Lezione 20 – il teorema di Cook-Levin

Lezione del 21/05/2024

La struttura di NP

- Come abbiamo già detto, la dispensa 9 studia due questioni "strutturali" relative alla classe NP
  - la struttura dei problemi che popolano la classe NP
  - la struttura della classe NP
- Sin qui, ci siamo occupati di studiare la struttura dei problemi che popolano NP
  - e abbiamo trovato un modo alternativo per dimostrare che un problema è in NP
- In questa lezione ci occupiamo della seconda questione: vogliamo capire se i problemi che popolano NP sono tutti uguali, per quel che riguarda la loro complessità, oppure ce ne sono alcuni più "difficili di" altri
- E qui, sono certa, starete scalpitando sulle vostre sedie…
  - ma come tutti uguali?! – starete dicendo
  - perché $P \subseteq NP$, e quindi dentro NP ci sono, sicuramente, problemi trattabili computazionalmente
  - ma dentro NP ci sono anche problemi che in P non si riesce proprio a collocarceli
- La domanda, ora, è: **fra i problemi in NP che non si riesce a collocare in P, ce ne sono alcuni più "difficili" di altri?**

La struttura di NP

- La domanda, ora, è: **fra i problemi in NP che non si riesce a collocare in P, ce ne sono alcuni più "difficili" di altri?**
- E qui, di nuovo, starete scalpitando sulle vostre sedie…
  - ma certo che ci sono problemi più difficili degli altri, in NP!
  - Sono i problemi NP-completi!
  - Perché, ce lo ricordiamo bene, se un problema NP-completo appartenesse alla classe P allora sarebbe P = NP!
- Perchè, ricordiamo, un problema (decisionale) Γ è NP-completo se
  - $\Gamma \in$ NP e per ogni altro problema $\Gamma_1 \in$ NP, si ha che $\Gamma_1 \leq \Gamma$
  - e P è chiusa rispetto a $\leq$
- Bene, tutto giusto: i problemi NP-completi sono i problemi "più difficili" in NP
- Certo, ammesso che esistano
- Perché: chi ce lo dice che esiste almeno un problema NP-completo?
- Ce lo dice il **Teorema di Cook-Levin** !

Il Teorema di Cook-Levin

- Ve lo ricordate il (caro, vecchio) problema SAT?
- " dati un insieme X di variabili booleane ed un predicato f, definito sulle variabili in X e in forma congiuntiva normale, decidere se esiste una assegnazione a di valori in {vero, falso} alle variabili in X tale che f(a(X))=vero "
- Dove, ricordiamo, un predicato f è in forma congiuntiva normale se
  - f è la congiunzione di un certo numero di clausole: $f = c_1 \wedge c_2 \ldots \wedge c_m$
  - e ciascuna c_j è la disgiunzione ($\vee$) di letterali, ad esempio $x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4$
- Ebbene, il Teorema di Cook-Levin dice, semplicemente, che

- **TEOREMA di Cook-Levin**: SAT è NP-completo

- Un enunciato facile facile…

Il Teorema di Cook-Levin

- **TEOREMA di Cook-Levin**: SAT è NP-completo

- Un enunciato facile facile…
- Per dimostrarlo occorre mostrare che è possibile ridurre a SAT ***ogni*** problema in NP
- ossia, dobbiamo prendere un ***qualsiasi*** problema Γ in NP mostrare come trasformare le sue istanze in istanze di SAT in modo tale che
  - se x è un'istanza sì di Γ allora l'istanza nella quale x viene trasformata è un'istanza sì di SAT
  - se x è un'istanza no di Γ allora l'istanza nella quale x viene trasformata è un'istanza no di SAT
- Ma in NP troviamo problemi in ambiti diversissimi
  - problemi di acquisto di biglietti aerei senza spendere una fortuna
  - problemi di suddivisione di oggetti sui due piatti di una bilancia mantenendoli in equilibrio
  - problemi di piastrellamento di un pavimento senza lasciare spazi scoperti
  - problemi di scelta di rappresentanti
  - …
- Come facciamo a mostrare come trasformare una ***qualsiasi*** istanza di un ***qualsiasi*** problema in NP in un'istanza di SAT?

Il Teorema di Cook-Levin

- **TEOREMA di Cook-Levin**: SAT è NP-completo

- Come facciamo a mostrare come trasformare un ***qualsiasi*** problema in NP a SAT se i problemi in NP sono così diversi gli uni dagli altri?
- Semplice: sfruttiamo l'unica caratteristica che tutti i problemi in NP hanno in comune: l'appartenere ad NP!
- Ossia la caratteristica di essere accettati da una macchina di Turing non deterministica in tempo polinomiale

- Consideriamo, allora, un generico problema $\Gamma \in$ NP e sia $L_\Gamma \subseteq \{0,1\}^*$ il linguaggio contenente la codifica ragionevole delle istanze sì di Γ
- **e cerchiamo di descrivere sotto forma di espressione booleana il predicato "$x \in L_\Gamma$"**
  - per il momento, ci disinteressiamo della forma congiuntiva normale
  - **la dimostrazione che vi presento in questa lezione è diversa da quella sulle dispense**

Il Teorema di Cook-Levin

- Consideriamo un generico problema $\Gamma \in NP$ e sia $L_\Gamma \subseteq \{0,1\}^*$ il linguaggio contenente la codifica ragionevole delle istanze sì di Γ
- **e cerchiamo di descrivere sotto forma di espressione booleana il predicato "$x \in L_\Gamma$"**
- sia NT_Γ una macchina di Turing non deterministica <u>ad un nastro</u> che accetta, anzi, che decide, L_Γ in tempo polinomiale: ossia, esiste un polinomio p tale che, per ogni $x \in \{0,1\}^*$
  - $ntime(NT_\Gamma, x) \leq p(|x|)$
  - $NT_\Gamma(x) = q_A$ se $x \in L_\Gamma$
  - $NT_\Gamma(x) \neq q_A$ se $x \notin L_\Gamma$
- L'affermazione "$x \in L_\Gamma$" è *logicamente equivalente* all'affermazione

  $\gamma(x)$ = "x (e nient'altro che x) è scritto sul nastro di NT_Γ
  <u>e</u> la testina di NT_Γ è posizionata sul primo carattere di x
  <u>e</u> NT_Γ è nel suo stato iniziale,
  <u>e</u> esiste una sequenza di al più $p(|x|)$ quintuple di NT_Γ che possono essere eseguite una di seguito all'altra e portano la macchina nello stato q_A"

