نظرية الأتومات

د. ريما الكحة — 2015 / 10 / 8

★ المحاضرة الأولى :

Automata theory and formal Languages

- هي نظرية الآلة : تمثل لخوارزمية معينة (برنامج معين)

تعتبر الأتومات واللغات الصورية في علوم الحاسب النظرية وتدرس النمذجة الرياضية في الحاسوب وعلاقة الأتومات باللغات الصورية وكيفية توليد لغة والتعرف عليها

للمادة تطبيقات عديدة ولطاقة في تصميم المترجمات ومعالجة النصوص.

★ اللغة الصورية : هي لغة نظرية غير ؟ مثل {111, 000}

Fundemental concepts: مفاهيم أساسية

- الرمز Symbol « الكائن غير القابل للتجزئة »

مثل الحروف اللاتينية A .... Z

a .... z

الحروف العربية ا ...... ي

الأرقام ٠ .... ٩

- الأبجدية Alphabet :

هي مجموعة منتهية ومحدودة وغير خالية من الرموز تتميز بأنه لا يمكن توليد أي رمز منها بواسطة بقية الرموز نرمز لها عادة بـ Σ

مثل : $\Sigma = \{a \dots z, A \dots Z\}$ [1]

الأبجدية في النظام العشري

$\Sigma = \{0, 1, \dots, 9\}$ [2]

إنّ المجموعة {٠,١,١٠} ليست أبجدية لأنه أمكن توليد (١٠) من تعاقب ٠ ثم ١ [3]

أي عناصرها (١٠)

المجموعة {2,4,7} تشكل أبجدية [4]

Alphabet NOTEBOOK

Subject

• الأبجدية String:

هي سلسلة منتهية من رموز الأبجدية متواضعة بجانب بعضها دون فراغات أو فواصل.

عادة نرمز للسلاسل برموز صغيرة . u, v, x, y

مثال: لتكن لدينا الأبجدية التالية $\Sigma = \{a, b\}$

فإن: $w = aabbbba$ سلسلة من الأبجدية Σ

$v = abbaa$ سلسلة من الأبجدية Σ

• طول السلسلة length of string

هو عدد الرموز المشكلة للسلسلة نرمز لطول السلسلة w بـ $|w|$

مثال: من المثال السابق

$|w| = |aabbbba| = 7$

• السلسلة الفارغة Empty string:

هي سلسلة لا تحوي أي رمز طولها = الصفر

يرمز لها بـ ε

مثال: $w = \varepsilon \Rightarrow |w| = 0$

• تعاقب السلاسل Concatenation:

"وصل، تعاقب"

تعاقب سلسلتين x و y هو سلسلة مشكلة من توضع رموز السلسلة الأولى x متبوعة مباشرة برموز السلسلة y

ونرمز لعملية التعاقب بـ (.) أو نحذف معظم الأيام

$x.y = xy$

مثال:

مثال: لتكن الأبجدية {0,1}=Σ ولنأخذ سلسلتين

$x = 00$

$y = 0100$

عندئذ: $x.y = 000100$

$y.x = 010000$

ملاحظة: عملية التعاقب هي عملية غير تبديلية ولكنها تجميعية و الحيادي هو ε

• بادئة سلسلة prefix:

تكون السلسلة u هي بادئة السلسلة v إذا وجدت سلسلة w بحيث يتحقق: $v = u.w$

مثال: لتكن $v = ababbbaa$ فإن مجموعة بادئات سلسلة v هي:

$w = \{\varepsilon, a, ab, aba, abab, ababb, ababbb, ababbba, ababbbaa\}$

• لاحقة سلسلة Suffix:

تكون السلسلة u لاحقة السلسلة v إذا وجدت سلسلة w بحيث يتحقق $v = w.u$

مثال: مجموعة لاحقات السلسلة السابقة:

خاطئة

$u = \{\varepsilon, a, aa, aab, aabb, aabbb, aabbba, aabbbab, aabbbaba$

$u = \{\varepsilon, a, aa, baa,$?

