Proof techniques: construction, contradiction, counterexample, case enumeration, induction, pigeonhole principle, proving cardinality, diagonalization

Languages, Strings

Alphabet: finite set of symbols
String: finite sequence of symbols from alphabet
Replication: $w^0 = \varepsilon, w^{i+1} = w^i w$
Reverse: $w^R = w = \varepsilon$ if $|w| = 0$, else $\exists a \in \Sigma$ and $\exists u \in \Sigma^*$ such that $w = ua$, then define $w^R = au^R$
$\boldsymbol{(wx)^R = x^r w^r}$: induction on $|x|$, base case $|x| = 0$ so $x = \varepsilon$, consider any string x where $|x| = n + 1$, then $x = ua$ for some character a and $|u| = n$, so $(wx)^R = (w(ua))^R = (wu)a^R = a(wu)^R = a(u^R w^R) = (au^R)w^R = (ua)^R w^R = x^R w^R$
Language: set of strings (finite/infinite) from alphabet, uncountably infinite number of these (power set of Σ^*)
$\boldsymbol{\Sigma^*}$: countably infinite with non-empty alphabet, enumerate with lexicographic order
$\boldsymbol{L_1 L_2}$: $\{w \in \Sigma^* : \exists s \in L_1(\exists t \in L_2(w = st))\}$
$\boldsymbol{L^*}$: $\{\varepsilon\} \cup \{w \in \Sigma^* : \exists k \geq 1(\exists w_1, w_2, \ldots w_k \in L(w = w_1 w_2 \ldots w_k))\}$ or $L^0 \cup L^1 \cup L^2 \cup \ldots$
$\boldsymbol{L^+}$: LL^* or $L^* - \{\varepsilon\}$ iff $\varepsilon \notin L$ or $L^0 \cup L^1 \cup L^2 \cup \ldots$
$\boldsymbol{(L_1 L_2)^R = L_2^R L_1^R}$: $\forall x(\forall y((xy)^R = y^R x^R))$ from before, then $(L_1 L_2)^R = \{(xy)^R : x \in L_1 \text{ and } y \in L_2\} = \{y^R x^R : x \in L_1 \text{ and } y \in L_2\} = L_2^R L_1^R$
Decision problem: problem to which answer is yes/no or true/false
Decision procedure: answers decision problem
Machine power hierarchy: FSM (regular), PDA (context-free), TM (semi-decidable & decidable)
Rule of least power: use least powerful language suitable for expressing info, constraints or programs on WWW
$\boldsymbol{firstchars(L)}$: $\{w : \exists y \in L(y = cx \wedge c \in \Sigma_L \wedge x \in \Sigma_L^* \wedge w \in c^*)\}$, closed under FIN but not INF (since result is first character *)
$\boldsymbol{chop(L)}$: $\{w : \exists x \in L(x = x_1 c x_2, x_1 \in \Sigma_L^*, x_2 \in \Sigma_L^*, c \in \Sigma_L, |x_1| = |x_2|$, and $w = x_1 x_2)$
, language where all strings have exact middle character removed, must have had odd length to begin with, closed under FIN but not INF (can get empty set if never odd length)
Extra: to describe language with at least 2 different substrings of length 2 $L = \{w \in \{a, b\}^* : \exists x, y(x \neq y \wedge |x| = 2 \wedge |y| = 2 \wedge Substr(x, w) \wedge Substr(y, w))\}$