1

# Capitolo B17
# macchine sequenziali ed elaborazioni procedurali

## Contenuti delle sezioni

a. prospettive degli esecutori di procedure p. 3
b. macchine sequenziali programmabili [1] p. 8
c. caratteristiche delle elaborazioni procedurali [1] p. 14
d. macchine sequenziali programmabili [2] p. 21
e. caratteristiche delle elaborazioni procedurali [2] p. 26
f. macchine equivalenti p. 28

29 pagine

**B17 0.01** In questo capitolo viene esaminato un esteso genere di automatismi per l'elaborazione delle informazioni che prenderemo in considerazione per allargare l'orizzonte delle attività con le quali si cercano di risolvere affidabilmente e sistematicamente i problemi che si devono fronteggiare.

Vengono dunque definiti i meccanismi automatici che chiamiamo "macchine sequenziali programmabili", in breve "MSP", che come le MSM agiscono su stringhe di caratteri di tipi definiti con le quali si intendono rappresentare tutte le informazioni che si devono gestire.

Le MSM sono state utilizzate a cominciare da B01e per le prime elaborazioni proposte, attività per le quali ci si preoccupava di convincere della attendibilità e della coincidenza dei risultati ottenibili sia da automatismi che da esecutori umani.

A questo punto servono più risultati e quindi preferiamo servirci delle MSP, macchine più versatili delle MSM grazie ai loro repertori di istruzioni notevolmente più ampi corrispondenti alla dotazione di dispositivi più prestanti e a organizzazioni interne più articolate e più adeguate a gestire attività elaborative più variegate in quanto mosse da maggiori interessi applicativi.

In effetti la definizione delle MSP riguarda le loro prestaioni e non la loro costituzione e questo cpnsente di lasciare la loro definizione piuttosto aperta in modo da poter consentire di aggiungere nuove specie di queste macchine che risultino pi'u adatte a trattare problemi nuovi.

Dalle MSP ci si aspetta una maggiore facilità per la progettazione e la descrizione delle possibili elaborazioni finalizzate alla soluzione di problemi che si prevedono via via più complessi.

Sulla organizzazione interna delle MSP non si avanzano richieste precise e stringenti, ma si lascia ampia libertà, limitandosi a presupporre, sulla base delle esperienze, la possibilità di chiarire tutti i loro dettagli.

Si mantiene comunque salda la richiesta della possibilità di ogni tipo di controlli a posteriori della affidabilità dei loro risultati e della riproducibilità delle loro prestazioni, e quindi la possibilità di consentire la possibilità di falsificazione della loro adeguatezza.

Occorre aggiungere che questa apertura vuole anche tenere presente la crescente disponibilità di dispositivi digitali dotabili di software molto efficienti e versatili, ed anche con prestazioni molto "certificate"; questi automatismi oggi possono essere pensati come gli strumenti primari per affrontare effettiva soluzione di problemi di tangibile consistenza.

Su queste macchine, anche se presentate a grandi linee, si intendono prospettare a un alto livello di generalità la nozione di elaborazione governata da indicazioni di ampia portata formalizzate nella nozione di procedura.

**B17 0.02** Delle macchine sequenziali programmabili vengono prospettati varie specie che presentano differenze costitutive anche rilevanti, ma che risulteranno avere le stesse portate o meglio potenzialità computazionali equivalenti, quando si decidedi non preoccuparsi dell'impigo dele risorse e in particolare del tempo impiegato.

Si va da macchine costituite di pochi tipi di dispositivi estremamente semplici a sistemi con una struttura molto elaborata paragonabili ai computers programmabili mediante i linguaggi procedurali oggi diffusamente disponibili.

Questa varietà consente di affrontare i moltissimi tipi di elaborazioni che si devono effettuare per affrontare i molteplici problemi che ci troviamo ad affrontare, da quelli di interesse pratico a quelli riguardanti gli obiettivi più ampi.

Quali soluzioni vanno giudicate come le più convenienti dipende fortemente dalle circostanze pratiche nelle quali vengono ricercate.
Caso per caso potrebbero imporsi le soluzioni più facilmente comprensibili, le più veloci, le più precise, le più generali, le più versatili e quant'altro.

L'aspetto al quale si ritiene opportuno prestare la maggiore attenzione è la ampiezza della portata, una caratteristica che tende ad imporsi nei tempi lunghi grazie alla disponibilità di crescente potenza e per la crescente tendenza di disporre di automatismi con prestazioni più integrate. Cresce quindi la tendenza a consolidare procedimenti risolutivi che si dimostrano convenienti in ampi ventagli di circostanze e che hanno maggiore possibilità di condurre a vere e proprie metodologie risolutive.

**B17 0.03** Questo capitolo continua a presentare molte nozioni, attività e scale di valori che presentano forti valenze formative e che consideriamo opportuno rendere rapidamente disponibili, anche in forma non definitiva e sotto forma di anticipazioni solo intuitive, rinviando la completezza dei dettagli e la robustezza logica.

Molte definizioni e molte affermazioni sono giustificate solo dalla loro plausibilità derivante dalla citazione di risultati ottenuti e consolidati in altra sede.

Per sostenere questo tipo di impostazione possiamo solo sforzarci di presentare con chiarezza esplicita i punti deboli delle argomentazioni e le affermazioni che possono solo essere anticipazioni e prospettive.

Occorre anche segnalare esplicitamente che in questo stadio dell'*esposizione* si rinviano del tutto le questioni dei precisi limiti delle prestazioni delle macchine proposte e della portata delle loro elaborazioni.

Conviene anche segnalare che l'introduzione delle procedure prepara la introduzione della nozione di infinito potenziale e fornisce un forte contributo alla formalizzazione delle conseguenze delle elaborazioni stesse, in vista del loro utilizzo per definire nuovi automatismi dotati di maggiori ambizioni provenienti in particolare dalla cosiddetta area dell'Intelligenza astificiale.

## B17 a. prospettive degli esecutori di procedure

**B17 a.01** In questo capitolo viene introdotta a grandi linee il genere esteso e ampliabile di apparecchiature in grado di operare autonomamente che chiamiamo **macchine sequenziali programmabili**, in sigla **MSP**, alle quali assegneremo il ruolo di esecutori artificiali dei processi che chiamiamo **procedure**.

Si tratta di **macchine formali**, cioè di sistemi in grado di effettuare trasformazioni di complessi di stringhe esprimenti informazioni concernenti problemi provenienti dall'esterno; le trasformazioni sono effettuate in conseguenza di regole anch'esse espresse da complessi di stringhe attribuibili a preesistenti costruttori delle macchiche, e che in un primo scenario si possono pensare scollegati dalle esigenze problematiche derivate da ambienti esterni.

Le macchine formali annunciate vogliono costituire modelli per molti tipi di dispositivi oggi disponibili e previsti; in particolare pensiamo ai computers programmabili oggi ampiamente disponibili e coinvolti in molte attività quotidiane.

Nel seguito dell'*esposizione* i computers, in forza della loro familiarità, verranno assunti, più o meno esplicitamente, come gli esecutori artificiali di prima scelta delle elaborazioni di dati quantitativi e di vari altri tipi di attività da effettuare automaticamente.

Delle MSP non diamo definizioni precise, ma ne presentiamo a grandi linee le componenti e le prestazioni con discorsi descrittivi cercando soprattutto di rendere le loro caratteristiche ampiamente e ragionevolmente accettabili.

In tal modo ci proponiamo di convincere che si tratta di un genere di macchine formali molto più versatile di quello delle MSM, le macchine sequenziali multinastro introdotte in B01, mentre rinviamo le definizioni più precise dei vari generi di macchine formali, a cominciare dalle macchine di Turing [C21] e rinviamo la dimostrazione del fatto che i vari tipi di automatismi proponibili hanno portate sostanzialmente equivalenti.

Scopo di questo modo di procedere è quello di procedere a introdurre svariati strumenti con i quali si possano discutere, talvolta anche limitandoci a ricorrere alla sola intuizione, molteplici attività che consentono di porsi problemi di interesse generale, in particolare la possibilità di servirsi della nozione di infinito potenziale.

Accettando di rimanere nel generico, si riesce a procedere con una certa speditezza alle descrizioni dei loro dispositivi e delle loro prestazioni e in particolare a rendere comprensibile la loro versatilità; in tal modo risulta più agevole procedere nella introduzione della vasta gamma delle loro prestazioni e delle conseguenti definizioni di numerose nozioni matematiche.

**B17 a.02** Mentre per le considerazioni iniziali sopra le elaborazioni automatiche come esecutrici artificiali sono state preferite le MSM, per proseguire più agevolmente ci appoggiamo su macchine esecutrici più elaborate grazie alla disponibilità di dispositivi più prestanti forniti dalla odierna tecnologia digitale e alla possibilità di crescere attraverso operazioni che consentono il riutilizzo di meccanismi adottati in precedenti esperienze altrui e attraverso il miglioramento delle proprie prestazioni derivato dalla riflessione sulla bontà dei propri comportamenti.

L'interesse nei confronti delle macchine formali e alle apparecchiature che essi schematizzano è dovuto alla convinzione che lo studio sistematico delle soluzioni dei problemi per essere efficace, sia sul piano sociale che su quello culturale, non può evitare di dare importanza primaria a questi strumenti concettuali e materiali.

Nelle considerazioni generali, dopo aver privilegiate le elaborazioni consistenti in manipolazioni di stringhe, per avvicinarci alle prestazioni dei computers programmabili prenderemo in considerazione anche le manipolazioni dei oggetti meno generici, a partire dai numeri interi, fino a servirci di strutture di dati piuttosto elaborate.

A questo potrà contribuire anche la introduzione di un linguaggio di programmazione, detto linguaggio c++, che viene esposta in B70 e B71.

**B17 a.03** Abbiamo introdotte come prestazioni fondamentali delle MSM alcune operazioni sopra le stringhe come copiatura, controllo di uguaglianza, giustapposizione e trasformazione nella stringa riflessa. Un operatore umano le attua senza difficoltà servendosi di carta e matita sia per rappresentare le stringhe che per effettuare le loro trasformazioni.
Un esecutore automatico si serve invece di manovre che si sono abbondantemente osservate risultare equivalenti.

Sulla base di queste constatazioni si possono garantire le operazioni strutturalmente più articolate che abbiamo visto nei capitoli precedenti.
In particolare gli operatori umani e artificiali sanno effettuare affidabilmente calcoli con numeri interi. L'esecutore umano sa eseguire operazioni aritmetiche non troppo complicate e sa servirsi della calcolatrice elettronica.
Inoltre si sanno redigere le formulazioni dei risultati di applicabilità di algoritmi collaudati mediante termini e formule che si possono proporre come ampiamente condivisibili.
Il fatto di servirsi di stringhe di caratteri scelti attraverso convenzion condivisibili, stante l'adattabilità delle convenzioni e la facilitazione della implementazione degli adattamenti consentita dai meccanismi disponibili (a cominciare da quelli elementari come la sostituzione di caratteri in stringhe [B04a] e la trasduzione a stati finiti [C13]) consente la estesa riutilizzabilità di risultati e schemi operativi in vista delle soluzioni dei nuovi problemi che ci si aspetta di dover affrontare.

**B17 a.04** Ci si chiede su cosa si basa la correttezza delle regole che governano le elaborazioni sopra stringhe da parte degli esecutori di procedure.

