$\mathfrak{R} \subseteq \mathfrak{E}$

Eugenio G. Omodeo

> *"It was in the summer of 1959 that Hilary and I really hit the jackpot. We decided to see how far we could get with the approach we had used at the Logic Institute in Ithaca, if, following Julia Robinson's lead, we were willing to permit variable exponents in our Diophantine equations."* [OP16, pp. 15–16]

> *"It wasn't until 2004 that Ben Green and Terence Tao proved that P.A.P. is true, thus validating our work as a complete proof, but only well after the fact."* [Dav10]

Trieste, 11/12.05.2021

Sunto

Il teorema di Davis-Putnam-Robinson asserisce che

$$\mathfrak{R} \subseteq \mathfrak{E}$$

i.e., che ogni insieme *enumerabile ricorsivamente* (in breve ‘*r.e.*’) può essere definito esistenzialmente in termini dell’esponenziazione.

$\|$

Sunto

Il teorema di Davis-Putnam-Robinson asserisce che

$$\mathfrak{R} \subseteq \mathfrak{E}$$

i.e., che ogni insieme *enumerabile ricorsivamente* (in breve ‘*r.e.*’) può essere definito esistenzialmente in termini dell’esponenziazione.

> Ne discende: *Non c’è* algoritmo in grado di stabilire, di ogni equaz. diofantea esponenziale, se abbia o no soluzione su $\mathbb{N}$.

Rilevanza

La dimostrazione di

$$\mathfrak{R} \subseteq \mathfrak{E}$$

permise di ridurre *all’*IPOTESI J.R. circa l’esistenza di relaz. diofantee polinomiali a crescita esponenziale, $\therefore$

$$\mathfrak{E} \subseteq \mathfrak{D}\,,$$

contemporaneamente la ‘DARING HYPOTHESIS’[1] di [Dav53]

$$\mathfrak{R} \subseteq \mathfrak{D}$$

e il 10º di Hilbert.

[1] V. [Mat93, pag. 99]

Scaletta

Enunciato del teorema DPR e come affrontarne la dim.

Teorema (Davis-Putnam-Robinson). Sia

$$g : \quad \mathbb{N}^{m} \longrightarrow \mathbb{N} \cup \{ \perp \}$$

una funzione computabile — anche solo parzialmente. Allora il GRAFO di g, i.e.

$$\mathcal{G}(a_0, a_1, \ldots, a_m) \iff_{\text{Def}} a_0 = g(a_1, \ldots, a_m)$$

è una relazione diofantea esponenziale.

Enunciato del teorema DPR e come affrontarne la dim.

Teorema (Davis-Putnam-Robinson). Sia

$$g : \quad \mathbb{N}^{m} \longrightarrow \mathbb{N} \cup \{ \perp \}$$

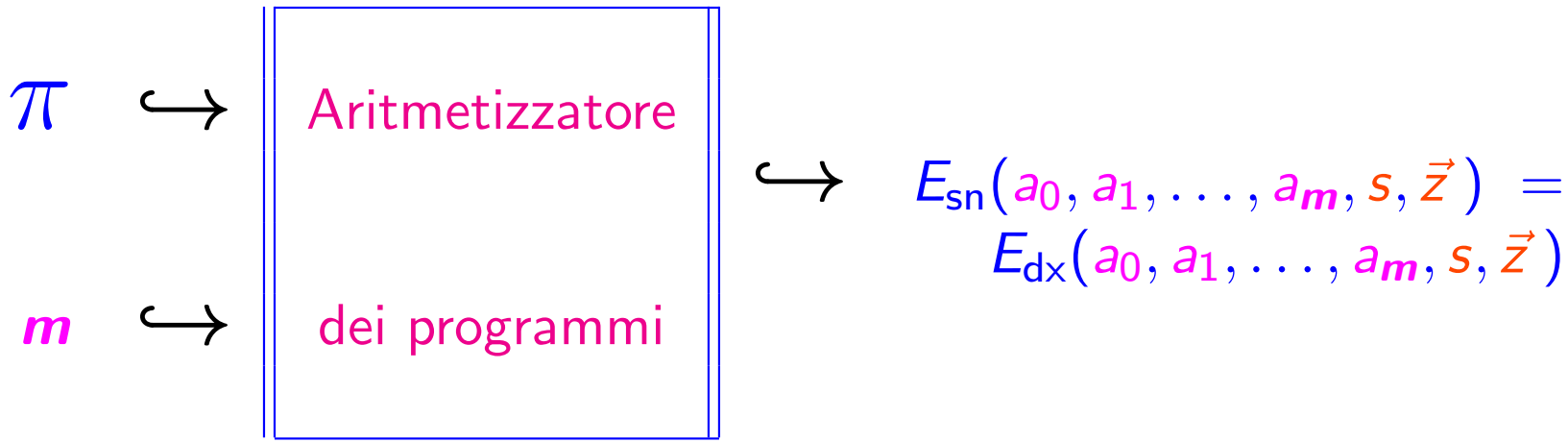
una funzione computabile — anche solo parzialmente. Allora il GRAFO di g, i.e.

$$\mathcal{G}(a_0, a_1, \ldots, a_m) \iff_{\text{Def}} a_0 = g(a_1, \ldots, a_m)$$

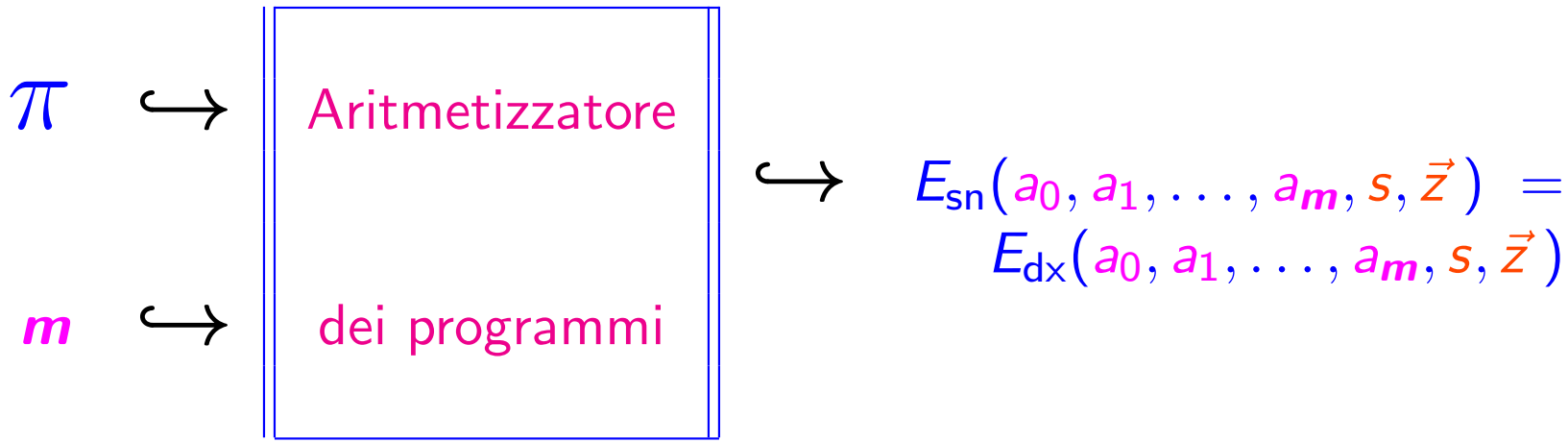
è una relazione diofantea esponenziale.

Dim. (traccia): Per definire esistenzialmente $\mathcal{G}$, specificheremo tramite equazioni diofantee esponenziali il funzionamento di un programma π computante g. ⊣

Costruz. che dimostrerà, alla Jones-Matiyasevich, il DPR

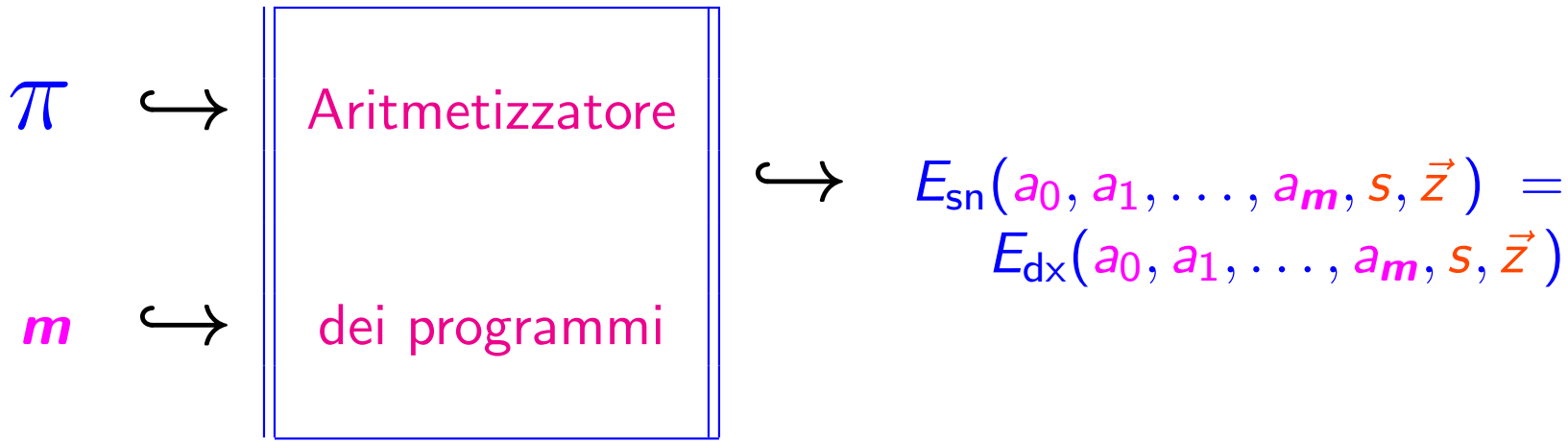


Costruz. che dimostrerà, alla Jones-Matiyasevich, il DPR



L'equazione $E_{sn} = E_{dx}$ avrà soluzione, per $\bigwedge_{i=0}^{m} a_i = \boldsymbol{a_i}$, sse l'esecuzione di π, avviata sui dati $\boldsymbol{a_1}, \ldots, \boldsymbol{a_m}$, giunge a STOP in un numero $\boldsymbol{s} \in \mathbb{N}$ di passi fornendo a conclusione il risultato $\boldsymbol{a_0}$.

