Alternation

Ταουσάκος Θανάσης
Αλγόριθμοι και Πολυπλοκότητα II
7 Φεβρουαρίου 2013

Alternation and Non-Determinism$_{1/2}$

Alternation: important generalization of non-determinism

Redefining Non-Determinism in terms of configurations: a configuration “lead to acceptance ” iff it is a final accepting configuration or at least one of its successors leads to Acceptance.

Each configuration is in some sense an implicit OR of its successor configurations.

In contrast the machine for the complement of the same language will have implicit AND’s as configurations.

Alternation and Non-Determinism$_{1/2}$

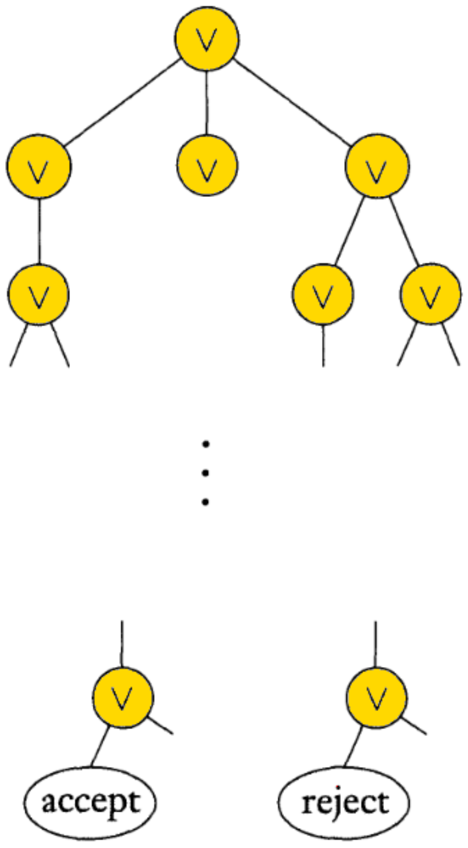
Suppose that we allow both modes in our nondeterministic machines

- AND configurations accept if all of their successors accept

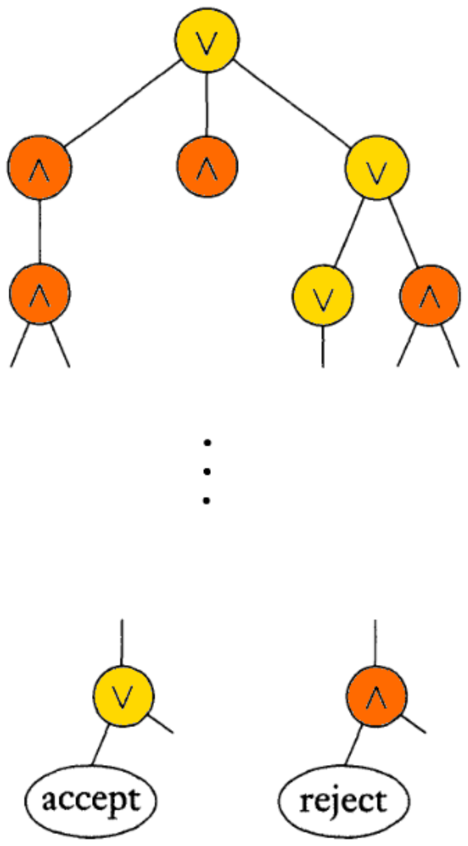
- OR configurations accept if at least one of their successors accept

- The machine accepts its input iff its initial configuration with this input does.

Computation trees



nondeterministic



alternating

Alternating Turing Machines – ATMs

ATM: a nondeterministic TM N = (K , Σ, Δ, s) in which the set of states K is partitioned into two sets, $K = K_{AND} \cup K_{OR}$. Let x be the input and consider the tree of computations of N on input x. Each node in this tree is a configuration of the precise machine, and includes the step number of the machine.

Alternating time and space

- **ATIME(f(n)):** the class of all languages decided by an ATM, all computations of which on input x halt after at most f(|x|) steps

$$\mathbf{AP} = \bigcup_c \mathbf{ATIME}(n^c)$$

- **ASPACE(f(n)):** the class of all languages decided by an ATM that uses no more than f(|x|) space on input x.

$$\mathbf{AL} = \mathbf{ASPACE}(\log n)$$

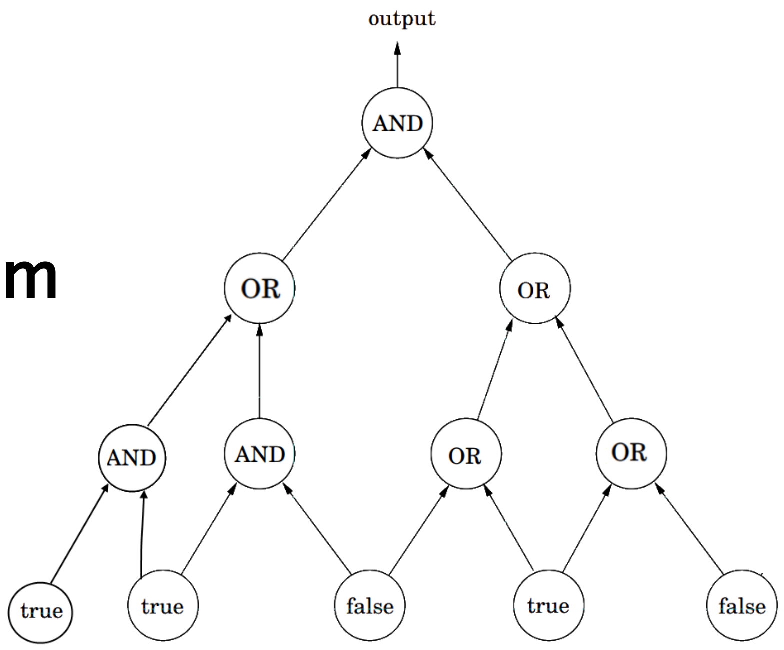
AL = P

Proof Sketch:

- MCVP is P-Complete (Ch.8)
- MCVP is AL-Complete (to be proven)
- Both classes are closed under reductions and the have the same complete problem.

