



Sintesi Sequenziale Sincrona

Sintesi comportamentale di reti sequenziali sincrone

Ant Brain

versione del 16/12/08





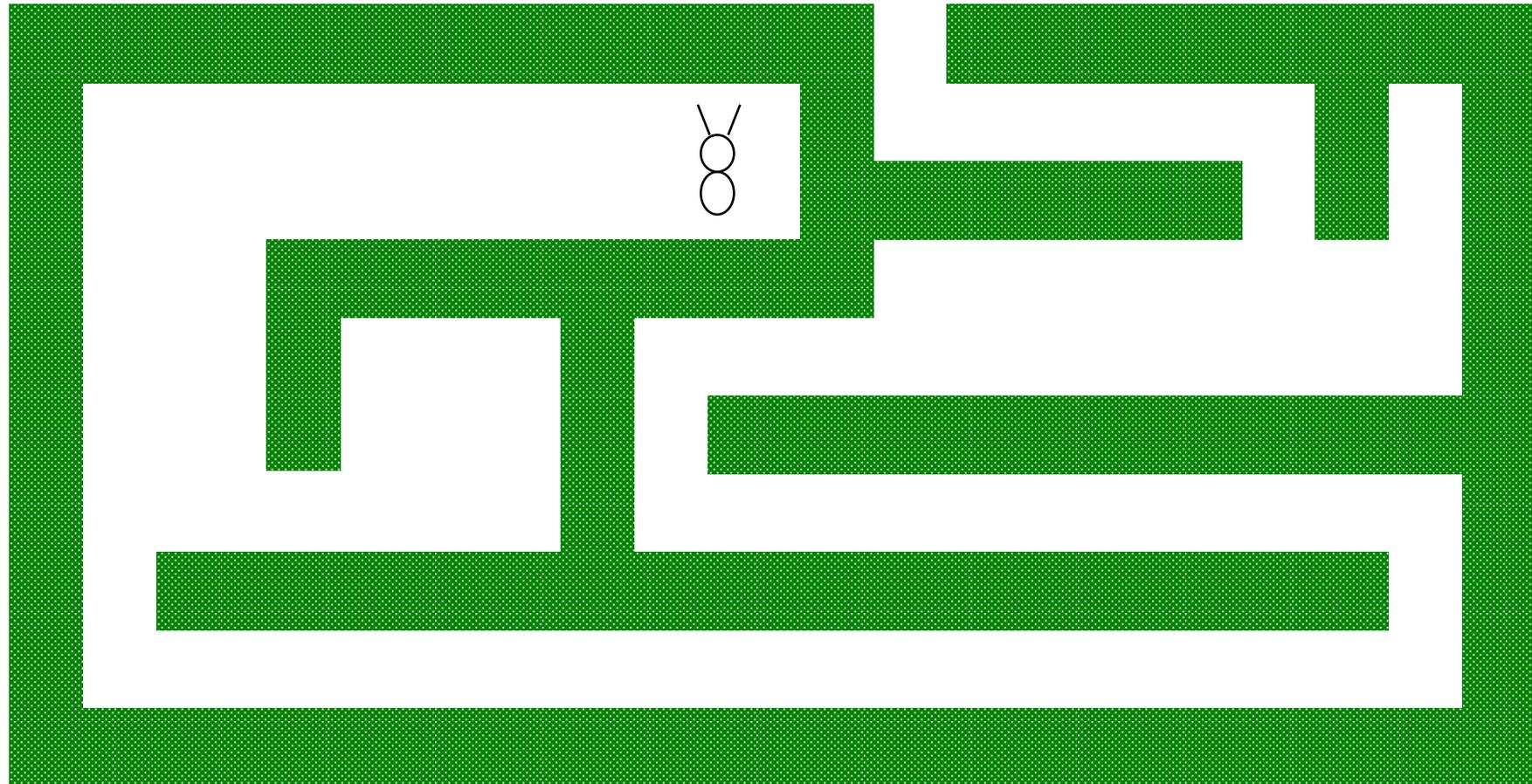
- La sintesi si svolge nei seguenti passi:
 1. Realizzazione del *diagramma degli stati* a partire dalle specifiche informali del problema
 2. Costruzione della *tabella degli stati*
 3. Riduzione del numero degli stati: *ottimizzazione*
 4. Costruzione della *tabella delle transizioni*
 - Assegnamento degli stati: *Codice & codifica*
 5. Costruzione della *tabella delle eccitazioni*
 - Scelta degli elementi di memoria
 6. Sintesi sia della rete combinatoria che realizza la funzione stato prossimo sia della rete combinatoria che realizza la funzione d'uscita





Ant Brain

μ -LAB





Ant Brain: Specifiche

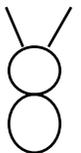
μ -LAB

- Sensori:
 - L e R antenne;
 - Assumono valore 1 se toccano il muro.