Costruz. che dimostrerà, alla Jones-Matiyasevich, il DPR



L'equazione $E_{sn} = E_{dx}$ avrà soluzione, per $\bigwedge_{i=0}^{m} a_i = \boldsymbol{a}_i$, sse l'esecuzione di π, avviata sui dati $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m$, giunge a STOP in un numero $\boldsymbol{s} \in \mathbb{N}$ di passi fornendo a conclusione il risultato $\boldsymbol{a}_0$.

Matiyasevich [Mat74]:

> È evitabile che l'eq. $E_{sn} = E_{dx}$ sia sotto-determinata.

Un linguaggio di programmazione Turing-completo

Programmazione di una macchina a registri –I

Introduciamo un LINGUAGGIO DI PROGRAMMAZIONE che comprende le variabili

$$R_0\,,\ R_1\,,\ R_2\,,\cdots \qquad (\ \textit{ad infinitum}\ )$$

a valori in $\mathbb{N}$.

Programmazione di una macchina a registri –I

Introduciamo un LINGUAGGIO DI PROGRAMMAZIONE che comprende le variabili

$$\mathrm{R}_0\,,\ \mathrm{R}_1\,,\ \mathrm{R}_2\,,\cdots \qquad (\ \textit{ad infinitum}\ )$$

a valori in $\mathbb{N}$.

Vi sono istruzioni di cinque sorte, ove $j, k \in \mathbb{N}$:

$\mathrm{R}_j \leftarrow \mathrm{R}_j + 1$	incremento
$\mathrm{R}_j \leftarrow \mathrm{R}_j - 1$	decremento ($\dot{-}$?)
IF $\mathrm{R}_j = 0$ **GOTO** k	salto condizionato
GOTO k	salto *in*condizionato
STOP	arresto

Programmazione di una macchina a registri –II

Per PROGRAMMA intendiamo una lista $\mathfrak{I}_0, \ldots, \mathfrak{I}_\ell$ d'istruzioni di dette sorte, con $\ell \in \mathbb{N}$, soggetta alle seguenti limitazioni:

-
-
-

Programmazione di una macchina a registri –II

Per PROGRAMMA intendiamo una lista $\Im_0, \ldots, \Im_\ell$ d'istruzioni di dette sorte, con $\ell \in \mathbb{N}$, soggetta alle seguenti limitazioni:

- Quando un'istruzione $\Im_i$ della lista ha la forma

  IF $\mathrm{R}_j = 0$ **GOTO** k oppure la forma **GOTO** k ,

  deve aversi $0 \leqslant k \leqslant \ell$ e $k \neq i+1$; inoltre $\Im_k$ non dev'essere un'istruzione di decremento.

-

-

Programmazione di una macchina a registri –II

Per PROGRAMMA intendiamo una lista $\mathfrak{I}_0, \ldots, \mathfrak{I}_\ell$ d'istruzioni di dette sorte, con $\ell \in \mathbb{N}$, soggetta alle seguenti limitazioni:

-
- Ogni istruzione $\mathrm{R}_j \leftarrow \mathrm{R}_j - 1$ di decremento dev'essere immediatamente preceduta da un 'istruzione

  IF $\mathrm{R}_j = 0$ **GOTO** k , dove il numero k è tenuto, ovviamente, a soddisfare le restrizioni del punto precedente.

-

Programmazione di una macchina a registri –II

Per PROGRAMMA intendiamo una lista $\Im_0, \ldots, \Im_\ell$ d'istruzioni di dette sorte, con $\ell \in \mathbb{N}$, soggetta alle seguenti limitazioni:

- Quando un'istruzione $\Im_i$ della lista ha la forma

  IF $\mathrm{R}_j = 0$ **GOTO** k oppure la forma **GOTO** k ,

  deve aversi $0 \leqslant k \leqslant \ell$ e $k \neq i+1$; inoltre $\Im_k$ non dev'essere un'istruzione di decremento.

- Ogni istruzione $\mathrm{R}_j \leftarrow \mathrm{R}_j - 1$ di decremento dev'essere immediatamente preceduta da un 'istruzione

  IF $\mathrm{R}_j = 0$ **GOTO** k , dove il numero k è tenuto, ovviamente, a soddisfare le restrizioni del punto precedente.

- $\Im_\ell$ è una **STOP**, l'unica che compare nella lista.

Programmazione di una macchina a registri –III

Un programma del genere può essere utilizzato per COMPUTARE una funzione

$$g : \quad \mathbb{N}^{m} \quad \rightharpoonup \quad \mathbb{N},$$

totale o parziale (di qui la mezza freccia), ad m operandi, come segue:

-

Programmazione di una macchina a registri –III

Un programma del genere può essere utilizzato per COMPUTARE una funzione

$$g : \quad \mathbb{N}^{m} \quad \rightharpoonup \quad \mathbb{N},$$

totale o parziale (di qui la mezza freccia), ad m operandi, come segue:

- nelle variabili $R_1, \ldots, R_m$ vengono inizialmente impostati i rispettivi valori $a_1, \ldots, a_m$ degli OPERANDI;

Programmazione di una macchina a registri –III

Un programma del genere può essere utilizzato per COMPUTARE una funzione

$$g : \quad \mathbb{N}^{m} \quad \rightharpoonup \quad \mathbb{N},$$

totale o parziale (di qui la mezza freccia), ad m operandi, come segue:

-
- le altre var.: R_0, $R_{m+1}, \ldots, R_r$ vengono inizialm. poste a 0;

Programmazione di una macchina a registri –III

Un programma del genere può essere utilizzato per COMPUTARE una funzione

$$g : \quad \mathbb{N}^{m} \quad \rightharpoonup \quad \mathbb{N} ,$$

totale o parziale (di qui la mezza freccia), ad m operandi, come segue:

-
- al programma viene dato avvio sull'istruzione $\mathfrak{S}_0$;

Programmazione di una macchina a registri –III

Un programma del genere può essere utilizzato per COMPUTARE una funzione

$$g : \quad \mathbb{N}^{m} \quad \rightharpoonup \quad \mathbb{N},$$

totale o parziale (di qui la mezza freccia), ad m operandi, come segue:

- nelle variabili $R_1, \ldots, R_m$ vengono inizialmente impostati i rispettivi valori $a_1, \ldots, a_m$ degli OPERANDI;
- le altre var.: R_0, $R_{m+1}, \ldots, R_r$ vengono inizialm. poste a 0;
- al programma viene dato avvio sull'istruzione $\mathfrak{I}_0$;
- *ecc. ecc.*;

Programmazione di una macchina a registri –IV

- se e quando il programma giunge all'istruzione **STOP**, si prende come RISULTATO $g(a_1, \ldots, a_m)$ il valore conservato nella variabile R_0;

Programmazione di una macchina a registri –IV

- se e quando il programma giunge all'istruzione **STOP**, si prende come RISULTATO $g(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m)$ il valore conservato nella variabile R_0;

- (g non associa alcun valore alla $\boldsymbol{m}$-upla $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m$ sse, quando avviato su tali valori, il programma prosegue per sempre).

Terminazione 'pulita' di un progr. della macchina a registri

Esercizio. Mostrare che se c'è un programma π che computa una certa funzione g, allora ce n'è uno che computa la stessa g e che, quando (e se) termina, lascia a 0 tutte le variabili distinte dalla R_0.

Terminazione 'pulita' di un progr. della macchina a registri

Esercizio. Mostrare che se c'è un programma π che computa una certa funzione g, allora ce n'è uno che computa la stessa g e che, quando (e se) termina, lascia a 0 tutte le variabili distinte dalla R_0.

Nota Bene. Nel seguito considereremo un programma che termina a questo modo.

Confronto fra ‘dialetti’ di Shepherdson–Sturgis

Esercizio. Mostrare che l’aggiunta di una nuova sorta d’istruzione, la

$$\textbf{IF} \quad R_j \neq 0 \quad \textbf{GOTO} \quad k \ ,$$

(di ovvio significato) non aumenterebbe il potere espressivo del linguaggio di programmazione e che anche l’istruzione di salto incondizionato è—noto che sia ***m***—eliminabile.

Come tradurre in $\mathfrak{E}$ un programma π, quale ad es.

