

Il Ciclo dell'Informazione

Comunicare Elaborare Interpretare

Introduzione al progetto realizzata con il contributo del dott. Giuseppe Pea,
esperto di didattica della Matematica e dell'Informatica nella Scuola

L'Informatica nei programmi curricolari

Cercando di interpretare i recenti orientamenti ministeriali relativamente all'introduzione dell'Informatica nella scuola, si possono fare le seguenti riflessioni:

- L'Informatica è auspicata, ma non come disciplina autonoma, almeno per quanto riguarda la scuola di base (attualmente elementari e medie). Ciò significa che si tratta prevalentemente di una accezione strumentale dell'Informatica, che non prevede la presenza di insegnanti con competenze specifiche. Questo orientamento tutto sommato riflette quanto già sperimentato da alcuni anni nei principali paesi Europei (si veda il Dossier "Tecnologie dell'Informazione e della Comunicazione nei sistemi educativi in Europa" redatto dalla Commissione Europea in collaborazione con Euridice e Eurostat, Febbraio 2000) e la possibilità di saper cogliere e valorizzare i contributi culturali dell'Informatica e di sfruttarne le potenzialità formative ricadrà sugli insegnanti delle discipline curricolari.
- Dal punto di vista di ogni disciplina che preveda l'uso degli strumenti informatici si tratterà di sviluppare un nuovo modo mentale per affrontare i problemi tipici di quella disciplina, problemi che potrebbero avere poco a che fare con i concetti e le metodologie proprie dell'Informatica. Senza sottovalutare in alcun modo il ruolo formativo di queste attività, è invece importante sottolineare che esse non garantiscono di per se un contributo allo sviluppo di una cultura informatica; in particolare non richiedono un approccio altrettanto rigoroso di quello necessario a dominare i problemi tipici dell'Informatica.
- Lo studio dell'Informatica come disciplina scientifica autonoma potrebbe essere eventualmente introdotto utilizzando gli spazi di autonomia che saranno lasciati ai diversi Istituti (anche se resta ancora da verificare quanti vincoli che incidono su questa potenziale autonomia saranno posti, per esempio, a livello regionale).

Informatica "ricca" e Informatica "povera"

Secondo la percezione comune, l'Informatica di oggi è prevalentemente legata alla multimedialità che, mettendo in maggior rilievo gli aspetti relativi alla comunicazione, rischia di rimanere principalmente confinata nell'ambito linguistico. Le attuali modalità di interazione con le macchine sono state concepite allo scopo di permetterne l'utilizzo anche agli utenti meno acculturati. In questo modo tutti possono servirsi del computer, però per fare altre cose, non per fare informatica. In quest'ottica si sono collocate di preferenza la maggior parte delle proposte didattiche, in particolare quelle al di fuori della matematica: l'informatica è stata proposta in questo senso per migliorare lo studio delle lingue, nell'educazione artistica e anche nell'educazione musicale. Si fa cioè uso di strumenti informatici per affrontare i problemi della lingua, che sia la lingua straniera o la lingua madre non ha importanza, per impadronirsi di nuovi linguaggi artistici oppure grafici, per fare musica.

Questi sviluppi hanno spostato l'informatica da un dominio prevalentemente matematico (come è stato nel passato) ad un dominio che tende ad essere pervasivo di tutte le attività scolastiche, di tutte le discipline. In questo spostamento però si perde, per quanto riguarda l'impostazione metodologica e l'educazione informatica, in rigosità; e si perde sia sul versante formativo, sia sul versante logico. È vero che si continuerà ad affrontare questo problema nell'ambito della matematica, come si è sempre fatto anche attraverso l'informatica, ma si aggiungono delle modalità di utilizzo degli strumenti che sono in qualche modo meno rigorose, in quanto cercano di sfruttare lo strumento il più possibile come tale, al di là del fatto che si capisca o non si capisca che cosa accade all'interno dello strumento: l'importante è ottenere dei risultati; in che modo poi si siano ottenuti e per quale motivo non ha altrettanta importanza. È un po' la logica della patente di guida, per conseguire la quale all'inizio si doveva conoscere la macchina come tale: i principi della sua meccanica, il funzionamento del motore, e così via; poi ci si è accontentati di conoscere come si deve guidare: galateo e segnaletica stradale, abilità di coordinamento, ecc. Anche per l'informatica si sta andando in una direzione che privilegia attività che potremmo definire "di coordinamento manuale" (non a caso l'esame relativo all'uso di strumenti informatici, riconosciuto a livello Europeo, è denominato European Computer Driving Licence, cioè Patente Europea "di guida del computer", <http://www.aicanet.it/ecdl.htm>). Cioè lo strumento informatico come una qualunque macchina, un qualunque elettrodomestico che noi utilizziamo per ottenere dei lavori, non per crescere cognitivamente, non per conquistare delle capacità mentali elevate.

