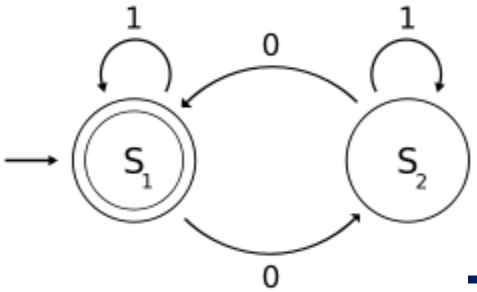
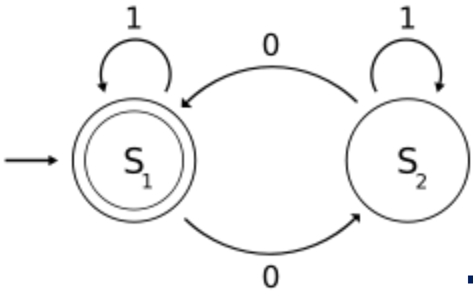
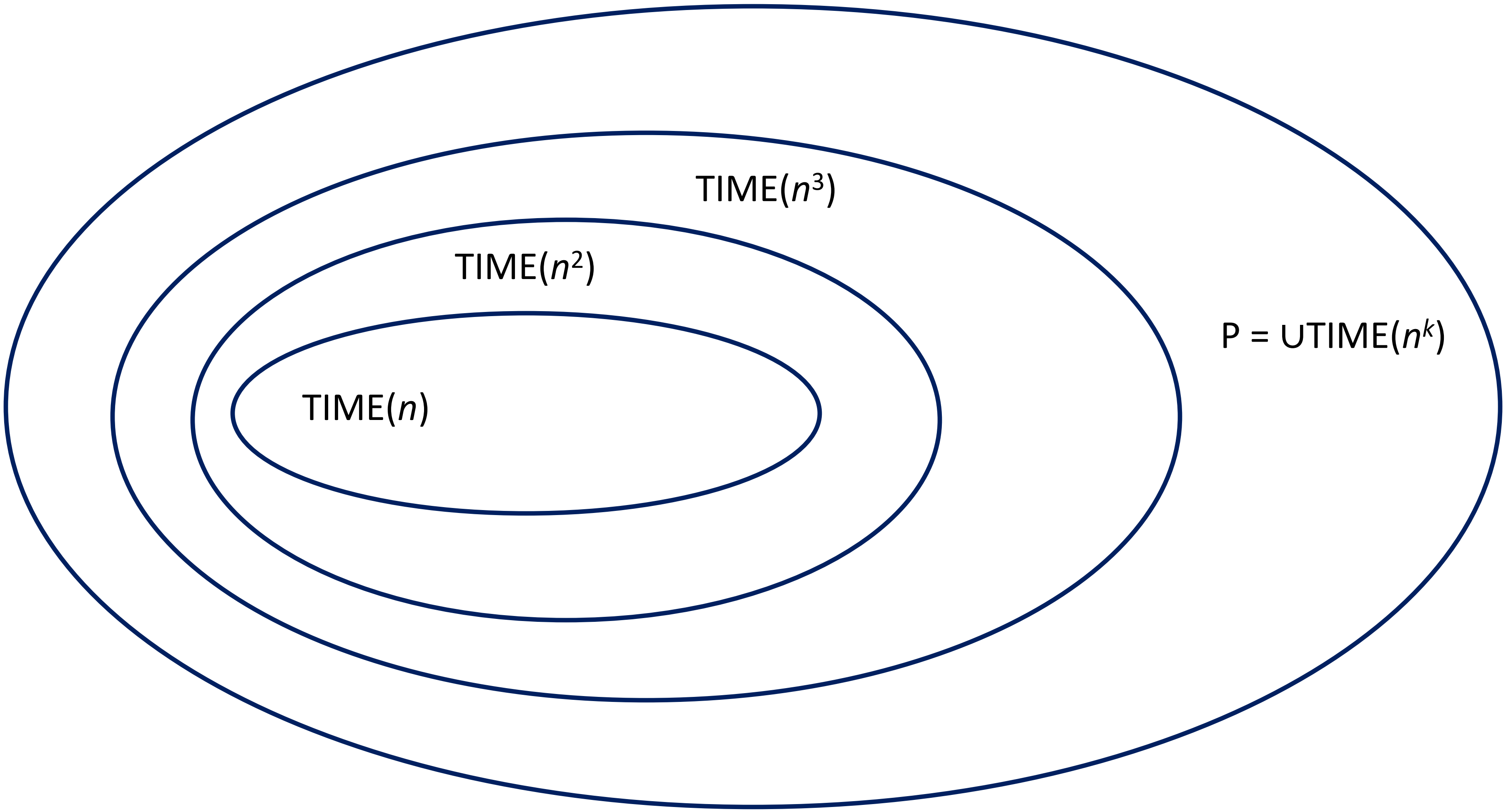