$\mathfrak{S}$	*Prodotto di due numeri*
0	IF $R_2 = 0$ GOTO 11
1	$R_2 \leftarrow R_2 - 1$
2	IF $R_1 = 0$ GOTO 7
3	$R_1 \leftarrow R_1 - 1$
4	$R_3 \leftarrow R_3 + 1$
5	$R_0 \leftarrow R_0 + 1$
6	GOTO 2
7	IF $R_3 = 0$ GOTO 0
8	$R_3 \leftarrow R_3 - 1$
9	$R_1 \leftarrow R_1 + 1$
10	GOTO 7

11	IF $R_1 = 0$ GOTO 14
12	$R_1 \leftarrow R_1 - 1$
13	GOTO 11
$\ell = 14$	STOP

$m = 2$,
$r = 3$

while $R_2 > 0$
  $R_2 --$
  while $R_1 > 0$
    $R_1 --$
    $R_3 ++$
    $R_0 ++$
  while $R_3 > 0$
    $R_3 --$
    $R_1 ++$

Emulazione di un programma tramite equazioni esponenziali

Incognite del sistema in cui tradurremo π

s : Numero di passi (ove $< \infty$) che precedono STOP

$\mathfrak{r}_j$: Decorso dei valori di ciascuna var. di programma R_j

▶

$\mathfrak{l}_i$: Tracciato delle attivazioni di ciascuna istruzione $\mathfrak{I}_i$

▶

Incognite del sistema in cui tradurremo π

s : Numero di passi (ove $< \infty$) che precedono STOP

$\mathfrak{r}_j$: Decorso dei valori di ciascuna var. di programma R_j

- i.e., la sequenza $\mathfrak{r}_{j,0}, \ldots, \mathfrak{r}_{j,s}$ costituita dall'iniziale $\mathfrak{r}_{j,0}$ e dai susseguenti valori $\mathfrak{r}_{j,t}$ della R_j, ove $\mathfrak{r}_{j,t}$ è il valore súbito dopo l'esecuzione del t-esimo passo.

$\mathfrak{l}_i$: Tracciato delle attivazioni di ciascuna istruzione $\mathfrak{S}_i$

-

Incognite del sistema in cui tradurremo π

s : Numero di passi (ove $< \infty$) che precedono STOP

$\mathfrak{r}_j$: Decorso dei valori di ciascuna var. di programma R_j

- i.e., la sequenza $\mathfrak{r}_{j,0}, \ldots, \mathfrak{r}_{j,s}$ costituita dall'iniziale $\mathfrak{r}_{j,0}$ e dai susseguenti valori $\mathfrak{r}_{j,t}$ della R_j, ove $\mathfrak{r}_{j,t}$ è il valore súbito dopo l'esecuzione del t-esimo passo.

$\mathfrak{l}_i$: Tracciato delle attivazioni di ciascuna istruzione $\mathfrak{S}_i$

- i.e., la sequenza $\mathfrak{l}_{i,0}, \ldots, \mathfrak{l}_{i,s}$ di 0 / 1 che soddisfa $\mathfrak{l}_{i,t} = 1$ in corrispondenza di quei t per cui $\mathfrak{S}_i$ è l'istruzione eseguita al $t+1$-esimo passo di π.

Incognite del sistema in cui tradurremo π

s : Numero di passi (ove $< \infty$) che precedono STOP

$\mathfrak{r}_j$: Decorso dei valori di ciascuna var. di programma R_j

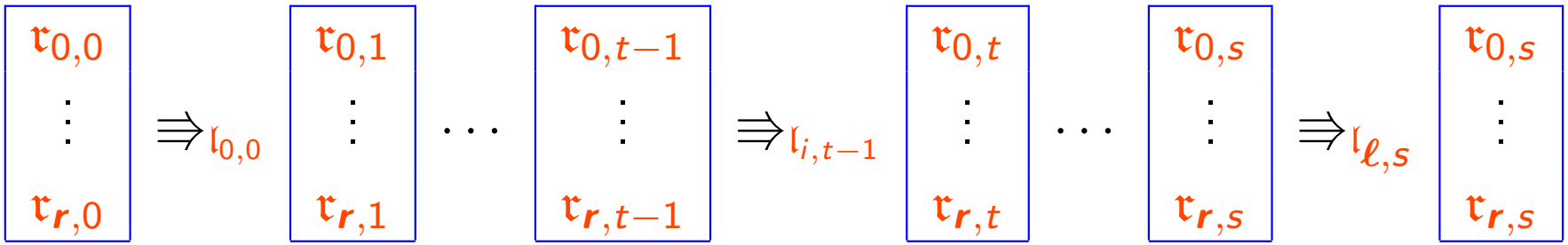
- i.e., la sequenza $\mathfrak{r}_{j,0}, \ldots, \mathfrak{r}_{j,s}$ costituita dall'iniziale $\mathfrak{r}_{j,0}$ e dai susseguenti valori $\mathfrak{r}_{j,t}$ della R_j, ove $\mathfrak{r}_{j,t}$ è il valore súbito dopo l'esecuzione del t-esimo passo.

$\mathfrak{l}_i$: Tracciato delle attivazioni di ciascuna istruzione $\mathfrak{I}_i$

- i.e., la sequenza $\mathfrak{l}_{i,0}, \ldots, \mathfrak{l}_{i,s}$ di 0 / 1 che soddisfa $\mathfrak{l}_{i,t} = 1$ in corrispondenza di quei t per cui $\mathfrak{I}_i$ è l'istruzione eseguita al $t+1$-esimo passo di π.

Come rappresentare ognuna di queste seq. tramite un numero ?

La STOP fa 'quadrare i conti' fra decorsi e tracciati



Solo per una i vale $\mathfrak{l}_{i,t-1} \neq 0$, a ciascun t

Un’altra incognita del nostro sistema

Q ✎ sarà la BASE prescelta di un sistema di numerazione posizionale a cifre **molto** ‘spaziose’

Un’altra incognita del nostro sistema

Q  sarà la BASE prescelta di un sistema di numerazione posizionale a cifre **molto** ‘spaziose’

Questa ci consentirà di assimilare ogni componente di ogni $\mathfrak{r}_j$ a una cifra del sistema di numerazione a base Q .

Un'altra incognita del nostro sistema

Q ✎ sarà la BASE prescelta di un sistema di numerazione posizionale a cifre **molto** 'spaziose'

Questa ci consentirà di assimilare ogni componente di ogni $\mathfrak{r}_j$ a una cifra del sistema di numerazione a base Q .

Cosí potremo riferirci, semplicemente, con

$\mathfrak{r}_j$: a quel numero che in base Q risulta espresso dalla seq. di *cifre* ☞ $\mathfrak{r}_{j,s} \cdots \mathfrak{r}_{j,1}\, \mathfrak{r}_{j,0}$

Un'altra incognita del nostro sistema

Q ✎ sarà la BASE prescelta di un sistema di numerazione posizionale a cifre **molto** 'spaziose'

Questa ci consentirà di assimilare ogni componente di ogni $\mathfrak{r}_j$ a una cifra del sistema di numerazione a base Q .

Cosí potremo riferirci, semplicemente, con

$\mathfrak{r}_j$: a quel numero che in base Q risulta espresso dalla seq. di *cifre* ☞ $\mathfrak{r}_{j,s} \cdots \mathfrak{r}_{j,1}\, \mathfrak{r}_{j,0}$

$\mathfrak{l}_i$: al numero analogam. espresso da $\mathfrak{l}_{i,s} \cdots \mathfrak{l}_{i,1}\, \mathfrak{l}_{i,0}$

Altra incognita ancora (foriera di sotto-determinazione?)

Per comodità, sia Q una potenza positiva del numero 2

Altra incognita ancora (foriera di sotto-determinazione?)

Per comodità, sia Q una potenza positiva del numero 2

Quest'ultimo criterio tira in ballo un altro valore incognito:
quel numero $\flat$ tale che $Q = 2 \cdot 2^{\flat}$

Altra incognita ancora (foriera di sotto-determinazione?)

Per comodità, sia Q una potenza positiva del numero 2

Quest’ultimo criterio tira in ballo un altro valore incognito:
quel numero $\flat$ tale che $Q = 2 \cdot 2^{\flat}$

(È facile rendere $\flat$ unica tramite un apposito ulteriore vincolo)

$2^{\flat} \nmid (2 a_1 + \cdots + 2 a_m + 2 s) \wedge \max(\ell + 1)$

$6 + r + \ell$ incognite assoggettate a 4 o 5 vincoli piú...

s : Numero dei passi che precedono STOP (se è finito)

$\mathfrak{r}_j$: Decorso dei valori di ciascuna var. di programma R_j

$\mathfrak{l}_i$: Tracciato delle attivazioni di ciascuna istruzione $\mathfrak{S}_i$

$Q, \mathfrak{b}, \mathrm{I}$: Valori tali che

$$2\,(\,a_1 + \cdots + a_m + s\,) \;<\; Q \;=\; 2 \cdot 2^{\mathfrak{b}} \;>\; \ell + 1$$

$$1 + (Q-1)\,\mathrm{I} \;=\; Q^{s+1}$$

dove $a_1, \ldots, a_m$ ed a_0 rappresentano i valori somministrati a π e il corrispettivo risultato.

