Alternation

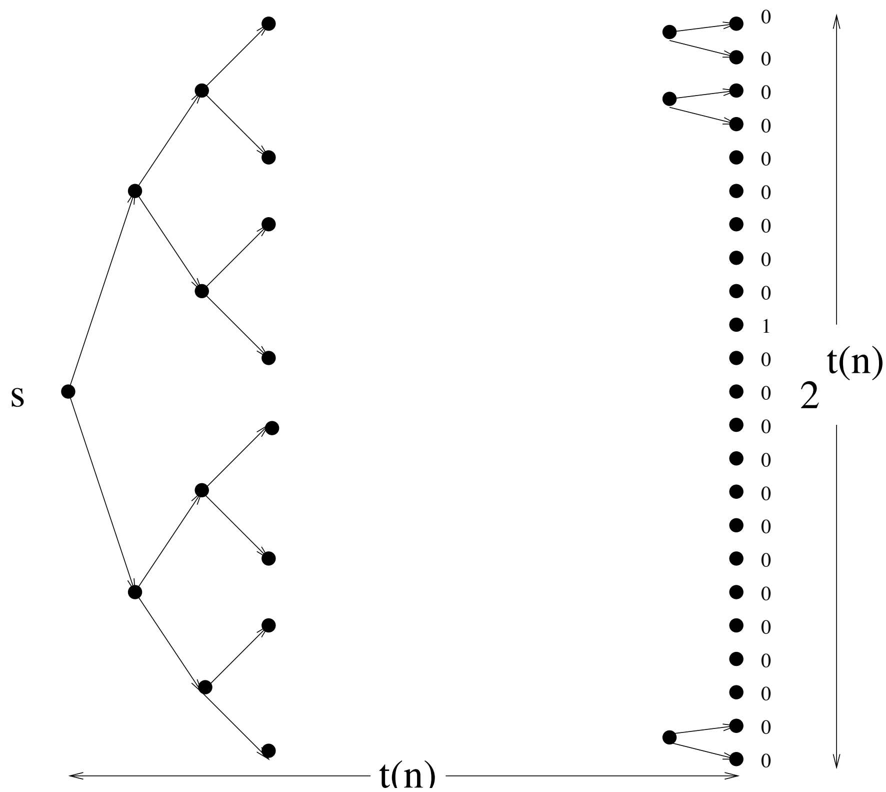
The concept of a nondeterministic acceptor of a boolean query has a long and rich history, going back to various kinds of nondeterministic automata.

It is important to remember that these are **fictitious machines**: we suspect that they cannot be built.

Open question: $\quad \text{NP} \quad ?= \quad \text{co-NP} \ = \ \left\{ \overline{A} \ \middle|\ A \in \text{NP} \right\}$

If one could really build an NP machine, then one could, with a single gate to invert its answer, also build a co-NP machine.

From a practical point of view, the complexity of a problem A and its complement, $\overline{A}$ are identical.



weak

communication

pattern

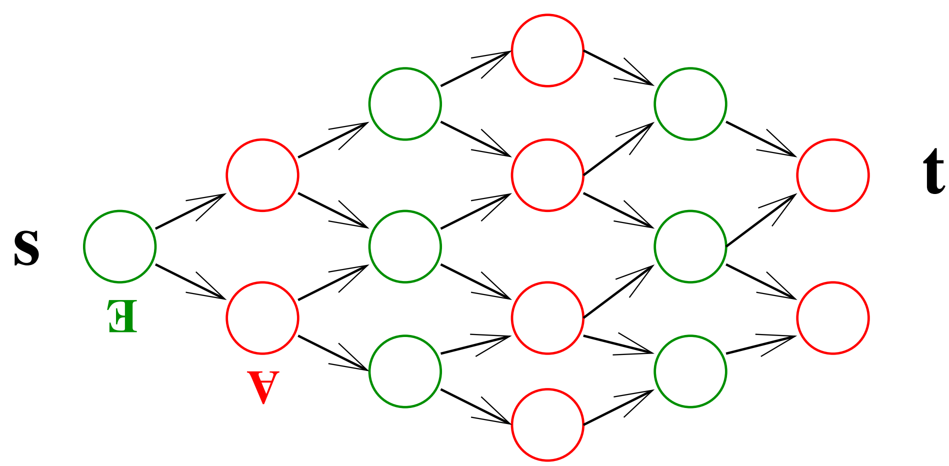
Value(ID) := Value(LeftChild(ID)) $\vee$ Value(Right(Child(ID))

