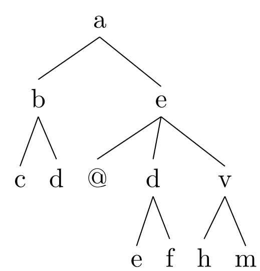
Expectation-Maximization in FSTA
The skeleton

I. Definitions

- Ranked alphabet: $\Sigma = \bigcup_k \Sigma_k$
  $\forall i \geqslant 0$, $\Sigma_i \subseteq \Sigma$
  - x[] $\in T_\Sigma$ if $x \in \Sigma_0$
  - x[t_1, t_2, t_3...t_k] $\in T_\Sigma$ if $x \in \Sigma_k$ and t_1, t_2, t_3...$t_k \in T_\Sigma$

- . $\quad\rightarrow$ context c: the context at the position of node n.



Context 1 = Nonroot (Node e) (context 2) ([]) ([$\underset{\displaystyle t_d}{\underset{\uparrow}{\boxed{\text{d[e[], f[]]}}}}$, $\underset{\displaystyle t_v}{\underset{\uparrow}{\boxed{\text{v[h[], m[]]}}}}$])

Context 2 = Nonroot (Node a) (context 3) ([$\underset{\displaystyle t_b}{\underset{\uparrow}{\boxed{\text{b[c[], d[]]}}}}$]) ([])

Context 3 = Root

II. Boolean Derivational Path

For any FSTA G with $G = \langle$Q, Σ, I, $\Delta\rangle$.
- Q: a finite set of states
- F: a finite set of initial state with $I \in Q$
- Δ: $\bigcup_k Q^k \times \Sigma_k \times Q$

Define `under`: $T_\Sigma \times Q \rightarrow \{0, 1\}$
Given root node is at state q, whether this tree can be generated by this grammar G

- `under`(x[])(q) = $\Delta([], x, q)$ when $x \in \Sigma_0$
- `under`(x[t_1, t_2, t_3...t_k])(q) = $\bigvee_{q_1 \in Q, q_2 \in Q ... q_k \in Q}$ `under`(t_1, q_1) $\wedge$ `under`(t_2, q_2) $\wedge$... $\wedge$`under`(t_k, q_k) $\wedge$ $\Delta([q_1, q_2 .... q_k], x, q)$

Define `over`: `context` $\times Q \rightarrow \{0, 1\}$
Given a node in a tree is at state q, whether such node can have this context in grammar G

- `over`(a)(q) = I(q) if a = `root`
- `over`(Nonroot (a) (context a') ([l_1, l_2, l_3...l_k]) ([r_1, r_2, r_3...r_m])(q)
  = $\bigvee_{q_1, q_2 ... q_k \in Q} \bigvee_{p_0 \in Q} \bigvee_{p_1, p_2, ... p_m \in Q}$ `under`(l_1, q_1) $\wedge$ `under`(l_2, q_2) $\wedge$... $\wedge$`under`(l_k, q_k)
  $\wedge$`under`(r_1, p_1) $\wedge$ `under`(r_2, p_2) $\wedge$... $\wedge$`under`(r_m, p_m)
  $\wedge$ `over` (context a', p_0)
  $\wedge\Delta([q_1, q_2 .... q_k, q, p_1, p_2 .... p_m], a, p_0)$

III. Probabilitistic Over and Under

- PFSTA

– ⟨Q, Σ, I, Δ⟩
  - Q: a finite set of states
  - I: $Q \to [0,1]$
  - Δ: $\bigcup_k (Q \times \Sigma_k \times Q^k \to [0,1])$

– *Normalized* iff
  - For all states q_0, the total probability of leaving q_0 is 1

$$\sum_{\sigma \in \Sigma_k} \sum_{p^k} \Delta(q_0, \sigma, p^k) = 1$$

  - [I(q_1), I(q_2)..., I(q_n)] is a stochastic vector

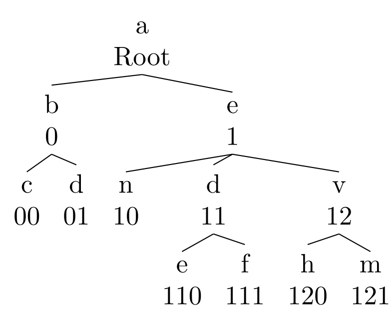
- For any FSTA G with G = ⟨Q, Σ, I, Δ⟩, define `under`: $T_\Sigma \times Q \to$ [0, 1]
  Given the root node is at state q, the probability this tree can be generated by this grammar G
  - `under`(x[])(q) = $\Delta(q, x, [])$ when $x \in \Sigma_0$
  - `under`(x[t_1, t_2, t_3...t_k])(q) = $\sum_{q_1 \in Q, q_2 \in Q \ldots q_k \in Q}$ ($\prod_{i=1}^{k}$ `under`(t_i, q_i)) × $\Delta(q, x, [q_1, q_2 \ldots. q_k])$

