

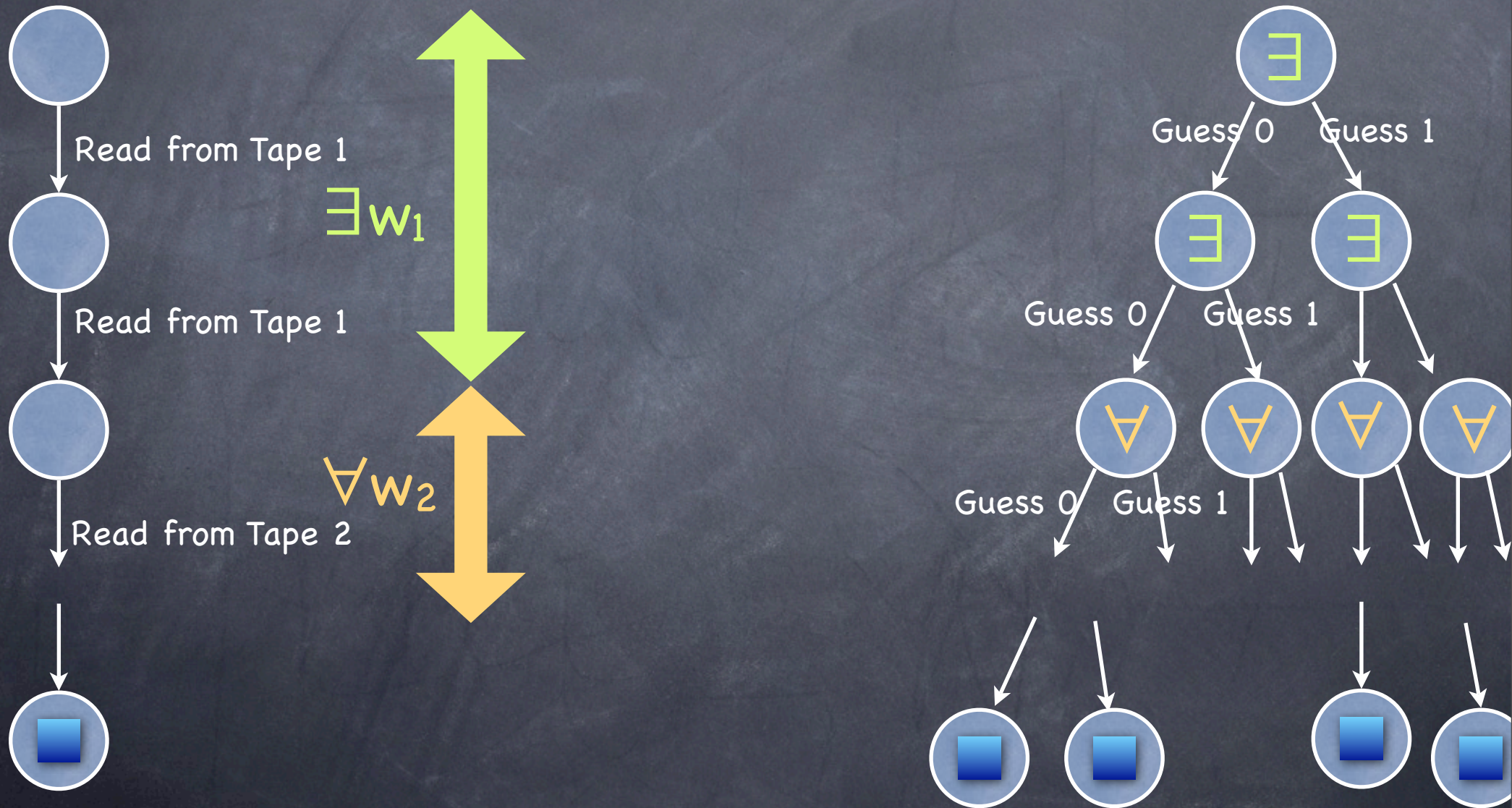
Computational Complexity

Lecture 9
More of the Polynomial Hierarchy
Alternation

PH is in terms of verification

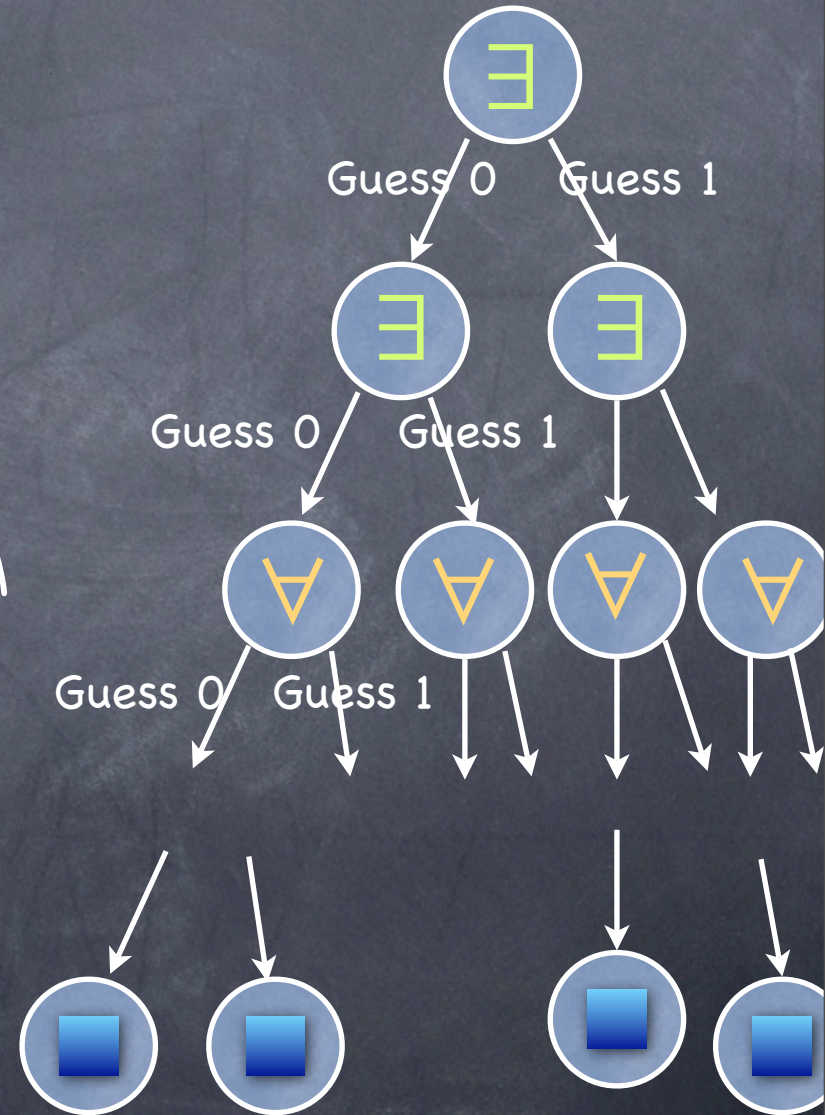
- Recall Σ_k^P
 - Languages $L = \{x \mid \exists w_1 \forall w_2 \dots Q w_k F(x; w_1, w_2, \dots, w_k)\}$, where F in P
 - Consider deterministic polynomial time machine M for F , with k read-once tapes for the certificates
 - Tapes read one after the other
 - x in L if $\exists w_1 \forall w_2 \dots Q w_k$ such that $M(x; w_1, w_2, \dots, w_k)$ accepts
- Plan: Formulate in terms of a non-deterministic TM (with no certificates)

