La Programma 101 e la notazione polacca inversa

Giovanni A. Cignoni, Progetto HMR,
Giugno 2017

Fra le diverse voci che girano sulla Programma 101 ce ne è una[1] che la dice far uso della notazione polacca inversa. Non è così.

Facciamo chiarezza non solo per amor di precisione: la voce che la P101 sia una calcolatrice RPN confonde le idee sulla RPN, colloca la P101 in una classe di macchine che non è la sua, non aiuta a spiegare come funziona la P101.

La notazione polacca inversa

È, appunto, una notazione, nata molto prima delle sue applicazioni informatiche. Il nome (abbreviato in RPN, per Reverse Polish Notation) è in onore di Jan Łukasiewicz, filosofo polacco che la introdusse nei suoi lavori di logica a partire dagli anni '20 del secolo scorso.

Il senso della RPN è principalmente poter scrivere espressioni aritmetiche equivalenti a quelle che si scrivono usando le parentesi e le convenzioni di precedenza degli operatori, ma senza le parentesi e le convenzioni. Usa cioè meno segni e meno regole.

Da questo punto di vista è una soluzione elegante e minimale. La minimalità però si paga. La RPN, almeno finché non ci si fa l'abitudine, risulta meno facile da usare; le regole sono poche ma bisogna applicarle mentalmente. Per capire meglio cosa significa basta pensare alle raccomandazioni che tipicamente si fanno ai programmatori: invece di sfruttare le regole di precedenza, meglio esplicitare qualche parentesi. L'espressione è più chiara e si usa la cortesia di evitare un esercizio (e possibili errori) a chi dovrà leggerla. Ecco, la RPN parte invece dall'opinione (lecita) che l'esercizio mentale sia cosa sana e giusta.

Per esempio, l'espressione[2] tradizionalmente scritta:

$$5 + ((1 + 2) \times 4) - 3$$

in RPN si scrive:

$$5\ 1\ 2 + 4 \times + 3 -$$

Da notare che fra '5', '1', e '2' c'è uno spazio, significativo affinché le tre cifre siano interpretate come tre operandi e non come un unico '512'. Nella notazione tradizionale invece gli spazi non sono significativi: gli operatori separano gli operandi.

Ovviamente, il risultato dell'espressione, comunque sia scritta, deve essere 14.

1 Ringraziamo David Pillitteri che ha sollevato la questione a *Vicoretrò 2017* durante la presentazione "HMR Racconta - Segreti per la vittoria: intelligenze artificiali di altri tempi (fatte girare sulla Programma101)" e ha in seguito fornito esempi che mostrano come la voce gira sulla rete.

2 Useremo questo esempio per mostrare che la P101 non è RPN, per garantire imparzialità è preso dalla pagina sulla RPN di Wikipedia (http://en.wikipedia.org/wiki/Reverse_Polish_notation).

La RPN e le calcolatrici

A un certo punto, la minimalità delle regole per interpretare la RPN è diventata un'opportunità per realizzare calcolatrici capaci di calcolare espressioni complesse, ma con un'organizzazione dell'hardware semplice, conveniente ed efficiente. Le calcolatrici RPN sono costruite intorno a una precisa soluzione implementativa: la memoria a pila (o *stack*).

La pila è una struttura dati in cui gli operandi sono impilati uno "sopra" l'altro e prelevati dalla "cima" – gli amanti degli acronimi inglesi dicono LIFO, per *Last In First Out*. Nelle calcolatrici è utile per il calcolo delle espressioni. Nei calcolatori è lo strumento principe per passare parametri/risultati in chiamate/ritorni di sottoprogrammi. La pila e la RPN sono anche alla base di alcuni linguaggi di programmazione, uno fra tutti il *Forth*.[3]

Riprendiamo il nostro esempio di espressione in RPN "5 1 2 + 4 × + 3 –" e interpretiamolo proprio come fa una calcolatrice RPN, cioè usando la pila. Per descrivere la pila usiamo una lista fra graffe, con la cima a destra, e.g {A, B}, B è l'ultimo inserito e il primo candidato ad essere prelevato. Battendo un operando, la calcolatrice semplicemente lo inserisce nella pila:

'5'	→	{5}
'1'	→	{5, 1}
'2'	→	{5, 1, 2}

Battendo un operatore, la calcolatrice estrae i primi due valori della pila, applica l'operatore e reinserisce il risultato nella pila, quindi:

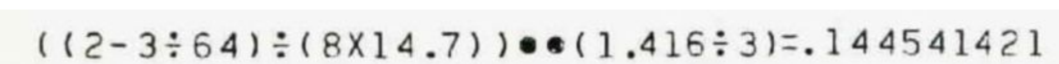
'+'	→	{5}, 1 + 2	→	{5, 3}

I passi successivi nell'interpretazione della nostra espressione sono chiari:

'4'	→	{5, 3, 4}		
'×'	→	{5}, 3 × 4	→	{5, 12}
'+'	→	{}, 5 + 12	→	{17}
'3'	→	{17, 3}		
'–'	→	{}, 17 – 3	→	{14}

Il valore che rimane nella pila, 14, è il risultato atteso.

Fra le prime calcolatrici costruite su un'architettura hardware a pila vanno citate la *Friden EC-130*, introdotta nel giugno del 1963 e la *Mathatron*, presentata nel novembre 1963. La Friden era una calcolatrice RPN a tutti gli effetti, la Mathatron invece usava pila e RPN solo internamente, per i programmi delle (notevoli) librerie di funzioni matematiche residenti nella memoria permanente. Da punto di vista utente invece la Mathatron si usava con la più comoda notazione tradizionale, con la macchina che gestiva parentesi e precedenza degli operatori. Ecco un'espressione, esattamente come veniva battuta e come si vedeva stampata sul nastrino di carta della macchina:[4]



Il risultato era calcolato e stampato dopo la pressione dell'uguale (il '●●' era l'operatore di elevazione a potenza). Notevole eh?

3 Il Forth nasce nei primi anni '70 dal lavoro di C.H. Moore, sarà portato (principalmente dalla Forth Inc.) su molte piattaforme e usato con successo per le applicazioni più varie, dai videogiochi allo spazio. È un linguaggio standard ANSI dal 1994 (http://www.forth.org/svfig/Win32Forth/DPANS94.txt).

4 Immagine dal *Old Calculator Web Museum* (http://www.oldcalculatormuseum.com/c-math8-48m.html).

La Programma 101, RPN? Potrebbe sembrare...

La Programma 101, presentata nel novembre 1965, non ha una memoria pila. Non ha neanche una memoria a celle indirizzabili, come tipicamente sono quelle dei calcolatori. È una calcolatrice a registri: ha 10 registri, ognuno accessibile per nome e indipendentemente. Non avendo l'architettura hardware di una calcolatrice RPN non può funzionare in RPN.

Tuttavia, avendo una conoscenza vaga della RPN e calcolando sulla P101 solo espressioni banali, si può essere tratti in errore. Anche la mancanza del tasto '=' sulla tastiera della P101 può ingannare; ma, se è vero che sulle calcolatrici RPN non usa, dedurre dalla sua assenza che la P101 è RPN è prendere per una prova quello che, al più, è un indizio.

Vediamo meglio come stanno le cose e come nascono le voci.

Prima occorre aggiungere qualcosa sui registri e sul funzionamento della P101:

- il registro A è l'accumulatore; ogni operazione aritmetica produce il risultato in A;
- M è il registro di trasferimento, tutti i valori battuti sulla tastiera finiscono in M;
- battere un operatore aritmetico, da solo, fa sì che sia applicato ai valori in A e in M (nell'ordine) con il risultato che, ovviamente, si forma in A.

Proviamo ora un paio di esempi, esprimendoli in RPN, mostrando come si calcolano su una vera calcolatrice RPN (useremo la *HP 35*[5]), e infine vedendoli sulla P101.

Consideriamo l'espressione "12 + 3", in RPN "12 3 +".

Abbiamo detto che in RPN lo spazio è significativo, su una calcolatrice RPN per separare gli operandi serve un tasto che svolga questo compito. Sulla HP 35 è 'ENTER↑', per cui, per calcolare la nostra espressione la sequenza da battere è:

    12 ENTER↑ 3 +

appena battuto il '+', come ogni calcolatrice RPN che si rispetti appare il risultato, 15.

