
Αυτοματοποιημένη Επαλήθευση

Στην ενότητα αυτή θα μελετηθούν τα εξής θέματα:

Έλεγχος Μοντέλου

- *Αλγόριθμοι γράφων*
- *Αλγόριθμοι αυτομάτων*

Αυτόματα ως προδιαγραφές

Έλεγχος μοντέλου

- Το σύστημα μοντελοποιείται ως ένας *γράφος*.
- Οι προδιαγραφές αναφέρονται στη γραφική αναπαράσταση.
- Για τον έλεγχο χρησιμοποιούμε αλγόριθμους γράφων.
- Με αυτόν τον τρόπο μπορούμε να ελέγξουμε ιδιότητες όπως
 - *Αμετάβλητη συνθήκη*: ιδιότητες που πρέπει να ισχύουν σε κάθε κατάσταση του μοντέλου
 - *Ανίχνευση αδιεξόδων*: υπάρχει κάποια κατάσταση από την οποία κάθε πρόοδος είναι αδύνατη;
 - *Νεκρός κώδικας*: υπάρχουν κομμάτια του κώδικα που δεν θα εκτελεστούν ποτέ;

Αλγόριθμος επαλήθευσης

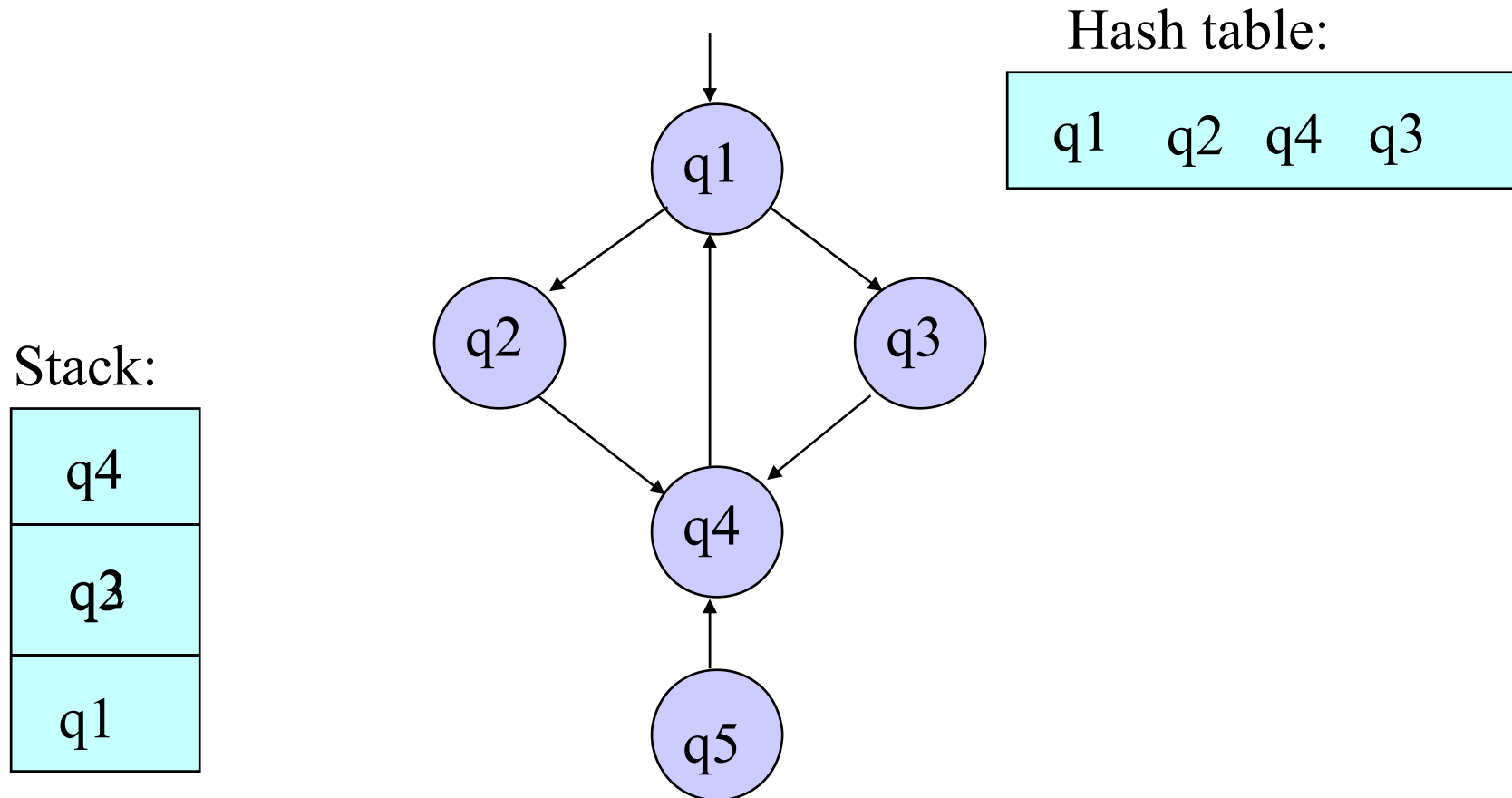
- Εφαρμόζουμε κάποια διαδικασία διάσχισης του γράφου (κατά βάθος ή κατά πλάτος διερεύνηση).
- Κατά τη διάσχιση ελέγχουμε τις καταστάσεις και τις μεταβάσεις.
- Αν διαπιστώσουμε κάποιο λάθος επιστρέφουμε αντιπαράδειγμα.

