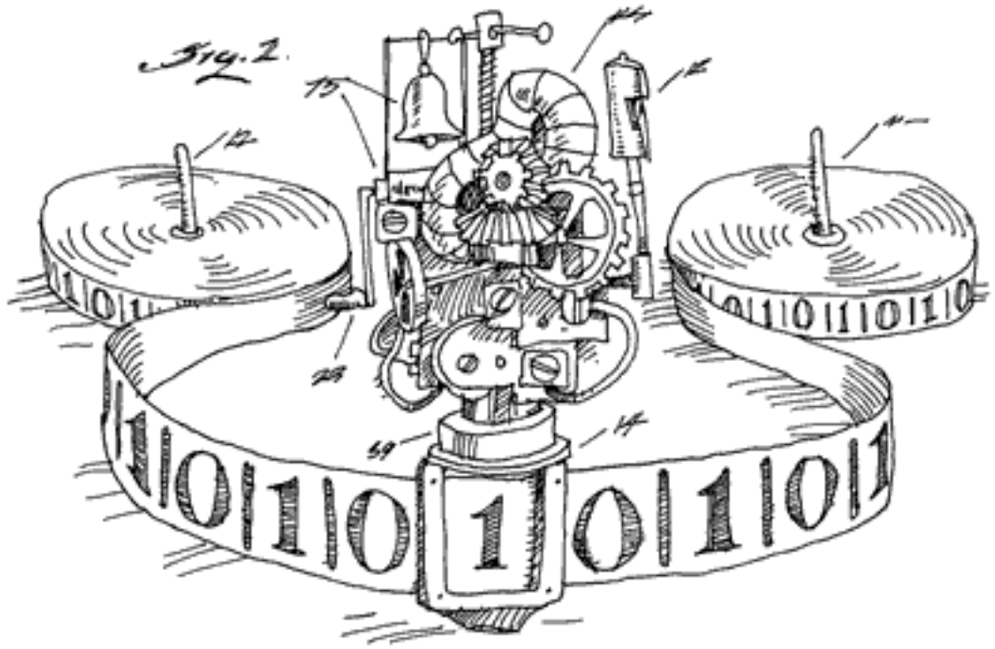
CSC4010: Introduction to Theoretical Computer Science



Lecture 16

Circuit Complexity of a Binary Language

Let $Y \subseteq \{0,1\}^*$. Let $Y_n = Y \cap \{0,1\}^n$.

Definition: The circuit complexity of Y is the function $S : \mathbb{N} \to \mathbb{N}$ defined by $S(n)$ = the size of the smallest circuit that computes Y_n.

Let $S : \mathbb{N} \to \mathbb{N}$ be a function

Definition: SIZE$(S) = \{Y \subseteq \{0,1\} * :$ the circuit complexity of Y is $O(S)\}$

Definition: PSIZE $= \{Y \subseteq \{0,1\}^* :$ circuit complexity of Y is poly$(n)\}$

$$\text{PSIZE} = \cup_{k=1}^{\infty} \text{SIZE}(n^k)$$

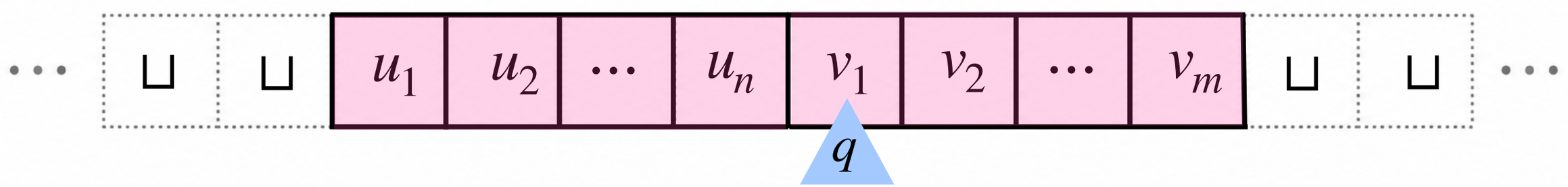
Circuit Complexity vs. time complexity

Theorem: Let $T : \mathbb{N} \to \mathbb{N}$ be any time constructible function (time bound). Then, TIME(T) $\subseteq$ SIZE(T^2) . In particular, $P \subseteq$ PSIZE .