The states of an **alternating Turing machine** are split into: **Existential states** ($\exists$) and **Universal states** ($\forall$).

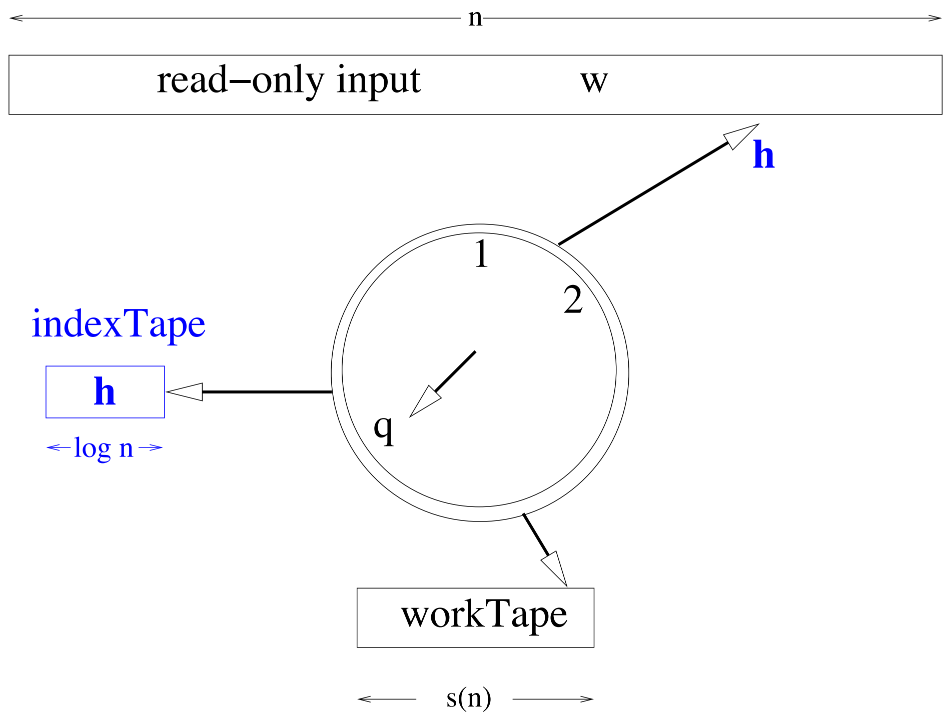
Definition 13.1 An alternating TM in ID_0 **accepts** iff

1. ID_0 is in a final accepting state, or
2. ID_0 is in an $\exists$ state and some next ID' accepts, or
3. ID_0 is in a $\forall$ state, has at least one next ID, and all next ID's accept.

□



From now on assume that our Turing machines have a **random access** read-only input. There is an **index tape** which can be written on and read like other tapes. Whenever the value h, written in binary, appears on the index tape, the read head will automatically scan bit h of the input.