Per gli esecutori MSM si ricorre al fatto che costituiscono uno schema di esecutore che rispecchia i comportamenti che molti esecutori umani hanno tenuti per loro attività che si sono osservate in grado di risolvere effettivamente vari problemi concreti.

Tra le prime regole rispettate nelle elaborazioni riconosciute affidabili vi sono la associatività e la noncommutatività della giustapposizione.
Per l'associatività si verifica che per molte terne di stringhe $\langle \alpha, \beta, \gamma \rangle$ accade che $(\alpha \cdot \beta) \cdot \gamma = \alpha \cdot (\beta \cdot \gamma)$ .
Per la noncommutatività si individuano numerose coppie $\langle \alpha, \beta \rangle$ di stringhe sopra un alfabeto costituito da almeno due caratteri per le quali si verifica materialmente, si constata, che $\alpha \cdot \beta \neq \beta \cdot \alpha$ .

Queste considerazioni dicono che i comportamenti fondamentali degli esecutori di procedure sono stati riscontrati affidabilmente corretti da osservazioni sistematiche e chiaramente evidenti.

Per quanto riguarda la noncommutatività è opportuno rilevare che essa consente di rispettare la noncommutatività dei processi tangibili, i quali si svolgono nel tempo: nello svolgimento temporale degli eventi la noncommutatività è una caratteristica ineludibile.

**B17 a.05** Prima di procedere a descrivere le MSP conviene precisare una caratteristica delle MSM che può essere estesa e notevolmente arricchita per le MSP: la possibilità che un automatismo utilizzi le prestazione di un altro automatismo collaudato dello stesso genere e la adozione metodica di questo

genere di composizione per far crescere il complesso degli automatismi disponibili per l'*apparato* e conseguentemente per far crescere il complesso dei problemi che si sanno risolvere affidabilmente.

Per questa adozione metodica si parla di **modularità** nella organizzazione degli automatismi e, a un livello più comprensivo, di modularità di approccio dei problemi.

Consideriamo di conoscere una MSM $\boldsymbol{M}_1$ in grado di risolvere le istanze di un problema $\mathcal{P}_1$: la soluzione della istanza di $\mathcal{P}_1$ determinata dal dato $D_i$ che attribuiamo a un tipo $T_1$ la descriviamo come la trasformazione della stringa in genere composita $D_i$ da parte di $\boldsymbol{M}_1$ attraverso un numero finito di passi nella stringa dei risultati $R_i$ che attribuiamo a un tipo $U_1$.

Si richieda inoltre una MSM $\boldsymbol{M}_2$ in grado di risolvere un problema tendenzialmente più elaborato $\mathcal{P}_2$ che coinvolge tra gli altri i dati $\overline{D}_i$ del tipo $T_2$ che risulta vantaggioso trasformare nei corrispondenti risultati $\overline{R}_i$ del tipo $U_1$, cosa attuabile con l'utomatismo $\boldsymbol{M}_1$.

Questa trasformazione viene considerata un sottoproblema per $\mathcal{P}_2$ e la trasformazione vantaggiosa si cerca di ottenerla con la transcodifica dei dati $\overline{D}_i$ nei $D_i$ con l'ottenimento dei risultati intermedi $R_i$ grazie al lavoro di $\boldsymbol{M}_1$ e con la transcodifica degli $R_i$ nei $\overline{R}_i$.

Se le due trasformazioni chiamate transcodifiche sono fattibili affidabilmente mediante operazioni che si possono considerare semplici il sottoproblema può venire risolto con il riutilizzo di $\boldsymbol{M}_1$ da parte di $\boldsymbol{M}_2$: questa macchina deve organizzare la prima transcodifica come trasformazione dei contenuti $\overline{D}_i$ registrati su una sua memoria nei corrispondenti $D_i$ su una memoria di ingresso di $\boldsymbol{M}_1$ per dare avvio dell'azione di questa macchina che in tal modo assume il ruolo di **sottomacchina** di $\boldsymbol{M}_2$ e con la successiva transcodifica degli $R_i$ da una memoria di uscita di $\boldsymbol{M}_1$ nei corrispondenti $\overline{R}_i$ su una memoria intermedia di $\boldsymbol{M}_2$.

Conviene sottolineare che le manovre per il riutilizzo di una sottomacchina, azione che viene chiamata anche **richiamo di una sottomacchina**, sono tutte evidentemente disponibili attraverso nastri per le MSM, e sono da supporre fattibilidalle MSP utilizzando semplici memorie, scrivendo gruppi di istruzioni per la manipolazione di dati e risultati e per le transcodifiche e gestendo cambiamenti di stati che consentano la organizzazione complessiva delle accennate manovre per il riutilizzo di $\boldsymbol{M}_1$.
Questo modo di organizzare il lavoro di una macchina $\boldsymbol{M}_2$ può essere ampiamente esteso al riutilizzo di molte MSM precedentemente studiate e sperimentate. Evidentemente si possono anche utilizzare (o richiamare) macchine che si servono di altre macchine con più livelli di richiamo.
Per questo modo di operare si parla di **organizzazione modulare** delle macchine.

**B17 a.06** In alternativa alla organizzazione sopra descritta si può pensare di evitare le manovre di richiamo di sottomacchine precedenti con un lavoro di trasformazione delle istruzioni delle sottomacchine richiamate $\boldsymbol{M}_1$ in istruzioni per la macchina richiamente $\boldsymbol{M}_2$. In tal modo si possono evitare manovre di collegamento come le transcodifiche e quindi si potrebbero avere elaborazioni più efficienti per quanto riguarda i tempi di esecuzione.
Per questo modo di fare parliamo di "fagocitazione delle istruzioni" contenute nella sottomacchina.

In linea di massima la fagocitazione delle istruzioni può essere opportuna solo nelle situazioni nelle quali ha grande peso la velocità dell'esecuzione e per questa risulta conveniente investire nel lavoro di adattamento delle istruzioni.

Dal punto di vista dello sviluppo complessivo delle attività computazionali l'organizzazione modulare è di gran lunga preferibile nella massima parte delle attività, soprattutto in quanto favorisce la crescita della gamma delle elaborazioni che si possono impiegare e in quanto rende molto più adattabile ogni parco di macchine uti;izzabili per scopi specifici.

In genere il lavoro di precisazione minuta delle istruzioni di un programma viene considerato troppo dispendioso rispetto ai vantaggi che può procurare il corrispndente aumento della velocità delle esecuzioni.
Viene valutata molto più importante la possibilità di affrontare rapidamente nuovi problemi nei quali è probabile si possano individuare sottoproblemi riconducibili a problemi già affrontati e risolti.

Sono frequenti le situazioni nelle quali si procede ad ampliare la portata di procedimenti collaudati facendo crescere la loro versatilità, al fine di consentire che nuove varianti dei problemi precedentemente risolti possano essere considerate agevolmente risolvibili.

Con i lavori di ampliamento della portata spesso si possono ottenere anche generalizzazioni delle nozioni matematiche che emergono dalla analisi delle possibili soluzioni e si possono avere arricchimenti delle relazioni tra le entità formali con le quali si ossono sviluppare nuovi procedimenti e nuove dimostrazioni.

Possiamo anche dire che preferire l'atteggiamento della modularità, rispetto all'atteggiamento della fagocitazione costituisce una scelta più "strategica" e "matematica" che "tecnico-efficientistica" e che essa porta a una più rapida crescita della cultura computazionale.

I vantaggi della modularità sono evidenti soprattutto dal punto di vista dei committentiabituali di nuove applicazioni. Essi possono confidare nella possibilità di disporre in tempi molto più brevi di procedure che consentono di affrontare nuovi problemi.
In termini di macchine si può pensare di mettere a punto nuove macchine e procedimenti innovativi in tempi relativamente contenuti e, soprattutto, in tempi prevedibili, con investimenti di risorse pianificabili e con un relativo contenimento dei rischi, una macchina risolutrice che si avvale del riutilizzo di una quantità di macchine collaudate quando deve affrontare la definizione di nuovi procedimenti innovativi con le incognite che comportano e che riesce a organizzare il complesso delle sue attività attraverso richiami di sottonmacchine con tecniche e pratiche che la "coltura della modularità" ha avuto modo di sviluppare.

**B17 a.07** Le considerazioni e le prospettive precedenti sono da inquadrare nel vasto scenario della evoluzione delle tecnologie.

Grande influenza hanno i progressi nella componentistica.
La miniaturizzazione consente di organizzare con la modularità gli stessi circuiti che si avvalgono di sottocircuiti che in anni recenti costituivano microcomputers con una loro marcata autonomia.

I progressi delle telecomunicazioni consentono di avere computers in locazioni distanti che colloquiano con elevati flussi di informazioni per sostenere collaborazioni per la soluzione di problemi specialistici molto articolati.

La modularità ha portato alla disponibilià di enormi librerie di sottoprogrammi, con il ruolo delle sottomacchine software; questi sottoprogrsmmi spesso sono scritti in diversi linguaggi di programmazione che mettono concretamente a disposizione vaste esperienze di risoluzioni di problemi.

Con la miniaturizzazione si dispone di sistemi di migliaia e milioni di computers che possono comunicare tra di loro attraverso canali paralleli molto performanti per portare avanti elaborazioni parallele che hanno rese affrontabili elaborazioni che pochi anni orsono erano giudicate intrattabili.
Con la costruzione dei cosiddetti sistemi cloud sono disponibili su richiesta e con con relativa facilità grandi prestazioni computazionali e accessi a grandi quantità di dati.
Lo sviluppo di Internet favorisce la collaborazione tra addetti ai lavori che sempre di più si scambiano informazioni e competenze che trovano vantaggioso condividere.

Tutti questi avanzamenti evidentemente favoriscono in misura molto rilevante la pratica della modularità e di conseguenza una sua cultura.

**B17 a.08** nella prospettiva della adozione della modularità e degli avanzamenti tecnologici e culturali ocorre riprendere le osservazioni secondo le quali tutte le elaborazioni, quale che sia la natura dell'esecutore che le effettua, richiedono risorse che si possono attribuire a tre grandi generi: supporti per la registrazione delle informazioni, tempo richiesto dalle elaborazioni e articolazione dei sistemi delle istruzioni operative.

Per le valutazioni di questi tre generi di risorse si parla anche, risp., di complessità spaziale, di complessità temporale e di complessità strutturale.

Come si è accennato è raramente determinante la velocità delle elaborazioni.
Per quanto riguarda i limiti raggiungibili conviene fare riferimento primariamente alle prestazioni dei computers oggi ampiamente disponibili, sia per le elevate velocità che possono raggiungere, sia per la possibilità di monitorare con accuratezza i loro comportamenti di fronte a problemi concreti.

Possono essere utili anche previsioni sulle prestazioni dei dispositivi la cui disponibilità nel prossimo futuro è ragionevolmente valutabile; queste previsioni spesso sono rafforzate dalla possibilità di analizzare i progetti che riguardano la utilizzazione attraverso modalità consolidate di componenti già collaudati nei laboratori e di possibili innovazioni indagate con sistematicità.