Verification → Non-determinism



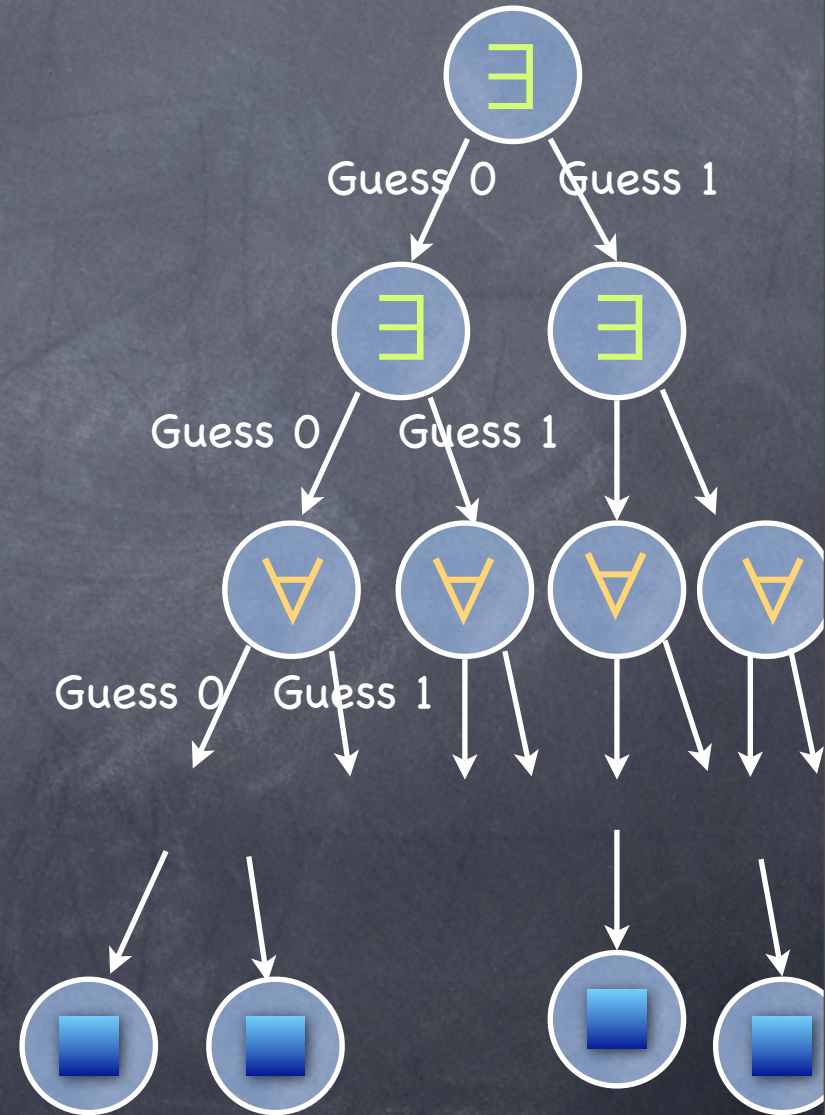
ATM

- Alternating Turing Machine
 - At each step, execution can fork into two
 - Exactly like an NTM or co-NTM
 - Accepting rule is more complex
 - Like in the game tree for QBF

