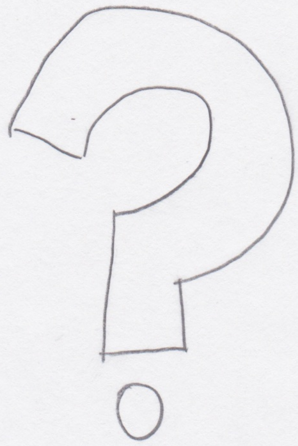
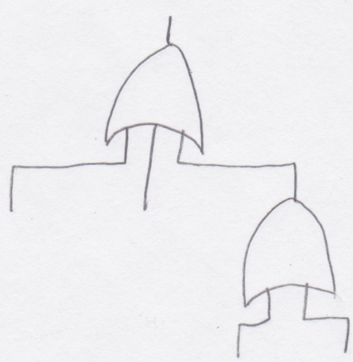


What is Alternation?

- Alternation is a generalization of nondetermenism.
- To understand alternation, we must first look at nondeterminism in detail.

Nondeterminism

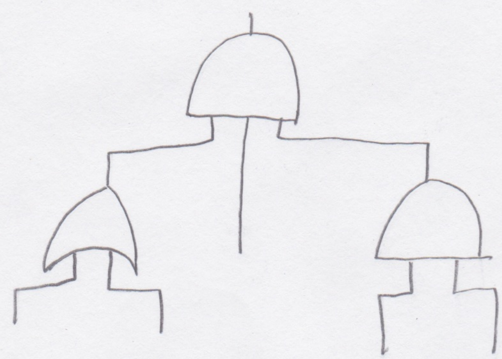
- When using a nondetermenistic algorithm, each time the possabilities branch out, the algorithm is asking if there exists any branch of the algorithm.
- Because nondetermenism asks the question if there exists any branch that resolves to true, each internal node of the tree can use the existential quantifier and ask if $\exists$ any children that resolve to true.
- To make it easier to understand, each node in a nondetermenistic tree could be thought of as an OR gate!



- If any children are true, the parent will be as well.

Alternation

- Alternation expands on nondetermenism's capabilities by being able to ask not only if ∃ a child that resolves to true, but also if all children resolve to true.
- The question if all children are resolved to true can be symbolized using the universal quantifier, ∀.
- Expanding on the logic gate metaphor for nondetermenism, each node in alternation can act like either an OR gate or an AND gate:



Alternating Turing Machine

- An alternating Turing machine is a nondetermenistic Turing machine with an extra feature.
- All states other than q_{accept} and q_{reject} are divided into 2 types:
  - Universal States: All children of that state must accept for the state to accept. Quantifier: ∀
  - Existential States: There must exist one child of the state that accepts in order for the state to accept. Quantifier: ∃
- The input to an alternating TM accepts if its start node accepts.

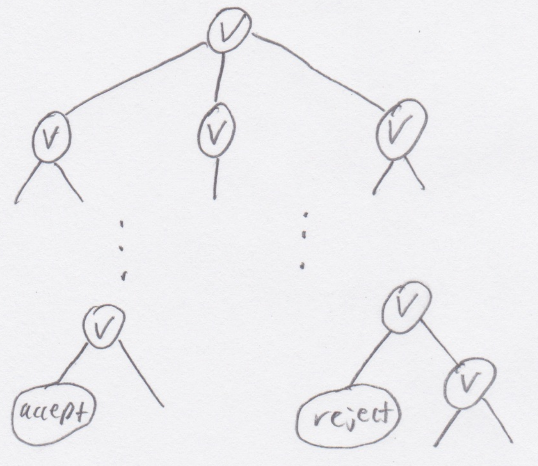
Representing alternating computation trees:

- In a computation tree, each child of a node is in one possible state after the parent state:

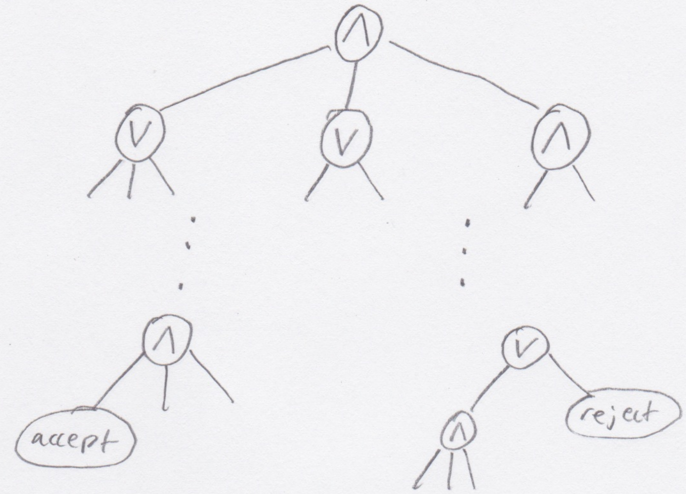
  $\wedge$: Universal State

  $\vee$: Existential State

Nondetermenistic Tree

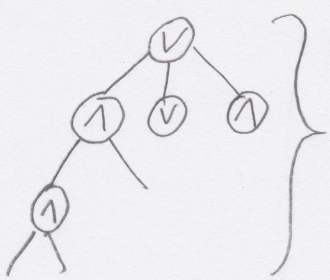


Alternating Tree



2 Types of alternating Turing machines

- for these definitions, i is a natural number
- An Σ_i-alternating TM starts with an existential step, and contains at most i runs of universal or existential steps on every input and computation branch.



Length is at most i

- A Π_i-alternating TM differs only in the fact that it starts with a universal step instead of an existential one.

Example: The Tautology Problem

- A tautology is a boolean formula that evaluates to 1, no matter what assignment of variables is used.

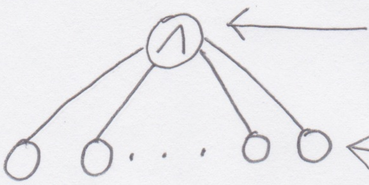
Algorithm:

TAUT = $\{\langle \phi \rangle \mid \phi$ is a tautology$\}$

On an input, ϕ:

1. Using a universal state, select all possible assignments for the variables in ϕ.
2. Evaluate ϕ for a particular assignment of variables
3. If ϕ evaluates to 1, accept. Otherwise reject.

Graphically



Accept only if all children evaluate to 1 and accept.

Each child is one assignment of variables in ϕ. These steps are detemenistic.

Another example - The MIN-FORMULA Problem

- This problem is not known to be in NP, but it is in ANP
- MIN-FORMULA states that for a boolean formula, $\emptyset$, there exists no other boolean formula that has the same variables and is equivelant and shorter.

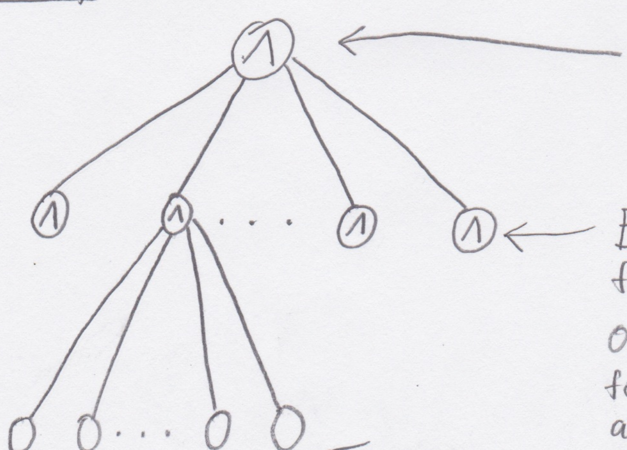
Algorithm:

On input, $\emptyset$:

1. Using a universal state, select all formulas, ψ, shorter than $\emptyset$
2. Using an existential state, select an assignment of variables for $\emptyset$.
3. Evaluate $\emptyset$ and ψ on the assignment.
4. Accept if the formulas evaluate to different values.

   Reject otherwise.

Graphically



Accept only if all children accept.

Each child is a shorter boolean formula, and will only accept if one of its children finds the formulas have different values and accepts

Each child is one assignment of the variables in $\emptyset$. These steps are determenistic and will accept if $\emptyset$ and ψ evaluate to different values and reject otherwise.