

# LIFO ante litteram

*Pierangelo Indolfi, Politecnico di Bari*

## Sommario

*La struttura dati astratta pila (stack) viene tradizionalmente introdotta nei corsi di programmazione attraverso la sua definizione formale, seguita da esempi applicativi. Questo articolo propone un percorso inverso: mostrare che il principio LIFO affiora spontaneamente in domini del sapere lontanissimi dall'informatica, molto prima che qualcuno pensasse di formalizzarlo. L'archeologia, la neurobiologia, la psicoanalisi, i protocolli di rete, i sistemi operativi a strati, il protocollo di decontaminazione in zona ad alto rischio biologico: ciascuno di questi contesti presenta la stessa geometria, quella in cui l'ordine di deposizione determina l'ordine inverso di accesso. La Torre di Hanoi occupa una posizione speciale in questa galleria: il suo caso rivela che lo stack non è uno strumento scelto per risolvere il problema, ma la struttura profonda del problema stesso. L'articolo discute le implicazioni didattiche di questo approccio, con particolare riferimento ai corsi di Ingegneria dei Sistemi Medicali, dove il dominio applicativo fornisce contesti motivanti di straordinaria concretezza. La tesi di fondo è epistemologica: formalizzare una struttura dati non è scoprire l'acqua calda, ma riconoscere una faccia che era già lì, e restituirla al mondo sotto forma di strumento capace di produrre cose nuove.*

**Parole chiave:** strutture dati, stack, LIFO, didattica dell'informatica, metafore cognitive, ingegneria dei sistemi medicali, ricorsione, Torre di Hanoi

## 1. Introduzione

Presentare una struttura dati astratta agli studenti di ingegneria seguendo l'ordine canonico – definizione formale seguita da esempi applicativi – produce un effetto prevedibile: lo studente acquisisce il meccanismo senza coglierne il significato. Sa cos'è uno stack, sa che funziona secondo il principio LIFO, sa come usare i comandi push e pop. Ciò che ignora è perché quella struttura esista, quale necessità del mondo reale abbia spinto qualcuno a isolarla e a darle un nome.

Le strutture dati traggono origine dall'osservazione di fenomeni ricorrenti che attendono di essere formalizzati, mentre i libri di testo ne danno notizia solo molto più tardi. Lo stack, in particolare, è un modello che emerge spontaneamente in ambiti molto distanti tra loro, ben prima che l'informatica esistesse come

disciplina. L'archeologia lo incontra ogni volta che uno scavo porta alla luce strati sovrapposti di civiltà. La neurobiologia lo riconosce nell'architettura del sistema nervoso centrale (MacLean, 1990). La psicoanalisi lo utilizza come metafora fondante dell'io (Freud, 1899). La teoria dei protocolli di rete lo formalizza come principio di incapsulamento (Tanenbaum, 2003).

Mostrare tutto questo agli studenti prima ancora di scrivere una sola riga di codice è un atto di onestà intellettuale nei confronti della natura delle idee, oltre che una scelta didattica più efficace (Papert, 1980).

## **2. Il modello di stratificazione**

La proprietà che definisce uno stack, ovvero il fatto che l'ultimo elemento depositato sia il primo ad essere recuperato, è una conseguenza inevitabile di un processo fisico elementare: la stratificazione. Ogni volta che oggetti, eventi o strutture si accumulano nel tempo senza mescolarsi, il risultato è una sequenza in cui l'ordine di deposito determina l'ordine di accesso, e l'elemento più recente occupa la posizione più esposta. Questo processo è talmente pervasivo da attraversare scale di grandezza e ambiti di conoscenza difficilmente comparabili. Il modello è tuttavia riconoscibile ovunque, con la stessa geometria: qualcosa viene aggiunto sopra, o fuori, o davanti, e per raggiungere ciò che giace in profondità bisogna attraversare a ritroso tutto ciò che era stato depositato strato dopo strato.