Q ✎ la BASE prescelta, a cifre **molto** 'spaziose'

Evoluz. di stato della memoria : $\ldots 2\,\boldsymbol{r}+2$ vincoli piú...

$$\begin{pmatrix} \mathfrak{r}_0 \\ \mathfrak{r}_1 \\ \vdots \\ \mathfrak{r}_{\boldsymbol{m}} \\ \mathfrak{r}_{\boldsymbol{m}+1} \\ \vdots \\ \mathfrak{r}_{\boldsymbol{r}} \end{pmatrix} = Q \left(\begin{pmatrix} \mathfrak{r}_0 \\ \mathfrak{r}_1 \\ \vdots \\ \mathfrak{r}_{\boldsymbol{m}} \\ \mathfrak{r}_{\boldsymbol{m}+1} \\ \vdots \\ \mathfrak{r}_{\boldsymbol{r}} \end{pmatrix} + \Delta \begin{pmatrix} \mathfrak{l}_0 \\ \mathfrak{l}_1 \\ \vdots \\ \mathfrak{l}_{\ell} \end{pmatrix} \right) + \begin{pmatrix} -Q^{s+1} a_0 \\ a_1 \\ \vdots \\ a_{\boldsymbol{m}} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

e $\boxed{\mathfrak{r}_j \sqsubseteq (\lfloor Q/2 \rfloor - 1)\,\mathrm{I}}$ per ogni j, ove

$$\Delta_{j,i} \quad =_{\mathrm{Def}} \quad \begin{cases} 0 & \text{quando } \Im_i \text{ non modifica } \mathrm{R}_j\,, \textit{ altrim.} \\ \pm 1 & \text{a seconda che } \Im_i \text{ sia } \mathrm{R}_j \leftarrow \mathrm{R}_j \pm 1\,. \end{cases}$$

Flusso di controllo : $\ldots 3\,\ell + 3$ vincoli (al piú)

$1 \sqsubseteq \mathfrak{l}_0, \qquad \mathfrak{l}_\ell = Q^s,$ $\qquad \mathfrak{l}_i \sqsubseteq \mathrm{I} = \sum_{h=0}^{\ell} \mathfrak{l}_h$ per $i = 0, \ldots, \ell - 1$;

$Q\mathfrak{l}_i \sqsubseteq \mathfrak{l}_{i+1}$ quando $\Im_i$ è di forma $\mathrm{R}_j \leftarrow \mathrm{R}_j \pm 1$;

$Q\mathfrak{l}_i \sqsubseteq \mathfrak{l}_k$ quando $\Im_i$ è di forma **GOTO** k ;

$Q\mathfrak{l}_i \sqsubseteq \mathfrak{l}_{i+1} + \mathfrak{l}_k$ quando $\Im_i$ è di forma **IF** $\mathrm{R}_j = 0$ **GOTO** k ;

$\boxed{Q\mathfrak{l}_i \sqsubseteq \mathfrak{l}_{i+1} + Q\,\mathrm{I} - 2\,\mathfrak{r}_j}$ quando $\Im_i$ è di forma **IF** $\mathrm{R}_j = 0$ **GOTO** k .

Flusso di controllo : $\ldots 3\,\ell + 3$ vincoli (al piú)

$1 \sqsubseteq \mathfrak{l}_0, \qquad \mathfrak{l}_\ell \sqsubseteq Q^s,$ $\qquad \mathfrak{l}_i \sqsubseteq \mathrm{I} = \sum_{h=0}^{\ell} \mathfrak{l}_h$ per $i = 0, \ldots, \ell - 1$;

$Q\mathfrak{l}_i \sqsubseteq \mathfrak{l}_{i+1}$ quando $\Im_i$ è di forma $\mathrm{R}_j \leftarrow \mathrm{R}_j \pm 1$;

$Q\mathfrak{l}_i \sqsubseteq \mathfrak{l}_k$ quando $\Im_i$ è di forma **GOTO** k ;

$Q\mathfrak{l}_i \sqsubseteq \mathfrak{l}_{i+1} + \mathfrak{l}_k$ quando $\Im_i$ è di forma **IF** $\mathrm{R}_j = 0$ **GOTO** k ;

$\boxed{Q\mathfrak{l}_i \sqsubseteq \mathfrak{l}_{i+1} + Q\,\mathrm{I} - 2\,\mathfrak{r}_j}$ quando $\Im_i$ è di forma **IF** $\mathrm{R}_j = 0$ **GOTO** k .

Richiamo sulla definibilità esist-/espon-enziale di $a \sqsubseteq b$

Dobbiamo esprimere che $\binom{b}{a}$ è dispari.[2]

Notiamo che $u > \binom{b}{h}$ quando $u > 2^b$.

Per il *teorema binomiale*:

$$(1+u)^b = u^{a+1} \sum_{h=a+1}^{b} \binom{b}{h} u^{h-a-1} + \binom{b}{a} u^a + \sum_{h=0}^{a-1} \binom{b}{h} u^h \,.$$

Richiediamo che

$$2x+1 = \left(\left\lfloor (u+1)^b / u^a \right\rfloor \,\%\, u\right) \quad \& \quad u = 2^b + 1 \,.$$

[2] Grazie al corollario di Lucas del teor. di Kummer.

Senso di $1 + (Q-1)\,\mathrm{I} = Q^{s+1}$, alla luce di $Q = 2^{\flat+1}$

Rispetto a qualsiasi base $Q > 1$ (anche $Q = 10$), il solo modo di soddisfare

$$(Q-1)\,\mathrm{I} \;=\; Q^{s+1} - 1$$

è di prendere

$$\mathrm{I} \;=\; \overbrace{1\ldots1}^{s+1 \text{ volte}} \quad \left(= \sum_{t=0}^{s} Q^t\right),$$

purché gli 1 siano pesati in accordo con Q.

In base 2, poiché stiamo imponendo $Q = 2^{\flat+1}$, I si espande:

$$\mathrm{I} \;=\; \overbrace{\underbrace{0\ldots0}_{\flat}1\ldots\underbrace{0\ldots0}_{\flat}1}^{s+1 \text{ volte}}.$$

Vantaggi che derivano dalla $Q = 2^{\flat+1} \geqslant 2$

Se esprimiamo posizionalmente due numeri u, v come

$$u \;=\; \sum_{i=0}^{K} u_i\, Q^i\,, \qquad v \;=\; \sum_{i=0}^{K} v_i\, Q^i$$

(equiparandone il numero K di cifre Q-arie), allora varrà

$$u \;\sqsubseteq\; v \quad \text{se e solo se} \quad u_i \;\sqsubseteq\; v_i \quad \text{per} \quad i = 0, 1, \ldots, K\,.$$

Senso delle $\mathfrak{r}_0, \quad \mathfrak{r}_1, \ldots, \mathfrak{r}_m, \quad \mathfrak{r}_{m+1}, \ldots, \mathfrak{r}_r$

Grazie alla condiz.

$$2\,(\,a_1 + \cdots + a_m + s\,) \;<\; Q\ ,$$

tenendo conto che il valore $\mathfrak{r}_{j,t}$ che si trova in una variabile R_j a qualsiasi istante[3] t soddisferà

$$\mathfrak{r}_{j,t} \;\leqslant\; \begin{cases} t + a_j & \text{per} \quad j = 1, \ldots, m\ , \\ t & \text{per} \quad j = 0 \text{ e per } j = m+1, \ldots, r\ , \end{cases}$$

possiamo rappresentare in base Q il decorso di R_j come

$$\mathfrak{r}_j \;=\; \textstyle\sum_{t=0}^{s} \mathfrak{r}_{j,t}\, Q^t\ ,$$

dove $\mathfrak{r}_{j,t} < Q/2 = \lfloor Q/2 \rfloor$.

$\big((\flat+1)$-esimo 'bit' immancabilmente $0\big)$

[3] Si pensi al tempo scandito dai passi di π.

Istantanee còlte nell'imminenza degli $s+1$ passi

In base Q, la memoria complessiva si evolve cosí:

$$\begin{pmatrix} \mathfrak{r}_0 \\ \mathfrak{r}_1 \\ \vdots \\ \mathfrak{r}_m \\ \mathfrak{r}_{m+1} \\ \vdots \\ \mathfrak{r}_r \end{pmatrix} = \begin{pmatrix} a_0 & \mathfrak{r}_{0,s-1} & \cdots & \mathfrak{r}_{0,1} & 0 \\ 0 & \mathfrak{r}_{1,s-1} & \cdots & \mathfrak{r}_{1,1} & a_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \mathfrak{r}_{m,s-1} & \cdots & \mathfrak{r}_{m,1} & a_m \\ 0 & \mathfrak{r}_{m+1,s-1} & \cdots & \mathfrak{r}_{m+1,1} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \mathfrak{r}_{r,s-1} & \cdots & \mathfrak{r}_{r,1} & 0 \end{pmatrix}$$

Istantanee còlte nell'imminenza degli $s+1$ passi

In base Q, la memoria complessiva si evolve cosí:

$$\begin{pmatrix} \mathfrak{r}_0 \\ \mathfrak{r}_1 \\ \vdots \\ \mathfrak{r}_m \\ \mathfrak{r}_{m+1} \\ \vdots \\ \mathfrak{r}_r \end{pmatrix} = \begin{pmatrix} a_0 & \mathfrak{r}_{0,s-1} & \cdots & \mathfrak{r}_{0,1} & 0 \\ 0 & \mathfrak{r}_{1,s-1} & \cdots & \mathfrak{r}_{1,1} & a_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \mathfrak{r}_{m,s-1} & \cdots & \mathfrak{r}_{m,1} & a_m \\ 0 & \mathfrak{r}_{m+1,s-1} & \cdots & \mathfrak{r}_{m+1,1} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \mathfrak{r}_{r,s-1} & \cdots & \mathfrak{r}_{r,1} & 0 \end{pmatrix}$$

Questa è una sommaria descrizione qualitativa, di cui abbiamo incontrato qualche lucido fa una precisazione quantitativa.