- Attuatori:
 - F: procedi dritto;
 - TL: Gira piano a sinistra;
 - TR : Gira piano a destra.

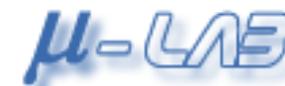
- Goal: trovare l'uscita dal labirinto

- Strategia: tenere il muro a destra

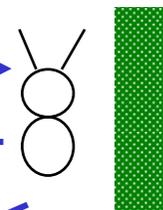




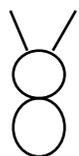
Comportamento Formica



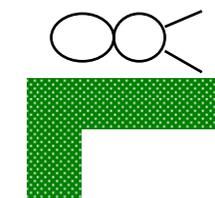
A: Segui il muro toccandolo.
Vai dritto, girando leggermente
a sinistra.



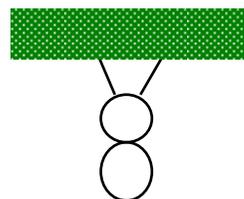
B: Segui il muro senza toccarlo.
Vai dritto, girando leggermente
a destra.



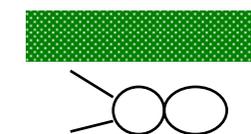
C: Rottura nel muro.
Vai dritto, girando leggermente
a destra.



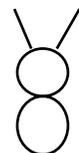
D: Toccato il muro nuovamente.
Torna allo stato A.



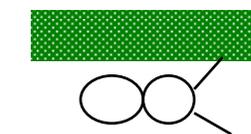
E: Muro di fronte.
Gira a sinistra finchè...



F: ...siamo qui, stessa
situazione dello stato B.



LOST: Dritto fino a quando non
Tocchi qualche cosa.



G: Gira a sinistra finchè...