- Polynomial Time Algorithms imply a family of polynomial size circuits for a problem.
- The proof is based on computational histories.

Recap: Configurations of a Turing Machine

- Let $M = (Q, q_{\text{accept}}, q_{\text{reject}}, q_0, \delta, \sqcup, \Sigma)$ be a Turing Machine.
- A configuration uqv of M is a string over $\Sigma \cup Q$
- The tape currently contains uv .
  - The machine is in state q and the “head” is pointing at the first symbol of v .



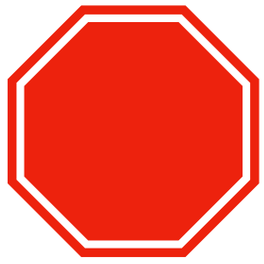
Recap: The Initial Configuration

- Let $w \in \{0,1\}^*$ be the input.
- The initial configuration of M on w is $q_0 \sqcup w$.

Recap: The "next" configuration

- For any configuration uqv, we define NEXT(uqv) as follows:
  - Break uqv into individual symbols: $uqv = u_1, u_2, \cdots, u_n q v_1, v_2, \cdots, v_m$.
  - If $\delta(q, v_1) = (q', b, R)$, then NEXT($uqv$) $= u_1 u_2 \cdots u_n b q' v_2, v_3 \cdots v_m$.
    - **Edge Case:** If $m = 1$, then NEXT(uqv) = $u_1 u_2 \cdots u_n b q' \sqcup$.
  - If $\delta(q, v_1) = (q', b, L)$, then NEXT($uqv$) $= u_1 u_2 \cdots u_{n-1} q' u_n b v_2, v_3 \cdots v_m$.
    - **Edge Case:** If $n = 0$, then NEXT(uqv) = $q' \sqcup b v_2 v_3 \cdots v_m$.
  - We write $\text{NEXT}_M(uqv)$ if M is not clear from context.

Recap: Halting Configurations



- An accepting configuration is of the form $uq_{\text{accept}}v$.
- An rejecting configuration is of the form $uq_{\text{reject}}v$.
- A halting configuration is either an accepting or a rejecting configuration.

Recap: Computational History

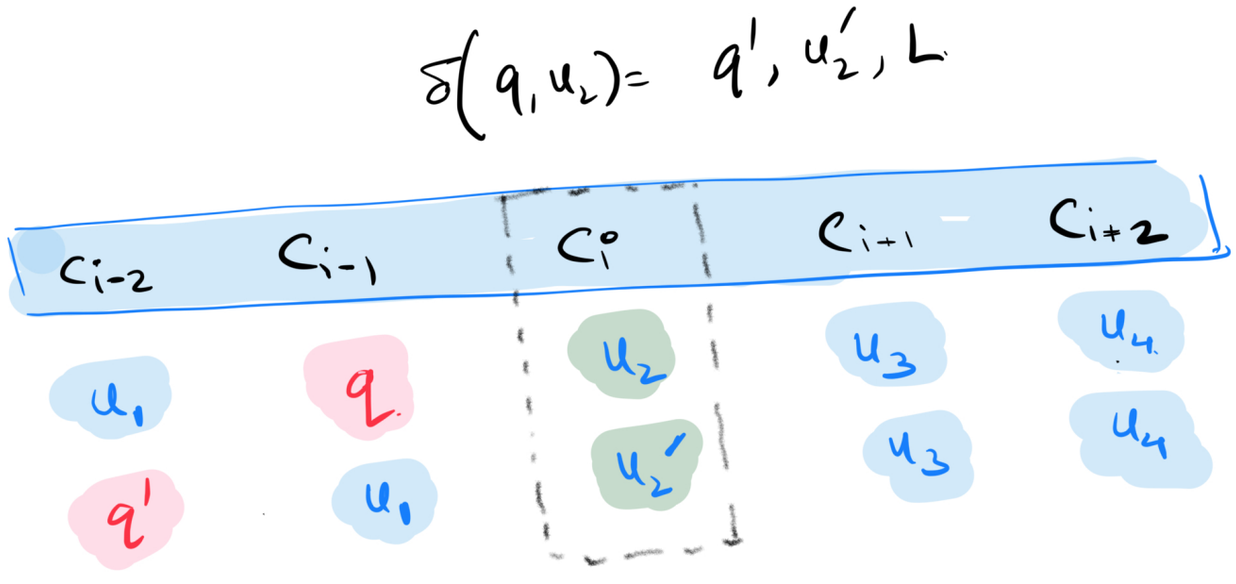
- Let $w \in \{0,1\}^*$ be an input string.
- Let C_0 be the initial configuration of M on w i.e. $C_0 = q_0 \sqcup w$.
- Inductively, for each $i \in \mathbb{N}$, let NEXT$(C_i) = C_{i+1}$.
- The computational history of M on w is the sequence $C_1, C_2, \cdots, C_T$, where C_T is the first halting configuration in the sequence.
- If there is no such C_T, then the computational history is infinite: $C_1, C_2, C_3, \cdots$.