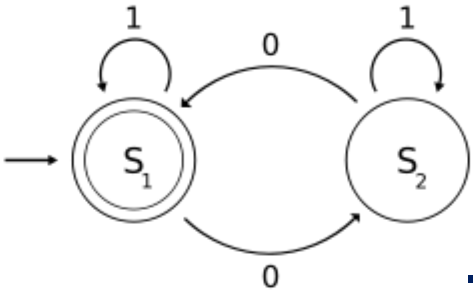
The Classes P and NP

Sipser: Section 7.2 pages 284 – 292; Section 7.3 pages 292 – 298



Polynomial Time

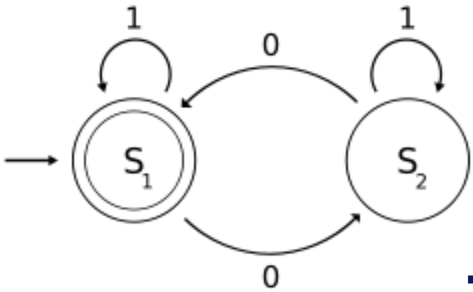




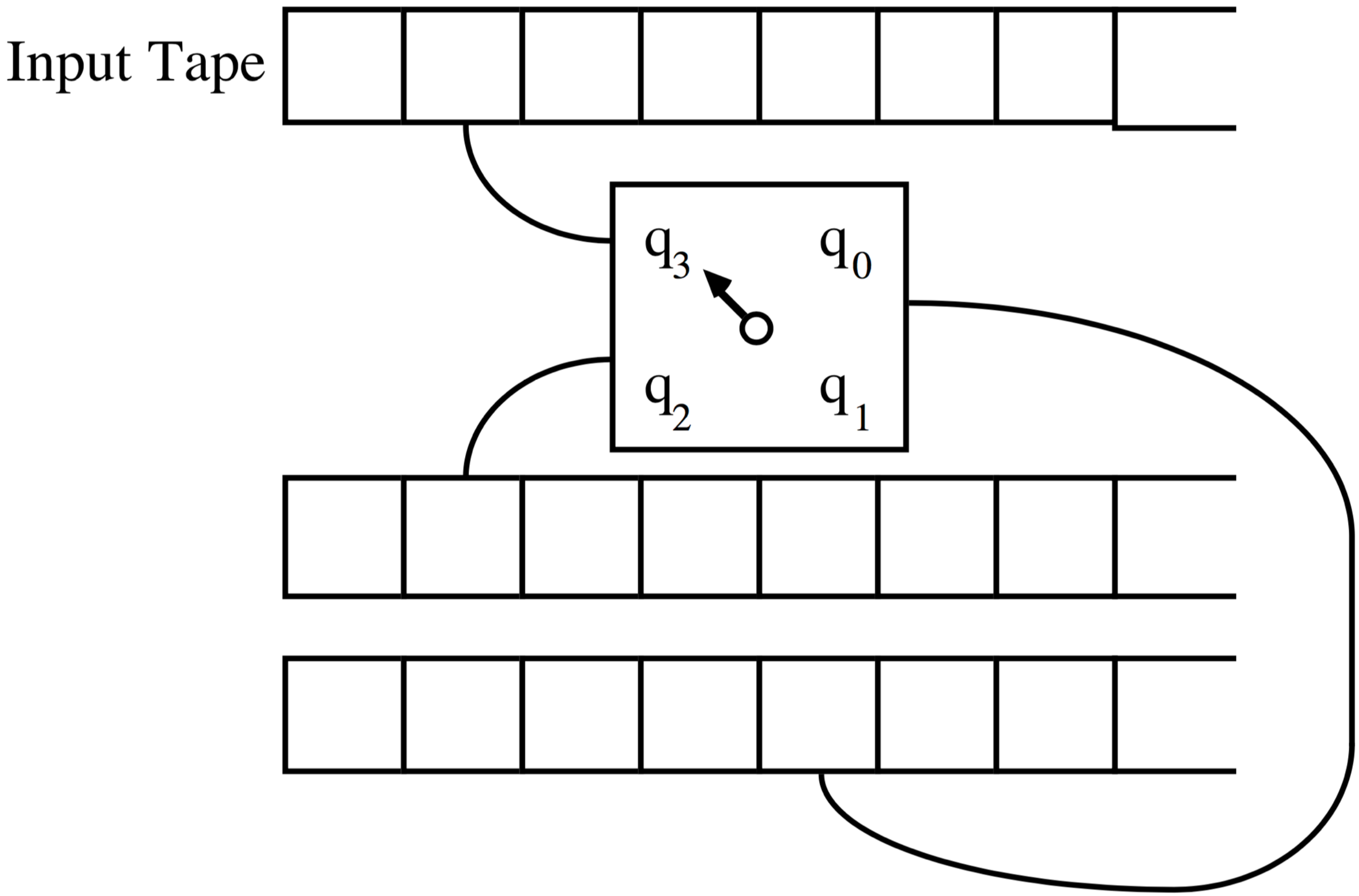
Tractable Problems

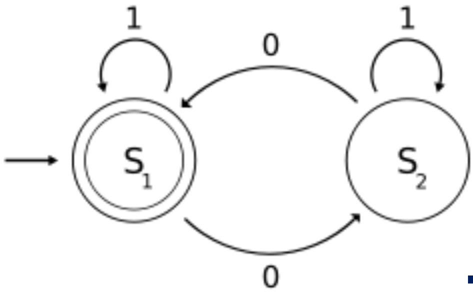
Size n

Time complexity function	10	20	30	40	50	60
n	.00001 second	.00002 second	.00003 second	.00004 second	.00005 second	.00006 second
n^2	.0001 second	.0004 second	.0009 second	.0016 second	.0025 second	.0036 second
n^3	.001 second	.008 second	.027 second	.064 second	.125 second	.216 second
n^5	.1 second	3.2 seconds	24.3 seconds	1.7 minutes	5.2 minutes	13.0 minutes
2^n	.001 second	1.0 second	17.9 minutes	12.7 days	35.7 years	366 centuries
3^n	0.59 second	58 minutes	6.5 years	3855 centuries	2×10^8 centuries	1.3×10^{13} centuries