  Define `over`: `context` $\times Q \to$ [0, 1]
  Given a node in a tree is at state q, the probability of such node can have a specific context in grammar G
  - `over`(a)(q) = I(q) if a = `root`
  - `over`(Nonroot (a) (context a') ([l_1, l_2, l_3...l_k]) ([r_1, r_2, r_3...r_m])(q)
    = $\sum_{q_1, q_2 \ldots q_k \in Q} \sum_{p_0 \in Q} \sum_{p_1, p_2, \ldots p_m \in Q}$ ($\prod_{i=1}^{k}$ `under`(l_i, q_i)) × ($\prod_{j=1}^{m}$ `under`(r_i, p_i))
    × $\Delta(p_0, a, [q_1, q_2 \ldots. q_k, q, p_1, p_2 \ldots. p_m])$ × `over`(a', p_0)

IV. Necessary functions

- `NodeIndex` a = Root | Non-Root Int

  Extending Huffman encoding: n-nary branching tree: {0, 1, 2, 3, 4...n}



- getContext :: `Tree` → `NodeIndex` → `Context`
- getSubtree::`Tree` → `NodeIndex` → [`Tree`]
- getLabel :: `Tree` → `NodeIndex` → `Node`
- isLabel :: `Tree` → (`Node`, `NodeIndex`) → Int
- indexList:: `Tree` → [(`NodeIndex`, `Node`)]

V. EM in FSTA

- Problem Setting:
  - The complete data: given a set of annotated trees **with the states information** of each nodes, estimate the distribution of Δ and I to maximize the likelihood of the observed data
  - The incomplete data: given a set of annotated trees, estimate the probability distribution of Δ and I with the tree-set.
- Formula
  - The MLE in PFSTA (Following Chi & Geman 1997)
    Given the set of annotated trees $\{t_1, t_2, t_3 ... t_n\}$ along with states information, the likelihood is

$$\begin{aligned} L &= L(p; t_1, t_2, t_3 .... t_n) \\ &= \prod_{i=1}^{n} \prod_{trans \in Q \times \Sigma_k \times Q^k} p(trans)^{freq(trans; w_i)} \\ log(L) &= \Sigma_r \Sigma_{i=1}^{n} freq(trans; w_i) log(p(trans)) \end{aligned}$$

$\forall q \in Q$, I(q)=$\frac{count(Q_0=q)}{\Sigma_{q' \in Q} count(Q_0=q')}$

$$\Delta(q_1, x, [p_1, p_2, p_3 .... p_k]) = \frac{count(q_1, x, [p_1, p_2, p_3 .... p_k])}{\Sigma_{s_1, s_2 ... s_k \in Q} \Sigma_{x' \in \Sigma_k} count(q_1, x', [s_1, s_2, s_3 .... s_k])}$$

  - Expected count
    $\forall q \in Q$, $\texttt{ecount}(Q_0 = q \mid T)$=Pr$(Q_0 = q \mid T) = \frac{I(q) \times \texttt{under(T)(q)}}{Pr(T)}$

$$\begin{aligned} &\texttt{ecount}(q, x, [p_1, p_2, p_3 .... p_k] \mid T) \\ &= \Sigma_{i \in \{Index\}} \Pr(\texttt{isLabel(i, x)}, Q_i = q, Q_{i+1} = [p_1, ... p_k] \mid T) \\ &= \Sigma_{i \in \{Index\}} \frac{\texttt{isLabel(i, x)} \times over(\texttt{getContext}(T,i), q) \times \Delta(q, x, [p_1, p_2, p_3 .... p_k]) \times \prod_{j \in \texttt{getSubtree(T, i)}} under(j, p_j)}{Pr(T)} \end{aligned}$$

- Algorithm: Initialize the parameters of Δ and I
  while not converge (←need threshold)
    Calculate the expected counts of each rule under such parametrization
    Update the parameters using the expected counts and MLE