ITSE321
Software Construction
بنـــاء البرمجيات

المحاضر: د. رضوان حسين
erudwan@yahoo.com

التعبيرات النظامية/الاعتيادية

REGULAR EXPRESSIONS

LEXEME PATTERNS

نمط المفردات

- **المفردات التي يجزءها محلل المفردات إلى بطاقات tokens تتبع قواعد rules تصف فئات/أنواع المفردات الموجودة بالبرنامج المصدري.**
- **هذه القواعد تسمى أنماط المفردات**
- **النمط: هو وصف للشكل الذي تأخذه المفردات في بطاقة ما a token وهو بُنية مركبة complex تتوافق عليها مفردات عديدة**
  - **بطاقة الكلمات المحجوزة تتبع نمطاً <keyword>**
    - سلسلة من الحروف
  - **بطاقة المعرفات/المتغيرات تتبع نمطاً <identifier>**
    - سلسلة من الحروف والأرقام ولا تبدأ برقم، وليست كلمة محجوزة

EXAMPLES OF TOKENS

أمثلة للبطاقات

```
int Index;
Index = 2 * count +17;
```

Lexemes	Tokens
int	type
Index	identifier
=	assignment
2	int_constant
*	operation
Count	identifier
+	operation
17	int_constant
;	semicolon

LEXEME PATTERNS SPECIFICATION

مواصفات نمط المفردات

- نحتاج لصياغة قادرة على التعبير عن أنماط البطاقات
- التعبيرات النظامية Regular Expressions تستخدم لوصف نمط صياغة المفردات
- مفردات أي لغة يعبر عنها من خلال الحروف الهجائية لتلك اللغة
  - ❖ اللغة العربية, حروفها الهجائية التي تشكل جميـــــع كلماتها هي أ – ي
  - ❖ اللغة الإنجليزية, حرفها الهجائية alphabet هي a – z, A - Z

GREEK ALPHABET

Αα	Ββ	Γγ	Δδ	Εε	Ζζ
ALPHA *Al-fah*	**BETA** *Bay-tah*	**GAMMA** *Gam-ah*	**DELTA** *Del-tah*	**EPSILON** *Ep-si-lon*	**ZETA** *Zay-tah*
Ηη	Θθ	Ιι	Κκ	Λλ	Μμ
ETA *Ay-tah*	**THETA** *Thay-tah*	**IOTA** *Eye-o-tah*	**KAPPA** *Cap-ah*	**LAMBDA** *Lamb-da*	**MU** *Mew*
Νν	Ξξ	Οο	Ππ	Ρρ	Σσς
NU *New*	**XI** *Zeye (if it stands alone) or Zee (if followed by a letter)*	**OMICRON** *Om-i-cron*	**PI** *Pie*	**RHO** *Row*	**SIGMA** *Sig-mah*
Ττ	Υυ	Φφ	Χχ	Ψψ	Ωω
TAU *Taw*	**UPSILON** *Oop-si-lon*	**PHI** *Fee (if stands alone) or Feye (if followed by a letter)*	**CHI** *Keye*	**PSI** *Sigh*	**OMEGA** *O-may-gah*

ALPHABET الحروف الهجائية للغة

- **تعتمد التعبيرات النظامية Regular Expressions على مبادئ مادة التراكيب المنفصلة Discrete Structures**
- **يعبر عن الحروف الهجائية للغة بالرمز الإغريقي سيقما Σ**
- حيث **Σ** عبارة عن فئة محدودة finite تحتوي على جميع الرموز (حروف وأرقام) وأيضاً العلامات (الفواصل, النقاط, وغيرها) التي يمكن أن تشكل **جملة** في لغة ما
- اللغة الإنجليزية: **Σ** = {a-z, A-Z}

مفردات فئة هجائية

- بفرض أن سيقما = فئة الحروف a, b, c, d
- $\Sigma=\{a,b,c,d\}$
- المفردات الممكنة من الهجائية Σ هي:
  - a
  - aa
  - aaa
  - aabbccdd
  - d
  - abab
  - ccccccccccacccc
  - وهكذا, أي تركيبة من الحروف الأربعة a, b, c, d

إذن يمكننا أن نصنع لغة ما لها شكل معين
- نحدد ماهي رموزها الهجائية
- ونحدد كيف تشكل تلك الرموز لتصنع مفردات اللغة

