Calcolabilità

DEF. **Insieme finito.** Un insieme A è **finito** se esiste f: A {1,..,n} con n N biunivoca

DEF. **Insiemi equipotenti.** Due insiemi sono **equipotenti** se esiste f: A B biunivoca

DEF. **Insieme infinito contabile** .Un insieme è **infinito contabile** se è equipotente ad N

DEF. **Insieme contabile.** Un insieme è **contabile** se è finito ed è equipotente ad N

DEF. **Insieme enumerabile** Un insieme I N è **enumerabile** se g: N I suriettiva

DEF. **Rappresentazione.**Dato F F_{calc} si dice che I N è una **rappresentazione** di F

se i I : i F e f F i I t.c. F = i

DEF. **Funzione caratteristica.** La funzione caratteristica di un insieme I è definita in modo tale che

$$f(x) = \begin{matrix} 1\ se\ x \in I \\ 0\ se\ x \notin I \end{matrix}$$

DEF. **Funzione semicaratteristica.** La funzione caratteristica di un insieme I è definita in modo tale che $f(x) = \begin{matrix} 1\ se\ x \in I \\ \uparrow\ se\ x \notin I \end{matrix}$

DEF. **Insieme Ricorsivo.** Un insieme $I \subseteq N$ è detto ricorsivo se la sua funzione caratteristica è calcolabile totale.

NB. Gli insiemi ricorsivi sono chiusi per unione, intersezione e complementazione. Se un insieme è finito allora è ricorsivo.

DEF. **Insieme Ricorsivamente enumerabile.** Un insieme I è ricorsivamente enumerabile se $I = \emptyset$ oppure $\exists f / I = cod(f)$ con f calcolabile totale.

DEF. **Insieme Semidecidibile.** Un insieme I è semidecidibile se la sua funzione semicaratteristica è calcolabile.

DEF. **Insieme Semidecidibile (alternativa).** Un insieme I è semidecidibile se $\exists f / dom(f) = I$ con f calcolabile parziale.

NB: I è ricorsivamente enumerabile se e solo se I è semidecidibile.
NB: Se I è ricorsivo allora I è anche ricorsivamente enumerabile.

Teorema di Post : sia $I \subseteq \mathbb{N}\ R.E.\ e\ \overline{I}\ R.E. \Rightarrow I\ ricorsivo$

DIM : $se\ I = \emptyset \Rightarrow I$ $ricorsivo\ per\ definizione$

$se\ I \neq \emptyset \wedge I \neq \mathbb{N} \Rightarrow \exists g, h\ \ calcolabili\ \ totale\ tali\ \ che\ cod(f) = I \wedge cod(h) = \overline{I}$

$$f(x) = \begin{cases} 1 \Leftrightarrow x \in I \Leftrightarrow x \in cod(g) \Leftrightarrow \exists y : g(y) = x \\ 0 \Leftrightarrow x \in \overline{I} \Leftrightarrow x \in cod(h) \Leftrightarrow \exists y : h(y) = x \end{cases}$$

```
f(x):= begin
        y:=0;
        while ( g(x)≠x∧h(y)≠x )do
                y:=y+1;
        if (f(y)=x) then
                f(x):=1;
        else
        f(x):=0;
end.
```

Questo ciclo while prima o poi termina perchè il dominio di f è $I \cup \overline{I} \Rightarrow \mathbb{N}$, quindi f è calcolabile.

Linguaggi Formali e teoria degli automi

Una **grammatica** a struttura di frase è una quadrupla G= (V,T,S,P) con: V=insieme finito non vuoto (vocabolario); T V insieme simboli terminali (alfabeto); V\T = insieme simboli non terminali; S simbolo iniziale non terminale; P insieme finito di produzioni con V^+ e V^*

Una grammatica è **generale** se non ci sono vincoli sulle produzioni
Una grammatica è **monotona** se le sue produzioni sono tipo con | | | |
Una grammatica è **dipendente dal contesto** se le sue produzioni sono del tipo A con , V^* A (V\T), V^+
Una grammatica è **libera** se le sue produzioni sono del tipo A con V^+
Una grammatica è **lineare** se le sue produzioni sono del tipo A con V^+ con al + 1 simbolo non terminale
Una grammatica è **lineare dx** se le sue produzioni sono del tipo A a o A a
Una grammatica è **lineare sx** se le sue produzioni sono del tipo A a o A a
Una grammatica libera è una forma **normale di chomsky** se le sue produzioni sono del tipo A BC o A a
Una grammatica libera è una forma **normale di Greibach** se le sue produzioni sono del tipo A a , con stringa di non terminali (possibilmente vuota)
Linguaggio Regolare: se è esprimibile attraverso un'expr. Reg. cioè se. E t.c. L = L(E) inoltre può essere generata da un ASFD
Una grammatica G è **ambigua** se data una stringa $\omega \in L(G)$ essa è generata da più di un albero di derivazione

