

نظریه زبان ها و ماشین

دانشگاه آل طه

رشته کامپیوتر

درس تخصصی

پاییز 1404

رئوس مطالب

□ هدف درس:

- معرفی مفاهیم پایه در نظریه محاسبات
- معرفی مدل های صوری مختلف در نظریه محاسبات
- روشن کردن محدودیت های مدل های مختلف محاسباتی

□ سرفصل:

- اتوماتای متناهی ، زبان های منظم و گرامر های منظم
- زبان ها و گرامر های مستقل از متن و اتومانای پشته ای
- زبان ها و گرامر های حساس به متن
- ماشین تورینگ و گرامر نامقید
- تصمیم پذیری
- پیچیدگی محاسباتی

جلسه اول

□ چی یاد میگیریم؟

۱. تعریف «زبان» از نظر کامپیوتر

۲. مدل سازی «ماشین‌هایی» که این زبان‌ها رو می‌فهمن

۳. مرز توانایی محاسبه – یعنی چه چیزهایی قابل حل هستن و چه چیزهایی نه

آلن تورینگ و آلونسو چرچ

■ سوال مهم

□ آیا هر مسئله‌ای با ماشین قابل حل است یا نه؟

□ ماشین تورینگ

یک مدل ریاضی محاسبات است که یک ماشین انتزاعی را توصیف می‌کند که نمادها را روی یک نوار طبق جدول قوانین دستکاری می‌کند.

با وجود سادگی مدل، قادر به اجرای هر الگوریتم کامپیوتری است.

مقدمات ریاضی

تعریف مجموعه:

دسته ای از اعضا که به جز عضویت دارای هیچ ساختاری نیستند.

$$S = \{0,1,2\}$$

$$S = \{i : i > 0, i \text{ is even}\}$$

مجموعه جهانی: تمامی عناصر ممکن

مجموعه تهی: هیچ عضوی نداشته $\emptyset$ یا $\{\}$

عملیات روی مجموعه ها

$$S_1 \cup S_2 = \{x : x \in S_1 \text{ or } x \in S_2\},$$

$$S_1 \cap S_2 = \{x : x \in S_1 \text{ and } x \in S_2\},$$

$$S_1 - S_2 = \{x : x \in S_1 \text{ and } x \notin S_2\}.$$

$$\bar{S} = \{x : x \in U, \ x \notin S\}$$

زیر مجموعه و زیر مجموعه محض

یک مجموعه S_1 زیر مجموعه S گفته می‌شود، اگر تمامی عناصر S_1 همگی در S موجود باشند. ما این گونه می‌نویسیم:

$$S_1 \subseteq S.$$

اگر $S_1 \subseteq S$ اما S دارای عنصری بود که در S_1 وجود نداشت، آن گاه گوییم S_1 یک زیر مجموعه محض از S است و می‌نویسیم:

$$S_1 \subset S.$$

خصوصیات مجموعه ها

$$۱) A \subset B \Rightarrow A \cap B = A, A \cup B = B$$

$$۲) A \cap B \subset A \cup B$$

$$۳) B \subset A \cup B, A \subset A \cup B$$

$$۴) A \cap B \subset B, A \cap B \subset A$$

$$۵) A \cap M = A, A \cup N = M \quad (M \text{ مجموعه‌ای مرجع است})$$

$$۶) A \cap \phi = \phi, A \cup \phi = A$$

$$۷) A - B = A \cap B'$$

$$۸) A \cap B = B \cap A, A \cup B = B \cup A$$

$$۹) A \cap (B \cap C) = (A \cap B) \cap C, A \cup (B \cup C) = (A \cup B) \cup C$$

$$۱۰) A \cap A' = \phi$$

$$۱۱) A \cup A' = M$$

$$۱۲) \phi \cup M = M, \phi \supset M = \phi$$

□ قانون دمورگان

$$\overline{S_1 \cup S_2} = \overline{S_1} \cap \overline{S_2},$$

$$\overline{S_1 \cap S_2} = \overline{S_1} \cup \overline{S_2},$$

- مجموعه متناهی: شامل تعداد محدودی از عناصر باشد.
- مجموعه نامتناهی: متناهی نباشد
- اندازه یک مجموعه متناهی: تعداد اعضای موجود در آن $||$
- مجموعه توانی: مجموعه تمامی زیر مجموعه های مجموعه

مثال ۱-۱: اگر مجموعه S ، $\{a, b, c\}$ باشد، آنگاه مجموعه توانی آن اینگونه است:

$$2^S = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}.$$

در اینجا $|S| = 3$ و $|2^S| = 8$. این از یک نتیجه کلی حاصل می شود، اگر S مجموعه ای متناهی باشد،
در این صورت: $|2^S| = 2^{|S|}$

زوج مرتب

اگر x و y دو شیء باشند، زوج مرتب x و y را به صورت (x, y) نشان می‌دهند. در این حالت، عنصر اول یعنی x را مولفه اول و عنصر دوم یعنی y را مولفه دوم می‌نامند. در زوج مرتب، ترتیب قرار گیری عناصر در مولفه‌ها مهم است.