- cioè **$x \in L_\Gamma$ se e soltanto se $\gamma(x)$ è vera**

Da computazione a espressione

- **x ∈ L_Γ se e soltanto se $\gamma(x)$ è vera**
- non resta che **descrivere $\gamma(x)$ sotto forma di una espressione booleana E(x) che sia soddisfacibile se e soltanto se $\gamma(x)$ è vera**
- **E(x)** deve descrivere una **computazione di NT_Γ che ha inizio con x scritto sul suo nastro**
- e, poiché ogni computazione di ogni macchina di Turing è una sequenza di *stati globali*,
- per costruire **E(x)** è necessario introdurre le variabili booleane che descrivano, per ogni passo t della computazione (con $0 \leq t \leq p(|x|)$) lo *stato globale* in cui si troverebbe NT_Γ al passo t della computazione $NT_\Gamma(x)$:
  - un insieme N di variabili booleane che permettano di rappresentare il carattere contenuto in *ciascuna cella* del nastro di lavoro di NT_Γ ad *ogni passo* della computazione NT_Γ (x) ;
  - un insieme M di variabili booleane che permettano di rappresentare lo stato interno di NT_Γ ad *ogni passo* della computazione NT_Γ (x) ;
  - un insieme R di variabili booleane che permettano di rappresentare la cella del nastro di lavoro sulla quale è posizionata la testina di NT_Γ ad *ogni passo* della computazione NT_Γ (x) .
- Vediamo, uno alla volta, questi insiemi di variabili
  - insieme alle condizioni che devono soddisfare perché rappresentino quel che devono rappresentare

Variabili per lo stato interno

- Iniziamo a descrivere l'insieme M di variabili booleane che permettono di rappresentare lo stato interno di NT_Γ ad *ogni passo* della computazione $NT_\Gamma(x)$
- Sia $Q = \{ q_0, q_1, q_2, \ldots, q_k \}$ l'insieme degli stati di NT_Γ ,
  - ove q_0 è lo stato iniziale, $q_1 = q_A$ e $q_2 = q_R$
- L'insieme M di variabili, insieme ad una porzione E_M dell'espressione **E(x)** che stiamo costruendo, servono a descrivere in quale stato interno si trova NT_Γ ad ogni passo della computazione $NT_\Gamma(x)$: ciò che vogliamo è che
  - **ogni volta che i valori assegnati alle variabili in M fanno assumere ad E_M il valore vero**,
  - **osservando i valori assegnati alle variabili contenute in M, dobbiamo essere in grado di rispondere a domande del tipo "è q_4 lo stato interno di NT_Γ al passo 25 della computazione $NT_\Gamma(x)$?".**
- Per ogni passo t (con $0 \leq t \leq p(|x|)$), e per ogni $i \in \{ 0,1,\ldots,k \}$, M contiene una variabile booleana M_i^t :

$$M = \{ M_i^t : 0 \leq t \leq p(|x|) \ \wedge i \in \{ 0,1,\ldots,k \} \}$$

- con il seguente significato: **assegnando a M_i^t il valore vero rappresentiamo il fatto che, al passo t della computazione $NT_\Gamma(x)$, la macchina NT_Γ si trova nello stato q_i**

Variabili per lo stato interno

- Per ogni passo t (con $0 \leq t \leq p(|x|)$), e per ogni $i \in \{0,1,\ldots,k\}$, M contiene una variabile booleana M^p_i :

$$\mathbf{M = \{M^t_i : 0 \leq t \leq p(|x|)) \wedge i \in \{0,1,\ldots,k\}}$$

- con il seguente significato: **assegnando a M^t_i il valore vero rappresentiamo il fatto che, al passo t della computazione $NT_\Gamma(x)$, la macchina NT_Γ si trova nello stato q_i**
- affinché le variabili in M descrivano effettivamente lo stato interno di NT_Γ ad ogni passo della computazione $NT_\Gamma(x)$, dobbiamo imporre che esse siano coerenti:
  - **ad ogni passo della computazione** $NT_\Gamma(x)$ la macchina NT_Γ si trova in uno (ed esattamente uno!) dei suoi stati interni
  - allora dobbiamo fare in modo che possano essere prese in considerazione solo quelle assegnazioni di valori alle variabili in M tali che, per ogni t compreso fra 0 e $p(|x|)$, ad una e una sola delle variabili M^t_0 , M^t_1 : ,..., M^t_k sia assegnato il valore **vero**.
- A questo scopo, introduciamo $p(|x|)+1$ espressioni: $\mathbf{E^0_M , E^1_M , \ldots , E^{p(|x|)}_M}$
- dove $\mathbf{E^t_M}$ è l'espressione nelle variabili in M che corrisponde all'affermazione

al passo t di $NT_\Gamma(x)$ la macchina NT_Γ si trova in uno (ed esattamente uno!) dei suoi stati interni

Variabili per lo stato interno

- E^t_M è l'espressione nelle variabili in M che corrisponde all'affermazione
  al passo t di NT_Γ (x) la macchina NT_Γ si trova in uno (ed esattamente uno!) dei suoi stati interni

- Allora,

$E^t_M = [\, M^t_0 \wedge \neg M^t_1 \wedge \neg M^t_2 \wedge \ldots \wedge \neg M^t_k \,]$ — **NT_Γ, al passo t di NT_Γ (x), si trova nello stato q_0 …**

$\vee \; [\, M^t_1 \wedge \neg M^t_0 \wedge \neg M^t_2 \wedge \ldots \wedge \neg M^t_k \,]$ — **… oppure si trova nello stato q_1 …**

$\vee \; [\, M^t_2 \wedge \neg M^t_0 \wedge \neg M^t_1 \wedge \ldots \wedge \neg M^t_k \,]$ — **… oppure si trova nello stato q_2 …**

$\vee \; \ldots$

$\vee \; [\, M^t_k \wedge \neg M^t_0 \wedge \neg M^t_1 \wedge \ldots \wedge \neg M^t_{k-1} \,]$ — **… oppure si trova nello stato q_k**

- Che assume il valore **vero** se e soltanto se ad esattamente una delle variabili $M^t_0, M^t_1, M^t_2, \ldots, M^t_k$ è assegnato il valore **vero**

Variabili per la posizione della testina

- Secondo step: descriviamo l'insieme R di variabili booleane che permettono di rappresentare la posizione della testina di NT_Γ ad *ogni passo* della computazione NT_Γ (x)

- Osserviamo che, poiché NT_Γ ha a disposizione p(|x|) passi per accettare x, allora non utilizza più di p(|x|) celle del nastro
- allora, assumiamo che NT_Γ (x) utilizzi le celle 1, 2, … , p(|x|)

