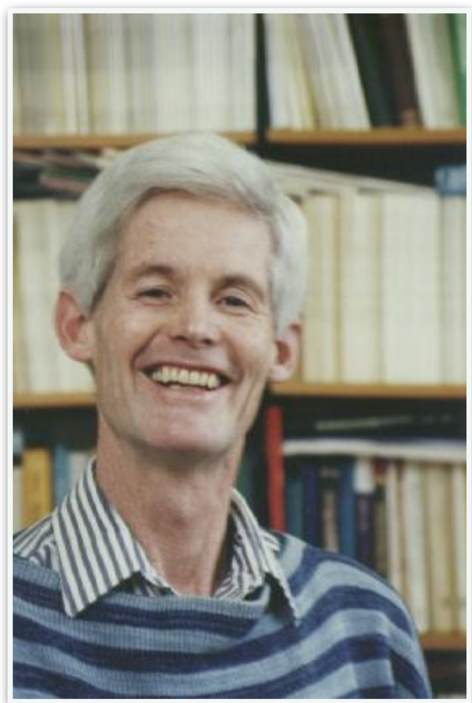
# Computabilità, Complessità e Logica

## Lezione 12

# Teorema di Cook-Levin

Indipendentemente Cook (1971) e Levin (1973) dimostrarono che esiste un problema NP-completo



**Stephen Cook**

**Leonid Levin**

SAT (soddisfacibilità Booleana) è $\mathrm{NP}$-completo

# Cosa è SAT?

- SAT chiede se data una formula in logica proposizionale esiste un assegnamento che la soddisfa.
- Per capire il problema ci serve sapere:
    - Cosa è una formula in logica proposizionale?
    - Cosa è un assegnamento?
    - Cosa significa che un assegnamento soddisfa una formula?

# Congiunzione, disgiunzione, negazione

- Un breve ripasso della notazione che utilizziamo
  - $a \wedge b$ è la congiunzione di $a$ e $b$
  - $a \vee b$ è la disgiunzione di $a$ e $b$
  - $\neg a$ è la negazione di $a$
- Useremo anche:
  - l'implicazione $a \rightarrow b$ come abbreviazione di $\neg a \vee b$
  - l'abbreviazione $a \leftrightarrow b$ per $(a \rightarrow b) \wedge (b \rightarrow a)$

# Formule

- Sia $V = \{x_1, \ldots, x_n\}$ un insieme di variabili proposizionali
- Le formule sono definite in modo induttivo come:
  - $\varphi \in V$ è una formula
  - Se $\varphi$ è una formula, allora $\neg\varphi$ è una formula
  - Se $\varphi$ e $\psi$ sono formule, allora $(\varphi \wedge \psi)$ e $(\varphi \vee \psi)$ sono formule

# Formule: esempi

- Se $V = \{A, B, C\}$ allora
  $(A \wedge B)$ è una formula,
  $A$ e $\neg A$ sono formule,
  $(A \vee (B \wedge C))$ è una formula, etc.

- Ma $A \wedge$ , $AB$, $A \neg B$ **non** sono formule ben formate

- Generalmente assegnamo un significato alle variabili.
  Per esempio “oggi piove” è $A$ “oggi c’è vento” è $B$, quindi
  $(A \wedge B)$ andrà a significare “oggi piove e oggi c’è vento”

# Assegnamenti

- Data una formula, per esempio $((A \vee B) \wedge (\neg B \vee C))$ possiamo assegnare ad ogni variabile un valore vero (true) o falso (false)

- Possiamo quindi vedere un assegnamento come una funzione $V \rightarrow \{t, f\}$ o come un vettore $\overrightarrow{x}$ in cui in posizione $x_i$ c'è il valore (in $\{t, f\}$) da assegnare all'i-esima variabile in $V$

- Per esempio $\overrightarrow{x} = (t, f, f)$ ci dice che assegnamo ad $A$ il valore $t$ e a $B$ e $C$ il valore $f$