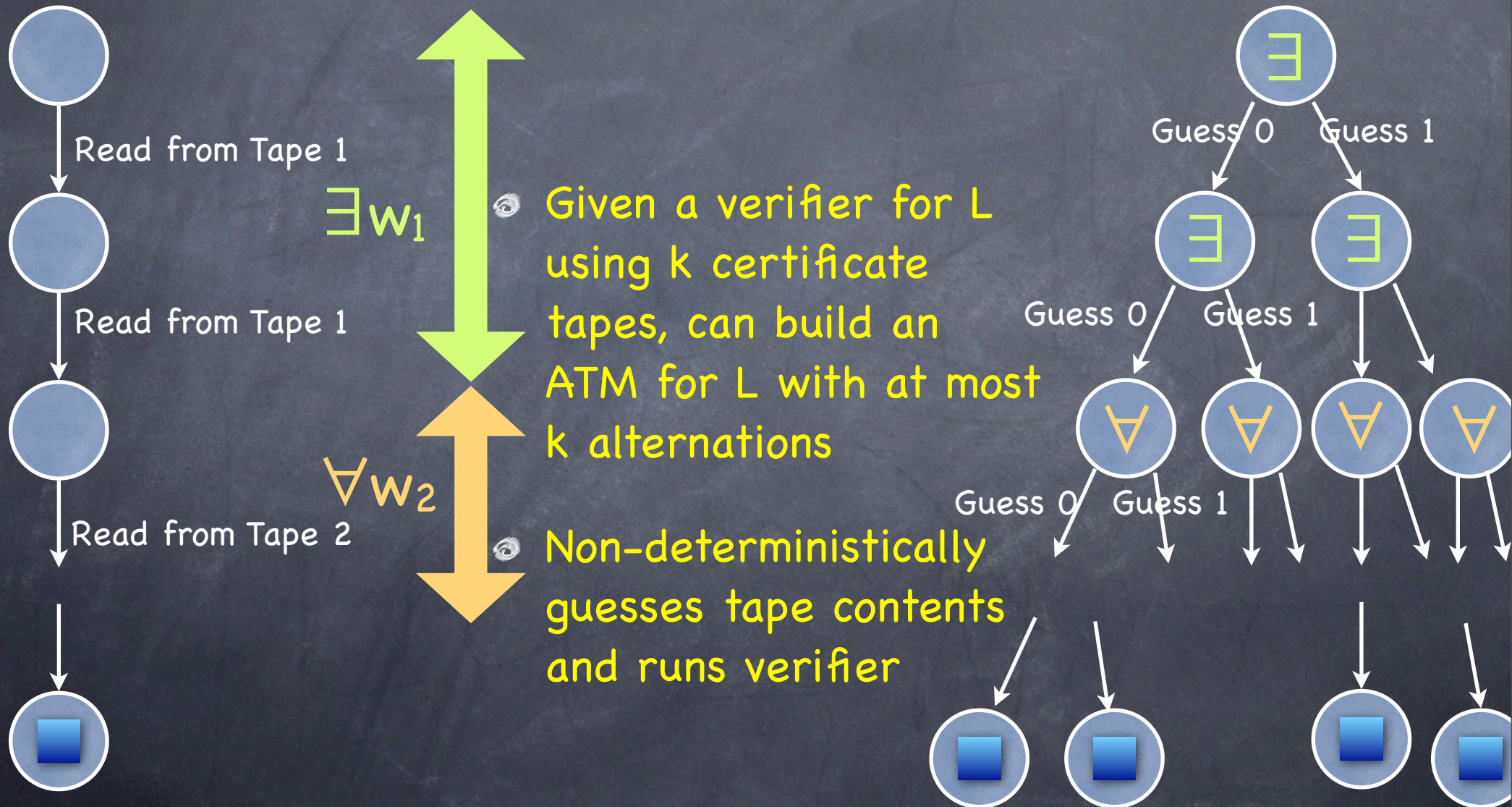


ATM

- Two kinds of configurations: \exists and \forall
 - Depending on the state, say
 - A \exists configuration is accepting if either child is accepting
 - A \forall configuration is accepting only if both children are accepting