La **classe dei linguaggi regolari** è la minima classe che contiene l'insieme vuoto e i singoletti $\{a_i\}$ con $a_i \in \Sigma$ che è chiusa per Unione,Conc. e S.d. Kleene

- APND
  è una tupla $M=(Q,\Sigma,\Gamma,t,q_0,Z_0,F)$:
  Q è un insieme finito di stati
  Σ alfabeto di imput (a,b,c,...)
  Γ alfabeto dello stack (A,B,C,...)
  t : $Qx(\Sigma x\{\varepsilon\}x\Gamma)x(Qx\Gamma^*)$
  q_0 è lo stato iniziale
  Z_0 è il simbolo iniziale della pila
  F è il simbolo degli stati finali o di accettazione
- ASFND
  è una 5-pla $M=(Q,A,t,q_0,F)$:
  Q è un insieme finito di stati
  A alfabeto finito
  $q_0 \in Q$ è lo stato iniziale
  $F \subseteq Q$ stati finali
  t : QxA^*xQ funzione di trasmissione
- APD
  è una tupla $M=(Q,\Sigma,\Gamma,t,q_0,Z_0,F)$:
  Q è un insieme finito di stati
  Σ alfabeto di imput (a,b,c,...)
  Γ alfabeto dello stack (A,B,C,...)
  t : $Qx(\Sigma x\{\varepsilon\}x\Gamma)\rightarrow 2^{(Qx\Gamma^*)}$
  q_0 è lo stato iniziale
  Z_0 è il simbolo iniziale della pila
  F è il simbolo degli stati finali o di accettazione
- ALND

è una tupla $M=(A,Q,B,t,q_0,F)$:
A è l'insime dei simboli dell alfabeto di imput $=\Sigma \cup \{¢,\$\}$
Q è un insieme finito di stati
B è l'insieme dei simboli dell alfabeto del mastro $A \subseteq B$ (sono quello che usa in piu rispetto all input)
$t : QxB \rightarrow 2^{(QxBx\{DxS\})}$ che equivale a dire $t \subseteq (QxB)x(QxBx\{DxS\})$
q_0 è lo stato iniziale
F è il simbolo degli stati finali o di accettazione

- ADTD

è una tupla $M=(A,Q,B,t,q_0,q_f)$:
A è l'insime dei simboli di imput
Q è un insieme finito di stati
B è l'insieme dei simboli del mastro (B contiene 'blank' usato per posizioni vuote, $A \subset B$)
t $QxB \rightarrow QxBx\{DxS\}$
q_0 è lo stato iniziale
q_f è lo stato finale

Semantica

Regole di Sematica Operazionale

– **Espressioni**

Var	$\langle v,\sigma\rangle \to_e \langle \sigma(v),\sigma\rangle$
Somma	$\dfrac{\langle e_0,\sigma\rangle \to_e \langle e_0',\sigma\rangle}{\langle e_0+e_1,\sigma\rangle \to_e \langle e_0'+e_1,\sigma\rangle}$ $\quad \dfrac{\langle e_1,\sigma\rangle \to_e \langle e_1',\sigma\rangle}{\langle m+e_1,\sigma\rangle \to_e \langle m+e_1',\sigma\rangle}$ $\langle m+m',\sigma\rangle \to_e \langle n,\sigma\rangle \quad n=m+m'$
Sottrazione	$\dfrac{\langle e_0,\sigma\rangle \to_e \langle e_0',\sigma\rangle}{\langle e_0-e_1,\sigma\rangle \to_e \langle e_0'-e_1,\sigma\rangle}$ $\quad \dfrac{\langle e_1,\sigma\rangle \to_e \langle e_1',\sigma\rangle}{\langle m-e_1,\sigma\rangle \to_e \langle m-e_1',\sigma\rangle}$ $\langle m-m',\sigma\rangle \to_e \langle n,\sigma\rangle \quad n=m-m' \wedge m\geq m'$
Moltiplicazione	$\dfrac{\langle e_0,\sigma\rangle \to_e \langle e_0',\sigma\rangle}{\langle e_0*e_1,\sigma\rangle \to_e \langle e_0'*e_1,\sigma\rangle}$ $\quad \dfrac{\langle e_1,\sigma\rangle \to_e \langle e_1',\sigma\rangle}{\langle m*e_1,\sigma\rangle \to_e \langle m*e_1',\sigma\rangle}$ $\langle m*m',\sigma\rangle \to_e \langle n,\sigma\rangle$