Senso delle $\mathfrak{l}_0, \ldots, \mathfrak{l}_\ell$

Intendiamo

$$\mathfrak{l}_{i,t} = \begin{cases} 1 & \text{se l'istruz. al passo } t \text{ è } \mathfrak{S}_i\,, \\ 0 & \text{se è un'altra,} \end{cases}$$

$$\mathfrak{l}_i = \sum_{t=0}^{s} \mathfrak{l}_{i,t}\, Q^t\,,$$

cosicché, in base Q:

$$\begin{pmatrix} \mathfrak{l}_0 \\ \mathfrak{l}_1 \\ \vdots \\ \mathfrak{l}_{\ell-1} \\ \mathfrak{l}_\ell \end{pmatrix} = \begin{pmatrix} 0 & \mathfrak{l}_{0,s-1} & \cdots & \mathfrak{l}_{0,1} & 1 \\ 0 & \mathfrak{l}_{1,s-1} & \cdots & \mathfrak{l}_{1,1} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \mathfrak{l}_{\ell-1,s-1} & \cdots & \mathfrak{l}_{\ell-1,1} & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{pmatrix}$$

Senso delle $\mathfrak{l}_0, \ldots, \mathfrak{l}_\ell$ – Instradamento condizionato

Dalla spiegazione che precede risultano chiari tutti i vincoli sul flusso di controllo, esclusi quelli dell’ultimo tipo — il piú difficile: l’instradamento condizionato.

Instradamento per l'istr. $\Im_i$: **IF** $\mathrm{R}_j = 0$ **GOTO** k

Consideriamo la tabella, formata da 4 righe di $(s+2)\cdot(\flat+1)$ bit:

	$s+1$	s		t		2	1	0
$Q\,\mathfrak{l}_i$	0	$\mathfrak{l}_{i,s-1}$	$\cdots$	$\mathfrak{l}_{i,t-1}$	$\cdots$	$\mathfrak{l}_{i,1}$	$\mathfrak{l}_{i,0}$	0
$\mathfrak{l}_{i+1}$	0	$\mathfrak{l}_{i+1,s}$	$\cdots$	$\mathfrak{l}_{i+1,t}$	$\cdots$	$\mathfrak{l}_{i+1,2}$	$\mathfrak{l}_{i+1,1}$	0
$Q\,\mathrm{I}$	1	1	$\cdots$	1	$\cdots$	1	1	0
$-2\,\mathfrak{r}_j$	-0	$2\,\mathfrak{r}_{j,s}$	$\cdots$	$2\,\mathfrak{r}_{j,t}$	$\cdots$	$2\,\mathfrak{r}_{j,2}$	$2\,\mathfrak{r}_{j,1}$	$2\,\mathfrak{r}_{j,0}$

La richiesta

$$Q\,\mathfrak{l}_i \sqsubseteq \mathfrak{l}_{i+1} + Q\,\mathrm{I} - 2\,\mathfrak{r}_j \qquad (\dagger)$$

è che quando $\mathfrak{l}_{i,t-1} = 1$, i.e., a $t-1$ è attiva un'istruz. $\Im_i$ di tipo condizionale, allora a t si attivi $\Im_{i+1}$ oppure l'istruz. rivale, a seconda che valga o meno

$$0 \quad \neq \quad \mathfrak{r}_{j,t-1} \quad (= \quad \mathfrak{r}_{j,t}\,).$$

Instradamento per l'istr. $\mathcal{S}_i$: **IF** $\mathrm{R}_j = 0$ **GOTO** k –II

Preliminarmente si osservi che la sottrazione produce una sequenza

$Q\,\mathrm{I} - 2\,\mathfrak{r}_j$

d_{s+1}	d_s	$\cdots$	d_t	$\cdots$	d_2	d_1	d_0

di cifre Q-arie dove la parità di d_t è disciplinata, per $t > 0$, dalla regola:

$$d_t \equiv 0 \pmod 2 \textbf{ sse } \mathfrak{r}_{j,t-1} \neq 0$$

Instradamento per l'istr. $\mathfrak{S}_i$: **IF** $\mathrm{R}_j = 0$ **GOTO** k –II

Preliminarmente si osservi che la sottrazione produce una sequenza

$Q\,\mathrm{I} - 2\,\mathfrak{r}_j$

d_{s+1}	d_s	$\cdots$	d_t	$\cdots$	d_2	d_1	d_0

di cifre Q-arie dove la parità di d_t è disciplinata, per $t > 0$, dalla regola:

$$d_t \equiv 0 \pmod 2 \text{ sse } \mathfrak{r}_{j,t-1} \neq 0\,,$$

semplificabile in

$$d_t \equiv 0 \pmod 2 \text{ sse } \mathfrak{r}_{j,t} \neq 0$$

quando $\mathfrak{l}_{i,t-1} = 1$, visto che la $\mathfrak{S}_i$ non modifica la memoria.

Instradamento per l'istr. $\mathfrak{S}_i$: **IF** $\mathrm{R}_j = 0$ **GOTO** k –III

Se ad un istante $t-1 < s$ in cui $\mathfrak{l}_{i,t-1} = 1$ vale $\mathfrak{r}_{j,t} = 0$, la parità di $(\mathfrak{l}_{i+1} + Q\,\mathrm{I} - 2\,\mathfrak{r}_j)_t$, cioè di $\mathfrak{l}_{i+1,t} + d_t$, è dunque data da

$$\mathfrak{l}_{i+1,t} + d_t \quad \not\equiv \quad \mathfrak{l}_{i+1,t} \quad (\textbf{mod } 2) \ ;$$

cosí, richiedendo che $\mathfrak{l}_{i+1,t} + d_t$ sia dispari la **(†)** impone che $\mathfrak{l}_{i+1,t} = 0$, cioè che il salto abbia luogo.

Instradamento per l'istr. $\mathfrak{S}_i$: **IF** $\mathrm{R}_j = 0$ **GOTO** k –III

Se ad un istante $t-1 < s$ in cui $\mathfrak{l}_{i,t-1} = 1$ vale $\mathfrak{r}_{j,t} = 0$, la parità di $(\mathfrak{l}_{i+1} + Q\,\mathrm{I} - 2\,\mathfrak{r}_j)_t$, cioè di $\mathfrak{l}_{i+1,t} + d_t$, è dunque data da

$$\mathfrak{l}_{i+1,t} + d_t \quad \not\equiv \quad \mathfrak{l}_{i+1,t} \quad (\mathbf{mod}\ 2)\ ;$$

cosí, richiedendo che $\mathfrak{l}_{i+1,t} + d_t$ sia dispari la **(†)** impone che $\mathfrak{l}_{i+1,t} = 0$, cioè che il salto abbia luogo. Viceversa, se $\mathfrak{r}_{j,t} \neq 0$, abbiamo

$$\mathfrak{l}_{i+1,t} + d_t \quad \equiv \quad \mathfrak{l}_{i+1,t} \quad (\mathbf{mod}\ 2)\ ;$$

cosí la richiesta che $\mathfrak{l}_{i+1,t} + d_t$ sia dispari comporta che debba valere $\mathfrak{l}_{i+1,t} = 1$, che il salto non abbia luogo.

Instradamento per l'istr. $\mathfrak{S}_i$: **IF** $\mathrm{R}_j = 0$ **GOTO** k –III

Se ad un istante $t-1 < s$ in cui $\mathfrak{l}_{i,t-1} = 1$ vale $\mathfrak{r}_{j,t} = 0$, la parità di $(\mathfrak{l}_{i+1} + Q\,\mathrm{I} - 2\,\mathfrak{r}_j)_t$, cioè di $\mathfrak{l}_{i+1,t} + d_t$, è dunque data da

$$\mathfrak{l}_{i+1,t} + d_t \quad \not\equiv \quad \mathfrak{l}_{i+1,t} \quad (\mathbf{mod}\ 2)\ ;$$

cosí, richiedendo che $\mathfrak{l}_{i+1,t} + d_t$ sia dispari la **(†)** impone che $\mathfrak{l}_{i+1,t} = 0$, cioè che il salto abbia luogo. Viceversa, se $\mathfrak{r}_{j,t} \neq 0$, abbiamo

$$\mathfrak{l}_{i+1,t} + d_t \quad \equiv \quad \mathfrak{l}_{i+1,t} \quad (\mathbf{mod}\ 2)\ ;$$

cosí la richiesta che $\mathfrak{l}_{i+1,t} + d_t$ sia dispari comporta che debba valere $\mathfrak{l}_{i+1,t} = 1$, che il salto non abbia luogo. In ogni caso, la **(†)** ben modella l'instradamento esercitato dall'istruzione $\mathfrak{S}_i$ in esame.

