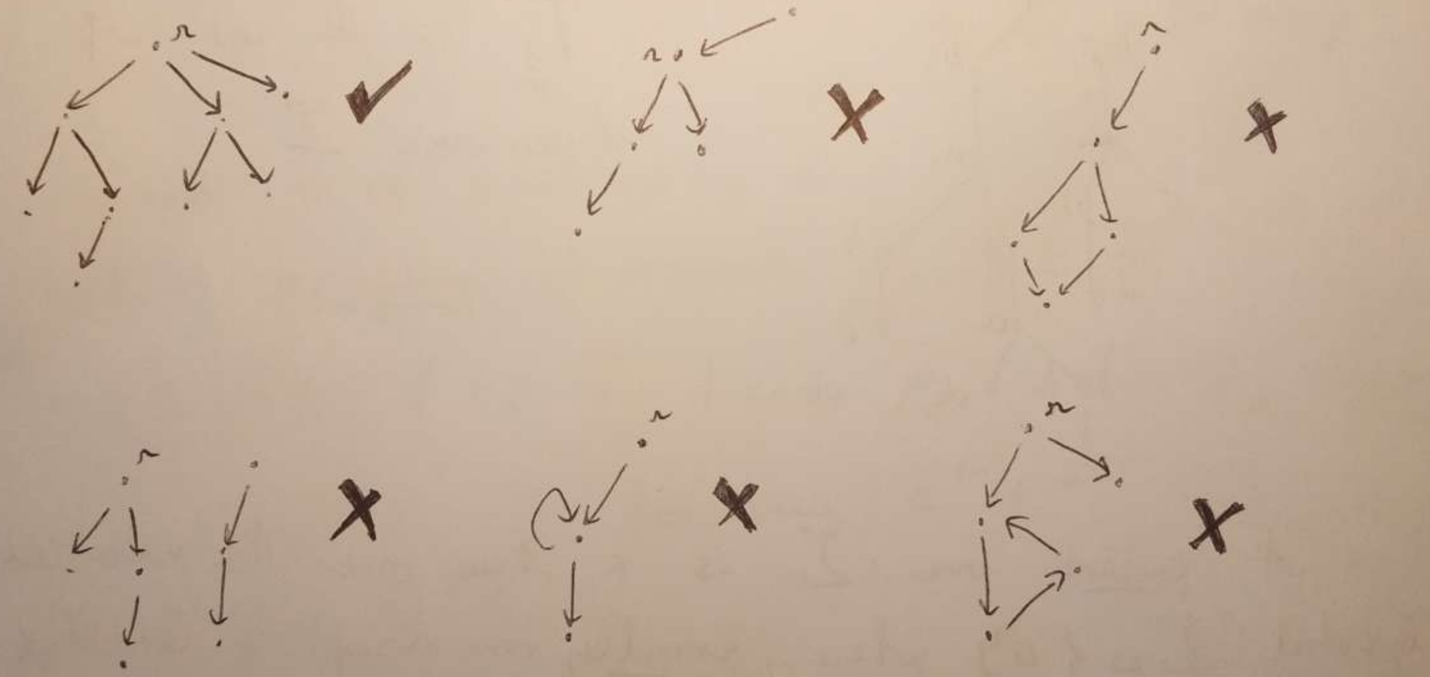
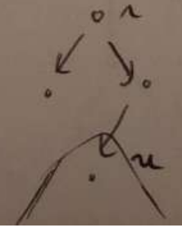
Chapter 5 - Automata (and logic) over finite trees

I] Labelled trees

• $G = \langle V, \to \rangle$ a $\underbrace{\text{directed graph}}^{\text{finite}}$ and $r \in V$, G is a <u>tree rooted in r</u> iff for every $v \in V$, there is a <u>unique</u> path $\underset{u_0}{\overset{\|}{r}} \to u_1 \to u_2 \to \dots \to u_{n-1} \to \underset{v}{\overset{\|}{u_n}}$ in G.