– **Espressioni booleane**

OR	$\dfrac{\langle b_0,\sigma\rangle \to_b \langle b_0',\sigma\rangle}{\langle b_0\vee b_1,\sigma\rangle \to_b \langle b_0'\vee b_1,\sigma\rangle}$ $\langle tt\vee b_1,\sigma\rangle \to_b \langle tt,\sigma\rangle$ $\langle ff\vee b_1,\sigma\rangle \to_b \langle b_1,\sigma\rangle$
EQ	$\dfrac{\langle e_0,\sigma\rangle \to_e \langle e_0',\sigma\rangle}{\langle e_0=e_1,\sigma\rangle \to_b \langle e_0'=e_1,\sigma\rangle}$ $\quad \dfrac{\langle e_1,\sigma\rangle \to_e \langle e_1',\sigma\rangle}{\langle m=e_1,\sigma\rangle \to_b \langle m=e_1',\sigma\rangle}$ $\langle m=n,\sigma\rangle \to_b \langle t,\sigma\rangle \quad con \quad t = \begin{matrix} tt & se\, m=n \\ ff & se\, m\neq n \end{matrix}$
NEG	$\dfrac{\langle b,\sigma\rangle \to_b \langle b',\sigma\rangle}{\langle \neg b,\sigma\rangle \to_b \langle \neg b',\sigma\rangle}$ $\quad \langle \neg t,\sigma\rangle \to_b \langle t',\sigma\rangle \quad con \quad t = \begin{matrix} tt & se\, t=ff \\ ff & se\, t=tt \end{matrix}$

– **Comandi**

NIL	$\langle nil,\sigma\rangle \to_c \sigma$
Assegnamento	$\dfrac{\langle e,\sigma\rangle \to_e^* \langle m,\sigma\rangle}{\langle v:=e,\sigma\rangle \to_c \sigma[m/v]}$
Composizione	$\dfrac{\langle c_0,\sigma\rangle \to_c \langle rigt\ c_0',\sigma'\rangle}{\langle c_0;c_1,\sigma\rangle \to_c \langle c_0';c_1,\sigma'\rangle}$ $\quad \dfrac{\langle c_0,\sigma\rangle \to_c \sigma'}{\langle c_0;c_1,\sigma\rangle \to_c \langle c_1,\sigma'\rangle}$
If	$\dfrac{\langle b,\sigma\rangle \to_b^* \langle tt,\sigma\rangle}{\langle \textbf{if}\ b\ \textbf{then}\ c_1\ \textbf{else}\ c_2,\sigma\rangle \to_c \langle c_1,\sigma\rangle}$ $\quad \dfrac{\langle b,\sigma\rangle \to_b^* \langle ff,\sigma\rangle}{\langle \textbf{if}\ b\ \textbf{then}\ c_1\ \textbf{else}\ c_2,\sigma\rangle \to_c \langle c_2,\sigma\rangle}$

While	$\dfrac{\langle b, \sigma\rangle \rightarrow_b^* \langle tt, \sigma\rangle}{\langle \textbf{\textit{while}}\ b\ \textbf{do}\ c, \sigma\rangle \rightarrow_c \langle c; \textbf{\textit{while}}\ b\ \textbf{do}\ c, \sigma\rangle}$ $\quad \dfrac{\langle b, \sigma\rangle \rightarrow_b^* \langle ff, \sigma\rangle}{\langle \textbf{\textit{while}}\ b\ \textbf{do}\ c, \sigma\rangle \rightarrow_c \langle \sigma\rangle}$
While (alternativa)	$\langle \textbf{\textit{while}}\ b\ \textbf{do}\ c, \sigma\rangle \rightarrow_c \langle \textbf{\textit{if}}\ b\ \textbf{\textit{then}}\ c; \textbf{\textit{while}}\ b\ \textbf{do}\ c\ \textbf{else}\ nil, \sigma\rangle$