

VARIANTI DI TM

Lo convinciamo che la nostra def. di TM ha alcune scelte arbitrarie, ma queste sono tutte dovute a convenzioni.

Alcuni esempi:

- Nastro infinito in 2 direzioni
- TM che muoversi e R, L, S ("stay put").
- TM con k -nastro.
- TM non deterministica.

Manterremo approccio ad alto livello,
come nella progettazione di algoritmi.
Vediamo prima il caso:

$$\delta' : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, S\}.$$

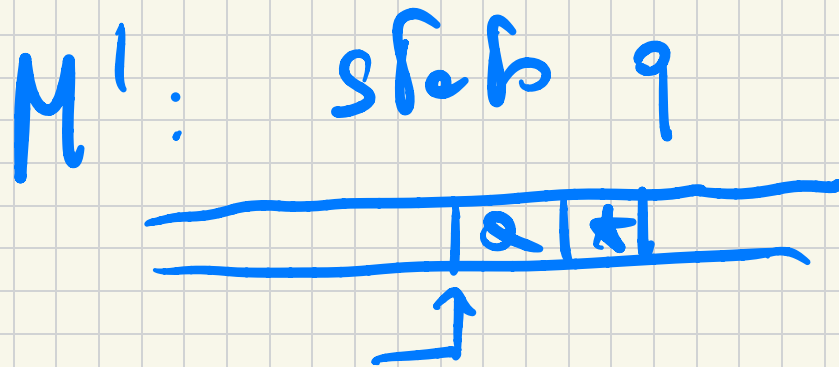
Semplicemente una TM standard può
simulare TM R' come sopra muovendo
prima e L e poi a R ogni volta
che R' si ferma.

Per il dettaglio: $\delta'(q, a) = (p, b, S)$

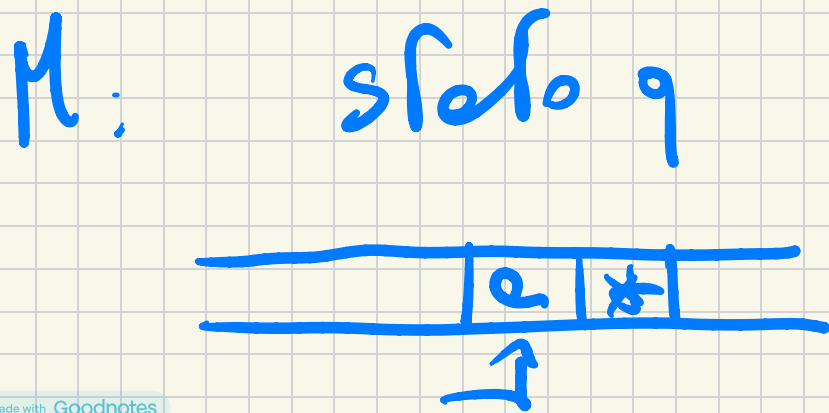
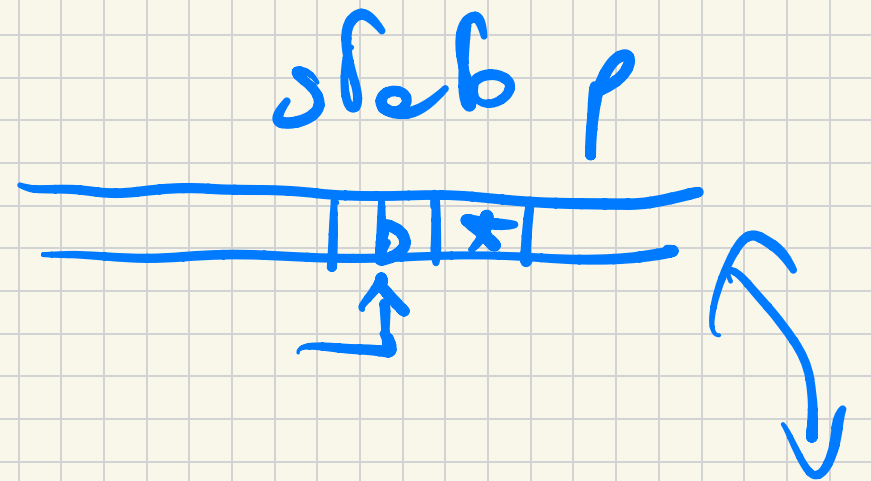
si produce un nuovo stato π e aggiungo

