

Informatica nella scuola: disciplina fondamentale e trasversale, ovvero “di cosa parliamo quando parliamo di pensiero computazionale”

ENRICO NARDELLI

Dipartimento di Matematica, Università degli Studi di Roma “Tor Vergata”

In questo articolo discuto l'importanza di introdurre fin dalla scuola primaria l'insegnamento dei concetti di base dell'informatica, per sviluppare negli studenti il cosiddetto “pensiero computazionale”. Dopo aver analizzato il significato di questo termine e le motivazioni per il suo uso, la discussione analizza l'importanza dell'informatica nel contesto dell'istruzione scolastica, sia come disciplina scientifica in sé che di valore trasversale rispetto alle altre. Esperienze di altri paesi, europei e non, sono presentate a supporto della tesi.

Inizio spiegando il perché del titolo: si tratta di un omaggio a Raymond Carver, scrittore americano che è considerato l'iniziatore della corrente del minimalismo [1]. In questo articolo provo a spiegare con parole semplici (come, appunto, si dice Carver facesse nelle sue opere) cosa intendiamo quando parliamo di pensiero computazionale e le relazioni di questo termine con l'informatica.

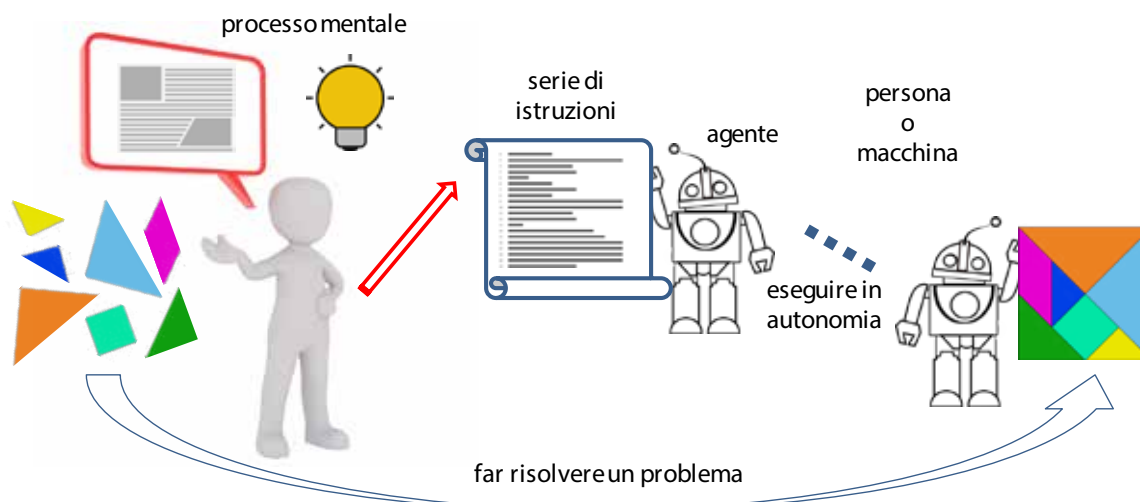
1. COSA SI INTENDE CON “PENSIERO COMPUTAZIONALE”

Inizio chiarendo che, sinteticamente, col termine “pensiero computazionale” indichiamo **gli aspetti culturali e scienti-**

fici dell'informatica, a prescindere da qualunque aspetto strumentale o tecnologico. In altre parole, possiamo dire che è ciò che rimane nelle persone in termini di competenze ed abilità dopo aver studiato (e praticato) adeguatamente l'informatica.

Il termine era stato introdotto da Seymour Papert, ma è stato popolarizzato da Jeannette Wing con un citatissimo articolo del 2006 [12]. In quest'articolo la Wing non diede una definizione completa, che in successivi interventi si è per lei stabilizzata nella seguente: «*Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent*». Tale definizione è presente in [13] ed è attribuita a un “work in progress” di Jan Cuny, Larry Snyder, and Jeannette Wing, di cui però a tutt'oggi non c'è traccia in letteratura. Per una rassegna sulla definizione di pensiero computazionale si possono consultare [14] e [15]. Il dibattito su quale sia l'esatta definizione di “*computational thinking*” è ancora aperto e a tutt'oggi si contano, nella sola Digital Library dell'ACM più di 400 articoli scientifici che ne trattano.