È inoltre opportuno segnalare la sostanziale continuità con la quale possiamo risalire dalle odierne prospettive delle tecnologie computazionali alle attività che si stanno concretamente svolgendo con la strumentazione e le competenze del presente, alle storie sui primi sviluppi degli studi quantitativi fino a confrontarle, con una perdonabile forzatura sull'efficienza, con i successi ottenuti per risolvere antichi problemi dai soli operatori umani che, ordinati e diligenti, sono stati capaci di portare a termine affidabilmente complessi di operazioni elementari come copiature, piccole modifiche e ordinamenti di stringhe e semplici calcoli.

## B17 b. macchine sequenziali programmabili [1]

**B17 b.01** Introduciamo qui le caratteristiche salienti delle elaborazioni eseguibili dalle MSP, gli automatismi dei quali intendiamo avvalerci partendo dalle esigenze che esse possono soddisfare e dalle loro possibilità, ponendoci l'obiettivo di individuare loro esemplari equivalenti che possano risultare ampiamente utili in relazione alle problematiche più rilevanti.

Non entreremo in molti dettagli, ma presenteremo richieste e risultati che potranno essere pienamente giustificati più avanti [in particolare v. C12a].

Mentre in linea di principio tutte le soluzioni di problemi si possono ottenere servendoci soltanto di confronti, di composizioni e di trasformazioni di stringhe, nella pratica risulta più vantaggioso e più significativo fare riferimento a entità come numeri di vari generi, digrafi, strutture algebriche, strutture relazionali, figure geometriche, ..., sorvolando sulle caratteristiche delle stringhe che in ogni caso consentono di rappresentarle.

**B17 b.02** Nei paragrafi successivi introdurremo le caratteristiche delle macchine sequenziali programmabili, automatismi in grado di elaborare stringhe e le varie entità rappresentabili con stringhe attraverso sequenze di manovre molto semplici o riconducibili a manovre molto semplici, secondo il piano di approccio che segue.

Fondamentalmente chiediamo che le MSP costituiscano modelli di meccanismi artificiali tali che ci si possa aspettare ampio consenso sulla effettiva possibilità di costruirli e farli funzionare affidabilmente. Un valido sostegno a questo consenso lo danno i risultati che hanno ottenuto e continuano a ottenere gli automatismi meccanici, elettrici ed informatici che sono stati utilizzati con risultti soddisfacenti r coerenti.

Queste macchine saranno definite senza entrare in molti dettagli costruttivi, ma prospettando modelli che possano essere ampiamente accettati come apparecchiature i cui dettagli sono ampiamente documentati, che possono essere realizzate e riprodotte senza difficoltà e che sono stati in grado di eseguire una estesa gamma di elaborazioni di concreta utilità.

Chiederemo anche che tutti i tipi di MSP possano essere simulate dalle macchine del genere MSM che abbiamo incontrate, con la conseguenza di poter disporre senza soluzioni di continuità di tutti i risultati ottenuti in precedenza.

Assumeremo un atteggiamento di genericità al fine di lasciare molto aperta la gamma dei modelli delle MSP, modelli che, come vedremo, possono comprendere macchine con componenti estremamente semplici come le macchine di Turing e sistemi estremamente articolati e differenziati come gli odierni computers. Consentiremo anche la possibilità di interconnetterli per costituire grandi sistemi compositi, anche distribuiti e in grado di cooperare per affrontare problemi molto impegnativi.

**B17 b.03** A ciascuna MSP **M** possono essere affidate diversi sistemi di dati iniziali, ciascuno dei quali esprimibile con una stringa D che può essere molto articolata.

Questa **M**, in seguito alla ricezione di una D avvia l'esecuzione di una elaborazione; le elaborazioni effettuabili dalla **M** costituiscono la cosiddetta **famiglia delle elaborazioni** eseguibili dalla **M**, famiglia i cui membri sono indicizzate dalle stesse singole D.

Le elaborazioni che possono essere effettuate dalla MSP **M** costituiscono le **prestazioni della macchina**. Queste elaborazioni possono aver termine, ma questo non chiediamo che possa essere garantito.

Quando una elaborazione giunge a termine risulta fornito un complesso di informazioni che costituiscono i cosiddetti **risultati della elaborazione**; questo può essere rappresentato da una stringa che potrebbe essere molto articolata.

Ogni MSP quindi determina una relazione funzionale costituita dalle coppie $\langle$ dati, risultati $\rangle$.

Due diverse MSP si dicono equivalenti sse sono in grado di accogliere gli stessi dati e per ciascuno di questi effettuano due elaborazioni finite che producono gli stessi risultati.
Quindi due MSP sono equivalenti sse presentano la stessa famiglia di elaborazioni, ossia sse posseggono lo stesso insieme di prestazioni.

Occorre osservare che i due risultati uguali forniti a parità di dati da due MSP equivalenti possono essere ottenuti con elaborazioni molto diverse e sulle differenze di queste attività possono essere esercitate preferenze che tengono conto di molti dettagli, ovvero di molte preferenze che possono richiedere di essere valutate con cura.

**B17 b.04** Un aspetto determinante dello studio delle prestazioni delle MSP sta nel fatto che ogni famiglia di elaborazioni può essere eseguita da una grandissima varietà di MSP equivalenti e può accadere che due tali macchine dispieghino manovre molto diverse le quali in particolare possono richiedere consumi di risorse molto diversi.

Possiamo anche anticipare che le MSP riescono a eseguire una gamma molto vasta di elaborazioni specifiche, ossia che posseggono una gamma molto estesa e variegata di prestazioni.

Ci possiamo quindi aspettare un vasto complesso di quelle che chiamiamo **elaborazioni eseguibili dalla macchina** MSP.

Si può osservare che il piano di approccio ora presentato rinuncia a collegarsi all'esecuzione di elaborazioni da parte di esecutori umani: in effetti per svolgere seriamente un tale esame, richiederebbe di affrontare dettagli di natura psicologica o comportamental, attività molto impegnativa che allontanerebbe la presentazione dei molti risultati specifici che consideriamo invece importanti ie in un certo senso urgenti.

Qui ci limitiamo semplicemente a ricordare che tutte le elaborazioni eseguibili da automatismi in linea di principio possono essere eseguite anche dagli operatori umani e ci limitiamo ad accennare che queste ipotizzate esecuzioni equivalenti a quelle delle MSP in genere richiederebbero risorse di tempo e di attenzione ai particolari tanto impegnative da essere giudicate improponibili.

La precedente affermazione di equivalenza in linea di principio si può sostenere con la constatazione che ogni operatore umano sufficientemente affidabile è in grado di effettuare le operazioni basilari sulle stringhe alle quali è riconducibile ogni elaborazione automatica da parte delle MSP (e quindi delle MSM).

**B17 b.05** Ad una MSP si chiede di operare per passi successivi in ciascuno dei quali effettua una manovra che va considerata semplice in relazione ai dispositivi operativi dei quali si suppone dotata.
Questo consente di sostenere che ogni MSP ha la capacità di realizzare **evoluzioni sequenziali deterministiche**.

Una MSP presenta varie componenti:

- un **controllo**, dispositivo che ha un ruolo di supervisore per le attività della macchina;
- dispositivi di memoria ad accesso diretto che potrebbero essere sostituiti da nastri indirizzabili i quali consentono la lettura e la registrazione dei dati attraverso canali di lettura/scrittura di alte prestazioni;

- un apposito dispositivo di memoria che chiamiamo **supporto delle istruzioni** nel quale in una fase che precede e prepara un gruppo di elaborazioni viene registrata una sequenza di stringhe dette **istruzioni [operative]**. Queste sono in grado di controllare operazioni sui dati la cui esecuzione demandano ad appositi circuiti;
- **circuiti operativi** che effettuano trasformazioni di raggruppamenti di dati da considerare semplici nella descrizione delle prestazioni della MSP stessa.

Per i dati trattabili chiediamo solo che possano essere ricondotti a stringhe sopra definiti alfabeti chiaramente riconoscibili da tutti gli esecutori delle squadre impegnate nelle varie problematiche; particolare importanza hanno i dati che possono essere rappresentati mediante sequenze binarie.
In effetti tutte le informazioni esprimibili con i caratteri di un qiualsiasi alfabeto si possono rappresentare con sequenze binarie.

**B17 b.06** Nella pratica della odierna elettronica digitale tutte le informazioni esprimibili con sequenze binarie vengono espresse in celle di memoria disposte sequenzialmente di un solo tipo o di pochi tipi multipli di un tipi di base.
Come cella di memoria di base si è imposto il byte, ossia l'ottetti di bits, dispositivo i cui possibili contenuti possono essere espressi dagli interi da 0 a $2^8 - 1 = 255$, dagli interi da -128 a + 127, oppure dalle coppie di cifre esadecimali 00 a FF.
Spesso si prwndono in considerazione celle più estese come halfwods di 2 bytes o 16 bits, words di 4 bytes o 32 bits,double words di 8 bytes o 64 bits e long words di 16 bytes o 128 bits.
Le disposizioni e le interpretazioni delle informazioni in queste celle seguono regole di codifica accettate internazionalmente e consistenti in codici come ASCII, ANSI o Unicode.

Per quanto riguarda le possibilit a funzionali chiediamo che il complesso delle MSP sia in grado di affrontare, direttamente o indirettamente, tutte le entità e tutti i problemi ben definiti incontrati finora servendosi di computers elettronici e in particolare tutto quanto è risultato trattabile affidabilmente con macchine MSM.

In dettaglio si chiede che siano trattabili valori di verità (falso o 0 e vero o 1), caratteri di alfabeti delle lingue naturali, numeri interi, delimitatori e separatori vari (e quindi liste, insiemi finiti e strutture finite dei vari generi incontrati), segni di operazioni numeriche e simboli convenzionali (e quindi espressioni matematica e dei linguaggi artificiali adottati dalle molteplici comunità scientifico-tecniche).

Dal punto di vista della organizzazione complessiva dello sviluppo dell'*apparato* risulta conveniente pensare a squadre di MSP che cooperano per affrontare le diverse problematiche.
In ciascuno di questi ambiti si individua un insieme dei tipi di dati trattabili, anche più articolati di quelli prospettati genericamente poco sopra, insieme al complesso di regole che riguardano la loro codifica in opportune celle di memoria.

Le celle della memoria possono trovarsi in diversi supporti corrispondenti a diverse caratteristiche costitutive e operative (ROM, RAM, su disco, su nastro, ...) e si chiede che possano essere indirizzate (direttamente o indirettamente) servendosi per ciascuna cella di un identificatore che la caratterizza.
Si chiede che questi identificatori possano essere ricondotti a numeri progressivi che costituiscono indirizzi di memorie sui quali si possano effettuare operazioni aritmetiche, utili in particolare per gestire scorrimenti e salti.

Una MSP con questi meccanismi di indirizzamento uò evidentemente simulare le MSM.
Si dimostra viceversa, attraverso molti dettagli, che una MSP può essere simulata da una MSM; questo richiede di precisare manovre di esteso scorrimento di lunghe sequenza di istruzioni per MSM carat-

terizzate da numerosissimi identificatori di registri in grado di simulare operazioni di indirizzamento della MSP ottenute con istruzioni apposite.
In questo modo si stabilisce che i programmi delle MSM possono essere sostituiti da programmi per le MSP che vengono eseguiti con instruzioni spesso consecutuive alternate da meno frequenti salti da un'istruzione ad una distante, in avanti o all'indietro; questo richiede che anche le istruzione delle MSP siano indirizzabili attraversi indici numerici gestibili con operazioni numeriche.