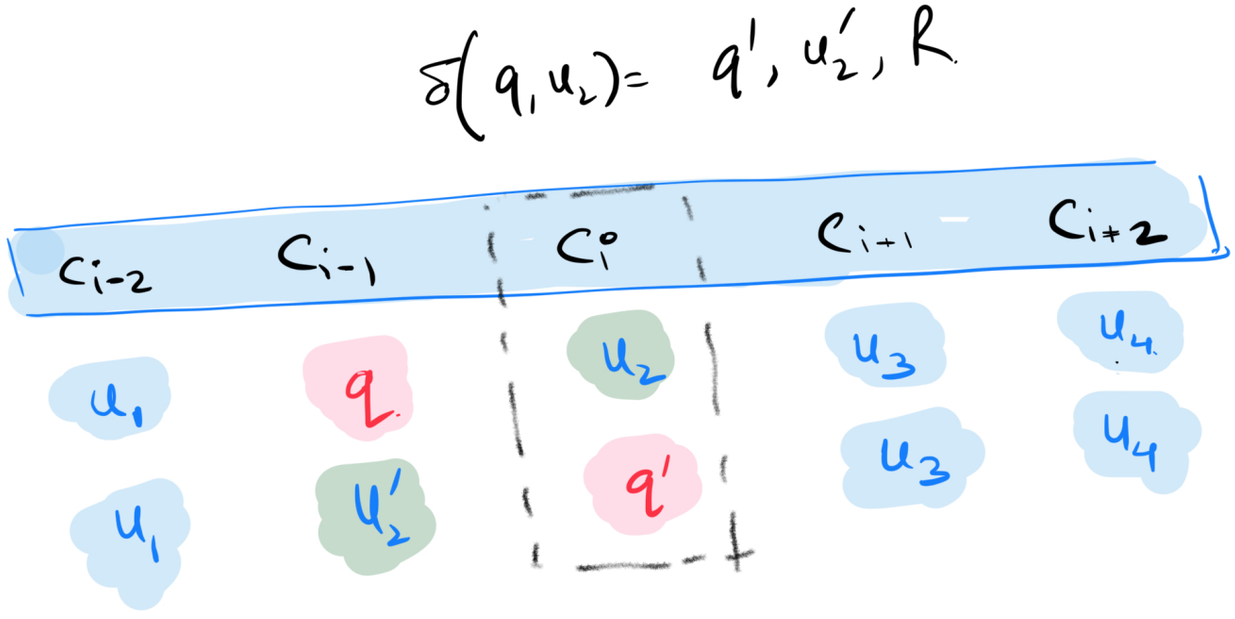
Locality of Computation

- Let C be a configuration of a Turing Machine M.
- We can write $C = c_1 c_2 \cdots c_l$ for some $c_i \in \Sigma \cup Q$.
- Then, NEXT$(C) = c_1' c_2' \cdots c_l'$ for some $c_i' \in \Sigma \cup Q$.

