NEIL IMMERMAN (HOME PAGE)

- FINITE MODEL THEORY (EBBINGHAUS-FLUM)
- DESCRIPTIVE COMPLEXITY (IMMERMAN)

$FO \subset NL \subset P \subset NP \subset PSPACE \subset EXPTIME$

↑ CLASSI DI PROBLEMI

$NL = NSPACE[\log n]$

FISSATO UN ALFABETO Σ (INSIEME FINITO DI SIMBOLI),

$\Sigma^* = \bigcup_n \Sigma^n$ (STRINGHE FINITE DI SIMBOLI).

UN <u>PROBLEMA</u> E' UN SOTTOINSIEME $L \subset \Sigma^*$ (DETTO ANCHE "LINGUAGGIO").

<u>ES.</u>: SIA 3-COL = CLASSE DEI GRAFI 3-COLORABILI.
OGNI GRAFO E' CODIFICABILE IN UNA STRINGA DI 0 E 1.

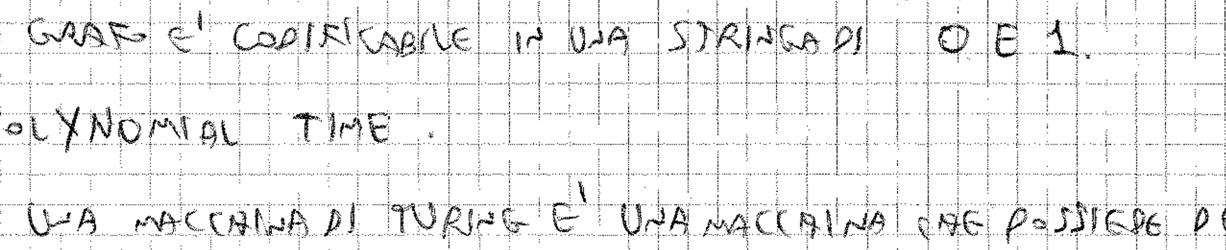
P = POLYNOMIAL TIME.

<u>DEF.</u>: UNA MACCHINA DI TURING E' UNA MACCHINA CHE POSSIEDE DEGLI STATI INTERNI, UN INDICATORE DI STATO, UN NASTRO INFINITO CHE CONTIENE DEI SIMBOLI DI UN ALFABETO $\Sigma' \supset \{0,1\}$, UN PUNTATORE CHE INDICA UNA POSIZIONE SUL NASTRO.

LA FUNZIONE DI TRANSIZIONE $\delta : \{STATI\} \times \Sigma \to \{STATI\} \times \Sigma \times \{SPOSTAMENTI\}$

$\left(SPOSTAMENTI = \{\overset{SX}{-1}, \overset{STOP}{0}, \overset{DX}{1}\}\right)$.

<u>ES.</u>: ADDIZIONE DI DUE NUMERI.



VOGLIAMO CHE SI FERMI SU:

REALIZZO LA MIA δ:

$(q_0, \overset{\text{LEGGE 1}}{1}) \mapsto (q_0, \overset{\text{SCRIVE}}{1}, \overset{\text{DESTRA}}{D})$

$(q_0, 0) \mapsto (\underset{\text{CAMBIA STATO}}{q_1}, 1, D)$

NEIL IMMERMAN (HOME PAGE)

- FINITE MODEL THEORY (EBBINGHAUS-FLUM)
- DESCRIPTIVE COMPLEXITY (IMMERMAN)

$FO \subset NL \subset P \subset NP \subset PSPACE \subset EXPTIME$

FO incluso L incluso NL

↑ CLASSI DI PROBLEMI

$NL = NSPACE[\log n]$

FISSATO UN ALFABETO Σ (INSIEME FINITO DI SIMBOLI),

$\Sigma^* = \bigcup_n \Sigma^n$ (STRINGHE FINITE DI SIMBOLI).

UN PROBLEMA E' UN SOTTOINSIEME $L \subset \Sigma^*$ (DETTO ANCHE "LINGUAGGIO").

ES.: SIA 3-COL = CLASSE DEI GRAFI 3-COLORABILI.
OGNI GRAFO E' CODIFICABILE IN UNA STRINGA DI 0 E 1.

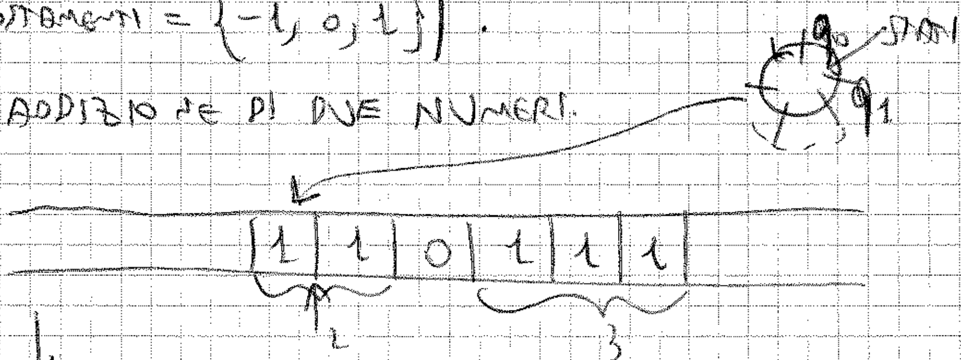
P = POLYNOMIAL TIME.

DEF.: UNA MACCHINA DI TURING E' UNA MACCHINA CHE POSSIEDE DEGLI STATI INTERNI, UN INDICATORE DI STATO, UN NASTRO INFINITO CHE CONTIENE DEI SIMBOLI DI UN ALFABETO $\Sigma' \supset \{0,1\}$, UN PUNTATORE CHE INDICA UNA POSIZIONE SUL NASTRO.

LA FUNZIONE DI TRANSIZIONE $\delta: \{STATI\} \times \Sigma \to \{STATI\} \times \Sigma \times \{SPOSTAMENTI\}$

(SPOSTAMENTI $= \{\overset{SX}{-1}, \overset{STOP}{0}, \overset{DX}{1}\}$).

ES.: ADDIZIONE DI DUE NUMERI.



2+3

VOGLIAMO CHE SI FERMI SU:

REALIZZO LA MIA δ: $(q_0, \overset{\text{LEGGE 1}}{1}) \longmapsto (q_0, \overset{\text{SCRIVE}}{1}, \overset{\text{DESTRA}}{D})$

$(q_0, 0) \longmapsto (q_1, 1, D)$

CAMBIA STATO

$(q_1, 1) \rightarrow (q_1, 1, D)$ (SCRIVI 1, VAI A DESTRA)

$(q_1, \#) \rightarrow (q_2, \#, S)$ (SCRIVI VUOTO, VAI A SINISTRA)

$\uparrow$ BLANK

$(q_2, 1) \rightarrow (q_3, \#, S)$

Definizioni di P ed NP

$(q_3, 1) \rightarrow (q_3, 1, S)$

$(q_3, \#) \rightarrow (q_4, \#, D)$

ABBIAMO DEFINITO δ_2. QUANTO TEMPO CI METTE?
PER FARE $n+m$, CI METTE $2(n+m+1)$ QUINDI UN TEMPO LINEARE (POLINOMIALE):

TIME $[f(n)]$ = SU INPUT DI LUNGA n SI FERMANO MENO DI $f(n)$.

DEF: $P = \bigcup_{p \text{ polinomio}} \text{TIME}(p(n))$

$\text{EXP} = \bigcup_{p \text{ polinomio}} \text{TIME}(2^{p(n)})$

SI SA CHE $P \neq EXP$. $P \subset NP \subset EXP$

$L \subset \Sigma^*$; $L \in P \Leftrightarrow \exists p$ polinomio $\exists$ MDT δ $\forall$ input $\sigma \in \Sigma^*$

$\delta(\sigma)\downarrow_{\leq p(|\sigma|)} = 1$ se $\sigma \in L$, $= 0$ se $\sigma \notin L$.

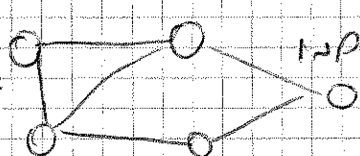
$L \in NP \Leftrightarrow \exists L' \in P$, $\exists p$ polinomio t.c. $\forall \sigma \in \Sigma^*$, $\sigma \in L \Leftrightarrow$
$\Leftrightarrow \exists \tau \in \Sigma^*$, $|\tau| \leq p(|\sigma|)$ t.c. $(\sigma, \tau) \in L'$.

τ È UNA DIMOSTRAZIONE CORRETTA DEL FATTO CHE $\sigma \in L$

~~ESEMPI~~

SI VEDE CHE $NP \subset EXP$:

ES: 3-COL

$\sigma = (G,V) =$  INPUT $\xrightarrow{\delta}$ {SI, NO}

UNA 3-COL. È τ: {VERTICI} $\rightarrow$ {G, V, B} T.C. SIA AMMISSIBILE.

ESISTONO 3^n 3-COLORAZIONI AMMISSIBILI O MENO (n = NUM. VERTICI)

$|\tau| \leq p(|G, V|)$, DEVO DIMOSTRARE CHE τ È CORRETTA, OSSIA
GRAFO VERTICI
τ COLORA CORRETTAMENTE σ

E COLORAZIONI SONO ESPONENZIALI MA UNA VOLTA FISSATA UNA DI QUESTE VERIFICARE CHE È CORRETTA È POLINOMIALE. QUINDI **3-COL $\in$ NP.**

UN ALTRO PROBLEMA IN NP È <u>SAT</u>:

SIA φ UNA FORMULA PROPOSIZIONALE; ES: $\varphi = (A \vee B \vee \neg C) \wedge (\neg A \vee \neg B \vee C)$

NE COSTRUISCO LA TABELLA DI VERITÀ; È LUNGA $2^{|VARIABILI|}$

$SAT = \{\varphi : \exists$ ALMENO UN 1 IN TABELLA$\}$, OPPURE

$\{\varphi : \exists f : \{VARIABILI\} \to \{0,1\}$ T.C. $\varphi^f = 1\}$

(SAT = SATISFIABLE). ANCHE QUI, ENUMERARE TUTTI I CASI È ESPONENZIALE, MA LA VERIFICA DELLA CORRETTEZZA DI UNA ASSEGNAZIONE È POLINOMIALE.

3-COL E SAT SONO NP-COMPLETI! SE FOSSERO IN P, AVREMMO DIMOSTRATO P = NP.

<u>ES</u>: DATO UN NUMERO NATURALE, STABILIRNE LA PRIMALITÀ.

$n \to \boxed{\text{MdT}}$ → (PRIMO) SÌ / NO

CON L'ALGORITMO BANALE, CI METTO UN TEMPO $\log_B n$ (B = BASE). SI DIMOSTRA CHE È POLINOMIALE.

<u>TEOR.</u> **DI FAGIN: NP = SO-$\exists$** con qualche proviso

INOLTRE: **P = FO(LFP)** ; **EXPTIME = SO(LFP)**

SIA L LINGUAGGIO DEL PRIMO ORDINE; ES.: $L = \{R\}$, R BINARIO;

~~$L = \{+, \cdot, 0, 1\}$~~ ; $L = \{\in\}$;

SENZA FUNZIONI

UNA L-STRUTTURA È UNA ~~TRIPLA~~ COPPIA (M, R^M), $R^M \subset M^2$.

ES φ FO: $\forall x \exists y\, R(x,y)$

$L = \{R\}$ (REL. BINARIA) ; φ T.C.

MOD φ = I GRAFI IN CUI OGNI VERTICE È CONNESSO AD ESATTAMENTE ALTRI DUE.

$\forall x \exists y,z\, ((x \neq y \wedge z \neq x \wedge y \neq z) \wedge R(x,y) \wedge R(x,z) \wedge$

$\forall u (R(u,x) \to u = y \vee u = z)) \wedge (\forall x,y\, R(x,y) \to R(y,x)) \wedge (\forall x\, \neg R(x,x))$

NB: LA PROPRIETÀ DI ESSERE CONNESSI NON È ESPRIMIBILE AL PRIMO ORDINE. DEVO SALIRE A FO(TC).

SE HO UNA L-STRUTTURA (M, R^M), φ È DEL PRIMO ORDINE SE COMPAIONO SOLO $\forall x, \exists x, \wedge, \vee, \neg, \rightarrow$.

AL 2° ORDINE, POSSO QUANTIFICARE SU RELAZIONI: $\exists Q \, (Q \subset M)$

ES: $G = (V, E^G)$ STRUTTURA, VOGLIO ESPRIMERE IL FATTO CHE IL GRAFO È 3-COLORABILE. VEDI FOTOCOPIE.

$(R^1: R \subset V \;;\; R^2: R \subset E)$

LA FRASE DEL 2° ORDINE È: $\exists R^1 Y^1 B^1 \ldots$

$MOD(\varphi) = \{3\text{-}COL\}$.