- L'insieme R di variabili, insieme ad una porzione E_R dell'espressione **E(x)** che stiamo costruendo, servono a descrivere su quale cella del nastro di NT_Γ è posizionata la testina ad ogni passo della computazione NT_Γ (x): ciò che vogliamo è che
  - **ogni volta che i valori assegnati alle variabili in R fanno assumere ad E_R il valore vero**,
  - **osservando i valori assegnati alle variabili contenute in R, dobbiamo essere in grado di rispondere a domande del tipo "la testina di NT_Γ è posizionata sulla cella 51 al passo 86 della computazione NT_Γ (x) ?".**

Variabili per la posizione della testina

- Secondo step: descriviamo l'insieme R di variabili booleane che permettono di rappresentare la posizione della testina di NT_Γ ad *ogni passo* della computazione NT_Γ (x)
- Per ogni passo t (con $0 \leq t \leq p(|x|)$), e per ogni $i \in \{ 1,...,p(|x|) \}$, R contiene una variabile booleana R^t_i :

$$R = \{R^t_i : 0 \leq t \leq p(|x|) \ \wedge \ 1 \leq i \leq p(|x|) \}$$

- con il seguente significato: **assegnando a R^t_i il valore vero rappresentiamo il fatto che, al passo t della computazione NT_Γ (x), la testina di NT_Γ si trova sulla cella i**
- affinché le variabili in R descrivano effettivamente la posizione della testina di NT_Γ ad ogni passo della computazione NT_Γ (x), dobbiamo imporre che esse siano coerenti:
  - ad ogni passo della computazione NT_Γ (x) la testina di NT_Γ è posizionata su una (ed esattamente una!) delle celle del suo nastro
  - allora dobbiamo fare in modo che possano essere prese in considerazione solo quelle assegnazioni di valori alle variabili in R tali che, per ogni t compreso fra 0 e p(|x|), ad una e una sola delle variabili R^t_1 , R^t_2 : , ... , $R^t_{p(|x|)}$ sia assegnato il valore **vero**.
- Come prima, a questo scopo introduciamo p(|x|)+1 espressioni: E^1_R , ... , $E^{p(|x|)}_R$

Variabili per la posizione della testina

- E^t_R è l'espressione nelle variabili in R che corrisponde all'affermazione
  al passo t di NT_Γ (x) la testina di NT_Γ è posizionata su una (ed esattamente una!) delle celle del suo nastro

- Allora,

$$E^t_R = [\, R^t_1 \wedge \neg R^t_2 \wedge \neg R^t_3 \wedge \ldots \wedge \neg R^t_{p(|x|)} \,]$$ **la testina di NT_Γ , al passo t di NT_Γ (x), si trova sulla cella 1…**

$$\vee \; [\, R^t_2 \wedge \neg R^t_1 \wedge \neg R^t_3 \wedge \ldots \wedge \neg R^t_{p(|x|)} \,]$$ **… oppure si trova sulla cella 2 …**

$$\vee \; [\, R^t_3 \wedge \neg R^t_1 \wedge \neg R^t_2 \wedge \ldots \wedge \neg R^t_{p(|x|)} \,]$$ **… oppure si trova sulla cella 3 …**

$$\vee \ldots$$

$$\vee \; [\, R^t_{p(|x|)} \wedge \neg R^t_1 \wedge \neg R^t_2 \wedge \ldots \wedge \neg R^t_{p(|x|)-1} \,]$$ **… oppure si trova sulla cella p(|x|)**

- Che assume il valore **vero** se e soltanto se ad <u>esattamente</u> una delle variabili $R^t_1 , R^t_2 , R^t_3 , \ldots , R^t_{p(|x|)}$ è assegnato il valore vero

Variabili per il contenuto delle celle

- Terzo step: descriviamo l'insieme N di variabili booleane che permettono di rappresentare il carattere contenuto in ciascuna cella del nastro di NT_Γ ad *ogni passo* della computazione NT_Γ (x)

- Ricordiamo che NT_Γ utilizza al più p(|x|) celle del nastro che assumiamo essere le celle 1, 2, … , p(|x|)
- e che $L_\Gamma \subseteq \{0,1\}^*$
  - e, perciò, una qualsiasi cella del nastro di NT_Γ , ad un qualunque passo della computazione NT_Γ (x), può contenere 0 oppure 1 oppure □

- L'insieme di variabili N, insieme ad una porzione E_N dell'espressione **E(x)** che stiamo costruendo, servono a descrivere quale simbolo è contenuto in ogni cella del nastro di NT_Γ ad ogni passo della computazione NT_Γ (x): ciò che vogliamo è che
  - **ogni volta che i valori assegnati alle variabili in N fanno assumere ad E_N il valore vero**,
  - **osservando i valori assegnati alle variabili contenute in N, dobbiamo essere in grado di rispondere a domande del tipo "è 1 il simbolo contenuto nella cella 12 di NT_Γ al passo 25 della computazione NT_Γ (x) ?".**

Variabili per il contenuto delle celle

- Terzo step: descriviamo l'insieme N di variabili booleane che permettono di rappresentare il carattere contenuto in ciascuna cella del nastro di NT_Γ ad *ogni passo* della computazione NT_Γ (x)
- Per ogni passo t (con $0 \leq t \leq p(|x|)$), per ogni $i \in \{ 0,1,\ldots,p(|x|) \}$, e per ogni $j \in \{ 0,1, \square \}$, N contiene una variabile booleana N^t_{ij} :

$$\mathbf{N = \{ N^t_{ij} : 0 \leq t \leq p(|x|) \ \wedge \ 1 \leq i \leq p(|x|) \wedge j \in \{ 0,1, \square \} \}}$$

- con il seguente significato: **assegnando a N^t_{ij} il valore vero rappresentiamo il fatto che, al passo t della computazione NT_Γ (x), la cella i del nastro di NT_Γ contiene il simbolo j**
- affinché le variabili in N descrivano effettivamente i contenuti delle celle del nastro di NT_Γ ad ogni passo della computazione NT_Γ (x), dobbiamo imporre che esse siano coerenti:
  - ad ogni passo della computazione NT_Γ (x) *ogni cella* di NT_Γ contiene un simbolo - ed esattamente uno!
  - allora dobbiamo fare in modo che possano essere prese in considerazione solo quelle assegnazioni di valori alle variabili in N tali che, per ogni t compreso fra 0 e p(|x|), e per ogni i compreso fra 1 e p(|x|), ad una e una sola delle variabili N^t_{i0} , N^t_{i1} , $N^t_{i\square}$ sia assegnato il valore **vero**.
- Di nuovo, a questo scopo introduciamo p(|x|)+1 espressioni: $\mathbf{E^0_N , E^1_N , \ldots , E^{p(|x|)}_N}$