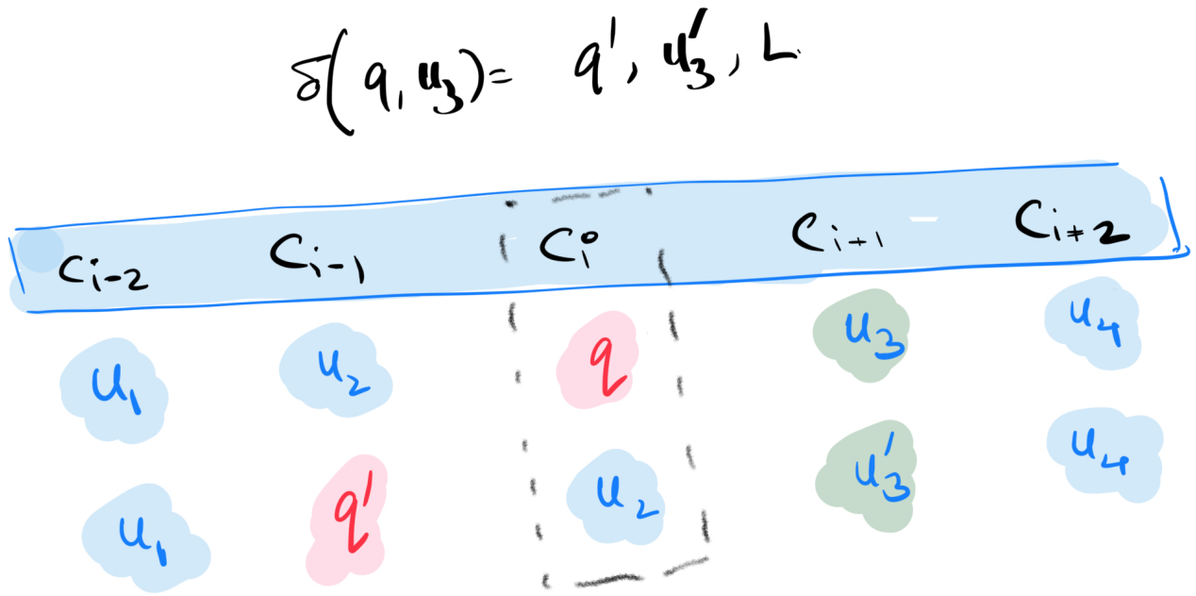
$c_i' \neq c_i \Rightarrow \quad c_i \in Q$ or $c_{i-1} \in Q$ or $c_{i+1} \in Q$.

