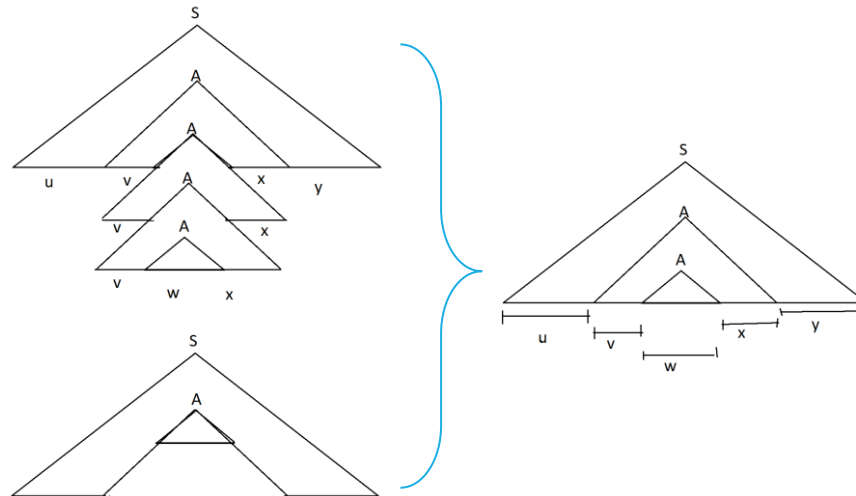
لم تزریق مام

L زمام است. آن گاه عدد ثابت n صرفا وابسته به L وجود دارد که تمامی رشته‌های z در L و $|z| \geq n$ از زیررشته‌های $uvwxy$ تشکیل شده‌اند به طوری که

الف $z = uvwxy$

ب $|vx| \geq 1, |vwx| \leq n$

ج $\forall i, i \geq 0: uv^iwx^iy \in L$



اختصارات

خ م: خودکاره متناهی

خ م م: خودکاره متناهی معین

خ م ن: خودکاره متناهی نامعین

ع ب: عبارت منظم

ص ب ن: صورت بکوس-ناتور

م ا م: مستقل از متن

Σ : مجموعه متناهی حروف الفباء دستور

BNF: صورت بکوس-ناتور

R: مجموعه قوانین تولید دستور

S: متغیر آغاز دستور

V: مجموعه متناهی متغیرهای دستور