A questo punto è forse opportuno fare un discorso sull'informatica un po' controcorrente, cioè contro la concezione prevalentemente associata alla multimedialità, allo strumento per potenziare le capacità comunicative. Se poi la multimedialità viene affrontata in modo spontaneo da parte del bambino, ma anche da parte dell'insegnante, attraverso strumenti di fruizione multimediale abbastanza semplici, come per esempio gli strumenti di presentazione che si prestano ad interazioni multimediali "ad effetto" dal punto di vista prettamente comunicativo, allora corriamo il rischio di impastare tante cose senza capirvi niente e senza favorire alcuna crescita mentale: si tratterebbe solo di un'improvvisazione comunicativa da cui né l'adulto né tanto meno il bambino potranno trarre un significativo beneficio. Portare la comunicazione al livello multimediale attraverso simili strumenti può essere bello, e può apparire anche gratificante per il bambino, ma cosa aggiunge sul piano della formazione? Il problema che dobbiamo affrontare quando proponiamo qualcosa attraverso strumenti informatici è: che cosa vogliamo dare all'allievo? Ciascun insegnante può ovviamente rimanere all'interno della propria disciplina, perché questo è l'uso dell'informatica suggerito dagli attuali orientamenti della politica scolastica. Il discorso però si presenta abbastanza complesso, perché disponiamo di numerosi strumenti informatici di notevole potenza e quindi anche una mente piuttosto intuitiva è in grado di ottenere risultati appariscenti sul piano della creazione di qualcosa di "informatico": riesce cioè ad ottenere dei bei risultati, ma non a formarsi una impostazione mentale adatta per andare oltre quei risultati spontanei. Quello che dobbiamo fare noi invece, come educatori, è cominciare a concepire una educazione mentale che permetta di superare il livello di superficialità delle facili forme di comunicazione multimediale che sono così spontanee, per arrivare a creare qualcosa che sia veramente importante dal punto di vista formativo.

Si impongono ora alcune questioni.

1. Gli insegnanti conoscono i fondamenti di informatica per poter impostare in modo rigoroso un percorso didattico che tratti (anche) di informatica? Perché se gli insegnanti stessi non hanno queste basi, in che modo possono comunicare contenuti informatici agli allievi? La domanda ha un senso perché la formazione di parecchi insegnanti, in particolare della scuola elementare, è una formazione che non ha contemplato mai alcunché di informatico. Almeno non secondo il normale curriculum di preparazione (non per colpa degli insegnanti, quindi, ma per le caratteristiche delle strutture scolastiche). E non è certamente garantito che gli insegnanti delle scuole medie abbiano affrontato argomenti di informatica nel corso degli studi universitari. Nasce quindi il problema di acquisire, da parte degli insegnanti, la capacità di riflettere sulla propria conoscenza informatica e di capire se è adeguata a questi obiettivi didattici o se deve essere integrata. Gli insegnanti che vogliono portare avanti assieme un discorso informatico, condividendo strumenti e materiali, si devono anche preoccupare di una forma di aggiornamento informatico, almeno per accertarsi che quando si parla di determinate cose tutti intendano in modo corretto, per condividere una stessa impostazione iniziale.

Dopo, i modi di fare e le metodologie possono diversificarsi, ma tutti devono partire da una stessa base. Capire i principi di base del funzionamento di un computer, per esempio, non è affatto difficile, e non serve essere dei tecnici o degli ingegneri: basta cominciare a familiarizzare con l'argomento e, una volta scoperte le idee fondanti, la macchina non si presenterà più come uno strumento estraneo o magico, ma semplicemente come uno strumento che funziona in un certo modo, avendo capito come e perché funziona in quel modo. Conoscere questi principi è di fondamentale importanza.

2. L'accento ai principi di funzionamento dell'elaboratore introduce il versante delle conoscenze e della cultura informatica. Ma va qui sottolineato che l'informatica come fatto culturale non può e non deve sfociare sempre nel computer. Questa è un'altra cosa estremamente importante: si può fare molta cultura informatica senza avere il computer davanti, senza nemmeno pensare a un computer. Questo può essere difficile da capire per gli insegnanti che non hanno avuto significative esperienze riguardo al modo in cui si risolvono i problemi tipici dell'informatica. In tal caso tende a prevalere lo stereotipo secondo cui l'informatica nella scuola è concepibile solo se c'è un laboratorio con tutti i computer e con tutto il software, e se c'è qualcosa che non va in un computer allora sembra che non si possa più fare informatica. Sul piano culturale il discorso sull'informatica è più articolato: c'è la parte rivolta al computer, ma c'è anche la parte che può fare a meno di esso. Una decina di anni fa, le due informatiche venivano identificate con i termini "informatica ricca", quella con il computer, e "informatica povera", quella senza il computer. Ora questa terminologia va man mano scomparendo, non perché abbiano perso di significato, ma perché si tende ad escludere la cosiddetta informatica povera. L'informatica ricca si sta oggi imponendo come unico concetto di informatica, e questo rischia di avere conseguenze negative perché prima di cominciare a lavorare con il computer ci sono tante cose da fare sul piano della formazione mentale, proprio attraverso l'informatica povera, povera nel senso che non fa spendere tanti soldi, ma che è invece estremamente ricca nel senso del processo formativo.

3. Il contrasto tra informatica ricca e povera è un problema ampiamente dibattuto in questi anni e probabilmente continuerà ad essere dibattuto per molti altri perché gli studenti, gli allievi, non vogliono fare l'informatica povera, preferiscono stare al computer quando si parla di informatica. Allora, e paradossalmente, per fare informatica povera è importante non dire che si sta facendo informatica.