Quando uno studente si imbatte per la prima volta nella definizione formale di una pila, riceve un nome e una notazione per qualcosa che ha già incontrato innumerevoli volte, in contesti che non avrebbe mai pensato di collegare. Il compito dell'insegnante è quello di rendere visibile una struttura già presente nell'esperienza dello studente, che attende solo di essere riconosciuta (Polya, 1945).

Le sezioni che seguono analizzano alcune delle manifestazioni più significative di questo modello, scelte per la diversità dei campi che attraversano e per la luce che ciascuna di esse getta su un diverso aspetto della struttura che stiamo descrivendo.

## **3. Le metafore, una per una**

### **3.1 Archeologia**

Quando un archeologo avvia uno scavo, il tempo gli viene incontro in ordine inverso. I reperti più vicini alla superficie appartengono alle civiltà più recenti; quelli più profondi, a quelle più antiche. Ogni strato è sigillato da quello sovrastante, e l'accesso a ciascuno di essi richiede la rimozione di tutti quelli depositati successivamente. Non è possibile estrarre un manufatto romano senza prima attraversare gli strati medievali, rinascimentali e moderni che lo ricoprono.

Questa sequenza non è una scelta metodologica dell'archeologo: è una proprietà fisica del terreno stratificato. L'ordine di deposito, determinato dal tempo, diventa l'ordine inverso di accesso. È la definizione stessa del LIFO, scritta nel terreno millenni prima che qualcuno pensasse di formalizzarla.

L'archeologia offre inoltre un'ulteriore intuizione che i libri di testo raramente rendono esplicita: la stratificazione non distrugge le informazioni, ma le conserva nell'ordine in cui sono state prodotte. Ogni strato è un'istantanea del momento in cui è stato depositato. La sequenza di estrazione, se condotta correttamente, è una macchina del tempo che percorre la storia a ritroso.

### **3.2 Il sistema nervoso centrale**

Il neurologo americano Paul MacLean propose negli anni '60 il modello del cervello triunale: una struttura in cui tre sistemi sovrapposti – il cervello rettiliano, il sistema limbico e la neocorteccia – riflettono altrettante fasi dell'evoluzione dei vertebrati (MacLean, 1990). Le strutture più antiche, che regolano le funzioni vitali di base, si trovano al centro e in profondità. Quelle più recenti, sede delle funzioni cognitive superiori, le avvolgono dall'esterno.

Il modello di MacLean è stato parzialmente rivisto dalla neurobiologia contemporanea, che ne ha messo in discussione la rigidità, ma conserva la sua validità descrittiva come metafora dell'accumulo evolutivo. Ciò che è venuto prima si trova in basso; ciò che è venuto dopo si trova in alto. Per raggiungere le strutture arcaiche bisogna attraversare, almeno anatomicamente, quelle più recenti. La stratificazione evolutiva produce una geometria che qualsiasi informatico riconoscerebbe immediatamente.

### **3.3 La stratificazione freudiana**

Sigmund Freud ha costruito il suo modello della psiche su un'esplicita metafora spaziale: i contenuti repressi affondano nell'inconscio, sepolti sotto strati successivi di vita psichica (Freud, 1899). Il lavoro analitico è un lavoro di scavo, e lo stesso Freud ha utilizzato più volte l'analogia archeologica per descriverlo. Il terapeuta rimuove strato dopo strato il materiale psichico più recente per raggiungere i nuclei profondi, che sono anche i più antichi e i più resistenti.

La direzione è la stessa che si riscontra nell'archeologia e nel cervello triunale: il tempo scorre verso il basso, e risalire verso l'origine significa attraversare a ritroso tutto ciò che era stato depositato strato dopo strato al di sopra. Freud non stava descrivendo una pila, ma utilizzava la stessa geometria.

Un proverbio napoletano, «'A capa è 'na sfoglia 'e cipolla» — la testa è uno strato di cipolla — coglie con popolare concisione la stessa intuizione: la mente è costituita da strati concentrici, ognuno dei quali nasconde quello sottostante, e non si può raggiungere il centro senza passare attraverso quelli esterni. Il proverbio non ha bisogno di Freud per essere vero, ed è probabilmente più antico di lui. È anche, come vedremo, una descrizione sorprendentemente accurata di certi sistemi informatici.

