**GPS1 & HW1 available**

**Last Time:**
- strings
- languages

**OPS on languages:**
- union, intersection, complement
- concatenation
  \[ L_1 L_2 = \{ xy : x \in L_1, y \in L_2 \} \]
- *Kleene star*
  \[ L^* = \bigcup_{i=0}^{\infty} L^i \]

**Regular Languages**

- all langs obtainable from union, concat, star
  (starting from trivial base cases)

**Formal Def's** by induction

1. \( \emptyset, \{ \varepsilon \}, \{ a \} \) are regular langs \( \forall a \in \Sigma \)
2. if \( L_1, L_2 \) are regular langs,
   then so are \( L_1 \cup L_2, L_2 \), and \( L^* \)
   (only langs obtained by finite # applns of (i),(ii)
   are regular)

\[ \Sigma = \{ 0, 1 \} \]

a) \( \{ 1001 \} \) is regular
   \[ = \{ 1 \} \cdot \{ 0 \} \cdot \{ 0 \} \cdot \{ 1 \} \]

b) \( \{ 1001, 10, 1 \} \) is regular
   \[ = \{ 1 \} \cdot \{ 0 \} \cdot \{ 0 \} \cdot \{ 1 \} \cup \{ 1 \} \cdot \{ 0 \} \cup \{ 1 \} \]
c) all finite langs are regular.

d) \( \{ x \in \{0,1\}^* : \text{\#1 is odd} \} \) is regular

\[
= \{00, 01, 10, 11\}^* \cdot \{0,1\}
\]

\[
= (\{(0\cup 1)\} \cdot \{(0\cup 1)\})^* \cdot \{0,1\}
\]

---

Notation

\( \phi, \varepsilon, a \) are regular expr for \( \emptyset, \{\varepsilon\}, \{a\} \) where \( a \in \Sigma \)

(ii) if \( r_1, r_2 \) are regular exprs for \( L_1, L_2 \) resp., then \( (r_1 + r_2) \) is reg expr for \( L_1 \cup L_2 \)

\( \{r_1^* \} \) is reg expr for \( L_1 \cdot L_2 \)

Let \( L(r) \) denote lang corresponding to expr \( r \).

---

\[ (d) \quad (0+1)(0+1)^* (0+1) \]

Rules:

- Omit unnecessary parentheses
- Precedence order: \(*, +, \cdot\)
- Shorthand: \( r^+ = r \cdot r^* \) (one or more occurrence)
- A lang may have many diff. reg expr

E.g. \( (0+1) \cdot ( (0+1)(0+1))^* \)

\( (0+1) \cdot (00+01+10+11)^* \)
\( \Sigma = \{0, 1\} \)

a) all strings with 00 as a substring
\[ (0+1)^* \text{00} (0+1)^* \]

b) all strings with 00 as a substring and having even length

2 cases: (i) part before 00: even \( \rightarrow \) even
Part after 00: even \( \rightarrow \) odd
(ii') part before 00: odd
Part after 00: odd
\[ ((0+1)(0+1))^* \text{00} ((0+1)(0+1))^* \]
\[ + \quad (0+1)(0+1)^* \text{00} (0+1)(0+1)(0+1)^* \]

c) all strings with even # of 0's
\[
\left(\begin{array}{c}
\text{00}^* \\
\text{01}^* \text{01}^* \\
\text{1}^*
\end{array}\right)
\]
\[ + \quad \left(\begin{array}{c}
\text{01}^* \\
\text{1}^*
\end{array}\right)
\]

or \( \left(\begin{array}{c}
\text{01}^* \text{0}^* \\
\text{1}^* \text{0}^* \\
\end{array}\right) \)

or \( \left(\begin{array}{c}
\text{\text{01}^*} \\
\text{\text{01}^*} \\
\end{array}\right) \)

d) all strings not beginning with 00
\[ \text{00} (0+1)^* \]

Cases:
begin with 1 \( \checkmark \) boundary case

or begin with 01 \( \checkmark \)

or 0 \( \checkmark \)
\[
1 \cdot (0+1)^* + 01 \cdot (0+1)^* + \varepsilon + 0 \\
= \ (1+01) \cdot (0+1)^* + \varepsilon + 0 \\
\]

e) all strings not containing 00 as a substring

\[
(1+01)^* \cdot (0+3)^* \\
+ (1+01)^* \cdot 0 \\
= \ (1^* + 01^*)^* \cdot (3+0) \\
\]

\[
\vdots \rightarrow (3+0) \cdot (1^0)^* \cdot 1^* \\
\]

f) all strings with even # of 0's and even # of 1's

idea - divide into blocks of two

\[
00 \ \text{or} \ 01 \ \text{or} \ 10 \\
\]

\[
\overbrace{00101010} \ \text{or} \ \overbrace{011010} \\
\]

# such blocks must be even

\[
(\# \text{of } 01 \text{ and } 10 \text{ blocks}) \text{ is even}. \\
\]

\[
(00+11)^* \cdot (01+10) \cdot (00+11)^* \cdot (01+10) \cdot (00+11)^* \cdot (01+10) \cdot (00+11)^* \\
\]

or

\[
(00+11)^* \cdot (01+10) \cdot (00+11)^* \cdot (01+10)^* \cdot (00+11)^* \\
\]
8) all strings with even # 0's 
   and # 1's div by 5 
   and not containing 01101
   hard but possible!

h) \{0^i 1^i : i \geq 0\} 
   impossible!! (how to prove it?)

---

Some identities

\[
L((r_1 + r_2) \cdot r_3) = L(r_1 r_3 + r_2 r_3) \\
L((r^*)^*) = L(r^*) \\
L((rs)^* r) = L(r (sr)^*) \\
L((r + s)^*) = L((r^* + s^*)^*) \\
\]

don't memorize!