Probabilmente a causa della sua novità, il **pensiero com-**



putazionale viene molte volte descritto in modo parziale e quindi impreciso. Per chiarezza presento e discuto qui la rielaborazione di una mia precedente definizione: “*il pensiero computazionale è un processo mentale per far risolvere problemi ad un agente, sia esso persona o macchina, fornendogli una serie di istruzioni che deve eseguire in autonomia*”, illustrata graficamente nel disegno della pagina precedente.

Tale definizione esplicita, in un contesto di divulgazione, tutte e sole le componenti essenziali, senza le quali il pensiero computazionale diventa altro. Nel discutere in dettaglio i vari termini segnalerò, in qualche caso, alcune semplificazioni che ne alterano la natura.

processo mentale: Specifica il concetto sottolineando la sua indipendenza da tecnologie, strumenti concreti e sistemi fisici. Identifica pertanto la natura eminentemente metodologica del pensiero computazionale, che è applicabile in molti ambiti disciplinari ed è riconosciuto come strumento concettuale per la conoscenza del mondo da scienziati come Peter J. Denning: «*Computing is the fourth great domain of science. It is as fundamental as the physical, life, and social sciences*» [17]. Inoltre, lungi dal trasformare gli studenti in robot, li aiuta a sviluppare il pensiero critico, cioè a «*non fermarsi all'apparenza dei fenomeni ma ... a chiedersi cosa ci sia dietro*» [18].

far risolvere problemi: Un errore fondamentale è descrivere il pensiero computazionale unicamente come la competenza che permette di risolvere problemi (*problem solving*). Se così fosse non sarebbe diverso dalle competenze di tipo logico-matematico. I matematici hanno risolto problemi per millenni. Ma solo quando hanno iniziato a riflettere sulla possibilità e sulle implicazioni di “far risolvere” i problemi a qualcun altro, si è messa in moto un'evoluzione che ha portato alla nascita della nuova disciplina scientifica, l'informatica. Correlata a questa semplificazione vi è quella di ritenere che l'essenza del pensiero computazionale sia decomporre il problema in sotto-problemi, per poi ricombinarli. Ma molte volte non è richiesto questo approccio, che non è caratterizzante per il pensiero computazionale.

ad un agente: Il termine “agente” indica chi esegue le istruzioni e ne mette in rilievo il suo essere distinto rispetto al soggetto in cui avviene il processo mentale. Anche questo è un elemento essenziale per caratterizzare appropriatamente l'informatica rispetto alla matematica da cui è nata. Chi specifica la soluzione di un problema nell'ambito della matematica è generalmente anche colui che la esegue. Nell'informatica la specifica della soluzione prevede necessariamente un esecutore – esterno e diverso dal risolutore – le cui capacità operative sono rigidamente e meccanicamente predefinite. Sottolineo che l'agente agisce sui dati che acquisisce per produrre dati come risultato della sua azione. Si tratta quindi di un agente in grado di elaborare informazione (*information processing agent*). È proprio **questo elemento che costituisce la novità scientifica ed il valore interdisciplinare dell'informatica**, perché consente di ottenere un punto di vista sul mondo in grado di complementare la descrizione

che sono in grado di farne gli altri tre domini della scienza. La bioinformatica offre in questo momento l'esempio più noto di tale vantaggio, ma tutte le discipline, anche quelle non scientifiche come l'arte [19], sono in grado di trarne giovamento.

