Regüler İfadeler ve Regüler Diller

Hafta 3

- Dilin izin verdiği sözcükleri tanımlamak için kullanılan araçlardan biri Regüler ifadelerdir.

Genel tanımlar:

- Alfabe sonlu simge/karakterler kümesi -- {a,b}, ASCII
- Katar sonlu simgeler dizisi (sözcük) – 001, bc, baba, while
- Uzunluk (|x|) katarı oluşturan simge sayısı
- Boş katar "Λ" uzunluğu sıfır olan katar
- Bitiştirme iki katarı birbirini izleyecek şekilde birleştirme
  - x= abc y= de → xy=abcde
  - Λx=x Λ=x
  - x^n x katarı kendisiyle "n" kez bitiştirilir, x^0 → Λ

- Bitiştirme (Concatenation):
  - $L_1L_2 = \{ s_1s_2 \mid s_1 \in L_1 \text{ and } s_2 \in L_2 \}$
- Birleşim (Union)
  - $L_1 \cup L_2 = \{ s \mid s \in L_1 \text{ or } s \in L_2 \}$
- Üs (Exponentiation):
  - $L^0 = \{\varepsilon\}$ $L^1 = L$ $L^2 = LL$
- Yıldız Kapanma (Kleene Closure):
  - $L^* = \bigcup_{i=0}^{\infty} L^i$
- Pozitif Kapanma (Positive Closure)
  - $L^+ = \bigcup_{i=1}^{\infty} L^i$

- Dil sonlu sayıda simgeden oluşan bir alfabeden üretilen katarlar kümesidir.
  - $\{\Lambda\}$ – boş katarı içeren küme
  - {0,1,00,01,10,11}
  - Doğal diller, programlama dilleri
- Bitiştirme işlemi: L ve M iki ayrı dil olmak üzere, LM dilinin elemanları, L'nin içerdiği tüm katarların M'nin içerdiği tüm katarlarla bitiştirilmesinden oluşur.
  - L={0,01,110} M={10,110},
    LM={010,0110,01110,11010,110110}
- $L_1 = \{a,b,c,d\}$ $L_2 = \{1,2\}$
- $L_1L_2 = \{a1,a2,b1,b2,c1,c2,d1,d2\}$

- $L_1 \cup L_2 = \{a,b,c,d,1,2\}$

- L_1^3 = üç uzunluklu bütün katarlar kümesi (a,b,c,d üzerinde tanımlı}
- L_1^* = a,b,c,d simgeleriyle oluşturulanbütün katarlar (Λ (empty string) dahil)
- L_1^+ = boş katarı içermez.

- Kleene Yıldızı Operatörü (*) İstenilen sayıda (belirsiz sayıda) bitiştirme için kullanılır.

- L^* L dilinin kendisiyle belirsiz sayıda bitiştirilme işlemi
  - D= 0,1,2...,9 D^* → rakamlardan oluşan tüm rakamlar, 0 dahil
  - L={aa} L^* → çift sayıda "a" karakterinden oluşan tüm katarlar

  - L^0={}, L^1={aa}, L^2={aaaa}...
- L^* "ε" da içerir. "ε" dışlamak için LL^* yazılmalıdır

- $L^+ = LL^*$ en az bir veya daha fazla sayıda bitiştirme işlemi

- Bir regüler ifade şöyle tanımlanır

  (R ve S regüler ifadeler olmak üzere)

  - a — Alfabenin her simgesi bir regüler ifadedir.
  - ε — boş katar bir regüler ifadedir
  - R+S — "R veya S" bir regüler ifadedir
  - RS — "R ve S" (bitiştirme) bir düzgün ifadedir
  - R* — 0 veya daha fazla R'nin bitiştirilmesiyle elde edilen bir regüler ifadedir.

Dil

- Bir R regüler ifadesi, L(R) nin ifade ettiği karakter katarlarını (sözcükler) tanımlar.
- L(R) = R'nin tanımladığı dil
  - L(abc) = { abc }
  - L(onay|red) = {evet|hayır}
  - L(1(0|1)*) = 1 ile başlayan tüm ikili sayılar
- Her sözcük bir düzgün ifade kullanarak tanımlanabilir

Örnek Düzgün İfadeler

- Düzgün İfade ..... L(R) dilinde örnek katarlar
  - a ..... “a”
  - ab ..... “ab”
  - a|b ..... “a”, “b”
  - (ab)* ..... “”, “ab”, “abab”, ...
  - (a| ε)b ..... “ab”, “b”
  - (aa|ab|ba|bb)* ..... “aa”, “bbbb”, “abbbaababb”
  - (a|b)(a|b)(a|b) ..... “bbb”, “aab”, “bab”, “abb”..

- Regüler İfade ..... L(R) dilinde örnek katarlar
  - rakam [0-9] .... “0”, “1”, “2”, ...
  - poztamsayı = rakam^{+} ..... “8”, “412”, ...
  - tamsayı = (-| ε) poztamsayı ..... “-23”, “34”, ...
  - reelsayı = tamsayı(ε |(.poztamsayı))“-1.65”, “24”, “1.085” (bu tanım “.58” ve “45.” sözcüklerine izin verir mi?)
  - harf [a-z] ..... “a”, “b”, “c”,....
  - değişken_adı = harf(harf | rakam)*..... “Ortalama”, “sayı”,...

- Öncelik
  - * en yüksek
  - concatenation sonra
  - | en düşük
- $ab^*|c$ şöyle değerlendirilir $(a(b)^*)|(c)$
  - Σ = {0,1} alfabesinde;
  - 0|1 => {0,1}
  - (0|1)(0|1) => {00,01,10,11}
  - 0^* => {ε ,0,00,000,0000,....}
  - $(0|1)^*$ => Boş katarda dahil olmak üzere 0 ve 1'li bütün katarlar.

Regular İfadeler

- Regüler ifadeleri programlama dillerinin token'larını tanımlamak için kullanırız.
- Bir regüler ifade yukarıda tanımlanan kurallarla basit regüler ifadelerin birleştirilmesiyle elde edilir.
- Her regüler ifade bir dil tanımlar.
- Bir rügüler ifade ile tanımlanan dil regüler bir kümedir.

Σ Üzerinde tanımlı Regüer ifadeler.

Reg. İfade	Dil
ε	$\{\varepsilon\}$
$a \in \Sigma$	$\{a\}$
$(r_1) \mid (r_2)$	$L(r_1) \cup L(r_2)$
$(r_1)\,(r_2)$	$L(r_1)\,L(r_2)$
$(r)^*$	$(L(r))^*$
(r)	$L(r)$
$(r)^+ = (r)(r)^*$	

- $(r)? = (r) \mid \varepsilon$

- Örnek: Belirleyiciler (Identifiers), PASCAL
- letter $\rightarrow$ A | B | ... | Z | a | b | ... | z

  digit $\rightarrow$ 0 | 1 | ... | 9

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

- Örnek :İşaretsiz tamsayı (Unsigned numbers):Pascal

  digit $\rightarrow$ 0 | 1 | ... | 9

  digits $\rightarrow$ digit $^+$

  opt-fraction $\rightarrow$ (. digits) ?

  opt-exponent $\rightarrow$ (E (+|-)? digits) ?

  unsigned-num $\rightarrow$ digits opt-fraction opt-exponent

Teşekkürler