### 3.4 Il modello ISO-OSI e l'incapsulamento

Il modello ISO-OSI descrive l'architettura delle reti di comunicazione come una sequenza di sette livelli, ciascuno dei quali aggiunge un involucro ai dati che scendono lungo la pila e lo rimuove man mano che risalgono (Tanenbaum, 2003). In trasmissione, ogni livello aggiunge la propria intestazione; in ricezione la rimuove in ordine inverso, fino a consegnare i dati originali all'applicazione.

In questo caso il principio LIFO non riguarda l'ordine temporale di deposito, bensì l'ordine logico di incapsulamento. La simmetria tra trasmissione e ricezione è perfetta, e non è casuale: si tratta di una proprietà strutturale di qualsiasi sistema a livelli in cui ogni livello ignora ciò che si trova al di sopra e al di sotto di esso.

### 3.5 Macchine virtuali e la struttura a cipolla

La metafora della cipolla, evocata dal proverbio napoletano, si rivela sorprendentemente azzeccata per descrivere l'architettura dei sistemi a livelli. Andrew Tanenbaum, nel descrivere la struttura dei sistemi operativi, utilizza il termine "macchina virtuale" in un senso preciso e più generale rispetto a quello corrente: ogni livello del sistema espone al livello superiore un'interfaccia che astrae tutto ciò che si trova al di sotto, creando l'illusione di una macchina dotata di determinate capacità (Tanenbaum, 2013). L'utente del livello N non sa e non ha bisogno di sapere in che modo il livello N-1 realizzi concretamente ciò che gli viene offerto. In questo senso ogni livello è una macchina virtuale per il livello superiore.

Come per gli strati di una cipolla e quelli della psiche freudiana, non è possibile raggiungere il centro senza attraversare quelli esterni. La differenza rispetto alla cipolla biologica è tuttavia significativa: gli strati di una cipolla crescono tutti insieme, mentre quelli di un sistema a strati sono definiti in sequenza, e tale sequenza determina l'ordine in cui possono essere attraversati. La geometria concentrica della cipolla si combina con la temporalità della pila.

Vale la pena notare che il termine «macchina virtuale» è utilizzato anche, in un senso più ristretto e tecnologicamente più recente, per riferirsi a software come VMware o VirtualBox, che emulano un intero ambiente hardware. Si tratta di un caso di omonimia: il concetto è diverso, anche se la parola è la stessa.

### 3.6 L'ascensore stretto

Tra tutti gli esempi raccolti, l'ascensore è l'unico in cui il comportamento LIFO non deriva da un'inevitabile proprietà fisica o biologica, ma da una scelta razionale e deliberata. In un ascensore di piccole dimensioni, dove non è possibile cambiare posizione una volta entrati, è conveniente che entrino per primi coloro che devono scendere ai piani più alti: in questo modo, quando arriva il momento, chi deve scendere prima si trova già vicino alla porta.

Nessuna legge fisica impone questa regola. Se le persone entrano nell'ordine sbagliato, l'ascensore funziona comunque, ma chi deve scendere per primo si ritrova intrappolato in fondo. Il LIFO, in questo caso, è una norma di comportamento collettivo che ottimizza l'uso di una risorsa condivisa. Vale la pena notare che questa è anche la situazione più vicina ad alcuni problemi di schedulazione nei sistemi operativi, dove l'ordine di esecuzione dei processi viene scelto deliberatamente per massimizzare l'efficienza (Tanenbaum, 2015).

### **3.7 Il protocollo di decontaminazione in una camera sterile**

In ambito medico, il personale che opera in zone ad alto rischio biologico indossa una serie di dispositivi di protezione individuale (DPI) secondo un ordine preciso e prestabilito: prima i dispositivi a contatto con il corpo, poi quelli che proteggono le vie di accesso più esposte e, infine, quelli che proteggono le mucose. Ogni dispositivo aggiunge uno strato di protezione a quello precedente.