FORMAL LANGUAGES اللغات الشكلية

- **الهجائية Σ**: سيقما هي فئة محدودة finite تحتوي على كل مدخلات inputs الرموز characters أو العلامات symbols
- **نهاية الهجائية Closure Σ^* : سيقما ستار** هي فئة كل المفردات الممكنة تشكيلها من سيقما Σ , ويشمل ذلك المفردة الخالية ε (إبسيلون) empty string
- **اللغة الرسمية formal language L** هي فئة جزئية من **Σ^* سيقما ستار**
  - **فهي فئة المفردات ذات المعنى في اللغة, كجزء من جميع المفرات الممكنة سيقما ستار Σ^***

FORMAL LANGUAGE OPERATIONS

عمليات اللغات الشكلية

- الاتحاد بين لغتين L و M Union: هو فئة المفردات التي تنتمي على الأقل لأحد اللغتين L أو M , "على الأقل" تعني أن بعض المفردات قد تنتمي لكلا اللغتين.

$$L \cup M = \{s | s \in L \; or \; s \in M\}$$

$$L \cup M = \{s | s \in L \vee s \in M\}$$

- مثال:

- $L = \{abb, baa, aba, bab\}, M = \{doo, ree, mee, baa\}$
  - $L \cup M = \{abb, baa, aba, bab, doo, ree, mee\}$

FORMAL LANGUAGE OPERATIONS

عمليات اللغات الشكلية

- التقاطع بين لغتين L و M Intersection: هو فئة المفردات التي تنتمي إلى كلا اللغتين.

$$L \cap M = \{s | s \in L \; and \; s \in M\}$$

$$L \cap M = \{s | s \in L \wedge s \in M\}$$

- مثال:

- $L = \{a, aa, aaa, aaaa\}, M = \{bd, bbdd, bdbd\}$
  - $L \cap M = \{\varepsilon\}$ إبسيلون: وهي فئة خالية

FORMAL LANGUAGE OPERATIONS

عمليات اللغات الشكلية

- لصق/ربط لغتين L و M Concatination: فئة كل المفردات على التشكيل

st حيث s عبارة عن مفردة من اللغة L و t عبارة عن مفردة من اللغة M

$$LM = \{st | s \in L \; and \; t \in M\}$$

- مثال:

- $L = \{a, aa\}, M = \{bd, bbdd\}$
  - $LM = \{abd, abbdd, aabd, aabbdd\}$

FORMAL LANGUAGE OPERATIONS

عمليات اللغات الشكلية

- نهاية كليين للغة L (Kleene closure) مسماة عن شخص اسمه كليين: فئة

كل المفردات الناتجة من لصق 0 مفردة أو أكثر من مفردات لغة ما L

$$L^* = \cup_{i=0}^{\infty} L^i$$

عندما $i = 0$, L^0 تعني لصق صفر من مفردات اللغة L وهو يساوي فئة خالية $\{\varepsilon\}$, وهي تعني لاشئ.

$$L^* = \{\varepsilon\} \cup \{wz \mid w \in L \ \wedge z \in L^*\}$$

مثلاً: $L = \{at, bat, cat\}$

$$L^* = \left\{ \begin{matrix} \varepsilon, at, atat, atbat, atcat, \\ bat, batat, batbat, \\ batcat, catat, catbat, \\ atbatcat, batatcat, catatbat, \ldots \end{matrix} \right\}$$

أي أن L^* هي فئة كل احتمالات لصق لمفردات اللغة بمافيها الفئة الخالية

FORMAL LANGUAGE OPERATIONS

عمليات اللغات الشكلية

- نهاية إجابية للغة L (Positive closure): فئة كل المفردات الناجمة عن لصق مفردة واحدة أو أكثر من مفردات لغة ما L, أي لا يوجد فئة خالية $i \neq 0$

$$L^+ = \cup_{i=1}^{\infty} L^i$$

مثلاً: $L = \{at, bat, cat\}$

$$L^+ = \left\{ \begin{array}{c} at, atat, bat, cat, atbat, atcat, batat, batcat, \\ catat, catbat, \\ atbatcat, batatcat, catatbat, \ldots \end{array} \right\}$$

أي فئة كل احتمالات لصق لمفردات اللغة ولكن بدون الفئة الخالية

FORMAL LANGUAGE OPERATIONS

عمليات اللغات الشكلية

- **الامتداد المحدود Finite exponential: فئة n للمفردات الناجمة عن لصق عدد محدد لمفردات لغة ما L, يساوي L^n**

