Spring 2015
Lecture 19: Structural induction and regular expressions
• An alphabet $\Sigma$ is any finite set of characters.
  
e.g. $\Sigma = \{0,1\}$ or $\Sigma = \{A, B, C, \ldots X, Y, Z\}$ or $\Sigma =$

<table>
<thead>
<tr>
<th></th>
<th>28</th>
<th>95</th>
<th>153</th>
<th>186</th>
<th>219</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>29</td>
<td>96</td>
<td>154</td>
<td>187</td>
<td>220</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>97-122 a-z</td>
<td>155</td>
<td>188</td>
<td>221</td>
</tr>
<tr>
<td>4</td>
<td>31</td>
<td>123</td>
<td>156</td>
<td>189</td>
<td>222</td>
</tr>
<tr>
<td>5</td>
<td>space</td>
<td>124</td>
<td>157</td>
<td>190</td>
<td>223</td>
</tr>
<tr>
<td>6</td>
<td>33</td>
<td>125</td>
<td>158</td>
<td>191</td>
<td>224</td>
</tr>
<tr>
<td>7</td>
<td>126</td>
<td>159</td>
<td>192</td>
<td>225</td>
<td>226</td>
</tr>
<tr>
<td>8</td>
<td># 127</td>
<td>160</td>
<td>193</td>
<td>226</td>
<td>227</td>
</tr>
<tr>
<td>9</td>
<td>$$ 128</td>
<td>161</td>
<td>194</td>
<td>226</td>
<td>227</td>
</tr>
<tr>
<td>10</td>
<td>% 129</td>
<td>162</td>
<td>195</td>
<td>228</td>
<td>228</td>
</tr>
<tr>
<td>11</td>
<td>&amp; 130</td>
<td>163</td>
<td>196</td>
<td>229</td>
<td></td>
</tr>
</tbody>
</table>

• The set $\Sigma^*$ of strings over the alphabet $\Sigma$ is defined by
  – Basis: $\varepsilon \in \Sigma^*$ ($\varepsilon$ is the empty string)
  – Recursive: if $w \in \Sigma^*$, $a \in \Sigma$, then $wa \in \Sigma^*$

$\varepsilon^*$ Basis, $a \forall a \in \varepsilon$, $w^*$ Recursive, $w \in \varepsilon^*$, $wa \in \varepsilon^*$
function definitions on recursively defined sets

Length:
len(ℇ) = 0;
len(𝐰𝚊) = 1 + len(w); for 𝐯 ∈ 𝜔*, 𝐚 ∈ 𝜔

Reversal:
ℇᴿ = ℇ
𝐰𝐚ᴿ = 𝐚𝐰ᴿ for 𝐯 ∈ 𝜔*, 𝐚 ∈ 𝜔

Concatenation:
x • ℇ = 𝑥 for 𝑥 ∈ 𝜔*
x • 𝐰𝐚 = (𝑥 • 𝐰)𝐚 for 𝑥, 𝐯 ∈ 𝜔*, 𝐚 ∈ 𝜔
rooted binary trees

- **Basis:**
  - is a rooted binary tree

- **Recursive step:**
  - If $T_1$ and $T_2$ are rooted binary trees,
    - then so is: $T_1 \cup T_2$
defining a function on rooted binary trees

**Basis**
- \( \text{size}(\cdot) = 1 \)

**Recursive**
- \( \text{size}\left( \begin{array}{c} T_1 \\ T_2 \end{array} \right) = 1 + \text{size}(T_1) + \text{size}(T_2) \)
- \( \text{height}(\cdot) = 0 \)
- \( \text{height}\left( \begin{array}{c} T_1 \\ T_2 \end{array} \right) = 1 + \max\{\text{height}(T_1), \text{height}(T_2)\} \)
How to prove $\forall x \in S, P(x)$ is true:

**Base Case:** Show that $P(u)$ is true for all specific elements $u$ of $S$ mentioned in the *Basis step*.

**Inductive Hypothesis:** Assume that $P$ is true for some arbitrary values of *each* of the existing named elements mentioned in the *Recursive step*.