4. Se si condivide l'importanza del contributo formativo che può venire dall'informatica povera, ci si deve anche chiedere se informatica povera può essere fatta in tutte le discipline o è caratteristica di una specifica disciplina o di un gruppo di discipline. Infatti, fin qui l'informatica povera è stata patrimonio degli insegnanti di matematica, o comunque di insegnanti specializzati (qui ci riferiamo soprattutto alle scuole elementari), cui veniva affidato il compito di affrontare i concetti e le metodologie dell'informatica sul piano proprio e non su quello strumentale. Rispondere alla domanda posta sopra è particolarmente importante se la cultura informatica deve essere portata avanti da tutti gli insegnanti, qualunque sia la disciplina di cui si occupano: in che modo gli insegnanti di lingua, per esempio, possono portare avanti la parte povera dell'informatica? In che modo gli insegnanti di educazione tecnica? O è meglio piuttosto limitarsi agli insegnanti di matematica come nel passato? Per gli insegnanti di lingua, per esempio, ci sono degli agganci per un motivo molto semplice: tutto ciò che ha a che fare con l'informatica ha a che fare con gli strumenti linguistici. Anche se si ritiene che la lingua sia cosa diversa rispetto ai linguaggi, agli strumenti linguistici, posizione per molti versi condivisibile, in ogni caso restano degli agganci a diversi livelli, per esempio a livello logico.

5. Le considerazioni appena fatte suggeriscono l'idea che l'informatica sia anche trasversale alle discipline, e non sul piano tecnologico (per la presenza del computer), ma su quello della formazione culturale. E siccome si sostiene insistentemente che, specialmente nella scuola di base, ci deve essere la trasversalità e non tante isolette separate, l'informatica può costituire proprio uno dei momenti in cui la trasversalità diventa quasi d'obbligo. L'importante è che anche gli insegnanti di lingua e di tutte le altre discipline si convincano, come gli insegnanti di matematica, che l'informatica è trasversale. Si cercherà di fornire alcuni spunti di riflessione in tal senso nei successivi paragrafi di questo documento.

Trasversalità della Cultura Informatica

Strumenti linguistici

Molte questioni legate ai linguaggi sono pertinenti all'ambito informatico. Per questo motivo alcuni collegamenti fra l'ambito linguistico e quello informatico saranno ripresi nei paragrafi seguenti con riferimento ad aspetti specifici. Qui ci limitiamo semplicemente ad un sintetico elenco di argomenti che hanno rilevanza sia in ambito linguistico sia in ambito informatico, anche se, va precisato subito, a parità di terminologia nei due ambiti non corrisponde sempre lo stesso concetto.

- Elementi della comunicazione: trasmissione e ricezione del messaggio, strumenti per comunicare, comunicazione uomo-macchina.
- Alfabeto e codice: simboli e combinazioni di simboli, parole, convenzionalità del codice.
- Sintassi di un linguaggio: regole di costruzione di frasi, formule ben formate, programmi, struttura dei costrutti linguistici.
- Semantica di un linguaggio: referente, significato, ruolo del contesto, ambiguità.
- Funzione delle frasi: frasi con funzione enunciativa, imperativa, descrittiva, ecc.
- Metalinguaggio: uso di un linguaggio per descrivere proprietà sintattiche e semantiche relative a un altro linguaggio o al linguaggio stesso.

Considerando le questioni legate al linguaggio più in generale, si può osservare che alla base della nostra cultura vi è sempre stata una riflessione critica sulla realtà e sulla sua percezione, riflessione che ha utilizzato come principale strumento il linguaggio, soprattutto nella sua forma scritta: ogni descrizione o spiegazione della realtà ha trovato la sua espressione privilegiata attraverso questa forma linguistica. Come conseguenza delle innovazioni tecnologiche degli ultimi decenni, soprattutto nell'ambito dell'informatica, la percezione della realtà sta diventando sempre più indiretta, in buona parte mediata da macchine di varia natura. La gran mole di dati resi disponibili ed elaborati dai nuovi strumenti tecnologici e la varietà di linguaggi simbolici a cui ciascun individuo è esposto fin dalla più giovane età sono accompagnate da una sensibile riduzione degli spazi di riflessione critica. Così, anche i linguaggi si stanno rapidamente trasformando in qualcosa di diverso da ciò a cui eravamo abituati. In particolare, il linguaggio scritto sta perdendo la sua preminenza assoluta come strumento per descrivere e spiegare la realtà. Sarà pertanto opportuno approfondire le caratteristiche, le potenzialità, ma anche i limiti delle nuove forme linguistiche in modo da consentire una fruizione più consapevole e, per quanto possibile, più creativa degli strumenti tecnologici, anche nell'ambito delle discipline umanistiche ed artistiche.