نکته: جابجایی مولفه‌ها، باعث ایجاد زوج مرتب جدید می‌شود. به این ترتیب برای دو زوج مرتب (x, y) و (y, x) لزوماً تساوی برقرار نیست. در نتیجه خواهیم داشت:

$$(x, y) \neq (y, x)$$

ضرب دکارتی

$$A \times B = \{(a, b) \mid a \in A \text{ and } b \in B\}$$

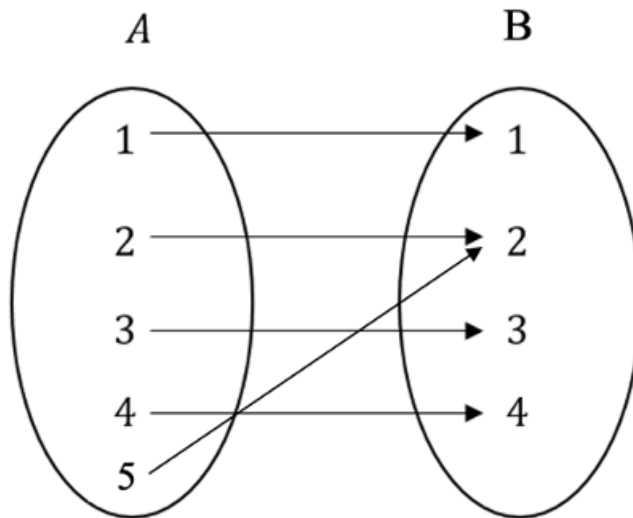
برای مثال فرض کنید که مجموعه $A = \{x, y, z\}$ به صورت A و مجموعه $B = \{1, 2, 3\}$ نیز به صورت B تعریف شده باشند. منظور از $A \times B$ مجموعه‌ای به صورت زیر است.

$$A \times B = \{(x, 1), (x, 2), (x, 3), (y, 1), (y, 2), (y, 3), (z, 1), (z, 2), (z, 3)\}$$

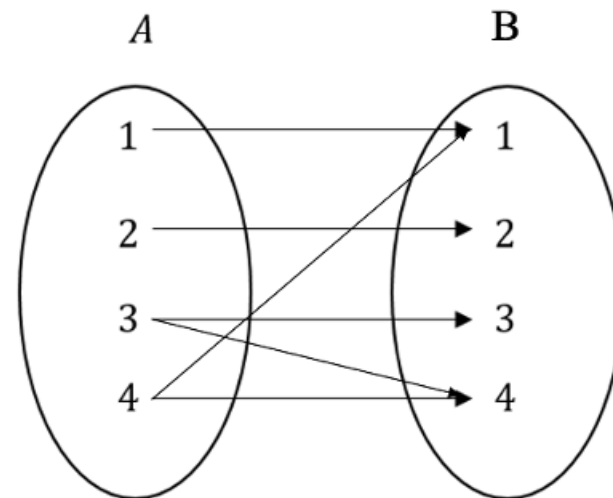
واضحا ضرب دکارتی خاصیت جابجایی ندارد.

توابع و روابط

تابع: قانون یا ضابطه ای که عناصر یک مجموعه را بصورت یکتا به عناصر مجموعه دیگر مربوط می سازد.



تابع است



تابع نیست

مجموعه اول دامنه و مجموعه دوم برد □

$$f: S_1 \rightarrow S_2$$

که بیانگر این است که دامنه f یک زیر مجموعه S_1 و برد آن یک زیر مجموعه از S_2 می باشد. اگر دامنه تابع f همه S_1 باشد، گوییم f یک تابع کلی روی S_1 است، در غیر این صورت f را تابع جزئی گویند.

سه مفهوم اساسی

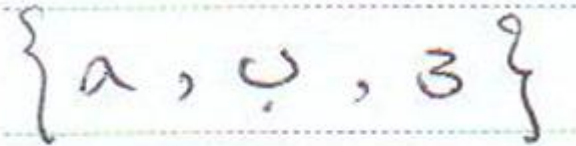
زبان ☐

گرامر ☐

ماشین ☐

تعریف الفبا

- تعریف نماد: یک شی غیرقابل تجزیه که معمولا همان علامت های روی کیبرد هستند. (علائم و نشانه های مجزا)
- الفبا مجموعه ای متناهی و غیرتهی از نمادهاست.