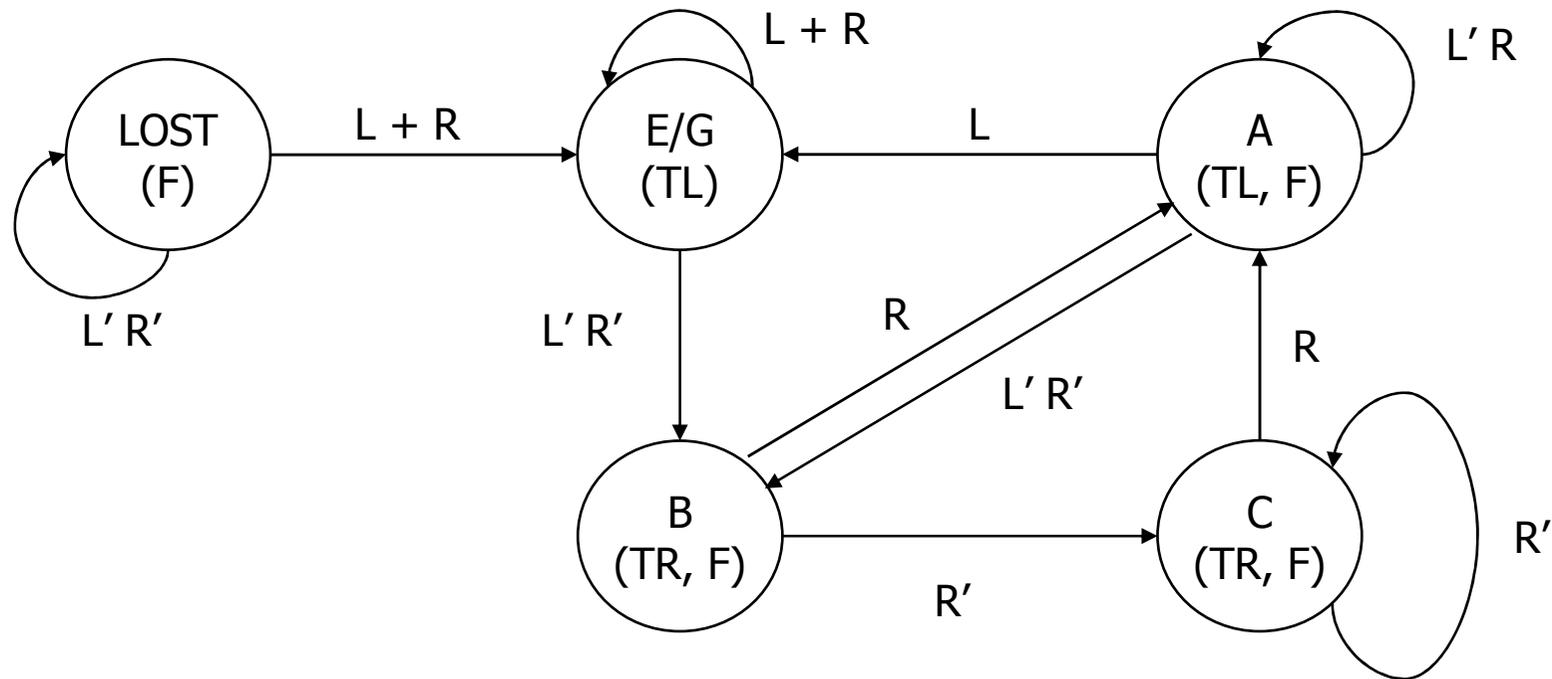


- La sintesi si svolge nei seguenti passi:
 1. **Realizzazione del *diagramma degli stati* a partire dalle specifiche informali del problema (15 min)**
 2. Costruzione della *tabella degli stati*
 3. Riduzione del numero degli stati: *ottimizzazione*
 4. Costruzione della *tabella delle transizioni*
 - Assegnamento degli stati: *Codice & codifica*
 5. Costruzione della *tabella delle eccitazioni*
 - Scelta degli elementi di memoria
 6. Sintesi sia della rete combinatoria che realizza la funzione stato prossimo sia della rete combinatoria che realizza la funzione d'uscita





Disegno del Cervello della Formica: Diagramma degli Stati





- La sintesi si svolge nei seguenti passi:
 1. Realizzazione del *diagramma degli stati* a partire dalle specifiche informali del problema
 2. **Costruzione della *tabella degli stati* (10min)**
 3. Riduzione del numero degli stati: *ottimizzazione*
 4. Costruzione della *tabella delle transizioni*
 - Assegnamento degli stati: *Codice & codifica*
 5. Costruzione della *tabella delle eccitazioni*
 - Scelta degli elementi di memoria
 6. Sintesi sia della rete combinatoria che realizza la funzione stato prossimo sia della rete combinatoria che realizza la funzione d'uscita





Tabella degli stati



Il comportamento di una FSM può essere descritto mediante la **Tabella degli stati**

- Gli indici di **colonna** sono i **simboli di ingresso** $i_\alpha \in I$
- Gli indici di **riga** sono i simboli di stato $s_j \in S$ che indicano lo **stato presente**
- Gli elementi della tabella sono:
 - **Macchine di Mealy**
 - la **coppia** $\{u_\beta, s_j\}$
 - $u_\beta = \lambda(i_\alpha, s_j)$ è il simbolo di uscita
 - $s_j = \delta(i_\alpha, s_j)$ è il simbolo stato prossimo

	i_1	i_2	...
s_1^t	s_j^{t+1}/u_j	s_k^{t+1}/u_k	...
s_2^t	s_m^{t+1}/u_m	s_1^{t+1}/u_1	...
...

- **Macchine di Moore**
- Il simbolo **stato prossimo** s_j
- $s_j = \delta(i_\alpha, s_j)$ è il simbolo stato prossimo
- i simboli d'uscita sono associati allo stato presente

	i_1	i_2	...	
s_1^t	s_j^{t+1}	s_k^{t+1}	...	u_1
s_2^t	s_m^{t+1}	s_1^{t+1}	...	u_2
...





Macchine di Mealy e Macchine di Moore

μ -LAB

□ Macchine di Mealy

- la funzione di uscita costituisce la risposta della macchina quando, trovandosi in un certo stato presente, riceve un simbolo di ingresso
- nelle macchine di Mealy, l'uscita va "letta" mentre la macchina subisce una transizione di stato