Διαδικασία διάσχισης κατά βάθος

```
DepthFirstSearch(graph G) {  
  για κάθε κατάσταση w  
    visited[w] = False;  
  για κάθε αρχική κατάσταση v  
    DFS(v) ;  
}
```

```
DFS(vertex v) {  
  visited[v] = True;  
  για κάθε w γειτονική της v  
    if (visited[w]==False)  
      DFS(w)  
}
```

Παράδειγμα DepthFirstSearch

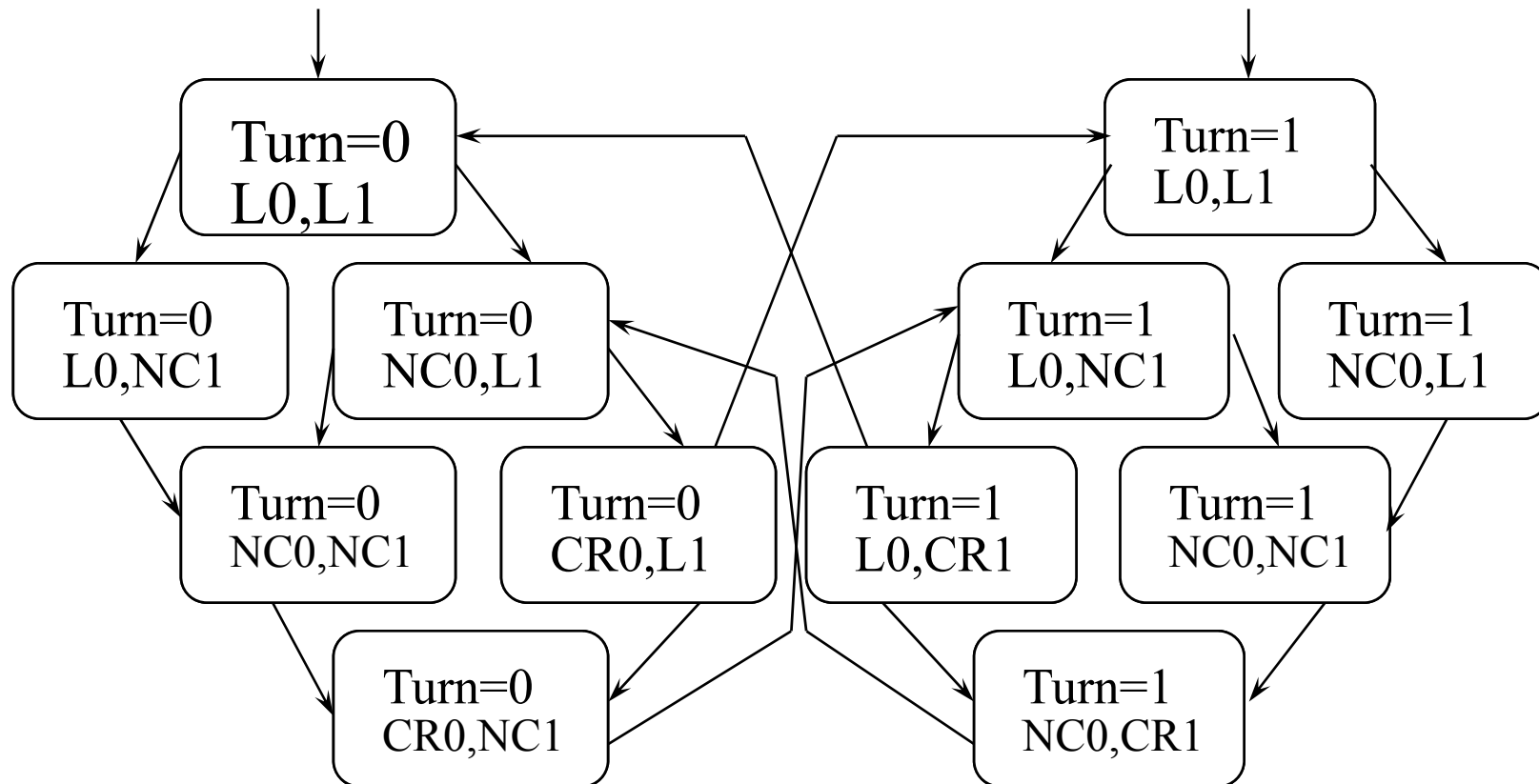


Έλεγχος ιδιοτήτων με τη DFS

- *Αμετάβλητη συνθήκη*: ελέγχουμε πως όλες οι εφικτές καταστάσεις ικανοποιούν τη ζητούμενη ιδιότητα. Διαφορετικά επιστρέφουμε μονοπάτι από την αρχική κατάσταση στην κατάσταση που παραβιάζει την ιδιότητα.
- *Ανίχνευση αδιεξόδων*: ελέγχουμε κατά πόσο υπάρχει εφικτή κατάσταση από όπου καμιά μετάβαση δεν επιτρέπεται.
- *Νεκρός κώδικας*: κατά τη DFS μαρκάρουμε όλες τις καταστάσεις από τις οποίες περνούμε. Αν υπάρχουν καταστάσεις που δεν έχουμε επισκεφθεί αντιστοιχούν σε νεκρό κώδικα.

$$G \neg (PC0 = CR0 \wedge PC1 = CR1)$$

η ιδιότητα αυτή είναι αμετάβλητη
συνθήκη του συστήματος



Μοντελοέλεγχος