$$\delta(q, a) = (\pi, b, R)$$

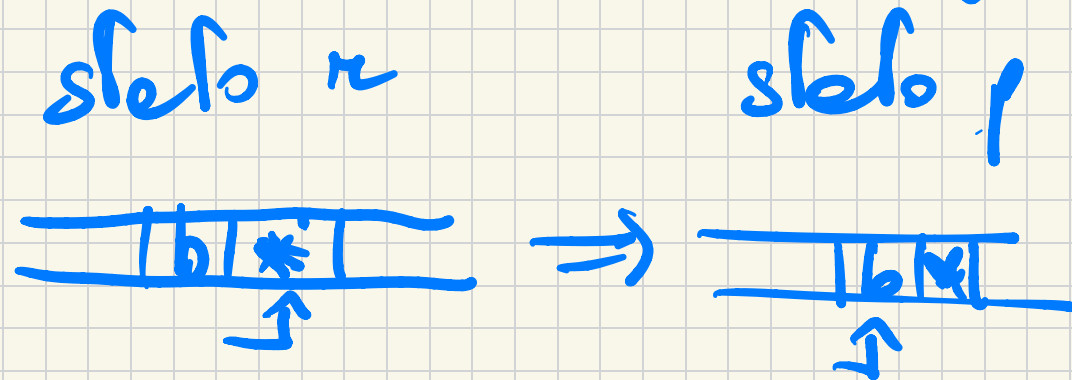
$$\delta(\pi, *) = (p, *, L)$$



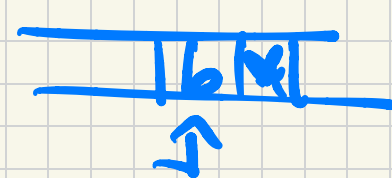
$\Rightarrow$



$\Rightarrow$



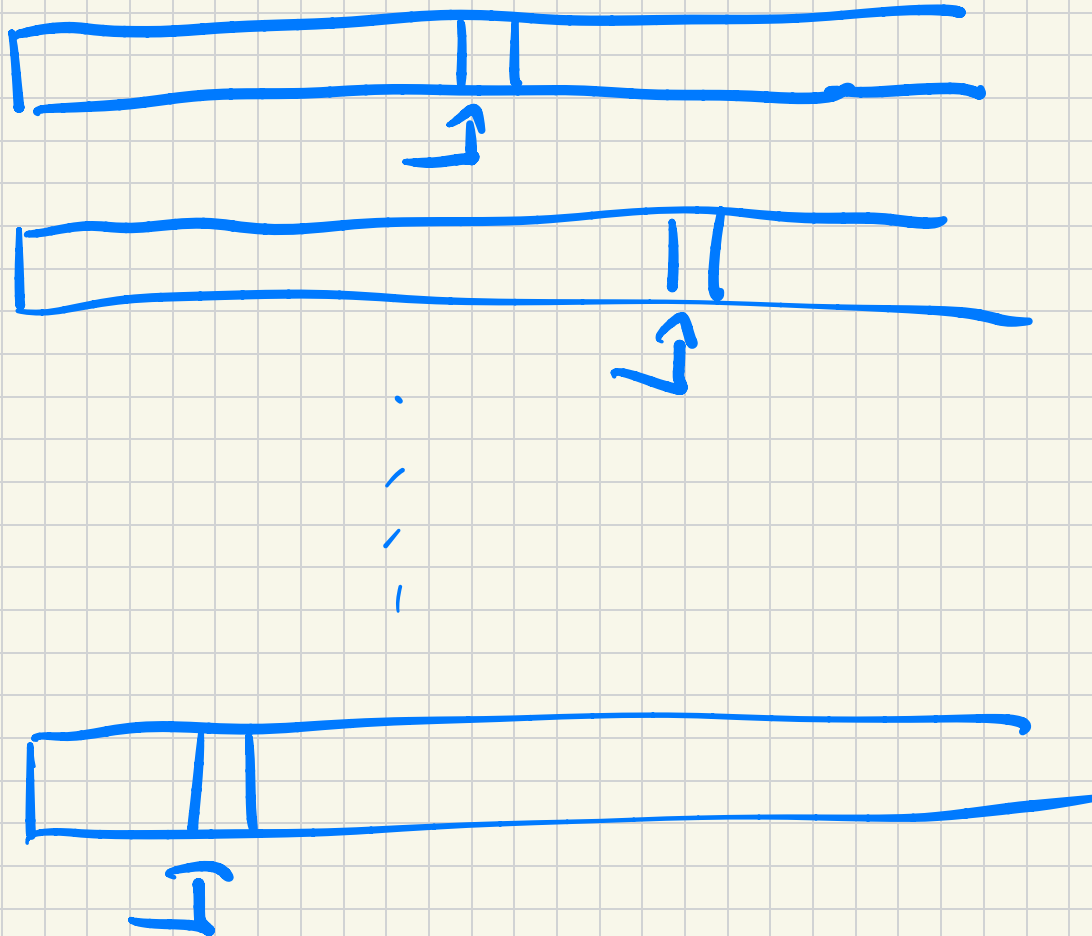
$\Rightarrow$



be TM multiset σ_0 :

$$\sigma' : Q \times \Gamma^K \rightarrow Q \times \Gamma^K \times \{L, R, S\}^K$$

Step 9



TEO Per ogni TM M' multinauto
ne esiste uno M singolo naivo equivalente.

COR Un linguaggio è $TURING-DEC.$

($TURING-REC$) sse esiste una
TM multinauto che lo decide/rec.

Take: Devo simulare sul singolo
naivo di M tutto il naivo di
 M' .

La TM M riconosce alcuni simboli
 speciali: $\#$ per separare i nostri
 da Π' ; $\forall a \in \Gamma$, almeno anche
 $\hat{a}$ la sua versione marcata per ricordare
 le posizioni delle χ definite su Π' .

Codice di M :