Definition 13.2 Let ASPACE$[s(n)]$ and ATIME$[t(n)]$ be the set of problems accepted by alternating TM's using $O(s(n))$ tape cells, $O(t(n))$ time, respectively, in any computation path on any input of length n. $\square$

Theorem 13.3 *[Alternation Thm.] For* $s(n) \geq \log n$, *and for* $t(n) \geq n$,

$$\bigcup_{k=1}^{\infty} \text{ATIME}[(t(n))^k] \quad = \quad \bigcup_{k=1}^{\infty} \text{DSPACE}[(t(n))^k]$$

$$\text{ASPACE}[s(n)] \quad = \quad \bigcup_{k=1}^{\infty} \text{DTIME}[k^{s(n)}]$$

Corollary 13.4 $\quad \text{ASPACE}[\log n] \; = \; \text{P} \quad$ *and* $\quad \text{ATIME}[n^{O(1)}] \; = \; \text{PSPACE}.$

Definition 13.5 The **monotone, circuit value problem** (MCVP) is the subset of CVP in which no negation gates occur. □

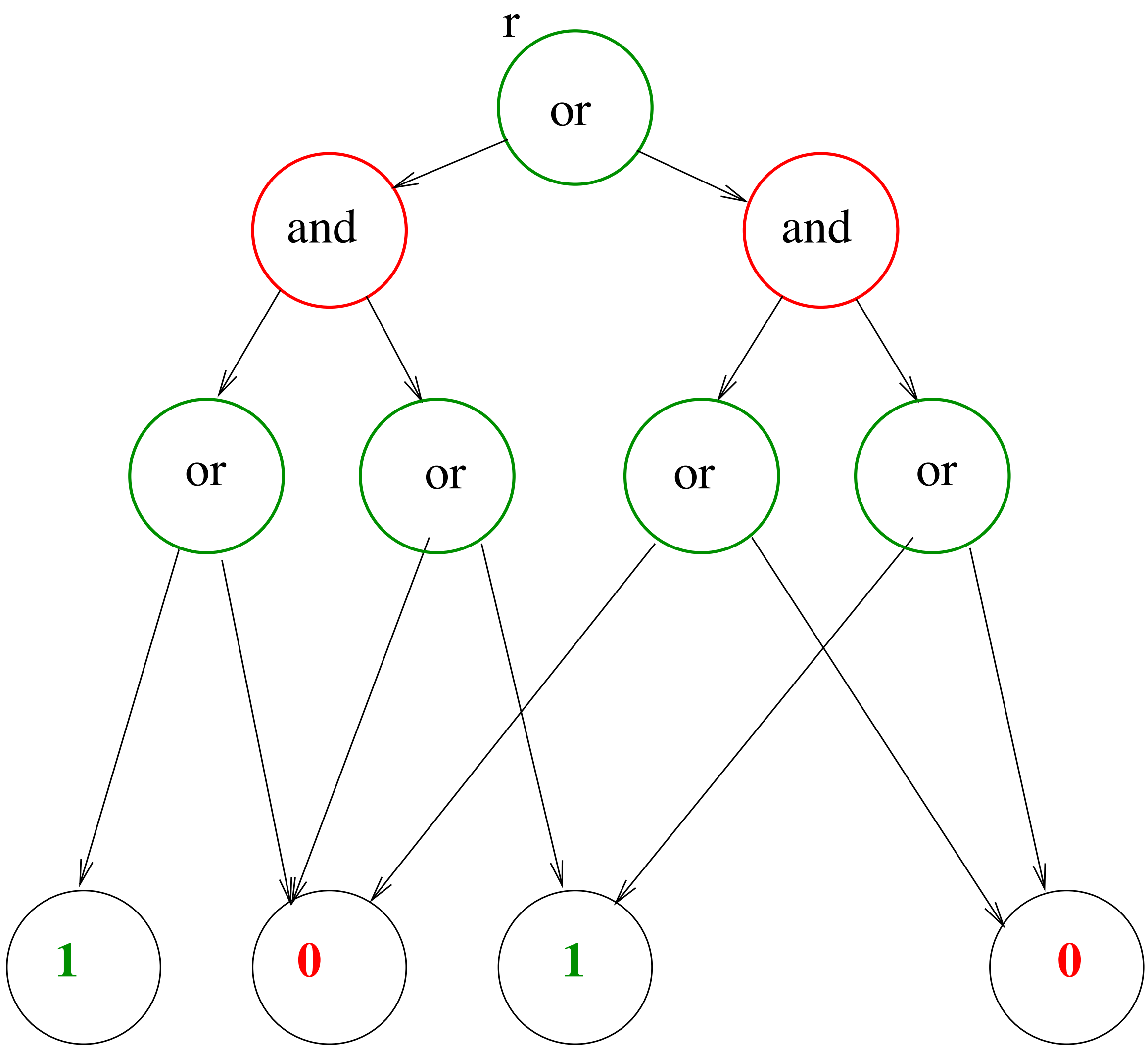
Proposition 13.6 MCVP $\in$ ASPACE$[\log n]$.