- r is the root
- $u \to v$, v child of u, ordered!
- leaf (no child)
- descendant $u \to^{+} v$
- elements of G are nodes
- subtree $N|_u$



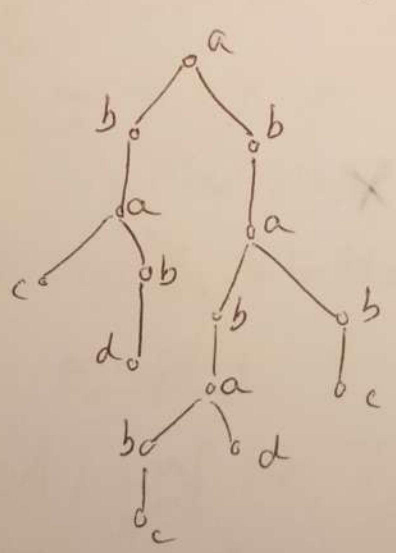
- depth of a tree / a node
- lex order

• <u>ranked alphabet</u>: every letter has an arity $^{(n)}$

ex: $\Sigma = \{ a^{(2)}, b^{(1)}, c^{(0)}, d^{(0)} \}$

• tree over Σ: map from some $\overbrace{\text{tree}}^{\text{shaped graph}}$ to Σ such that every n mapped to $a \in \Sigma$ has $ar(a)$ children (in particular it is a leaf iff a has arity 0 (i.e. is nullary)). We identify the map $t : \mathrm{Nodes}(G) \to \Sigma$ with the graph itself. (and therefore write $\mathrm{Nodes}(t)$, etc.)

<u>ex</u>: Σ as above,

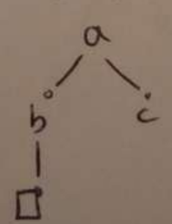
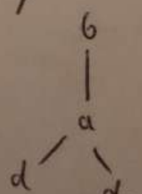
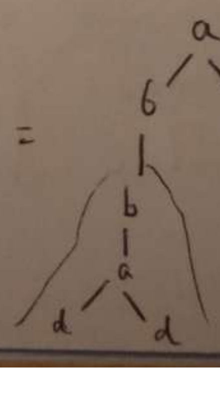


T_Σ is the set of trees over Σ

• A <u>context</u> over Σ is a tree over the extended alphabet $\Sigma \sqcup \{\square^{(0)}\}$ where <u>exactly</u> one leaf is labelled by the nullary letter $\square$.

$\mathcal{C}_\Sigma$ is the set of contextes over Σ.

• $C \in \mathcal{C}_\Sigma$, $t \in T_\Sigma$, $C[t]$ is the tree obtained by plugging t at the blank symbol of C

<u>ex</u>: $C =$  , $t =$  $C[t] =$ 

• similar operation $C[\;] : \mathcal{C}_\Sigma \to \mathcal{C}_\Sigma$ (t can actually be a context)

__Fact__: $(C_1[C_2])[C_3] = C_1[C_2[C_3]]$ $\square$

• definition C^n makes sense (with $C^0 = \square$)

context with only one leaf, labelled by $\square$.

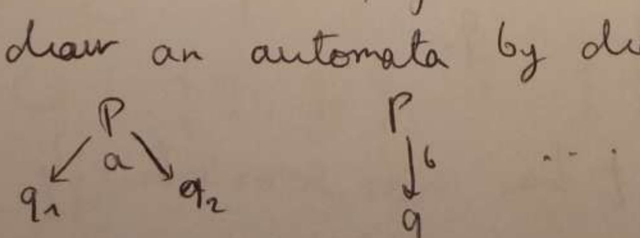
II] Automata

α) Top-down automata

• Top-down automata over the ranked alphabet Σ is a tuple $\mathcal{A} = \langle Q, I, (\delta_a)_{a \in \Sigma}, (F_a) \rangle$ where

- Q finite set of __states__;
- $I \subseteq Q$ set of __initial states__;
- for every a of arity $n \geq 1$, $\delta_a \subseteq Q \times Q^n \simeq Q^{n+1}$ is the set of __transitions__ for a;
- $F_a \subseteq Q$ set of final states for every a __nullary__