sia esso persona o macchina: È una rigidità inutile ritenere che il pensiero computazionale serva solo per gestire macchine, cioè computer. Istruire una squadra di collaboratori a risolvere un problema richiede almeno le stesse capacità cognitive allenate dal pensiero computazionale: questo costituisce uno dei suoi apporti formativi che lo rendono estremamente utile per l'istruzione moderna. Non va dimenticato che questo deve avvenire a complemento e non in sostituzione di altre discipline e sempre tenendo presente che servono ovviamente altre capacità: cognitive, emozionali e relazionali.

fornendogli una serie di istruzioni: Vuol dire scrivere un programma, cioè *codice informatico* (o fare *coding*). Ritenere che il pensiero computazionale sia finalizzato ad imparare a programmare (cioè sia il *coding*) è una semplificazione snaturante perché scrivere un programma informatico vuol dire niente altro che dare concretezza al processo mentale retrostante. Da questo punto di vista sussiste, tra pensiero computazionale e *coding*, un po' la stessa relazione che esiste tra pensiero e linguaggio: quest'ultimo è la forma mediante cui rendiamo concreto il pensiero, cioè ne costituisce una *rappresentazione*, così come un programma (il *codice informatico*) costituisce una rappresentazione del pensiero computazionale. Ritenere che l'obiettivo primario della formazione al pensiero computazionale nelle scuole sia scrivere codice è come pensare che quello dell'insegnamento della matematica sia imparare a fare i calcoli.

che deve eseguire in autonomia: L'ultimo elemento, da non trascurare, è l'autonomia dell'esecutore rispetto al soggetto in cui il processo mentale avviene. Nel caso si tratti di un agente “umano”, il programma che costituisce la rappresentazione di quel pensiero computazionale che ha specificato la soluzione del problema può, in una certa misura, contare sull'intelligenza dell'esecutore. Nel caso di un agente “meccanico”, cioè di una macchina qual è un computer, non possiamo far leva su alcuna intelligenza (nel senso umano del termine) e dobbiamo quindi esplicitare ogni più piccolo dettaglio. L'allenamento fatto in questo modo da studenti sarà utile nel lavoro che sarà svolto come adulti, quando gli esecutori cui daremo istruzioni saranno persone in grado di colmare le eventuali lacune.

Un'ulteriore discussione di alcuni termini chiave del pensiero computazionale è presente in [16].

2. PERCHÉ PARLARE DI “PENSIERO COMPUTAZIONALE”

Ritorno adesso sul perché usiamo questa espressione, ovvero “di cosa parliamo quando parliamo di pensiero computazionale”. Comincio con un paio di citazioni.

Donald Knuth, scienziato famosissimo sia dai matematici che dagli informatici, nel 1974 ha scritto «*In realtà una persona non ha davvero capito qualcosa fino a che non è in*

grado di insegnarla ad un computer». [2, p.327]

George Forsythe, analista numerico ed uno dei padri della formazione universitaria in informatica (è stato tra i fondatori del Dipartimento di Computer Science di Stanford – uno dei primi a nascere e tuttora tra i migliori al mondo), nel 1968 ha scritto: «Le acquisizioni più valide nell'educazione scientifica e tecnologica sono quegli strumenti mentali di tipo generale che rimangono utili per tutta la vita. Ritengo che il linguaggio naturale e la matematica siano i due strumenti più importanti in questo senso, e l'informatica sia il terzo». [3, p.456]

Linguaggio naturale e matematica sono, giustamente, insegnati fin dal primo anno delle elementari, perché costituiscono delle competenze “fondamentali e trasversali”, cioè utili sia in sé e per sé che nell'applicazione a qualunque altra materia.

È infatti evidente che per qualunque relazione sociale serve conoscere il linguaggio naturale ed è altrettanto evidente che per discutere di quantità è necessario sapere la matematica: soltanto con “niente”, “poco” e “tanto” non si fanno molti progressi. Linguaggio e matematica sono quindi fondamentali.