A handwritten representation of the set notation {a, b, c} on a piece of lined paper. The characters are written in a cursive style, with 'a' in black, 'b' in blue, and 'c' in red. The set is enclosed in curly braces.

$\Sigma = \{a, b, c\}$

□ تعریف رشته: دنباله متناهی از حروف الفبا

□ $w_1 = \text{bacc}$

□ $W_2 = \text{ca}$

□ طول رشته w را با $|w|$ نشان میدهیم.

□ $|W_2| = 2$

□ $|W_1| = 4$

رشته با طول صفر را رشته تهی می نامیم و با ϵ (یا λ) نمایش می دهیم. پس داریم: $|\epsilon| = 0$

عملیات روی رشته ها

concatenation اتصال □

$$w_1 = \text{bacc}$$

$$w_2 = \text{ca}$$

$$w_1 \circ w_2 = \text{baccaca}$$

$$w \circ \varepsilon = \varepsilon \circ w = w$$

$$w \circ w = w^2$$

$$w^n = \underbrace{w \circ w \circ \dots \circ w}_{n \text{ times}} = w \circ w^{n-1}$$

$$w^0 = \varepsilon$$

عملیات روی رشته ها

معکوس Reverse □

$$w_1 = bacc$$

$$w_2 = ca$$

$$w_1^R = ccab \quad - \quad w_2^R = ac$$

$$\varepsilon^R = \varepsilon$$

$$(u.v)^R = v^R.u^R$$

Σ^+ و Σ^*

اگر Σ یک الفبا باشد، آنگاه از Σ^* برای نمایش مجموعه رشته‌هایی که از اتصال صفر یا بیشتر از نمادهای Σ حاصل می‌شود، استفاده می‌شود. مجموعه Σ^* همیشه شامل λ است. برای خارج کردن رشته نهی تعریف می‌کنیم:

$$\Sigma^+ = \Sigma^* - \{\lambda\}.$$

درحالی‌که Σ بفرض محدود است، Σ^* و Σ^+ همیشه نامتناهی هستند، زیرا هیچ محدودیتی روی طول رشته‌ها در این مجموعه‌ها وجود ندارد.

زبان

□ تعریف زبان: زبان مجموعه ای از رشته های ساخته شده از الفبای Σ است.

□ پس میتوان گفت زیر مجموعه ای از Σ^* است.

□ $\Sigma = \{a, b, c\}$

□ $L_1 = \{ab, b\}$

□ $L_2 = \{\epsilon, bc, cc\}$

□ $L_3 = \{a, aa, aaa, aaaa, \dots\}$

□ زبان یک مجموعه است پس عملیات روی مجموعه ها مانند اجتماع، اشتراک، تفاضل را میتوانیم روی آن ها انجام دهیم.

اتصال زبان ها

□ علاوه بر اجتماع، اشتراک، تفاضل عملیات دیگری هم روی زبان ها تعریف می شود.

$$L^2 = L \circ L$$

$$L_2 = \{\epsilon, bc, cc\}$$

$$L_2^2 = \{bc, cc, bc bc, bc cc, \epsilon, cc bc, cc cc\}$$

$$L^n = L \circ L^{n-1}$$

$$L^0 = \{\epsilon\}$$

متمم زبان

- $\bar{L} = \Sigma^* - L$

$$L_2 = \{ \varepsilon, bc, cc \}$$

$$\Sigma^* = \{ \varepsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, \dots \}$$

$$\bar{L}_2 = \{ a, b, c, aa, ab, ac, ba, bb, ca, cb, \dots \}$$

زبان ها

- $L^- = \Sigma^* - L$
- $L^R = \{W^R : W \in L\}$
- $L_1 L_2 = \{xy : x \in L_1, y \in L_2\}$
- $L^0 = \{\lambda\}, L^1 = L$
- $L^* = L^0 \cup L^1 \cup L^2 \dots$
- $L^+ = L^1 \cup L^2 \dots$

زبان ها

□ $L^* = L^0 \cup L^1 \cup L^2 \dots$

□ بستار ستاره ای star clouser:

□ اتصال 0 یا بیشتر رشته های L

□ $L^+ = L^1 \cup L^2 \dots$

□ بستار بعلاوه positive clouser:

□ اتصال 1 یا بیشتر رشته های L

عبارت منظم (Regular Expression)

□ روشی برای توصیف یک زبان منظم است

□ عبارت منظم الگویی است که مشخص می کند چه رشته هایی به یک زبان تعلق دارند.