عندما $n = 0$ تكون $L^0 = \{\varepsilon\}$,

وكلما كانت $n > 0$ فإن $L^n = L^{n-1}L$

عندما $n = 4$ فإن $L^4 = L^3\ L$

$L^4 = L^2 LL$

$L^4 = LLLL$

FORMAL LANGUAGE OPERATIONS

عمليات اللغات الشكلية

عندما $n = 0$ تكون $L^0 = \{\varepsilon\}$, وكلما كانت $n > 0$ فإن $L^n = L^{n-1}L$

مثلاً اللغة

$$\boldsymbol{L} = \{a,\ \ b,\ \ c,\ \ dd, ee, ff\}$$

عندما $n = 1$, $L^1 = L = \{a,\ \ b,\ \ c,\ \ dd,\ \ ee,\ \ ff\}$

$n = 2$, $L^2 = LL = \{$ $\}$

$n = 3$, $L^3 = LLL = \{$ $\}$

FORMAL LANGUAGE OPERATIONS

عمليات اللغات الشكلية

- مثلاً:

L هي فئة الحروف الإنجليزية الصغيرة والكبيرة

$$L = \{A, B, \dots, Z, a, b, \dots z\}$$

و D هي فئة الأرقام

$$D = \{0, 1, \dots 9\}$$

فإن $L(L \cup D)^*$ تعبر عن فئة كل المفردات من حروف وأرقام وتبدأ بحرف ويعتبر هذا تعبيراً عن نمط صياغة المعرفات (أسماء المتغيرات identifiers)

التعبيرات النظامية

- التعبيرات النظامية هي صياغة مختصرة لوصف نص المفردات
- المعرفات في لغات البرمجة عبارة عن حرف متبوع بصفرٍ أو أكثر من الحروف أو الأرقام
- Identifier→letter(letter|digit)*
- $ID \rightarrow L(L \cup D)^*$
- المُعرِف← حرف(حرف|رقم)*
- العمود | يعبر عن الاتحاد، وينطق أيضاً "أو" or
- الأقواس ( ) تستخدم لتجميع التعبيرات الجزئية
- النجمة * :تعني حدوث صفر أو أكثر من مابين الأقواس، Kleene Closure

التعبيرات النظامية

- التعبيرات النظامية يتم بناؤها بالتكرار recursively من خلال تعبيرات نظامية صغيرة، وذلك باتباع قواعد معينة rules وهي:

1. إبسيلون ε تعبير نظامي يشير denotes إلى اللغة $L(\varepsilon)$ وتساوي $\{\varepsilon\}$, وهي فئة خالية لاتحتوي على نص
2. إذا كانت a هي رمز في الهجائية Σ , فإن a تكون تعبيراً نظامياً يشير إلى الفئة $\{a\}$ وهي تحتوي على النص a

a حرف من حروف سيقما, فإن وجدت لوحدها فهي تعبير نظامي للغة $L(a)$ التي تساوي $\{a\}$ كلغة من مفردةٍ واحدة بطول رمز واحد

التعبيرات النظامية

- تقليدياً, تكتب التعبيرات بالخط **العريض**, مثل $\boldsymbol{a}$, وتكتب الرموز بالخط المائل العادي, مثل a.

$$\boldsymbol{L(a) = \{a\}}$$

لغة التعبير النظامي **a**

- بما أن التعبيرات الكبيرة تتألف من تعبيرات نظامية صغيرة, فهذه 4 تعبيرات جزئية أساسية:

1. **(r)|(s)** هو تعبير "أو" ويشير إلى اتحاد اللغتين $\boldsymbol{L(r) \cup L(s)}$
2. **(r)(s)** هو تعبير يشير إلى لصق اللغتين $\boldsymbol{L(r)L(s)}$
3. **(r)*** هو تعبير يشير إلى $\boldsymbol{(L(r))^*}$
4. **(r)** هو تعبير عن اللغة $\boldsymbol{L(r)}$

REGULAR EXPRESSIONS PRECEDENCE

أسبقيات التعبيرات النظامية

- التعبيرات النظامية تحتوي أقواس يمكن الاستغناء عنها بشرط اتباع أسبقيات مشغلاتها
- المشغل * له الأسبقية الأعلى
- للصق concatenation الأسبقية الثانية
- المشغل أو | له الأسبقة الأخيرة
- طبعاً قراءة التعبيرات من اليسار إلى اليمين