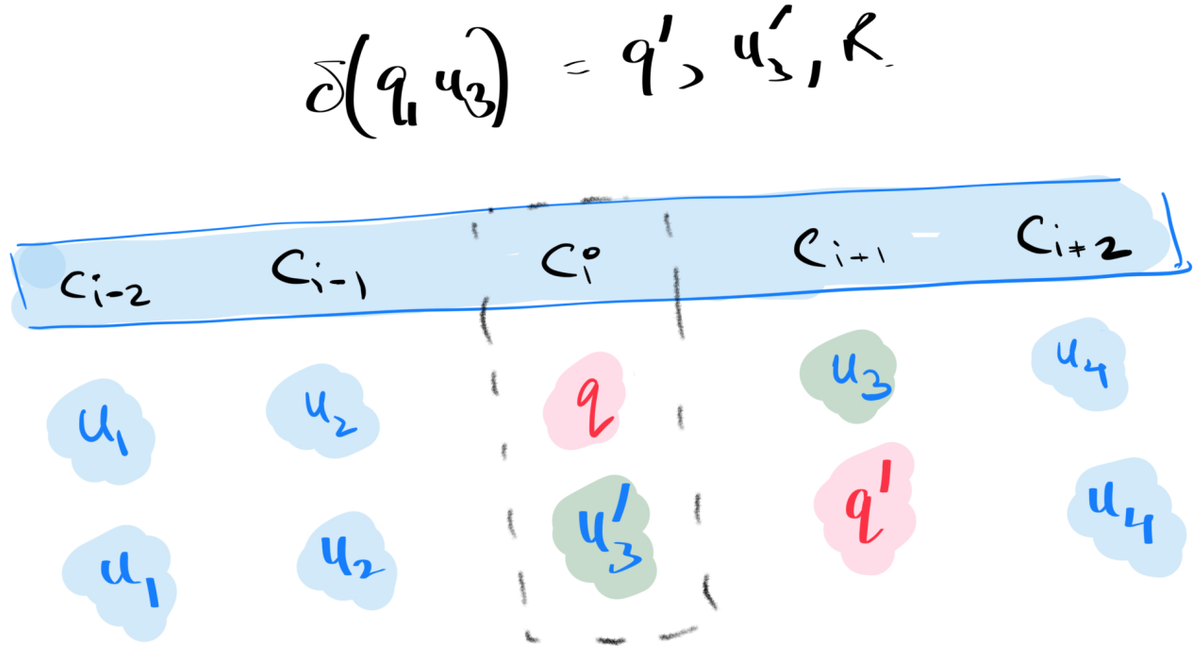
Locality of Computation



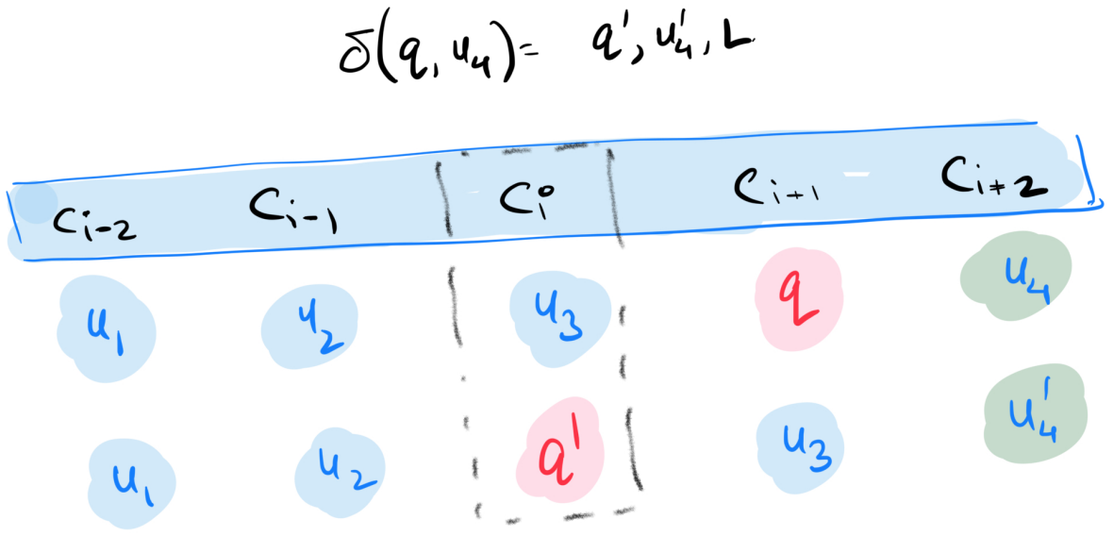


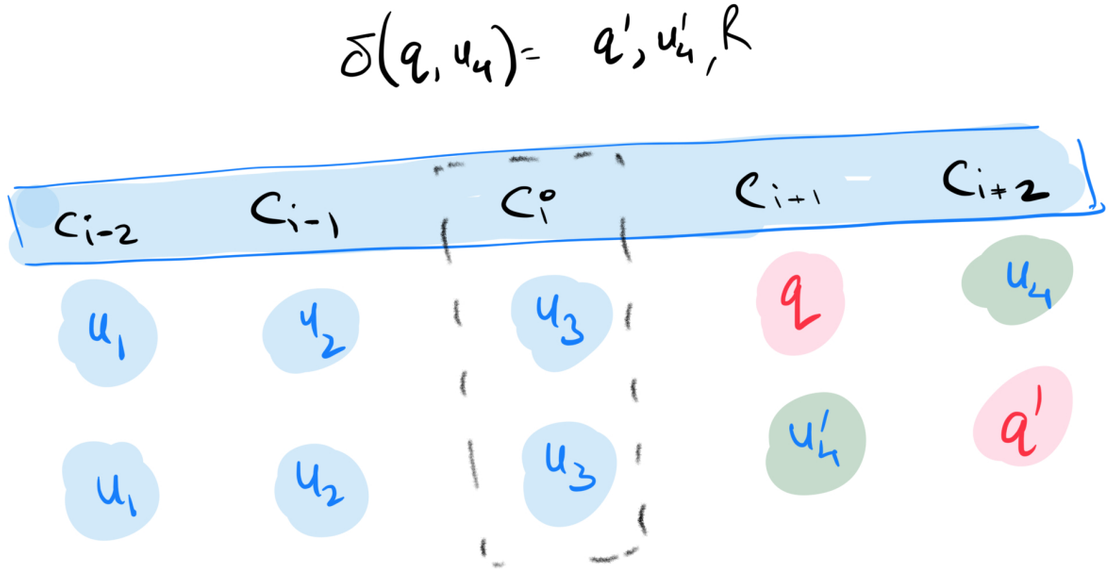
Locality of Computation





Locality of Computation





Encoding Configurations in Binary

- Let C be a configuration of a Turing Machine M, say $C = u_1 u_2 \cdots u_k q v_1 v_2 \cdots v_m$.
- Each symbol/state $b \in \Sigma \cup Q$ can be encoded in binary as $\langle b \rangle \in \{0,1\}^r$ for some $r = O(1)$.
- We define $\langle C \rangle = \langle u_1 \rangle \langle u_2 \rangle \cdots \langle u_k \rangle \langle q \rangle \langle v_1 \rangle \cdots \langle v_m \rangle$.

From one configuration to the next