**B17 b.07** Il realismo impone che i supporti fisici per la registrazione delle informazioni, i dispositivi di memoria, siano finiti.
Si consente tuttavia che nel corso di una elaborazione di un nuovo tipo, i suddetti supporti possano venire estesi al fine di soddisfare nuove esigenze impreviste che si vengano a manifestare nel corso della elaborazione stessa.
Questo in linea di principio; nella pratica una possibile estensione risulta condizionata dalla effettiva disponibilità delle nuove risorse richieste e dalla valutazione della convenienza di proseguire nella elaborazione corrente.

Le istruzioni devono potersi organizzare in sequenze che si propongono di essere in grado di governare evoluzioni della MSP che abbiano senso applicativo, cioè che possano contribuire a risolvere incisivamente problemi consistenti.

Si chiede che le sequenze di istruzioni di una MSP possano essere presentate seguendo una loro adeguata struttura organizzativa, ovvero si chiede che possano considerarsi le stringhe di un linguaggio formale specifico che soddisfi le diverse esigenze di natura linguistica (lessicale, sintattica e semantica).

Inoltre occorre pensare che vi sia la possibilità di un intervento sistematico di operatori umani, non pochi, incaricati della redazione delle istruzioni per le MSP, operatori che vengono chiamati **programmatori**.
Al linguaggio delle istruzioni si chiede anche di essere soddisfacente sul piano pragmatico, cioè di essee facilmente comprensibile per i programmatori; questi devono essere in grado di collaborare alla redazione di liste di istruzioni richieste frequantemente che chiamiamo **routines**.
Per le routines sono da auspicare caratteristiche di elevata riutilizzabilità, adattabilità e versatilità.

Sono auspicabili complessi organici di routines orientate alla soluzione di problemi di rilevanza applicativa; quelli più consistenti vengono detti **programmi per la MSP** e si dice che costituire la **implementazione di una procedura**.

Abbiamo dunque presentate molte caratteristiche delle macchine sequenziali programmabili sotto forma di auspici; per ciascun genere di queste macchine si dovrà dimostrare, o almeno convincere, che sia realizzabile ed efficace.

**B17 b.08** Il controllo, o unità di controllo, di una MSP a ogni passo evolutivo si prevede che svolga in successione le tre attività che seguono.

- Riconosce l'istruzione attuale (il suo tipo, i suoi dati, ...) e la interpreta per demandare la sua esecuzione agli opportuni circuiti operativi;
- attende che i circuiti restituiscano le informazioni risultato della istruzione eseguita e cura che esse vengano registrate nelle dovute unità di memoria;
- stabilisce quale altra istruzione dovrà essere eseguita nel passo successivo.

Il controllo inoltre gestisce (cioè consulta e modifica) una particolare informazione variabile che può assumere un insieme finito e predeterminato di valori che potrebbero essere rappresentati da interi

naturali scelti con un criterio convenzionale e che contribuisce in modo rilevante a determinare il comportamento del controllo stesso.
I suddetti valori si dicono contraddistinguere gli **stati interni della macchina** MSP e si dice che il controllo a ogni passo si viene a trovare in uno stato interno.
A questo scopo si consente che il controllo si serva di proprie unità di memoria che chiamiamo registri interni, o in breve **registri**.

**B17 b.09** I successivi passi di una elaborazione conviene che siano riferiti alle successive componenti di una sequenza di tempi; questa la rappresentiamo con una sequenza iniziale di interi naturali, numeri che consideriamo come i valori di un parametro che diciamo **tempo discreto** e che specifichiamo servendoci di una apposita variabile "$\mathfrak{t}$": i successivi passi sono dunque contraddistinti dai successivi valori $\mathfrak{t} = 0$, $\mathfrak{t} = 1$, $\mathfrak{t} = 2$, ... .

Ogni passo evolutivo vede una transizione della MSP da una precisa configurazione attuale a una configurazione successiva come conseguenza determinata dell'esecuzione dell'istruzione attuale.

Una evoluzione di una MSP, ovvero una elaborazione che essa esegue, consiste nella sequenza delle configurazioni che essa via via assume. Nell'istante iniziale, caratterizzato dall'istante temporale $\mathfrak{t} = 0$, il meccanismo si trova nella sua **configurazione iniziale**: questa corrisponde a una sequenza formata dalle seguenti componenti:

- un particolare stato interno, detto **stato iniziale**;
- le registrazioni su un adeguato numero di unità di memoria dei dati iniziali; può essere utile descrivere tali dati come forniti da un committente dell'elaborazione che sta per iniziare il quale si aspetta che la MSP fornisca un risultato che dia una soluzione a una sua istanza di problema o che almeno porti qualche chiarimento a tale istanza;
- le posizioni di specifici supporti di memoria dedicati alla registrazione di specifici dati in corso di elaborazione; in particolare talune di queste registrazioni all'inizio dell'esecuzione riguardano i dati iniziali e alla fine dell'esecuzione riguardano i risultati.

**B17 b.10** Esaminiamo l'esecuzione di una istruzione **I**.

Il controllo, dopo averla riconosciuta si predispone a farla eseguire: inizialmente si fa fornire da particolari unità di memoria (di varia costituzione fisica ma riconducibili a celle di nastri controllabili con le testine di letture opportunamente posizionate) i dati con il ruolo di operandi sui quali la **I** è deputata ad agire.

Successivamente l'unità di controllo, a partire da questi dati e da altri eventuali dati a sua disposizione (contenuti nella **I**, in propri registri o implicati dal suo stato interno), decide:

- se chiedere ai circuiti operativi di modificare i contenuti delle posizioni da lei giudicate da trasformare ed eventualmente di modificare lo stato interno e propri registri;
- se modificare propri registri in uno dei modi previsti (in particolare se muovere le testine di alcuni nastri in una delle direzioni percorribili);
- se passare a una delle istruzioni che il programma considerava come possibili successive della istruzione corrente.

Assumiamo come requisito essenziale di ogni MSP e di ogni suo programma che le sue istruzioni e i dati a sua disposizione siano interpretabili dal controllo senza incertezze e le sue istruzioni siano eseguibili univocamente.

L'elaborazione procede secondo le modalità ora descritte in tutti i passi che seguono l'iniziale e determina le configurazioni successive riferibili, come si è detto, ai valori della variabile tempo discreto $\mathtt{t} = 1,\ \mathtt{t} = 2,\ \mathtt{t} = 3,\ ....$

Con il procedere dei passi evolutivi il controllo potrebbe incontrare una una istruzione esplicita di **arresto [dell'eleborazione]**, oppure trovarsi in una configurazione che vede esauriti i dati che dovrebbero alimentare nuovi passi evolutivi e di conseguenza decidere di porre fine alla sua evoluzione, o, in altre parole, di decidere di effettuare il proprio arresto.

## B17 c. caratteristiche delle elaborazioni procedurali [1]

**B17 c.01** Le elaborazioni automatiche nell'odierna società rivestono importanza primaria, in quanto consentono di affrontare una massa di problemi che senza ricorrere ad automatismi non potremmo neppure iniziare ad affrontare.
Le elaborazioni automatiche, sempre più spesso attraverso il Web, permettono il funzionamento di dispositivi che sono utilizzati sostanzialmente in tutte le attività: reggono l'organizzazione dei sistemi amministrativi, produttivi e logistici; gestiscono conoscenze basate su archivi digitali, sostengono tutti i tipi di ricerche innovative.

Grande importanza rivestono le richieste generali di riproducibilità e di controllabilità delle elaborazioni automatiche.
Si vuole che ogni elaborazione effettuata da un esecutore sia potenzialmente analizzabile da qualche altro esecutore (umano o artificiale) che in tale azione svolge un ruolo di supervisore. Questo supervisore deve essere in grado di conoscere quello che si riscontra nelle successive configurazioni che costituiscono l'elaborazione e di giustificare ogni passaggio da una configurazione alla successiva sulla base dell'istruzione corrente, dello stato interno e delle informazioni utilizzabili dall'istruzione stessa.
Per un quadro più completo conviene prospettare che un supervisore possa anche avere il compito di riferire ai committenti delle elaborazioni gli elementi che riguardano le loro aspettative.

**B17 c.02** Vediamo una prima classificazione delle valutazioni di una elaborazione che un supervisore può segnalare in relazione alle aspettative dei committenti.

Una elaborazione si dice **elaborazione finitamente pertinente** sse giunge ad arrestarsi in una certa configurazione finale nella quale si individua un complesso di stringhe risultato che forniscono una risposta pertinente alla richiesta che ha originato l'elaborazione stessa; La valutazione della risposta da parte del committente può andare dall'essere pienamente soddisfacente all'essere del tutto deludente.

Sul piano formale l'arresto può vedersi come la conclusione di una serie di trasformazioni di rappresentazioni di oggetti che riguardano il problema con la individuazione di una o più stringhe che esprimono obiettivi attesi o paventati dal committente.

Nei casi più semplici si ha una stringa che determina un numero, un nome, una risposta positiva o negativa a una domanda a due vie o una valutazione di possibili scelte per decisioni che il committente si prepara a compiere.
In altri casi si ottengono più stringhe risultato che esprimono entità matematiche o strutture informative che possono anche essere molto elaborate (si pensi per esempio, al programma compilato che produce un programma `compilatore (we)`].

Alternativamente può accadere che una elaborazione si concluda con l'arresto delle operazioni in una **configurazione anomala** che riguarda una situazione che il committente non aveva prevista o aveva temuta oppure una situazione che non consente di individuare stringhe risultato che si possano interpretare come pertinenti. Una tale elaborazione si può chiamare **elaborazione con conclusione erronea**.

Può invece accadere che dopo un elevato numero di passi l'elaborazione non sia giunta a una conclusione. Una tale situazione deve essere riferita a un ben determinato stadio dell'evoluzione, va giudicata adeguata o meno in relazione alle risorse, soprattutto temporali, impiegate e va collegata alle possibilità di prosecuzione.
Questa eventualità che qui ci limitiamo a segnalare, dovrà essere adeguatamente analizzata più avanti, a cominciare da B18a e B18f.

**B17 c.03** La presentazione delle elaborazioni fatta finora è solo una presentazione di prospettiva, in quanto non pretende di fornire una descrizione esauriente e generale delle elaborazioni e neppure intende dare una solida giustificazione di una qualche loro validità.
Si è soltanto inteso prospettare argomentazioni volte a fornire una ragionevole convinzione che le elaborazioni che si stanno per incontrare nei prossimi capitoli siano riconducibili ai meccanismi descritti solo schematicamente e che questi meccanismi siano costruibili con elevata affidabilità e che le elaborazioni loro demandate siano eseguibili concretamente, siano riproducibili e siano supervisionabili e conseguentemente si intende convincere che i procedimenti costruttivi che tali elaborazioni mettono in atto siano ampiamente riutilizzabili.

**B17 c.04** Introduciamo ora la relazione di **simulazione tra macchine sequenzialiprogrammabili**, relazione essenzialeper organizzare con sufficiente chiarezza le conoscenze sopra le possibilità operative e le prestazioni delle apparecchiature alle quali demandiamo le elaborazioni condivisibilmente affidabili.
Va anticipato che queste prestazioni si confida siano molto incisive e molto variegate.