We draw an automata by drawing the set of transitions



~~(- for every a of arity 0, $\delta a \subseteq Q$ is the set of empty transitions for a)~~

- $t \in T_\Sigma$, a run over $\mathcal{A}$ via t is a mapping ~~such that~~ $\rho_t : Nodes(t) \to Q$ such that
  - $\rho(root) \in I$
  - for every $u \in Nodes(t)$ ~~of an~~ with $n \geq 1$ children, labelly by a, $\begin{tikzcd} & \rho(u) \arrow[dl] \arrow[d] \arrow[dr] & \\ \rho(u_0) & \rho(u_1) \; \cdots & \rho(u_{n-1}) \end{tikzcd}$ $\in \delta_a$.

$u_0, u_1, u_2 \dots, u_{n-1}$

It is accepting if moreover $\rho(\ell) \in F_a$ for every $\ell \in Leaves(t)$ labelled by a.

$$\mathcal{L}(\mathcal{A}) = \{ t \in T_\Sigma \mid \text{exists } \rho_t : t \xrightarrow{run} Q_{\mathcal{A}} \text{ accepting} \}$$

ex: $\Sigma = \{ a^{(2)}, b^{(2)}, \ell^{(0)} \}$

$Q = \{q_s, q_v\}$, $I = \{q_s\}$, $F_\ell = \{q_v\}$

$$\begin{tikzcd} & q_s \arrow[dl] \arrow[dr] & \\ q_s & a & q_v \end{tikzcd} \quad \begin{tikzcd} & q_s \arrow[dl] \arrow[dr] & \\ q_v & a & q_s \end{tikzcd} \quad \begin{tikzcd} & q_s \arrow[dl] \arrow[dr] & \\ q_v & b & q_v \end{tikzcd}$$

$$\begin{tikzcd} & q_v \arrow[dl] \arrow[dr] & \\ q_v & a & q_v \end{tikzcd} \quad \begin{tikzcd} & q_v \arrow[dl] \arrow[dr] & \\ q_v & b & q_v \end{tikzcd}$$

$\mathcal{L}(\mathcal{A}) = \{ t \in T_\Sigma \mid t \text{ has a } b \text{ somewhere} \}$

$\mathcal{A}$ deterministic if for every $a^{(n)} \in \Sigma$, $\delta_{\mathcal{A},a}$ is a function from Q to Q^n.

B) Bottom-up automaton

- $\mathcal{A} = \langle Q, (I_a), (\delta_a)_{a \in \Sigma}, F \rangle$ where
  - Q states
  - I_a initial states for every a nullary
  - $\delta_a \subseteq Q^n \times Q \cong Q^{n+1}$ transitions for a
  - F final states

A run over $\mathcal{A}$ on t is a mapping $\rho_\uparrow : \mathrm{Nodes}(t) \to Q$
such that
- $\rho(v) \in I$ for every leaf v of t
- $\begin{array}{c} \rho(u) \\ \nearrow \; \nearrow \quad a \quad \nwarrow \\ \rho(u_0) \; \rho(u_1) \quad \rho(u_{n-1}) \end{array} \in \delta_a$ for every

inside node u ~~of~~ labelled by a.