Logica

Già da molto tempo, e pressoché universalmente, si è riconosciuto nella logica uno dei temi più ricchi di connessioni interdisciplinari, in particolare con gli ambiti linguistico, filosofico, matematico e informatico. Questo risulta evidente anche dallo spazio che i libri di testo hanno dedicato all'argomento, fin dai primi livelli di istruzione. Benché oggi si possono ascoltare voci più critiche riguardo alla presupposta valenza formativa della logica, soprattutto se calata nella scuola di base, resta tuttavia il fatto che un minimo di dimestichezza con enunciati e connettivi logici stanno alla base di qualunque attività che affronti l'analisi e la ricerca di soluzioni algoritmiche a problemi. Quando si introducono i connettivi logici è interessante confrontare i modi in cui vengono utilizzati in informatica e in matematica, piuttosto che in filosofia o nel linguaggio naturale, evidenziando dove ci sono

corrispondenze di significato nei vari ambiti culturali e dove invece i significati si differenziano. In questo modo la logica diventa un'occasione per realizzare attività interdisciplinari.

Strutture e linguaggi per esprimere strutture

Un'ulteriore occasione di trasversalità è fornita dal concetto di struttura, che è senza dubbio uno dei concetti fondanti in ambito informatico. Non si può infatti fare informatica se non si è capaci di cogliere e immaginare strutture. L'informatica, intesa da questo punto di vista, contribuisce anche a dare una formazione logica elevata, perché quando si affrontano le strutture si affrontano nessi, relazioni, sequenzialità, simultaneità. Oltre a saper cogliere le strutture, occorre poi poterle tradurre sul piano linguistico, pervenendo a una forma di comunicazione scritta. A tal fine, si deve saper scegliere fra le varie forme linguistiche quelle che meglio delle altre descrivono le strutture considerate.

Per esempio, in una situazione in cui si analizza un brano con determinate caratteristiche affinché gli allievi possano inventare storie a ricalco è necessario che essi riconoscano la struttura che deve poi essere applicata alle nuove storie con caratteristiche simili. Lo stesso succede quando si esamina la struttura di una poesia o di altre opere d'arte. In tutti questi casi l'allievo deve individuare una struttura, per quanto semplificata, per poter imitare l'artista. Benché ci sia chi sostiene che queste operazioni mentali sono troppo elevate per l'età dei bambini che frequentano la scuola elementare, riteniamo che educare a cogliere le strutture possa comunque avere effetti estremamente positivi, purché i bambini siano motivati a fare queste esperienze. Come ulteriore esempio, quando si risolve un problema in matematica è opportuno evidenziare la struttura complessiva della risoluzione e non limitarsi ad esprimere la sequenzialità delle operazioni. Mettendo in evidenza solo la sequenzialità delle operazioni, infatti, non si fa una buona educazione informatica in quanto quella sequenziale è la struttura più povera.

Oggi si può dire che in un certo senso l'educazione ai linguaggi per esprimere strutture, una volta affrontati soprattutto nell'ambito della logica, è stata quasi assorbita dall'informatica. Non per nulla nei libri della scuola elementare, anche nei libri di lingua, si trovano molti diagrammi di flusso, molti diagrammi ad albero, che servono precisamente a mettere in evidenza strutture che con la lingua italiana difficilmente si potrebbero descrivere in modo efficace.

Equivalenze fra strutture

Nell'educazione informatica bisogna portare lo studente a dominare le corrispondenze tra strutture diverse, non le strutture prese individualmente ma l'equivalenza fra strutture diverse. Anche questo è un concetto estremamente importante: non basta cogliere la struttura e saperla esprimere in una forma linguistica, ma anche capire quando due strutture all'apparenza diverse sono fra loro equivalenti, cioè pur non essendo uguali si equivalgono. Questo discorso sul piano informatico è rilevante perché un linguaggio informatico, in questo caso anche un linguaggio di programmazione, non permette di esprimere efficacemente la soluzione di qualsiasi problema. Di conseguenza, quando ci si trova di fronte ad una situazione operativa in cui lo strumento linguistico non è più adeguato, si deve trovare fra i tanti mondi linguistici una forma equivalente che si presti meglio ad affrontare la specifica situazione.

In quest'ottica, l'uso dei diagrammi di flusso o dei diagrammi ad albero come forme linguistiche che esprimono determinate strutture, e che devono poi essere tradotte in un linguaggio di programmazione, permette di affrontare il problema della corrispondenza fra strutture e di acquisire la necessaria flessibilità per passare agevolmente da un forma linguistica ad un'altra. Sul piano educativo questo è un punto estremamente importante perché gli studenti tendono ad evitare questo addestramento, cioè tendono a pensare solo con una forma linguistica.

Nella scuola di base il discorso sull'equivalenza delle strutture viene portato avanti soprattutto nell'ambito dello studio di una lingua straniera, facendolo diventare uno dei principi di

alfabetizzazione. Perché nella lingua straniera le traduzioni non vengono fatte mettendo in corrispondenza parola con parola, ma mettendo in corrispondenza struttura (per esempio sintattica) con struttura. Ad una struttura in una determinata lingua corrisponde una struttura diversa ma equivalente nell'altra lingua. Problematiche simili si possono affrontare in tutte le discipline: se la soluzione di un problema si esprime attraverso un determinato linguaggio, come si può esprimere in un altro linguaggio? Con le stesse strutture o con strutture diverse?