Proof: Let G be a monotone boolean circuit. For $a \in V^G$, define "EVAL(a)",

1. **if** (InputOn(a)) **then accept**
2. **if** (InputOff(a)) **then reject**
3. **if** $(G_\wedge(a))$ **then** universally choose child b of a
4. **if** $(G_\vee(a))$ **then** existentially choose child b of a
5. Return(EVAL(b))

M simply calls EVAL(r). EVAL(a) returns "**accept** " iff gate a evaluates to one.

Space used for naming vertices a, b: $O(\log n)$. □



The above circuit is a member of MCVP because it just has $\wedge$ and $\vee$ gates and it evaluates to 1.

Def: The **quantified satisfiability problem** (QSAT) is the set of true formulas of the following form:

$$\Psi \quad = \quad Q_1 x_1 \, Q_2 x_2 \, \cdots \, Q_r x_r \, (\varphi)$$

For any boolean formula φ on variables $\overline{x}$,

$$\begin{array}{ccc} \varphi \in \mathrm{SAT} & \Leftrightarrow & \exists \overline{x}\, (\varphi) \in \mathrm{QSAT} \\ \varphi \notin \mathrm{SAT} & \Leftrightarrow & \forall \overline{x}\, (\neg\varphi) \in \mathrm{QSAT} \end{array}$$

Thus QSAT logically contains SAT and $\overline{\mathrm{SAT}}$.

Proposition 13.7 QSAT $\in$ ATIME$[n]$.

Proof: Construct ATM, A, on input, $\Phi \equiv$

$$\begin{array}{ccccccccc} \exists x_1 & \forall x_2 & \cdots & \exists x_{2k-1} & \forall x_{2k} & \bigwedge_{i=1}^{r} & \bigvee_{j=1}^{s} & \ell_{ij} \\ b_1 & b_2 & \cdots & b_{2k-1} & b_{2k} & i & j & \ell_{ij}(b_1, \ldots, b_{2k}) \end{array}$$

Quantifiers:

- in $\exists$ state, A writes a bit b_1 for x_1,
- in $\forall$ state, A writes a bit b_2 for x_2, and so on.

Boolean operators:

- in $\forall$ state, A chooses i,
- in $\exists$ state, A chooses j

Final state: accept iff $\ell_{ij}(b_1, \ldots, b_{2k})$ is true.

$$A \text{ accepts } \Phi \quad \Leftrightarrow \quad \Phi \text{ is true.} \qquad \square$$

Theorem 13.8 *For any* $s(n) \geq \log n$, $\quad \text{NSPACE}[s(n)] \ \subseteq \ \text{ATIME}[s(n)^2] \ \subseteq \ \text{DSPACE}[s(n)^2]$.

Proof: NSPACE$[s(n)] \subseteq$ ATIME$[s(n)^2]$:

Let N be an NSPACE$[s(n)]$ Turing machine.