Inoltre, per riuscire bene in storia, geografia, scienze, arte, il linguaggio naturale è indispensabile, altrimenti non riusciremmo a spiegare e descrivere i soggetti che stiamo trattando, ed è altrettanto necessaria la matematica, pena l'impossibilità – nel discutere di questi soggetti – di confrontare quantità, stabilire relazioni numeriche, ordinare valori. Linguaggio e matematica hanno quindi anche un valore trasversale, cioè un'utilità ed un'applicabilità interdisciplinari.

Non accade – però – che si insegnino il “pensiero linguistico” o il “pensiero matematico” unicamente per il loro valore interdisciplinare. Si insegnano, giustamente, “italiano” e “matematica” come materie di base, e poi le competenze linguistiche e matematiche vengono naturalmente travasate e messe in gioco quando si parla – ad esempio – di arte o di storia.

Nel seguito spiegherò perché anche l'**informatica è una disciplina fondamentale e con un valore trasversale**, come italiano e matematica.

Devo però prima spiegare perché usiamo il termine “pensiero computazionale” parlando di insegnamento dell'informatica nella scuola [4].

Questo dipende essenzialmente dal fatto che se si usa il termine “informatica”, si possono intendere concetti che vanno dalla “teoria della complessità computazionale” a “quale formato ha il file system sulla pennetta USB”. Questa è una maledizione da cui forse solo l'informatica è afflitta. Nessuno confonde più ormai il medico con l'infermiere, l'ingegnere con il meccanico, tutti mestieri nobilissimi ed ugualmente rispettabili, ma che hanno ovviamente diverse aree di competenza.

Usiamo quindi questa espressione affinché tutti abbiano ben chiaro che, che quando parliamo della necessità di “insegnare informatica nella scuola”, l'obiettivo non è insegnare l'uso di un certo strumento o applicazione o di una determi-

nata tecnologia o sistema, quanto l'apprendimento dei concetti scientifici di base. Esattamente come a tutti è chiaro che insegnare matematica nella scuola non vuol dire insegnare a diventare esperti nel fare i conti, ma far sviluppare competenze relative alle quantità ed alle loro relazioni e manipolazioni.

3. PERCHÉ L'INFORMATICA È UNA DISCIPLINA FONDAMENTALE

Ho discusso dell'importanza di questa formazione per lo sviluppo delle capacità cognitive razionali in [11], chiarendo che essa deve accompagnare e NON sostituire altre discipline, egualmente importanti per un'equilibrata e completa maturazione degli studenti. Ovviamente, come per la matematica alle elementari non si insegna il calcolo differenziale (e non lo si fa neanche alle medie o alle superiori) così per l'informatica bisogna insegnare i concetti di base in modo adatto allo specifico livello di maturazione degli studenti, ed in modo indipendente dalla tecnologia.

Fornisco adesso argomenti già articolati in altri paesi, più recenti rispetto alle due citazioni “storiche” precedentemente riportate.

Una **prima risposta** è quanto si sta facendo negli Stati Uniti (non certo una nazione di secondario rilievo). A dicembre del 2015 sia la Camera che il Senato, con un appoggio bipartisan ed a larghissima maggioranza, hanno approvato lo “Every Student Succeeds Act” (ESSA). Questa legge, firmata dal Presidente Obama il 10 dicembre 2015 [5], riconosce che l'informatica (*computer science*) è un soggetto fondamentale per l'educazione scolastica K-12 (negli USA si usa quest'espressione per indicare i 12 anni di educazione dalla prima elementare all'ultimo anno delle superiori, la cui durata è di 4 anni).

La legge riconosce che l'informatica è un'abilità fondamentale per fornire a tutti gli studenti un'educazione bilanciata e adeguata al 21-mo secolo. Di conseguenza la inserisce, a pari merito con discipline più tradizionali (quali la madrelingua, la matematica, le scienze, solo per citarne alcune) nell'insieme delle materie (*well rounded education subjects*) che dovranno far parte dei programmi educativi che i singoli stati dovranno definire e che saranno obiettivo di programmi di sviluppo professionale degli insegnanti.