SO-$\exists$: I QUANTIFICATORI DEL 2° ORDINE SONO TUTTI ESISTENZIALI E ALL'INIZIO DELLA FORMULA. PER FAGIN QUESTO È NP.

GIOCHI DI ERENFEUCHT-FRAÏSSÉ

L = LINGUAGGIO DEL PRIMO ORDINE (SIMBOLI COSTANTI, $\exists$, $\forall$, E RELAZIONI)

A, B = L-STRUTTURE

Giochi di EF

$a_1, \dots, a_l \in A$, $b_1, \dots, b_l \in B$.

COSA VUOL DIRE CHE $A, a_1, \dots, a_l \sim_m B, b_1, \dots, b_l$?

PER $l=0$, CIÒ SI RIDUCE A: $A \sim_m B$.

ES.: DUE GRAFI.

SI FA UN GIOCO FRA DUE GIOCATORI ($\forall$, $\exists$) CHE DURA m MOSSE.

$\forall$ SCEGLIE UN ELEMENTO DA A O DA B

$\exists$ SCEGLIE UN ELEMENTO DALL'ALTRA

AD OGNI MOSSA, $\exists$ DEVE "COPIARE" $\forall$ (OSSIA, SCEGLIERE ELEMENTI CON RELAZIONI SIMILI).

IL DUPLICATORE HA UNA STRATEGIA PER RESISTERE m MOSSE.

DEF.: 0-EQUIVALENZA: $\Leftrightarrow$ LA FUNZIONE PARZIALE $f: A \hookrightarrow B$ T.C. $f(a_i)=b_i$, $f(C^A)=C^B$ E' UN ISOMORFISMO FRA LE STRUTTURE GENERATE.

$\forall$ C = SIMBOLO ~~[illegible]~~ IN L

QUESTO E' EQUIVALENTE A RICHIEDERE CHE PER OGNI L-FORMULA ATOMICA $\varphi(x_1, \dots, x_l)$ (VARIABILI LIBERE CONTENUTE IN $\{x_1, \dots, x_l\}$)

$A \models \varphi(x_1, \dots, x_l) \Leftrightarrow B \models \varphi(x_1, \dots, x_l)$.

$m+1$-EQUIVALENZA: IL DUPLICATORE PUÒ RESISTERE $(m+1)$ MOSSE, OSSIA: PER OGNI MOSSA DI "$\forall$", IL DUPLICATORE "$\exists$" HA UNA CONTROMOSSA CHE LO PORTA IN UNA SITUAZIONE IN CUI RESISTE ALTRE m MOSSE:

$$\begin{cases} \forall a \in A, \exists b \in B \text{ T.C. } A, a_1, \dots, a_l, a \sim_m B, b_1, \dots, b_l, b \text{ E} \\ \forall b \in B, \exists a \in A \text{ T.C. (IDEM).} \end{cases}$$

(CONDIZIONI "VA E VIENI")

DEF.: φ FORMULA; ~~[illegible]~~ $\rho(\varphi)$ SI DEFINISCE COME:

$\rho(\varphi)=0 \Leftrightarrow \varphi$ NON HA $\forall, \exists$; $\rho(\neg\varphi)=\rho(\varphi)$;

$\rho(\varphi\wedge\psi)=\rho(\varphi\vee\psi)=\max\{\rho(\varphi), \rho(\psi)\}$;

$\rho(\exists x \varphi) = 1 + \rho(\varphi) = \rho(\forall x \varphi)$.

TEOR. (E.F.): $\mathcal{A}, a_1, \dots, a_\ell \sim_m \mathcal{B}, b_1, \dots, b_\ell \iff$
$\iff (\forall \varphi(x_1, \dots, x_\ell), \rho(\varphi) \leq m, \mathcal{A} \models \varphi(x_1, \dots, x_\ell) \iff$
$\iff \mathcal{B} \models \varphi(x_1, \dots, x_\ell))$.

Enunciato teorema di EF

COR.: $\mathcal{A} \sim_m \mathcal{B}\ \forall m \iff \mathcal{A} \equiv \mathcal{B}$.

ES.: $L = \{<\}$; $(\mathbb{R}, <^{\mathbb{R}}) \not\sim_3 (\mathbb{Z}, <^{\mathbb{Z}})$. INFATTI:

$a^{\exists}\quad b^{\exists}\quad 1^{\forall}\quad 2^{\forall}$

T.C. $b > a$

$\left(\frac{a+b}{2}\right)^{\forall}$

ED $\exists$ PERDE.

INFATTI LA FORMULA CHE DISTINGUE LE DUE STRUTTURE HA RANGO 3: $\forall x \forall y \exists z (x \neq y \rightarrow x < z \wedge z < y)$;

INVECE, $(\mathbb{R}, <) \sim_m (\mathbb{Q}, <)\ \forall m$. ANZI, SONO

∞-EQUIVALENTI: IL DUPLICATORE PUÒ VINCERE ANCHE SENZA SAPERE QUANTO DURA IL GIOCO.

ES.: $\mathbb{Z} + \mathbb{Z} = \mathbb{Z} \times \{0\} \cup \mathbb{Z} \times \{1\}$, CON $(a,0) < (b,1)\ \forall a,b$ E ALL'INTERNO DELLE DUE COPIE VALE L'ORDINE DI $\mathbb{Z}$.

$\mathbb{Z} + \mathbb{Z} \sim_m \mathbb{Z}\ \forall m$ MA NON ∞-EQUIVALENTE.

GIOCHI DI EHRENFEUCHT-FRAISSE 6/3/2008

A, B L-STRUTTURE, $a_0, \dots, a_k \in A$, $b_0, \dots, b_k \in B$;

$A, a_0, \dots, a_k \overset{0}{\sim} B, b_0, \dots, b_k \Leftrightarrow$ LA FUNZIONE $a_i \mapsto b_i$ INDUCE UN ISOMORFISMO TRA $\langle a_0, \dots, a_k \rangle_A$ E $\langle b_0, \dots, b_k \rangle_B$ $\Leftrightarrow$

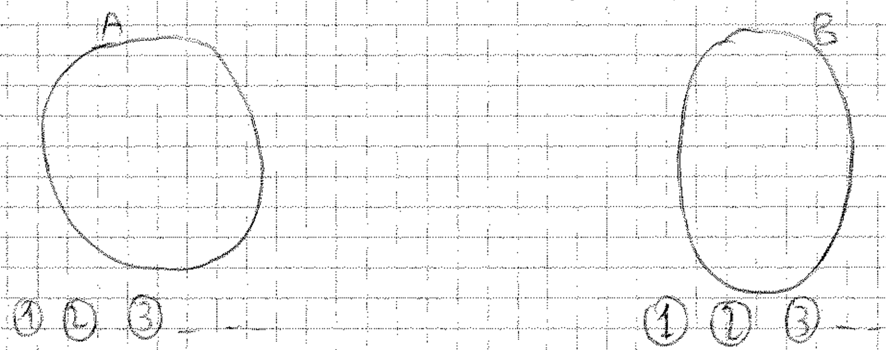
$\Leftrightarrow$ PER OGNI FORMULA $\varphi(x_0, \dots, x_k)$ ATOMICA, $A \models \varphi \Leftrightarrow B \models \varphi$.

INDUTTIVAMENTE DEFINIAMO $A, a_0, \dots, a_k \sim_{m+1} B, b_0, \dots, b_k \Leftrightarrow$

$\Leftrightarrow \forall a \in A, \exists b \in B$ (E $\forall b \in B \ \exists a \in A$) T.C.

$A, a_0, \dots, a_k, a \sim_m B, b_0, \dots, b_k, b$.

VARIANTE: $\overset{k}{\sim}_m$; m E' IL NUMERO DELLE MOSSE, K E' IL NUMERO DELLE PEDINE. K E' FISSATO ALL'INIZIO DEL GIOCO.



peable games

AD OGNI MOSSA, ∀ PIAZZA UNA PEDINA ⓘ SU UN ELEMENTO DI A O DI B, ∃ RISPONDE CON LO STESSO ⓘ.

~~DOPO K+1~~ ALLA K+1-ESIMA MOSSA, ∀ DEVE SPOSTARE UNA PEDINA ~~A B~~ ED ∃ LA STESSA PEDINA NELL'ALTRA STRUTTURA.

FORMALMENTE: $A, a_0, \dots, a_k \overset{k}{\sim}_{m+1} B, b_0, \dots, b_k$ SE

$\forall i \le k$, $\forall a \in A \ \exists b \in B$ (E $\forall b \in B \ \exists a \in A$) T.C.

$A, a_0, \underset{\uparrow}{a}, a_k \overset{k}{\sim}_m B, b_0, \underset{\uparrow}{b}, b_k$ (DOVE a_i, b_i SCOMPAIONO).

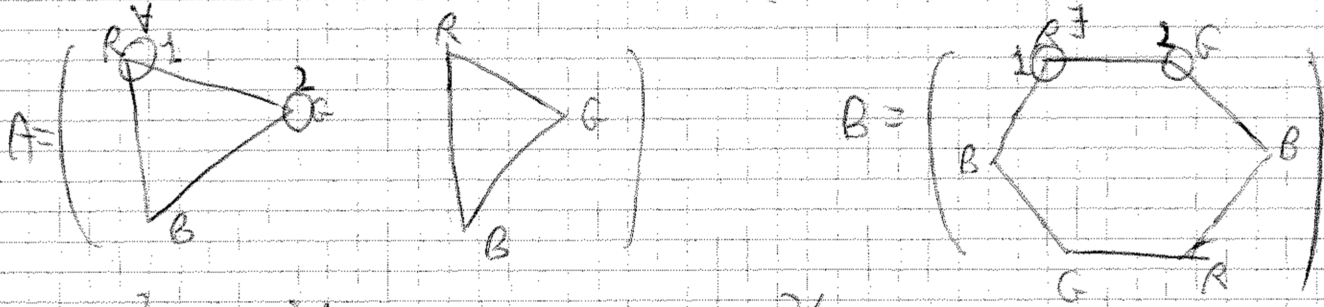
POSTO i-ESIMO POSTO i-ESIMO

~~DA~~ E CI POSSONO ESSERE DEGLI SPAZI VUOTI (QUANDO LE K PEDINE NON SONO ANCORA STATE PRESE TUTTE).

OSS: $\sim_m \Rightarrow \overset{k}{\sim}_m \ \forall m$

ES.: $L = \{E^2, G^1, R^1, B^1\}$ (E BINARIA, R, G, B UNARIE)

L-STRUTTURA $\sim$ GRAFO COLORATO (SE TUTTI I NODI HANNO UN SOLO COLORE)



$A \overset{2}{\underset{m}{\sim}} B$ $\forall m$, MA $A \overset{3}{\underset{m}{\not\sim}} B$.

TEOR.: $A, a_0, \dots, a_\ell \overset{k}{\underset{m}{\sim}} B, b_0, \dots, b_\ell \iff$ PER OGNI FORMULA $\varphi(x_0, \dots, x_\ell)$ DI RANGO DI QUANTIFICAZIONE $\le m$ E AL MASSIMO K VARIABILI, $A \models \varphi(a_0, \dots, a_\ell) \iff B \models \varphi(b_0, \dots, b_\ell)$.

NELL'ESEMPIO SOPRA, LA FORMULA $\exists x, y, z\, (E(x,y) \wedge E(y,z) \wedge E(z,x))$ E' VERA IN A MA NON IN B.

ES.: $L = \{<^2, S^1\}$, $<$ ORDINE TOTALE;

UNA STRINGA BINARIA PUO' ESSERE VISTA COME UN ORDINE TOTALE IN CUI GLI "1" INDICANO GLI ELEMENTI PER CUI VALE S.

DICIAMO CHE LA STRINGA BINARIA $\models \varphi \iff$ LA STRINGA HA ALMENO 3 "1". PER QUESTO BASTANO 2 VARIABILI:

$$\exists x\, [S(x) \wedge \exists y\, [y > \underset{\substack{\uparrow \\ \text{LIBERA}}}{x} \wedge S(y) \wedge \exists \underset{\substack{\uparrow \\ \text{LEGATA}}}{x}\, [x > y \wedge S(x)]]].$$

stringhe binarie

QUINDI, $01011 \overset{2}{\not\sim} 01001$.

NOTARE: $\overset{k}{\sim} \iff \overset{k}{\underset{m}{\sim}}\ \forall m$, MENTRE SI VEDE DAL TEOREMA CHE $\underset{m}{\sim} \iff \overset{k}{\underset{m}{\sim}}\ \forall k$.

ES.: $L = \{<^2, \text{MIN}, \text{MAX}\}$, $<$ ORDINE TOTALE. A, B L-STRUTTURE $(A, <^A)$, $(B, <^B)$; $|A| > 2^m$, $|B| > 2^m$ ALLORA SONO NECESSARIAMENTE m-EQUIVALENTI.