Variabili per il contenuto delle celle

- Ma, ora, abbiamo bisogno di un passo intermedio
- Indichiamo con $E^{t\,i}_{N}$ l'espressione nelle variabili in N che corrisponde all'affermazione

  al passo t di NT_Γ (x) la cella i di NT_Γ
  contiene un elemento (ed esattamente uno!) dell'insieme { 0,1, □ }

- $E^{t\,i}_{N} = [\, N^{t}_{i\,0} \wedge \neg N^{t}_{i\,1} \wedge \neg N^{t}_{i\,\square} \,]$ **nella cella i di NT_Γ , al passo t di NT_Γ (x), si trova il simbolo 0…**

  $\vee \; [\, N^{t}_{i\,1} \wedge \neg N^{t}_{i\,0} \wedge \neg N^{t}_{i\,\square} \,]$ **… oppure si trova il simbolo 1…**

  $\vee \; [\, N^{t}_{i\,\square} \wedge \neg N^{t}_{i\,0} \wedge \neg N^{t}_{i\,1} \,]$ **… oppure si trova il simbolo □**

- Che assume il valore **vero** se e soltanto se ad <u>esattamente</u> una delle variabili $N^{t}_{i\,0}$, $N^{t}_{i\,1}$, $N^{t}_{i\,\square}$ è assegnato il valore vero

Variabili per il contenuto delle celle

- Dunque $E^{t\,i}_{N}$ è l'espressione nelle variabili in R che corrisponde all'affermazione
  al passo t di NT_Γ (x) la cella i di NT_Γ contiene un elemento (ed esattamente uno!) dell'insieme { 0,1, □ }
  - $E^{t\,i}_{N}$ assume il valore **vero** se e soltanto se ad esattamente una delle variabili $N^{t}_{i\,0}$, $N^{t}_{i\,1}$, $N^{t}_{i\,\square}$ è assegnato il valore vero

- Infine, per ogni t compreso fra 0 e p(|x|), poniamo

$$E^{t}_{N} \;=\; E^{t\,1}_{N} \wedge E^{t\,2}_{N} \wedge \ldots \wedge E^{t\,p(|x|)}_{N}$$

- E^{t}_{N} assume il valore **vero** se e soltanto se, **per ogni cella i del nastro di NT**, ad esattamente una delle variabili $N^{t}_{i\,0}$, $N^{t}_{i\,1}$, $N^{t}_{i\,\square}$ è assegnato il valore vero
- ossia, **E^{t}_{N} assume il valore vero se e soltanto se, per ogni cella i del nastro di NT_Γ, al passo t di NT_Γ (x) la cella i di NT_Γ contiene un elemento (ed esattamente uno!) dell'insieme { 0,1, □ }**

Ricapitolando…

- Siamo partiti da un generico problema $\Gamma \in NP$ e dal linguaggio $L_\Gamma \subseteq \{0,1\}^*$ contenente una codifica ragionevole delle istanze sì di Γ
- Abbiamo considerato una macchina di Turing non deterministica NT_Γ che accetta, le parole x in L_Γ in tempo $p(|x|)$ – polinomiale in $|x|$
- E abbiamo osservato che l'affermazione "$x \in L_\Gamma$ " è *logicamente equivalente* all'affermazione

  $\gamma(x)$ = **"x (e nient'altro che x) è scritto sul nastro di NT_Γ**
  e la testina di NT_Γ è posizionata sul primo carattere di x
  e NT_Γ è nel suo stato iniziale,
  e esiste una sequenza di al più $p(|x|)$ quintuple di NT_Γ che possono essere eseguite una di seguito all'altra e portano la macchina nello stato q_A"
- cioè **$x \in L_\Gamma$ se e soltanto se $\gamma(x)$ è vera**
- Ci siamo proposti di **descrivere $\gamma(x)$ sotto forma di espressione booleana E(x) che sia soddisfacibile se e soltanto se $\gamma(x)$ è vera**
- e poiché **E(x)** deve descrivere una **computazione di NT_Γ che ha inizio con x scritto sul suo nastro**, ossia, una sequenza di *stati globali* tale che si passa da uno stato globale al successivo mediante l'esecuzione di una quintupla
- abbiamo definito le variabili booleane che ci permettono di descrivere gli stati globali che compongono la computazione NT_Γ (x) (e le condizioni per la loro consistenza!)
- non ci resta che descrivere configurazione iniziale, stati globali e computazione…

Rappresentare un generico stato globale

- Descriviamo, dunque, all'interno di **E(x)** gli stati globali che compongono la computazione $NT_\Gamma(x)$
- Ma lo abbiamo già fatto!
- Le variabili che descrivono uno stato globale SG_t in cui si trova la macchina NT_Γ al passo t di una ***generica*** computazione di p(|x|) passi sono:
  - le variabili $M^t_0, M^t_1, \dots, M^t_k$ per lo stato interno
  - le variabili $R^t_1, R^t_2, \dots, R^t_{p(|x|)}$ per la posizione della testina
  - le variabili $N^t_{10}, N^t_{11}, N^t_{1\square}, \dots, N^t_{p(|x|)\,0}, N^t_{p(|x|)\,1}, N^t_{p(|x|)\,\square}$ che il contenuto delle p(|x|) celle del nastro utilizzate durante la computazione
- e gli stati globali di NT_Γ al passo t di $NT_\Gamma(x)$ sono completamente descritti da tutte e sole le assegnazioni di verità che soddisfano $\mathcal{S}^t = E^t_M \wedge E^t_R \wedge E^t_N$
- infatti, una assegnazione di verità che soddisfa $\mathcal{S}^t$ rappresenta
  - l'unico stato interno in cui si trova NT_Γ al tempo t, l'unica cella del nastro sulla quale è posizionata la testina di NT_Γ al tempo t, e l'unico simbolo in { 0,1, □ } contenuto in ciascuna cella del nastro di NT_Γ al tempo t
- viceversa, dato uno stato globale di NT_Γ al tempo t, è facile derivare da esso una assegnazione di verità che soddisfa $\mathcal{S}^t$

Rappresentare la computazione $NT_\Gamma(x)$

- A questo punto, sappiamo come rappresentare uno stato globale generico e lo stato globale iniziale **della computazione $NT_\Gamma(x)$** mediante una assegnazione di verità alle variabili in M, R e N
- dobbiamo rappresentare allo stesso modo le computazioni **della macchina NT_Γ** che accettano in al più $p(|x|)$ passi

- Ossia: **esiste una sequenza di al più $p(|x|)$ quintuple di NT_Γ che possono essere eseguite una di seguito all'altra e portano la macchina nello stato q_A**
- Possiamo allora dire che **$NT_\Gamma(x)$ è una computazione accettante in $p(|x|)$ passi** se:
  - al passo 0, NT_Γ esegue una quintupla e
  - al passo 1, NT_Γ è nello stato q_A oppure esegue una quintupla e
  - al passo 2, NT_Γ è nello stato q_A oppure esegue una quintupla e
  - …
  - al passo $p(|x|) - 1$, NT_Γ è nello stato q_A oppure esegue una quintupla e
  - al passo $p(|x|)$, NT_Γ è nello stato q_A.