Metalinguaggi

Un altro significativo strumento che ritroviamo nell'informatica è quello del metalinguaggio. Ogni linguaggio può infatti essere generativo di un linguaggio più potente attraverso la creazione di strutture che sono equivalenti a quelle presenti nella nostra mente. In particolare, è importante mettere in risalto il ruolo del linguaggio come metalinguaggio quando si usa una forma linguistica per esprimere come una struttura possa essere tradotta in un'altra equivalente. In matematica, scegliendo strumenti metalinguistici adeguati, si può mettere in evidenza la struttura che ha permesso di risolvere il problema, al di là della sequenzialità operativa. Tali strumenti possono essere diagrammi ad albero o diagrammi di flusso, per esempio, dove al posto dei dati numerici si rappresenta il significato delle grandezze (che spesso sono implicite nei problemi).

Potenzialità risolutive del linguaggio

Una forma linguistica, nel momento in cui viene codificata e nel momento in cui i codici vengono accettati, serve normalmente per comunicare e la comunicazione avviene nel rispetto dei criteri di codifica stabiliti. Quando esaminiamo un linguaggio da questo punto di vista, lo stiamo esaminando come strumento di comunicazione. E il linguaggio si ritrova in matematica come in logica, in informatica come nelle scienze: ogni forma di sapere sviluppa le proprie competenze linguistiche. Tuttavia c'è una differenza fra il linguaggio inteso in senso logico, matematico, informatico e il linguaggio inteso in senso linguistico. Nel secondo caso la principale preoccupazione risiede nell'efficacia della comunicazione: questo è l'obiettivo prioritario dell'esame da parte dei linguisti. E l'efficacia della comunicazione talvolta si ottiene proprio non rispettando le regole della sintassi, perché quando si parla di efficacia il problema non è più il rigore ma il modo in cui si riesce a far passare certi messaggi. Molte volte la forma linguistica è tanto più efficace quanto meno le regole vengono rispettate. Questa è un fenomeno importantissimo che permette alle forme linguistiche (studiate nella cosiddetta comunicazione antropologica) di evolversi costantemente, l'evoluzione traendo impulso proprio dall'abbattimento di regole per creare nuove regole, per ottenere una efficacia sempre maggiore. Questo è anche l'obiettivo degli educatori di lingua e degli studiosi dei linguaggi impegnati sul versante della comunicazione.

Per quanto riguarda invece la matematica, per quanto riguarda le scienze, il rigore diventa un fatto fondamentale, il mancato rispetto delle regole compromette la comunicazione. E questa è una distinzione rilevante: i linguaggi delle discipline scientifiche non hanno solo potenzialità e finalità comunicative, ma anche potenzialità e finalità di natura risolutiva (prescrittiva, operativa). Ciò significa che questi linguaggi forniscono strumenti per risolvere i problemi, in qualche modo automaticamente. E la risoluzione automatica attraverso le forme linguistiche è un altro dei fondamenti dell'informatica. A dire il vero anche i linguaggi naturali forniscono strumenti che a volte permettono di risolvere problemi. Ma in matematica, tra tanti potenziali linguaggi, tutti in grado di esprimere in modo corretto gli elementi che devono essere espressi, quale linguaggio si preferisce? Quello che, oltre alla correttezza, dà i migliori strumenti per risolvere "automaticamente" i problemi in esame. Si prenda come esempio la scrittura dei numeri con la notazione Romana e quella con le cifre Arabe. Entrambi sono modi per comunicare correttamente i numeri in forma scritta. Ma se scriviamo gli operandi di una divisione o di una moltiplicazione come li avrebbero potuti scrivere i Romani, comunichiamo senza dubbio le quantità del dividendo e del divisore, ma poi come facciamo a trovare il quoziente o il resto utilizzando quella scrittura? Si tratta di un ottimo strumento di comunicazione, ma non ha in sé

potenzialità risolutive tali da introdurre degli automatismi. Attraverso le cifre Arabe e la corrispondente notazione posizionale, la soluzione di questo tipo di problemi può invece essere affrontata basandosi su meccanismi “algoritmici”.

Nella scuola, e in particolare in quella elementare, la lingua è utilizzata prevalentemente sul versante espressivo. Anche quando si insegnano le tecniche di calcolo, in genere non si mette nella dovuta evidenza dove sta la potenzialità risolutiva del linguaggio utilizzato e in che senso. Molta parte dell'informatica, invece, si è sviluppata in relazione alle forme linguistiche con potenzialità risolutive, basti pensare ai linguaggi di programmazione. I costrutti if-then-else, while-do, repeat-until, ecc. sono di fatto strumenti che permettono di risolvere automaticamente, sono macrostrutture dotate di potenzialità risolutiva. Basta scriverle per risolvere un problema.

Strumenti linguistici risolutivi dell'Informatica povera

Uno degli obiettivi principali dell'informatica è quello di risolvere problemi: è al fine di risolvere problemi che si affrontano i principi della programmazione. Ci sono strumenti linguistici anche nell'informatica povera che permettono di “risolvere” alcune classi di problemi. I diagrammi ad albero, per esempio, non sono solo uno strumento espressivo, ma anche uno strumento risolutivo. Tradizionalmente un problema viene inizialmente risolto in modo sequenziale. Per educare anche all'informatica, e per educare alle strutture, è opportuno poi esprimere in che relazione sono stati posti fra di loro tutti gli elementi in gioco, e per fare questo l'insegnante può costruire un albero. A questo punto si è svolto un lavoro organizzativo, mettendo in evidenza la struttura risolutiva, a partire dai dati per arrivare alla risposta. L'uso del linguaggio in questo senso è ancora espressivo, non risolutivo: chiarisce una cosa che è già conosciuta e ha finalità comunicativa.