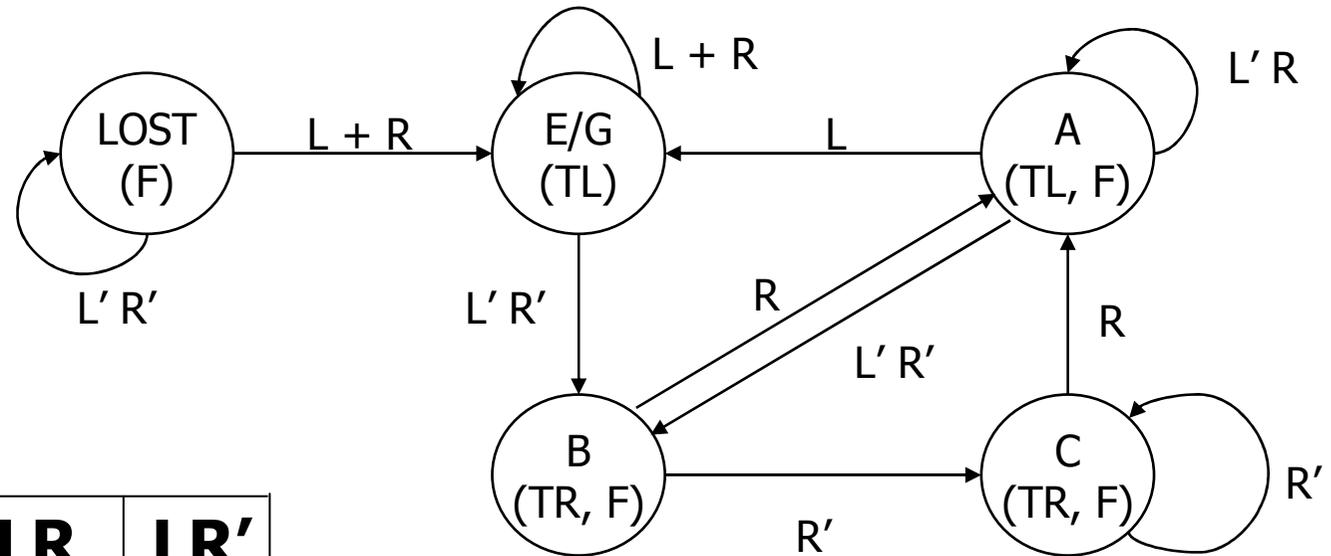
□ Macchine di Moore

- la funzione di uscita costituisce la risposta della macchina associata allo stato in cui si trova nelle macchine di Moore, l'uscita viene letta mentre la macchina si trova in un determinato stato
-
- E' possibile trasformare una macchina di Mealy in una macchina equivalente di Moore, e viceversa





Tabella degli stati: Moore

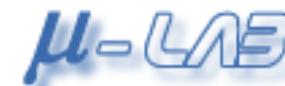


	L'R'	L'R	LR	LR'	
Lost	Lost	E/G	E/G	E/G	F TR' TL'
E/G	B	E/G	E/G	E/G	F' TR' TL
A	B	A	E/G	E/G	F TR' TL
B	C	A	A	C	F TR TL'
C	C	A	A	C	F TR TL'





Tabella degli stati: Mealy



	L'R'	L'R	LR	LR'
Lost	Lost/ _{F TR' TL'}	E/G/ _{F TR' TL'}	E/G/ _{F TR' TL'}	E/G/ _{F TR' TL'}
E/G	B/ _{F' TR' TL}	E/G/ _{F' TR' TL}	E/G/ _{F' TR' TL}	E/G/ _{F' TR' TL}
A	B/ _{F TR' TL}	A/ _{F TR' TL}	E/G/ _{F TR' TL}	E/G/ _{F TR' TL}
B	C/ _{F TR TL'}	A/ _{F TR TL'}	A/ _{F TR TL'}	C/ _{F TR TL'}
C	C/ _{F TR TL'}	A/ _{F TR TL'}	A/ _{F TR TL'}	C/ _{F TR TL'}



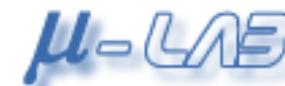


- La sintesi si svolge nei seguenti passi:
 1. Realizzazione del *diagramma degli stati* a partire dalle specifiche informali del problema
 2. Costruzione della *tabella degli stati*
 - 3. Riduzione del numero degli stati: *ottimizzazione* (15 min)**
 4. Costruzione della *tabella delle transizioni*
 - Assegnamento degli stati: *Codice & codifica*
 5. Costruzione della *tabella delle eccitazioni*
 - Scelta degli elementi di memoria
 6. Sintesi sia della rete combinatoria che realizza la funzione stato prossimo sia della rete combinatoria che realizza la funzione d'uscita





Riduzione del numero degli stati: macchine equivalenti



- Date due macchine completamente specificate M1 e M2 si dicono **equivalenti** se e solo se:
 - per ogni stato s_i di M1, esiste uno stato s_j di M2 tale che ponendo la macchina M1 in s_i e la macchina M2 in s_j
 - e applicando alle due macchine una qualunque sequenza di ingresso I
 - le due sequenze di uscita sono identiche.
 - E viceversa per M2 rispetto ad M1

- **Nota:** nella **definizione di equivalenza** sono considerate solo le relazioni **ingresso-uscita** quindi le due macchine possono avere un insieme di stati diverso e in particolare di **diversa cardinalità**