Rappresentare la computazione $NT_\Gamma(x)$

- Dobbiamo mostrare come esprimere
  - **al passo t, NT_Γ è nello stato q_A oppure esegue una quintupla**
- Sia $\langle q_{i1}, s1, s2, q_{i2}, m \rangle$ una quintupla di NT_Γ
  - con m = -1 se la testina si muove a sinistra, m = 0 se rimane ferma, m = +1 se si muove a destra
- L'affermazione "la quintupla $\langle q_{i1}, s1, s2, q_{i2}, m \rangle$ è eseguita al passo t **mentre la testina è posizionata sulla cella u**" è equivalente all'espressione

$$\mathcal{G}^t(u, \langle q_{i1}, s1, s2, q_{i2}, m \rangle) = M^t_{i1} \wedge R^t_u \wedge N^t_{u\,s1} \wedge N^{t+1}_{u\,s2} \wedge M^{t+1}_{i2} \wedge R^{t+1}_{u+m}$$

- ossia: "al passo t la macchina è nello stato q_{i1}, la testina è posizionata sulla cella u e legge il simbolo s1, e al passo t+1 la macchina è nello stato q_{i2}, nella cella u è stato scritto s2 e la testina è stata spostata sulla cella u + m"
- L'espressione $\mathcal{G}^t(u, \langle q_{i1}, s1, s2, q_{i2}, m \rangle)$ significa: "al passo t, la testina è posizionata sulla cella u e viene eseguita la quintupla $\langle q_{i1}, s1, s2, q_{i2}, m \rangle$"
- *ma, al passo t, la testina potrebbe essere posizionata su qualunque cella…*

Rappresentare la computazione $NT_\Gamma(x)$

- L'affermazione "la quintupla $\langle q_{i1}, s1, s2, q_{i2}, m \rangle$ è eseguita al passo t **mentre la testina è posizionata sulla cella u**" è equivalente all'espressione

$$\mathcal{G}^t(u, \langle q_{i1}, s1, s2, q_{i2}, m \rangle) = M^t_{i1} \wedge R^t_u \wedge N^t_{u\,s1} \wedge N^{t+1}_{u\,s2} \wedge M^{t+1}_{i2} \wedge R^{t+1}_{u+m}$$

- L'espressione $\mathcal{G}^t(u, \langle q_1, s1, s2, q_2, m \rangle)$ significa: "al passo t, la testina è posizionata sulla cella u e viene eseguita la quintupla $\langle q_1, s1, s2, q_2, m \rangle$", *ma, al passo t, la testina potrebbe essere posizionata su qualunque cella…*
- Allora, per esprimere che **"su qualunque cella sia posizionata la testina, la quintupla $\langle q_{i1}, s1, s2, q_{i2}, m \rangle$ è eseguita al passo t"** scriviamo l'espressione

$$\mathcal{G}^t(\langle q_{i1}, s1, s2, q_{i2}, m \rangle) = \mathcal{G}^t(1, \langle q_{i1}, s1, s2, q_{i2}, m \rangle) \vee \mathcal{G}^t(2, \langle q_{i1}, s1, s2, q_{i2}, m \rangle) \vee \ldots \vee \mathcal{G}^t(p(|x|), \langle q_{i1}, s1, s2, q_{i2}, m \rangle)$$

- che significa "al passo t la macchina è nello stato q_{i1}, la testina è posizionata su una *qualsiasi* cella u (con $1 \leq u \leq p(|x|)$) e legge il simbolo s1, e al passo t+1 la macchina è nello stato q_{i2}, nella cella u è stato scritto s2 e la testina è stata spostata sulla cella u + m"

Rappresentare la computazione $NT_\Gamma(x)$

- Per esprimere che "**su qualunque cella sia posizionata la testina, la quintupla $\langle q_{i1}, s1, s2, q_{i2}, m \rangle$ è eseguita al passo t**" scriviamo l'espressione

  $\mathcal{G}^t(\langle q_{i1}, s1, s2, q_{i2}, m \rangle) = \mathcal{G}^t(1, \langle q_{i1}, s1, s2, q_{i2}, m \rangle) \vee \mathcal{G}^t(2, \langle q_{i1}, s1, s2, q_{i2}, m \rangle) \vee \ldots$
  $\vee \mathcal{G}^t(p(|x|), \langle q_{i1}, s1, s2, q_{i2}, m \rangle)$

- Se l'insieme delle quintuple di NT_Γ è

  $\{\langle q_{11}, s11, s12, q_{12}, m1 \rangle, \langle q_{21}, s21, s22, q_{22}, m2 \rangle, \ldots, \langle q_{h1}, sh1, sh2, q_{h2}, mh \rangle\}$

- allora per esprimere l'affermazione

  " **al passo t viene eseguita una quintupla di NT_Γ** "

  scriviamo l'espressione

  $\mathcal{G}^t = \mathcal{G}^t(\langle q_{11}, s11, s12, q_{12}, m1 \rangle) \vee \mathcal{G}^t(\langle q_{21}, s21, s22, q_{22}, m2 \rangle) \vee \ldots$
  $\ldots \vee \mathcal{G}^t(\langle q_{h1}, sh1, sh2, q_{h2}, mh \rangle)$

Rappresentare la computazione $NT_\Gamma(x)$

- Per esprimere l'affermazione

  "**al passo t viene eseguita una quintupla di NT_Γ** "

  scriviamo l'espressione

$$\mathcal{G}^t = \mathcal{G}^t(\langle q_{11}, s11, s12, q_{12}, m1 \rangle) \vee \mathcal{G}^t(\langle q_{21}, s21, s22, q_{22}, m2 \rangle) \vee \ldots$$
$$\ldots \vee \mathcal{G}^t(\langle q_{h1}, sh1, sh2, q_{h2}, m2 \rangle)$$

- Allora, per esprimere l'affermazione
  - **al passo t, NT_Γ è nello stato q_A oppure esegue una quintupla**
    - ricordando che $q_A = q_1$
- scriviamo: $M^t_1 \vee \mathcal{G}^t$

Rappresentare la configurazione iniziale di $NT_\Gamma(x)$

- Descriviamo, ora, una espressione che andrà a comporre **E(x)** che permette di descrivere lo stato globale iniziale di $NT_\Gamma(x)$, ossia, la prima parte di $\gamma(x)$:

  " x (e nient'altro che x) è scritto sul nastro di NT_Γ e la testina di NT_Γ è posizionata sul primo carattere di x e NT_Γ è nel suo stato iniziale "

- **NT_Γ è nel suo stato interno iniziale**
  - Facile: è sufficiente imporre che ad M^0_0 debba essere assegnato il valore vero
- $\wedge$ **la testina di NT_Γ è posizionata sul primo carattere di x**
  - Facile: è sufficiente imporre che ad R^0_1 debba essere assegnato il valore vero
- $\wedge$ **x (e nient'altro che x) è scritto sul nastro di NT_Γ**
  - sia x = x1 x2 ... xn (ovviamente, xi ∈ { 0,1, □ } per i = 1, … , n)
  - allora, è sufficiente imporre che

    per i = 1, … , n, ad $N^0_{i\,xi}$ debba essere assegnato il valore **vero**

    per i = n+1, … , p(n) ad $N^0_{i\,□}$ debba essere assegnato il valore **vero**

Rappresentare la configurazione iniziale di $NT_\Gamma(x)$

- Descriviamo, ora, una espressione che andrà a comporre **E(x)** che permette di descrivere lo stato globale iniziale di $NT_\Gamma(x)$, ossia, la prima parte di $\gamma(x)$:

  " x (e nient'altro che x) è scritto sul nastro di NT_Γ e la testina di NT_Γ è posizionata sul primo carattere di x e NT_Γ è nel suo stato iniziale "

- Quindi, lo stato globale iniziale di $NT_\Gamma(x)$ è completamente descritto da una assegnazione di verità che soddisfa

$$\mathcal{H} = M^0_0 \wedge R^0_1 \wedge N^0_{1\,x1} \wedge N^0_{2\,x2} \wedge \ldots \wedge N^0_{n\,xn} \wedge N^0_{n+1\,\square} \wedge N^0_{n+2\,\square} \wedge \ldots \wedge N^0_{p(n)\,\square}$$

- **esempio: se x = 1001,**

$$\mathcal{H} = M^0_0 \wedge R^0_1 \wedge N^0_{1\,1} \wedge N^0_{2\,0} \wedge N^0_{3\,0} \wedge N^0_{4\,1} \wedge N^0_{5\,\square} \ldots \wedge N^0_{p(4)\,\square}$$

Finalmente, E(x)

- Possiamo, infine, mettere insieme tutti i mattoncini che abbiamo sin qui costruito
  - il predicato $\mathcal{H}$ - che assume valore vero se e soltanto se alle variabili in M, R, N vengono assegnati valori di verità corrispondenti alla presenza di x (e nient'altro) sul nastro, alla testina posizionata sulla cella 1, e alla macchina che si trova nello stato interno iniziale
  - per ogni t, il predicato $\mathcal{S}^t$ - che assume valore vero se e solo se alle variabili che descrivono lo stato globale al passo t vengono assegnati valori di verità consistenti (la macchina è in uno ed un solo stato ecc.)
  - per ogni t, il predicato $\mathcal{G}^t$ - che assume valore vero se e solo se alle variabili vengono assegnati valori di verità che descrivono l'esecuzione di una quintupla al passo t
  - ricordando che M^t_1 è la variabile che descrive se al passo t NT_Γ è nello stato q_A
  - e ricordando che **$NT_\Gamma(x)$ è una computazione accettante** se:
    - al passo 0, NT_Γ esegue una quintupla,
    - e al passo 1, NT_Γ è nello stato q_A oppure esegue una quintupla e …
    - al passo p(|x|) - 1, NT_Γ è nello stato q_A oppure esegue una quintupla, e al passo p(|x|), NT_Γ è nello stato q_A.
- per ottenere l'espressione E(x)

$$E(x) = \mathcal{H} \wedge \mathcal{S}^0 \wedge (M^0_1 \vee \mathcal{G}^0) \wedge \mathcal{S}^1 \wedge (M^1_1 \vee \mathcal{G}^1) \wedge \mathcal{S}^2 \wedge \ldots \ldots \wedge \mathcal{S}^{p(|x|)-1} \wedge (M^{p(|x|)-1}_1 \vee \mathcal{G}^{p(|x|)-1}) \wedge \mathcal{S}^{p(|x|)} \wedge M^{p(|x|)}_1$$

$x \in L_\Gamma$ se e solo se E(x) è soddisfacibile

- A partire da NT_Γ e da $x \in \{0,1\}^*$ abbiamo ottenuto

$$E(x) = \mathcal{H} \wedge \mathcal{S}^0 \wedge (M^0{}_1 \vee \mathcal{G}^0) \wedge \mathcal{S}^1 \wedge (M^1{}_1 \vee \mathcal{G}^1) \wedge \ldots$$
$$\ldots \wedge \mathcal{S}^{p(|x|)-1} \wedge (M^{p(|x|)-1}{}_1 \vee \mathcal{G}^{p(|x|)-1}) \wedge \mathcal{S}^{p(|x|)} \wedge M^{p(|x|)}{}_1$$

- Ricordiamo che stiamo mostrando che $\mathbf{L_\Gamma}$ è riducibile polinomialmente a SAT (senza curarci della forma congiuntiva normale)
  - e che, quindi, abbiamo mostrato come trasformare x in E(x)
  - la macchina NT_Γ gioca il ruolo di costante: non è l'istanza!
- Dobbiamo, a questo punto, dimostrare che

$x \in L_\Gamma$ se e soltanto se
esiste una assegnazione di verità per le variabili in M, R, N che soddisfa E(x)

se x ∈ L_Γ allora E(x) è soddisfacibile

- A partire da NT_Γ e da x ∈ {0,1}* abbiamo ottenuto

$$E(x) = \mathcal{H} \wedge \mathcal{S}^0 \wedge (M^0{}_1 \vee \mathcal{G}^0) \wedge \mathcal{S}^1 \wedge (M^1{}_1 \vee \mathcal{G}^1) \wedge \ldots$$
$$\ldots \wedge \mathcal{S}^{p(|x|)-1} \wedge (M^{p(|x|)-1}{}_1 \vee \mathcal{G}^{p(|x|)-1}) \wedge \mathcal{S}^{p(|x|)} \wedge M^{p(|x|)}{}_1$$