# Valutazione

- Data una formula ed un assegnamento possiamo valutare la formula
- Il valore di verità di una variabile proposizionale è dato direttamente dall'assegnamento
- $\neg\varphi$ è vera se la formula $\varphi$ è falsa
- $\varphi \land \psi$ è vera se entrambe le formule che la compongono sono vere
- $\varphi \lor \psi$ è vera se almeno una delle due formule che la compongono è vera
- Diciamo che una formula $\varphi$ è soddisfatta se esiste un assegnamento che la rende vera

# Notazione

- Indicheremo con
  $$\bigwedge_{i=1}^{m} \varphi_i$$
  la formula $\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_m$

- Indicheremo con
  $$\bigvee_{i=1}^{m} \varphi_i$$
  la formula $\varphi_1 \vee \varphi_2 \vee \ldots \vee \varphi_m$

- Per esempio
  $$\bigwedge_{i=1}^{2} \bigvee_{j=1}^{2} \varphi_{i,j} = (\varphi_{1,1} \vee \varphi_{1,2}) \wedge (\varphi_{2,1} \vee \varphi_{2,2})$$

# SAT è contenuto in $\mathrm{NP}$

- Data una formula $\varphi$ e un assegnamento $\overrightarrow{x}$ di variabili possiamo verificare che $\varphi$ è soddisfatta calcolando $\varphi(\overrightarrow{x})$,

- La verifica che una formula soddisfi un assegnamento in tempo polinomiale. Quindi data $\varphi$ possiamo usare $\overrightarrow{x}$ come certificato.

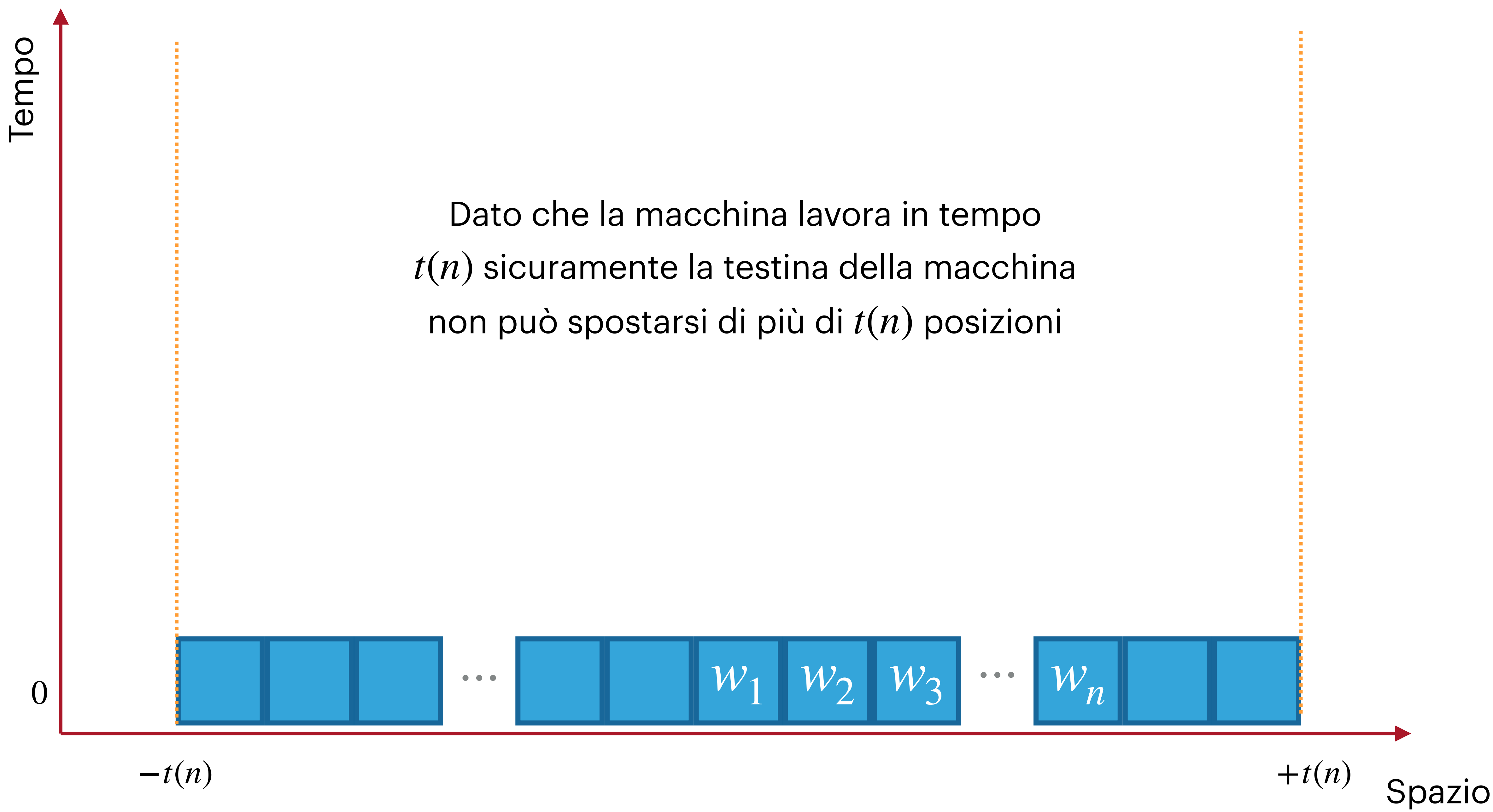
- Se vogliamo invece usare la definizione di $\mathrm{NP}$ con macchine non deterministiche, possiamo generare in modo non deterministico un assegnamento $\overrightarrow{x}$ di $\varphi$ e verificare se $\overrightarrow{x}$ soddisfa $\varphi$ e, in quel caso accettare. Questo richiede tempo polinomiale non-deterministico