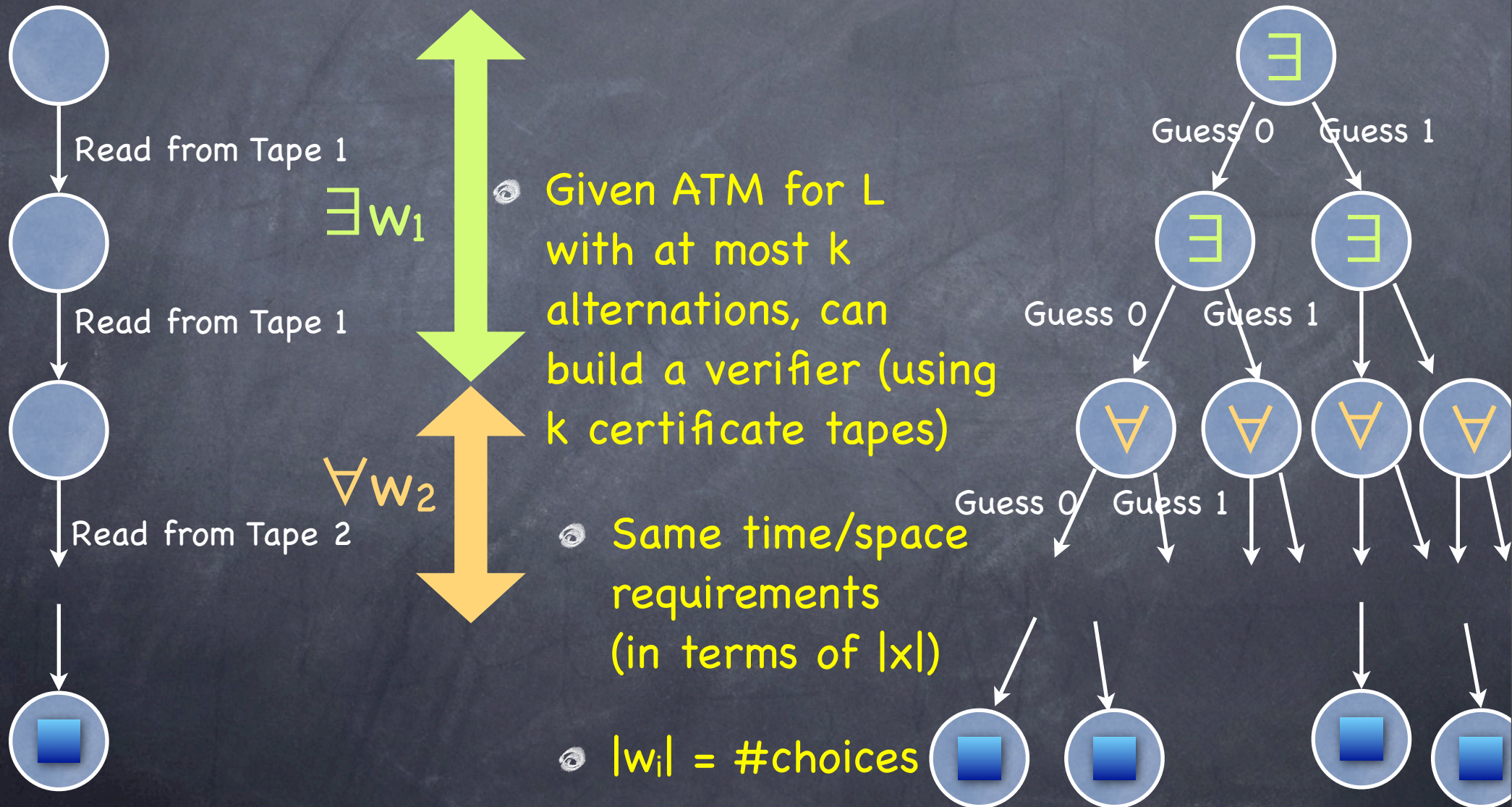


Verification → Non-determinism



Verification ←

Non-determinism



Time, Space, Alternations

- Complexity measures
 - Time: Maximum number of steps in any thread
 - Space: Maximum space in any configuration reached
 - Alternations: Maximum number of quantifier switches in any thread

ATIME

- $\Sigma_k \text{TIME}$, $\Pi_k \text{TIME}$

- $\Sigma_k \text{TIME}(T)$: languages decided by ATMs with at most k alternations starting with \exists , in time $T(n)$

- $\Sigma_k \text{TIME}(\text{poly}) = \Sigma_k^P$

- Latter being exactly the certificate version

- ATIME

- $\text{ATIME}(T)$: languages decided by ATMs in time $T(n)$

ATIME vs. DSPACE

- $ATIME(T) \subseteq DSPACE(T^2)$
 - c.f. $NTIME(T) \subseteq DSPACE(T)$
 - $AP \subseteq PSPACE$
- But $PSPACE \subseteq AP$
 - TQBF in AP (why?)
- $AP = PSPACE$

$ATIME(T) \subseteq DSPACE(T^2)$

- Evaluate if the start configuration is accepting, recursively
 - A \exists configuration is accepting if either child is and a \forall configuration is accepting if both children are
- Space needed: depth \times size of configuration
 - Depth = # alternations = $O(T)$. Config size = $O(T)$ as any thread runs for time $O(T)$
 - $O(T^2)$

ASPACE vs. DTIME

- $ASPACE(S) = DTIME(2^{O(S)})$
 - Recall, already seen $NSPACE(S) \subseteq DTIME(2^{O(S)})$
 - Poly-time connectivity in configuration graph of size at most $2^{O(S)}$
 - Instead of connectivity, can recursively label all accepting nodes (2 lookups per node: in $\text{poly}(S)$ time). So $ASPACE(S) \subseteq DTIME(2^{O(S)})$
 - To show $DTIME(2^{O(S)}) \subseteq ASPACE(S)$

$DTIME(2^{O(S)}) \subseteq ASPACE(S)$

- To decide, is configuration after t steps accepting
 - Accept configuration, with unique first cell α (blank tape cell and unique accept state)
 - Once there, stays there
 - Is first cell of config after t steps α
 - $C(i,j,x)$: if after i steps, j^{th} cell of config is x
 - Need to check $C(t,1,\alpha)$

ATM for TM simulation

- $C(i,j,x)$: if after i steps, j^{th} cell of config is x
 - Recall reduction in Cook's theorem
 - If $C(i-1,j-1,a), C(i-1,j,b), C(i-1,j+1,c)$ then $C(i,j,x)$ iff $x=F(a,b,c)$
 - $C(i,j,x): \exists a,b,c$ st $x=F(a,b,c)$ and $C(i-1,j-1,a), C(i-1,j,b), C(i-1,j+1,c)$
 - Base case: $C(0,j,x)$ easy to check from input
 - Naive recursion: Extra $O(S)$ space at each level for $2^{O(S)}$ levels!

