

# INFORMATICA, MODELLI MENTALI E RAGIONAMENTO PROFONDO

ENRICO NARDELLI

*Università di Roma «Tor Vergata»*

Ho accolto con piacere l'invito a commentare l'articolo bersaglio di Bucciarelli (2019) sull'importanza di sviluppare il ragionamento profondo nella scuola. Il dilagare di dispositivi e strumenti digitale pone infatti la società contemporanea di fronte ad un sfida educativa senza precedenti. Quell'intermediazione tra produttori e consumatori di informazioni che veniva fino a pochi anni orsono gestita da un insieme di soggetti autorevoli e selezionati, in grado quindi di assicurare qualità ed affidabilità, sta ormai scomparendo. Nell'infosfera digitale chiunque può immettere contenuti, che vengono poi ritrovati da motori di ricerca e forniti in risposta alle richieste ad essi attinenti. È secondo me vano sperare che si possano attuare dei meccanismi più o meno automatici di filtraggio di qualità su quanto immesso in rete, anche considerando (o meglio, nonostante) l'impetuosa crescita delle tecniche di apprendimento automatico esibita dall'intelligenza artificiale. La scuola deve quindi, ancora di più che in passato, allenare al ragionamento profondo, perché sempre di più la determinazione dell'autorevolezza di una sorgente sarà il risultato di una decisione autonoma di chi legge. Come argomenterò nel seguito commentando l'articolo bersaglio, penso che lo studio nella scuola dell'informatica, intesa soprattutto come disciplina scientifica più che come abilità tecnologica, possa offrire un valido contributo per questo obiettivo.

Il modello mentale fondamentale dell'informatica è quello dell'automa, cioè dell'esecutore di algoritmi. Una sua componente fondamentale è l'insieme di istruzioni che è in grado di comprendere e quindi di eseguire. Solo algoritmi espressi mediante le istruzioni che fanno parte del repertorio di un automa potranno essere eseguiti da esso (Nardelli, 2019b). Non abbiamo, in questo caso, soltanto «una rappresentazione iconica della struttura degli stati di cose» ma un modello totalmente dinamico, in cui la capacità di rappresentare mentalmente lo stato corrente del sistema e di ragionare sulle sue possibili traiettorie future diventa essenziale. Ritengo che la capacità di simulare mentalmente la dinamica dell'esecutore possa contribuire a raf-

forzare lo sviluppo del ragionamento profondo e penso che sia opportuno investigare sul campo se esiste una correlazione positiva tra competenza informatica e capacità di ragionamento profondo.

Ricordiamo che l'espressione di un algoritmo mediante le istruzioni che l'automa «comprende» costituisce il programma informatico che l'automa esegue. La ricerca sulle metodologie per l'insegnamento della programmazione informatica ha introdotto il concetto di *notional machine* per indicare le caratteristiche – da un punto di vista comportamentale – della macchina che si sta imparando a controllare (du Boulay, 1986). Una *notional machine*, in altre parole, è la rappresentazione in un'ottica pedagogica del concetto teorico di esecutore, cioè dell'automa, e della sua semantica informatica. Diversi studi hanno rilevato che l'aver interiorizzato in modo corretto la *notional machine* di riferimento per il linguaggio di programmazione che si sta usando costituisce un'abilità fondamentale per lo sviluppo delle competenze informatiche. Per una rassegna si vedano Sorva (2013) e Lewis (2019).