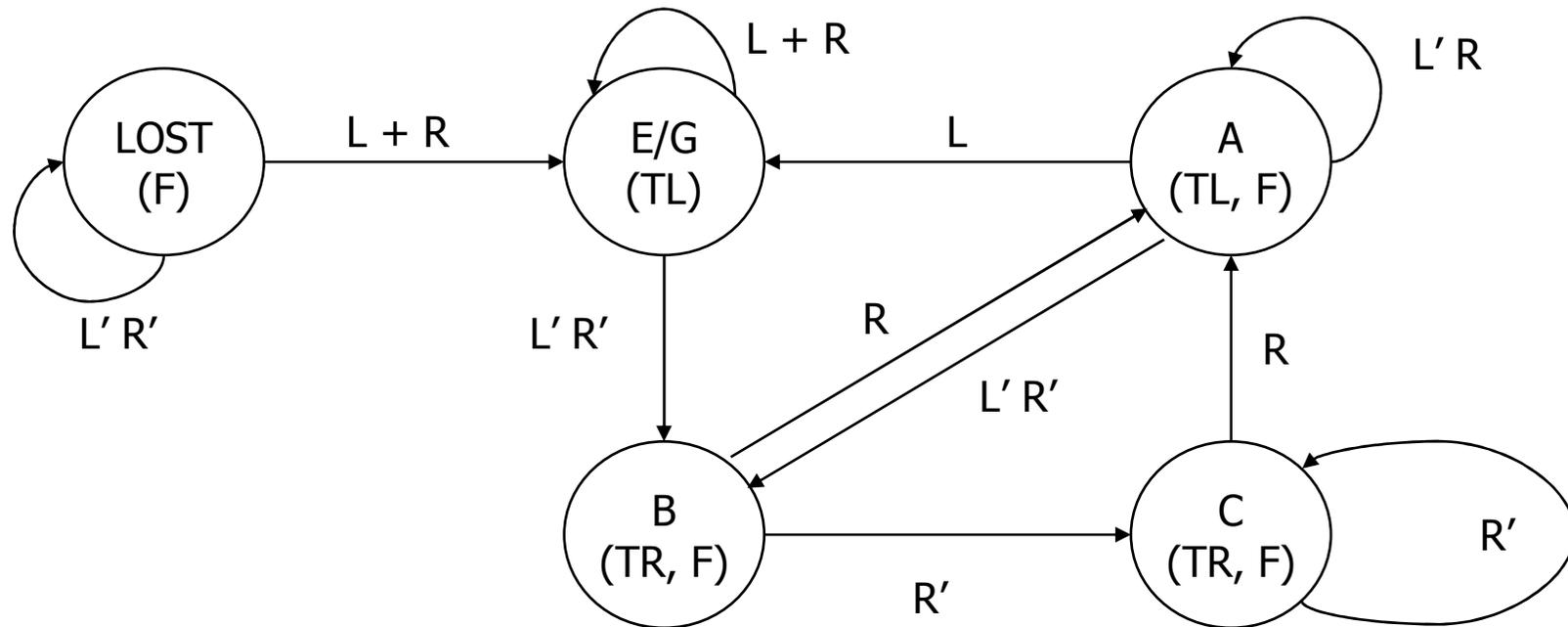




Rivisitazione dell'Ant Brain



- Nessuno stato equivalente?





Riduzione del numero degli stati: stati indistinguibili di una stessa macchina



- Data una macchina completamente specificata, siano:
 - I_α - una generica sequenza di ingresso i_j, \dots, i_k
 - U_α - la sequenza d'uscita ad essa associata ottenuta attraverso λ .
 - s_i, s_j - due generici stati

- I due stati s_i e s_j appartenenti ad S sono *indistinguibili* se:

$$U_{\alpha,i} = \lambda(s_i, I_\alpha) = \lambda(s_j, I_\alpha) = U_{\alpha,j} \quad \forall I_\alpha$$

- ponendo la macchina in s_i oppure in s_j e applicando una qualsiasi sequenza di ingresso, le uscite sono identiche.

- L'indistinguibilità tra s_i e s_j si indica con: $s_i \sim s_j$





Riduzione del numero degli stati: identificazione degli stati equivalenti



- La definizione di **indistinguibilità** tra stati è di difficile applicabilità poiché richiederebbe di considerare *tutte* le sequenze di ingresso (a priori infinite)
- Si ricorre ad una regola introdotta da **Paull - Unger**
 - Due stati s_i e s_j appartenenti ad S sono indistinguibili se e solo se per ogni simbolo di ingresso i_a :
 $\lambda(s_i, i_a) = \lambda(s_j, i_a)$ (Le uscite sono uguali per ogni simbolo di ingresso)
 $\delta(s_i, i_a) \sim \delta(s_j, i_a)$ (Gli stati prossimi sono indistinguibili)
- La **regola di Paull - Unger** è iterativa