Date due MSP, $\mathbf{M}_1$ e $\mathbf{M}_2$, si dice che $\mathbf{M}_2$ **simula la macchina** $\mathbf{M}_1$ sse si individuano altre due macchine $\mathbf{C}$ e $\mathbf{D}$ con le seguenti proprietà:

$\mathbf{C}$ è in grado di trasformare con un numero finito di passi ogni programma $\mathsf{P}_1$ per la $\mathbf{M}_1$ e ogni sistema di dati di ingresso $D_1$ per per la $\mathbf{M}_1$, risp., in un programma $\mathsf{P}_2$ per la $\mathbf{M}_2$ e in un sistema di dati di ingresso $D_2$ per $\mathbf{M}_2$;

$\mathbf{D}$ è in grado di trasformare con un numero finito di passi ogni complesso di risultati $R_2$ ottenuti da $\mathbf{M}_2$ con l'elaborazione a partire da $D_2$ retta da $\mathsf{P}_2$ nel complesso di risultati $R_1$ ottenibile da $\mathbf{M}_1$ con l'elaborazione a partire da $D_1$ retta da $\mathsf{P}_1$.

Dunque quali che siano $\mathsf{P}_1$ e $D_1$, i risultati $R_1$ della corrispondente elaborazione della $\mathbf{M}_1$ si possano ottenere facendo agire la $\mathbf{D}$ sugli $R_2$.

Sulla situazione descritta si dice che $\mathbf{C}$ **codifica** programmi e dati della $\mathbf{M}_1$ per la $\mathbf{M}_2$ e che $\mathbf{D}$ **decodifica** per la $\mathbf{M}_1$ i risultati ottenuti dalla $\mathbf{M}_2$.

La definizione precedente richiede una precisazione di carattere generale: come si è detto [e02], oltre alle elaborazioni che sicuramente giungono ad arrestarsi, si possono avere elaborazioni che si dimostrano in grado di proseguire illimitatamente ed elaborazioni che sono giunte a uno stadio nel quale non si sono arrestate e che esecutore e supervisore non sanno prevedere se possano mai arrestarsi.

Queste eventualità per ora non la prendiamo in considerazione e ci limitiamo a ricordare che quando si pone il problema della relazione di simulazione tra due MSP in linea di massima sui rendono necessarie analisi specifiche.

La definizione precedente comunque fa capire che quando si riesce a simulare una macchina $\mathbf{M}_1$ con una $\mathbf{M}_2$, la prima può essere sostituita dalla seconda e si può considerare la macchina simulante efficace almeno quanto la macchina simulata; in altre parole si può affermare che la portata della $\mathbf{M}_2$ è almeno estesa quanto quella della $\mathbf{M}_1$.

La relazione di simulazione tra le MSP comporta una sorta di ordinamento tra le MSP per portata ed efficacia che conviene precisare.

**B17 c.05** Per questo riteniamo conveniente denotare la relazione di simulazione con un simbolo specifico: qui per segnalare che la $\mathbf{M}_2$ può simulare la $\mathbf{M}_1$ scriviamo $\mathbf{M}_1 \preceq_{smul} \mathbf{M}_2$ . Osserviamo che il simbolo dissimmetrico introdotto è in accordo con altre notazioni di relazioni di ordinamento. []

La relazione di simulazione si può considerare riflessiva senza difficoltà: ogni macchina è in grado di simulare se stessa, e per questa come algoritmi di codifica e decodifica si possono assumere le semplici operazioni di copiatura di istruzioni, di dati e di risultati.

Quando si individuano una coppia di macchine $\langle \mathbf{M}_a, \mathbf{M}_b \rangle$ tali che $\mathbf{M}_a$ può simulare la $\mathbf{M}_b$ e viceversa si dice che si hanno due **macchine MSP equivalenti per portata**; più semplicemente si dice che si hanno due macchine intercambiabili.

Si osserva anche che se si hanno tre macchine $\mathbf{M}_1$, $\mathbf{M}_2$ e $\mathbf{M}_3$ e accade che $\mathbf{M}_1 \preceq_{smul} \mathbf{M}_2$ e che $\mathbf{M}_2 \preceq_{smul} \mathbf{M}_3$, allora $\mathbf{M}_3$ risulta in grado di simulare la $\mathbf{M}_1$, cioè $\mathbf{M}_1 \preceq_{smul} \mathbf{M}_3$.
Infine si osserva che se si hanno tre macchine $\mathbf{M}_1$, $\mathbf{M}_2$ e $\mathbf{M}_3$ e accade che $\mathbf{M}_1 \preceq_{smul} \mathbf{M}_2$ e che $\mathbf{M}_2 \preceq_{smul} \mathbf{M}_3$, allora $\mathbf{M}_3$ risulta in grado di simulare la $\mathbf{M}_1$, cioè $\mathbf{M}_1 \preceq_{smul} \mathbf{M}_3$.

Questa implicazione la si rende evidente ampliando la figura precedente per rappresentare anche il fatto che $\mathbf{M}_3$ simula la $\mathbf{M}_2$ servendosi di una nuova macchina di codifica $\mathbf{C}_\alpha$ di dati e programmi per $\mathbf{M}_3$ e di una nuova macchina $\mathbf{D}\alpha$ per la decodifica dei risultati della $\mathbf{M}_3$ per la $\mathbf{M}_2$. È allora evidente che $\mathbf{M}_3$ può simulare la $\mathbf{M}_1$ servendosi della codifica $\mathbf{C} \circ_{lr} \mathbf{C}_\alpha$ e della decodifica $\mathbf{D}\alpha \circ_{lr} \mathbf{D}$.

L'implicazione precedente si esprime anche affermando che la relazione di simulazione è una relazione transitiva.

Complessivamente si dice che la relazione di simulazione è una relazione d'ordine parziale tra macchine sequenziali programmabili.

A questo punto possiamo anticipare che la relazione di simulazione riveste grande interesse pratico quando si studiano macchine molto articolate, cioè macchine che si servono di molte sottomacchine alle quali demandano compiti specifici precedentemente conosciuti. attraverso i meccanismi della organizzazione modulare dei programmi.

Questo genere di situazione si esprime dicendo che è opportuno organizzare le macchine articolate con modalità modulare.

In effetti quando si progetta una nuova macchina articolata $\mathbf{M}$ risulta vantaggioso individuare macchine note $M_i$ in grado di effettuare manovre parziali previste tra i compiti della $\mathbf{M}$ in modo che queste $M_i$ possano essere riutilizzate come sue sottomacchine.
Adottando sistematicamente questo modo di organizzare le macchine si vengono ad individuare collezioni di macchine con comportamenti consolidati che possono essere utilizzate come sottomacchine di macchine con portate sempre più estese.
A questo punto risulta importante riuscire a individuare macchine note in grado di simulare sottomacchine per nuove macchine articolate: in tal modo si può far crescere il parco delle macchine disponibili servendosi di esperienze precedenti: è evidente che questa organizzazione modulare delle macchine risulta molto vantaggiosa per la crescita complessiva della strumentazione dell'intero *apparato*.

Conviene anticipare anche che si riesce a individuare una vasta gamma di tipi di MSP che possono presentare differenze costitutive anche molto rilevanti, ma che si dimostrano essere equivalenti in portata, cioè che sono in grado di effettuare lo stesso insieme (possiamo dire lo stesso insieme-E) di trasformazioni da dati in risultati.

Tra queste varianti vi sono le macchine di Turing, macchine sequenziali dotate di componenti operative estremamente semplici; questa semplicità costitutiva rende queste macchine formali particolarmente convenienti per dimostrare molte proprietà generali delle attività computazionali.

All'opposto, si possono individuare varianti con strutture molto elaborate le quali costituiscono generi formali dei moltissimi generi di computers o di reti di computers che sono concretamente disponibili o che stanno per essere costruite.

**B17 c.06** Ciascuno dei più rilevanti generi di MSP va visto anche in relazione alla gamma di problemi che consente di risolvere.
Le dimostrazioni delle loro equivalenze per portata consentono di individuare una raccolta di problemi affrontati e affrontabili mediante procedure, che costituiscono una collezione estremamente estesa e variegata; coppie di queste procedure equivalenti in portata spesso presentano differenze che le fanno preferire in molti diversi contesti applicativi.

Tutto questo contribuisce a consolidare l'idea della vastità della portata delle MSP. In particolare la loro collezione comprende tutte le elaborazioni effettivamente eseguite con gli strumenti informatici con i quali si è operato nei decenni recenti e con i quali si continua a operare massicciamente.

Segnaliamo anche che questo induce a congetturare che l'insieme delle elaborazioni eseguibili da MSP consente di coprire tutti i problemi che si possono affrontare con procedimenti operativi affidabili, riproducibili e analizzabili, ovvero tutti i problemi ben definiti (matematizzabili0.

È questa la cosiddetta `congettura di Church-Turing (we)` che verrà ripreso in C21.

A questo punto si può ritenere ragionevole giudicare soddisfacente lo studio delle elaborazioni eseguibili da MSP e dare fiducia a questi strumenti, mantenendo comunque la riserva di poter cambiare giudizio qualora si incontrassero elaborazioni non eseguibili da MSP, elaborazioni che costituirebbero controesempi per la congettura di Church-Turing.

Lo scenario prospettato e la libertà consentita ai tipi di dati trattabili e alle istruzioni di cui si possono dotare le MSP consente di individuare una ampia popolazione di elaborazioni formali in grado di dare risposta a problemi di concreto interesse.

Conseguentemente si riesce a prospettare una vasta gamma di iniziative articolate attraverso le quali gli organizzatori possono pianificare azioni incisive che contano sulla possibilità di servirsi delle prestazioni di molte MSP (anche di tipi molto diversi e anche operanti in collegamento diretto) per supportare attività tangibili, sviluppi conoscitivi e decisioni strategiche.

Questi collegamenti possono consistere in semplici forniture di informazioni che sono risultati per una procedura "precedente" e dati di ingresso per una procedura "successiva" che, in un panorama di strategie per la soluzione di problemi ben formulabili, va eseguita in uno stadio successivo.

**B17 c.07** È anche opportuno prospettare le attività di apposite MSP che siano in grado di produrre i testi dettagliati di programmi utilizzabili per risolvere problemi a partire da richieste applicative e avvalendosi di librerie di procedure collaudate per la risoluzione di problemi circoscritti.
Similmente si possono prospettare MSP che a partire da formulazioni di requisiti e avvalendosi di repertori di MSP siano in grado di generare descrizioni schematiche di nuovi modelli di MSP.

Un importante elemento a sostegno dell'importanza delle attività che si servono sistematicamente delle MSP è la componibilità delle loro prestazioni e la riutilizzabilità di risultati consolidati.

Va sottolineato che la componibilità e la riutilizzabilità delle macchine e delle procedure vanno concepite e attuate nell'ambito di scenari molto ampi, planetari.
L'odierna telematica consente di sviluppare azioni di monitoraggio, individuazione di decisioni reattive e anche di intervento diretto grazie alla efficienza delle interconnessioni avanzate (sottomarine,

satellitari, ...) tra apparecchiature cooperanti (sensori, computers, server farms, smart phones, robots, servomeccanismi, controllori di impianti, ...).

