A Hierarchy of formal Languages and Automata

Friday, September 23, 2011
10:32 AM

سلسله مراتب زبان‌های رسمی و ماشین‌ها

Accept (پذیرش)

Decide (تصمیم‌گیری)

تعریف: به زبان L شمارش‌پذیر بازگشتی (RE: Recursively Enumerable) گفته می‌شود اگر ماشین تورینگی وجود داشته باشد که آنرا بپذیرد (accept).

تعریف: به زبان L روی الفبای Σ بازگشتی (REC: Recursive) گفته می‌شود اگر ماشین تورینگ M وجود داشته باشد که زبان L را بپذیرد (accept) و روی تمام رشته‌های $w \in \Sigma^+$ متوقف شود (halt). به عبارت دیگر یک زبان بازگشتی است اگر و تنها اگر برای آن یک الگوریتم عضویت داشته باشیم (Membership algorithm).

نکته: فرض کنید S یک مجموعه شمارای نامتناهی باشد آنگاه مجموعه توانی S (2^S) ناشمارا است.

www.amitisweb.com

$\Sigma \qquad \Sigma^*$

2^{Σ^*}

از بین مفاهیم شمارا و ناشمارا در جای REC و RE ؟ می‌شد شمارا و ناشمارا به ترتیب.

از یک مجموعه ناشمارا که زیرمجموعه یک مجموعه شمارا است

قضیه

می‌توان یک زبان RE پیدا کرد که مکمل آن RE نباشد.

(برهان خلف) $L: RE \to$ yes ✓ / no ✗

فرض $\to \bar{L}: RE \to$ yes ✓ / no ✗ $\Rightarrow L: REC$

مثالی از یک زبان RE که REC نباشد: (مکمل اثبات)

$L = \{a^i \mid a^i \in L(M_i)\}$, $\Sigma = \{a\}$

همان بدون کورش است

① $L: RE$ ✓

a^i معادل که ماشین M_i را بپذیرد.

فرض $\bar{L}: Re \to \bar{L} = L(M_k)$

فرض کنیم شماره ماشین k است.

www.amitisweb.com

تناقض

$a^k \in L \Rightarrow a^k \in L(M_k) \Rightarrow a^k \in \bar{L}$

$a^k \notin L \Rightarrow a^k \in \bar{L} \Rightarrow a^k \in L(M_k) \Rightarrow a^k \in L$

تناقض

پس آیا ماشین تورینگی نداریم که بپذیرد.

$\Rightarrow \bar{L}$: RE نیست $\Rightarrow L$: REC نیست

قضیه: اگر یک زبان L و مکمل آن $\bar{L}$ هر دو RE باشند، آنگاه هر دو REC نیز هستند. اگر L بازگشتی باشد، آنگاه $\bar{L}$ هم بازگشتی است. بنابراین هر دو RE نیز هستند. (REC تحت مکمل بسته اند)

قضیه: می‌توان یک زبان RE پیدا کرد که REC نباشد. به عبارت دیگر خانواده زبان‌های REC زیرمجموعه خانواده زبان‌های RE هستند.

مسائل ۶ و ۷ از بخش ۱۱.۱ (به عنوان مثال):

L_1 : RE

L_2 : RE

$L_1 \cup L_2$: RE?

$L_1 \cap L_2$: RE? ← می‌شد این رشته را از هر دو داد.

تمام مسائل بخش ۱-۱۱ مهم هستند.

* می‌شد هر دو را جمع و ند پیش برد.

پس هم $\cup$ و $\cap$ ، Re است.

Unrestricted Grammars: گرامرهای نوع صفر

تنها محدودیت این که سمت چپ λ نباشد

$u \to v$

$u \in (V \cup T)^+$

$v \in (V \cup T)^*$

قضیه: هر زبان تولید شده توسط یک گرامر بدون محدودیت یک زبان Re است و بالعکس.

www.amitisweb.com

فصل ۱-۱۱ : ۱، ۷، ۸، ۹

۱۱-۳ گرامرهای حساس به متن

Context-Sensitive Grammars

$x \to y$

$x, y \in (V \cup T)^+$

$|x| \leq |y|$

تعریف: زبان L حساس به متن است اگر و فقط اگر گرامر حساس به متن G وجود داشته باشد به نحوی که

$L = L(G)$ or $L = L(G) \cup \{\lambda\}$

قضیه: به ازای هر زبان CS که فاقد رشته λ باشد یک LBA وجود دارد و به تناسب آن یک گرامر CS داریم.

$L = \{ww \mid w \in \{a,b\}^*\}$ حساس به متن است.

