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

\[
\Sigma = \{
\]

• 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 \( wa \in \Sigma^* \)
**Length:**

\[
len(\varepsilon) = 0; \\
len(wa) = 1 + len(w); \text{ for } w \in \Sigma^*, a \in \Sigma
\]

**Reversal:**

\[
\varepsilon^R = \varepsilon \\
(wa)^R = aw^R \text{ for } w \in \Sigma^*, a \in \Sigma
\]

**Concatenation:**

\[
x \cdot \varepsilon = x \text{ for } x \in \Sigma^* \\
x \cdot wa = (x \cdot w)a \text{ for } x, w \in \Sigma^*, a \in \Sigma
\]
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

• \( \text{size}(\bullet) = 1 \)

\[
\text{size} \left( \begin{array}{c}
\text{T}_1 \\
\text{T}_2
\end{array} \right) = 1 + \text{size}(\text{T}_1) + \text{size}(\text{T}_2)
\]

• \( \text{height}(\bullet) = 0 \)

\[
\text{height} \left( \begin{array}{c}
\text{T}_1 \\
\text{T}_2
\end{array} \right) = 1 + \max\{\text{height}(\text{T}_1), \text{height}(\text{T}_2)\}
\]
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 $\forall x \in S, P(x)$
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.
proof continued?
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 \(x \in \Sigma^*\)"

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

• size(\( \cdot \)) = 1

• \[
\text{size } \left( \begin{array}{c}
\text{T}_1 \\
\text{T}_2
\end{array} \right) = 1 + \text{size}(\text{T}_1) + \text{size}(\text{T}_2)
\]

• height(\( \cdot \)) = 0

• \[
\text{height } \left( \begin{array}{c}
\text{T}_1 \\
\text{T}_2
\end{array} \right) = 1 + \max\{\text{height}(\text{T}_1), \text{height}(\text{T}_2)\}\]
Claim: For every rooted binary tree $T$, $\text{size}(T) \leq 2^{\text{height}(T)+1} - 1$
Sets of strings that satisfy special properties are called languages.

Examples:
- English sentences
- Syntactically correct Java/C/C++ programs
- $\Sigma^* = \text{All strings over alphabet } \Sigma$
- 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