Vincolo sull'evoluzione di stato della memoria –I

Direttamente dalla definizione $\mathfrak{r}_j =_{\text{Def}} \sum_{t=0}^{s} \mathfrak{r}_{j,t} Q^t$ discende

$$\mathfrak{r}_j - Q\,\mathfrak{r}_j = \mathfrak{r}_{j,0} + \sum_{t=1}^{s} (\mathfrak{r}_{j,t} - \mathfrak{r}_{j,t-1})\, Q^t - \mathfrak{r}_{j,s}\, Q^{s+1}$$

Vincolo sull'evoluzione di stato della memoria –I

Direttamente dalla definizione $\mathfrak{r}_j =_{\text{Def}} \sum_{t=0}^{s} \mathfrak{r}_{j,t} Q^t$ discende

$$\mathfrak{r}_j - Q\,\mathfrak{r}_j = \mathfrak{r}_{j,0} + \sum_{t=1}^{s} (\mathfrak{r}_{j,t} - \mathfrak{r}_{j,t-1})\, Q^t - \mathfrak{r}_{j,s}\, Q^{s+1}\,,$$

dove

$$\mathfrak{r}_{j,t} - \mathfrak{r}_{j,t-1} = \Delta_{j,i} \qquad \text{se } i \text{ è l'indice per cui vale } \mathfrak{l}_{i,t-1} = 1$$

$$\therefore\ \mathfrak{r}_{j,t} - \mathfrak{r}_{j,t-1} = \sum_{i=0}^{\ell} \Delta_{j,i}\, \mathfrak{l}_{i,t-1}\,.$$

Vincolo sull'evoluzione di stato della memoria –I

Direttamente dalla definizione $\mathfrak{r}_j =_{\text{Def}} \sum_{t=0}^{s} \mathfrak{r}_{j,t} Q^t$ discende

$$\mathfrak{r}_j - Q\,\mathfrak{r}_j = \mathfrak{r}_{j,0} + \sum_{t=1}^{s} (\mathfrak{r}_{j,t} - \mathfrak{r}_{j,t-1})\, Q^t - \mathfrak{r}_{j,s}\, Q^{s+1}\,,$$

dove

$$\mathfrak{r}_{j,t} - \mathfrak{r}_{j,t-1} = \Delta_{j,i} \qquad \text{se } i \text{ è l'indice per cui vale } \mathfrak{l}_{i,t-1} = 1$$

$$\therefore\ \mathfrak{r}_{j,t} - \mathfrak{r}_{j,t-1} = \sum_{i=0}^{\ell} \Delta_{j,i}\, \mathfrak{l}_{i,t-1}\,.$$

Pertanto

$$\mathfrak{r}_j - Q\,\mathfrak{r}_j = \mathfrak{r}_{j,0} - \mathfrak{r}_{j,s} Q^{s+1} + \sum_{i=0}^{\ell} \sum_{t=1}^{s} \Delta_{j,i}\, \mathfrak{l}_{i,t-1}\, Q^t$$

Vincolo sull'evoluzione di stato della memoria –I

Direttamente dalla definizione $\mathfrak{r}_j =_{\text{Def}} \sum_{t=0}^{s} \mathfrak{r}_{j,t} Q^t$ discende

$$\mathfrak{r}_j - Q\,\mathfrak{r}_j = \mathfrak{r}_{j,0} + \sum_{t=1}^{s} (\mathfrak{r}_{j,t} - \mathfrak{r}_{j,t-1})\, Q^t - \mathfrak{r}_{j,s}\, Q^{s+1}\,,$$

dove

$$\mathfrak{r}_{j,t} - \mathfrak{r}_{j,t-1} = \Delta_{j,i} \qquad \text{se } i \text{ è l'indice per cui vale } \mathfrak{l}_{i,t-1} = 1$$

$$\therefore\ \mathfrak{r}_{j,t} - \mathfrak{r}_{j,t-1} = \sum_{i=0}^{\ell} \Delta_{j,i}\, \mathfrak{l}_{i,t-1}\,.$$

Pertanto

$$\mathfrak{r}_j - Q\,\mathfrak{r}_j = \mathfrak{r}_{j,0} - \mathfrak{r}_{j,s} Q^{s+1} + \sum_{i=0}^{\ell} \sum_{t=1}^{s} \Delta_{j,i}\, \mathfrak{l}_{i,t-1}\, Q^t$$

e, considerato che quando $\Delta_{j,i} \neq 0$ vale $\mathfrak{l}_{i,s} = 0$, onde

$$\sum_{t=1}^{s} \mathfrak{l}_{i,t-1} Q^t = \sum_{t=0}^{s} \mathfrak{l}_{i,t} Q^{t+1} = Q\,\mathfrak{l}_i\,,$$

Vincolo sull'evoluzione di stato della memoria –I

Direttamente dalla definizione $\mathfrak{r}_j =_{\text{Def}} \sum_{t=0}^{s} \mathfrak{r}_{j,t} Q^t$ discende

$$\mathfrak{r}_j - Q\,\mathfrak{r}_j = \mathfrak{r}_{j,0} + \sum_{t=1}^{s} (\mathfrak{r}_{j,t} - \mathfrak{r}_{j,t-1})\, Q^t - \mathfrak{r}_{j,s}\, Q^{s+1}\,,$$

dove

$$\mathfrak{r}_{j,t} - \mathfrak{r}_{j,t-1} = \Delta_{j,i} \qquad \text{se } i \text{ è l'indice per cui vale } \mathfrak{l}_{i,t-1} = 1$$

$$\therefore\ \mathfrak{r}_{j,t} - \mathfrak{r}_{j,t-1} = \sum_{i=0}^{\ell} \Delta_{j,i}\, \mathfrak{l}_{i,t-1}\,.$$

Pertanto

$$\mathfrak{r}_j - Q\,\mathfrak{r}_j = \mathfrak{r}_{j,0} - \mathfrak{r}_{j,s} Q^{s+1} + \sum_{i=0}^{\ell} \sum_{t=1}^{s} \Delta_{j,i}\, \mathfrak{l}_{i,t-1}\, Q^t$$

e, considerato che quando $\Delta_{j,i} \neq 0$ vale $\mathfrak{l}_{i,s} = 0$, onde

$$\sum_{t=1}^{s} \mathfrak{l}_{i,t-1} Q^t = \sum_{t=0}^{s} \mathfrak{l}_{i,t} Q^{t+1} = Q\,\mathfrak{l}_i\,,$$

possiamo riscrivere

$$\mathfrak{r}_j - Q\,\mathfrak{r}_j = \mathfrak{r}_{j,0} - \mathfrak{r}_{j,s}\, Q^{s+1} + Q \sum_{i=0}^{\ell} \Delta_{j,i} \mathfrak{l}_i\,.$$

Vincolo sull'evoluzione di stato della memoria –II

Per tirar le somme, non serve altro che richiamare:

$$\mathfrak{r}_{j,0} = \begin{cases} 0 & \text{se} \quad j = 0 \quad \vee \quad j > \boldsymbol{m} \\ a_j & \text{se} \quad 0 < j \leqslant \boldsymbol{m}\,, \end{cases}$$

$$\mathfrak{r}_{j,s} = \begin{cases} 0 & \text{se} \quad j \neq 0 \\ a_0 & \text{se} \quad j = 0\,. \end{cases}$$

⊣

Congettura di Goldbach

La congettura di Christian Goldbach (ca. 1742) asserisce che:

"ogni numero pari maggiore di 2 può essere scritto come somma di due primi".

Congettura di Goldbach espressa come eq. $G(0, x, \vec{z}) = 0$

Si scriva un programma γ che computi la funzione

$$g(a_1) = \begin{cases} 1 & \text{se vi sono numeri primi } p, q \\ & \quad \text{tali che } 2\,a_1 + 4 = p + q, \\ 0 & \text{altrimenti,} \end{cases}$$

Congettura di Goldbach espressa come eq. $G(0, x, \vec{z}) = 0$

Si scriva un programma γ che computi la funzione

$$g(a_1) = \begin{cases} 1 & \text{se vi sono numeri primi } p, q \\ & \quad \text{tali che } 2\,a_1 + 4 = p + q, \\ 0 & \text{altrimenti,} \end{cases}$$

utilizzando un procedimento che determini i numeri primi (ad es. `http://en.wikipedia.org/wiki/Sieve_of_Eratosthenes`).

Congettura di Goldbach espressa come eq. $G(0, x, \vec{z}) = 0$

Si scriva un programma γ che computi la funzione

$$g(a_1) \quad = \quad \begin{cases} 1 & \text{se vi sono numeri primi } p, q \\ & \quad \text{tali che } 2\,a_1 + 4 = p + q, \\ 0 & \text{altrimenti,} \end{cases}$$

utilizzando un procedimento che determini i numeri primi (ad es. `http://en.wikipedia.org/wiki/Sieve_of_Eratosthenes`).

Alla stregua della dimostrazione del teorema DPR, si ricavi da γ un sistema di equazioni diofantee esponenziali —e poi una singola eq. $G(a_0, a_1, \vec{z}) = 0$— descrivente il grafo $\mathcal{G}(a_0, a_1)$ di g.

Congettura di Goldbach espressa come eq. $G(0, x, \vec{z}) = 0$

Si scriva un programma γ che computi la funzione

$$g(a_1) \quad = \quad \begin{cases} 1 & \text{se vi sono numeri primi } p, q \\ & \quad \text{tali che } 2\,a_1 + 4 = p + q, \\ 0 & \text{altrimenti,} \end{cases}$$

utilizzando un procedimento che determini i numeri primi (ad es. `http://en.wikipedia.org/wiki/Sieve_of_Eratosthenes`).