- Suppose $\langle C \rangle = B_1 B_2 \cdots B_l$ where $B_i \in \{0,1\}^r$
- Suppose $\langle \text{NEXT}(C) \rangle = B'_1 B'_2 \cdots B'_l$ where $B'_i \in \{0,1\}^r$
- If we want to know B'_i, we do not need to look at all of $\langle C \rangle$.
- For $\langle B'_i \rangle$ sufficient to consider $B_{i-1}, B_i, B_{i+1}, B_{i+2}$.

From one configuration to the next

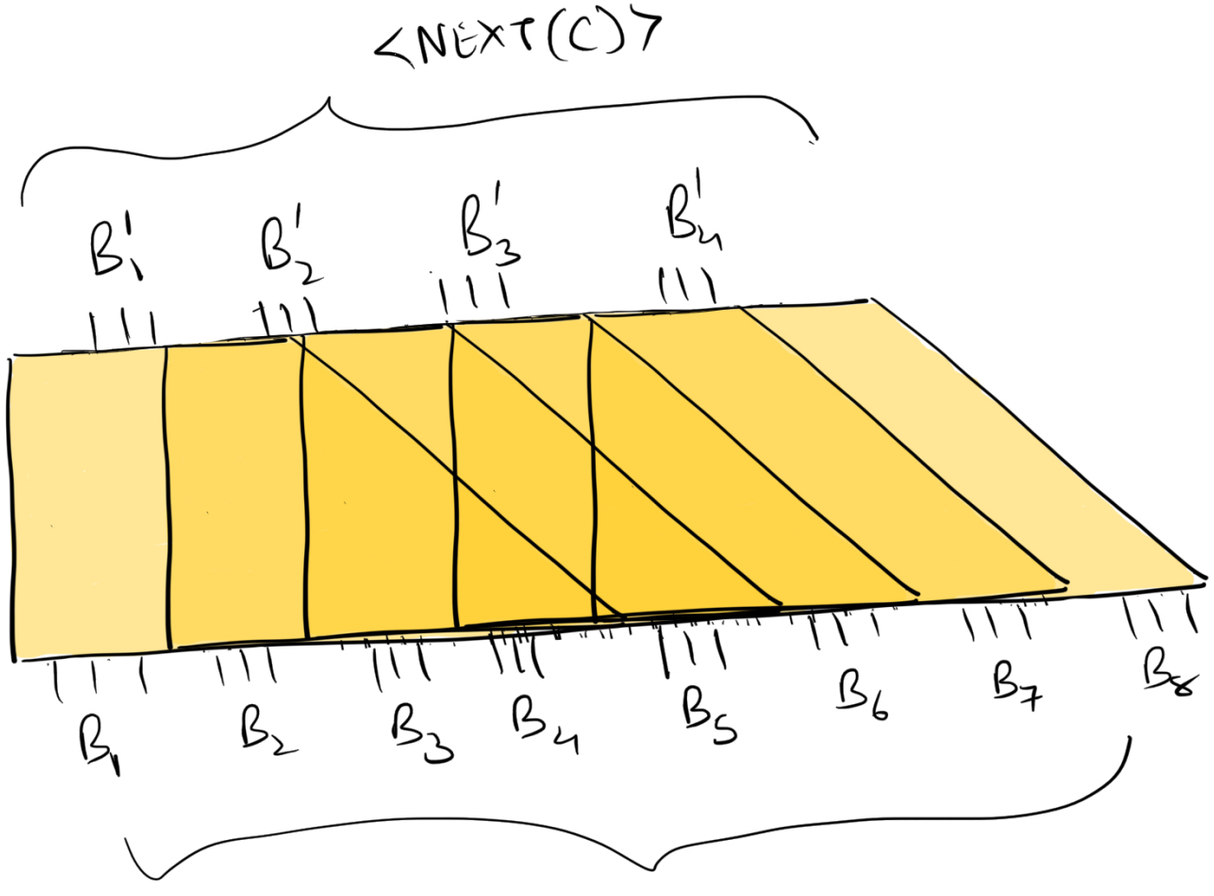
- There is a function $\Delta_M : \{0,1\}^{4r} \to \{0,1\}^r$ such that for every configuration C, if $\langle C \rangle = B_1 B_2 \cdots B_l$ and $\langle \mathsf{NEXT}(C) \rangle = B'_1 B'_2 \cdots B'_l$ then for every i, we have,