Sulla P101[6] la sequenza è simile, cambia solo l'etichetta del tasto:

    12 A↓ 3 +

Ma, dopo aver battuto il '+', sulla P101 non succede niente. Per vedere il risultato occorre chiedere la stampa del contenuto di A battendo i tasti 'A' e '◇'; la sequenza completa è:

    12 A↓ 3 + A ◇

Assumendo la richiesta esplicita della stampa un dettaglio, in effetti sembra RPN.

Proviamo un'altra espressione: "(12 + 3) × 4", in RPN "12 3 + 4 ×". Sulla HP 35:

    12 ENTER↑ 3 + 4 ×

e sul display, appena battuto il '×', appare il risultato, 60.

Sulla P101:

    12 A↓ 3 + 4 ×

e questa volta il risultato appare subito. La P101 ha un comportamento asimmetrico: dopo somme e sottrazioni non stampa A, ma lo fa dopo moltiplicazioni e divisioni. Però, asimmetrie a parte, la P101 continua a sembrare proprio RPN.

5 La RPN era stata usata anche sulla *HP 9100A* del 1968, ma la HP 35 del 1972 è la prima scientifica tascabile ed ebbe un successo straordinario; "12 3 +" è il primo esempio sul manuale originale della 35. Per un buon sim della HP 35: http://www.hpmuseum.org/simulate/hp35sim/hp35sim.htm.

6 Per il simulatore "ufficiale" della P101: http://p101.unicas.it/p101/.

Sembra RPN, ma non lo è

Riprendiamo l'espressione del nostro primo esempio "$5 + ((1 + 2) \times 4) - 3$", che in RPN diventa "5 1 2 + 4 × + 3 −"; sulla HP 35 la battiamo seguendo pari pari la RPN:

5 ENTER↑ 1 ENTER↑ 2 + 4 × + 3 −

e sul display, appena battuto il '−', come ci si aspetta da una vera calcolatrice RPN, appare il risultato corretto, 14.

Sulla P101 invece, se proviamo a battere:

5 A↓ 1 A↓ 2 + 4 × + 3 − A ◇

con 'A' e '◇' aggiunti solo per stampare il contenuto di A, otteniamo 13: sbagliato!

Ovviamente: la P101 non è una calcolatrice RPN, tentare di usarla credendola tale porta a combinare pasticci. Se in qualche caso funziona è, appunto, un caso.

Questo è quel che succede nella P101 passo per passo:

'5'	→	M=5	A=0
'A↓'	→	M=5	A=5
'1'	→	M=1	A=5
'A↓'	→	M=1	A=1
'2'	→	M=2	A=1
'+'	→	M=2	A=3
'4'	→	M=4	A=3
'×'	→	M=4	A=12
'+'	→	M=4	A=16
'3'	→	M=3	A=16
'−'	→	M=3	A=13

L'effetto di 'A↓' è un trasferimento fra registri, non un'inserzione in una pila (una *push*, direbbero gli Inglesi).

Volendo, in termini di strutture dati, si potrebbe dire che un registro è una pila di lunghezza 1, ma è un'argomentazione buona per una discussione teorica – con tutto il rispetto: aiuta per esempio a spiegare i casi in cui la P101 sembra RPN.

Tuttavia, la lunghezza 1 è un caso limite. Per dire, un registro è anche una coda[7] lunga 1. Con lunghezza 1, pila, coda e registro sono la stessa cosa: non si possono usare per caratterizzare seriamente diversi modi di realizzare le macchine calcolatrici.

Più interessante invece è notare che la P101 non solo non è RPN, ma neanche ha le parentesi. Quindi, se vogliamo calcolare correttamente la nostra espressione tocca a noi sciogliere le parentesi e fare i calcoli nell'ordine giusto, "$5 + ((1 + 2) \times 4) - 3$" diventa:

1 A↓ 2 + 4 × 5 + 3 − A ◇

Se, a questo punto, uno si domandasse: ma allora la P101 che notazione usa?
La risposta corretta è: la sua.

Va da sé che, anche se non è RPN e non ha le parentesi, la P101 rimane un bell'oggetto :)

7 Altra tipica struttura dati amata dagli informatici, in acronimo FIFO, per *First In First Out*.