• السلسلة الجزئية Substring:

تكون السلسلة u سلسلة جزئية من السلسلة v إذا وجدت سلسلتين x و y بحيث يتحقق:

$v = xuy$

أي أن كل رموز السلسلة u موجودة في v مع المحافظة على التسلسل

مثال: في المثال السابق

$v = ababbbaa$

فإن:

$$\left.\begin{array}{l} u = abb \\ u = baa \\ u = bbaa \\ u = ababb \\ u = a \\ u = bb \end{array}\right\}$$ سلاسل جزئية من v

أما $u = aab$ ليست سلسلة جزئية من v.

• قوة الأبجدية power of Alphabet:

هي مجموعة كل السلاسل التي يمكن تركيبها باستخدام عملية التقاطب بين رموز الأبجدية Σ

نرمز لمجموعة الكلمات المعرفة على Σ ذات الطول n بـ Σ^n

مثال: لتكن الأبجدية $\Sigma = \{0,1\}$

- $\Sigma^0 = \{\varepsilon\}$
- $\Sigma^1 = \{0,1\} = \Sigma$
- $\Sigma^2 = \{0,1\}.\{0,1\} = \{00,01,10,11\}$
- $\Sigma^3 = \Sigma^2.\Sigma^1 = \Sigma^1.\Sigma^2 = \Sigma.\Sigma.\Sigma$

$= \{00,01,10,11\}.\{0,1\}$

$= \{000,001,010,011,100,101,110,111\}$

Subject / /

نعرف Σ^* بأنها:

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \ldots$$

حيث Σ^*: مجموعة كل السلاسل من حروف الأبجدية Σ وهي مجموعة غير منتهية وتشمل الـ ϵ دوماً.

ملاحظة (1):

أحياناً يلزمنا تعريف كل السلاسل على الأبجدية Σ ماعدا السلسلة الفارغة ϵ عندئذ لذلك بالرمز.

$$\Sigma^+ = \Sigma^* - \{\epsilon\}$$

$$= \Sigma^* - \Sigma^0$$

$$\Rightarrow \Sigma^* = \Sigma^+ \cup \Sigma^0$$

ملاحظة (2):

$$\{a^*\} = \{\epsilon, a, aa, aaa \ldots\}$$

تكرار a من 0 ← ∞

نسمي a^* ، b^* : دوماً بالتعبير المنتظم.

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

$$aba \notin \{(ab)^*\} = \{\epsilon, ab, abab, ababbab, \ldots\}$$

$$aba \notin \{a^*b\} = \{b, ab, aab, aaab, aaaab \ldots\}$$

$$\epsilon \notin \{ab^*c^*\} = \{a, ac, ab, abc, abbc, abbbc, \ldots\}$$

$$bca \notin \{a^*b^*c^*\} = \{\epsilon, a, b, c, ab, ac, bc, abc, aabbbc, \ldots\}$$

سؤال

$$bca \in \{(a^*b^*c^*)^*\}$$ الأولوية للتكرار

$L((a^*b^*))^* =$ الأولوية للقوى

شرح

$$(a^n b^m)^k = (a^3 b^4)^2 = a^3 b^4 a^3 b^4$$

$$(a^n b^m)^2 = (a^3 b^4)(a^2 b) = aaabbbbaab$$

مرتين ولكن كل مرة تشكل مجموعة

تكرار مرتين

المحاضرة الثانية

$L(a^*) = \{a^*\}$

اللغة التي يولدها التعبير المنتظم a^* = مجموعة السلاسل التي يمكن توليدها من a^*

$L((a^* b^*))^* = \{\epsilon, a, b, ab, aab, aaaabbbbb, aabbbaaaaab, aaba \dots\}$

Σ^* مجموعة كل السلاسل المتشكلة من الأبجدية Σ

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

$\Sigma^* = \{\epsilon,$

اللغة: وهي مجموعة السلاسل المختارة من Σ^* والمعرفة على الأبجدية Σ ونرمز لها عادةً بالرمز L
أي أنه إذا كان Σ أبجدية وكان $L \subseteq \Sigma^*$ فإن L هي لغة مولدة من ~~لغة~~ الأبجدية Σ

ملاحظة: إن أي لغة معرفة على أبجدية لا تحوي بالضرورة على كل السلاسل المولدة من الأبجدية.