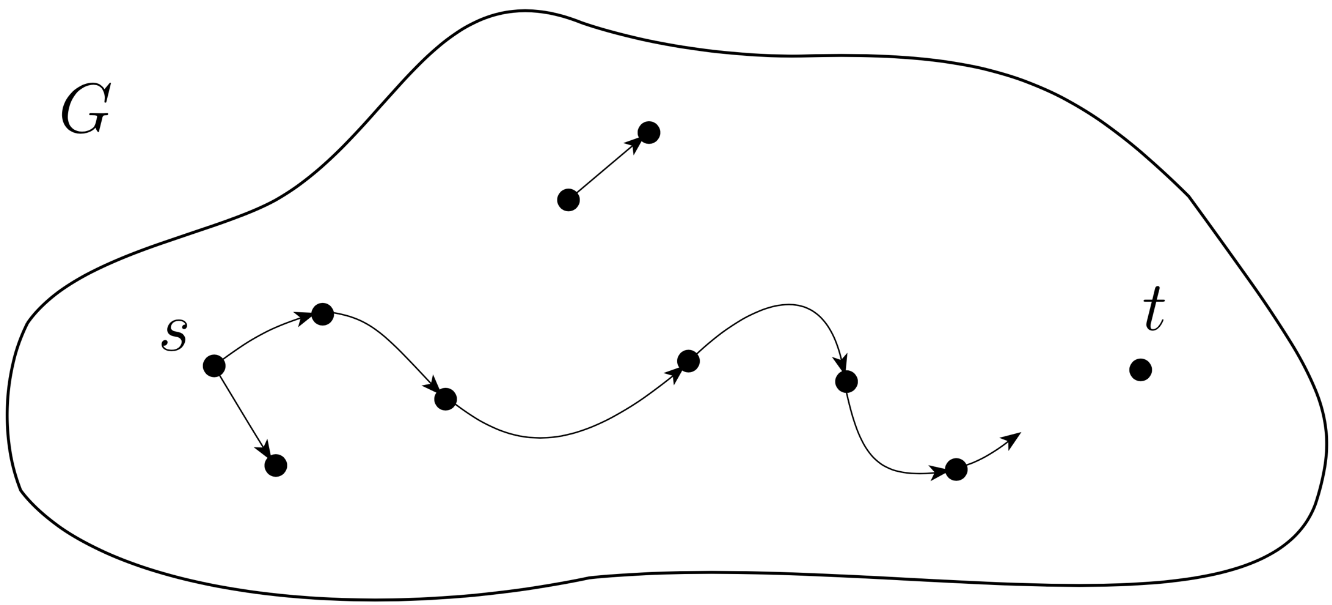
Polynomially Equivalent Models

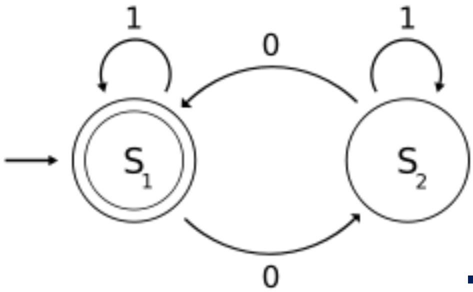




Graph Theory

Definition. *PATH* = {<*G*, *s*, *t*> | ∃ a directed path from *s* to *t* in *G*}