Con il raggiungimento della reversibilità del pensiero, già nel secondo ciclo delle elementari, si può passare dalla pura espressione di un procedimento risolutivo che si è applicato alla prescrizione del procedimento in sé. Va comunque tenuto presente che esistono due età mentali diverse, alle quali corrispondono fasi linguistiche diverse. Partendo dai sette anni e mezzo, si può proporre lo strumento linguistico dei diagrammi ad albero semplicemente in funzione espressiva. Solo verso la fine della quarta si può cominciare a chiedere al bambino se prima di fare le operazioni è capace di disegnare il diagramma che rappresenta la risoluzione. L'utilizzo del linguaggio in veste risolutiva è invece appropriato dopo i nove anni e mezzo. Prima di quell'età non si può contare sulla necessaria reversibilità del pensiero; se invece di dare il problema in forma numerica lo si dà in termini di grandezze, non accompagnate da numeri, per i bambini questo può significare che non si è in grado fare operazioni, e quindi non ci sono nemmeno le basi per pensare in termini operativi.

Il linguaggio visto sotto il profilo risolutivo sta alla base di tutta la programmazione. Non si risolve il problema facendo le operazioni, ma indicando in che modo si devono succedere le operazioni. Poiché il calcolo lo farà eventualmente il computer, si deve solo indicare la struttura che permette di giungere alla risoluzione del problema. I diagrammi ad albero possono costituire una delle prime esperienze di programmazione, e si può fare molta informatica in questo modo. Più tardi si incontreranno alcuni problemi che non si risolvono semplicemente per mezzo di operazioni aritmetiche, ma richiedono confronti e scelte conseguenti ai risultati dei confronti. E via via per quanto riguarda altri costrutti sempre più sofisticati. I diagrammi ad albero però colgono un nesso solo di tipo “operazione binaria” (o poco più) e non si prestano ad altri costrutti che hanno a che fare con la programmazione. Opportuni diagrammi risolutivi sono allora i diagrammi di flusso, o in alcuni casi i diagrammi a blocchi, ecc. Tuttavia, per quanto riguarda la scuola elementare e la scuola media, anche questi diagrammi devono essere affrontati nello spirito dei linguaggi prima visti come strumenti espressivi, poi fatti diventare strumenti risolutivi.

Anche ai fini della programmazione, la soluzione di un problema si dovrebbe dapprima affrontare attraverso diagrammi di flusso o schemi a blocchi) cioè sulla carta. Solo quando questa fase è terminata si utilizzeranno i diagrammi (gli schemi) come traccia per tradurre l'algoritmo nel linguaggio di programmazione prescelto, sia esso il Pascal, il C, Java o un altro linguaggio. E con l'operazione di

traduzione siamo di fronte a quella forma di educazione informatica che è stata già individuata, importantissima, che è la corrispondenza tra le strutture. Questo lavoro può essere fatto verso la fine della scuola media e all'inizio delle superiori. È un lavoro importante che gli insegnanti potrebbero fare sempre, non consentendo agli studenti di mettersi al computer e di improvvisare la soluzione lavorando direttamente con un linguaggio di programmazione. È sempre meglio, sul versante formativo ed educativo, far risolvere il problema prima nella sua essenzialità con gli strumenti linguistici propri dell'informatica povera (su carta, quindi) e poi tradurlo applicando le corrispondenze tra strutture e aggiungendo i dettagli peculiari dei linguaggi adatti alle macchine (come, per fare un esempio, le dichiarazioni, che non si trattano con i diagrammi di flusso). Inoltre, i diagrammi di flusso educano a riesaminare la soluzione proposta passo dopo passo, analizzando accuratamente il processo risolutivo.

Generalizzazione

Saper generalizzare è un altro dei fondamenti dell'educazione informatica. Generalizzare i problemi significa evitare che un problema sia risolto in modo contingente, solo per quei particolari dati, e cercare una strategia risolutiva che vada bene per un gran numero di problemi correlati, capire che si sta risolvendo una classe di problemi. Perché poi la programmazione non è finalizzata a risolvere un problema, ma una classe di problemi. Bisogna comunque prestare attenzione al fatto che generalizzare non significa solo sostituire un numero con la grandezza corrispondente, benché si tratti già di una generalizzazione, ma significa anche capire che quando si esprimono determinate grandezze certi valori non possono essere assunti. Ad esempio, quando si vuole generalizzare un problema la cui soluzione prevede una divisione, indicare l'operazione nei termini generali "A diviso B" non comporta automaticamente la potenzialità di risolvere una vasta gamma di problemi, perché se B può assumere il valore zero, allora la generalizzazione non risulta corretta. Nella generalizzazione bisogna porre attenzione alle condizioni nelle quali un certo problema non può essere risolto con quel algoritmo. La capacità di prevedere quali sono i casi in cui un algoritmo non può essere applicato presuppone una modalità di ragionamento piuttosto sofisticata, che non si affronta nella scuola elementare. A quell'età si può introdurre la generalizzazione nella sua forma più elementare, semplicemente dando dei nomi alle grandezze, ma senza porre condizioni. Nella scuola media è invece opportuno cominciare a rendere il bambino consapevole del fatto che se capita una particolare combinazione di dati l'algoritmo che avevo congegnato non risolve il problema.