$\Sigma^* = \{$ كل، أكل، .... $\}$

كل $\notin$ L

مثال:

لتكن لدينا الأبجدية $\Sigma = \{a,b\}$

$L \subseteq \Sigma^*$ هي اللغة التي تحوي على التعاقب aa

(سلاسل ~~[illegible]~~

وبالتالي :

$$L=\{\boxed{aa},\ \boxed{aa}bbb,\ baa\boxed{aa}bba, \ldots\}$$

ليس من الضروري أن تكون اللغة منتهية أو غير منتهية .

$$babbb \notin L$$

التعابير المنتظمة: لتكن لدينا الأبجدية $\Sigma=\{0,1\}$ عندئذ

▪ التعبير ϵ هو التعبير المنتظم الذي يحدد اللغة التي تحوي السلسلة الفارغة أي اللغة التي تولدها ϵ $\quad L(\epsilon)=\{\epsilon\}$

▪ التعبير 0 " " " " " " " " " 0

أي ، $L(0)=\{0\}$

▪ التعبير 1 " " " " " " " " " 1

أي : $L(1)=\{1\}$

▪ التعبير 0^* هو التعبير المنتظم الذي يحدد اللغة التي تحوي جميع السلاسل الحاوية على عدد من الأصفار مساوي من الصفر، الواحد، ... أي:

$$L(0^*)=\{0\}^*=\{\epsilon,0,00,000,\ldots\}$$

▪ التعبير 1^* هو التعبير المنتظم

$$L(1^*)=L(1)^*=\{1\}^*=\{\epsilon,1,11,111,\ldots\}$$

▪ التعبير المنتظم $1+0$: يحدد اللغة التي تحوي إحدى السلسلتين أي:

$$L(0+1)=L(0)\cup L(1)=\{0\}\cup\{1\}=\{0,1\}$$

التعبير المنتظم 011 تتابع يحدد اللغة " " " " أي :

$$L(011)=L(0).L(1).L(1)=\{0\}.\{1\}\{1\}$$
$$=\{011\}$$

Subject

ملاحظة: أفضلية العمليات في التعابير المنتظمة هي:

1) عملية التكرار * ثم 2) التعاقب (.) ثم 3) الـ + (أو)

فائدة:

$$L((0+1)^*) = (L(0+1))^* = (\{0\} \cup \{1\})^*$$

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

$$= \{\epsilon, 0, 1, 00, 11, 011, 10000, \ldots\}$$

توضيح

$$(0+1)^1 = 0$$

$$(0+1)^2 = \underbrace{(0+1)}_{\text{اختر}} . \underbrace{(0+1)}_{\text{اختر}}$$

$$= 1 \, . \, 1$$

$$(0+1)^3 = (0+1).(0+1).(0+1)$$

$$= 0 \quad 1 \quad 1$$

$$L(0.0 \boxed{+}_{\text{أو}} 1.1)^* = \{\epsilon, 00, 11, 0000, 0011, 1100 \ldots$$

$$(L(00)^* + L(11)^*)^* = 0011, 0000$$

$$L((01^*0 \boxed{+}_{\text{أو}} 1)^*) = \{\epsilon, 00, 01, 0110, 01111, 1, 0110011110$$

$$\{01^*0 , 1\}$$

مجموعة السلاسل

التعبير 0^*110^* المنتظم: هو التعبير المنتظم الذي قد يبدأ بأي عدد من الأصفار متبوعة بواحدين وينتهي أيضاً بأي عدد من الأصفار

$$L(0^*110^*) = \{11, 011, 110, 0011, 00011$$

$$L(0^*1^*\emptyset^*0^*) = \{\epsilon, 0001100, \ldots\}$$

$$L(a^+) = \{a, aa, aaa, \ldots\}$$

IR^+

للدلالة على أننا نعتبر IR تتكرر مرة واحدة على الأقل أي 1، 2، 3، ...

Alphabet NOTEBOOK