La rimozione dei DPI al termine dell'intervento deve seguire rigorosamente l'ordine inverso. La ragione è di natura biologica: la superficie esterna di ciascun dispositivo è potenzialmente contaminata e rimuovere un dispositivo interno prima di uno esterno significa esporre le mani o il viso alla contaminazione di quest'ultimo durante le manovre successive. Episodi documentati di infezione da Ebola nel personale sanitario sono stati ricondotti proprio a errori nella sequenza di rimozione (OMS, 2014).

Tra tutti gli esempi raccolti in questa sezione, il protocollo di decontaminazione è quello in cui la proprietà LIFO cessa di essere una comodità o una proprietà emergente e diventa un vincolo di sicurezza biologica. La violazione dell'ordine di estrazione non produce un errore logico rilevato da un compilatore: produce un rischio per la vita dell'operatore.

## **4. La Torre di Hanoi come caso speciale**

Tra tutti gli esempi raccolti, la Torre di Hanoi occupa una posizione particolare. Negli altri casi il modello LIFO emerge in fenomeni che esistono indipendentemente dall'informatica, e l'informatica li riconosce e li formalizza. Nella Torre di Hanoi il rapporto è invertito: il problema è stato deliberatamente costruito come un rompicapo matematico, ma la sua soluzione rivela che lo stack è la struttura profonda del problema stesso, non uno strumento scelto per comodità di implementazione.

Il rompicapo, ideato dal matematico francese Édouard Lucas nel 1883 (Lucas, 1883), consiste nel trasferire una torre di dischi di diametro decrescente da un perno di partenza a un perno di destinazione, utilizzando un perno ausiliario, con il vincolo che un disco di diametro maggiore non possa mai essere posizionato sopra uno di diametro minore. Il numero minimo di mosse necessarie per trasferire una torre di  $n$  dischi è  $2^n - 1$ .

La soluzione ricorsiva è elegante e ben nota: per spostare  $n$  dischi dal piolo A al piolo C utilizzando B come piolo ausiliario, si spostano prima i  $n-1$  dischi superiori da A a B, poi si sposta il disco più grande da A a C, quindi si spostano i  $n-1$  dischi da B a C. La ricorsione si applica a ciascuno dei due sottoproblemi fino a raggiungere il caso base, ovvero un singolo disco, che viene spostato direttamente (Cormen et al., 2009).

Ciò che questa descrizione non rende immediatamente evidente è che ogni chiamata ricorsiva aggiunge un frame allo stack delle chiamate e che la sequenza di esecuzione segue esattamente il principio LIFO: l'ultima chiamata inserita nello stack è la prima ad essere risolta. Un disco non può essere spostato finché tutti quelli impilati sopra di esso non sono stati rimossi: è la definizione stessa di stack, applicata a oggetti fisici di legno o avorio.

La soluzione iterativa del problema della Torre di Hanoi richiede l'uso di uno stack esplicito per simulare ciò che la ricorsione fa implicitamente. Eliminare la ricorsione non elimina lo stack: lo rende semplicemente visibile. Questa è una dimostrazione concreta di un principio che vale in generale: la ricorsione e lo stack sono due facce della stessa struttura, e ogni algoritmo ricorsivo porta al suo interno uno stack, indipendentemente dal fatto che esso emerga o meno nel codice (Knuth, 1968).

## **5. Implicazioni didattiche**

### **5.1 L'ordine di presentazione**

La prassi consolidata prevede che lo stack venga presentato dopo che lo studente ha già acquisito una certa familiarità con la programmazione: viene definita la struttura, ne vengono descritte le operazioni fondamentali e vengono illustrate alcune applicazioni. Questo ordine ha una sua logica, ma chi studia seguendo questa sequenza rischia di apprendere il meccanismo senza coglierne il significato: sa come usare push e pop senza avere un'immagine mentale solida a cui aggrapparsi, conosce la struttura senza capire perché esiste.

Un ordine alternativo consiste nell'invertire la sequenza. Le metafore discusse nelle sezioni precedenti rendono questo approccio fattibile anche con studenti che non hanno ancora affrontato il concetto formale di pila, poiché attingono alle esperienze e alle conoscenze che tali studenti già possiedono.