$$\Delta_M(B_{i-1} B_i B_{i+1} B_{i+2}) = B'_i$$

- Intuitively, Δ_M is a version of the transition function of M.

From one configuration to the next

- There is a circuit of size $O(1)$ that computes Δ_M
- Combine the copies in parallel:

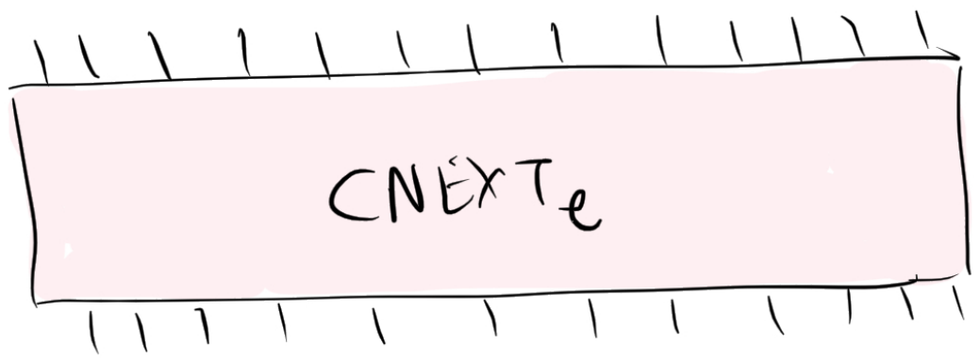


From one configuration to the next

- The construction on the previous slide shows that for every $\ell \in \mathbb{N}$, there is a circuit $\mathrm{CNEXT}_\ell : \{0,1\}^{r\ell} \to \{0,1\}^{r\ell}$ satisfying:
  - If C is a configuration such that $|C| = |\mathrm{NEXT}(C)| = l$, then