Aver aggiunto l'informatica all'insieme dei *well rounded education subjects* significa che il legislatore americano ha ritenuto necessario esporre gli studenti da subito e con continuità a questo soggetto di studio, in modo integrato con le altre parti del curriculum, allo scopo di apportare benefici durevoli alla società e all'economia.

Successivamente, nel Gennaio 2016, il Presidente Obama ha proposto un piano da 4 miliardi di dollari affinché «tutti gli studenti americani dall'asilo al liceo imparino l'informatica e acquisiscano la competenza di pensiero computazionale necessaria per essere creatori, e non semplici consumatori, nell'economia digitale, e cittadini attivi di una società sempre più tecnologica» [6].

Una **seconda risposta** viene dall'Académie des Sciences,

l'accademia nazionale degli scienziati della Francia, che nel Maggio 2013 ha prodotto il rapporto "*L'enseignement de l'informatique en France. Il est urgent de ne plus attendre*" [7] che osserva che la maggior parte degli avanzamenti tecnologici più innovativi degli ultimi decenni sono il risultato diretto dell'informatica (dai motori di ricerca, alle reti di comunicazione, ai computer miniaturizzati presenti in ogni oggetto).

Il documento poi ricorda che la capacità dell'informatica di elaborare simboli di qualunque tipo la rende utile per tutte le altre discipline scientifiche alle quali l'informatica fornisce nuovi modi di pensare ed argomenta la necessità di insegnare l'informatica nelle scuole, a tutti i livelli, al di là dell'insegnamento a livello utente degli strumenti di base hardware e software. L'approccio suggerito è quello di iniziare esponendo i bambini alle nozioni fondamentali dell'informatica come scienza nella scuola primaria, proseguendo nella secondaria inferiore con l'acquisizione e l'approfondimento dei concetti – che conduca gli studenti ad un loro uso in autonomia, e sviluppando ulteriormente la formazione nella secondaria superiore.

Il rapporto francese sottolinea l'importanza che tutti i cittadini ricevano un'adeguata formazione sull'informatica, in modo da fornire loro le chiavi per comprendere un futuro in cui tutto sarà sempre più digitalizzato. Questa comprensione è necessaria per capire e prendere attivamente parte alla sua evoluzione invece di consumare e subire ciò che è stato deciso altrove. Il rapporto ricorda anche che in tutte le professioni, da quelle letterarie a quelle sociali, scientifiche o artistiche, l'interazione con strumenti digitali sta diventando la norma e la comprensione dei principi scientifici di base – che sono sempre gli stessi indipendentemente dai settori applicativi – è fondamentale.

In questi mesi tali indicazioni stanno diventando in Francia documenti che saranno operativi a partire dal prossimo anno scolastico.

Infine, una **terza risposta** è stata data dalla Royal Society, l'istituzione del Regno Unito analoga all'Académie des Sciences francese, che a Gennaio 2012 ha messo nero su bianco nel rapporto "*Shut down or restart? The way forward for computing in UK schools*" [8] che ogni bambino dovrebbe avere la possibilità di imparare i concetti e i principi dell'informatica, sia per il proprio beneficio personale che nell'ottica della prosperità della nazione. Il rapporto rimarca l'importanza di apprenderli sin dall'inizio della scuola primaria.

Anche per la Royal Society le motivazioni per insegnare l'informatica sono l'importanza per gli studenti di comprendere il mondo digitale che ci circonda e la possibilità di giocare un ruolo attivo e non essere consumatori passivi di una tecnologia misteriosa ed opaca. Inoltre osserva che cittadini che conoscono le basi scientifiche della tecnologia informatica sono in grado di partecipare in modo più informato e razionale alla discussione di problemi fondamentali per la società contemporanea come i brevetti software, furti d'identità, ingegneria genetica, votazioni elettroniche.

Sulla base di questo rapporto, a partire da settembre 2014 ogni scuola del Regno Unito, ad ogni livello, deve insegnare i principi base dell'informatica [9].

