Cosa vuol dire computare

Computare significa ridurre un'informazione da una forma implicita ad una forma esplicita in modo effettivo
→ Rielaborazione informazioni → Esplicitarla

Definizione di modello computazionale

Lo possiamo definire come un modello o formalismo matematico, in cui esso viene rappresentato in modo astratto e rappresentiamo un modo di definire un input, un output, un passo e come organizzare i passi degli elementi di computazione durante un processo effettivo (algoritmo) che permette di rendere esplicito ciò che era implicito in partenza

Diversi modelli computazionali

- Funzioni ricorsive
- Lambda - Calcolo
- Macchine di Turing (automi con la massima potenza computazionale)

Teoria dei linguaggi formali

Gli ASFD riconoscono le stringhe appartenenti a linguaggi regolari

- NON possono memorizzare informazioni a parte la stringa di input

Analisi di Alan Turing ⇒ concetto di procedura effettiva

Turing provò a formalizzare la classe di tutte le procedure effettive (1936) e questo rappresenta la 1° analisi che descrive come ha luogo la computazione

---

Le configurazioni

Prendiamo 2 configurazioni C_i e C_j e indichiamo con $C_i \vdash_{A} C_j$ la correlazione di S tramite una relazione di transizione, vale a dire C_j deriva da C_i per effetto dell'applicazione della funzione di transizione di A. ovviamente se l'automa a cui ci si riferisce è ben definito possiamo scrivere $C_i \vdash C_j$

Se andiamo a considerare una sequenza di configurazioni $C_0, \dots C_n$ che ha lunghezza finita ⇒ $C_n \vdash_{A} C$ → configurazione finale

↓

terminare in uno stato di accettazione o rifiuto

Quando la configurazione ha lunghezza finita ed è massimale essa termina

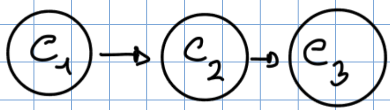
tipi di configurazione

Accettazione: situazione in cui l'automa, letta tutta la stringa di input si trova in uno stato q finale ($q \in F$)

Rifiuto / non accettazione: situazione in cui l'automa non ha letto tutta la stringa di input oppure se l'ha letta, non si trova in uno stato finale

2.5.2 automi deterministici e non

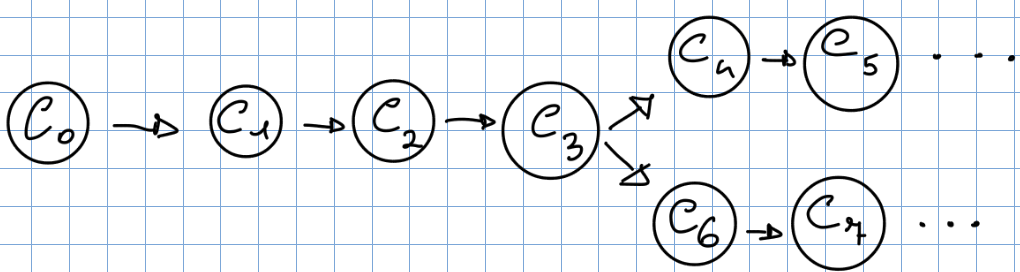
Definizione: Un automa è detto deterministico se ogni stringa di input associa una computazione e quindi una singola sequenza di configurazioni. Possiamo dire che un automa deterministico, data una stringa in input può eseguire una sola computazione: se la computazione termina in una configurazione di accettazione, allora la stringa viene accettata



Computazione di un automa deterministico

Definizione: un automa è non deterministico se esso associa ad ogni stringa di input un numero qualunque ($n > 1$) di computazioni. osserviamo che l'automa deterministico è un caso particolare del non det per $n = 1$ non det $\Leftrightarrow$ det

Indicheremo con grado di non determinismo di un automa il numero massimo di configurazioni che la funzione di transizione associa ad una configurazione

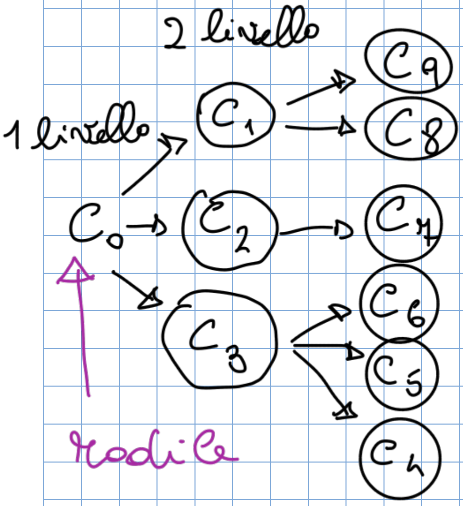


Abbiamo diverse continuazioni di una stessa computazione di un automa non deterministico