INFATTI SIANO $d(x,y) = \#\{z : x < z \le y \vee y < z \le x\}$

E $d_\ell = \{ d(x,y)$ SE $d(x,y) < 2^\ell$

LA STRATEGIA PER IL DUPLICATORE È MANTENERE $d_j(x, y)$, CON j È IL NUMERO DELLE MOSSE MANCANTI ALLA FINE. OSSIA:

$A, a_0, \dots, a_l \overset{< d_m}{\sim} B, b_0, \dots, b_l$ SE $d_{m}(a_{i_1}, a_{i_2}) = d_{m}(b_{i_1}, b_{i_2})$.

HO CHE $\overset{d_{m+1}}{\sim}$ SODDISFA LE STESSE PROPRIETÀ INDUTTIVE DI $\underset{m}{\sim}$: $\overset{d_{m+1}}{\sim} \iff \forall a \in A \; \exists b \in B : A, a_0, \dots, a_l, a \overset{< d_m}{\sim} B, b_0, \dots, b_l, b$
$\forall b \in B \; \exists a \in A$

SUPPONIAMO $a_0 < a_1 < \dots < a_l$ E $a_0 < a < a_1$; ALLORA $b_0 < b_1 < \dots < b_m$. SE VALE $\overset{d_{m+1}}{\sim}$ VOGLIAMO DIMOSTRARE CHE $\exists\, b,\ b_0 < b < b_1$ T.C. $\overset{d_m}{\sim}$.

SE $d_{m+1}(a_0, a_1) = \infty$, ~~ALLORA $d(a_0, a_1) \geq 2^{m+1}$~~, $= d_{m+1}(a_1, a)$; ALLORA $d(a_0, a) \geq 2^{m}$, $d(a_1, a) \geq 2^{m} \Rightarrow$
$\Rightarrow d(a_0, a_1) \geq 2^{m+1} \Rightarrow d(b_0, b_1) \geq 2^{m+1}$ (PERCHÉ SONO d_{m+1}-EQUIV.) $\Rightarrow \exists\, b$ CHE SODDISFA. IDEM PER GLI ALTRI CASI (ESERCIZIO). OSS.: $\overset{< d_0}{\sim} = \overset{<}{\sim}$.

OSS.: $A, \vec{a} \overset{d_m}{\sim} B, \vec{b} \Rightarrow A, \vec{a} \underset{m}{\sim} B, \vec{b}$: INFATTI LA STRATEGIA PER RESISTERE m MOSSE È PRESERVARE LA DISTANZA m-ESIMA.

Even is not FO

DA CUI, $|A| \geq 2^{m}$, $|B| > 2^{m} \Rightarrow A \underset{m}{\overset{d_m}{\sim}} B \Rightarrow A \underset{m}{\sim} B$.

ES.: $m = 3$, $|A| = 9$, $|B| = 10$: $1 \text{———} 9 \underset{3}{\sim} 1 \text{———} 10$.

COROLL.: NON ESISTE $\varphi \in L\{<, \mathrm{MIN}, \mathrm{MAX}\}$ T.C. $A \models \varphi \iff |A|$ EVEN, DOVE A INTERPRETA $<$ COME ORDINE TOTALE.

INFATTI, SE $\mathrm{rk}(\varphi) = m$, SCELGO $|A|, |B| > 2^{m}$, $A \underset{m}{\sim} B$ E ALLORA BASTA PRENDERE UNA PARI E L'ALTRA DISPARI. $\boxed{\text{OK}}$

RICORDIAMO CHE: $\rho(\exists x\, \varphi) = \rho(\varphi) + 1$, $\rho(\varphi \overset{\wedge}{\vee} \psi) = \max\{\rho(\varphi), \rho(\psi)\}$, $\rho(\neg \varphi) = \rho(\varphi)$.

DIM TEOR.: $A, \vec{a} \underset{m}{\sim} B, \vec{b} \iff [\forall \varphi(\vec{x}),\ g(\varphi) \leq m,\ A \models \varphi(\vec{a}) \iff B \models \varphi(\vec{b})]$. DIMOSTRIAMO $\Rightarrow$.

$m=0$ OK. $m \Rightarrow m+1$: $A, \vec{a} \underset{m+1}{\sim} B, \vec{b}$; LA MINIMA FORMULA SU CUI DIFFERISCONO DEVE INIZIARE CON UN QUANTIFICATORE. SUPPONIAMO CHE SIA $\exists$: $\varphi(\vec{x}) = \exists y\, \vartheta(\vec{x}, y) \Rightarrow g(\vartheta) = m$

E $g(\varphi) = m+1$

dai giochi all'equivalenza elementare

$A \models \exists y\, \vartheta(\vec{a}, y) \Rightarrow \exists\, a' \in A$ T.C. $A \models \vartheta(\vec{a}, a')$. ORA, ESSENDO A E B $m+1$ EQUIV., SO CHE $\exists\, b'$: $A, \vec{a}, a' \underset{m}{\sim} B, \vec{b}, b'$; PER IP. INDUTTIVA, $B \models \vartheta(\vec{b}, b') \Rightarrow B \models \exists y\, \vartheta(\vec{b}, y)$. $\boxed{OK}$

ESERC.: LA VERSIONE $\overset{k}{\underset{m}{\sim}}$ È IDENTICA MA CONSIDERO FORMULE CON $x_1, \dots, x_k$: QUINDI L'a' INVECE DI AGGIUNGERLO LO METTO IN i-ESIMA POSIZIONE.

Dall'equivaleza elementare ai giochi. Lacunoso.

dipende dalla cardinalita' di A

posposto

$\Leftarrow$ $A, \vec{a} \underset{m}{\equiv} B, \vec{b}$ SE VERIFICANO LE STESSE FORMULE DI RANGO $\leq m$. A MENO DI EQUIVALENZA, $\forall m\ \exists$ # FINITO DI FORMULE DI RANGO m. L'IDEA È: DATA UNA L-STRUTTURA A, DATI $\vec{a} = (a_1, \dots, a_l)$ IN A, DEFINISCO $\varphi^m_{A,\vec{a}}(x_1, \dots, x_l)$ T.C. $B, \vec{b} \underset{m}{\sim} A, \vec{a} \iff B \models \varphi^m_{A,\vec{a}}(\vec{b})$. A QUEL PUNTO IL GIOCO SAREBBE FATTO. FISSO $A, \vec{a}$;

$B, \vec{b} \underset{m+1}{\sim} A, \vec{a}$: $\displaystyle\bigwedge_{a' \in A} \exists y\, \varphi^m_{A,\vec{a},a'}(x_1, \dots, x_l, y) \wedge \forall y \bigvee_{a' \in A} \varphi^m_{A,\vec{a},a'}(x_1, \dots, x_l, y) = \varphi^{m+1}_{A,\vec{a}}$ INOLTRE, $\varphi^0_{A,\vec{a}}(\vec{x}) = \bigwedge \psi$

ψ ATOMICA O NEGAZ. ATOMICA $A \models \psi$

ORA, DATO $B, \vec{b}$, $B \models \varphi^{m+1}_{A,\vec{a}}(\vec{b})$ SE $\forall a' \in A\ \exists b' \in B\ B \models \varphi^m_{A,\vec{a},a'}(\vec{b}, b')$ *

MA * PER INDUZIONE EQUIVALE A $A, \vec{a}, a' \underset{m}{\sim} B, \vec{b}, b'$. $\boxed{OK}$

ESERC.: OGNI CLASSE FINITA DI STRUTTURE È SCRIVIBILE COME MODELLO DI UNA TEORIA (OGNI CLASSE È ASSIOMATIZZABILE), MA NON SEMPRE CON UN NUMERO FINITO DI ASSIOMI.

7/3/2008

L LINGUAGGIO DEL 1° ORDINE, K = ~~LINGU~~ CLASSE DI L-STRUTTURE FINITE CHIUSA PER ISOMORFISMO;

Mod(T) = CLASSE DEI MODELLI FINITI DI T (T = INSIEME DI L-ENUNCIATI)

ALLORA: TEOR.: OGNI K E' DELLA FORMA Mod(T), NON NECESSARIAMENTE DELLA FORMA Mod(φ), φ SINGOLO L-ENUNCIATO.

DEF.: K E' UNA CLASSE ELEMENTARE SE $\exists\, \varphi$: K = Mod(φ).

DIM.: A L-STRUTTURA, $|A| = n < +\infty$ (L FINITO); ALLORA $\exists$ UNA L-FORMULA φ_A T.C. $\forall B$, $B \models \varphi_A \iff B \simeq A$.

{ESEMPIO: $L = \{+\}$, $A = (\mathbb{Z}/_{2\mathbb{Z}}, +^A)$; ALLORA

$\varphi_A = \exists x, y\ (x+x = x \wedge x+y = y \wedge y+x = y \wedge y+y = x) \wedge \forall z (z = x \vee z = y)$
$x \neq y$ ECC.

E QUINDI, $B \models \varphi_A \iff B \simeq A$.

IN GENERALE, DUNQUE, $\varphi_A = \exists x_1, \dots, x_n \left(\bigwedge_{\substack{\varphi \text{ ATOMICA} \\ A \models \varphi(a_1,\dots,a_n)}} \varphi(x_1,\dots,x_n) \right. \wedge$

$\left. \bigwedge_{\substack{\varphi \text{ ATOMICA} \\ A \models \neg\varphi}} \neg\varphi(x_1,\dots,x_n) \right) \forall z (z = x_1 \vee \dots \vee z = x_n)$ $x_i \neq x_j$ PER $i \neq j$.}

ogni classe sarebbe "elementare" se ammettessimo infiniti assiomi

{ES.: $K = \{\mathbb{Z}/_2, \mathbb{Z}/_5, \mathbb{Z}/_{11}, \dots\}$; $\forall n \in \mathbb{N}$, $\exists$ # FINITO DI L-STRUTTURE DI CARD. n A MENO DI ISOMORFISMO.}

DATO n, SIANO $A_1^n, \dots, A_k^n \in K$ DI CARD. n IN K, NON ISOMORFE CON K MASSIMALE (SONO TUTTE). $\psi_n = \exists x_1, \dots, x_n,\ x_i \neq x_j$, $\forall z, z = x_1 \vee \dots \vee z = x_n$. ALLORA, $T = \{\psi_n \Rightarrow \varphi_{A_1^n} \vee \dots \vee \varphi_{A_k^n}, n \in \mathbb{N}\}$ $\boxed{\text{OK}}$

QUANDO K = Mod(φ)?

TEOR.: $\exists \varphi\, K = \text{Mod}\, \varphi \iff \exists\, n \in \mathbb{N}$: K E' CHIUSO PER $\underset{n}{\simeq}$.

DIM.: ~~KASSA~~ $\Rightarrow$ K = Mod(φ); SIA $n = \rho(\varphi)$;

$A \in K$, $B \underset{n}{\simeq} A$, $A \models \varphi \Rightarrow B \models \varphi$ (PERCHE' $A \equiv_n B$) $\Rightarrow B \in K$.

~~...~~,

DEF.: A L-STRUTTURA; $Th(A) = \{\varphi : A \models \varphi\}$ (TEORIA DELLA STRUTTURA)

$a \in A$: $\text{TIPO}_A(a) = (L \cup \{x\}\text{-TEORIA}) \cap \{\varphi(x) : A \models \varphi(a)\}$.

ES.: IN $(\mathbb{R}, >, +, 0)$ CI SONO SOLO 3 TIPI: POSITIVO, NEGATIVO, NULLO

$\text{TIPO}(3) = \text{TIPO}(5) \neq \text{TIPO}(-3) \neq \text{TIPO}(0)$: $\exists\, f: A \to A$ AUTOMORFISMO T.C. $f(3) = 5$ ($x \mapsto \frac{5}{3}x$).

IN GENERALE, SE $f: A \to B$ ISOMORFISMO, $A \models \varphi(a) \Leftrightarrow B \models \varphi(f(a))$

A L-STRUTTURA, $a \in A$; $P^n_{A,a} = \{\varphi(a) \mid A \models \varphi(a),\ \rho(\varphi) \leq n\}$;

È EQUIVALENTE A $\varphi^n_{A,a}(x)$

(FINE DIM. LA PROSSIMA VOLTA)

$\varphi^n_A \equiv Th(A) \cap \{\varphi : \rho(\varphi) \leq n\}$;

$\forall$ L-STRUTTURA B, $b_1, \dots, b_l \in B$; $B, b_1, \dots, b_l \models \varphi^n_{A, a_1, \dots, a_l}(x_1, \dots, x_l) \Leftrightarrow$
$\Leftrightarrow B, b_1, \dots, b_l \sim_n A, a_1, \dots, a_l$.