إذن يمكن استبدال التعبير $((a)|((b)^*(c)))$ بالتعبير $(a)|(b^*c)$ أو $a|b^*c$

كلا التعبيرين يشير إلى فئة المفردات وهي: a أو b closure ملصقة بـها c

{a, c, bc, bbc, bbbc, bbbbc, bbbbbc, bbbbbbc, ...} =

أمثلة التعبيرات النظامية

- ما هي لغة التعبير $(a|b)(a|b)$
  - $\{aa,ab,ba,bb\}$ a أو b يلتصق مع a أو b
- هل يمكنك كتابة تعبير آخر يعطي نفس اللغة الناتجة من التعبير $(a|b)(a|b)$
  - $aa|\ ab|ba|bb$
- ما هي لغة التعبير a^*
  - $\{\varepsilon, a,aa,aaa,aaaa, \dots\}$ لاشئ, و كل احتمالات لصق a مع نفسها
- ما هي لغة التعبير $(a|b)^*$
  - $\{\varepsilon,a,b,aa,bb,aaa,bbb, \dots\}$ لاشئ, و كل احتمالات a أو احتمالات b
- ما هي لغة التعبير $a|a^*b$
  - $\{a,b,ab,aab,aaab, \dots\}$ المفردة a و 0 أو عدد من a ملصق مع b

PROPERTIES OF REGULAR EXPRESSION

خواص التعبيرات النظامية

مثل رياضي	وصف	قانون المسلَّمة Axiom
3+6 =3+6	commutative تبديلية |	$r\|s = s\|r$
4+(3+6) = (4+3)+6	associative ترابطية |	$r\|(s\|t) = (r\|s)\|t$
	اللصق ترابطية	$(rs)t = r(st)$
4 (3+6) = (4x3+4x6)	اللصق توزيعي على |	$r\,(s\|t) = rs\|rt$ $(s\|t)r = sr\|tr$
3+0 = 3 0+3 = 3	ε محايد للصق identity	$\varepsilon r = r$ $r\varepsilon = r$
	علاقة ε مع *	$r^* = (r\|\varepsilon)^*$
	* تكرارها كحدوثها مرة واحدة idempotent	$r^{**} = r^*$

REGULAR DEFINITIONS

تعريفات نظامية

- يمكن كتابة صيغة لتعريفات نظامية على النحو التالي:

- $d_1 \rightarrow r_1$
- $d_2 \rightarrow r_2$
- ...
- $d_n \rightarrow r_n$

- حيث أن كل d_i عبارة عن اسم لتعريف definition يطلع على تعبير نظامي r_i
- فئة هجائيات هذه التعريفات تتكون من جميع الرموز التي تعبر عنها التعريفات $\Sigma \cup \{d_1, d_2, \ldots, d_n\}$ وتساوي جميع رموز هذه التعريفات

REGULAR DEFINITIONS أمثلة لتعريفات نظامية

$$letter \rightarrow (a|b|c|\dots|z|A|B|C|\dots|Z)$$

$$digit \rightarrow (0|1|2|3|4|5|6|7|8|9)$$

$$id \rightarrow letter(letter|digit)*$$

$$integer \rightarrow (+|-|\varepsilon)digit^{+}$$

$$decimal \rightarrow integer.digit^{*}$$

$$real \rightarrow (integer|decimal)E(+|-)digit^{+}$$

REGULAR DEFINITIONS

اختصارات للتعريفات نظامية

$$\boldsymbol{letter} \rightarrow (\boldsymbol{a}|\boldsymbol{b}|\boldsymbol{c}|\ldots|\boldsymbol{z}|\boldsymbol{A}|\boldsymbol{B}|\boldsymbol{C}|\ \ldots|\boldsymbol{Z})$$

$\boldsymbol{letter} \rightarrow [\text{a} - \text{z A - Z}]$ **يمكن أن تختصر:**

$$\boldsymbol{digit} \rightarrow (\boldsymbol{0}|\boldsymbol{1}|\boldsymbol{2}|\boldsymbol{3}|\boldsymbol{4}|\boldsymbol{5}|\boldsymbol{6}|\boldsymbol{7}|\boldsymbol{8}|\boldsymbol{9})$$

$\boldsymbol{digit} \rightarrow [\boldsymbol{0} - \boldsymbol{9}]$ **يمكن أن تختصر:**

REGULAR DEFINITIONS أمثلة لتعريفات نظامية

$$\boldsymbol{if} \rightarrow \text{if}$$

$$\boldsymbol{then} \rightarrow \text{then}$$

$$\boldsymbol{else} \rightarrow \text{else}$$

$$\boldsymbol{relop} \rightarrow \text{< | > | = | <= | >= | <>}$$

$$\boldsymbol{while} \rightarrow \text{while}$$

$$\boldsymbol{int} \rightarrow \text{int}$$

تمارين التعبيرات النظاميـة

REGULAR EXPRESSION

Exercises