Let w be an input to N, $n = |w|$.

$$w \in \mathcal{L}(N) \quad \Leftrightarrow \quad \text{CompGraph}(N, w) \in \text{REACH}$$

$w \in \mathcal{L}(N) \quad \Leftrightarrow \quad$ CompGraph$(N, w) \in$ REACH

$$\begin{aligned} P(d,x,y) &\equiv \text{" In CompGraph}(N,w)\text{, dist}(x,y) \le 2^d \text{ "} \\ P(d,x,y) &\equiv \exists z \, (P(d-1,x,z) \, \wedge \, P(d-1,z,y)) \end{aligned}$$

1. Existentially: choose middle ID z.
2. Universally: $(x, y) := (x, z) \quad \& \quad (z, y)$
3. Return($P(d-1, x, y)$)

$$\begin{aligned} T(d) &= O(s(n)) + T(d-1) \; = \; O(d \cdot s(n)) \\ d &= O(s(n)) \\ T(d) &= O((s(n))^2) \end{aligned}$$

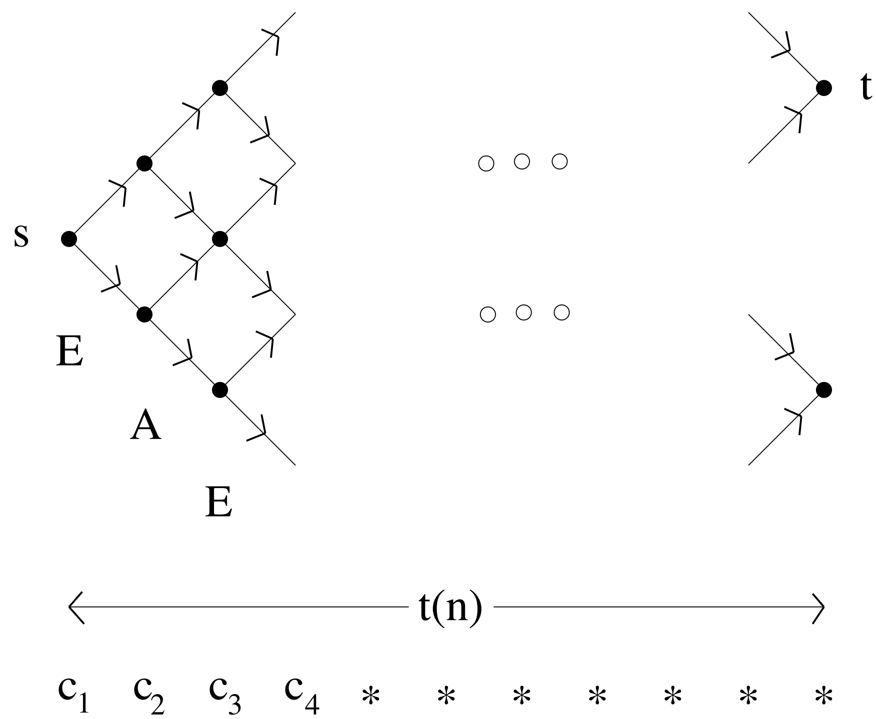
ATIME$[t(n)] \subseteq$ DSPACE$[t(n)]$

Let A be an ATIME$[t(n)]$ machine, input w, $n = |w|$.

CompGraph(A, w) has depth $c(t(n))$ and size $2^{c(t(n))}$, for some constant c.