# SAT è contenuto in $\mathrm{NP}$

- Dobbiamo ora mostrare che ogni problema in $\mathrm{NP}$ si riduce (in tempo polinomiale) a SAT
- Dato che per ogni problema in $\mathrm{NP}$ esiste una MdT non deterministica che lavora in tempo polinomiale che lo decide…
- …possiamo vedere se possiamo usare una istanza di SAT per "simulare" una MdT non deterministica
- Data una MdT non deterministica $M$ che lavora in tempo $t(n)$ e un input $w$ dobbiamo scrivere una formula $\varphi$ che è soddisfacibile se e solo se esiste una computazione accettante di $M$ su input $w$

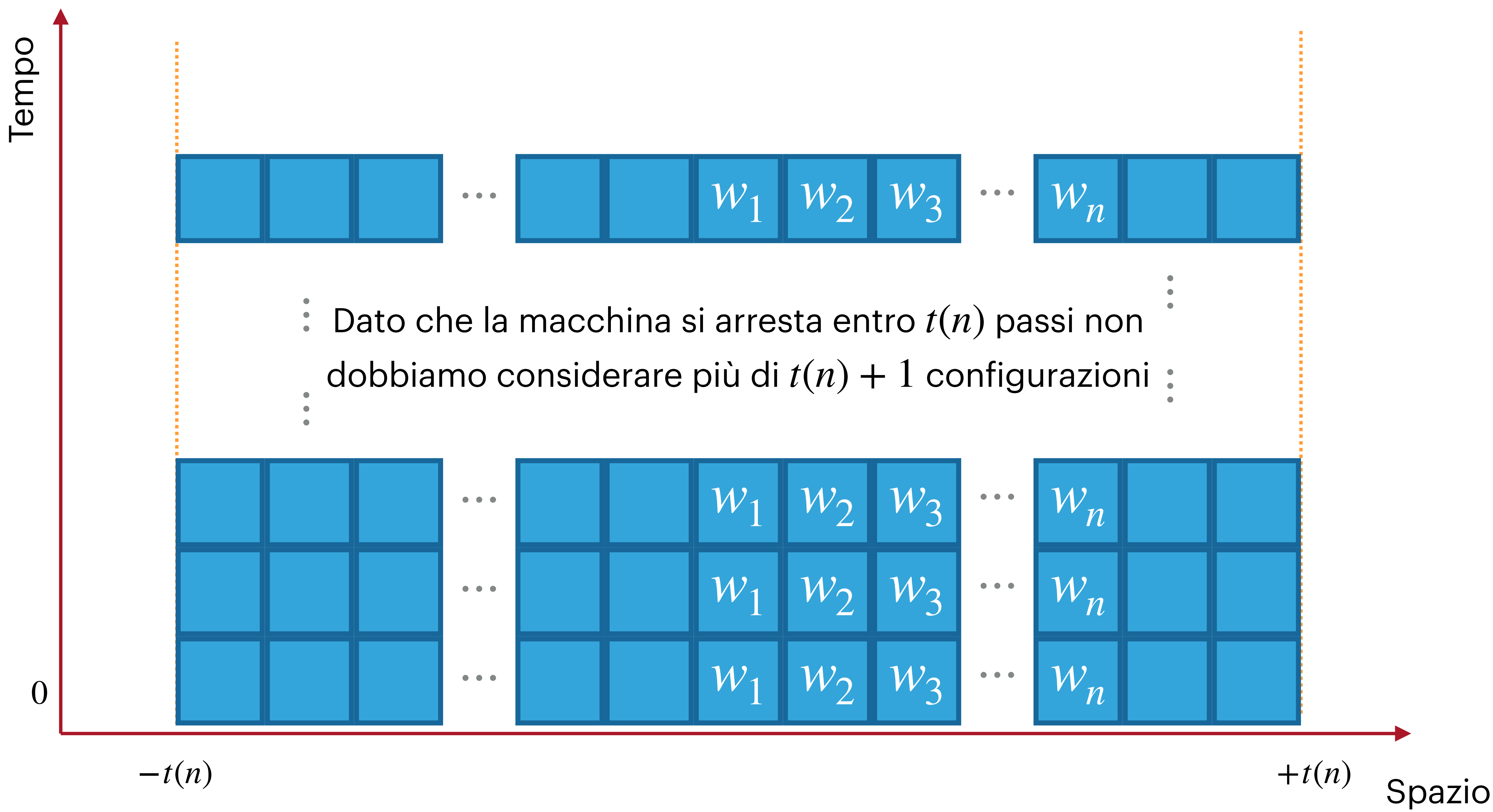
# Diagramma spazio-tempo di una MdT

Supponiamo di avere una NDTM $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ che lavora in tempo $t(n)$



# Diagramma spazio-tempo di una MdT

Supponiamo di avere una NDTM $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ che lavora in tempo $t(n)$