### **5.2 La discussione in classe come punto di partenza**

Prima di avviare un ambiente di sviluppo, agli studenti viene chiesto di elencare situazioni tratte dalla vita quotidiana o dal proprio percorso di studi in cui qualcosa si accumula per stratificazione e deve essere recuperato in ordine inverso. La discussione che ne segue porta quasi invariabilmente a esempi che coprono l'intero spettro esplorato in questo articolo: qualcuno cita l'archeologia, qualcuno i livelli di rete studiati in un corso precedente, qualcuno il proprio zaino. Il docente raccoglie e organizza le intuizioni già in possesso degli studenti. Solo a questo punto viene definito lo stack, formalizzata la proprietà LIFO e scritto il codice. Il formalismo si presenta come il punto di arrivo di un percorso che lo studente ha già percorso, non come un punto di partenza che richiede uno sforzo di astrazione partendo da zero.

Questa sequenza presenta un ulteriore vantaggio che riguarda la Torre di Hanoi. Quando il rompicapo viene proposto come esercizio dopo che lo studente ha già interiorizzato lo schema LIFO attraverso le metafore,

la soluzione ricorsiva non appare come un espediente matematico da memorizzare. Appare invece come la trasposizione naturale di qualcosa che lo studente riconosce: per spostare il disco più grande bisogna prima rimuovere tutti quelli che si trovano sopra di esso, esat e come in archeologia, esattamente come nella rimozione dei DPI in una sala sterile. La ricorsione diventa comprensibile perché lo schema che essa implementa è già familiare (Polya, 1945).

### **5.3 Il dominio di applicazione come contesto motivante: il caso dei sistemi medici**

Le metafore di uso quotidiano hanno un immediato potere evocativo, ma rischiano di rimanere a livello di semplice illustrazione se non sono ancorate all'ambito professionale a cui è rivolto il corso. Il protocollo di decontaminazione descritto nella sezione 3.7 offre questa opportunità in modo particolarmente diretto: per uno studente di Ingegneria dei sistemi medici non si tratta semplicemente di un altro esempio, ma di un contesto in cui un giorno potrebbe trovarsi a progettare o certificare un vero e proprio sistema informatico (OMS, 2014).

Proporre la modellizzazione di quel protocollo come esercizio di programmazione, con una classe PPE, una classe VestitionProtocol che gestisce una LinkedList utilizzata come stack e un metodo verifyRemoval che segnala le violazioni della sequenza, significa chiedere allo studente di tradurre in codice qualcosa che ha già compreso a livello clinico. La struttura dati viene riconosciuta come lo strumento naturale per modellare un vincolo biologico reale, e lo stack cessa di essere una convenzione da accettare per diventare la risposta a una domanda posta dal dominio stesso.

## **6. Conclusione**

William Butler Yeats, in una poesia breve e densa, immagina una donna che si trucca davanti allo specchio non per vanità, ma per riconoscimento: sta cercando, al di là delle convenzioni che il tempo ha depositato, il volto che aveva prima che il mondo fosse creato (Yeats, 1933). Il gesto è l'opposto di ciò che sembra: non aggiunge, ma scava.

Chi formalizza una struttura dati compie un gesto analogo. Lo stack non è stato inventato da chi ha scritto il primo compilatore: era già presente nel terreno che l'archeologo scava, nell'architettura del sistema nervoso, nella geometria della psiche, nella sequenza rigorosa con cui un operatore sanitario si toglie i dispositivi di protezione in una sala sterile. L'informatica lo ha scoperto, lo ha riconosciuto per quello che era e gli ha dato un nome e un'interfaccia. Ha trovato il volto che era già lì prima che il mondo fosse creato.