- se $x \in L_\Gamma$, allora esiste una computazione di $NT_\Gamma(x)$ che termina in q_A in al più $p(|x|)$ passi
- cioè, esistono
  - una sequenza di stati globali $SG_0, SG_1, \ldots, SG_u$, con $u \leq p(|x|)$
  - e una sequenza di u quintuple, dove la quintupla t è $\langle q_{t1}, s_{t1}, s_{t2}, q_{t2}, m_t \rangle$, con $0 \leq t \leq u-1$
- tali che
  - SG_0 è lo stato globale in cui la macchina è nello stato q_0, la testina è posizionata sulla cella 1, le prime $|x|$ celle contengono i bit di x, e le rimanenti $p(|x|)-x$ celle contengono □
  - per t = 0, … , u-1, lo stato interno di SG_t è q_{t1} e il simbolo letto dalla testina è s_{t1}, e SG_{t+1} è lo stato globale corrispondente all'esecuzione della t-esima quintupla della sequenza a partire da SG_t
  - lo stato interno di SG_u è q_A

se x ∈ L_Γ allora E(x) è soddisfacibile

- A partire
  - dalla sequenza di stati globali $SG_0, SG_1, \dots, Sg_u$, con $u \le p(|x|)$
  - e dalla sequenza di u quintuple, dove la quintupla i è $\langle q_{i1}, s_{i1}, s_{i2}, q_{12}, m_i \rangle$, con $0 \le i \le u-1$
- costruiamo una assegnazione di verità **a** che soddisfa

$$E(x) = \mathcal{H} \wedge \mathcal{S}^0 \wedge (M^0{}_1 \vee \mathcal{G}^0) \wedge \mathcal{S}^1 \wedge (M^1{}_1 \vee \mathcal{G}^1) \wedge \dots \wedge \mathcal{S}^u \wedge (M^u{}_1 \vee \mathcal{G}^u) \wedge \dots \wedge \mathcal{S}^{p(|x|)-1} \wedge (M^{p(|x|)-1}{}_1 \vee \mathcal{G}^{p(|x|)-1}) \wedge \mathcal{S}^{p(|x|)} \wedge M^{p(|x|)}{}_1$$

- **1: usiamo SG_0.** Poniamo
  - **a**$(M^0{}_0)$ = **a**$(R^0{}_1)$ = **vero**
  - per j = 1, … , |x| , se il bit j di x è 0 poniamo **a**$(N^0{}_{j0})$ = **vero** altrimenti poniamo **a**$(N^0{}_{j1})$ = **vero**
  - per j = |x| , … ,p(|x|), poniamo **a**$(N^0{}_{j\ \square})$ = **vero**
  - **a** assegna falso a tutte le altre variabili in M^0, R^0, N^0
  - pertanto, **a** soddisfa $\mathcal{H} \wedge \mathcal{S}^0$

se x ∈ L_Γ allora E(x) è soddisfacibile

- **2: usiamo SG_1, SG_2, … , SG_u**. Definiamo $\mathbf{a}(M^t_i)$, $\mathbf{a}(R^t_i)$, $\mathbf{a}(N^t_{ji})$, usando SG^t analogamente a quanto abbiamo fatto al punto 1.
- Questo garantisce che, per ogni t=1, … , u,
  - esiste uno ed un solo i tale che $\mathbf{a}(M^t_i)$ = **vero**
  - esiste uno ed un solo i tale che $\mathbf{a}(R^t_i)$ = **vero**
  - per ogni j = 1, … ,p(|x|), esiste uno ed un solo i tale che $\mathbf{a}(N^t_{j\,i})$ = **vero**
- e quindi che, per ogni t=1, … , u, **a** soddisfa $\mathcal{S}^t$
- **3: usiamo le quintuple**. Poiché per ogni t = 0, … , u-1 può essere eseguita la quintupla t della sequenza, allora, per ogni t = 0, … , u-1, **a** soddisfa $\mathcal{G}^t$
- **4: lo stato interno di SG_u è q_A** . Allora, $\mathbf{a}(M^u_1)$=**vero**; perciò, benché al passo u non venga eseguita alcuna quintupla, **a** soddisfa $(\mathbf{M^u_1} \vee \mathcal{G}^u)$
- **5: t > u**. Per ogni i = 0, … , k
  poniamo $\mathbf{a}(M^t_i)= \mathbf{a}(M^u_i)$,
  per ogni j = 1, … , h poniamo $\mathbf{a}(R^t_i)= \mathbf{a}(R^u_i)$, $\mathbf{a}(N^t_{i0})= \mathbf{a}(N^u_{i0})$, $\mathbf{a}(N^t_{i1})= \mathbf{a}(N^u_{i1})$,
  $\mathbf{a}(N^t_{i\,□})= \mathbf{a}(N^u_{i\,□})$,
  ed è facile verificare che **a** soddisfa $\mathcal{S}^t$ e $\mathbf{M^t_1}$
- **Questo dimostra che a soddisfa E(x)**

se E(x) è soddisfacibile allora x ∈ L_Γ

- A partire da NT_Γ e da x ∈ {0,1}* abbiamo ottenuto

$$E(x) = \mathcal{H} \wedge \mathcal{S}^0 \wedge (M^0{}_1 \vee \mathcal{G}^0) \wedge \mathcal{S}^1 \wedge (M^1{}_1 \vee \mathcal{G}^1) \wedge \ldots \ldots \wedge \mathcal{S}^{p(|x|)-1} \wedge (M^{p(|x|)-1}{}_1 \vee \mathcal{G}^{p(|x|)-1}) \wedge \mathcal{S}^{p(|x|)} \wedge M^{p(|x|)}{}_1$$

- supponiamo, ora, che esista una assegnazione di verità **a** alle variabili in M, R, N che soddisfa **E(x)**
- ossia, **a** soddisfa $\mathcal{H}$, $\mathcal{S}^{p(|x|)}$ e $M^{p(|x|)}{}_1$
- e, inoltre, per ogni t = 0, 1, … , p(|x|)-1, **a** soddisfa $\mathcal{S}^t$ e $(M^t{}_1 \vee \mathcal{G}^t)$
- Poiché, ricordiamo, ogni assegnazione di verità che soddisfa $\mathcal{S}^t = E^t{}_M \wedge E^t{}_R \wedge E^t{}^1{}_N \wedge \ldots \wedge E^t{}^{p(|x|)}{}_N$ descrive uno stato globale di NT_Γ
  - specificando che, per ogni ogni t = 0, 1, … , p(|x|)-1, NT_Γ è in uno (ed un solo) stato interno, con la testina posizionata su una (e una sola) cella che contiene un (ed un solo) elemento in {0, 1, □}
- allora **a** descrive una sequenza $SG^1, \ldots, SG^{p(|x|)-1}$ di stati globali di NT_Γ
  - dove, per ogni t = 0, 1, … , p(|x|)-1, SG^t è lo stato descritto da **a**($\mathcal{S}^t$)

se E(x) è soddisfacibile allora x ∈ L_Γ

- A partire da NT_Γ e da x ∈ {0,1}* abbiamo ottenuto

$$E(x) = \mathcal{H} \wedge \mathcal{S}^0 \wedge (M^0{}_1 \vee \mathcal{G}^0) \wedge \mathcal{S}^1 \wedge (M^1{}_1 \vee \mathcal{G}^1) \wedge \ldots \ldots \wedge \mathcal{S}^{p(|x|)-1} \wedge (M^{p(|x|)-1}{}_1 \vee \mathcal{G}^{p(|x|)-1}) \wedge \mathcal{S}^{p(|x|)} \wedge M^{p(|x|)}{}_1$$