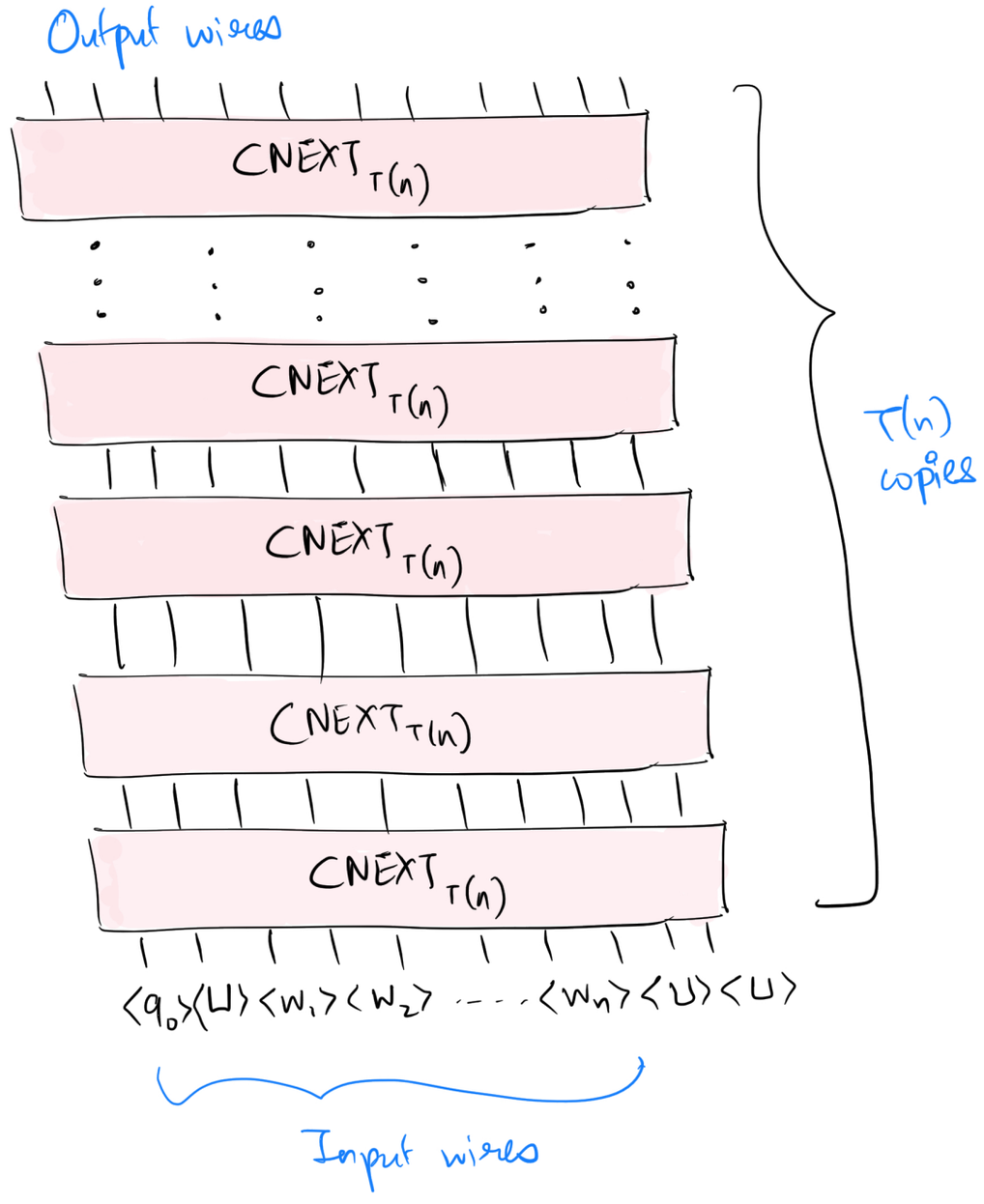
$$\mathrm{CNEXT}_\ell(\langle C \rangle) = \langle \mathrm{NEXT}(C) \rangle$$

- The size of CNEXT_ℓ is $O(\ell)$.



Converting TM to Circuit

- Let M be a Turing Machine that decides L with time complexity $T(n)$.
- We get a circuit of size $O(T(n)^2)$ computing $L_n = \{0,1\}^n \cap L$.



Circuit Complexity vs. time complexity

Theorem: Let $T : \mathbb{N} \to \mathbb{N}$ be any time constructible function (time bound). Then, TIME(T) $\subseteq$ SIZE(T^2) . In particular, $P \subseteq$ PSIZE .