ATM for TM simulation

- ATM to check if $C(i,j,x)$

- $C(i,j,x): \exists a,b,c$ st $x=F(a,b,c)$ and $C(i-1,j-1,a), C(i-1,j,b), C(i-1,j+1,c)$

- **Tail-recursion** (in parallel forks)

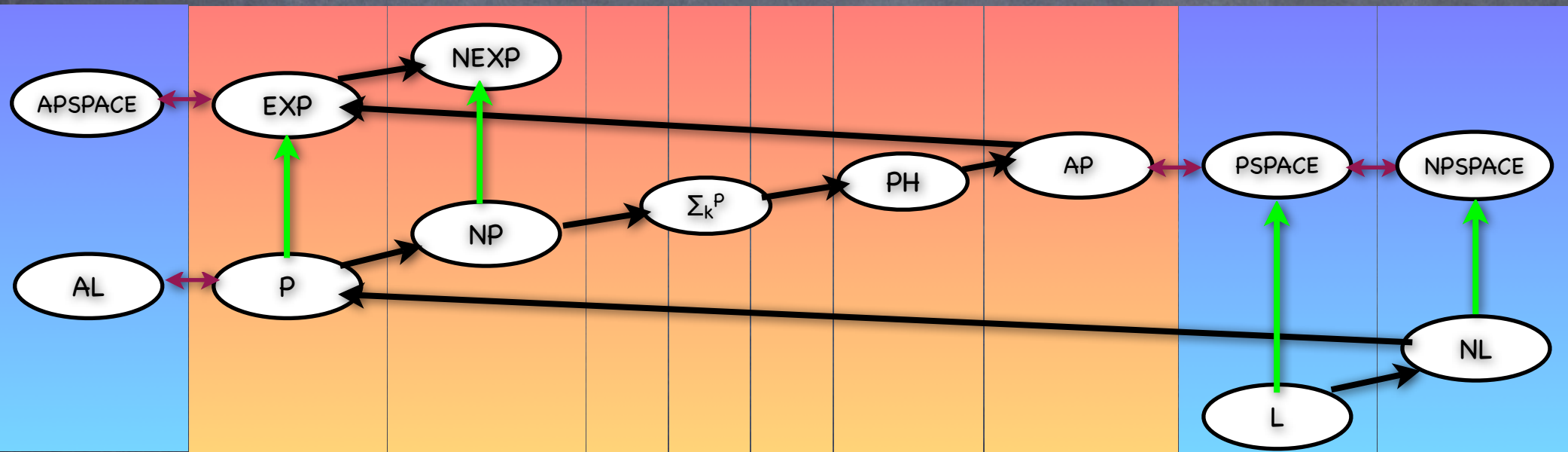
Gets the AND check for free. No need to use a stack.

- Check $x=F(a,b,c)$; then enter universal state, fork out for each of the three configurations to be checked
- Overwrite $C(i,j,x)$ with $C(i-1,...)$ and reuse space
- Stay within the same $O(S)$ space at each level!

ASPACE vs. DTIME

- $ASPACE(S) = DTIME(2^{O(S)})$
 - $APSPACE = EXP$
 - $AL = P$

Zoo



DTISP(T,S)

- **Theorem:** $\text{NTIME}(n) \not\subseteq \text{DTISP}(n^{1+\epsilon}, n^\delta)$ for some $\epsilon, \delta > 0$
- i.e., cannot solve SAT in some slightly super-linear time and slightly super-logarithmic space
 - Commonly Believed: can't solve in less than exponential time or with less than linear space
- Follows (after careful choice of parameters) from
 - $\text{DTISP}(T,S) \subseteq \Sigma_2\text{TIME}(T^{1/2} S)$
 - $\text{NTIME}(n) \subseteq \text{DTIME}(n^{1+\epsilon}) \Rightarrow \Sigma_2\text{TIME}(T) \subseteq \text{NTIME}(T^{1+\epsilon})$
 - $\text{NTIME}(n) \subseteq \text{DTISP}(n^{1+\epsilon}, n^\delta) \Rightarrow \text{NTIME}(n^\dagger) \subseteq \text{NTIME}(n^{\dagger(1/2+\epsilon')}) !$

quantification to guess intermediate configs, check consecutive ones good

Today

- ATM to define levels of PH
 - ATIME and ASPACE
 - $AP = PSPACE$ and $APSPACE = EXP$
- Using Σ_2 TIME for a DTISP lower-bound