- supponiamo, ora, che esista una assegnazione di verità **a** che soddisfa **E(x)**
- allora **a** descrive una sequenza $SG^1, \ldots, SG^{p(|x|)-1}$ di stati globali di NT_Γ
- poiché , **a** soddisfa $\mathcal{H}$, **a**($\mathcal{S}^0$) descrive lo stato globale in cui x (e solo x) è scritto sul nastro, NT_Γ è nello stato q_0, e la testina è posizionata sul carattere più a sinistra dell'input
- inoltre, per ogni t = 0, 1, … , p(|x|)-1, **a** soddisfa $(M^t{}_1 \vee \mathcal{G}^t)$
- allora, per ogni t = 0, 1, … , p(|x|)-1,
  - **o** viene eseguita una quintupla (se **a**($\mathcal{G}^t$) = **vero**) che fa passare da SG^t a SG^{t+1}
  - **oppure** NT_Γ è in q_A (se **a**($M^t{}_1$) = **vero**)
  - osserviamo che non può accadere **a**($\mathcal{G}^t$) = **vero** e **a**($M^t{}_1$) = **vero** in quanto esiste uno e un solo i tale che **a**($M^t{}_i$) = **vero** e non esistono quintuple che partono da $q_1=q_A$

se E(x) è soddisfacibile allora $x \in L_\Gamma$

- supponiamo, ora, che esista una assegnazione di verità **a** che soddisfa **E(x)**
- **a** corrisponde ad una sequenza $SG^1, \ldots, SG^{p(|x|)-1}$ di stati globali di NT_Γ
  - dove, per ogni t = 0, 1, … , p(|x|)-1, SG^t è lo stato globale corrispondente a $a(\mathcal{S}^t)$e $\mathcal{S}^0$ corrisponde allo stato globale iniziale di $NT_\Gamma(x)$ aaa
- inoltre, per ogni t = 0, 1, … , p(|x|)-1, se $a(\mathcal{G}^t)$ = **vero** allora viene eseguita una quintupla che fa passare da SG^t a SG^{t+1},
- D'altra parte, poiché a soddisfa E(x), allora deve essere $a(M^{p(|x|)}_1)$ = **vero**
  - e questo significa che esiste un indice h tale che $a(M^h_1)$ = **vero**
  - e che (come è facile verificare), per ogni $t \geq h$, $a(M^t_1)$ = **vero**
- sia u {0, 1, … , p(|x|) } il primo intero tale che $a(M^t_1)$ = **vero**
  - ossia, per ogni t = 0, 1, ... , u-1, $a(\mathcal{G}^t)$ = **vero**
  - e, quindi, per ogni t = 0, 1, ... , u-1, viene eseguita una quintupla che fa passare da SG^t a SG^{t+1}
- allora $\langle SG_0, SG_1, \ldots, SG^u \rangle$ è una computazione accettante di $NT_\Gamma(x)$
- e, quindi, $x \in L_\Gamma$

Quanto costa calcolare E(x)?

- A partire da NT_Γ e da $x \in \{0,1\}^*$ abbiamo ottenuto

$$E(x) = \mathcal{H} \wedge \mathcal{S}^1 \wedge (M^1{}_1 \vee \mathcal{G}^1) \wedge \mathcal{S}^2 \wedge (M^2{}_1 \vee \mathcal{G}^2) \wedge \ldots$$
$$\ldots \wedge \mathcal{S}^{p(|x|)-1} \wedge (M^{p(|x|)-1}{}_1 \vee \mathcal{G}^{p(|x|)-1}) \wedge \mathcal{S}^{p(|x|)} \wedge M^{p(|x|)}{}_1$$

- Ricordiamo che stiamo mostrando che $\mathbf{L_\Gamma}$ è riducibile polinomialmente a SAT (senza curarci della forma congiuntiva normale)
  - e che, quindi, abbiamo mostrato come trasformare x in E(x)
  - la macchina NT_Γ gioca il ruolo di costante: non è l'istanza!
- Ma quanto tempo occorre a calcolare E(x) a partire da NT_Γ e da $x \in \{0,1\}^*$?
- È facile verificare che, per ogni t = 0, 1, … , p(|x|), calcolare $\mathcal{S}^t$ e $\mathcal{G}^t$ richiede O(p(|x|)) passi
  - e, quindi, calcolarli tutti richiede $O([p(|x|)]^2)$ passi
- Calcolare $\mathcal{H}$ richiede un numero di passi proporzionale a p(|x|)
- In conclusione, calcoliamo E(x) in $O([p(|x|)]^2)$ passi

Il Teorema di Cook-Levin

- Abbiamo considerato un qualunque $L_\Gamma \in NP$, e da $x \in \{0,1\}^*$ abbiamo costruito

$$E(x) = \mathcal{H} \wedge \mathcal{S}^1 \wedge (M^1{}_1 \vee \mathcal{G}^1) \wedge \mathcal{S}^2 \wedge (M^2{}_1 \vee \mathcal{G}^2) \wedge \ldots$$
$$\ldots \wedge \mathcal{S}^{p(|x|)-1} \wedge (M^{p(|x|)-1}{}_1 \vee \mathcal{G}^{p(|x|)-1}) \wedge \mathcal{S}^{p(|x|)} \wedge M^{p(|x|)}{}_1$$

- e abbiamo dimostrato che E(x) è calcolabile in $O([p(|x|)]^2)$ passi
- e abbiamo dimostrato che $x \in L_\Gamma$ se e soltanto se E(x) è soddisfacibile
- Come abbiamo osservato, E(x) non è in forma congiuntiva normale
- Tuttavia, è semplice trasformare E(x) in forma congiuntiva normale
  - è sufficiente applicare le leggi distributive di $\wedge$ e $\vee$, separatamente, a ciascun $\mathcal{S}^t$ e $\mathcal{G}^t$
  - ma non lo facciamo!
- e questo richiede $O(p(|x|))$ passi
  - anche se non lo dimostriamo
- E, poiché l'algoritmo non deterministico che abbiamo utilizzato per mostrare che $3SAT \in NP$ prova anche che $SAT \in NP$, questo completa la dimostrazione del Teorema di Cook-Levin:

SAT è NP-completo