TEOR.: $\vartheta(x_1, \dots, x_l)$, $\rho(\vartheta) \leq n$; $A \models \vartheta(a_1, \dots, a_l) \Rightarrow$
$\Rightarrow \models \varphi_{A, a_1, \dots, a_l}(x_1, \dots, x_l) \Rightarrow \vartheta(\vec{x})$ ($\models \varphi$ VUOL DIRE CHE È VERIFICATA IN OGNI L-STRUTTURA) E SE $A \models \neg\vartheta$, $\models \varphi \Rightarrow \neg\vartheta$.

DIM.: $B, b_1, \dots, b_l \models \varphi^n_{A, a_1, \dots, a_n}(x_1, \dots, x_l)$ VUOL DIRE

$B, b_1, \dots, b_l \sim_n A, a_1, \dots, a_l$ MA PER IL TEOR. DI EHRENFEUCHT-FRAÏSSÉ, $B, \vec{b} \equiv_n A, \vec{a}$ DA CUI SEGUE. $\boxed{\text{OK}}$

15/3/2008

T MACCHINA DI TURING, T CALCOLA UNA F. PARZIALE DA STRINGHE BINARIE $\longrightarrow$ STRINGHE BINARIE

T POLINOMIALE $\Rightarrow \left(|w| \leq n \Rightarrow T(w)\downarrow_{\leq poly(|w|)} \right)$

DEFINIAMO $P = \bigcup_{k \in \mathbb{N}} DTIME(n^k)$; $EXP = \bigcup_k DTIME(2^{n^k})$

DETERMINISTIC TIME

$P \neq EXP$; $P \subset NP \subset EXP$, MA NP NON SI SA DOVE STA.

SAT $= \{ \varphi$ BOOLEANA IN CNF [CONJUNCTED NORMAL FORM] T.C. φ E' SODDISFACIBILE $\}$ SAT $\subset$ FORMULE.

CNF $\bigwedge_i \bigvee_j \pm x_{ij}$ (CONGIUNZIONE E NEGAZIONE DI FORMULE)

definizione di SAT e CLIQUE

SAT E' NP-COMPLETO.

CLIQUE $= \{ (G, K)$, G GRAFO, $K \in \mathbb{N}$, G HA UNA CLIQUE DI CARD. $K \}$

$G = (V, E)$, $E \subset V^2$, $E(x,y) \rightarrow E(y,x)$, $\neg E(x,x)$

UNA CLIQUE E' UN SOTTOGRAFO COMPLETO.

ANCHE CLIQUE E' NP-COMPLETO.

CODIFICHE

SAT: {FORMULE} , SAT E' UN SOTTOINSIEME DI L-STRUTTURE

$\downarrow$

L-STRUTTURE

$\downarrow$ (BIN)

STRINGHE BINARIE

LA PRIMA CODIFICA E' ABBASTANZA NATURALE, LA SECONDA DIPENDE DALL'ORDINE.

DEF: $A \subset$ {STRINGHE BINARIE}; $A \in$ PTIME $\Leftrightarrow$ ESISTE T MDT CHE LAVORA IN TEMPO POLINOMIALE E CHE CI METTE UN TEMPO POLINOMIALE PER RICONOSCERE SE UNA STRINGA STA IN A O NO. IDEM PER EXPTIME.

A E' IN NP SE $\exists B \in P$ T.C. $w \in A \Leftrightarrow \exists x, (x, w) \in P$

E $|x| \leq p(|w|)$ (POLINOMIO).

SAT $\in$ NP: $\varphi \in SAT \Leftrightarrow \exists v$ T.C. $\varphi^v = 1$ E $|v| \leq poly(\varphi)$.

(v E' UNA VALUTAZIONE: $VAR(\varphi) \to \{0, 1\}$).

$A \subset \{$STR. BIN.$\}$ E' NP-COMPLETO SE $A \in NP$ E $\forall B \in NP$, $B \leq_p A$;

$B \leq_p A \Leftrightarrow \exists f$: STR. $\to$ STR., $f \in$ PTIME, T.C.

$w \in B \Leftrightarrow f(w) \in A$.

$B \leq_p A$: $A \in P \Rightarrow B \in P$.

LO DIMOSTREREMO

IL FATTO CHE SAT E' NP-COMPLETO CI DICE:

$NP = P \Leftrightarrow SAT \in P$.

VOGLIAMO DIMOSTRARE CHE CLIQUE $\in$ NP-COMPLETO, OSSIA $SAT \leq_p CLIQUE$

$$CNF \ni \varphi_0 = (x_1 \wedge \neg x_2 \vee x_3) \wedge \underbrace{(\neg x_1 \vee \neg x_2 \vee x_4)}_{\text{CLAUSOLA}} \wedge (x_2 \vee \neg x_3 \vee x_5) \wedge (x_4 \vee \neg x_4) \wedge (x_5 \vee \neg x_5).$$

x_i = VARIABILE;

$\downarrow$

L-STRUTTURA $L = \{P^2, N^2\}$;

codifica di una CNF con una struttura

LETTERALE = $\{x_i, \neg x_i\}$

φ_0 LA CONVERTO IN A_φ = ?

$m = \max\{\#\text{CLAUSOLE}, \#\text{VARIABILI}\}$; $m = \{1, \dots, m\}$.

LA L-STRUTTURA CHE CODIFICA φ_0 E': $A_\varphi = (m, P^A, N^A)$ DOVE:

$P(i, j) \Leftrightarrow$ LA VARIABILE x_j COMPARE POSITIVAMENTE NELLA CLAUSOLA

C_i; $N(i, j) \Leftrightarrow$ NEGATIVAMENTE.

NELL' ESEMPIO: $A_\varphi = (5, P, N)$, $P = \{(1,1), (1,3), (2,4), (3,2), (3,5), (4,4), (5,5)\}$

$N = \{(1,2), (2,1), (2,2), (3,3), (4,4), (5,5)\}$.

L-STRUTTURE → STRINGHE (bin): DATA UNA STRUTTURA (A, P, N), $P, N \subset A^2$, FISSIAMO UN ORDINE $\leq$ TOTALE SU A; ESSENDO A FINITA, $|A| = n$, POSSIAMO IDENTIFICARE $A = n$. ALLORA $P \subset n \times n \cong 2^5$

OSS.: LE STRINGHE SI POSSONO INTERPRETARE COME L-STRUTTURE IN CUI $L = \{S^1\}$. ALLORA UNA REL. BINARIA SU n DIVENTA UNA STRINGA DI LUNGHEZZA n^2: $P(a, b) \iff P'(a n + b)$. P' È PROPRIO $\underbrace{}_{\text{bin } P}$ bin P, LA CODIFICA BINARIA DI P.

codifica di una struttura ordinata con una stringa binaria

ALLORA, $\text{bin}(A, P^2, N^2) \iff (\text{bin } P)(\text{bin } N)$.

OSS.: (STRINGHE → L-STRUTTURA → STRINGHE) È L'IDENTITÀ.

ORA, SAT $\in$ NP $\iff \{\text{bin}(\varphi) : \varphi \in \text{SAT}\} \in$ NP.

ABBIAMO COSÌ CODIFICATO SAT. ORA, CODIFICHIAMO CLIQUE.

CLIQUE $= \{(G, k) : G$ HA UNA k-CLIQUE$\}$

codifica di clique con una struttura ordinata

(G, k) DIVENTA UNA L-STRUTTURA ORDINATA CON $L = \{\leq, E^2, c\}$.

LA COSTANTE c SI CODIFICA IN UNA STRINGA LUNGA $\lceil \log_2 n \rceil$.

ORA DIMOSTRIAMO SAT $\leq_p$ CLIQUE.

$\exists g$: L-STRUTTURE → L'-STRUTTURE T.C. $\varphi \in$ SAT $\iff g(\varphi) \in$ CLIQUE E g È CALCOLABILE IN TEMPO POLINOMIALE (OSSIA, $\exists g'$: STRINGHE → STRINGHE, POLINOMIALE, T.C. $g'(\text{bin}(\varphi)) = \text{bin}(g(\varphi))$).

$\varphi \in$ CNF $\to A_\varphi = (G, k)$; φ ABBIA n VARIABILI E r CLAUSOLE;

SCEGLIAMO $A_\varphi = (G, r+1) = ((r \times 2n) + 1, E^2, r+1)$
VERTICI

PTime riduzione SAT < Clique

L'INSIEME DEI VERTICI DI G È $((\text{CLAUSOLE} \times \text{LETTERALI}) \cup \{0\})$
(I LETTERALI SONO $x_1, \neg x_1, \ldots, x_n, \neg x_n$)

GLI ARCHI IN G SONO TUTTI TRANNE QUELLI FRA UN LETTERALE E UN LETTERALE NEGATO APPARTENENTE A UN'ALTRA CLAUSOLA NÉ QUELLI FRA x_i E $x_i \neg$, $\forall i$. GLI ARCHI DA V_0 ~~AL NOD~~ PARTONO SOLO VERSO LE VARIABILI PRESENTI IN CIASCUNA CLAUSOLA.
(OSSIA, DOVE COMPARE CIASCUN LETTERALE, NEGATO O MENO)

$E(\langle c, l \rangle, \langle c', l' \rangle) \iff (c \neq c' \land l \neq \neg l')$

$E(w_0, \langle c, l \rangle) \iff l$ APPARE IN c.

SICCOME NON CI SONO COLLEGAMENTI ORIZZONTALI, UNA $n+1$-CLIQUE DEVE AVERE UNO E UN SOLO PUNTO PER CLAUSOLA, PIÙ w_0.

ORA, C'È UNA CORR. BIUNIVOCA FRA CLIQUE E VALUTAZIONI CHE SODDISFANO LA FORMULA.

È POLINOMIALE: BASTA OSSERVARE CHE $|G|$ È POLINOMIALE RISPETTO A φ.

L-STRUTTURA A T.C. $|A| = n$, QUANT'È $|bin(A)|$?

C'È UNA RELAZIONE POLINOMIALE.

C'È DI PIÙ: $3SAT \leq CLIQUE$ TRAMITE RIDUZIONE AL PRIMO ORDINE.

QUERY: L-STRUTTURE $\longrightarrow L'$-STRUTTURE

ES: LA SOMMA BINARIA, $L = \{\leq, A^1, B^1\}$, $L' = \{\leq, S^1\}$ (ECC. ECC.)

ADDIZIONE BINARIA VISTA COME QUERY: 24/5/2008

$I: L\text{-STRUTTURE} \longrightarrow L'\text{-STRUTTURE}$

Addizione binaria

NEL NOSTRO CASO, COPPIE DI STRINGHE $\longrightarrow$ STRINGHE

$L = (X, \leq A^1, B^1)$ $L' = (X, \leq, S^1)$

DOVE X AD ES. PUÒ ESSERE T.C. $|X| = 5$ (QUINDI $X \simeq 5$, ESSENDO ORDINATO).

DEFINIRE S A PARTIRE DA A E B: DEFINIRE $\varphi_S(x) \in L^?$ CHE EQUIVALGA ALLA SOMMA.

DEFINIAMO UNA FORMULA AUSILIARIA: $\varphi_{RIPORTO}(x) \equiv (\exists y . x < y)$
$[A(y) \wedge B(y) \wedge (\forall z . x < z < y)\ A(z) \vee B(z)]$.

A QUESTO PUNTO, DEFINISCO: $\alpha \oplus \beta \equiv (\alpha \vee \beta) \wedge \neg(\alpha \wedge \beta) \equiv$
$\equiv \neg(\alpha \Leftrightarrow \beta)$ (OR ESCLUSIVO)

E ALLORA $\varphi_{SOMMA}(x) = A(x) \oplus B(x) \oplus \varphi_{RIPORTO}(x)$

I È UNA QUERY (INTERROGAZIONE) DEL 1° ORDINE (FO), IN PARTICOLARE È CALCOLABILE IN TEMPO POLINOMIALE ($FO \subset PTIME$). IN REALTÀ, $FO_{\leq} \subset LOGSPACE \subset PTIME$.

OSS.: SE HO UNA STRUTTURA $A = (A, \leq, \ldots)$, POSSO IDENTIFICARE A CON n, SE $|A| = n$. POSSO DEFINIRE UNA RELAZIONE $PLUS \subset A \times A \times A$ T.C. $PLUS(i, j, k) \Leftrightarrow i + j = k$, SIMILMENTE $TIMES(i, j, k) \Leftrightarrow ij = k$.

QUESTO MI DEFINISCE $FO_{\leq, +, \cdot}$ (OLTRE A FO E $FO_{\leq}$).

ESERC.: POSSIAMO DEFINIRE $BIT \subset A \times A$, $BIT(i, j) \Leftrightarrow$ IL J-ESIMO BIT DI $(i)_2 = 1$. AL PRIMO ORDINE, DEFINIRE BIT TRAMITE PLUS E TIMES E VICEVERSA (SEMPRE CON $\leq$).