Tabella delle implicazioni

	L'R'	L'R	LR	LR'
Lost	Lost/ $F TR' TL$	E/G/ $F TR' TL'$	E/G/ $F TR' TL'$	E/G/ $F TR' TL'$
E/G	B/ $F' TR' TL$	E/G/ $F' TR' TL$	E/G/ $F' TR' TL$	E/G/ $F' TR' TL$
A	B/ $F TR' TL$	A/ $F TR' TL$	E/G/ $F TR' TL$	E/G/ $F TR' TL$
B	C/ $F TR TL'$	A/ $F TR TL'$	A/ $F TR TL'$	C/ $F TR TL'$
C	C/ $F TR TL'$	A/ $F TR TL'$	A/ $F TR TL'$	C/ $F TR TL'$





Tabella delle implicazioni

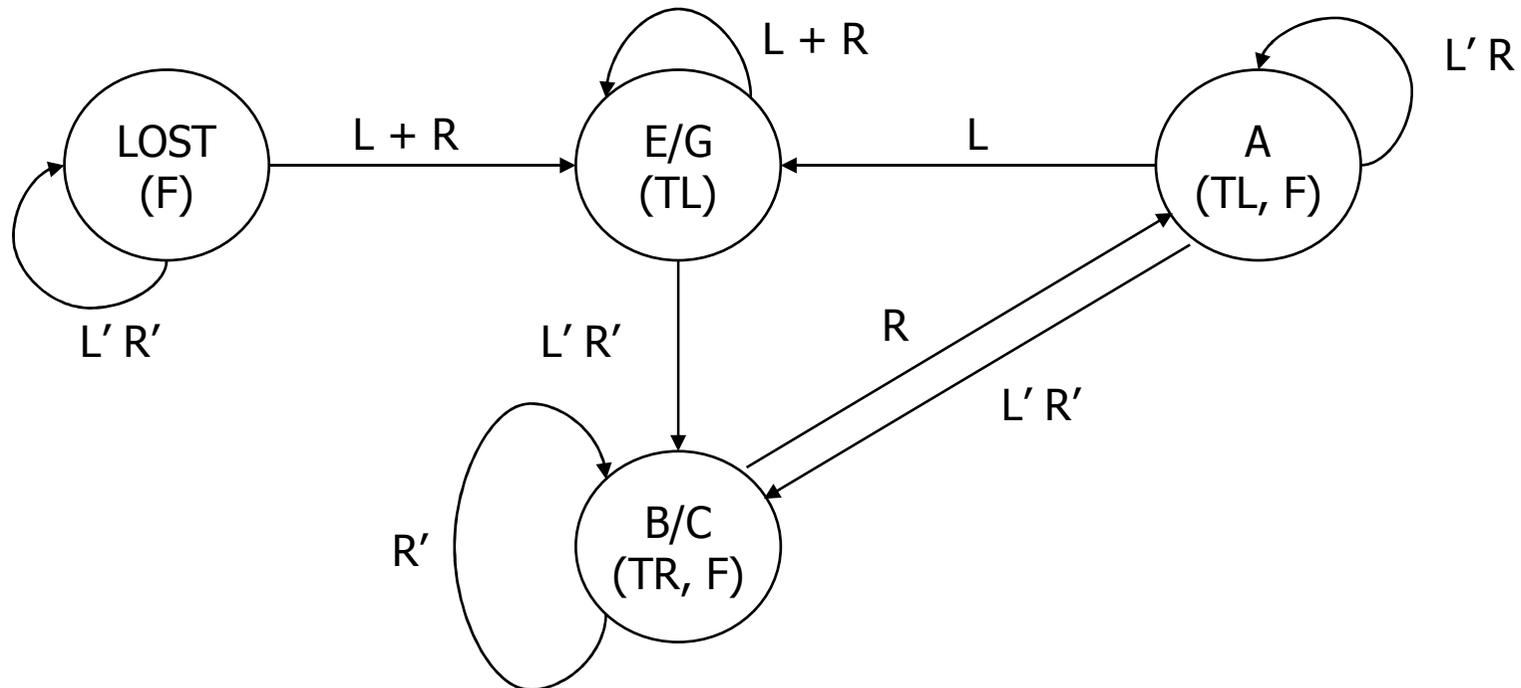
E/G	x			
A	x	x		
B	x	x	x	
C	x	x	x	~
	Lost E/G	A	B	





Nuovo Cervello Migliorato

- Unione degli stati B e C;
- Il comportamento è esattamente quello del cervello a 5 stati;
- Ora ci servono solo 2 variabili di stato invece di 3.