Alla stregua della dimostrazione del teorema DPR, si ricavi da γ un sistema di equazioni diofantee esponenziali —e poi una singola eq. $G(a_0, a_1, \vec{z}) = 0$— descrivente il grafo $\mathcal{G}(a_0, a_1)$ di g.

La congettura è dunque vera sse $G(0,\ x,\ \vec{z}) = 0$ *manca* di soluz.

Un insieme semidecidibile ma non decidibile

Indichiamo per comodità con ψ_π la funzione monadica

$$\psi_\pi : \quad \mathbb{N} \quad \rightharpoonup \quad \mathbb{N} \,,$$

totale o parziale, computata da un programma π.

Un insieme semidecidibile ma non decidibile

Indichiamo per comodità con ψ_π la funzione monadica

$$\psi_\pi : \quad \mathbb{N} \quad \rightharpoonup \quad \mathbb{N} \,,$$

totale o parziale, computata da un programma π.

La teoria dell'enumerabilità ricorsiva ci permette di scrivere un programma κ tale che per ogni altro programma π accade che

$$\{\, x : \langle x, y\rangle \in \psi_\pi \,\} \quad \neq \quad \mathbb{N} \setminus \{\, x : \langle x, y\rangle \in \psi_\kappa \,\} \,;$$

cioè, i dominî delle funz. computate non sono mai insiemi complementari.

Un insieme semidecidibile ma non decidibile

Indichiamo per comodità con ψ_π la funzione monadica

$$\psi_\pi : \quad \mathbb{N} \quad \rightharpoonup \quad \mathbb{N} \,,$$

totale o parziale, computata da un programma π.

La teoria dell'enumerabilità ricorsiva ci permette di scrivere un programma $\mathcal{K}$ tale che per ogni altro programma π accade che

$$\{\, x : \langle x, y\rangle \in \psi_\pi \,\} \quad \neq \quad \mathbb{N} \setminus \{\, x : \langle x, y\rangle \in \psi_\kappa \,\} \,;$$

cioè, i dominî delle funz. computate non sono mai insiemi complementari.

In altre parole

$$\mathcal{K} \quad =_{\mathsf{Def}} \quad \{\, x : \langle x, y\rangle \in \psi_\kappa \,\}$$

è semidecidibile ma non decidibile.

Un insieme semidecidibile ma non decidibile (cont.)

Il Teor. DPR ci permette, poi, di costruire un pol. diofanteo esponenziale parametrico K tale che

$$b = \psi_K(a) \iff \exists\, \vec{z} \quad K(a, b, \vec{z}) = 0\,;$$

Un insieme semidecidibile ma non decidibile (cont.)

Il Teor. DPR ci permette, poi, di costruire un pol. diofanteo esponenziale parametrico K tale che

$$b = \psi_\kappa(a) \iff \exists\, \vec{z} \quad K(a, b, \vec{z}) = 0\,;$$

cosí $\mathcal{K}$ risulta esistenzialmente definito dall'equazione

$$K(a, x, \vec{z}) = 0\,,$$

dove il secondo parametro è stato 'declassato' a nuova incognita.

Un problema algoritmicamente insolubile

Discende di qui l’insolubilità algoritmica del X di Hilbert riferito alle equazioni diofantee esponenziali—anziché a quelle polinomiali.

Un problema algoritmicamente insolubile

Discende di qui l'insolubilità algoritmica del X di Hilbert riferito alle equazioni diofantee esponenziali—anziché a quelle polinomiali.

Ove riuscissimo a risolvere quel problema in totale generalità, potremmo infatti, per qualsiasi assegnato valore $\boldsymbol{a}$, stabilire se l'equazione $K(\boldsymbol{a}, x, \vec{z}) = 0$ abbia o no soluzione.

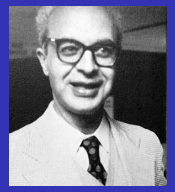
Un problema algoritmicamente insolubile

Discende di qui l'insolubilità algoritmica del X di Hilbert riferito alle equazioni diofantee esponenziali—anziché a quelle polinomiali.

Ove riuscissimo a risolvere quel problema in totale generalità, potremmo infatti, per qualsiasi assegnato valore $\boldsymbol{a}$, stabilire se l'equazione $K(\boldsymbol{a}, x, \vec{z}) = 0$ abbia o no soluzione.

Ma allora potremmo decidere se $\boldsymbol{a}$ stia in $\mathcal{K}$ oppure no — contraddizione.

Celebre abbaglio in una recensione di [DPR61]

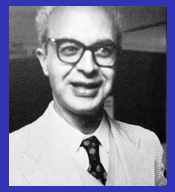


(Georg Kreisel, 1923–2015)

"*These results are superficially related to Hilbert's tenth Problem on (ordinary, i.e., non-exponential) Diophantine equations. The proof of the authors' results, though very elegant, does not use recondite facts in the theory of numbers nor in the theory of r.e. sets, and so it is likely that the present result is not closely connected with Hilbert's tenth Problem.*

[Kre62]

Celebre abbaglio in una recensione di [DPR61]



(Georg Kreisel, 1923–2015)

"*These results are superficially related to Hilbert's tenth Problem on (ordinary, i.e., non-exponential) Diophantine equations. The proof of the authors' results, though very elegant, does not use recondite facts in the theory of numbers nor in the theory of r.e. sets, and so it is likely that the present result is not closely connected with Hilbert's tenth Problem. Also it is not altogether plausible that all (ordinary) Diophantine problems are uniformly reducible to those in a fixed number of variables of fixed degree, which would be the case if all r.e. sets were Diophantine.*" [Kre62]

L'aspetto paradossale segnalato da Georg Kreisel

Alla luce del teorema di universalità di Turing, discende dal teorema DPR che

Corollario: Per ogni $m \in \mathbb{N}$, esiste un'equazione diofantea esponenziale

$$U(a_1, \ldots, a_m, a_{m+1}, x_1, \ldots, x_k) \quad = \quad 0$$

tale che gli insiemi diofantei esponenziali di dimensione m sono tutti e soli gli insiemi definiti da una delle equazioni

$$U(a_1, \ldots, a_m, \underline{a}, x_1, \ldots, x_k) \quad = \quad 0$$

che risultano dal variare di $\underline{a}$ in $\mathbb{N}$. $\dashv$

Storia della dimostrazione del teorema DPR

- Davis e Putnam dedussero nel 1959 [DP59, Part III] l'asserto

$$\mathfrak{R} \subseteq \mathfrak{E}$$

da una *congettura*, denominata $\mathrm{P.A.P.}$: che vi siano progressioni aritmetiche arbitrariamente lunghe formate per intero da numeri primi

Storia della dimostrazione del teorema DPR

- Davis e Putnam dedussero nel 1959 [DP59, Part III] l'asserto

$$\mathfrak{R} \subseteq \mathfrak{E}$$

da una *congettura*, denominata $\mathrm{P.A.P.}$: che vi siano progressioni aritmetiche arbitrariamente lunghe formate per intero da numeri primi

- Julia Robinson rese la dimostrazione indipendente da tale congettura nel 1960 (ne risultò [DPR61])
-
-

Storia della dimostrazione del teorema DPR

- Davis e Putnam dedussero nel 1959 [DP59, Part III] l'asserto

$$\mathfrak{R} \subseteq \mathfrak{E}$$

da una *congettura*, denominata $\mathrm{P.A.P.}$: che vi siano progressioni aritmetiche arbitrariamente lunghe formate per intero da numeri primi

- Julia Robinson rese la dimostrazione indipendente da tale congettura nel 1960 (ne risultò [DPR61])
- Yuri V. Matiyasevich perfezionò la dim., eliminandone la sotto-determinazione [Mat74]
-

Storia della dimostrazione del teorema DPR

- Davis e Putnam dedussero nel 1959 [DP59, Part III] l'asserto

$$\mathfrak{R} \subseteq \mathfrak{E}$$

da una *congettura*, denominata $\mathrm{P.A.P.}$: che vi siano progressioni aritmetiche arbitrariamente lunghe formate per intero da numeri primi

- Julia Robinson rese la dimostrazione indipendente da tale congettura nel 1960 (ne risultò [DPR61])
- Yuri V. Matiyasevich perfezionò la dim., eliminandone la sotto-determinazione [Mat74]
- Una dimostrazione molto tersa —quella qui presentata sulla scorta di [Dav93]— fu prodotta (tra il 1981 e il 1983) da James Jones e Yuri V. Matiyasevich [JM84]

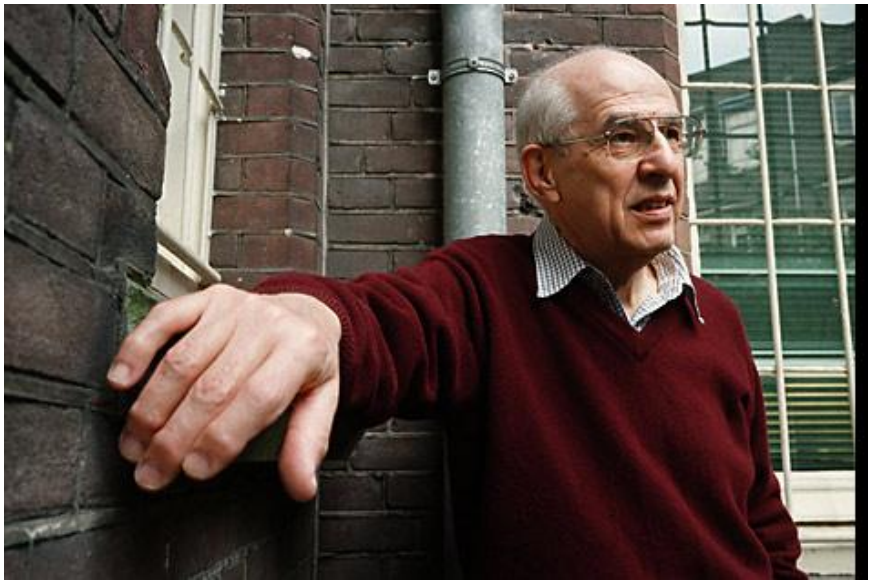
In merito alla P.A.P.