**Inductive Step:** Prove that $P(w)$ holds for each of the new elements $w$ constructed in the *Recursive step* using the named elements mentioned in the Inductive Hypothesis.

**Conclude** that $\forall x \in S, P(x)$.

*structural induction*
Let $S$ be a set of strings over $\Sigma = \{a, b\}$ defined by

**Basis:** $a \in S$

**Recursive:**

If $w \in S$ then $wa \in S$ and $wba \in S$

If $u, v \in S$ then $uv \in S$

**Claim:** If $w \in S$ then $w$ has more $a$'s than $b$'s.

Base Case: $P(a)$ holds because $a$ has more $a$'s than $b$'s.

IH: $P(w), P(ua), P(v)$ hold for some $w, u, v \in S$

IS: $\#_a(wa) = 1 + \#_a(w) > 1 + \#_b(w) > \#_b(w) \Rightarrow P(wa)$

$\#_a(wba) = 1 + \#_a(w) > 1 + \#_b(w) = \#_b(wba) \Rightarrow P(wba)$
 proof continued?

\[ \#_a (uv) = \#_a (u) + \#_a (v) > \#_b (u) + \#_b (v) \]

\[ \text{IH } p(u) \land p(v) \]
\[ = \#_b (uv) \]
\[ =) p(uv) \text{ holds} \]

Conclusion: p(w) holds for all w ∈ S.
prove: \( \text{len}(x \cdot y) = \text{len}(x) + \text{len}(y) \) for all \( x, y \in \Sigma^* \)

Let \( P(y) \) be “\( \text{len}(x \cdot y) = \text{len}(x) + \text{len}(y) \) for all \( x \in \Sigma^* \)”

Base Case: \( P(\varepsilon) \) holds
\[
\text{len}(x \cdot \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon) \checkmark
\]
\[ \text{def.} \quad \text{def of len} \]

IH: \( P(y) \) holds for some \( y \in \Sigma^* \)

IS: \( \forall a \in \Sigma, \ P(ya) \) holds.

Fix \( x \in \Sigma^* \)
\[
\text{len}(x \cdot ya) = \text{len}((x \cdot y)a) = \text{len}(x \cdot y) + 1
\]
\[ \text{def.} \quad \text{def of len} \]

IH \( \Rightarrow \)
\[
= \text{len}(x) + \text{len}(y) + 1
\]
\[ = \text{len}(x) + \text{len}(ya)
\]
\[ \text{def of len} \Rightarrow P(ya) \text{ holds} \]

Length:
\[
\text{len} (\varepsilon) = 0;
\text{len} (wa) = 1 + \text{len}(w); \text{ for } w \in \Sigma^*, a \in \Sigma
\]
defining a function on rooted binary trees

- \( \text{size}(\cdot) = 1 \)

- \[
\text{size} \left( \begin{array}{c}
T_1 \\
\cdot \\
T_2 
\end{array} \right) = 1 + \text{size}(T_1) + \text{size}(T_2)
\]

- \( \text{height}(\cdot) = 0 \)

- \[
\text{height} \left( \begin{array}{c}
T_1 \\
\cdot \\
T_2 
\end{array} \right) = 1 + \max\{\text{height}(T_1), \text{height}(T_2)\}
\]
Claim: For every rooted binary tree $T$, $\text{size}(T) \leq 2^{\text{height}(T)+1} - 1$

Base Case: $P(\cdot)$

$\text{size}(\cdot) = 1 \leq 2^{0+1} - 1 = 2 - 1 = 1$

IH: $P(T_1), P(T_2)$ hold for some $T_1, T_2$.

IS: $P(T_3)$ holds

Size: 7

Height: 2

$2^{2+1} - 1 = 7$
Sets of strings that satisfy special properties are called languages.

Examples:

- English sentences
- Syntactically correct Java/C/C++ programs
- $\Sigma^* =$ All strings over alphabet $\Sigma$
- Palindromes over $\Sigma$
- Binary strings that don’t have a 0 after a 1
- Legal variable names, keywords in Java/C/C++
- Binary strings with an equal # of 0’s and 1’s