L'eseguibilità diretta del modello mentale (che dovrebbe più propriamente essere chiamato meta-modello, dal momento che i veri modelli sono poi gli algoritmi che vengono eseguiti) costituisce quella che ho chiamato «la grande bellezza dell'informatica» (Nardelli, 2017b) e la sua grandissima utilità dal punto di vista pedagogico, dal momento che consente di agire la capacità di falsificare le ipotesi. Come rilevato dal matematico francese Cédric Villani (vincitore della medaglia Field, definito informalmente come «il premio Nobel della matematica»), l'informatica è infatti pressoché l'unica materia in cui lo studente può ottenere facilmente da solo la verifica alla propria ipotesi (Villani, 2015). Oltre che essere pedagogicamente interessante di per sé, nell'ottica di quell'allenamento al ragionamento deliberato che molto giustamente l'autrice identifica come una delle missioni fondamentali della scuola, lo sviluppo di tale abilità costituisce una componente fondamentale di quell'esercizio al «pensiero critico» che sempre di più è necessario nel diluvio informativo della società contemporanea (Nardelli, 2016a). Ma non solo. L'utilità pedagogica dell'informatica dipende anche dalla sua capacità di rappresentare qualunque realtà si voglia esplorare o inventare, definendo un insieme di regole e dando loro vita mediante un'esecuzione su di un appropriato dispositivo automatico. Le sue fenomenali capacità di elaborare simboli di qualunque tipo ed in qualunque modo, costituiscono la sua forza interdisciplinare in un contesto didattico. Con l'informatica è infatti possibile costruire una simulazione di fenomeni fisici, biologici o sociali di cui si conoscono (anche in modo parziale ed approssimato) le leggi, osservare cosa accade e «cosa accadrebbe se...». È lampante il valore educativo di un approccio «costruttivista» di questo tipo,

perché niente è utile per apprendere un concetto come farne esperienza concreta, ed è questa caratteristica che rende, appunto, l'informatica una disciplina di valore trasversale, da insegnare quindi non soltanto di per sé ma anche come ausilio didattico per approfondire la comprensione di altre discipline (Nardelli, 2017c; Caspersen, Gal-Ezer, McGettrick e Nardelli, 2019).

Risulta molto interessante l'osservazione sperimentale, rilevata nelle ricerche condotte dall'autrice, sui bambini che riescono a formulare con migliore precisione gli algoritmi quando accompagnano con gesti la loro formulazione. Mi risuona con quanto ho letto in diverse pubblicazioni scientifiche di psicologia cognitiva che riconoscono l'importanza della scrittura manuale come meccanismo per rafforzare l'apprendimento e, più in generale, osservano il miglioramento nella capacità di memorizzazione e ragionamento che si ottiene quando viene coinvolto anche il corpo. Ritengo, anche sulla base di ciò, che l'apprendimento dell'informatica nella scuola primaria dovrebbe avere una forte base di concretezza. In quest'ottica, ad esempio, imparare a formulare algoritmi lavorando direttamente su oggetti fisici quali i vagoni e i binari degli studi della «ferrovia giocattolo» descritti dall'autrice **costituisca** una strategia certamente valida. Sarebbe meritorio cercare di allargare il più possibile il corredo degli strumenti didattici non digitali a supporto dell'insegnamento **dell'informatico**. È mia ferma convinzione che in questa fascia di scuola un insegnamento «concreto» dei principi dell'informatica (cioè basato il più possibile su oggetti fisici) sia un passaggio obbligato ed aiuti a formare una corretta visione dell'informatica come scienza dell'automazione della computazione, indipendentemente dal substrato fisico con cui l'automata è realizzato. Inoltre, va osservato che quello che ho precedentemente descritto come un singolo meta-modello (l'automata) è in realtà l'esempio più noto (l'automata procedurale) di un insieme di meta-modelli di diversa natura. Sono possibili, infatti, automi logici, funzionali, reattivi, orientati agli oggetti. Una questione su cui la ricerca in didattica dell'informatica è ancora nelle fasi iniziali è quale sia il meta-modello più adeguato per le varie fasi dello sviluppo cognitivo dello studente.