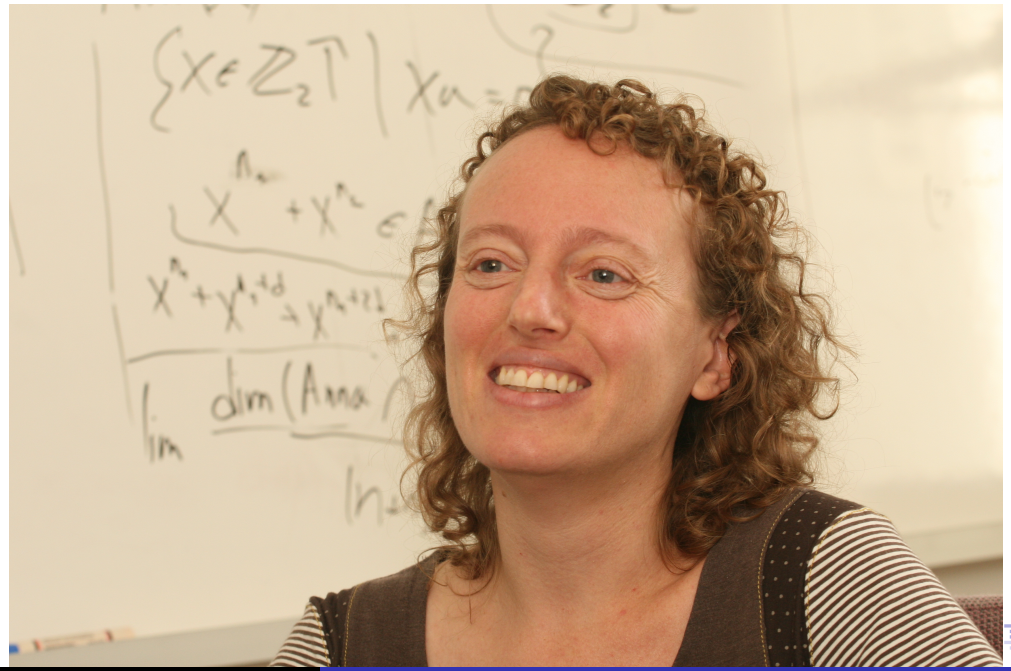
"*It was in the summer of 1959 that Hilary and I really hit the jackpot. We decided to see how far we could get with the approach we had used at the Logic Institute in Ithaca, if, following Julia Robinson's lead, we were willing to permit variable exponents in our Diophantine equations.*" [OP16, pp. 15–16]

In merito alla P.A.P.

"*It was in the summer of 1959 that Hilary and I really hit the jackpot. We decided to see how far we could get with the approach we had used at the Logic Institute in Ithaca, if, following Julia Robinson's lead, we were willing to permit variable exponents in our Diophantine equations.*" [OP16, pp. 15–16]

"*It wasn't until 2004 that Ben Green and Terence Tao proved that P.A.P. is true, thus validating our work as a complete proof, but only well after the fact.*" [Dav10]





Da una congettura di Davis-Putnam al teor. Tao-Ziegler

In merito alla congettura

> "*per ogni* q*, esistono interi positivi* r, s *tali che i numeri* $r,\ r+s,\ r+2s, \ldots,\ r+qs$ *risultino tutti primi.*"

Da una congettura di Davis-Putnam al teor. Tao-Ziegler

In merito alla congettura

> "*per ogni* q*, esistono interi positivi* r, s *tali che i numeri* $r,\ r+s,\ r+2s, \ldots,\ r+qs$ *risultino tutti primi.*"

Harold N. Shapiro può averne, tramite conversazioni, stimolato l'ideazione (v. [DPR61])

Da una congettura di Davis-Putnam al teor. Tao-Ziegler

In merito alla congettura

> "*per ogni q, esistono interi positivi r, s tali che i numeri r, $r+s$, $r+2s, \ldots, r+qs$ risultino tutti primi.*"

Harold N. Shapiro può averne, tramite conversazioni, stimolato l'ideazione (v. [DPR61])

Ben Joseph Green & Terence Tao l'hanno dimostrata nel 2004

Da una congettura di Davis-Putnam al teor. Tao-Ziegler

In merito alla congettura

> "*per ogni q, esistono interi positivi r, s tali che i numeri r, $r+s$, $r+2s, \ldots, r+qs$ risultino tutti primi.*"

Harold N. Shapiro può averne, tramite conversazioni, stimolato l'ideazione (v. [DPR61])

Ben Joseph Green & Terence Tao l'hanno dimostrata nel 2004

Terence Tao & Tamar Debora Ziegler ne hanno dimostrato una formidabile generalizzazione nel 2006 (v. [TZ08]):

Da una congettura di Davis-Putnam al teor. Tao-Ziegler

In merito alla congettura

> "*per ogni q, esistono interi positivi r, s tali che i numeri r, $r+s$, $r+2s, \ldots, r+qs$ risultino tutti primi.*"

Harold N. Shapiro può averne, tramite conversazioni, stimolato l'ideazione (v. [DPR61])

Ben Joseph Green & Terence Tao l'hanno dimostrata nel 2004

Terence Tao & Tamar Debora Ziegler ne hanno dimostrato una formidabile generalizzazione nel 2006 (v. [TZ08]):

> " *Dati i polinomi $P_0, P_1, \ldots, P_q \in \mathbb{Z}[x]$, con $P_0(0) = P_1(0) = \cdots = P_q(0) = 0$, e un $\epsilon > 0$, vi sono infiniti interi $\boldsymbol{r}$ ed $\boldsymbol{s}$, con $1 \leqslant \boldsymbol{s} \leqslant \boldsymbol{r}^{\epsilon}$, tali che $\boldsymbol{r} + P_0(\boldsymbol{s}), \boldsymbol{r} + P_1(\boldsymbol{s}), \ldots, \boldsymbol{r} + P_q(\boldsymbol{s})$ risultino primi.*"

Voci bibliografiche

Martin Davis.
Arithmetical problems and recursively enumerable predicates.
The Journal of Symbolic Logic, 18(1):33–41, 1953.

Martin Davis. *Lecture Notes in Logic*.
Courant Institute of Mathematical Sciences, New York University, 1993.

Martin Davis.
Il decimo problema di Hilbert: equazioni e computabilità.
In Claudio Bartocci and Piergiorgio Odifreddi, editors, *La matematica – Pensare il mondo*, Volume IV, Grandi Opere. Einaudi, 2010.

Martin Davis and Hilary Putnam.
A computational proof procedure; Axioms for number theory; Research on Hilbert's Tenth Problem.
Technical Report AFOSR TR59-124, U.S. Air Force, October 1959. (Part III reprinted in [OP16, pp. 411-430]).


Martin Davis, Hilary Putnam, and Julia Robinson.
The decision problem for exponential Diophantine equations.
Annals of Mathematics, Second Series, 74(3):425–436, 1961.


J. P. Jones and Y. V. Matijasevič.
Register machine proof of the theorem on exponential Diophantine representation of enumerable sets.
The Journal of Symbolic Logic, 49(3):818–829, 1984.


Georg Kreisel.
A3061: Davis, Martin; Putnam, Hilary; Robinson, Julia. The decision problem for exponential Diophantine equations.
Mathematical Reviews, 24A(6A):573, 1962.


Yu. V. Matiyasevich.
Sushchestvovanie neèffektiviziruemykh otsenok v teorii èkponentsial'no diofantovykh uravneniĭ.
Zapiski Nauchnykh Seminarov Leningradskogo Otdeleniya Matematicheskogo Instituta im. V. A. Steklova AN SSSR (LOMI), 40:77–93, 1974.

(Russian. Translated into English as Yu. V. Matiyasevich, Existence of noneffectivizable estimates in the theory of exponential Diophantine equations, *Journal of Soviet Mathematics*, 8(3):299–311, 1977).


Yuri Vladimirovich Matiyasevich.
Hilbert's tenth problem.
The MIT Press, Cambridge (MA) and London, 1993.


Eugenio G. Omodeo and Alberto Policriti, editors.
Martin Davis on Computability, Computational Logic, and Mathematical Foundations, volume 10 of *Outstanding Contributions to Logic*.
Springer, 2016.


Terence Tao and Tamar Ziegler.
The primes contain arbitrarily long polynomial progressions.
Acta Mathematica, 201:213–305, 2008.