# Idea della dimostrazione

- Definiamo delle variabili che ci definiscono tutte le $t(n) + 1$ configurazioni (ognuna consistente di $2t(n) + 1$ celle)

  - Esempio: "la cella $i$ al tempo $j$ contiene il simbolo $a$"

- Definiamo una formula che è vera se e solo se esiste un assegnamento delle variabili che rappresenta una computazione accettante

- Peer avere una riduzione in tempo polinomiale questa formula deve essere costruibile in tempo polinomiale a partire dall'input $w \in \Gamma^\star$ di lunghezza $n$

# Variabili utilizzate

- $c_{\sigma,i,j}$ per $\sigma \in \Gamma$, $i \in \{-t(n), \ldots, t(n)\}$, e $j \in \{0,\ldots,t(n)\}$ indica che la cella $i$ contiene il simbolo $\sigma$ al tempo $j$
  - Esempio: $c_{a,4,5}$ significa "la cella $4$ al tempo $5$ contiene il simbolo $a$
- $p_{i,j}$ per $i \in \{-t(n), \ldots, t(n)\}$ e $j \in \{0,\ldots,t(n)\}$ indica che la testina della macchine è sulla cella $i$ al tempo $j$
- $e_{q,j}$ per $q \in Q$ e $j \in \{0,\ldots,t(n)\}$ indica che la macchina si trova nello stato $q$ al tempo $j$