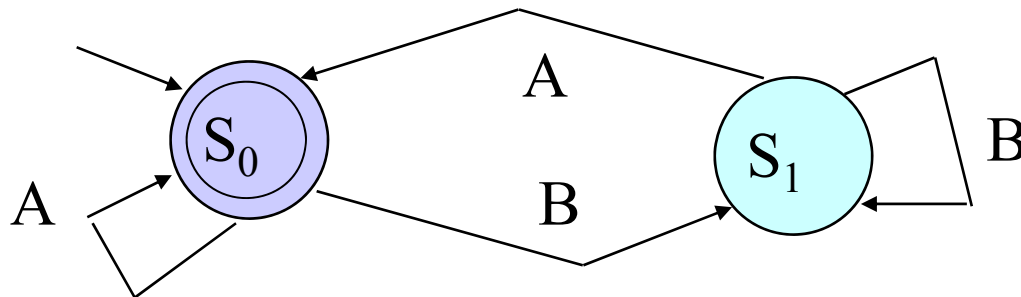
- Μας ενδιαφέρουν πολλές άλλες ιδιότητες εκτός από τις ιδιότητες τύπου G p.
- Απαιτείται αλγόριθμος ελέγχου οποιασδήποτε ιδιότητας της PLTL.
- Αυτό μπορεί να επιτευχθεί με τη *μετατροπή ιδιοτήτων χρονικής λογικής σε γράφους* (πεπερασμένα αυτόματα).
- Με αυτό τον τρόπο ο ίδιος *φορμαλισμός* χρησιμοποιείται τόσο για την μοντελοποίηση όσο και για την προδιαγραφή συστημάτων.

Αυτόματα πεπερασμένων λέξεων

- Ένα *αυτόματο* ορίζεται ως μία πλειάδα $A = \langle \Sigma, S, \Delta, I, F \rangle$ όπου
 - Σ είναι το αλφάβητο του αυτομάτου
 - S είναι το σύνολο των καταστάσεων
 - $\Delta : S \times \Sigma \times S$, είναι η σχέση μεταβάσεων
 - $I \subseteq S$, είναι το σύνολο των αρχικών καταστάσεων
 - $F \subseteq S$, είναι το σύνολο των αποδεκτών καταστάσεων

Παράδειγμα

- Έστω $M = \langle \Sigma, S, \Delta, I, F \rangle$ όπου
 - $\Sigma = \{A, B\}$,
 - $S = \{S_0, S_1\}$,
 - $\Delta = \{(S_0, A, S_0), (S_0, B, S_1), (S_1, B, S_1), (S_1, A, S_0)\}$,
 - $I = \{S_0\}$,
 - $F = \{S_0\}$.