MONOTONE CIRCUIT VALUE is AL-Complete 1/3

- The CIRCUIT VALUE problem is the following: when the laws of Boolean logic are applied to the gates in topological order, does the output evaluate to true?
- Monotone Circuit: with only AND and OR gates.
- if the one input changes from false to true the value of the function cannot change from true to false.



MONOTONE CIRCUIT VALUE is AL-Complete 2/3

- The input of our ATM is a circuit.

The machine examines the output gate of the circuit.

- If it is an AND gate, then the machine enters an AND state;
- if the output gate is an OR gate, then it enters an OR state.

- In either case, the machine determines the two gates that are

predecessors of the output (it does so by remembering the output gate while examining all edges), and it nondeterministically chooses one.

- The same process is repeated at the new gate, till the input gate where the machine accepts if it is a true gate, and rejects if it is a false gate.

Only logarithmic space is needed.

MONOTONE CIRCUIT VALUE is AL-Complete 3/3

- Any language L is reducible to the MCVP.
- Construct a monotone circuit C:
  - C's output $\longrightarrow x \in L$
  - Construction:
    - Gates (C,i) where C is a configuration for input x
      - There is an arc between $(C_1,i) \rightarrow (C_2,j)$ iff $C_2 \rightarrow C_1$, $j = i + 1$
      - Gate type depends on the state:
        - K_{OR} = OR
        - K_{AND} = AND
        - Yes = true
        - No = false
        - Output= initial configuration
    - and i the step number $0 < i < |x|^k$ the time-bound in order to for the circuit to be acyclic

AP = PSPACE

Proof Sketch:

- AP = ATIME(n^c)
- QSAT is PSPACE-Complete
- QSAT is AP-Complete
- Both classes are closed under reductions and the have the same complete problem.

QSAT is AP-Complete$_{1/3}$

- Also known as QBF (Quantified Boolean Formula)
- Given a Boolean expression ϕ in CNF, Φ is satisfied by the overall truth assignment?

$$\exists x_1 \forall x_2 \exists x_3 \ldots Q_n x_n \quad \phi?$$

QSAT is AP-Complete$_{2/3}$

QSAT is in AP

The computation will guess the truth values of the variables X_1, X_2, . .. one- by-one, where existentially quantified variables are guessed at states in K_{OR} , while universally quantified ones at states in K_{AND}.
A final state is accepting if the guessed truth assignment satisfies the expression, and rejecting otherwise.
It follows from the definition of acceptance for alternating machines that a quantified expression is accepted iff it is true; the time needed is polynomial.

QSAT is AP-Complete$_{3/3}$

- The computation of a polynomial-time ATM can be captured by a table, with extra nondeterministic choices.
- In the resulting expression the quantifiers for the nondeterministic choices are
  - universal if the current state is in K_{AND}
  - existential if the current state in K_{OR}.
- We can standardize our ATMs so that the successors of K_{OR} configurations are K_{AND}, and vice-versa;
- The variables standing for nondeterministic choices at even levels are existentially quantified, and at odd levels universally.
- All other variables (the gates of the circuit) are quantified existentially.
- The ATM accepts the input iff the resulting quantified expression is true.

Interesting combination of time and space

- Alternating space is precisely deterministic time, only one exponential higher

ASPACE = EXP

- Alternating time is roughly equivalent to deterministic space

$$\mathrm{ASPACE}(f(n)) = \mathrm{TIME}\ (k^{f(n)})$$

ATMs restricted to a fixed number of alternations. 1/2

For every i ∈ N, we define Σ_iTIME(T(n)) to be the set of languages accepted by a T(n)–time ATM M whose initial state is labeled "∃" and on which every input and on every (directed) path from the starting configuration in the configuration graph, M can alternate <u>at most i−1 times</u> from states with one label to states with the other label.

For every $i \in \mathbb{N}$ $\Sigma_i^p = \bigcup_c \Sigma_i \text{TIME}(n^c)$

ATMs restricted to a fixed number of alternations. 2/2

For every i ∈ N, we define Π_iTIME(T(n)) to be the set of languages accepted by a T(n)–time ATM M whose initial state is labeled "∀" and on which every input and on every (directed) path from the starting configuration in the configuration graph, M can alternate at most i−1 times from states with one label to states with the other label.

For every $i \in \mathbb{N}$ $\Pi_i^p = \bigcup_c \Pi_i \mathrm{TIME}(n^c)$

The class TISP

For every two functions $S,\ T : \mathbb{N} \to \mathbb{N}$,
the class $\mathrm{TISP}(T(n), S(n))$
is the set of languages decided by a TM
M that on every input $x \in \{0,1\}^*$ takes
at most $O(T(|x|))$ steps and uses at most
$O(S(|x|))$ cells of its read/write tape

SAT and TISP

One can show that SAT $\notin \mathrm{TISP}(n^c, n^d)$
for every constants c, d such that $c(c+d) < 2$.

Bibliography

- Christos Papadimitriou, Computational Complexity, Addison Wesley, 1994
- Sanjeev Arora and Boaz Barak, Computational Complexity: A Modern Approach, Cambridge University Press, 2009
- Michael Sipser , Introduction to the Theory of Computation, Thomson Course Technology, 2006