Complessivamente dobbiamo osservare che l'odierno scenario delle elaborazioni eseguibili affidabilmente risulta, da decenni, in continua espansione quantitativa e in continua evoluzione qualitativa.

Questo da un lato porta ad allargare le possibilità di risolvere problemi impegnativi e dall'altro fa nascere preoccupazioni sopra la eventualità di conseguenze negative imprevedibili, anche sui piani psicologico, sociale e politico.

Da tutto questo dovrebbe discendere la necessità di impegnarsi decisamente per sostenere una diffusa seria consapevolezza sulle potenzialità degli strumenti computazionali.

**B17 c.08** i proponiamo ora di convincere che abbia senso descrivere una elaborazione senza analizzare i dettagli dei processi materiali (meccanici, elettrici, elettronici, magneto-ottici, ...) che portano a ogni nuova configurazione ricorrendo alla convinzione che dimostreremo essere ampiamente condivisa che gran parte delle implementazioni delle macchine MSP siano in grado di eseguire correttamente e ripetutamente i loro compiti.

Dalla esigenza di riproducibilità delle elaborazioni affidabili segue l'esigenza di essere in grado di descrivere senza incertezze l'evoluzione di ogni elaborazione e attraverso un suo modello discreto.

Osserviamo che la richiesta di trattare solo elaborazioni realizzabili effettivamente e riproducibili perfettamente in pratica elaborazioni con possibilità di errori estremamente basse) ha condotto a richiedere che ogni elaborazione sia retta da un complesso di istruzioni espresse finitamente e interpretabili senza incertezze.

In precedenza, ragionevolmente, abbiamo iniziato descrivendo elaborazioni semplici che rispondono a esigenze semplicemente definibili e manovre eseguibili anche da operatori umani in grado di effettuare prestazioni chiaramente descrivibili e collaudate.
Tali descrizioni hanno buone possibilità di essere accettate come convincenti da una vasta platea di persone interessate.

Le prime elaborazioni sono state descritte seguendo uno schema discreto comprendente l'esecutore, gli strumenti dei quali si avvale, le informazioni che elabora e le operazioni che sa eseguire.
In particolare sono state introdotte istruzioni espresse finitamente prive di ambiguità considerandole le sole che si possono considerare effettivamente realizzabili e produttive.

**B17 c.09** Ogni procedura eseguibile affidabilmente è deducibile da un complesso di istruzioni che un opportuno esecutore umano o artificiale a partire da un determinato complesso di dati iniziali, è in grado di interpretare e attuare, servendosi l'umano delle proprie competenze, l'artificiale dei dispositivi dei quali è dotato nella prospettiva di risolvere l'istanza di problema che corrisponde ai dati iniziali.

Precisare quali procedure può effettuare un esecutore richiede di definire la classe dei complessi di dati iniziali che gli possono essere sottoposti perché esso possa effettuare una eleborazione con finalità costruttive, ovvero che dei complessi di dati in grado di avviare una sua evoluzione con conseguenze che rivestano interesse applicativo.

Tale classe di complessi di dati iniziali viene detta **dominio della procedura**; ciascuno di tali complessi si può ricondurre a un insieme di stringhe, cioè a un linguaggio formale, in particolare a una unica stringa opportunamente articolata. .

Anche ogni risultato di una procedura è riconducibile a un complesso di stringhe oppure a una sola stringa adeguatamente strutturata.

Quindi ogni procedura definisce un meccanismo che a ogni stringa del suo dominio (esprimente un complesso di dati iniziali) associa una stringa esprimente i risultati della elaborazione innescata dai dati iniziali.
Questa corrispondenza la chiamiamo **funzione dati-risultati**; essa si può considerare associata alla procedura, oppure alla MSP, cioè all'esecutore.

**B17 c.10** Cerchiamo di delineare quali caratteristiche possono presentare le funzioni dati-risultati delle MSP e delle procedure.

Si possono individuare, innanzi tutto, procedure che, quale che sia il complesso dei dati iniziali facente parte del loro dominio, portano avanti elaborazioni che giungono sicuramente a termine dopo un numero finito di passi. Queste situazioni risultano particolarmente ben definite e confortanti sul piano delle decisioni computazionali per fini applicativi.

Si incontrano invece anche delle MSP che, in dipendenza dei dati iniziali, presentano sia evoluzioni che dopo un numero finito di passi si arrestano e forniscono un risultato (più o meno soddisfacente), sia evoluzioni che dopo un numero finito di passi non hanno raggiunto una conclusione e lasciano aperte le due possibilità, il raggiungimento di una conclusione oppure la inconcludente prosecuzione illimitata delle elaborazioni.

Una situazione negativa ma chiara è quella nella quale si riesce a dimostrare che si potrebbe avere una evoluzione illimitata, senza possibilità di arresto.

Nel caso permanga l'incertezza i responsabili della esecuzione possono scegliere di proseguire o meno e per essere realisti devono tenere conto delle risorse a loro disposizione (disponibilità di tempo di calcolo, di energia elettrica, di spazio per memorie, di finanziamenti, di supporto tecnico, di urgenze, ...) .
Se scelgono di proseguire rischiano di consumare inutilmente delle risorse; se scelgono di arrestare l'esecuzione rinunciano a raggiungere un auspicato risultato.

Questa scelta in una fase di incertezza di una elaborazione può essere difficile e richiedere la valutazione di vari fattori che dipendono dalle circostanze attuali; possono essere determinanti il giudicare le risorse disponibili abbondanti oppure scarse e l'importanza attribuita al risultato atteso; possono influire i precedenti noti; può essere necessario ricorrere a statistica e simulazioni.

La distinzione tra le evoluzioni per le quali è garantito l'arresto con costi ragionevoli e quelle in cui si ha incertezza può essere molto rilevante.

Si utilizza il termine **procedura algoritmica** o **algoritmo** per designare ogni procedura che governa elaborazioni che per qualsiasi complesso di dati iniziali dichiarato lecito giungono a conclusione. Le rimanenti le chiameremo **procedure nonalgoritmiche**.

Si osserva che un algoritmo determina una funzione dati-risultati definita sopra tutto il suo dominio, mentre a una procedura non algoritmica risulta associata una funzione dati-risultati che si presenta definita solo per una parte del suo dominio,che potrebbe essere non facile delimitare.

Conveniamo di denotare con PRCDR l'insieme di tutte le procedure e con ALGRT l'insieme delle procedure algoritmiche e .

Evidentemente risulta importante riuscire a stabilire se una procedura sia algoritmica o meno.
Per ora ci limitiamo a segnalare che, data una procedura, lo stabilire se e per quali dati sia algoritmica in genere non è facile; anzi si trova che a certi livelli di generalità il problema è impossibile da risolvere [C21] .

**B17 c.11** In gran parte delle circostanze le procedure nonalgoritmiche sono considerate pericolose, da evitare, in quanto avviare la loro esecuzione comporta il rischio di consumare risorse senza alcuna garanzia di ottenere risultati.
In effetti per risolvere la maggior parte dei problemi di interesse pratico si ricorre a procedure con comportamenti algoritmici ben garantiti; questo accade soprattutto quando si affrontano problemi consolidati, e soprattutto problemi che richiedono di minimizzare dei rischi e assicurare tempi prevedibili.

Occorre segnalare anche che spesso si può ridurre una procedura a un più rassicurante algoritmo riducendo oculatamente l'insieme dei dati che è lecito sottoporlee, ossia il dominio della corrispondente funzione-dr.
Accade però che in molte circostanze si riesce a individuare abbastanza agevolmente una procedura che affronta un problema, mentre risulta più difficile individuare un sottoinsieme del suo dominio che garantisca la conclusione in tempi finiti delle elaborazioni innescate e che sia sufficientemente ampia da consentire di controllare una gamma soddisfacentemente estesa e utile di istanze del problema.

Volendo affrontare su un piano tendenzialmente generale i problemi delle elaborazioni automatiche si è indotti a prendere in considerazione l'intera classe delle procedure PRCDR.
In tal modo ci si pone in grado di fare riferimento a una gamma ampia di meccanismi elaborativi basandosi sulla definizione di procedura relativamente semplice, "programmatica e permissiva" data in precedenza, in accordo con il proposito di esaminare le elaborazioni della più ampia portata.
Per molti problemi se si volessero prendere in considerazione solo procedure con garantito comportamento algoritmico, si sarebbe costretti ad affrontare preliminarmente la questione tendenzialmente non facile della delimitazione del suo dominio.

Viceversa va segnalato che, nell'ambito delle considerazioni computazionali generali, è risultato spesso utile servirsi di proprietà delle procedure nonalgoritmiche per conoscere meglio meccanismi che possano portare a procedure algoritmiche auspicabilmente applicabili a insiemi di dati interessanti.

Va infine segnalato che nella pratica computazionale non di rado accade di servirsi euristicamente di procedure che hanno la possibilità di evolversi illimitatamente ma possono essere messe in campo con relativa facilità nella speranza di ottenere in tempi finiti risultati e conoscenze parziali che possono portare suggerimenti per successive più consapevoli azioni nella direzione delle possibili soluzioni finite del problema in esame.
In qualche caso si riesce ad ottenere una bozza di classificazione delle situazioni che consentano di individuare algoritmi che possano essere concretamente utili per gamme di situazioni circoscritte, ma comunque interessanti.

## B17 d. macchine sequenziali programmabili [2]

**B17 d.01** Procediamo a sviluppare considerazioni generali sulle procedure facendo riferimento quasi esclusivamente alle macchine sequenziali programmabili.
Questi esecutori vengono individuati costruttivamente ma solo per grandi linee e con argomentazioni intuitive sono stati riconosciuti in grado di effettuare le elaborazioni affidabili che in precedenza sono state trovate eseguibili dalle MSM, le macchine sequenziali multinastro.

Nell'insieme delle macchine sequenziali programmabili, che denoteremo con MSP, si possono porre, non solo tutte le MSM, ma anche tutte le macchine che si dimostreranno essere equivalenti a esse, a cominciare dai comuni tipi di computers dotati di un linguaggio di programmazione procedurale di livello medio-basso, in particolare dotati del linguaggio c++ introdotto in B70 e B71.
A questo per ora vago insieme MSP si possono ancheassimilare gli esecutori umani che, come si è detto, si possono considerare in grado di risolvere affidabilmente gli stessi problemi risolvibili dalle MSP artificiali (quando si trascurino le questioni di efficienza).

Dell'insieme MSP esamineremo alcuni sottoinsiemi definiti con richieste sulle loro prestazioni, ancora senza definizioni precise sulla loro costituzione.

Chiaramente anche questo modo di procedere va considerato provvisorio e fiduciario: più avanti dovremo definire con accettabile precisione alcuni tipi di MSP che possono essere utilizzati per risolvere problemi specifici e conseguentemente per ampliare l'*apparato* AMI. Essenziali saranno le dimostrazioni della equivalenza delle rispettive portate.

**B17 d.02** Riprendiamo le caratteristiche salienti delle elaborazioni eseguibili affidabilmente da meccanismi artificiali muovendoci sul piano delle esigenze e delle possibilità generali, con l'obiettivo di individuare generi equivalenti di esecutori artificiali.
Non entreremo in molti dettagli, ma presenteremo risultati che potranno essere pienamente giustificati solo più avanti, [in particolare v. C21].