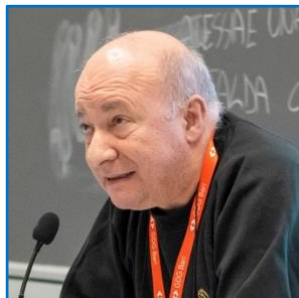
Ciò non riduce l'informatica al ruolo di reinventare la ruota. La formalizzazione non è una scoperta passiva: è un atto creativo con conseguenze che il modello originale non conteneva. Il suolo stratificato esiste da miliardi di anni, ma nessun geologo aveva mai pensato di utilizzare la sua logica per gestire le chiamate ricorsive di un algoritmo (Knuth, 1968). La psiche freudiana aveva la geometria di una pila, ma Freud non

avrebbe potuto scrivere un sistema operativo. Il valore aggiunto dell'informatica risiede proprio in questo trasferimento: prendere un modello antico, trasferirlo in un nuovo ambito e scoprire che lì produce risultati impensabili nell'ambito originario.

La donna di Yeats, dopo aver ritrovato il volto originario, non lo tiene per sé: vuole che il suo amato ami quella cosa, quella che esisteva prima che il mondo fosse creato. È un atto di restituzione. L'informatica fa qualcosa di simile: prende un modello che la realtà aveva già prodotto, lo formalizza e lo restituisce al mondo sotto forma di strumento. Con quello strumento si costruiscono compilatori, sistemi operativi, protocolli di rete, software per la gestione delle camere bianche. Il modello era antico; le cose che rende possibili sono nuove. La questione di quante altre strutture dati abbiano una storia naturale simile a quella dello stack rimane aperta, e vale la pena lasciarla aperta. La coda, la lista, l'albero, il grafo: ciascuno di essi ha probabilmente una propria galleria di manifestazioni pre-informatiche, una propria serie di ambiti in cui il modello emerge prima ancora di essere denominato. Esplorarle con lo stesso metodo applicato in questo articolo costituisce un programma di ricerca didattica che il presente testo non esaurisce ma, si spera, contribuisce ad avviare.

## Bibliografia

- Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C. (2009). *Introduction to Algorithms*, 3rd ed. MIT Press, Cambridge (MA).
- Freud, S. (1899). *Die Traumdeutung*. Franz Deuticke, Leipzig-Wien. [Engl. transl.: *The Interpretation of Dreams*, Basic Books, New York, 1955.]
- Knuth, D.E. (1968). *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*. Addison-Wesley, Reading (MA).
- Lucas, É. (1883). *Récréations Mathématiques*. Gauthier-Villars, Paris.
- MacLean, P.D. (1990). *The Triune Brain in Evolution: Role in Paleocerebral Functions*. Plenum Press, New York.
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York.
- Polya, G. (1945). *How to Solve It*. Princeton University Press, Princeton (NJ).
- Tanenbaum, A.S. (2003). *Computer Networks*, 4th ed. Prentice Hall, Upper Saddle River (NJ).
- Tanenbaum, A.S. (2013). *Structured Computer Organization*, 6th ed. Pearson, Upper Saddle River (NJ).
- Tanenbaum, A.S., Bos, H. (2015). *Modern Operating Systems*, 4th ed. Pearson, Upper Saddle River (NJ).
- WHO — World Health Organization (2014). *Guidance on Personal Protective Equipment in the Context of Filovirus Disease Outbreak Response*. WHO Press, Geneva.
- Yeats, W.B. (1933). *The Winding Stair and Other Poems*. Macmillan, London.



**Pierangelo Indolfi**

[indolfi@gmail.com](mailto:indolfi@gmail.com)

Pierangelo Indolfi tiene il corso di Laboratorio di Progettazione Software per gli studenti di Ingegneria dei Sistemi Medici presso il Politecnico di Bari. Con oltre trent'anni di esperienza nella formazione ICT nelle scuole secondarie tecniche, è stato istruttore della Cisco Networking Academy, supervisore ICDL e possiede certificazioni in Arduino, Raspberry Pi e competenze digitali (DigCompEdu C2). È vincitore del MIT App Inventor Foundation Teacher Excellence Award 2023. I suoi interessi di ricerca includono la didattica dell'informatica, le strutture dati astratte e i fondamenti epistemologici dell'educazione informatica.