Dati e informazioni

Un problema che è interessante porsi riguarda la differenza fra dato e informazione. Almeno per gli educatori dovrebbe essere una distinzione fondamentale. Se qualcuno ci propone una comunicazione scritta, leggiamo le stesse frasi, le stesse parole, ma nella nostra mente non si forma lo stesso tipo di comprensione, cioè non vi leggiamo esattamente la stessa informazione. Che cosa è accaduto? È accaduto che ciascuno di noi interpreta quanto trova scritto in un modo un po' diverso. Quanto è scritto è il dato; l'interpretazione è invece l'informazione. Ciò che viene trasmesso dagli insegnanti agli allievi, come ciò che viene trasmesso da questo testo a chi lo legge è costituito da un insieme di dati, non da informazioni. È solo la cultura di chi riceve i dati, la sua esperienza, il suo passato, che li può far diventare informazioni, ma se dovesse mancare una forma di cultura, allora i dati sarebbero ricevuti esclusivamente come tali. Quando si intende di comunicare, in genere si pensa in termini di informazione, ma nella realtà vengono trasferiti solo dati; l'informazione sta nel modo in cui i dati vengono tradotti nella mente di chi li riceve e nel significato che vi assumono.

Analogamente, la distinzione fra dati e informazioni è importante nel caso in cui operiamo con un computer. Utilizzando il computer noi lavoriamo prevalentemente con dei dati rappresentabili all'interno del computer, dati che dovrebbero tradursi in informazioni, ma non è detto che ciò accada. Si può dire che una delle principali differenze fra informatica e semplice studio del computer, come oggetto tecnologico, sta nel fatto che nell'informatica c'è sempre il significato delle cose. Il computer opera esclusivamente sui dati: esso legge, scrive, riceve, invia, cancella, modifica, elabora dati e questo

può avvenire in modo puramente meccanico, al di là del significato che i dati assumeranno per noi. È importante far capire questa distinzione agli studenti delle superiori, in quanto per gli allievi della scuola elementare e della scuola media questi discorsi sono troppo impegnativi. La comprensione di come siano correlati il piano sintattico (dei dati e delle elaborazioni meccaniche) e il piano semantico (dell'interpretazione dei dati, dei risultati dell'elaborazione e dei processi computazionali) è in ogni caso un problema fondamentale della formazione informatica, anche all'università. Anche qui la questione dei linguaggi è importante, perché attraverso le forme linguistiche noi lavoriamo con dei segni e questi segni costituiscono dei dati, mentre le informazioni sono le interpretazioni che diamo ai segni.

Linguaggi iconografici e linguaggi ideografici

Alla base della maggior parte dei linguaggi c'è una convenzione. Esistono però alcune forme linguistiche che non richiedono una vera e propria base convenzionale: ad esempio i linguaggi di natura iconografica. Un linguaggio iconografico è un linguaggio che dovrebbe comunicare delle informazioni attraverso la forma, l'immagine, l'icona appunto. Quindi la percezione dell'immagine dovrebbe immediatamente richiamare l'informazione, sempre che, nell'esperienza di chi legge l'iconogramma, quella forma sia associata a dei vissuti. Il linguaggio iconografico, per questo motivo, è uno dei linguaggi più primitivi, ma anche fra quelli più costantemente usati nello sviluppo della civiltà. È un linguaggio abbastanza immediato, ma ha dei limiti che non ne consentono lo sviluppo oltre a un certo punto. Innanzitutto non tutte le cose che possiamo avere nella mente hanno una forma.

Per superare questo ostacolo si sono inventati dei segni che vengono messi convenzionalmente in corrispondenza a quei concetti. Ma ciò significa passare da un linguaggio iconografico a un linguaggio ideografico. E il linguaggio ideografico è un linguaggio che comincia a svincolarsi dall'apparenza della realtà. Il linguaggio ideografico perciò è un linguaggio che finalmente ha la potenzialità di esprimere ogni aspetto dell'esperienza. Tuttavia, esso determina a sua volta dei problemi, almeno alla luce delle funzioni che il linguaggio è venuto ad assumere nella cultura occidentale. I Cinesi e i Giapponesi hanno già sperimentato questo tipo di problemi, in particolare il gigantismo, che significa che ogni concetto nuovo deve avere un segno nuovo diverso da tutti gli altri già esistenti. Ma allora la capacità di distinguere i segni grafici è una capacità che richiede un notevole addestramento e una notevole intelligenza. Per esempio, nel linguaggio Cinese ci sono circa 90.000 segni; tuttavia è stato dimostrato che la capacità umana, per tanto che sia sviluppata, difficilmente può arrivare a distinguere più di 16.000 segni. Distinguere meno di 16.000 segni su 90.000 a disposizione significa che un Cinese non può leggere tutti i documenti che ha prodotto la sua cultura.