SE $|A| = n$, BIT PERMETTE DI PENSARE UN EL. DI A COME UNA STRINGA DI LUNGHEZZA $\log n$.

TORNIAMO ORA AL NOSTRO PROBLEMA: $SAT \leq CLIQUE$.

LA FUNZIONE $\varphi \longmapsto A_\varphi = (G_\varphi, k)$ È UNA QUERY DEL 1° ORDINE.

FO-Riduzione SAT < Clique

CNF = L-STRUTTURE, $L = \{P^2, N^2\}$; GRAFI = L'-STRUTTURE, $L' = \{E^2, c, \leq\}$

~~QUESTO~~ IL GRAFO CORRISPONDENTE A φ PUÒ ESSERE VISTO COME SOTTOINSIEME DI A^3:

$$I(A) = (G_\varphi, m+1) = (V_\varphi, E^2, m+1)$$

$V_\varphi \subset A^3$; UN VERTICE È UNA TERNA <CLAUSOLA, VARIABILE, SEGNO>,

"SEGNO" È 0 SE COMPARE NEGATIVAMENTE, 1 ~~ALTRIMENTI~~ SE COMPARE POSITIVAMENTE. IL VERTICE w_0 SI DENOTA CON $\langle 0, 0, 2 \rangle$

$$E(\langle c, v, s\rangle, \langle c', v', s'\rangle) \iff [s, s' < 2 \wedge [c \neq c' \wedge (v = v' \Rightarrow s = s')]$$

$$\vee [s = 2 \wedge (s' \neq 1 \wedge P(c', v')) \vee [s' = 0 \wedge N(c', v')]]] \vee [s' = 2 \wedge \text{IDEM}]$$

DOVE IL LINGUAGGIO DI PARTENZA ERA $(A, [\leq, 0, 1, max], P, N)$

INOLTRE, LA FORMULA CHE DEFINISCE IL DOMINIO:

$$V_\varphi \subset A^3 : V_\varphi = \{(x, y, z) : \underbrace{z < 2 \vee (x = 0 \wedge y = 0 \wedge z = 2)}_{FO_{\leq, 0, 1, max}}\}.$$

QUINDI, SAT ≤ CLIQUE È FO-DEFINIBILE ⇒ È PTIME.

DEF.: I : L-STRUTTURE → L'-STRUTTURE

$$A \longmapsto I(A) = (B, E^2)$$

(k-ARIA: $B \subset A^k$)
$E^2 \subset A^{2k}$

ESEMPIO: $L = \{R_1^2, R_2^3\}$ $L' = \{E^2\}$

SIA IL DOMINIO CHE LE RELAZIONI SIANO DEFINIBILI NEL LINGUAGGIO DI PARTENZA.

MACCHINE DI TURING ALTERNANTI

SONO PIÙ POTENTI DELLE MdT NON DETERMINISTICHE, CHE A LORO VOLTA SONO PIÙ POTENTI DELLE MdT DETERMINISTICHE.

IN TERMINI DI CLASSE DI COMPLESSITÀ, $AP \supseteq NP \supseteq P$

SONO UN MODELLO DI CALCOLO PARALLELO.

Alternanza

MdT DETERMINISTICHE: $\delta : \{SIMBOLI\}^{(k)} \times \{STATI\} \longrightarrow \{SIMBOLI\} \times \{STATI\} \times \{S, D\}^{(k)}$

(A UN SOLO NASTRO; K = NUM. NASTRI)

MdT NON DET.: $\delta : \{SIMBOLI\}^{k} \times \{STATI\} \rightarrow P(\{SIMBOLI\}^{k} \times \{STATI\} \times \{S,D\}^{k})$

NELLE ALTERNANTI, GLI STATI SONO PARTIZIONATI IN $STATI^{\forall} \cup STATI^{\exists}$ ($\cup$ STATI DETERMINISTICI)

UNIVERSALI ESISTENZIALI

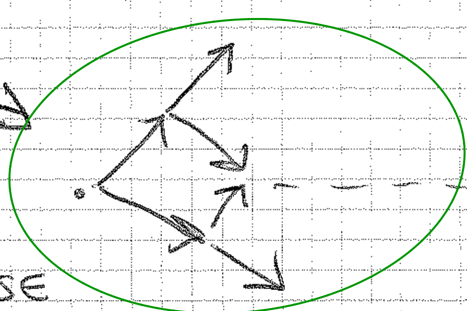
ID = DESCRIZIONE ISTANTANEA = CONFIGURAZIONE = = < CONTENUTO DEI NASTRI, + STATO IN CUI SI TROVA LA MACCHINA, + POSIZIONE TESTINE >

SIA M = MdT DETERM.; ALLORA, INPUT $w \longmapsto ID_w \equiv C_w$ (CONFIG. INIZIALE);

DATE C, C' CONFIGURAZIONI, $C \vdash_M C'$; SE LA MACCHINA È DETERMINISTICA, C' È BEN DETERMINATO. SI HA QUINDI

$C_w \vdash C'_w \vdash C''_w \vdash \ldots$

SE INVECE È NON DETERMINISTICA, POSSO AVERE UN GRAFO DELLE CONFIGURAZIONI: G^M_w



ALCUNE CONF. SONO FINALI; POSSONO ESSERE ACCETTANTI O RIFIUTANTI.

UNA CONF. UNIVERSALE È ACCETTANTE SE TUTTI I FIGLI ACCETTANO (AND LOGICO); UNA CONF. ESISTENZIALE ACCETTA SE ALMENO UNO DEI FIGLI ACCETTA (OR LOGICO).

UNA MdT ALT. IN UNA CONF. $C = \langle S, St, \{S,D\} \rangle$ ACCETTA SE:

1) C È IN UNO STATO FINALE ACCETTANTE, OPPURE
2) C È IN UNO STATO ESISTENZIALE ED $\exists C'$ RAGGIUNGIBILE IN 1 PASSO DA C T.C. C' ACCETTA; OPPURE
3) C È IN UNO STATO UNIVERSALE E TUTTI I C' RAGGIUNGIBILI IN 1 PASSO ACCETTANO

OSS.: (1) E' UN CASO PART. DI 3).

QUINDI, $M(w)$ ACCETTA $\Leftrightarrow C_w$ ACCETTA (CONF. INIZIALE).

UNA MdT NON DET. E' UNA MdT ALTERNANTE SENZA STATI UNIVERSALI.

<u>DEFINIZIONI</u>: $t: \mathbb{N} \to \mathbb{N}$

DTIME$(t(n))$: MACCHINE DETERMINISTICHE CHE OPERANO IN UN TEMPO $t(n)$ E $w : |w| = n$ E' ACCETTATO $\Leftrightarrow w \in L \subset \Sigma^*$.

$\cap$

NTIME$(t(n))$: NON DET.

$\cap$

ATIME$(t(n))$: ALTERN.

time constructible

IDEM PER DSPACE$(s(n)) \subset$ NSPACE$(s(n)) \subset$ ASPACE$(s(n))$; IN QUESTO CASO CONTA IL NUMERO DI CELLE DI MEMORIA OCCUPATE.

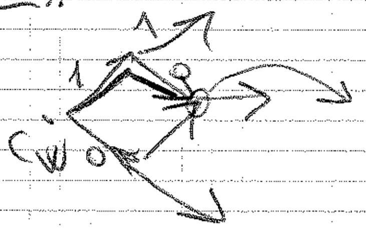
UNA FUNZIONE E' TIME-CONSTRUCTIBLE SE $\exists M$ DETERMINISTICA SE SU INPUT $(0, \dots, 0)$ (LUNGA n) SI FERMA IN UN TEMPO $t(n)$. IDEM PER SPACE-CONSTRUCTIBLE.

CON QUESTA DEFINIZIONE, E' SEMPRE POSSIBILE INCORPORARE UN OROLOGIO IN M CHE IMPONGA DI FERMARSI DOPO $t(n)$.

$NP = \bigcup_k NTIME(n^k)$ (AVEVAMO DETTO: $NP = \exists P$, CON $\exists$ LIMITATO POLINOMIALMENTE: $L \in NP \Leftrightarrow \exists L' \in P : w \in L \Leftrightarrow \exists t : (t, w) \in L'$ E $|t| \leq poly(|w|)$). <u>ESERC.</u>: LE DUE DEFINIZIONI SONO EQUIVALENTI.

<u>DIM.</u>: IL CAMMINO CHE PORTA ALL'ACCETTAZIONE E' IL TESTIMONE:

due definizioni di NP



CAMMINO $= (1\ 0)$; QUINDI LE STRINGHE CHE CODIFICANO I CAMMINI SONO POLINOMIALI IN $|w|$. IL PROBLEMA ASSUMIAMO E' QUELLO CHE DICE SE IL CAMMINO E' POLINOMIALE. IL VICEVERSA E' ANALOGO. $\boxed{ok}$

CLASSE DI COMPL. $\mathcal{C} = \begin{cases} NP \\ NPTIME(n^2) \end{cases}$ ECC. ...)

$co\text{-}\mathcal{C} = \{ L \subset \Sigma^* : \Sigma^* \setminus L \in \mathcal{C} \}$. <u>ES.</u>: $\{3\text{-COLOR}\} \in NP$; $\{NON\ 3\text{-COLOR}\} \in co\text{-}NP$.

OSS.: $\text{co-ATIME}(t(n)) = \text{ATIME}(t(n))$; $\text{co-DTIME}(t(n)) = \text{DTIME}(t(n))$
INVECE CIÒ NON VALE PER NTIME.

ATime = co-ATime

BASTA SCAMBIARE I SÌ CON I NO (E, PER ATIME, GLI $\exists$ CON I $\forall$).

PROBLEMA APERTO: $NP \stackrel{?}{=} \text{co-}NP$ ($\Leftarrow P = NP$) NEL SENSO DEL TEMPO

TEOR. (SAVITCH): $\text{NSPACE}(s(n)) \subset \text{DSPACE}(s(n)^2)$. IN PARTICOLARE,

COR.: $\text{NPSPACE} = \bigcup_k \text{NSPACE}(n^k) = \bigcup_k \text{DSPACE}(n^{2k}) = \text{DPSPACE}$.

QUINDI, NEL SENSO DELLO SPAZIO, $P = NP$ E $NP = \text{co-}NP$.

OSS.: $\text{NTIME}(t(n)) \subset \bigcup_c \text{DTIME}(c^{t(n)})$. IN PART., $P \subset NP \subset \text{EXPTIME}$

TEOR. (2.32 IMMERMAN): $\text{NSPACE}[s(n)] \subset \text{ATIME}(s(n)^2) \subset \text{DSPACE}(s(n)^2)$,
SE $s(n) \geq \log(n)$ E COSTRUIBILE.

Nspace = Dspace = Atime

TEOR. (2.25 IMMERMAN): $(t(n) \geq n,\ s(n) \geq \log n)$

$\bigcup_k \text{ATIME}(t(n)^k) = \bigcup_k \text{DSPACE}(t(n)^k)$; $\text{ASPACE}(s(n)) = \bigcup_k \text{DTIME}(k^{s(n)})$.

OSS.: TIME $\subset$ SPACE PER TUTTE LE CLASSI.

$NL = \bigcup_n \text{NSPACE}(\log(n))$

CONSEGUENZE: $L = \bigcup_n \text{DSPACE}(\log(n))$; $L \subset NL \subset P \subset NP \subset \text{PSPACE} \subset$
$\subset \text{EXPTIME} = \bigcup_{P(n)} \text{TIME}(2^{P(n)})$. INOLTRE, $P \neq \text{EXPTIME}$.

ES.: $\exists x: P(x,y) \Leftrightarrow P(\text{BEST}_P(y), y)$, DOVE BEST È LA FUNZIONE CHE
SCEGLIE UN x: $P(x,y)$ SE ESISTE, SENNÒ UNA COSA QUALUNQUE. È
EQUIVALENTE ALL'ASSIOMA DELLA SCELTA.

$\forall x\, P(x,y) \Leftrightarrow P(\text{WORST}_P(y), y)$, DOVE WORST È LA FUNZIONE CHE SCEGLIE
UN ELEMENTO T.C. SE LUI VINCE, VINCONO TUTTI. $\text{WORST}_P = \text{BEST}_{\neg P}$.

UNA MdT ALTERNANTE QUINDI È UNA MdT NON DETERMINISTICA MUNITA DELLE
FUNZIONI BEST E WORST.

Best, Worst

ESERC.: (2.29) $M\underset{\text{CIRCUITO}}{C}VP \in \text{ASPACE}(\log(n))$, DOVE:

UN CIRCUITO BOOLEANO È UN GRAFO IN CUI I NODI HANNO $\neg, \vee, \wedge$.

UN CIRCUITO È LEGGERMENTE PIÙ PICCOLO DI UNA FORMULA: IN UNA
FORMULA, SE DEVO USARE UNA PROP. PIÙ VOLTE, DEVO SCRIVERLA PIÙ VOLTE.