Osservazione

Asimmetria tra accettazione e rifiuto di una stringa tra det e non det. Infatti nel caso della non det la stringa viene accettata se una qualunque delle computazioni definite è di accettazione, mentre non lo è se tutte le possibili computazioni che terminano non sono di accettazione

Per un automa non deterministico, consideriamo come se l'automa esegue una sola computazione non deterministica per la quale, ad ogni passo, assume non una sola configurazione ma un insieme di configurazioni, transitando ad ogni passo, non da una conf. e un'altra conf ma da un'insieme di config ad un insieme di conf



Albero di computazione

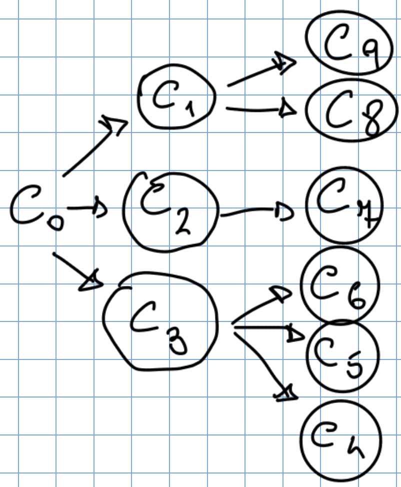
C_0, C_1, C_9

C_0, C_1, C_8

Con un albero computazionale definisco "Singole" computazioni aventi come prefissi le sequenze di configurazioni C_0, C_3, C_4 ; C_0, C_1, C_8 ; C_0, C_3, C_6 corrispondenti ai cammini con origine nella radice dell'albero

Il concetto di non determinismo non deve essere confuso con la definizione reale, perché questo concetto è sostanzialmente un artificio matematico e ci

Consente di rappresentare un calcolo, invece di una traiettoria definita in uno spazio di stati come un albero di traiettorie. Ciò che ci interessa è poter definire un concetto di accettazione legato al fatto che almeno uno dei rami dell'albero conduce ad uno stato finale



Espressioni regolari $\rightarrow$ descrivono tutti i linguaggi appartenenti a una classe

Dato un'alfabeto Σ e dato l'insieme di simboli
$\{+, *, (,), \cdot, \emptyset\}$

Si definisce espressione regolare sull'alfabeto Σ una stringa

$$r \in \left(\Sigma \cup \{+, *, (,), \cdot, \emptyset\}\right)^+$$

tale che valga una delle seguenti condizioni:

1. $r = \emptyset$
2. $r \in \Sigma$
3. $r = (s + t)$, oppure $r = (s \cdot t)$ oppure $r = s^*$

dove s e t sono espressioni regolari sull'alfabeto Σ

Corrispondenze tra le espressioni regolari e i linguaggi:

ESP. REG	LINGUAGGIO
$\emptyset$	Λ
a	$\{a\}$
$(S+T)$	$L(S) \cup L(T)$
$(S \cdot T)$	$L(S) \circ L(T)$
S^*	$(L(S))^*$

con $L(r)$ generico denoto il linguaggio rappresentato dall'espressione regolare r

→ Se s e t sono due espressioni regolari posso scrivere (st) anziché $(s \cdot t)$

→ Diamo precedenza al simbolo $*$ sul simbolo $\cdot$, $\cdot$ ha precedenza su $+$ e possiamo tenere conto delle proprietà associative di tali applicazioni

ES:

l'espressione regolare $(a+(b\cdot(c\cdot d)))$ definita sull'alfabeto $\Sigma=\{a,b,c,d\}$ può essere riscritta

$(a+(b(cd))) = a+bcd$

anche per espressioni regolari, come per i simboli caratteri o stringhe possiamo introdurre le potenze scrivendo $(r)^3 \to \underbrace{rrr}$ espressione regolare

la chiusura non riflessiva r^+ per indicare $r(r)^*$

Osservazione

Per rappresentare la stringa vuota può essere utile a volte usare il simbolo ε nelle espressioni regolari, con lo scopo di indicare il linguaggio $\{\varepsilon\}$

$\Lambda = \{\varepsilon\}$