Structural Induction

Recursively (inductively) defined sets

- Define $\{0,1\}^*$ = set of all finite bit strings
  - Base case λ_0 = the empty string
  - Constructor case $s \in \{0,1\}^*$ IMLIES $S_0 \in \{0,1\}^*$ AND $S_1 \in \{0,1\}^*$
    - Ex. $s = 001, s_0 = 0010, s_1 = 0011$
    - If Σ is a finite set of letters, then Σ^* is the set of all words that's in Σ

- Define brkts by the finite string of matched brackets $\subseteq \{[,]\}$
  - Base case $\lambda \in$ Brkts
  - Constructor case $s \in$ Brkts IMPLIES $[s] \in$ Brkts, $(s \in \text{Brkts}.\ t \in \text{Brkts})$IMPLIES $st \in$ Brkts
    - Alternatively $(s \in \text{Brkts}.\ t \in \text{Brkts})$IMPLIES $[s]t \in$ Brkts
    - E.x. s = a", t="b, ["a]"b" $\in$ Brkts

- Define S = Set of syntactically correct formulas of propositional logic
  - Base case propositional variables are in S
  - Constructor case $(f \in S.\ f' \in S)\text{IMPLIES} \begin{pmatrix} (f \text{ AND } f') \in S \text{ (binary)} \\ \text{AND } (f \text{ OR } f') \in S \text{ (binary)} \\ \text{AND } (\text{NOT } f) \in S \text{ (unary)} \end{pmatrix}$

- Define M = set of syntactically correct moniton formulas of propositional logic
  - Base case propositional variables are in S
  - Constructor case $(f \in S.\ f' \in S)\text{IMPLIES}\big((f \text{ AND } f') \in S \text{ AND } (f \text{ OR } f') \in S\big)$
  - M is the smallest set that containing the base cases and closed under the constructor cases

- Define natural numbers
  - Base case $0 \in \mathbb{N}$
  - Constructor case $n \in \mathbb{N}$ IMPLIES $n + 1 \in \mathbb{N}$

- Define binary relations recursively
  - Base case $(0,0) \in \mathbb{N} \times \mathbb{N}$
  - Constructor case: $(m,n) \in \mathbb{N} \times \mathbb{N}$ IMPLIES($(m+1,n) \in \mathbb{N} \times \mathbb{N}$ AND $(m,n+1) \in \mathbb{N} \times \mathbb{N}$)

Structural Induction

Let $P: S \to \{T, F\}$ be a predicate, where S is a recursively defining set
Prove P(s) for all base cases of the definition
Prove P(s) for the constructor cases assuming it's true for the component
$\forall s \in S.\ P(s)$ strong induction

Justifying the correctness of structural induction

Let E_0 be the element of S because of the base case
Let E_i be the element of S obtained from the elements of E_0 by applying the constructor cases i times
$(E_i = \{ x \in P(S) \mid |x| = i \})$
When we do structural induction, we are performing strong induction on the size of the elements of S.

Thrm define M set of moninton propositional formulas, $\forall f \in M.\ N_v(f)$ = #occurance of propositional variables, $N_c(f)$ = #occurance of propositional connectives. $N_v(f) = 1 + N_c(f)$
Proof For all $f \in M$, let P(f) defined $N_v(f) = 1 + N_c(f)$

Let f be a propositional variable, then $N_v(f) = 1, N_c(f) = 0$
Let $f = (f' \text{ OR } f'')$, assume $P(f')$, assume $P(f'')$
$N_v(f) = N_v(f') + N_v(f'') = 1 + N_c(f') + 1 + N_c(f'') = 1 + \big(N_c(f') + N_c(f'') + 1\big)$
$= 1 + N_c(f)$
Similarly, P(f) holds for $f = (f' \text{ AND } f'')$
$\forall f \in M.\, P(f)$ structural induction

Thrm recursively define B: the set of binary trees.
Base case: The empty tree $\perp \in B$
Constructor case: $t_1 \in B, t_2 \in B$, r is a root IMPLIES $t = t_1 \leftarrow r \rightarrow t_2 \in B, t_1 = \text{left}(t), t_2 = \text{right}(t)$
Define $N(l) = \#\text{nodes in } l$
Base case: $N(\perp) = 0$
Constructor case: $N(t) = 1 + N\big(\text{left}(t)\big) + N\big(\text{right}(t)\big)$
Define $L(l) = \#\text{leaves of the tree}$
Base case: $L(\perp) = 0, L(f) = 1$ if f is a tree with one node $(N(f) = 1)$
Constructor case: $L(f) = L\big(\text{left}(f)\big) + L\big(\text{right}(f)\big)$
Then, a binary tree with n nodes has at most $\left\lceil \frac{n}{2} \right\rceil$.
Proof For all $t \in B$, let P(t) defined $L(t) \leq \left\lceil \frac{N(t)}{2} \right\rceil$
Let $t = \perp, L(t) = 0 \leq 0 = \left\lceil \frac{0}{2} \right\rceil = \left\lceil \frac{N(t)}{2} \right\rceil$
Let t be the binary tree with only one node, $L(t) = 1 = \left\lceil \frac{1}{2} \right\rceil = \left\lceil \frac{N(t)}{2} \right\rceil$
P(t) holds for base cases
Let $t \in B, N(t) > 1$ be arbitrary, assume $P\big(\text{left}(t)\big)$ AND $P\big(\text{right}(t)\big)$, hence

$$L(t) = L\big(\text{left}(t)\big) + L\big(\text{right}(t)\big) \leq \left\lceil \frac{N\big(\text{left}(t)\big)}{2} \right\rceil + \left\lceil \frac{N\big(\text{right}(t)\big)}{2} \right\rceil \text{ (IH)}$$
$$\leq \frac{N\big(\text{left}(t)\big)+1}{2} + \frac{N\big(\text{right}(t)\big)+1}{2} \text{ (by definition of ceiling)}$$
$$\leq \frac{N(t)+1}{2}$$

Since $L(t) \in \mathbb{N}$ by definition, $L(t) \leq \left\lfloor \frac{N(t)+1}{2} \right\rfloor \leq \left\lceil \frac{N(t)}{2} \right\rceil$
P(t) holds for constructor cases
$\forall t \in B.\, P(t)$ structural induction

Induction vs Contradiction
Sometimes a proof by induction can be disguised as a proof by contradiction
Thrm every integer greater than 1 can be expressed as a product of primes
Suppose the statement is false, let n be the smallest integer greater than 1 that can't be expressed as a product of primes.
n is not prime because any prime number is a product of itself.
Therefore, $\exists k \in \mathbb{N}.\, \exists m \in \mathbb{N}.\, 1 < k < n \text{ AND } 1 < m < n \text{ AND } km = n$ by the definition of not prime
Because n is the smallest integer that is greater than 1 and can't be expressed as product of primes, k,m must be expressed as a product of primes, then $n = km$ are a product of primes
By contradiction, every integer greater than 1 can be expressed as a product of primes