$C = (\overset{\text{VERTICI}}{V}, \overset{\text{ARCHI}}{E^2}, \overset{\text{AND}}{G_\wedge^1}, \overset{\text{OR}}{G_\vee^1}, \overset{\text{NOT}}{G_\neg^1}, \overset{\text{FOGLIE}}{\text{LEAF}^1}, \overset{\text{F. VERE}}{I^1}, \overset{\text{USCITA}}{r})$

DESCRIVIAMO UN ALGORITMO CHE, SU INPUT C, DICE SE π=VERO.
C=CIRCUITO;
M=MONOTONO (OSSIA, SENZA NEGAZIONI);

SIA C UN CIRCUITO MONOTONO, PER $\omega \in V$ VOGLIAMO CALCOLARE EVAL(ω).
- SE IN INPUT: OUTPUT=VERO $\iff I^{1}(\omega)$, FALSO ALTRIMENTI.
- SE $G_{\wedge}(\omega)$ (ω=AND): IN UNO STATO UNIVERSALE, SCEGLIAMO $b: E(b,\omega)$ (b=WORST(ω)).
- SE $G_{\vee}(\omega)$ (ω=OR): IN UNO STATO ESISTENZIALE, SCEGLIAMO $b: E(b,\omega)$ (BEST(ω)).
- RETURN EVAL(b) E POI CONTINUARE.
- SE ω=π, SCEGLI L'UNICO FIGLIO.

ORA, $E \in 2^{n^2}$, MA LA CODIFICA DI UN NODO IMPIEGA $\log_2 n$. $\boxed{\text{OK}}$

MCVP e' ASPACE(log n). [Imm. p. 37]

IL GO E' P-SPACE COMPLETO

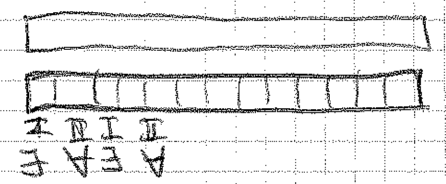
2/11/2006

ECCEZIONI: GOBAN $n \times n$, NIENTE "PASSO", IL GIOCO FINISCE DOPO n^2 MOSSE.

PROBLEMA **QSAT**: DATA UNA FORMULA PROPOSIZIONALE, STABILIRE QUANDO E' VERA.

GEO: DATO UN GRAFO ORIENTATO, STABILIRE SE C'E' UNA STRATEGIA VINCENTE PER IL GIOCO "GEOGRAPHY": PARTENDO DA UNA CITTA' CI SI SPOSTA SU CITTA' VICINE, PERDE CHI NON PUO' PIU' MUOVERSI SENZA VISITARE UNA CITTA' GIA' VISITATA.

PER PERCORRERE IL GRAFO DI GEO: BASTANO DUE ARRAY: UNO DI NUMERI LUNGO n, UNO DI VALORI DI VERITA' LUNGO n.



SE HO UNA SERIE DI CASELLE NON INIZIALIZZATE SEGUITE DA UNA LETTERA, QUESTA LETTERA MI DICE SE IN QUEL PUNTO HO UNA STRATEGIA VINCENTE.
L'ARRAY SUPERIORE AUMENTA OGNI VOLTA DI 1; OGNI VOLTA CHE SI RIPORTA AL POSTO PRECEDENTE, AZZERO LE CASELLE SUCCESSIVE DEL SECONDO ARRAY E CAMBIO LA CASELLA CORRISPONDENTE AL RIPORTO: SE E' $\exists$, SE C'E' UN V SUCCESSIVO LA PONGO V, SENNO' F; SE E' $\forall$, SE TUTTE LE CASELLE SUCCESSIVE SONO V, LA PONGO V, SENNO' F.

QSAT e' PSpace completo

QSAT E' PSPACE-COMPLETO

NEL GRAFO DELLE CONFIGURAZIONI, $\exists$? UN PERCORSO ~~DA~~ DALLA CONF. INIZ. DI M (DATO x) AD UN STATO ACCETTANTE? ($M = MdT$, x = STATO INIZIALE, $|x| = n$, L = LINGUAGGIO)

~~# CONFIGURAZIONI~~ MEMORIA $= n^k$ PER QUALCHE K $\Rightarrow 2^{n^k}$ CONFIGURAZIONI.

QUINDI TUTTO SI CONDUCE AI: DATI A E B, SONO DISTANTI MENO DI 2^{n^k}?

RICORSIVAMENTE: $\exists\, C: d(A,C) \leq 2^{n^k-1}$ E $d(C,B) \leq 2^{n^k-1}$?

DATA UNA FORMULA DI BASE $\varphi_0(A,B)$, DEVE RISPONDERE ALLA DOMANDA:

$\varphi_0(A,B) =$ "$(A \vdash_M B) \vee (A = B)$"?

A=B SI SCRIVE IN FORMA DISGIUNTIVA: $\bigwedge_{i \leq m} (a_i \wedge b_i) \vee (\neg a_i \wedge \neg b_i) \Leftrightarrow$

$\Rightarrow \bigvee_{f} \left(f(a_0) \wedge f(b_0) \wedge f(a_1) \wedge f(b_1) \wedge \ldots \right)$ ← è solo un modo per scrivere A=B

DOVE:

$f(w_1) = \begin{cases} w_{01} & \text{SE } \sigma_1 = V \\ \neg w_{01} & \text{SE } \sigma_1 = F \end{cases}$ ECC. ; SIANO $C_1, \dots, C_m$ VARIABILI DI CONTROLLO (C_i VERA E LE ALTRE FALSE $\Leftrightarrow w_i = b_i$).

ALLORA H: $\forall C_1, \dots, \forall C_m \; (\dots) \; (w_1 = b_1) \wedge C_1 \vee (w_1 \neq b_1) \wedge \neg C_1$ CONTROLLA SE $w_i = b_i$.

$\psi_i(A,B)$ = A E B DISTANO AL PIÙ $2^{m^{i-1}}$;

guess middle configuration

$\psi_i(A,B) = \exists Z \, \forall X \, \forall Y \, [(X = A \wedge Y = Z) \vee (X = Z \wedge Y = B) \rightarrow \psi_{i-1}(X,Y)]$

POI: $P \Rightarrow Q \longrightarrow (\neg P) \vee Q$. POI, PER ELIMINARE LA NEGAZIONE, SI ANALIZZANO SINGOLI CASI IN CUI ~~LA PRIMA~~ L'IPOTESI È FALSA (SONO AL PIÙ $16 m^{2k}$).

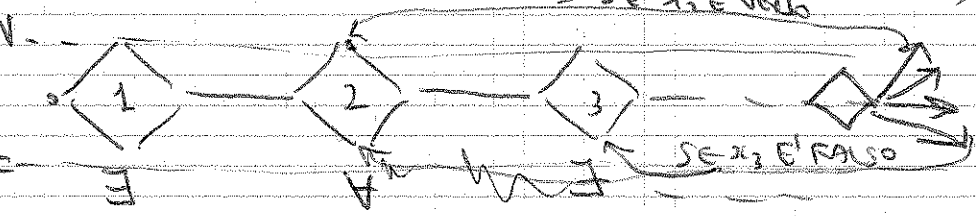
ALLA FINE ABBIAMO UNA FORMULA $\psi_i(A,B)$ LUNGA $c m^k + c \cdot i^2 \simeq c m^k + c m^{2k}$.

QUINDI HO RICONDOTTO UN QUALUNQUE PROBLEMA PSPACE A UNA FORMULA IN QSAT.

$|\psi_i| = c n^k + |\psi_{i-1}| = i c n^k \leq n^k c n^k$

GG È PSPACE-COMPLETO:

SE HO $\exists x_1 \forall x_2 \exists x_3 \dots \varphi(x_1, \dots, x_n)$

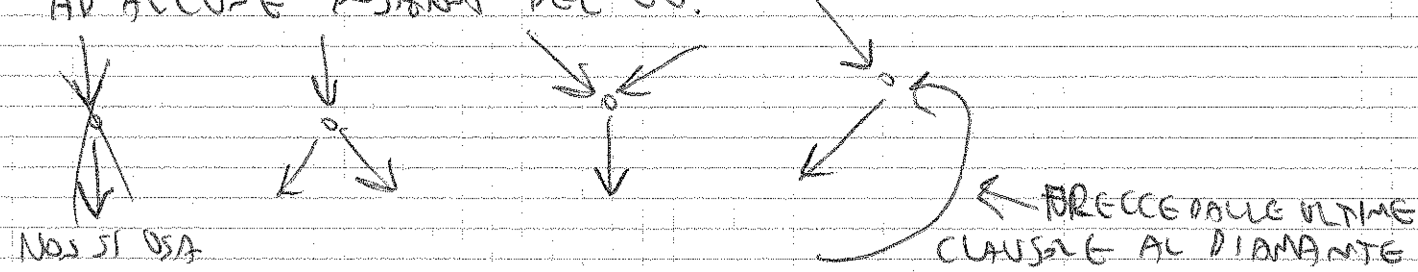


$\exists$ = DIAMANTI DISPARI

$\forall$ = DIAMANTI PARI

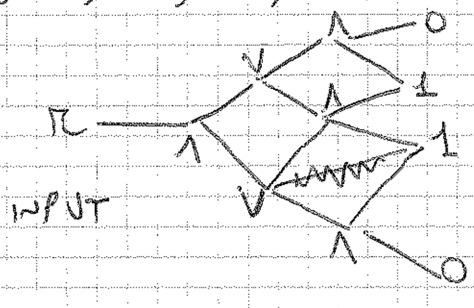
OGNI SCELTA IN UN DIAMANTE i CORRISPONDE A DARE UN VALORE DI VERITÀ AL CORRISPONDENTE x_i. SE DALL'ULTIMA FRECCIA RIESCO AD ANDARE SU UNA CIMA NON VISITATA ($\forall$ SCELTA DELL'ALTRO) HO VINTO, SENNÒ NO.

INFINE: GEOGRAPHY SI RICONDUCE A GO (IN REALTÀ UN CASO PARTICOLARE DI GEOGRAPHY CHE È PSPACE-COMPLETO): I GRAFI DI GEOGRAPHY SI RICOLLEGANO AD ALCUNE POSIZIONI DEL GO:



NON SERVONO

$C = (V, E^2, G^1_\wedge, G^1_\vee, \cancel{\text{Terminali}}, I, r)$ (CIRCUITO)

ES.: 

$\Rightarrow$ OUTPUT: $r = 1$? $r = 0$?

circuit val e' ASpace(log)

SI PUÒ RISOLVERE IN ASPACE($\log n$).

UNA MdT ALTERNANTE È UNA MdT T.C. LA FUNZIONE DI TRANSIZIONE

δ: CONTENUTO CELLA MEMORIA, STATO $\longrightarrow P($CONTENUTO CELLA', STATO' ~~AGGIORNATO~~, SPOSTAMENTO$)$

$\forall, \exists$ — INSIEME DELLE PARTI (MdT NON DETERMINISTIC

LA MACCHINA TIENE IN MEMORIA AL MASSIMO UNO O DUE NODI. "SE LEGGI $w \in V$ E $w \in G_\wedge$ ALLORA ENTRA IN UNO STATO UNIVERSALE E VAI A LEGGERE UN QUALSIASI ALTRO NODO CONNESSO CON w. VICEVERSA, SE $w \in G_\vee$, ENTRA IN UNO STATO ESISTENZIALE. SE w È TERMINALE, ENTRA IN UNO STATO ACCETTANTE O RIFIUTANTE A SECONDA CHE w SIA VERO O FALSO."

PROP.: $ATIME(t(n)) \subset DSPACE(t(n))$ $\big(\overset{\text{ovvio}}{\subset} ASPACE(t(n))\big)$.

IN GENERALE,

$$\begin{array}{c|ccccc} & D & & N & & A \\ \hline TIME & & \subset & & \subset & \\ & \cap & & \cap & & \cap \\ SPACE & & \subset & & \subset & \end{array} \qquad f(n)$$

INOLTRE, $ASPACE(f(n)) = \bigcup_K DTIME(K^{f(n)})$ ECC.