- All'inizio, M riconosce su
 questa configurazione

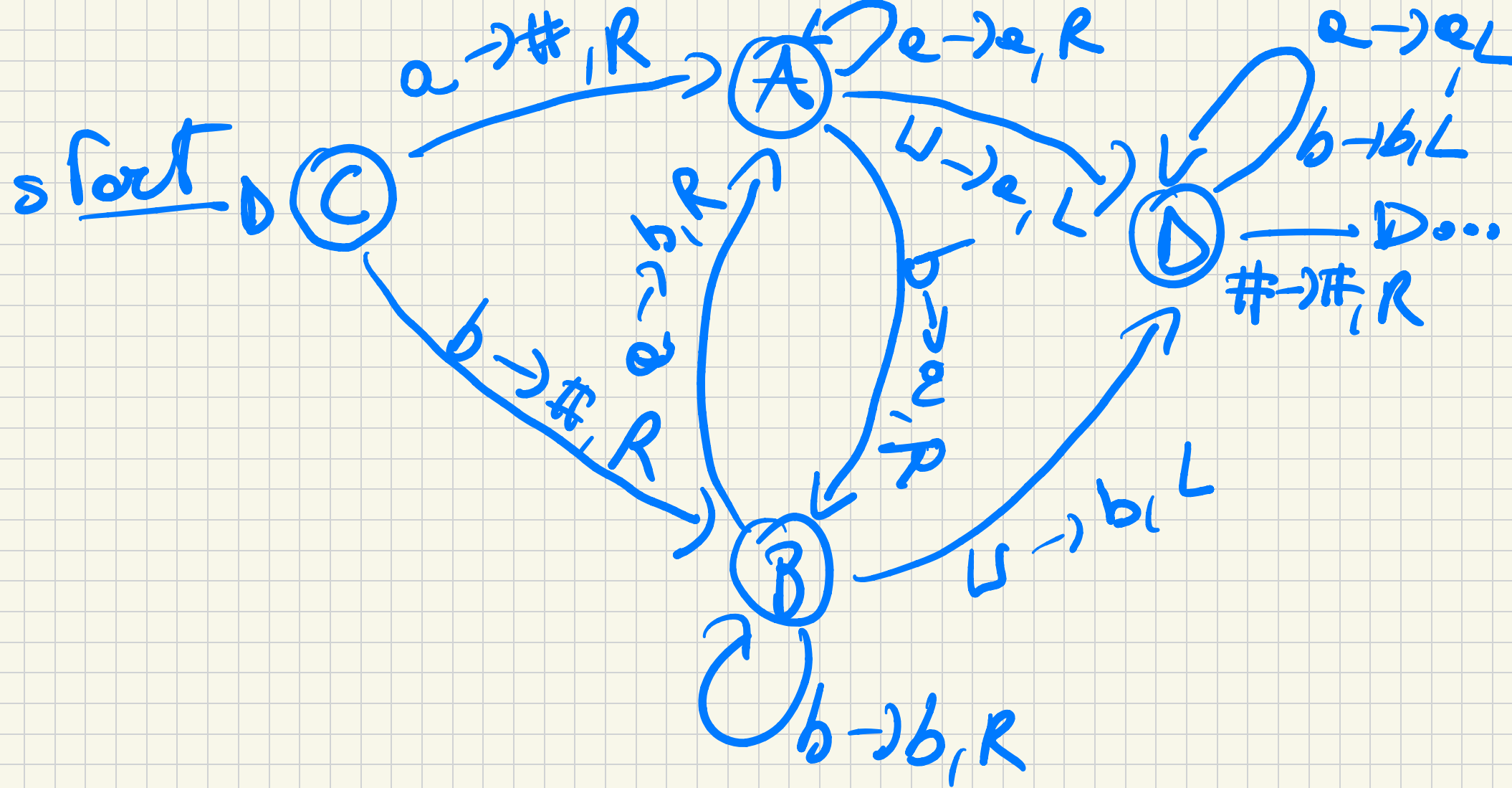
$\# \hat{w}_1 \dots w_n \# \hat{q} \# \dots \# \hat{q} \# \sqcup$

- Simulo una mossa di M' :
 - Spostano il matto per leggere
i simboli marcati in ognuno
dei k mattoni
 - Torno indietro al primo #.
 - Spostano nuovamente andando ad
aggiornare i simboli marcati
in accordo alla g' .
- Se M' costringe M a muovere
le pedine emulato da un suo

maestro su $\#$, M sposta tutto
il maestro a R da una posizione
e inserisce il simbolo $\hat{L}$.

~~Il~~
A livello implementativo, mostrare
il simbolo è una semplice sostituzione.
Per spostare il maestro o ricalcolare
il numero del maestro:

$$\text{Su } \Sigma = \{a, b\}$$



Le TM non deterministico (NTM):

$$\delta': Q \times \Gamma \rightarrow P(Q \times \Gamma \times \{L, R\})$$

Niente di nuovo, come per $\sim$ PDA e $\sim$ DFA. Le NTM accetta se esiste un cammino di computazione accettabile. Gli altri cammini possono rifiutare o non terminare mai.