Preliminarmente ricordiamo che tutte le elaborazioni di informazioni eseguibili da esecutori umani o artificiali che abbiamo viste finora sono riconducibili a composizioni e trasformazioni di stringhe.
In effetti abbiamo già osservato che a manovre su stringhe si possono ricondurre anche tutte le elaborazioni effettuate affidabilmente che incontreremo in seguito e quindi tutte le costruzioni che contribuiscono allo sviluppo delle attività computazionali e della matematica e tutte le elaborazioni con lo scopo di scegliere tra due possibilità.

In linea di principio si può pensare che le questioni fondamentali sulle elaborazioni si possano affrontare considerando soltanto costruzioni formali che effettuano composizioni e trasformazioni di stringhe.
Nella pratica invece in genere si trova più conveniente e significativo fare riferimento a entità come numeri, vettori, matrici, digrafi, relazioni, funzioni, trasformazioni, strutture algebriche, ..., spesso senza soffermarsisorvolando sulle stringhe che si devono usare per rappresentare queste entità nei contesti applicativi.

Ricordiamo anche che tutte le elaborazioni, quale che sia la natura dell'esecutore che le effettua, richiedono risorse che si possono attribuire a tre generi: supporti per la registrazione delle informazioni, tempo di elaborazione e complessi di istruzioni operative.

**B17 d.03** Le macchine sequenziali programmabili sono introdotte come le apparecchiature che riteniamo più adatte per definire e giustificare le elaborazioni che in seguito ci interesseranno maggiormente, seguendo il piano di approccio che segue.

Fondamentalmente chiediamo che le MSP si basino su descrizioni di meccanismi artificiali tali che ci si possa aspettare ampio consenso sulla effettiva possibilità di costruirli e farli funzionare correttamente, ossia in modo da realizzare gli obiettivi che si sono proposti.
Un valido sostegno a questo consenso lo danno le caratteristiche di portata, di versatilità e di attendibilità degli strumenti che le tecnologie informatiche continuano a rendere disponibili e le caratteristiche dei risultati che essi continuano a ottenere.

Queste macchine saranno definite senza entrare in molte caratteristiche costruttive, ma prospettando loro esemplari che possano essere ampiamente accettati come apparecchiature i cui dettagli possono essere precisati senza difficoltà dagli esperti del settore r che ragionevolmente sono in grado di eseguire una estesa gamma di elaborazioni di concreto interesse.

Questo atteggiamento di genericità consente di lasciare molto aperta la gamma dei generi delle MSP; come vedremo, questa comprenderà macchine con componenti estremamente semplici come le macchine di Turing e sistemi estremamente complessi come gli odierni computers con la possibilità di essere interconnessi in sistemi estremamente articolati.

A ciascuna MSP **M**, che qui pensiamo dotata di un programma, possono essere affidati diversi sistemi di dati iniziali e a partire da ciascun di essi essa è in grado di eseguire una elaborazione.
Le elaborazioni effettuabili dalla **M** costituiscono la cosiddetta **famiglia delle elaborazioni eseguibili** dalla **M**, famiglia i cui esemplari sono caratterizzati dai singoli sistemi di dati iniziali D.

Ad ogni MSP **M** è associata anche la funzione che fa corrispondere a ogmi scelta di dati di ingresso e i risultati della corrispondente elaborazione.
Questa funzione viene chiamata **trasformazione dati-risultati della macchina M**.

Due diverse MSP si dicono equivalenti sse eseguono con gli stessi risultati la stessa famiglia di elaborazioni, ossia sse realizzano la stessa funzione dati-risultati.

Un aspetto cruciale dello studio delle prestazioni delle macchine sequenziali e in particolare delle MSP, sta nel fatto che ogni famiglia di elaborazioni può essere eseguita da una grandissima varietà di MSP equivalenti.

Inoltre si giunge a verificare che le MSP riescono a eseguire una gamma molto vasta di famiglie di elaborazioni specifiche, ossia che le MSP posseggono una gamma molto variegata di prestazioni. Ha quindi grande importanza il complesso di quelle che chiamiamo **elaborazioni eseguibili dalla macchina** MSP.

Si può osservare che il precedente tipo di approccio rinuncia ad esaminare l'esecuzione di elaborazioni da parte di esecutori umani: in effetti un tale esame dovrebbe ricorrere a discipline complesse e non consolidate ma in fase di crescita come sociologia, psicologia e neuroscienze, argomenti al di l'a delle prospettive di questa *esposizione*.
Ci si limita invece ad affermare che tutte le elaborazioni eseguibili meccanicamente possono essere eseguite anche dagli operatori umani, senza dimenticare che questi potrebbero richiedere risorse di tempo e di attenzione ai particolari enormemente maggiori.
Questa affermazione si può basare sulla constatazione che ogni operatore umano sufficientemente affidabile è in grado di effettuare le operazioni basilari sulle stringhe alle quali è riconducibile ogni elaborazione automatica.

**B17 d.04** Alle MSP si chiede di operare per passi successivi in ciascuno dei quali effettua una sola manovra; questa caratteristica delle MSP determina quella che viene detta **evoluzione sequenziale deterministica** delle MSP.
Una MSP deve presentare varie componenti:

- dispositivi di memoria riconducibili a nastri indirizzabili che consentono la lettura e la registrazione dei dati attraverso testine di lettura/scrittura;
- dispositivi per la trasformazione di piccoli raggruppamenti di dati che chiameremo **circuiti operativi**;
- un apposito dispositivo di memoria che chiamiamo **nastro delle istruzioni** nel quale prima di una famiglia di elaborazioni viene registrata una sequenza di stringhe dette **istruzioni** operative; queste esprimono operazioni sui dati che i circuiti sono in grado di eseguire;
- un **controllo**, dispositivo definito come il supervisore per le attività della macchina.

Per i dati trattabili chiediamo solo che possano essere rappresentati con stringhe finite sopra un alfabeto finito riconoscibile da tutti gli esecutori di una famiglia cooperante; in particolare in ogni caso possono essere rappresentati mediante sequenze binarie.
Dal punto di vista delle applicazioni chiediamo che siamo trattabili i numeri interi naturali, i valori di verità vero e falso (rappresentabili, risp., con 1 e 0), le sequenze e le matrici di bits, di caratteri basilari (per esempio ASCII).
Inoltre chiediamo sia possibile operare con strutture di dati (discrete) esprimibili mediante ragionevoli combinazioni di interi naturali e di stringhe di caratteri.
Si chiede anche di poter disporre di famiglie di MSP cooperanti che possano condividere raggruppamenti di dati trattabili e tipi di dati.

Per nastro indirizzabile intendiamo, come al solito, un dispositivo di memoria costituito da celle in grado di registrare un dato di qualcuno dei tipi trattabili. Le celle dei nastri più semplici, sono disposte in sequenze e possono essere scorse passo a passo nelle due direzioni; esse inoltre possono essere indirizzate direttamente servendosi per ciascuna cella di un identificatore che la caratterizza, informazione riconducibile a un numero progressivo che individua la sua collocazione in tabelle organizzate nelle memoria.
Per rispetto della realtà si chiede che tutti i nastri siano finiti; si possono avere nastri poco estesi o grandi quantità di celle singolarmente indirizzabili. Si avanza anche la richiesta tendenziale che alcuni nastri possano venire estesi nel corso delle elaborazioni tanto da soddisfare nuove esigenze che vengano a manifestarsi nel corso delle esecuzioni.

Le istruzioni devono potersi organizzare in sequenze che siano efficaci, cioè che siano in grado di governare evoluzioni della MSP che abbiano senso applicativo, ossia che riescano a risolvere problemi rilevanti e incisivi.
Si chiede che le sequenze di istruzioni di ogni genere di MSP possano essere presentate con una loro adeguata struttura organizzativa, ovvero si chiede che possano considerarsi le stringhe di un linguaggio formale specifico che soddisfi esigenze di varia natura (lessicale, sintattica e semantica); inoltre questo linguaggio deve essere soddisfacente sul piano pragmatico.

Una sequenza di istruzioni orientata alla soluzione di un problema che desta interesse viene detta **programma per la MSP** e si dice che costituisce la **implementazione di una procedura**.

Abbiamo dunque presentate molte caratteristiche delle macchine solo come auspici. Per convincere che queste caratteristiche possano essere realizzate bisogna ricorrere alla segnalazione di una cospicua gamma di apparecchiature realizzate e trovate in grado di ottenere risultati soddisfacenti.

**B17 d.05** Il controllo, o unità di controllo, di una MSP a ogni passo evolutivo in genere svolge in successione le tre attività che seguono:

- riconosce l'istruzione attuale (il suo tipo, le informazioni sulle quali deve operare, e le informazioni che deve produrre) e la interpreta per demandare la sua esecuzione agli opportuni circuiti operativi,

in genere dopo aver trasferito loro dati specifici ottenuti da unità di memoria (registri, celle di memoria ocaselle di nastri).
- attende che i circuiti restituiscano le informazioni risultato e cura che esse vengano registrate in determinate unità di memoria;
- stabilisce quale altra istruzione dovrà essere eseguita nel passo successivo.

Il controllo inoltre gestisce (cioè consulta e modifica) una particolare informazione variabile che può assumere un insieme finito e predeterminato di valori che possono essere rappresentati da interi naturali e che contribuisce a determinare il comportamento del controllo della MSP; i suddetti valori si dicono contraddistinguere gli **stati interni della macchina** e si dice che il controllo a ogni passo si viene a trovare in uno stato interno. A questo scopo il controllo in genere si serve anche di propri registri privilegiati.

**B17 d.06** I successivi passi di una elaborazione conviene che siano riferiti alle successive componenti di una sequenza di tempi; questa la rappresentiamo con una sequenza iniziale di interi naturali, numeri che consideriamo come i valori di un parametro che diciamo **tempo discreto** e che specifichiamo servendoci di una apposita variabile "$\mathfrak{t}$": i successivi passi sono dunque contraddistinti dai successivi valori $\mathfrak{t} = 0$, $\mathfrak{t} = 1$, $\mathfrak{t} = 2$, ... .

Ogni passo evolutivo vede una transizione della MSP da una precisa configurazione attuale a una configurazione successiva come conseguenza determinata dell'esecuzione dell'istruzione attuale.
Una evoluzione di una MSP, ovvero una elaborazione da essa eseguita, consiste nella sequenza delle configurazioni che essa via via assume.
Nell'istante iniziale, caratterizzato dallo stato temporale $\mathfrak{t} = 0$, il meccanismo si trova nella sua **configurazione iniziale**: questa corrisponde a una sequenza formata dalle seguenti componenti:

- lo stato interno definito **stato iniziale**;
- le registrazioni sui nastri di input dei dati iniziali; può essere utile descrivere tali dati come forniti da un committente dell'elaborazione che sta per iniziare il quale si aspetta che la MSP fornisca un risultato che dia una soluzione a un suo problema o che almeno porti qualche chiarimento a tale problema;
- le posizioni delle testine di lettura/scrittura sopra le prime caselle dei nastri dedicati alla registrazione dei dati iniziali e/o disponibili per i primi risultati dell'elaborazione.