ATime incluso DSpace

DIM.: SIA A UNA ATIME$(t(n))$; SU INPUT w, SIA G_w = GRAFO DELLE COMPUTAZIONI DI A SU INPUT w. OSSIA, UN NODO DI G_w È UNA TERNA $\langle$CONTENUTO MEMORIA, STATO, POSIZIONE$\rangle = C \in G_w$; $C \vdash C'$ SE CI SI VA IN UN PASSO TRAMITE IL PROGRAMMA DI A. $|G_w| = O(1)^{t(n)}$, QUINDI MOLTO GRANDE. MA $A(w)$ ACCETTA $\Leftrightarrow$ IN G_w LA RADICE RICEVE VALORE 1. ORA, DATI DUE SOLI NODI, POSSO ESIBIRE UNA FUNZIONE CHE DICA SE SI PUÒ ANDARE IN UN PASSO DA UNA ALL'ALTRA:

C = [a|b|a|c|q|-|-|-|-|-|-]   C' = [| | | | | | | | | | | |]

AFFINCHÉ C PASSI A C', LE POSIZIONI DELLE DUE FRECCE (puntatori) DEVONO ESSERE DISTANTI AL PIÙ 1, POI C E C' NELLE PARTI LONTANE DA q E q' E NEI PUNTI IN CUI SONO DIVERSE, DEVONO RISPETTARE δ.

LA STRINGA $\langle C_1, \dots, C_n \rangle$, $C_i \in \{0, 1\}$, INDICA IL NODO A CUI SI ARRIVA PRENDENDO QUELLE SCELTE NEL GRAFO. ALLA FINE DELLA STRINGA ACCODIAMO UN SIMBOLO: ? $\Rightarrow$ NON SAPPIAMO SE A ACCETTA O RIFIUTA

SI $\Rightarrow$ ACCETTA

NO $\Rightarrow$ RIFIUTA

QUINDI, ALL'INIZIO ABBIAMO $(\varepsilon, ?)$ (ε = RADICE). LE REGOLE SONO:

$C_1, \dots, C_n, ? \Rightarrow C_1, \dots, C_n, 0, ?$ (0 = A SINISTRA, 1 = A DESTRA)

$\underbrace{C_1, \dots, C_{MAX}}_{\uparrow \text{FOGLIA}}, ? \Rightarrow C_1, \dots, C_{MAX}, \{$SI O NO A SECONDA SE LA FOGLIA ACCETTI O RIFIUTI$\}$

$\langle C_1, \dots, C_n, 0 \rangle$ SI, CON C_n $\exists$ $\Rightarrow \langle C_1, \dots, C_n \rangle$ SI

$\langle C_1, \dots, C_n, 0 \rangle$ NO CON C_n $\exists$ $\Rightarrow \langle C_1, \dots, C_n, 1 \rangle$?

$\langle C_1, \dots, C_n, 1 \rangle^{NO}_{SI} \Rightarrow \langle C_1, \dots, C_n \rangle^{NO}_{SI}$

$\langle C_1, \dots, \underset{\forall}{C_n}, 0 \rangle$ NO $\Rightarrow \langle C_1, \dots, C_n \rangle$ NO

$\langle C_1, \dots, \underset{\forall}{C_n}, 0 \rangle$ SI $\Rightarrow \langle C_1, \dots, C_n, 1 \rangle$?

QUINDI, NEL GRAFO D'ESEMPIO, $(\wedge = \forall, \vee = \exists)$:

ε ?	ε 0 1 1 SI	ε $\underset{\exists}{1}$ 0 SI
ε 0 ?	ε 0 1 SI	ε 1 SI
ε 0 0 ?	ε $\underset{\forall}{0}$ SI	ε SI
ε 0 0 0, NO	ε 1 ?	FINE
ε $\underset{\exists}{0}$ 0, NO	ε 1 0 ?	L'ALGORITMO È DETERMINISTICO,
ε 0 1 ?	ε 1 $\underset{\forall}{0}$ 0 ? $\Rightarrow$ SI	OCCUPA SPAZIO $t(n)$, TEMPO
ε 0 1 0 SI	ε 1 0 1 ? $\Rightarrow$ SI	$o(1)^{t(n)}$ $\boxed{OK}$

ASpace(s) incluso DTime(c^s)

PROP.: CON UN RAGIONAMENTO SIMILE, SI DIMOSTRA CHE $ASPACE(s(n)) \subseteq DTIME(o($

DIM.: $A \in ASPACE(s(n))$, G_w, INPUT w, G_w HA NODI DI LUNGH. $\leq s(n)$ E HA $\leq o(1)^{s(n)}$ VERTICI. DI QUI, COME SOPRA. $\boxed{OK}$

PROP.: $NSPACE(s(n)) \subseteq ATIME(s(n)^2)$.   **NSpace(s) = ATime(s^2)**

DIM.: SIA $N \in NSPACE(S(n))$, G_w = GRAFO DELLE COMPUTAZIONI DI N SU INPUT w.
$N(w)$ ACCETTA $\iff \exists$ CAMMINO NEL GRAFO DA C_w INIZIALE A UNA ACCETTANTE
(INFATTI TUTTI GLI STATI SONO $\exists$, QUINDI BASTA UN CAMMINO).

G_w HA $O(1)^{S(n)}$ VERTICI; LA MACCHINA UTILIZZA UNA SUBROUTINE
$P(d,x,y)$ T.C. $d \in \mathbb{N}$, $x,y \in G_w$ T.C.: $P(d,x,y) \iff \exists$
CAMMINO DA x A y IN G_w LUNGO AL PIÙ 2^d PASSI. OSSERVIAMO CHE:

guess middle

$P(0,x,y) \iff E(x,y)$ (OPPURE $x=y$);

$P(d+1,x,y) \iff \exists z$ T.C. $[P(d,x,z) \wedge P(d,z,y)]$ $(\forall d)$.

L'ALGORITMO ALLORA PROCEDE:

$\boxed{d|x|y} \Rightarrow d\ x\ y\ (\exists) \Rightarrow d\ x\ y\ z\ (\forall) \Rightarrow \begin{cases} d\ x\ y\ z\ 1 \\ d\ x\ y\ z\ 2 \end{cases}$

in stato esistenziale scelgo z, in stato universale uno dei due congiunti

$d\ x\ y\ z\ 1 \longrightarrow (d-1)\ x\ z\ ?$; $d\ x\ y\ z\ 2 \longrightarrow (d-1)\ z\ y\ ?$.

E IL FATTO DI AVERE UNA MACCHINA ALTERNANTE PERMETTE DI CALCOLARE
$\forall$ E $\exists$. QUANTO TEMPO OCCORRE?
L'INIZIALIZZAZIONE AVVIENE CON: x = CONF. INIZ., y = CONF. FIN.,
$d = c \cdot S(n)$. SIA $T(d)$ = TEMPO PER CALCOLARE $P(d,x,y)$ SU
UNA ATM; $T(d)$ = TEMPO PER SCRIVERE z + TEMPO PER CALCOLARE $T(d-1)$.
OSSIA: $O(S(n)) + T(d-1)$ CHE ALLA FINE DÀ $S(n)^2$. $\boxed{OK}$

PROP.: $DTIME(O(1)^{S(n)}) \subseteq ASPACE(S(n))$ (ABBIAMO GIÀ DIMOSTRATO L'INCLUSIONE OPPOSTA, QUINDI VALE $=$)
DIM.: POSPOSTA.

ABBIAMO DUNQUE:

$NSPACE(S(n)) \subseteq ATIME(S(n)^2)$

$ATIME(t(n)) \subseteq DSPACE(t(n))$

SE NE DEDUCE:

$APTIME = DPSPACE$ (PARALLEL TIME) $= NPSPACE$,
$APTIME = \bigcup_k ATIME(n^k)$, $DPSPACE = \bigcup_k DSPACE(n^k)$

Poly-hierarchy

$APSPACE = DEXPTIME$; $P = ASPACE(\log n)$
$APSPACE = \bigcup_k ASPACE(n^k)$, $DEXPTIME = \bigcup_k TIME(2^{(n^k)})$

NUM. RISULTATO DI ALTERNANZE

(POLYNOMIAL HIERARCHY) $PH = \bigcup_{k,\ell} ATIME\text{-}ALT(n^k, \ell)$ (LOGICA DEL 2° ORDINE)

$P \subseteq NP \subseteq PH \subseteq AP \subseteq EXPTIME$; $P \neq EXPTIME$, IL RESTO E' UN PROBLEMA APERTO.

$AP = APSPACE$

$LOGSPACE \subseteq P \subseteq AP$ E $LOGSPACE \neq AP$.

Inclusioni tra le principali classi

PROP.: $DTIME(K^{S(n)}) \subset ASPACE(S(n))$ $\left(= DTIME\left(O(1)^{S(n)}\right)\right)$

DIM.: DATA $M \in DTIME(K^{S(n)})$, DATO UN INPUT w, VOGLIAMO SAPERE SE $M(w)=SI$ USANDO SPAZIO $S(n)$ IN UNA MACCHINA ALTERNANTE.

G_w^M (GRAFO DI M SU INPUT w) E' UN GRAFO LINEARE: $C_0, C_1, C_2, \ldots, C_{K^{S(n)}}$

OGNI C_i E' IL CONTENUTO DEL NASTRO, LO STATO E LA POSIZIONE DELLA TESTINA.

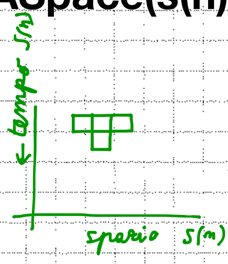
IL TUTTO SI PUÒ RAPPRESENTARE IN UNA MATRICE $K^{S(n)} \times K^{S(n)}$ (TEMPO X SPAZIO)

UNA CONFIG. C_i E' UNA LISTA IL CUI n-ESIMO ELEMENTO ~~...~~ E' IL CONTENUTO DELLA n-ESIMA CELLA DI MEMORIA, A MENO CHE IN QUELLA CONFIG. LA TESTINA SI TROVI NELLA n-ESIMA POSIZIONE: IN QUEL CASO, L'ELEMENTO n-ESIMO (S_n) E' UGUALE A < CONT. CELLA, STATO MACCHINA >.

STATI = $q_0, \ldots, q_n$
ALFABETO = 0, 1

0	1	q_1 0	1	1	0	0

DTime(k^s(n)) = ASpace(s(n))



COME GESTIRE UNA MATRICE $K^{S(n)} \times K^{S(n)}$ IN ASPACE?

$C_0 =$ | w_0 | w_1 | w_2 | ... | w_n | (q_0 su w_0) ; $C_{FINALE} =$ | 0 FINALE | 0 | ... | 0 | (STATO ACCETTANTE)

IN OGNI C_i, IL CONTENUTO DI OGNI CASELLA E' DETERMINATO DAL CONTENUTO DELLE 3 CHE "LE STANNO SOPRA":

SIA $C(t, p, a) \iff$ AL TEMPO t IN POS. p C'E' IL SIMBOLO a.

$C(t, p, a)$ E' DETERMINATO DA $C(t-1, p', a')$, CON $p' \in \{p-1, p, p+1\}$

$$C(0, p, a) \iff \begin{cases} a = w_p, & p > 0 \\ a = \langle w_0, q_0 \rangle, & p = 0 \end{cases} ;$$

$C(t+1, p, a) \iff \exists\, a_{-1}, a_0, a_1$ T.C. $\left((a_{-1}, a_0, a_1) \xrightarrow{M} a \wedge \forall i \in \{-1, 0, 1\}\ C(t, p+i, a_i)\right)$. INFATTI:

$M = \{$ MEMORIA X STATO $\longrightarrow$ MEMORIA X STATO X SPOSTAMENTO

(ES.: $M(0, q) = (1, q', -1)$). COME FARE TUTTO CIÒ CON UNA ATM?

SCRIVIAMO UN "ALGORITMO" ALTERNANTE PER CALCOLARE $C(t+1, p, a)$:

- PASSO IN UNO STATO (∃): NON DETERMINISTICAMENTE SCELGO a_{-1}, a_0, a_1;
- PASSO IN UNO STATO (∀): CONTROLLO SE $(a_{-1}, a_0, a_1) \xrightarrow{M} a$
- SCELGO NON DET. $i \in \{-1, 0, 1\}$; RIPARTO CON $t, p+i, a_i$

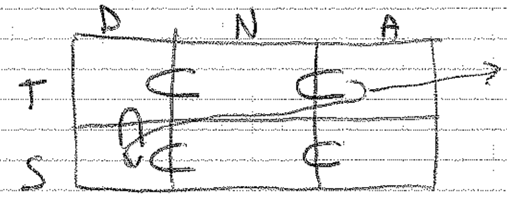
IN PRATICA, SI COSTRUISCE UNA MTA CHE SI COMPORTA COSÌ:

∃ t, p, a → $t-1, p, a, a_{-1}, a_0, a_1$ ∀(a_{-1}, a_0, a_1), CONTROLLA SE a_{-1}, a_0, a_1 DETERMINANO a, SENNÒ LI SCARTA E RIPETE; SE SONO BUONI, SCEGLIE $i \in \{-1, 0, 1\}$ E CONTROLLA: -1: $\boxed{t, p-1, a_{-1}}$ STATO ∃; 0: ECC.; 1: ECC. E CONTINUA...

QUANTO SPAZIO OCCUPA?