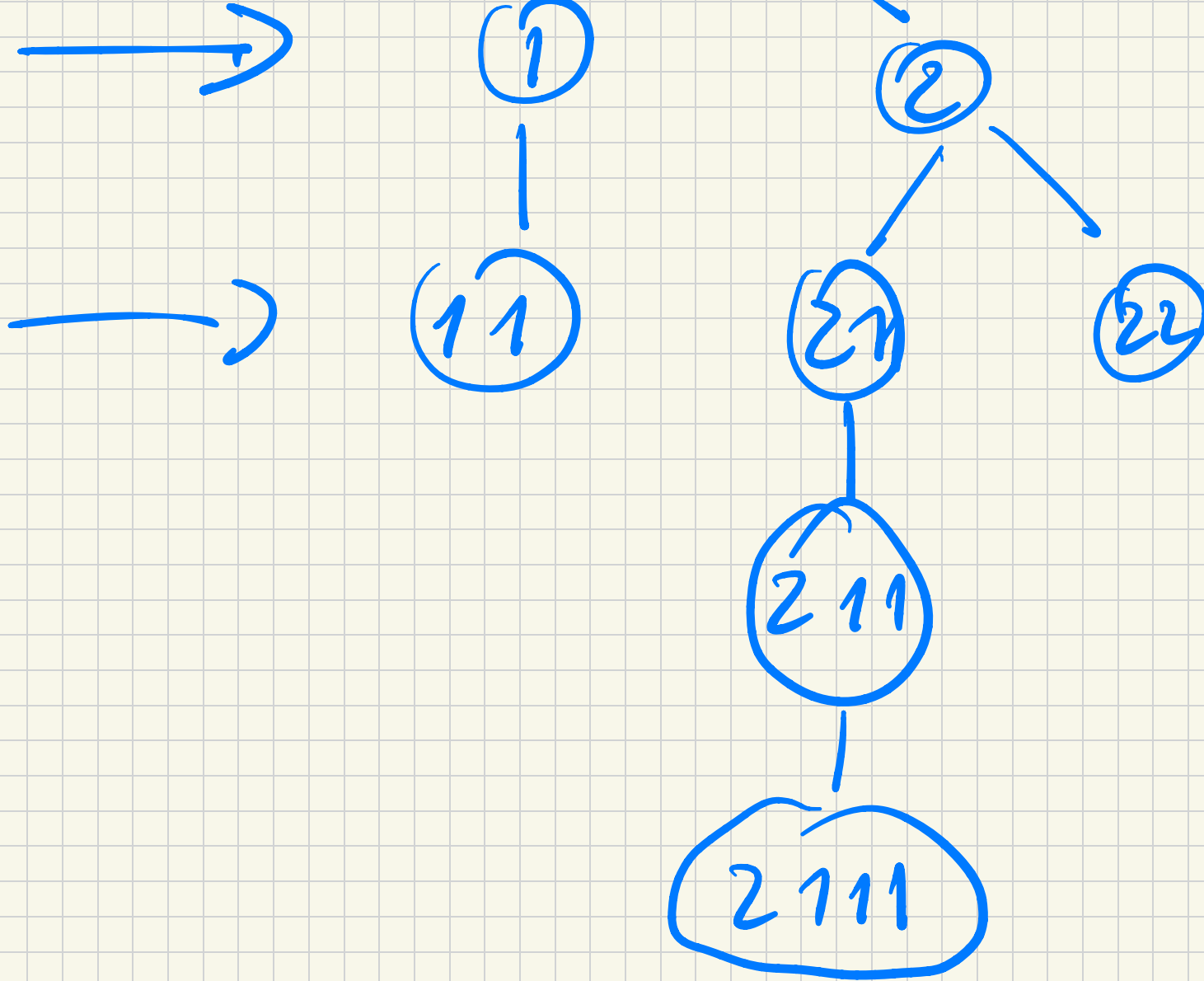
TEO Per ogni NTM N esiste una
TM M equivalente.

COR Un linguaggio è Turing-Decidibile
sse $\exists$ NTM che lo decide/rec.

Take: la TM M prova tutte le
possibili scelte non-det. di N . Se
ne esiste un'unica che accetta, M
accetta.

la nostra NTM avrà un altro
computer;

grado di
non-det. 2.



Vlog. assieme M con 3 master:

- Primo master contiene $v = v_1 \dots v_n$
- Secondo master, master di lavoro
- Terzo master: controller per monitorare l'andamento del master e cui mi devo fermare nelle simulazioni di un cammino per N. Come nelle figure sopra.

Importante: Esploro l'albero di
N in ordine; prima a noi
con movimento corrispondente e profon-
dità 1, poi 2, poi 3, ...
Così non vedo mai un loop e prima
o poi se N è nella w trovo un
cammino da eccelle. Appena questo
successo eccelle; strumento incrementale
l'input e ricomincia con la
simulazione fino al prossimo node. 