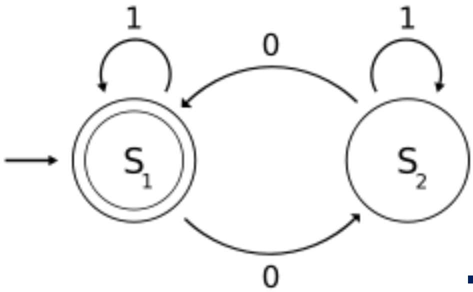




PATH ∈ P

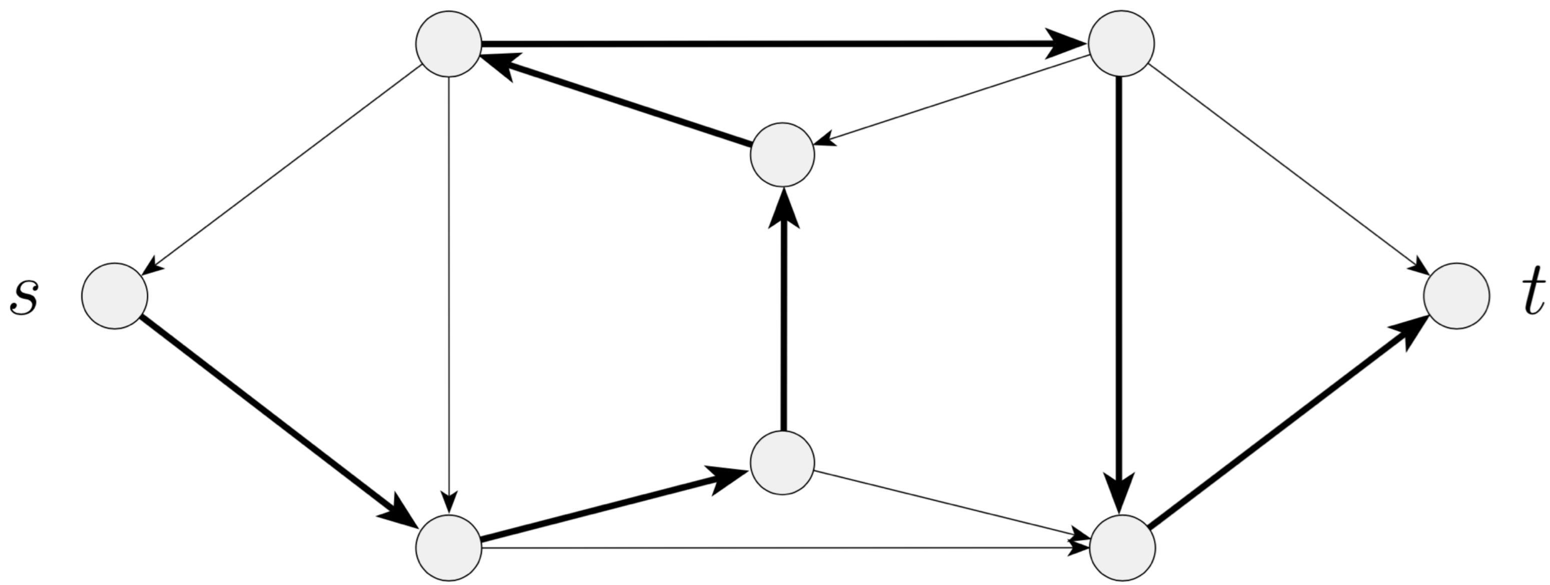
M = "On input ⟨*G*, *s*, *t*⟩:

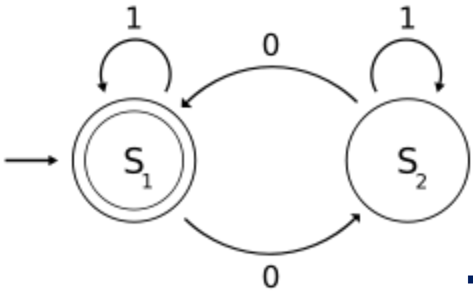
1. Place a mark on node *s*.
2. Repeat until no additional nodes are marked.
3. Scan all edges of *G*. If (*a*, *b*) found from marked node to unmarked node, mark *b*.
4. If *t* is marked, *accept*. Otherwise, *reject*."



Hamiltonian Paths

Definition. *HAMPATH* = {<*G*, *s*, *t*> | ∃ a Hamiltonian path from *s* to *t*}

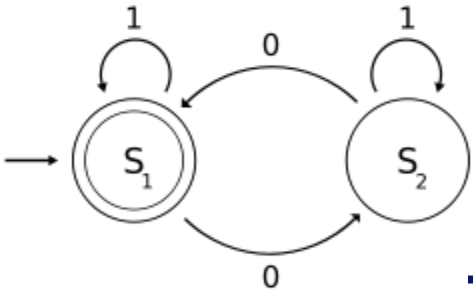




Checking for Hamiltonian Paths

E = "On input <*G*, *s*, *t*>:

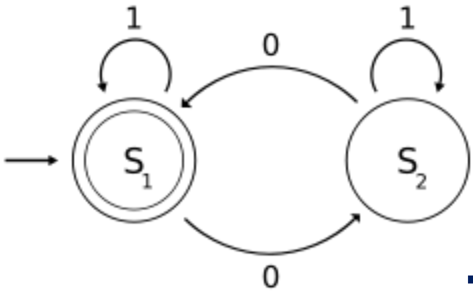
1. Generate all orderings, p_1, p_2, …, p_n, of the nodes in *G*.
2. Check whether $s = p_1$ and $t = p_n$.
3. For each $i = 1$ to n-1, check whether (p_i, p_{i+1}) is an edge in *G*. If any are not, *reject*. Otherwise, *accept*.



Guessing a Solution

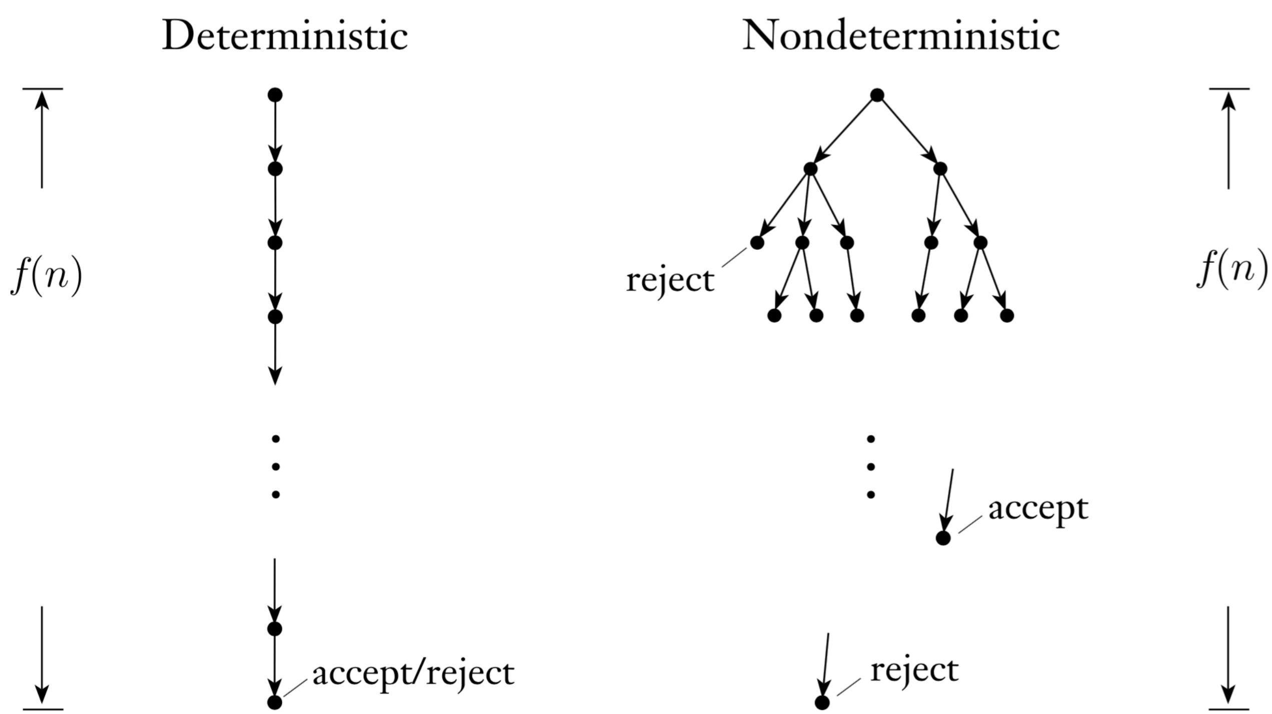
N = "On input <*G*, *s*, *t*>:

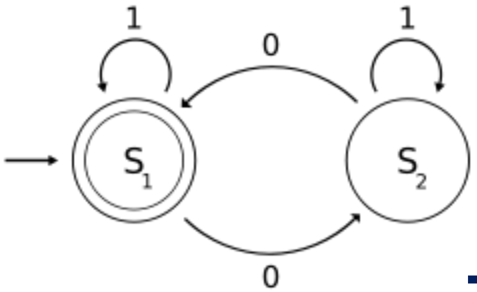
1. Guess an ordering, $p_1, p_2, \ldots, p_n$, of the nodes in *G*.
2. Check whether $s = p_1$ and $t = p_n$.
3. For each $i = 1$ to n-1, check whether (p_i, p_{i+1}) is an edge in *G*. If any are not, *reject*. Otherwise, *accept*.



Nondeterministic Time Complexity

Definition. Let *N* be a NTM. The ***running time*** of *N* is a function $f: \mathbb{N} \longrightarrow \mathbb{N}$, where $f(n)$ is the maximum number of steps that *N* uses on any branch of its computation on any input of length *n*.

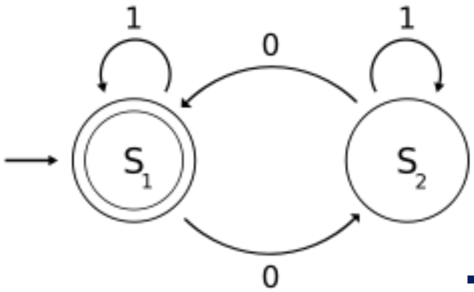




The Class NP

Definition. NTIME($t(n)$) = { L | L is decided in $O(t(n))$ time by an NTM.}

Corollary. NP = $\cup_k$ NTIME(n^k)



Certificates

NP is the class of languages that have polynomial time verifiers.

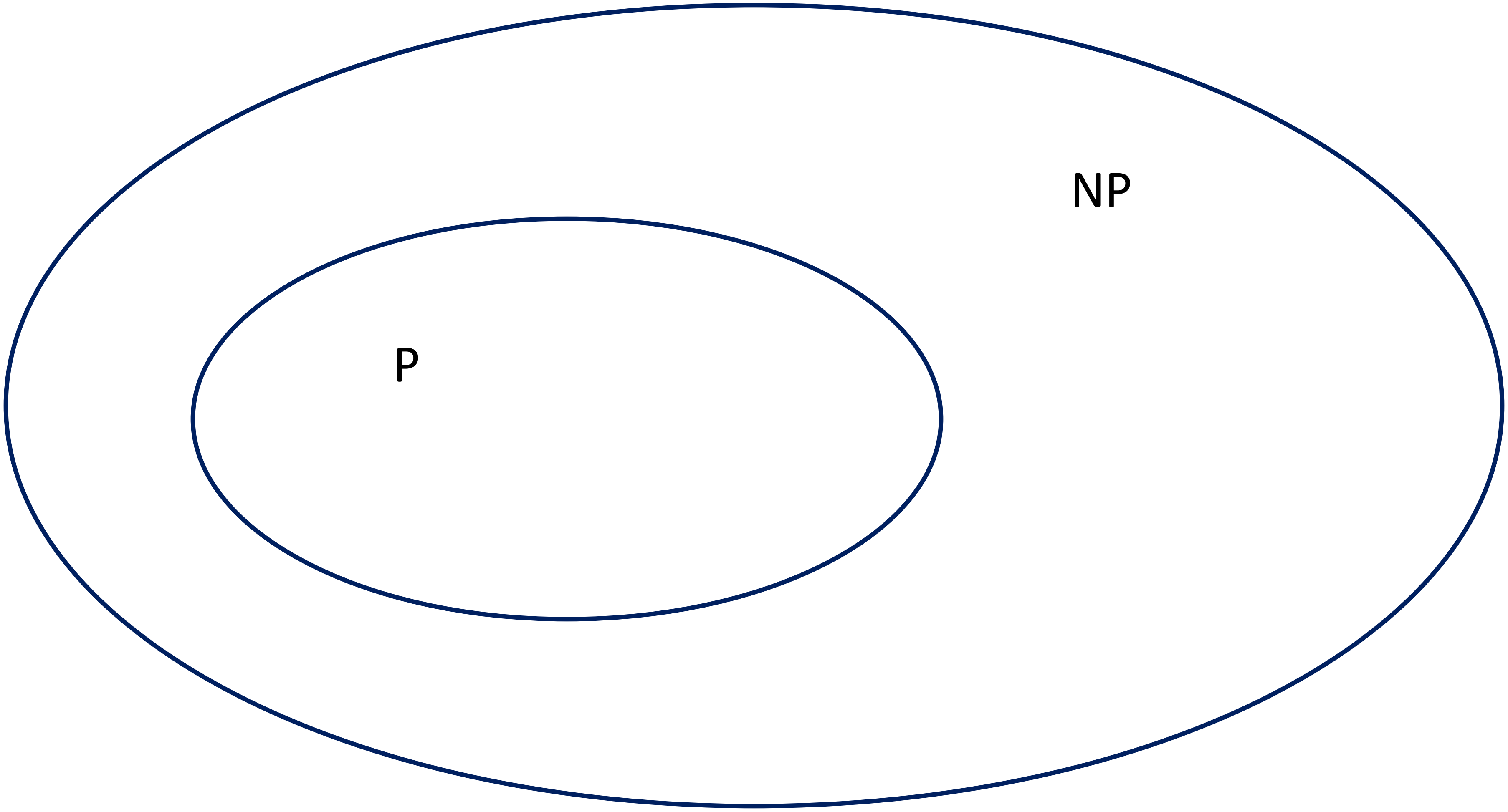
What is a certificate for *PATH*?