$t \le K^{S(n)}$, $p \le K^{S(n)}$, $\Rightarrow |t| \le O(S(n))$, $|p| \le O(S(n))$ E LA MEMORIA OCCUPATA DAGLI a_i È PICCOLA. $\boxed{OK}$

RICORDIAMO:



(ATIME ⊂ DSPACE)

INOLTRE, $ASPACE(S(n)) = DTIME(K^{S(n)})$.

ALLORA,
$P (= TIME(n^{O(1)})) = ASPACE(\log n)$

P=AL

SIANO FO = FIRST ORDER, LFP = LEAST FIXING POINT;

VOGLIAMO DIMOSTRARE CHE $FO(LFP) = P$.

L = LINGUAGGIO, $L = \{E^{(2)}\}$, $G = (V, E)$, $E \subset V^2$.

E^* = CHIUSURA TRANSIT. DI E; $E^*(x,y) \iff x = y \vee (\overset{\text{PLEONASTICO}}{E(x,y) \vee})\ \exists z$ T.C. $[E(x,z) \wedge E^*(z,y)]$. E^* DEVE ESSERE LA MINIMA RELAZIONE CHE SODDISFA CIÒ. COME ESPRIMERLO? PER OGNI R, DEFINISCO:

$\varphi(R, x, y) \iff x = y \vee \exists z\, (E(x,z) \wedge R(z,y))$. VOGLIO TROVARE R T.C. $\varphi(R) = R$. QUESTO DETERMINA UNA FUNZIONE $R \to R'$;

$f_{\varphi, R, x, y}$

$R' = \{(x,y) : \varphi(R, x, y)\}$.

f È ~~MONOTONA~~ CRESCENTE: $R \subset S \Rightarrow f(R) \subset f(S)$. $f : \mathcal{P}(V^2) \to \mathcal{P}(V^2)$.

PER IL TEOR. DI KNASTER-TARSKI, $f: \mathcal{P}(U) \to \mathcal{P}(U)$ MONOTONA CRESCENTE, U FINITO $\Rightarrow$ f HA UN MINIMO PUNTO FISSO (UNICO).

(DIM.: $\emptyset \subset f(\emptyset) \subset f(f(\emptyset)) \subset \dots$ E PRIMA O POI SI FERMA: $f^k(\emptyset) = f^{k+1}(\emptyset)$ CON $k \leq |U|$; $R = f^k(U)$ E' PUNTO FISSO ED E' MINIMO PERCHE' SE $f(S) = S$, ALLORA $\forall k$ $f^k(\emptyset) \subset S$ E SI FA PER INDUZIONE: $f^0(\emptyset) = \emptyset \subset S$, E SE $f^k(\emptyset) \subset S \Rightarrow f^{k+1}(\emptyset) \subset f(S) = S$. $\boxed{OK}$)

FISSATA LA STRUTTURA $(G = (V, E))$, $f_{\varphi, R, x, y} : \mathcal{P}(V^2) \to \mathcal{P}(V^2)$ HA MINIMO PUNTO FISSO; QUINDI $E^*(a, b) = (LFP_{\varphi, R, x, y})(a, b)$.

DIMOSTRIAMO ORA CHE $FO(LFP) = P$. DATA $\varphi \in FO(LFP)_L$ CHIUSA, $MOD(\varphi) = \{ A \mid A \models \varphi \} \subset L$-STRUTTURE. TRAMITE CODIFICA BINARIA, UN INSIEME DI STRUTTURE DIVENTA UN INSIEME DI STRINGHE, QUINDI $MOD(\varphi) \subset \{$STRINGHE DI $0,1\}$

QUESTO DA' UNA BIGEZIONE FRA P E $FO(LFP)$.

<u>ES.</u>: $G = (V, E, \underset{\text{SOURCE}}{s}, \overset{\text{TARGET}}{t})$; OUTPUT: C'E' UN CAMMINO DA s A t? $n = |V|$

PRENDIAMO $\varphi = E^*(s, t)$ E RISPONDIAMO "SI'". $MOD(\varphi) = \{ G : \text{OUTPUT} = \text{SI} \}$

QUESTO E' IL PROBLEMA REACH $= \{ (V, E, s, t) : E^*(s, t) \}$.

$REACH_{ALT} = \{ (V, E^2, G^1_\forall, G^1_\exists, s, t) : E^*_{ALT}(s, t) \}$ ($E^*_{ALT}(s,t)$ = RAGGIUNGIBILE SU UN GRAFO ALTERNANTE). SI HA: $REACH_{ALT} \in PTIME$.

<u>ESERC.</u>: ESPRIMERE E^*_{ALT} COME PUNTO FISSO DI QUALCOSA.

Problema Reach-Alt

p26-27 guess middle

$$\begin{array}{c} ATime(t^2) \\ \cup \end{array}$$

$$TIME(t) \subset NTime(t) \subset ATime(t) \subset DSpace(t) \subset NSpace(t) \subset ASpace(t) =$$

$$= Time(O(1)^t)$$

p26
visito Gw con backtracking

$\subseteq$ p26
visito Gw con backtracking

$\supseteq$ p.30
uso la località delle computazioni

$$LSpace \subset ALSpace = P \underset{\neq}{\subset} ExpTime = APSpace$$

LEGGE 0-1 PER IL 1° ORDINE

DATO UN VOCABOLARIO τ, $\dfrac{|\{A\ \tau\text{-STRUTTURE}: A\models\varphi,\ |A|=n\}|}{|\{A\ \tau\text{-STRUTTURE}: |A|=n\}|} \xrightarrow{n\to\infty} 0$ o 1

DATO L, A, B L-STRUTTURE, $\mathcal{G}_k^m$ (GIOCO CON k PEDINE E m MOSSE)

DUE GIOCATORI, DALILA E SANSONE; DALILA VINCE SE, DATI
$\exists$ $\forall$

$a_0, \dots, a_\ell \in A$, $b_0, \dots, b_\ell \in B$, $\forall i$, LE STRUTTURE SONO ISOMORFE PER L'ISOM. CHE MANDA $a_i \to b_i$. $(A \sim_m^k B)$

SIA $A \equiv_m^k B \iff \left(\forall \varphi \in \mathcal{L}_m^k,\ A\models\varphi \iff B\models\varphi\right)$.

TEOR.: D VINCE $\iff A\equiv_m^k B$.

SE $\forall k$ $A\sim_m^k B$, ALLORA $A\sim_m B$; ANALOGAMENTE, SE $\forall m$, $A\sim_m^k B$ ALLORA $A\sim^k B$.

FORMULA DI ESTENSIONE

$\gamma_k = (\exists x_1, \dots, x_{k-1}$ "DIVERSI"$) \wedge (\forall x_1, \dots, x_{k-1}$ "DIVERSI" $(\exists x_k$ "DIVERSO DAGLI ALTRI" $(E(x_1, x_k) \wedge \dots \wedge E(x_{k-1}, x_k)) \wedge (\exists x_k$ $(E(x_1, x_k) \wedge \dots \wedge \neg E(x_{k-1}, x_k)) \wedge \dots \wedge \exists x_k : (E(x_1, x_2) \wedge \neg E(x_2, x_3) \wedge \dots)))$

($\exists x_k$ COLLEGATO SOLO AI PRIMI 2 VERTICI, x_k COLLEGATO SOLO AI PRIMI 3 ECC.).

$\mu_n = \dfrac{|\{G : |G|=n \quad G\models\gamma_k\}|}{|\{G : |G|=n\}|} \xrightarrow{n\to\infty} 1$. INFATTI:

SIANO $v_1, \dots, v_{k-1}$ VERTICI, x UNO DEGLI ALTRI $n-k+1$ VERTICI, C UNO DEI CONGIUNTI DI γ_k; LA PROB. CHE x NON SODDISFI C E' $\left(1-\left(\frac{1}{2}\right)^{k-1}\right)$; LA PROB. CHE NON VERIFI C, CI SIANO x CHE NON VERIFICHINO C ~~NESSUNO DI ESSI~~ E' α^{n+1-k}.

~~LA PROB. CHE NESSUN x VERIFICHI ALCUN C E'~~

LA PROB. CHE ALMENO UN C NON SIA MAI VERIFICATO E' $\leq k\cdot\alpha^{n+1-k}$.

AL VARIARE DI $v_1, \dots, v_k$ LA PROB. E' $\leq \binom{n}{k} k\,\alpha^{n+1-k} < n^{k-1} k\cdot\alpha^{n+1-k} \to 0$

E QUINDI LA PROB. CHE γ_k SIA VERIFICATA $\to 1$.

LEMMA: G, H GRAFI, $(G \models \gamma_k, H \models \gamma_k) \Rightarrow G \sim^k H$.

DIM: INDUZIONE SU m: $G \sim^k_m H$.

$m=0$: OK A VUOTO.

$m \Rightarrow m+1$: DA UNA STRATEGIA VINCENTE DI $\mathcal{G}^k_m$ SE NE DERIVA UNA PER $\mathcal{G}^k_{m+1}$

LE PRIME m MOSSE SI GIOCANO COME LA STRATEGIA VINCENTE;

PRENDIAMO $v \in G$; $g_1, \dots, g_{k-1}\ v \in G$, $h_1, \dots, h_{k-1} \in H$;

$\exists v' \in H$: $G \models E(g_i, v) \Leftrightarrow H \models E(h_i, v')$. ESISTE PER VIA DI γ_k. $\boxed{OK}$

$\tau < R_1^{a_1}, \dots, R_j^{a_j} >$; $\mathcal{J} = \{$FORMULE ATOMICHE $R(a_{i_1}, \dots, a_{i_m})$ T.C.

$x_k \in \{a_{i_1}, \dots, a_{i_m}\} \subset \{x_1, \dots, x_k\}\}$.

$\gamma_k(\tau) = (\forall x_1, \dots, x_{k-1}$ "DIVERSI"$) \bigwedge_{S \subset \mathcal{J}} (\exists x_k$ "DIVERSO"$)\left(\bigwedge_{\alpha \in S} \alpha \wedge \bigwedge_{\alpha \notin S} \neg\alpha\right)$.

SE

LEMMA: ALLORA VALE: $\mu_n(\gamma_k) = \dfrac{|\{A\ \tau\text{-STRUTTURE } |A|=n\ A \models \gamma_k\}|}{|\{A\ |A|=n\}|}$, $\mu_n(\gamma_k) \xrightarrow{n} 1$.

DIM: $a_1, \dots, a_{k-1} \in A$; $x \notin \{a_1, \dots, a_k\}$; LA PROB. CHE IL COSTRUTTO CONGIUNTO

S NON SIA VERIFICATO È $\alpha = \left(1 - \left(\frac{1}{2}\right)^{|\mathcal{J}|}\right) < 1$; C.S. LA PROB. CHE γ_k NON SIA

VERIFICATA È $\leq \binom{n}{k-1} 2^{|\mathcal{J}|} \alpha^{n+1-k} < n^{k-1} 2^{|\mathcal{J}|} \alpha^{n+1-k} \to 0$. $\boxed{OK}$

LEMMA: A, B τ-STRUTTURE T.C. $A \models \gamma_k \Leftrightarrow B \models \gamma_k$; ALLORA $A \sim^m B$.

DIM: INDUZIONE; $m=0$: OK A VUOTO.

$m \Rightarrow m+1$: $a_1, \dots, a_{k-1}, a \in A$, $b_1, \dots, b_{k-1} \in B$; COME SOPRA,

SCEGLIAMO b T.C., DATA $\varphi \in \mathcal{J}$, $A \models \varphi(a_{i_1}, \dots, a_{i_t})$ $(a \in \{a_{i_1}, \dots, a_{i_t}\})$

$\Leftrightarrow B \models \varphi(b_{i_1}, \dots, b_{i_t})$ $(b \in \{b_{i_1}, \dots, b_{i_t}\})$.

LEGGE 0-1

DATO $S \subset \mathcal{L}^k$; $\mu_n(S) = \dfrac{|\{A\ \tau\text{-STRUTTURE}, |A| = n, \forall \varphi \in S, A \models \varphi\}|}{|\{|A| = n\}|}$

ALLORA $\lim_{n \to \infty} \mu_n(S) \in \{0, 1\}$.

DIM.: $\forall \varphi \in \mathcal{L}^k$, $\gamma_k \vdash \varphi$ OPPURE $\gamma_k \vdash \neg\varphi$ (QUESTO DISCENDE DALL'ULTIMO LEMMA: $A \models \varphi \Leftrightarrow B \models \varphi$).

SE S E' INFINITO, $\{\varphi_l\}_l = S \subset \mathcal{L}^k$, AVREMO $\gamma_k \vdash_{l \in N} \neg\varphi_l$ E $\gamma_k \vdash_{l \in J} \varphi_l$; SE $\exists\, l$: $\gamma_k \vdash \neg\varphi_l$, LA PROB. E' 0, SENNO' E' 1.

$\boxed{\text{OK}}$