4. PERCHÉ L'INFORMATICA HA UN VALORE TRASVERSALE

L'informatica possiede inoltre una caratteristica, nuova e sua specifica, che la rende ancora più interessante in un'ottica **interdisciplinare**, rispetto alla lingua naturale ed alla matematica, quando si tratta di rappresentare situazioni o scenari tipici di altre discipline.

Ci arriviamo progressivamente.

Il linguaggio naturale è il mezzo base di espressione, ed in quanto tale permette di trattare qualunque soggetto, dai più concreti a quelli più rarefatti. Ovviamente ha un enorme "potere evocativo" ma purtroppo, o meglio per fortuna, possiede una sua intrinseca "ambiguità", che non gli consente quella precisione che però in alcuni casi è necessaria. Quando leggiamo "I tre moschettieri" è la nostra fantasia che dà vita ai dettagli. Infatti, ogni regista che si è confrontato con l'opera ha dato vita alla sua rappresentazione del romanzo.

A compensare questa caratteristica della lingua naturale interviene la matematica, che fa del rigore e dell'esattezza la sua forza. Se si vuole descrivere qualcosa con precisione assoluta, allora il formalismo matematico è la soluzione giusta, ed una formula matematica ha sempre un preciso significato nel contesto della sua teoria di riferimento. Ma questa forza può essere inadeguata, in molti casi della vita. Perché non sempre ciò che trattiamo è suscettibile di quella formulazione così rigorosa di cui la matematica ha bisogno per esplicitare il suo potere descrittivo. Appunto, non è possibile descrivere "I tre moschettieri" in formule.

La matematica è certamente in grado di definire simboli che rappresentano, ad opportuno livello di astrazione, concetti del mondo. Però, se non ha teorie, cioè teoremi, in grado di dedurre conseguenze pratiche, tale formalizzazione non è poi così produttiva, soprattutto nel trattamento quantitativo dei problemi delle altre discipline.

Ne abbiamo un esempio lampante nella biologia, in cui diversi aspetti erano stati modellati matematicamente già molti decenni fa. Ma è stato solo quando ad essi si è applicata l'informatica, con le sue peculiari caratteristiche sopra ricordate, che in quest'area "formale" della biologia è scoppiata una fioritura di risultati senza precedenti.

Un ulteriore esempio di questo tipo è relativo alla manipolazione del linguaggio naturale. Non si può costruire una teoria matematica precisa del linguaggio naturale, perché troppe sono le eccezioni e le sfumature. Ma l'informatica, che come lo stesso Knuth ricorda [2, p.326], eccelle nel trattare sia casi particolari ed eccezioni, che relazioni che cambiano subito dopo che sono state definite, riesce a gestire queste situazioni. Infatti gli impressionanti avanzamenti degli ultimi anni nella comprensione del linguaggio da parte dei calcolatori sono concretamente possibili solo grazie all'informatica, anche se basati su teorie matematiche molto sofisticate.

La **forza dell'informatica in un contesto educativo** è co-

stituita dalla capacità di concretizzare mondi virtuali, grazie alle sue possibilità apparentemente illimitate di **elaborare in qualunque modo simboli di qualunque tipo**.

Con l'informatica è possibile rappresentare qualunque mondo vogliamo esplorare e dargli senso, definire un insieme di regole e dar loro vita, animandole mediante un'esecuzione su di un appropriato dispositivo automatico. Con l'informatica è 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.

Concludo con una nota di cautela osservando che estendendo queste simulazioni diventa possibile costruire quelle "realtà virtuali" di cui stiamo vedendo l'incalzante diffusione ed alle quali è comunque necessario fare attenzione per le possibili conseguenze sulla vita reale.