# Variabili utilizzate

- Le variabili della forma $c_{\sigma,i,j}$ sono $|\Gamma|(t(n)+1)(2t(n)+1)$
- Le variabili della forma $p_{i,j}$ sono $(t(n)+1)(2t(n)+1)$
- Le variabili della forma $e_{q,j}$ sono $|Q|(t(n)+1)$
- Dato che $t(n)$ è polinomiale rispetto a $|w|=n$, il numero di variabili rimane polinomiale

# Cosa deve verificare la formula

- Le $t(n) + 1$ configurazioni sono valide:
  - C'è sempre esattamente un simbolo per cella
  - C'è esattamente uno stato della macchina
  - La testina è in esattamente una posizione sul nastro
- La configurazione iniziale è corretta
- L'ultima configurazione è accettante
- Le transizioni sono valide
- *Vediamo come codificare ciascuna di queste condizioni*

# Cosa deve verificare la formula

- Per ognuna di queste condizioni daremo una formula
- Consideriamo tutte le formule messe in congiunzione
- Ovvero tutte queste sotto-formule devono essere vere tutte affinché la formula risultate sia vera
- Avremo quindi che la formula è soddisfacibile se:
  - Ognuna della $t(n) + 1$ configurazioni è valida, le transizioni sono valide, la computazione è accettante
  - Ovvero se la MdT non deterministica $M$ che andiamo a modellare accettare su input $w$ entro $t(n)$ passi

# C'è esattamente un simbolo per cella

- La formula è

$$\bigwedge_{i=-t(n)}^{t(n)} \bigwedge_{j=0}^{t(n)} \bigvee_{\sigma\in\Gamma} \left( c_{\sigma,i,j} \wedge \bigwedge_{\sigma'\neq\sigma} \neg c_{\sigma',i,j} \right)$$

- Che significa che per ogni posizione $i$ del nastro e per ogni istante temporale $j$ deve valere che c'è un simbolo $\sigma$ nella casella $j$ e nessun altro simbolo nella stessa casella

- Per esempio, per $i = 1,\ j = 0$ e alfabeto $\Gamma = \{a, b, \#\}$ avremo la formula:
$(c_{a,1,0} \wedge \neg c_{b,1,0} \wedge \neg c_{\#,1,0}) \vee (c_{b,1,0} \wedge \neg c_{a,1,0} \wedge \neg c_{\#,1,0}) \vee (c_{\#,1,0} \wedge \neg c_{a,1,0} \wedge \neg c_{b,1,0})$

# C'è esattamente uno stato della macchina

- La formula è
$$\bigwedge_{j=0}^{t(n)} \bigvee_{q \in Q} \left( e_{q,j} \wedge \bigwedge_{q' \neq q} \neg e_{q',j} \right)$$

- Che significa che in ogni istante temporale $j$ quando lo stato della macchina è $q$ allora non può essere in nessuno stato $q' \neq q$

- Per esempio all'istante temporale $j = 0$ per $Q = \{q, r\}$ avremmo la formula: $(e_{q,0} \wedge \neg e_{r,0}) \vee (e_{r,0} \wedge \neg e_{q,0})$

# La testina è in esattamente in una posizione sul nastro

- La formula è
$$\bigwedge_{j=0}^{t(n)} \bigvee_{i=-t(n)}^{t(n)} \left( p_{i,j} \wedge \bigwedge_{i' \neq i} \neg p_{i',j} \right)$$

- Che significa che in ogni istante temporale $j$ quando la testina è in posizione $i$ sul nastro allora non è in nessuna altra posizione $i' \neq i$

- Per esempio all'istante temporale $j = 0$ per $t(n) = 1$ avremmo la formula:
$(p_{-1,0} \wedge \neg p_{0,0} \wedge \neg p_{1,0}) \vee (p_{0,0} \wedge \neg p_{-1,0} \wedge \neg p_{1,0}) \vee (p_{1,0} \wedge \neg p_{-1,0} \wedge \neg p_{0,0})$

# La configurazione iniziale è corretta

- Mettiamo una congiunzione delle seguenti formule:

- $e_{q_0,0}$ ovvero al tempo $0$ lo stato è $q_0$

- $p_{0,0}$ ovvero al tempo $0$ la testina è sulla cella $0$

- $c_{w_i,i,0}$ per $i \in \{0,\ldots,n-1\}$ con $w = w_0 w_1 \cdots w_{n-1}$.
  Ovvero in posizione $i$ del nastro c'è l'i-esimo simbolo dell'input

- $c_{\#,i,0}$ per $i \in \{-p(n),\ldots,-1,n,\ldots,p(n)\}$.
  Ovvero, dove non c'è l'input c'è il simbolo di blank

# L'ultima configurazione è accettante

- Per semplificare la notazione consideriamo che, se anche la macchina si arresta prima del tempo $t(n)$ tutte le configurazioni successive al tempo di arresto non cambiano

- Quindi ci basta controllare che lo stato al tempo $t(n)$ sia accettante

- Si esprime con: $e_{q_{\text{final}},t(n)}$

# Le transizioni sono valide

- Questa è la parte più complessa
- Dobbiamo esprimere tre cose:
  - La posizione sotto la testina cambia in modo compatibile con la funzione di transizione
  - Lo stato cambia in modo compatibile con la funzione di transizione
  - Nessuna delle altre celle cambia

# Nessuna delle altre celle cambia

- Per ogni transizione dal tempo $j-1$ al tempo $j$ definiamo la formula
$$\bigwedge_{i=-p(n)}^{p(n)} \neg p_{i,j-1} \to \left( \bigwedge_{\sigma\in\Gamma} c_{\sigma,i,j-1} \leftrightarrow c_{\sigma,i,j} \right)$$

- Che dice "se al tempo $j-1$ la testina non era sulla cella $i$ allora il contenuto della cella $i$ coincide al tempo $j-1$ e $j$"

- Questo significa che il contenuto del nastro non cambia per le posizioni dove non c'è la testina

# Applicazione della funzione di transizione

- Per ogni transizione dal tempo $j-1$ al tempo $j$ definiamo la formula
$$\bigwedge_{i=-t(n)}^{t(n)} \bigwedge_{q\in Q} \bigwedge_{\sigma\in\Gamma} \left(p_{i,j-1} \wedge e_{q,j-1} \wedge c_{\sigma,i,j-1} \to \psi_{q,\sigma,i}\right)$$
- Dove $\psi_{q,\sigma,i}$ è definito come
$$\bigvee_{(q',\sigma',d)\in\delta(q,\sigma)} \left(c_{\sigma',i,j} \wedge e_{q'j} \wedge p_{i+d,j}\right)$$
dove usiamo $d = \{-1,1\}$ per i movimenti ← e → della testina (ci serve indicare come varia la posizione)

# Verso la fine della dimostrazione

- Se prendiamo tutte le formule definite fino ad ora e le mettiamo in congiunzione allora abbiamo che la formula è soddisfacibile solo se rispetta tutte le condizioni elencate

- Ognuna delle sotto-formule ha dimensione polinomiale rispetto a $n$ e anche la formula finale ha dimensione polinomiale rispetto ad $n$ e le operazioni per costruirla sono eseguibili in tempo polinomiale

- Abbiamo quindi mostrato che SAT è $\mathrm{NP}$-completo

# Conseguenze

- Se esiste una soluzione efficiente (in tempo polinomiale deterministico) per SAT allora esiste per ogni problema in $\mathrm{NP}$. E questo mostrerebbe $\mathrm{P} = \mathrm{NP}$

- Chiaramente questo non è stato mostrato

- SAT non è l'unico problema $\mathrm{NP}$-completo noto, ma il primo trovato

- Se abbiamo un problema $L \in \mathrm{NP}$ e vogliamo mostrare che è $\mathrm{NP}$-completo ci basta mostrare che $\mathrm{SAT} \leq_P L$

- Dato che SAT è $\mathrm{NP}$-completo questo implica che anche $L$ lo sia