Inoltre, il gigantismo impedisce lo sviluppo di strumenti di comunicazione come la tipografia, perché classificare un numero di caratteri tipografici dell'ordine di 90.000, per poi potervi accedere al momento in cui servono non è certo un problema da poco. È possibile poi sviluppare una macchina da scrivere con questo tipo di scrittura? E la tastiera del computer come dovrebbe essere fatta? È di nuovo interessante osservare come certe forme linguistiche, per la loro natura, impediscano di fatto anche lo sviluppo di certe tecnologie di comunicazione. Perciò non possiamo dire che qualunque forma linguistica vada altrettanto bene dal punto di vista informatico.

Come vedremo, i linguaggi basati su un alfabeto presentano un notevole vantaggio rispetto a quelli iconografici e ideografici in quanto permettono di introdurre l'operatività. Tuttavia i linguaggi iconografici e ideografici sono preferibili in determinati contesti perché consentono una maggiore immediatezza di lettura e interpretazione. Per questo motivo, per esempio, la segnaletica stradale è un linguaggio che utilizza ideogrammi (molti dei quali sono di fatto anche iconogrammi), e solo raramente introduce informazioni in linguaggio alfabetico. Nella segnaletica stradale, infatti, la comunicazione deve essere immediata e non può essere condizionata dai tempi di lettura dei linguaggi alfabetici: l'ideogramma in quel caso è estremamente più efficace.

Operatività del linguaggio alfabetico

Le forme linguistiche a cui siamo abituati, fondate sugli alfabeti, sono più adatte ad esprimere ogni cosa senza la fatica di ricordare troppi segni. I segni sono pochi, si riconoscono e si distinguono facilmente in base al loro aspetto grafico. Il significato non è invece affidato al segno (l'alfabeto è composto da segni senza significato), ma al modo in cui si combinano i segni, cioè si introduce il concetto di parola. Questa modalità di costruire le forme linguistiche rende il linguaggio estremamente più adatto a realizzare elaborazioni, più adatto ad una forma di operatività, proprio per come è stato congegnato attraverso combinazioni di segni.

I vantaggi della forma alfabetica sono evidenti e in particolare, è opportuno sottolineare il ruolo del linguaggio alfabetico nel realizzare tutte le possibilità di comunicazione a distanza. In sostanza, con un linguaggio alfabetico è possibile esercitare la lettura e la scrittura non obbligatoriamente associata alla comprensione, cosa che con gli altri linguaggi non poteva accadere. Il concetto è molto importante, e in particolare dal punto di vista informatico: è possibile svincolare la comprensione dalla lettura e dalla scrittura, la semantica dalla sintassi, fino al punto in cui la lettura e la scrittura possono realizzarsi anche attraverso delle macchine, che ovviamente non capiscono. Il linguaggio alfabetico diventa così un fondamento dall'elaborazione dei dati, elaborazione che può avvenire nelle macchine a livello puramente sintattico, mentre l'interpretazione resta di pertinenza degli operatori umani consapevoli del significato dei dati e delle operazioni su di essi.

Linguaggi basati su un alfabeto binario

Un'altra importante conseguenza dell'introduzione dei linguaggi alfabetici consiste nella possibilità di definire linguaggi basati su due soli simboli, in qualche modo portando all'estremo l'idea di definire tutte le parole utilizzando combinazioni di pochi simboli diversi. Questo modo di operare risulta particolarmente vantaggioso nel caso in cui si vogliono utilizzare delle macchine, poiché il mondo fisico fornisce numerosi esempi di grandezze che possono assumere due valori chiaramente distinti e perciò adatti a rappresentare su un supporto fisico concreto i due segni dell'alfabeto. Ne sono esempio il nord e il sud magnetico, presi come segni; la proprietà di riflettere o meno; la presenza o meno di tensione, oppure la tensione positiva o negativa; il passaggio o meno di corrente. In ognuno di questi casi si possono usare le grandezze fisiche binarie per rappresentare parole di un linguaggio basato su un alfabeto binario. L'alfabeto nord/sud, l'alfabeto passa/non-passa, e così via, sono alfabeti diversi ma fra loro perfettamente equivalenti. Quando non siamo interessati ad entrare nello specifico della grandezza coinvolta, noi sintetizziamo il fatto che l'alfabeto è binario utilizzando i simboli 0 e 1 che diventano il nostro alfabeto di riferimento.

Trasformazioni fra linguaggi

I linguaggi basati su un alfabeto binario non si prestano alla comunicazione se questa coinvolge interlocutori umani. Pertanto, si presenta la necessità di tradurre da linguaggi ad alto livello più adatti alle modalità comunicative umane (con alfabeti più ricchi) a linguaggi più adatti alle modalità di funzionamento delle macchine, e viceversa. Anche tale compito viene normalmente delegato alle macchine e automatizzato, quantomeno se i linguaggi ad alto livello sono comunque linguaggi artificiali, concepiti allo scopo di interagire con le macchine. Anzi, si può forse dire che la maggior parte del lavoro che il computer normalmente svolge consiste proprio nel tradurre linguaggi (artificiali) in altri linguaggi artificiali collocabili a un diverso livello di astrazione. È questa, in particolare, la funzione delle interfacce, il cui ruolo è mettere in comunicazione mondi che hanno forme linguistiche diverse. In ogni elaboratore ve ne sono moltissime e ognuna di esse realizza un passaggio da una forma linguistica ad un'altra forma linguistica (e sulla base di corrispondenze strutturali).