1. تهی ، اپسیلون و هر حرف تنهایی مثل a یک عبارت منظم است.

2. اگر r_1 و r_2 عبارت منظم باشند آنگاه عبارت های زیر هم منظم هستند.

$$r_1 + r_2 \quad r_1 \cdot r_2 \quad r_1^* \quad (r_1)$$

3. عباراتی منظم هستند که از ترکیب قوانین اول و دوم به دست بیاد.

4. هیچ عبارت دیگری منظم نیست.

عبارت منظم (Regular Expression)

* تعریف رسمی

اگر Σ (سیگما) یک الفبا باشد (یعنی مجموعه‌ای از نمادها یا حروف مثل $\{a, b\}$),
آنگاه عبارت منظم روی Σ طبق قوانین زیر ساخته می‌شود:

1. $\emptyset$ (مجموعه تهی) یک عبارت منظم است، که زبانی را نشان می‌دهد که هیچ رشته‌ای ندارد.
2. ϵ (ایسیلون) یک عبارت منظم است، و زبانی را نشان می‌دهد که فقط رشته‌ی تهی را دارد.
3. اگر $a \in \Sigma$ باشد، آنگاه a یک عبارت منظم است (زبان $\{a\}$ را نشان می‌دهد).
4. اگر r_1 و r_2 عبارت‌های منظم باشند، آنگاه:
 - $(r_1 \mid r_2)$ یا $(r_1 + r_2)$ یعنی اجتماع زبان‌ها $\rightarrow$ یکی از دو حالت
 - $(r_1 r_2)$ یعنی الحاق (concatenation) $\rightarrow$ پشت سر هم آمدن دو الگو
 - $(r_1)^*$ یعنی بسته‌ی کلین (Kleene star) $\rightarrow$ تکرار صفر یا بیشتر بار از r_1

عبارت منظم (Regular Expression)

مثال □

1. a^* → تمام رشته‌هایی که از صفر یا چند تا a تشکیل شده‌اند:

$\dots, \epsilon, a, aa, aaa$

2. $(a|b)^*$ → تمام رشته‌هایی که فقط از a و b تشکیل شده‌اند:

$\dots, \epsilon, a, b, ab, ba, aab, bba$

3. $abb^*(a|b)$ → همه رشته‌هایی که در آخرشان abb آمده:

$\dots, abb, aabb, babb, aaabb, ababb$

زبان معادل عبارت منظم

$$\emptyset \rightarrow \{\}$$

$$\varepsilon \rightarrow \{\varepsilon\}$$

$$a \rightarrow \{a\}$$

$$r_1 + r_2 \rightarrow L(r_1) \cup L(r_2)$$

$$r_1 \cdot r_2 \rightarrow L(r_1) \cdot L(r_2)$$

$$r_1^+ \rightarrow (L(r_1))^+$$

$$(r_1) \rightarrow L(r_1)$$

گرامر

ابزاری برای توصیف ساختار زبان ها

تعریف ۱-۱: یک گرامر G بصورت چهارتایی زیر تعریف می شود:

$$G = (V, T, S, P),$$

جاییکه V یک مجموعه متناهی از اشیاء به نام متغیرها است،

T یک مجموعه متناهی از اشیاء به نام نمادهای پایانی است،

$S \in V$ یک نماد خاص به نام متغیر شروع است،

P یک مجموعه متناهی از قوانین تولید است.

فرض می شود که مجموعه های V و T غیر تهی و مجزا هستند.

از نماد شروع شروع می کنی و با استفاده از قوانین تولید، مرحله به مرحله جایگزینی می کنی تا فقط از نمادهای نهایی تشکیل بشه.

مثال گرامر

گرامر $G = (\{s\}, \{a,b\}, S, P)$ که p به صورت زیر است:

$$S \rightarrow aSb$$

$$S \rightarrow \lambda$$

جمله تولید شده و شکل جمله ای را به دست آورید.

این گرامر چه زبانی را تولید میکند؟

مثال گرامر

گرامری پیدا کنید که زبان زیر را تولید کند:

$$L = \{a^n b^{n+1} : n \geq 0\}$$

نکته گرامری

دو گرامر G_1 و G_2 معادل هستند اگر زبان یکسانی را تولید نمایند.

$$L(G_1) = L(G_2)$$

یعنی شکل ظاهری آن ها مهم نیست.