5. CONCLUSIONI

Questi sono i motivi per cui parliamo di "pensiero computazionale" nella scuola. Ma non chiediamo che sia questo il nome della materia, non ha senso. Non chiediamo che in tutte le materie ci sia "il momento del coding", non ha senso. Bisogna insegnare "informatica", così come si insegna "italiano" e "matematica". E poi, se questo viene fatto in modo adeguato, da docenti che sono stati adeguatamente preparati a questo scopo, allora i benefici verranno.

Il nostro Paese, in cui nel corso di soli due anni scolastici (2014-15 e 15-16) circa 25.000 insegnanti hanno portato più di un milione di studenti a svolgere quasi 10 ore a testa di informatica [10], ha dimostrato di essere pronto ed interessato a seguire questa strada.

Non perdiamo quest'occasione.

RIFERIMENTI

- [1] <http://www.treccani.it/enciclopedia/raymond-carver/>
- [2] Donald Knuth, Computer Science and its relation to Mathematics, *The American Mathematical Monthly*, vol.81, n.4, pp 323-343, April 1974. <http://www.maa.org/programs/maa-awards/writing-awards/computer-science-and-its-relation-to-mathematics>
- [3] George E. Forsythe, What to do till the computer scientist comes, *The American Mathematical Monthly*, vol.75, n.5, pp 454-462, May 1968. https://www.maa.org/sites/default/files/pdf/upload_library/22/Ford/GeorgeEForsythe.pdf
- [4] Enrico Nardelli, Tutti possono parlare di informatica?, Gennaio 2017. <http://www.roars.it/online/tutti-possono-parlare-di-informatica/>
- [5] The White House, Report on the "Every Student Succeeds Act", December 2015. <https://www.whitehouse.gov/the-press-office/2015/12/10/white-house-report-every-student-succeeds-act>
- [6] The White House, Report on the initiative "Computer Science for All", January 2016, <https://www.whitehouse.gov/blog/2016/01/30/computer-science-all>
- [7] Académie des Science, *L'enseignement de l'informatique en France. Il est urgent de ne plus attendre*, Mai 2013.
- [8] The Royal Society, *Shut down or restart? The way forward for computing in UK schools*, January 2012.
- [9] UK Department for Education, *National curriculum in England: computing programmes of study*, September 2013. <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study>
- [10] Progetto "Programma il Futuro", <http://www.programmailfuturo.it/progetto/descrizione-del-progetto>
- [11] Enrico Nardelli, *Informatica: il nuovo latino che tutti amano*, Giugno 2016. <http://link-and-think.blogspot.it/2016/06/informatica-il-nuovo-latino-che-tutti.html>
- [12] Jeannette Wing, Computational Thinking, *Communications of the ACM*, vol.42, n.3, pp. 33-35, March 2006.
- [13] The LINK, Research Notebook: Computational Thinking – What and Why?, *The Magazine of Carnegie Mellon University's School of Computer Science*, March 2011. <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>
- [14] Cynthia C. Selby e John Woollard, Computational Thinking: The Developing Definition, *Information Technology in Computer Science Education conference*, 2013, (ITiCSE-13).
- [15] Michael Lodi, *Imparare il pensiero computazionale. Imparare a programmare*. Tesi di Laurea Magistrale in Informatica, Univ. Bologna, 2013.
- [16] Enrico Nardelli, *Cinque parole chiave del pensiero computazionale*, Aprile 2016. <http://link-and-think.blogspot.it/2016/04/cinque-parole-chiave-pensiero-computazionale.html>
- [17] Peter J. Denning e Paul S. Rosenbloom, Computing: the fourth great domain of science, *Communications of the ACM*, vol.52, n.9, pp 27-29, September 2009.
- [18] Enrico Nardelli, *Gli studenti dell'era digitale: automi o pensatori critici?*, <http://www.ilfattoquotidiano.it/2016/04/21/gli-studenti-dellera-digitale-automi-o-pensatori-critici/2654546/>
- [19] Enrico Nardelli, A viewpoint on the Computing-Art dialogue, *Leonardo*, vol. 47, n.1, pp. 43-49, 2014.