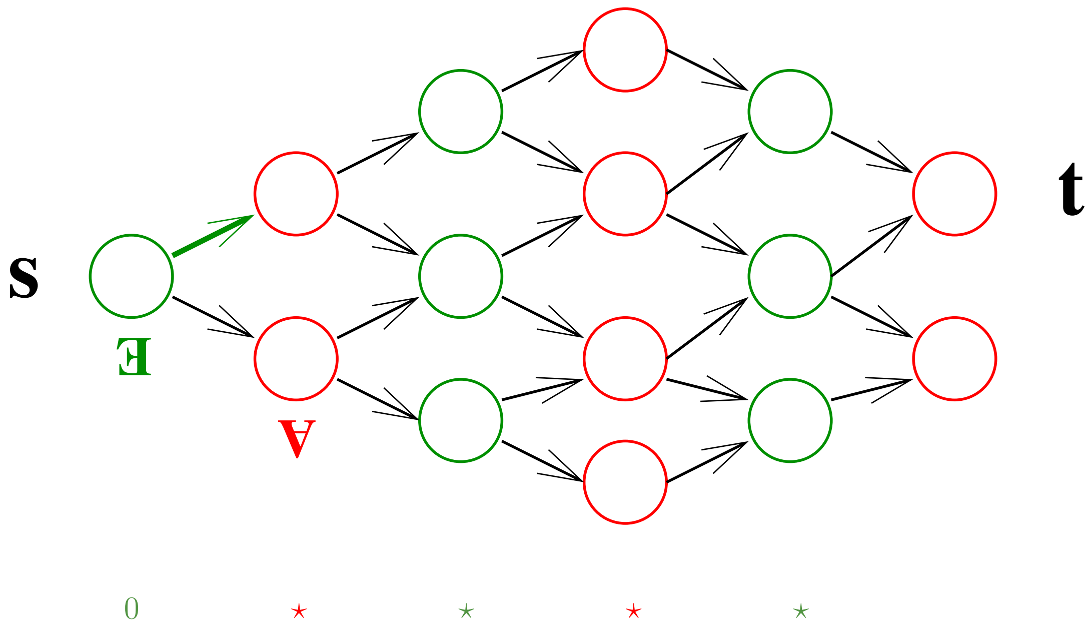
Search this and/or graph systematically using $c(t(n))$ extra bits of space.

$$\text{ATIME}[t(n)] \quad \subseteq \quad \text{DSPACE}[t(n)]$$



Evaluate computation graph of ATIME$[t(n)]$ machine using $t(n)$ space to cycle through all possible computations of A on input w.

Example: ATIME$[t(n)]$ $\subseteq$ DSPACE$[t(n)]$



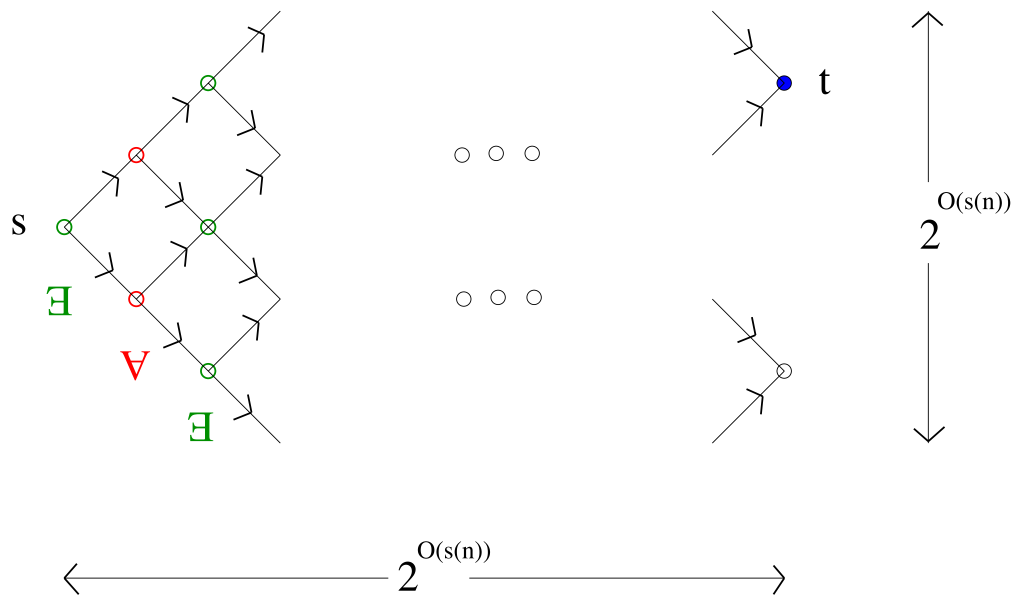
Theorem 13.9 ASPACE$[s(n)] \quad = \quad$ DTIME$[2^{O(s(n))}]$

Proof: ASPACE$[s(n)] \subseteq$ DTIME$[2^{O(s(n))}]$:

Let A be an ASPACE$[s(n)]$ machine, w an input, $n = |w|$.