کوتاهترین رشته این زبان

$S \to c \mid acq_a\square \mid bcq_b\square$

$q_a a \to aq_a$, $q_a b \to bq_a$, $q_b a \to aq_b$, $q_b b \to bq_b$

$q_a\square \to q'a\square \mid a$, $q_b\square \to q'b\square \mid b$

$aq' \to q'a$, $bq' \to q'b$

$cq' \to acq_a \mid bcq_b$

www.amitisweb.com

حساس به متن است چون فضای حافظه‌ای که می‌خواهیم باندازه طول رشته است.

abcab

$S \Rightarrow acq_a\square \Rightarrow acq'a\square \Rightarrow abcq_ba\square \Rightarrow abcaq_b\square \Rightarrow abcab$

$S \to c \mid acq_a\square \mid bcq_b\square \mid aa \mid bcb$, aacaa abcab cbc bb

$q_a a \to aq_a$, $q_a b \to bq_a$, $q_b a \to aq_b$, $q_b b \to bq_b$

$q_a\square \to q'a\square$, $q_b\square \to q'b\square$

$aq' \to q'a$, $bq' \to q'b$

$cq' \to acq_a \mid bcq_b \mid aacq_{aa} \mid abcq_{ab} \mid bacq_{ba} \mid bbcq_{bb}$

$q_{aa} a \to aq_{aa}$, $q_{aa} b \to bq_{aa}$

q_{ab} ---

q_{ba} --

q_{bb} --

$q_{aa}\square \to aa$, $q_{ab}\square \to ab$ ---

c باشد فقط در گام آخر آن را حذف کنیم.

ww ؟

زبان حساس به متن بازگشتی است.

تمرین ۳-۱۱ شماره ص ۱۴۴.

Karzati

www.amitisweb.com

هر زبان حساس به متن بازگشتی است.

$L: CS \Rightarrow G: CS$

$AB \to BA,\ BA \to AB$

الگوریتمی بدهیم که تشخیص دهد آیا رشته حساس به متن است یا نه؟
می‌دانیم آن زبان حساس به متن است چون گرامرش را داریم.

$S \to x_1 \to x_2 \to \dots \to x_n \to w$

$A \to B$ مثل زیر
$B \to A$

$|x_j| \le |x_{j+1}|$

$|x_j| < |x_{j+m}|$ ، $m = f(|V \cup T|, |w|) = \dots$

A,B A → a B → a
a, b, c A → b B → b
A → c B → c
A → B

بعد از یک مدتی بعد از m مرتبه حتماً بزرگتر است یا مساوی هم هست

تکرار می‌کنیم ۱-۱ محدود است.

$w \in L$?

$m \times |w|$

قضیه:
می‌توان یک زبان بازگشتی پیدا کرد که حساس به متن نباشد.

notation ارائه یک

$T = \{a, b\}$

$V = \{V_0, V_1, V_2, \dots\}$

$x_1 \to y_1;\ x_2 \to y_2;\ \dots;\ x_m \to y_m$ $|x_i| \le |y_i|$

تا اینجا حساس به متن است.

ثابت کنید زبانی پیدا کنید که Rec است

$h(a) = 010$
$h(b) = 01^2 0$
$h(\to) = 01^3 0$
$h(;) = 01^4 0$
$h(V_i) = 01^{5+i} 0$

گرامر را می‌سازیم $V_0 \to 0V_0;\ V_0 \to 1V_0;\ V_0 \to 0;\ V_0 \to 1$ $(0+1)^+$

$01^5 0 01^3 0 0 1^5 0 1^4 0 1^5 0$

گرامر را می‌سازیم $V_0 \to 1V_0;\ V_0 \to 1$ 1^+

$01^5 0 01^3 0 010 \dots$

Karghi
Karghi
Karghi
AMITISWEB.COM 0912
www.amitisweb.com

$\in L((011^*0)^*)$ یک گرامر CS، G_i را که کدبندی این خاصیت را دارند

$L = \{ w_i \mid w_i \text{ defines a CS grammar } G_i \text{ and } w_i \notin L(G_i) \}$

① REC ✓ Rec است اما ما سی به قبلیم نیست

گام ۱ - اول ضرب در شده است یا نه که آیا الت Decider ← آیا خروجی Decide شده است یا نه؟

② CS X

فرض: $L : CS \Longrightarrow L = L(G_k) \quad G_k : CS \ \& \ G_k : w_k$

$w_k \in L \Longrightarrow w_k \notin L(G_k) \Longrightarrow w_k \notin L$

$w_k \notin L \Longrightarrow w_k \notin L(G_k) \Longrightarrow w_k \in L$

$\Longrightarrow L : CS \ X$

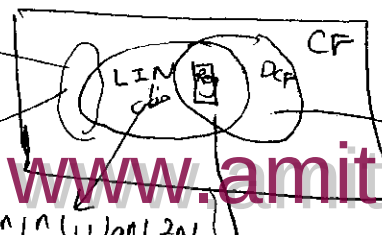
نه ندارد

پس حساسیت به متغیر

$L = \{ a^i \mid a_i \notin L(M_i) \}$

Chomskey Hierarchy: 1959

مطلق نشدن صهم حتی مستقل غیر مطلق است.