- La sintesi si svolge nei seguenti passi:
 1. Realizzazione del *diagramma degli stati* a partire dalle specifiche informali del problema
 2. Costruzione della *tabella degli stati*
 3. Riduzione del numero degli stati: *ottimizzazione*
 4. **Costruzione della *tabella delle transizioni* (10 min)**
 - **Assegnamento degli stati: *Codice & codifica***
 5. Costruzione della *tabella delle eccitazioni*
 - Scelta degli elementi di memoria
 6. Sintesi sia della rete combinatoria che realizza la funzione stato prossimo sia della rete combinatoria che realizza la funzione d'uscita





Sintesi: Scelta del codice



- Il processo di codifica degli stati ha l'obiettivo di identificare per ogni rappresentazione simbolica dello stato una corrispondente rappresentazione binaria.
- Due problemi paralleli:
 - Scelta del codice.
 - A minimo numero di bit
 - n° di elementi di memoria = $\lceil \log_2 |S| \rceil$ (codifica densa)
 - One-Hot
 - n° di elementi di memoria = $|S|$ (codifica sparsa)
 - Distanza Minima
 - Gli stati che sono in corrispondenza delle transizioni più frequenti sono poste a distanza Hamming più piccola possibile ponendo il vincolo del minor numero possibile di bit.
 - ...
 - Identificazione della codifica di ogni stato.





Assegnamento degli stati



- 2 Stati:
 - Richieste almeno 2 variabili di stato: X, Y;
 - Almeno 2 bistabili.

LOST	- 00
E/G	- 01
A	- 10
B/C	- 11





Tabella delle transizioni

μ -LAB

	00	01	11	10	
00	00	01	01	01	100
01	11	01	01	01	001
11	11	10	10	11	110
10	11	10	01	01	101





Assegnamento degli stati: caso non ottimizzato

μ-LAB

- 5 Stati:
 - Richieste almeno 3 variabili: X, Y, Z

LOST	- 000
E/G	- 001
A	- 010
B	- 011
C	- 100





Tabella delle transizioni: caso non ottimizzato

μ-LAB

	L'R'	L'R	LR	LR'	
000	000	001	001	001	100
001	011	001	001	001	001
010	011	010	001	001	101
011	100	010	010	100	110
100	100	010	010	100	110





- La sintesi si svolge nei seguenti passi:
 1. Realizzazione del *diagramma degli stati* a partire dalle specifiche informali del problema
 2. Costruzione della *tabella degli stati*
 3. Riduzione del numero degli stati: *ottimizzazione*
 4. Costruzione della *tabella delle transizioni*
 - Assegnamento degli stati: *Codice & codifica*
 - 5. Costruzione della *tabella delle eccitazioni* (10 min)**
 - Scelta degli elementi di memoria**
 6. Sintesi sia della rete combinatoria che realizza la funzione stato prossimo sia della rete combinatoria che realizza la funzione d'uscita





Implementazione: caso ottimizzato



- Le uscite sono funzioni solo dello stato corrente:
Macchiana di Moore

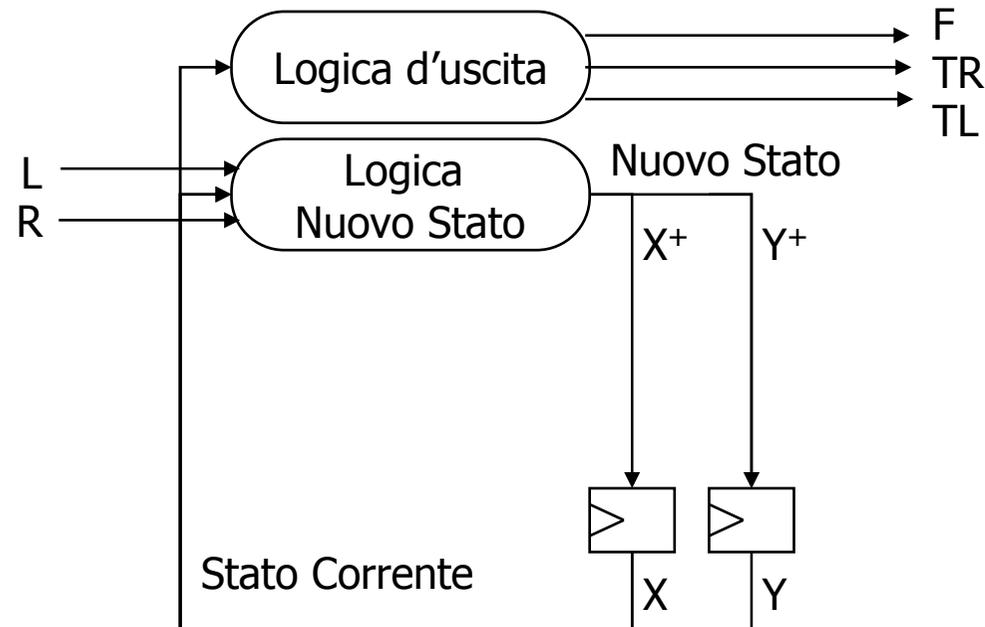




Tabella delle eccitazioni



Tabella delle eccitazioni
di un bistabile di tipo D

Q	Q'	D
0	0	0
0	1	1
1	0	0
1	1	1

Tabella delle transizioni

	00	01	11	10
00	00	01	01	01
01	11	01	01	01
11	11	10	10	11
10	11	10	10	01

Tabella delle
eccitazioni (con D)