CompGraph$(A(w))$ has size $\leq 2^{O(s(n))}$

Marking algorithm evaluates this in DTIME$2^{O(s(n))}$.



DTIME$[2^{O(s(n))}] \subseteq$ ASPACE$[s(n)]$:

Let M be DTIME$[2^{k(s(n))}]$ TM, w an input, $n = |w|$.

alternating procedure $C(t, p, a)$ accepts iff contents of cell p at time t in M's computation on input w is symbol a.

$C(t+1, p, b)$ holds iff the three symbols a_{-1}, a_0, a_1 in tape positions $p-1, p, p+1$ lead to a "b" in position p in one step of M's computation.

$$C(t+1,p,b) \;\equiv\; \bigvee_{(a_{-1},a_0,a_1)\overset{M}{\to}b} \quad \bigwedge_{i\in\{-1,0,1\}} C(t,p+i,a_i)$$

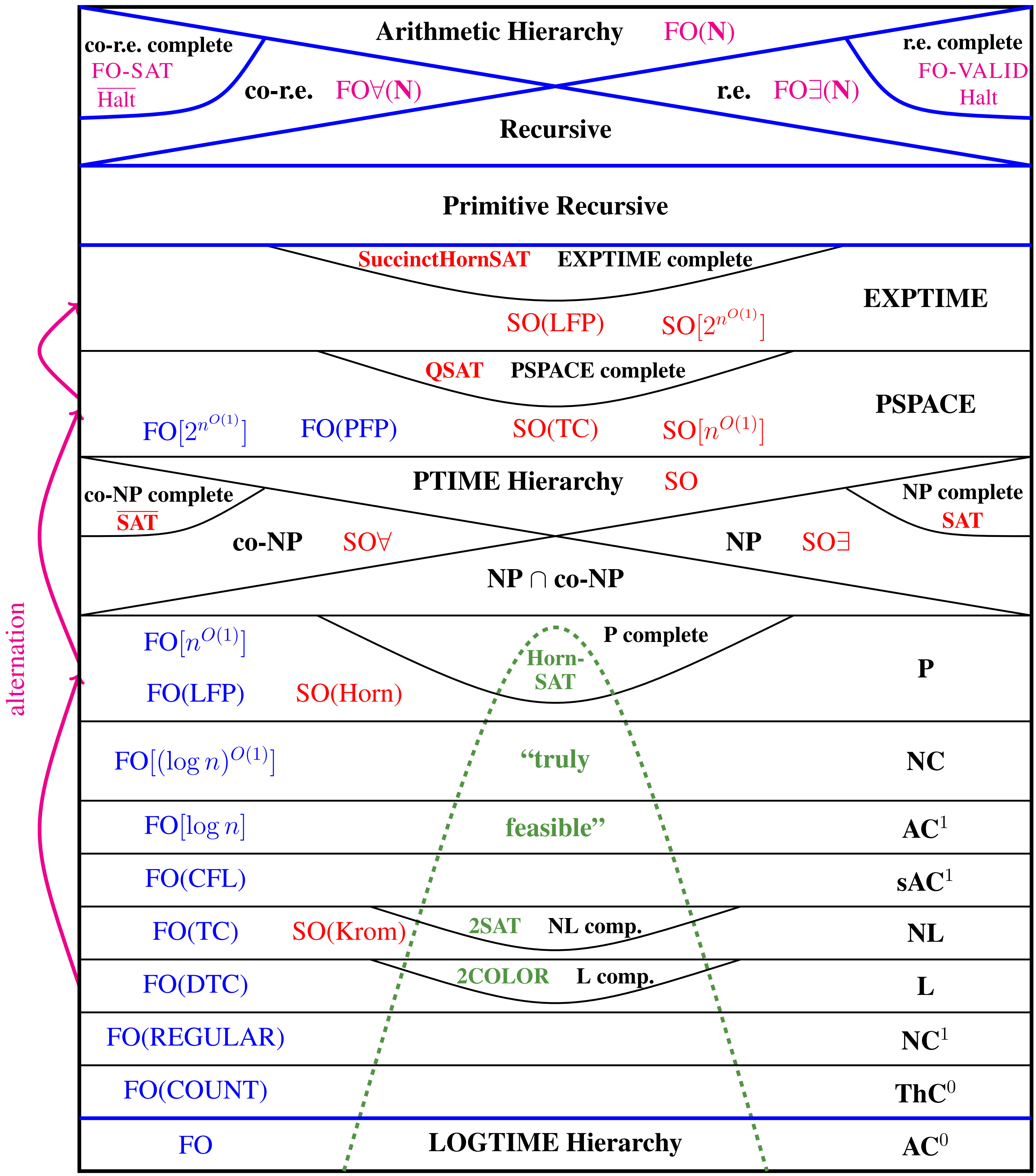
Space needed is $O(\log 2^{k(s(n))})$ = $O(s(n))$.