L'articolo bersaglio tocca anche il tema dell'apprendimento della ricorsione. In informatica il termine «ricorsione» denota un concetto tecnico ben preciso, quello di una funzione che calcola un risultato mediante l'invocazione di sé stessa su un insieme di dati di dimensione inferiore a quello ricevuto in ingresso. Ritengo importante precisare che non esiste, in informatica, un concetto «rilassato» o meno ristretto di ricorsione. L'informatica, come ogni altra scienza esatta, vive della non ambiguità semantica dei suoi termini. Ritengo quindi che parlare di una «nozione più ampia di ricorsione» per intendere

la specifica di un *loop* di operazioni rischi di creare incomprensione metodologica per le ricerche sul campo (e, semmai, sarebbe vero che è la ricorsione a rappresentare una forma più generale del *loop*). D'altro canto, l'osservazione dell'autrice secondo cui mediante la ricorsione viene creato un *loop* è certamente corretta ed interessante da utilizzare per studiare le modalità mediante le quali la ricorsione (nel senso tecnico dell'informatica) viene appresa. Sarebbe infatti utile studiare se esiste una correlazione positiva tra la competenza nell'uso dei *loop* (e dei *loop* annidati) e la capacità di scrivere correttamente algoritmi ricorsivi (competenza che l'autrice giustamente osserva è difficile da padroneggiare anche per gli adulti: elemento noto in letteratura e confermato empiricamente sia dalla mia personale esperienza didattica che da quella dei miei colleghi informatici). Ritengo – dal mio punto di vista tecnico – che elementi importanti per la comprensione della ricorsione siano anche un'estesa memoria di lavoro (necessaria, a mio avviso, per tener traccia dello stato corrente dell'algoritmo, di come tale stato è legato a quello precedente ed influenza il successivo) e la capacità di ragionamento astratto (coerentemente – ad esempio – con la pratica dell'insegnamento della matematica in cui si inizia l'introduzione dell'algebra, cioè dell'astrazione dell'aritmetica, soltanto nella fascia 11-13 anni, quella delle scuole secondarie inferiori). Anche in questo caso sarebbe necessaria un'approfondita ricerca sul campo per comprendere meglio la situazione.

Nell'ambito del gruppo di lavoro (costituito da molti colleghi universitari di informatica) su «Informatica e Scuola», istituito dal CINI – Consorzio Interuniversitario Nazionale per l'Informatica – e da me coordinato, abbiamo preparato una proposta di curriculum per l'insegnamento dell'informatica nella scuola<sup>1</sup>, pubblicata anche a convegno internazionale (Forlizzi *et al.*, 2018). Il curriculum è stato costruito in base alle migliori esperienze internazionali, tra le quali una posizione di preminenza spetta alle indicazioni del «Curriculum on Computing» obbligatorio per le scuole di ogni livello del Regno Unito a partire dal settembre 2014 (UK Government, 2013), e basato sulla pratica del nostro lavoro nella scuola in collaborazione con insegnanti e pedagogisti. Ovviamente, essendo l'informatica una disciplina la cui didattica nella scuola, diversamente da scienze di più lunga storia, è ancora all'inizio del suo percorso, non vi è quell'accumulo di studi e sperimentazioni che ha permesso di capire quali sono le migliori metodologie e gli approcci più efficaci per l'insegnamento dei concetti fondamentali. Sarà quindi necessaria nei prossimi anni una

<sup>1</sup> Liberamente scaricabile dalla pagina web del Gruppo di Lavoro CINI su «Informatica e Scuola», <http://consorzio-cini.it/gdl-informatica-scuola>.

forte collaborazione tra informatici, psicologi e pedagogisti, come avviene nell'ambito del progetto «Programma il Futuro» per sperimentare l'introduzione dell'insegnamento dell'informatica nella scuola<sup>2</sup> (Corradini, Lodi e Nardelli, 2017a, 2017b, 2018; Corradini e Nardelli, 2018). Ciò che in altre discipline si è potuto studiare con calma nel corso dei decenni, deve essere affrontato in tempi molto più stretti, per l'informatica, pena il rischio di costruire percorsi di apprendimento squilibrati ed inefficaci.