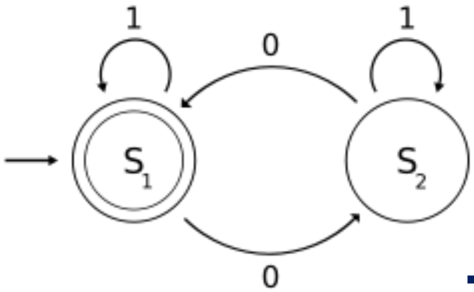
What is a certificate for *HAMPATH*?

What is a certificate for *COMPOSITE*?

What is a certificate for $\overline{HAMPATH}$?

The Classes P and NP

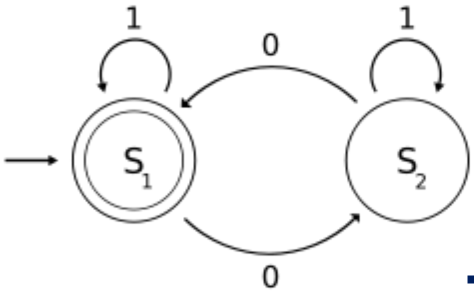




P = NP?

P ⊆ NP ⊆ PSPACE = NPSPACE ⊆ EXPTIME

Proper containment



Exercises

Let *CONNECTED* = { <*G*> | *G* is a connected undirected graph}.
Is *CONNECTED* in NP? Is *CONNECTED* in P?

A ***triangle*** in an undirected graph is a 3-clique.
Let *TRIANGLE* = { <*G*> | *G* contains a triangle}.
Is *TRIANGLE* in NP? Is *TRIANGLE* in P?

Call the graphs *G* and *H* ***isomorphic*** if the nodes of *G* may be reordered so that it is identical to *H*.
Let *ISO* = { <*G*, *H*> | *G* and *H* are isomorphic graphs}.
Is *ISO* in NP? Is *ISO* in P?