Note that M accepts w iff $C(2^{k(s(n))}, 1, \langle q_f, 1\rangle)$

	Space						
	0	1	$\bar{s}$	$n-1$	n		$2^{ks(n)}$
0	$\langle q_0, w_0\rangle$	w_1	$\cdots$	w_{n-1}	$\sqcup$	$\cdots$	$\sqcup$
1	w_0	$\langle q_1, w_1\rangle$	$\cdots$	w_{n-1}	$\sqcup$	$\cdots$	$\sqcup$
Time	$\vdots$	$\vdots$	$\vdots$			$\vdots$	
$\bar{t}$		$\boxed{a_{-1}}$	$\boxed{a_0}$	$\boxed{a_1}$			
$\bar{t}+1$			$\boxed{b}$				
	$\vdots$	$\vdots$	$\vdots$			$\vdots$	
$2^{ks(n)}$	$\langle q_f, 1\rangle$	$\sqcup$	$\cdots$	$\sqcup$	$\sqcup$	$\cdots$	$\sqcup$

$$C(t+1,p,b) \;\equiv\; \bigvee_{(a_{-1},a_0,a_1)\overset{M}{\to}b} \quad \bigwedge_{i\in\{-1,0,1\}} C(t,p+i,a_i)$$

This completes the proof of the Alternation Thm. $\square$


Arithmetic Hierarchy
FO(N)
co-r.e. complete
FO-SAT
$\overline{\text{Halt}}$
co-r.e.
FO∀(N)
r.e.
FO∃(N)
r.e. complete
FO-VALID
Halt
Recursive
Primitive Recursive
SuccinctHornSAT
EXPTIME complete
EXPTIME
SO(LFP)
$SO[2^{n^{O(1)}}]$
QSAT
PSPACE complete
PSPACE
$FO[2^{n^{O(1)}}]$
FO(PFP)
SO(TC)
$SO[n^{O(1)}]$
PTIME Hierarchy
SO
co-NP complete
$\overline{\text{SAT}}$
co-NP
SO∀
NP
SO∃
NP complete
SAT
NP ∩ co-NP
$FO[n^{O(1)}]$
P complete
Horn-SAT
P
FO(LFP)
SO(Horn)
$FO[(\log n)^{O(1)}]$
"truly
NC
$FO[\log n]$
feasible"
AC^1
FO(CFL)
sAC^1
FO(TC)
SO(Krom)
2SAT
NL comp.
NL
FO(DTC)
2COLOR
L comp.
L
FO(REGULAR)
NC^1
FO(COUNT)
ThC^0
FO
LOGTIME Hierarchy
AC^0
alternation