ذاتاً مبهم هستند

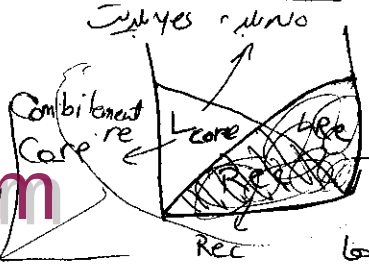
IA

$\{a^n b^n\} \cup \{a^n b^{2n}\}$

$\{a^n b^n\}$

$w = w^R$

قطعی در حین شدن



کلاس Re

تک تک این LRe را مکمل کنیم آنهایی که Rec

www.amitisweb.com

Limits of Algorithmic Computation

Friday, September 23, 2011
10:36 AM

Problem → Decidable
Problem → Undecidable

Function → Computable
Function → Uncomputable

REC: Decidable

RE : ?

* معادل هر مساله یک زبان وجود دارد
* دامنه یک مساله روی خواص آن تأثیر گذارند.

Halting Problem:

M : TM

$w \in \Sigma_M^*$ ⟶ if M halts on w or not?

www.amitisweb.com

مسأله [illegible] ماشین تورینگ M است که رفتارش به شکل زیر از خود ارائه می دهد:

اگر M روی w متوقف می شود ، $q_0 w_M w \vdash^* x_1 q_y x_2$

اگر M روی w متوقف نمی شود ، $q_0 w_M w \vdash^* y_1 q_n y_2$

که q_y و q_n عضو مجموعه حالات نهایی H هستند.

قضیه: هیچ ماشین تورینگی وجود ندارد که مسئله H را رفع کند. بنابراین مسأله halting تصمیم ناپذیر است.

اثبات: ؟

قضیه: اگر مسأله halting تصمیم پذیر می بود آنگاه تمام زبان های RE، REC می شدند. بنابراین مسأله halting تصمیم ناپذیر است.

Reduction (کاهش)

A: problem

B: problem

$$A \xrightarrow{reduce} B$$

① A → Decidable

② Undecidable → B

مسأله ۱۰ هیلبرت:

$$5x^2y^3 + 10y^4 + 7 = 0$$

$$5x^3 + 7x^4 + 3 = 0$$

www.amitisweb.com

M : TM

$w \in \Sigma^+$

$q \in Q$

halting problem $\longrightarrow$ State-entry problem

(M, w, q_f)

: Blank-tape halting problem : مثال

M

halting problem $\longrightarrow$ BTHP

$M \searrow$
$w \nearrow M_w$

$M \searrow$
$aab \nearrow M_{aab}$:

$\delta(q, \square) = (q', b, R)$
$\delta(q', \square) = (q'', a, L)$
$\delta(q'', \square) = (q''', a, R)$
$\delta(q''', \square) = (q_0, \square, R)$

www.amitisweb.com

blank-tape متوقف می‌شود.

تمرین‌های 12.1 : ۳، ۴، ۵، ۶، ۹، ۱۰، ۱۲، ۱۳، ۱۵ و ۱۶

13. $B = \{M \mid M \text{ halts on blank-tape}\}$

$C = \{M \mid M \text{ does not halt on blank-tape}\}$

16. Any problem whose domain is finite is decidable!

9. halting problem on DFAs?

" " on DPDAs?

" " on PDAs? $\delta(q_0, \lambda, z) = (q_0, z)$

قضیه ناپذیری برای زبان‌های RE

قضیه: فرض کنید G یک گرامر بدون محدودیت باشد. آنگاه $L(G) = \emptyset$ تصمیم‌ناپذیر است.

اثبات: فرض کنید M یک ماشین تورینگ باشد. آنگاه این مسئله که آیا $L(M)$ متناهی هست یا نه تصمیم‌ناپذیر است.

www.amitisweb.com

آیا L(M) منظم هست یا نه ...

Rice :

هر خصوصیت non-trivial از زبان‌های RE تصمیم‌ناپذیر است. به خصوصیت non-trivial
گفته می‌شود که برای بعضی زبان‌های RE برقرار باشد، لیکن برای تمام زبان‌های RE برقرار نباشد.

های بخش 12.2: ۳، ۴، ۵، ۶، ۷، ۸

3. $L(M_1) \subseteq L(M_2)$?

4. G_1: UN

Is $L(G)^R$ RE?

6, 7) G_1: UN

G_2: Reg.

Is $L(G_1) \cap L(G_2) = \emptyset$ decidable?

www.amitisweb.com

تصمیم ناپذیری برای زبان‌های مستقل از متن :

. : هیچ الگوریتمی وجود ندارد که بتواند تصمیم بگیرد آیا یک گرامر مستقل از متن داده شده مبهم هست یا نه .

. : هیچ الگوریتمی وجود ندارد که برای دو گرامر مستقل از متن دلخواه به نام‌های G_1 و G_2 تعیین کند که آیا

$L(G_1) \cap L(G_2) = \emptyset$

هست یا نه .

ش 12.3 : ۲ الی ۸

6. G : CF

$L(G) = \Sigma^*$ decidable or not ?

5. L_1 : Reg

$L(G)$: CF

$L_1 = L(G)$ decidable or not ?

$L_1 \subseteq L(G)$

$L(G) \subseteq L_1$

www.amitisweb.com

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

Created with Microsoft Office OneNote 2007
One place for all your notes and information

www.amitisweb.com