## BIBLIOGRAFIA

- BUCCIARELLI M. (2019). Imparare a ragionare... e continuare a farlo. *Giornale Italiano di Psicologia*, 46 (4), ???
- CASPERSEN M.E., GAL-EZER J., MCGETTRICK A., NARDELLI E. (2019). Informatics as a fundamental discipline for the 21st Century. *Communications of the ACM*, April 2019, 62 (4), 58.
- CORRADINI I., LODI M., NARDELLI E. (2017a). *Computational thinking in Italian schools: Wuantitative data and teachers' sentiment analysis after two years of «Programma il Futuro»*. Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE '17. Bologna, Italy: ACM.
- CORRADINI I., LODI M., NARDELLI E. (2017b). *Conceptions and misconceptions about computational thinking among Italian primary school teachers*. Proceedings of the 2017 ACM Conference on International Computing Education Research. ICER '17. Tacoma, Washington, USA: ACM, pp. 136-144.
- CORRADINI I., LODI M., NARDELLI E. (2018). *An investigation of Italian primary school teachers' view on coding and programming*. ISSEP. Vol. 11169. Springer Lecture Notes in Computer Science, pp. 228-243.
- CORRADINI I., NARDELLI E. (2018). *Awareness in the online use of digital technologies of Italian students*. Proceedings of the 11th International Conference of Education, Research and Innovation. Sevilla, Spain.
- DU BOULAY B. (1986). Some difficulties of learning to program. *Journal of Educational Computing*, 2 (1) 57-73.
- FORLIZZI L., LODI M., LONATI V., MIROLO M., MONGA M., MONTRESOR A., MORPURGO A., NARDELLI E. (2018). *A core informatics curriculum for Italian compulsory education*. 11th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives. St. Petersburg, Russia. Springer: Lecture Notes in Computer Science vol. 11169.
- LEWIS C.M., CLANCY M.J., VAHRENHOLD J. (2019). Student knowledge and misconceptions. In S.A. Fincher, A.V. Robins (eds.), *The Cambridge handbook of computing education research*. Cambridge: Cambridge University Press.
- NARDELLI E. (2016a). Gli studenti dell'era digitale: automi o pensatori critici? *Il Fatto Quotidiano*, <https://www.ilfattoquotidiano.it/2016/04/21/gli-studenti-dellera-digitale-automi-o-pensatori-critici/2654546/>.

<sup>2</sup> <http://programmailfuturo.it>

- NARDELLI E. (2017b). La «grande bellezza dell'informatica». *Link&Think*, <http://link-and-think.blogspot.com/2017/05/la-grande-bellezza-dellinformatica.html>.
- NARDELLI E. (2017c). Informatica nella scuola: disciplina fondamentale e trasversale, ovvero «di cosa parliamo quando parliamo di pensiero computazionale». *Scienze e Ricerche Magazine*, 47, 36-40.
- NARDELLI E. (2019a). Do we really need computational thinking? *Communications of the ACM*, 62 (2), 32-35.
- NARDELLI E. (2019b). Informatica e cittadino digitale: dal *coding* al *computational thinking*. In C. Petracca (a cura di), *Per un'idea di Scuola*. Teramo: Lisciani Editore, pp. 145-166.
- SORVA J. (2013). Notional machines and introductory programming education. *ACM Transactions on Computing Education*, 13 (2), art. 8.
- UK GOVERNMENT (2013). National curriculum in England: Computing programmes of study. Department of Education. <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study>.
- VILLANI C. (2015). *La programmation est – presque – la seule discipline où l'enfant réalise son auto-correction*. Intervista di Lila Meghraoua, Luglio 2015, <https://atelier.bnpparibas/life-work/article/cedric-villani-la-programmation-seule-discipline-enfant-realise-auto-correction>.

### Informatics, mental model, and deep reasoning

*La corrispondenza va inviata a Enrico Nardelli, Dipartimento di Matematica, Università di Roma «Tor Vergata», Via della Ricerca Scientifica 1, 00133 Roma. E-mail: [nardelli@mat.uniroma2.it](mailto:nardelli@mat.uniroma2.it)*