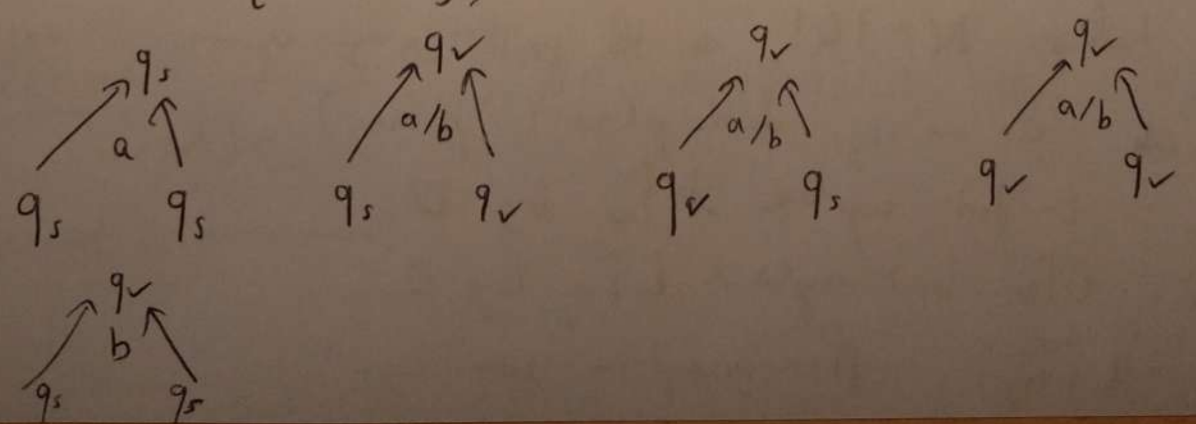
It is accepting if moreover $\rho(r) \in F$.

$$\mathcal{L}(\mathcal{A}) = \{ t \in T_\Sigma \mid \text{exists } \rho_\uparrow : t \xrightarrow{\text{run}} Q_\mathcal{A} \text{ accepting} \}$$

$\mathcal{A}$ deterministic if for every $a^{(n)} \in \Sigma$, δ_a is a function from Q^n to Q.

ex: $\Sigma = \{ a^{(2)}, b^{(2)}, \ell^{(0)} \}$

$Q = \{ q_s, q_v \}$, $I = \{ q_s \}$, $F = \{ q_v \}$



γ) Equivalences

prop: $L \subseteq T_\Sigma$ recognised by an NTDA iff it is reco by a NBUA.

proof: the same in reverse...

prop: Any L recognisable (i.e. reco by a NTDA or a NBUA) is recognisable by a DBUA.

proof / algo: powerset, as for finite words

non prop: $DTDA \subset NTDA$.

III) Pumping lemma

Pumping lemma: for every L recognisable, there exists N s.t. for every tree $t \in L$, every $v \in Leaf(t)$ of depth $\geq N$, t can be written $C[D[t']]$ such that:

- $D \neq \square$
- $v \in t'$
- $C[D^n[t']] \in L$ for every $n \in \mathbb{N}$.

proof: take $N = |Q|$; on the path $r \to u_1 \to \dots \to v$, $\; v = u_n$, must exist $u_i \to^* u_j$ with $\rho(u_i) = \rho(u_j)$

$C = t$ but replace $t\restriction_{u_i}$ by $\square$,

$D = t\restriction_{u_i}$ but replace $t\restriction_{u_j}$ by $\square$,

$t' = t\restriction_{u_j}$. Then pump as you can!

application: $L = \{ t \in T_\Sigma \mid$ every branch of t has prime length $\}$

is not ~~definable~~ recognisable.

IV] Monadic Second-Order Logic

Here, we assume that $\Sigma = \{ a^{(2)}, a^{(0)}, b^{(2)}, b^{(0)} \dots \}$, we identify $a^{(2)}$ with $a^{(0)}$. Hence, a tree over Σ is binary.

α) Syntax and semantics

$$\varphi := a(x) \mid L(x,y) \mid R(x,y) \mid x \leq_{lex} y \mid \exists x.\ \varphi \mid \exists X.\ \varphi \mid \neg \varphi \mid \varphi \vee \psi$$

models are trees over Σ, FO variables are interpreted as nodes of the tree.

ex: $\mathrm{leaf}(x) := \forall y.\ \neg L(x,y) \wedge \neg R(x,y)$

$\varphi := \exists y.\ c(y) \wedge \forall x.\ x \leq_{lex} y \rightarrow a(x)$

$\mathcal{L}(\varphi)$

β) Correspondence with automata

Theorem: $L \subseteq T_\Sigma$ is recognisable iff it is definable in MSO.

proof: as usual