		LR			
		00	01	11	10
XY	00	0 0	0 1	0 1	0 1
	01	1 1	0 1	0 1	0 1
	11	1 1	1 0	1 0	1 1
	10	1 1	1 0	1 0	0 1





Tabella delle eccitazioni



Tabella delle eccitazioni di un bistabile di tipo SR

Q	Q'	SR
0	0	0-
0	1	10
1	0	01
1	1	-0

Tabella delle transizioni

	00	01	11	10
00	00	01	01	01
01	11	01	01	01
11	11	10	10	11
10	11	10	10	01

Tabella delle eccitazioni (con SR)

		LR			
		00	01	11	10
XY	00	0- 0-	0- 10	0- 10	00 10
	01	10 -0	0- -0	0- -0	0- -0
11	-0 -0	-0 01	-0 01	-0 -0	
10	-0 10	-0 0-	-0 0-	01 10	





Sintesi: Mappe di Karnaugh



Le quattro mappe di Karnaugh che si ottengono sono quindi:

	00	01	11	10
00	0	1	1	1
01	-	-	-	-
11	-	0	0	-
10	1	0	0	1

Set₀ = ...

	00	01	11	10
00	-	0	0	0
01	0	0	0	0
11	0	1	1	0
10	0	-	-	0

Res₀ = ...

	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	-	-	-	-
10	-	-	-	0

Set₁ = ...

	00	01	11	10
00	-	-	-	0
01	0	-	-	-
11	0	0	0	0
10	0	0	0	1

Res₁ = ...

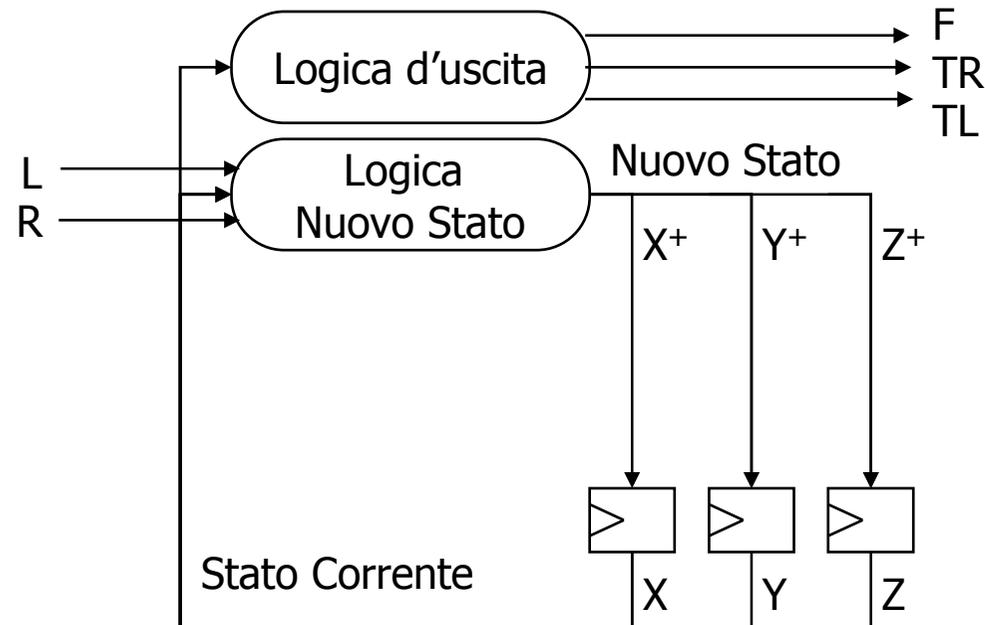




Implementazione: caso non ottimizzato



- Le uscite sono funzioni solo dello stato corrente:
Macchiana di Moore





- La sintesi si svolge nei seguenti passi:
 1. Realizzazione del *diagramma degli stati* a partire dalle specifiche informali del problema
 2. Costruzione della *tabella degli stati*
 3. Riduzione del numero degli stati: *ottimizzazione*
 4. Costruzione della *tabella delle transizioni*
 - Assegnamento degli stati: *Codice & codifica*
 5. Costruzione della *tabella delle eccitazioni*
 - Scelta degli elementi di memoria
 - 6. Sintesi sia della rete combinatoria che realizza la funzione stato prossimo sia della rete combinatoria che realizza la funzione d'uscita (*casa*)**