**B17 d.07** Torniamo all'esecuzione di una istruzione **I**. Il controllo, dopo averla riconosciuta si prepara a eseguirla: inizialmente si fa fornire da alcune testine di lettura come dati richiesti dalla stessa **I** le informazioni registrate a partire dalle posizioni sulle quali le testine sono collocate.
Successivamente l'unità di controllo, a partire da questi dati e da altri eventuali dati a sua disposizione (contenuti nella **I**, nello stato interno o in propri registri), decide

- se chiedere ai circuiti operativi di ricavare nuovi dati con i quali modificare i contenuti delle posizioni puntate dalle testine di alcuni nastri (tipicamente di nastri di emissione e ausiliari, talvolta anche di nastri di lettura) ed eventualmente modificare stato interno e propri registri;
- se muovere le testine di alcuni nastri in una delle direzioni percorribili;
- se passare all'istruzione che segue quella eseguita oppure ad una istruzione diversa determinata dai dati a sua disposizione.

Assumiamo come requisito essenziale di ogni MSP e di ogni suo programma che le sue istruzioni e i dati a sua disposizione siano interpretabili dal controllo senza incertezze e le sue istruzioni siano eseguibili univocamente.

L'elaborazione procede secondo le modalità ora descritte in tutti i passi che seguono l'iniziale e determina le configurazioni successive riferibili, come si è detto, ai valori della variabile tempo discreto $\mathtt{t} = 1,\ \mathtt{t} = 2,\ \mathtt{t} = 3,\ ....$

Con il procedere dei passi evolutivi il controllo potrebbe incontrare una esplicita **istruzione di arresto** dell'eleborazione, oppure trovarsi in una configurazione che vede esauriti i dati che dovrebbero alimentare nuovi passi evolutivi e di conseguenza decidere di porre fine alla sua evoluzione; in tutti questi casi l'esecuzione è giunta alla sua conclusione, ovvero al proprio arresto.

**B17 d.08** Torniamo sulle differenze tra le macchine sequenziali programmabili (MSP) e le macchine sequenziali multinastro (MSM).

Come le MSM, le MSP sono caratterizzate da un controllo, da un sistema di istruzioni, da registri interni e dalla possibilità di servirsi di nastri di memoria.

Le MSP possono disporre di una maggiore gamma di dispositivi di memoria che possono registrare oggetti che descriviamo come diversi dalle stringhe e di conseguenza di un repertorio di istruzioni molto più eterogeneo.
Infatti si consente la possibilità di gestire in modo diretto (e quindi con maggiore facilità) una gamma più estesa di tipi di dati in quanto si dà la possibilità di definire istruzioni che permettono di agire su oggetti più articolati delle mere stringhe che li possono comunque rappresentare.
È quindi evidente che ogni MSM può essere facilmente simulata da una MSP e quindi che la portata delle MSP è superiore o uguale a quella delle MSM.

Dunque La differenza maggiore riguarda i sistemi di istruzioni.
Le MSM dispongono di un insieme di istruzioni e a ogni passo il controllo deve scorrere tale insieme alla ricerca dell'istruzione giusta, che deve essere univocamente riconoscibile.
I sistemi di istruzioni per le MSP presentano liste di istruzioni e a ogni passo per default si passa alla successiva o si va a una istruzione diversa dalla successiva e fornita dalla esecuzione dell'istruzione corrente.

## B17 e. caratteristiche delle elaborazioni procedurali [2]

**B17 e.01** Le elaborazioni automatiche nella societàodierna rivestono importanza primaria, in quanto consentono di affrontare con buone possibilità di successo una quantità di problemi che senza ricorrere ad automatismi non potremmo sensatamente sperare di risolvere.

Sulle elaborazioni automatiche si reggono dispositivi automatici utilizzati in quasi tutte le attività che non riguardino esclusivamente i singoli individui o i piccoli gruppi di persone: si costituiscono sistemi per le comunità (amministrativi, commerciali, produttivi, dei trasporti, della comunicazione, ...), si organizzano efficientemente conoscenze complesse (attraverso archivi digitali, strumenti di searching e reti), si sostengono tutti i tipi di ricerche e di sviluppi innovativi.

Procediamo ora a esaminare le elaborazioni eseguite da un esecutore che effettua trasformazioni di stringhe esprimenti i dati di istanze di un problema al fine di ottenere stringhe che costituiscono i risultati che risolvono la detta istanza.

A queste elaborazioni si avanzano come richieste generali, di principio, la riproducibilità e la controllabilità delle manovre grazie alle quali si sviluppano.

In relazione a tali richieste si chiede che ogni elaborazione effettuata da un esecutore possa essere analizzata da un'altro agente (umano o artificiale) che in questo scenario assume un ruolo di **supervisore delle elaborazioni**.

Un supervisore di una elaborazione, in linea di principio, deve essere in grado di conoscere i dettagli delle successive configurazioni attraverso le quali l'elaborazione si sviluppa e di giustificare ogni passaggio da una configurazione alla successiva facendo riferimento a una istruzione (la corrente), allo stato interno dell'esecutore e alle informazioni che l'istruzione stessa rende disponibili.

Per un quadro più comprensivo può essere utile prospettare che un supervisore abbia anche il compito di riferire a un soggetto che chiamiamo **committente della elaborazione** alcuni risultati intermedi che risultano significativi rispetto alle aspettative del committente stesso.

**B17 e.02** Diamo una prima segnalazione delle valutazioni sul procedere di una elaborazione che un supervisore può ritenere opportuno segnalare al committente.

Una elaborazione viene detta **elaborazione finitamente pertinente** se giunge ad arrestarsi in una certa configurazione finale nella quale si individuano stringhe risultato che forniscono una risposta pertinente alla richiesta che ha originato l'elaborazione stessa; una tale risposta dal committente potrebbe ricevere un giudizio che si colloca tra l'essere pienamente soddisfacente e l'essere del tutto deludente.

Dal punto di vista dell'esecutore l'arresto va visto come la conclusione di una sequenza di trasformazioni delle stringhe che rappresentano gli oggetti che caratterizzano l'istanza del problema con la individuazione di un complesso di stringhe che esprimono oggetti attesi o paventati dal committente.

Nei casi più semplici si ha una stringa che determina un numero, una grandezza fisica, un nome, una risposta positiva o negativa a una domanda a due vie o una valutazione che il committente potrà valutare in relazione a scelte che egli e l'organismo del quale fa parte intendono decidere.

In altri casi si ottengono più stringhe risultato che esprimono entità matematiche o strutture informative che possono anche essere molto elaborate (si pensi per esempio al testo di programma in un linguaggio vicino al linguaggio dei circuiti del computer che viene generato da un programma `compilatore (wi)`).

Alternativamente può accadere che una elaborazione si concluda con l'arresto delle operazioni in una **configurazione anomala** che riguarda una situazione che il committente e l'esecutore non avevano prevista, oppure con l'arresto in una configurazione che non consente di individuare stringhe risultato che si possano interpretare come una risposta pertinente alla richiesta che ha dato origine alla elaborazione. Una tale elaborazione si può chiamare elaborazione con conclusione imprevista.

Può anche accadere che dopo un consistente numero di passi l'elaborazione non sia giunta a una conclusione.
Una tale situazione va riferita a un ben determinato stadio $S_1$ della elaborazione e la sua segnalazione al committente può comportare valutazioni diverse.
Queste valutazioni sono da porre in relazione alle risorse che la elaborazione ha richieste (in particolare alle risorse di tempo) e influiscono principalmente sulla opportunità di proseguire l'eleborazione.

Anche per decidere questa prosecuzione può rendersi necessario che il committente svolga analisi anche laboriose che possono toccare questioni che non si limitano alla sola istanza del problema da risolvere.

## B17 f. macchine equivalenti

**B17 f.01** Abbiamo visto come talune trasformazioni di stringhe si possono ottenere con manovre diverse da una stessa macchina sequenziale multinastro.
Questi non sono che esempi limitati di una situazione molto generale: in effetti per tutti i problemi risolvibili affidabilmente si conoscono o si potrebbero individuare molti procedimenti risolutivi, spesso molto diversi.

Questa varietà di procedimenti scende anche dal fatto che ogni soluzione di un problema si può ottenere con macchine costruite diversamente.
Questo fatto verrà ripreso più circostanziatamente più avanti quando cercheremo di individuare la totalità delle macchine in grado di ottenere delle SAP, delle soluzioni affidabili dei problemi.

Per ora vogliamo solo accennare a grandi linee come si possa allargare la gamma delle macchine equivalenti.

**B17 f.02** Diciamo che due macchine sequenziali (multinastro o programmabili) $\mathsf{M}_1$ ed $\mathsf{M}_2$ sono **macchine funzionalmente equivalenti** sse per tutti i sistemi di dati che individuano correttamente istanze di un problema che entrambe possono risolvere esse forniscono le le stesse soluzioni.

Per accertare la equivalenza funzionale, in breve la **equivalenza-f**, di due macchine $\mathsf{M}_1$ ed $\mathsf{M}_2$ con strutture diverse si ricorre alla nozione di **simulazione**.

Si dice che un programma $P_1$ per la macchina $\mathsf{M}_1$ viene simulato da un programma $P_2$ per la $\mathsf{M}_2$ sse ogni elaborazione governata da $P_1$ conduce a risultati che possono essere ottenuti da una elaborazione governata da $P_2$.
Si dice che $\mathsf{M}_2$ può simulare la $\mathsf{M}_1$ sse si individua una trasformazione di ciascuna delle istruzioni per la $\mathsf{M}_1$ in una o più istruzioni per la $\mathsf{M}_2$ tale da trasformare ogni programma $P_1$ per la $\mathsf{M}_1$ in un programma $P_2$ per la $\mathsf{M}_2$ in grado di simulare $P_1$.

Quindi due macchine $\mathsf{M}_1$ e $\mathsf{M}_2$ sono funzionalmente equivalenti sse ciascuna delle due è in grado di simulare l'altra.
Per affermare concisamente questa equivalenza scriviamo

$$\mathsf{M}_1 \sim_{smul} \mathsf{M}_2 \ .$$

**B17 f.03** Quello che si riesce di fare è procedere a individuare vari tipi di macchine funzionalmente equivalenti alle macchine MSM. Queste macchine possono essere sia di costruzione sensibilmente più semplice delle MSM, sia costruite come si sono tratteggiate le MSP nelle pagine precedenti.

Si tratta quindi di procedere a individuare da un lato macchine dotate di dispositivi più semplici e dall'altro macchine dotate di dispositivi molto più elaborati, tutte funzionalmente equivalenti.

Le prime, in particolare le macchine di Turing, hanno il vantaggio di essere facilmente rappresentate da numeri naturali e quindi consentono di tenere sotto controllo in modoelativamente facile la loro portata sul piano dei concetti generali.
Le seconde comprendono gli odierni computers dotati di una vasta gamma di linguaggi di programmazione e riescono a essere molto più efficienti, versatili e adattabili.

Va tenuto presente che la grande varietà dei tipi di macchine funzionalmente equivalenti si ottiene prescindendo drasticamente da questioni pratiche e in particolare da considerazioni di efficienza.

Sul piano dei concetti generali l'ampliamento delle macchine funzionalmente equivalenti conduce a una visione straordinariamente unitaria delle possibilità di trovare concretamente soluzioni affidabili di problemi impegnativi.

Testo fruibile in https://www.mi.imati.cnr.it/alberto/ e https://arm.mi.imati.cnr.it/Matexp/matexp_main.php