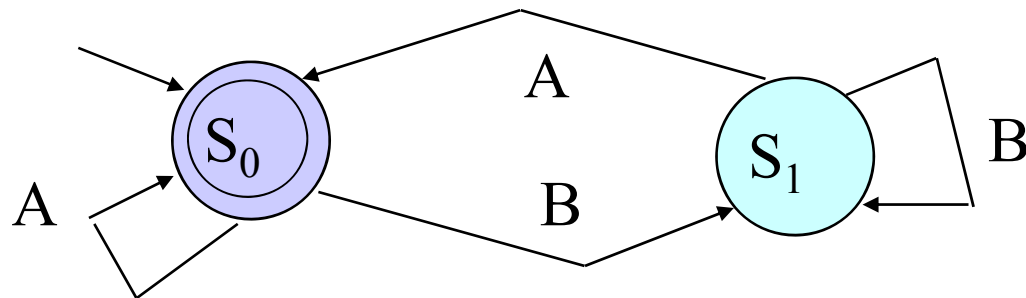


Η γλώσσα ενός αυτόματου

- **Λέξη** ενός αλφάβητου Σ ονομάζουμε μια ακολουθία γραμμάτων από το Σ , π.χ. ABAAB.
- Μία ακολουθία από καταστάσεις, π.χ. $S_0S_1S_2\dots S_n$ ονομάζεται **εκτέλεση της λέξης** $a_1a_2\dots a_n$ αν και μόνο αν
 - η S_0 είναι αρχική κατάσταση, και
 - για κάθε i , (S_{i-1}, a_i, S_i) είναι μετάβαση.Ονομάζεται **αποδεκτή εκτέλεση της λέξης** αν η S_n είναι αποδεκτή κατάσταση.
- Μία λέξη είναι **αποδεκτή** αν υπάρχει αποδεκτή εκτέλεση της λέξης από το αυτόματο.
- **Γλώσσα** του αυτομάτου είναι το σύνολο των λέξεων που είναι αποδεκτές από αυτό.

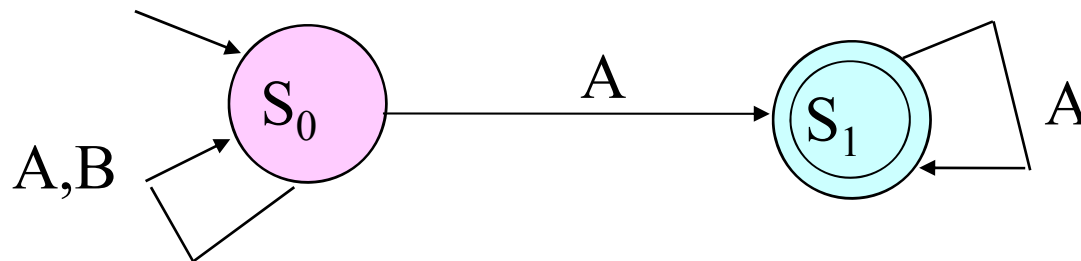
Παράδειγμα (συν.)

- Οι λέξεις AABBBBA, ABBBBBA ανήκουν στη γλώσσα του αυτομάτου, ενώ...
- Οι λέξεις ABAB, ABBB δεν ανήκουν.
- Ποια είναι η γλώσσα του αυτομάτου;

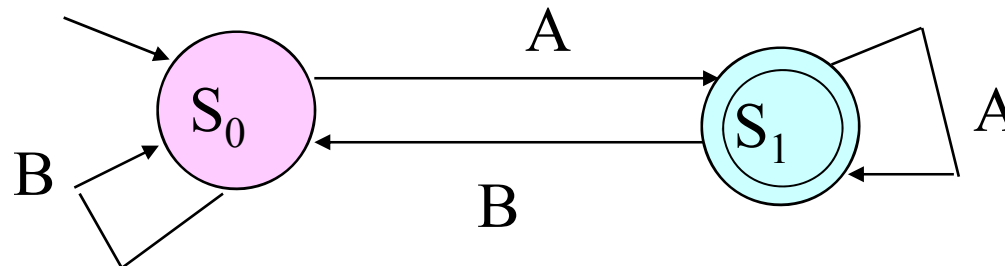


Μη ντετερμινιστικό αυτόματο

- $\Delta = \{(S_0, A, S_0), (S_0, B, S_0), (S_0, A, S_1), (S_1, A, S_1)\}$.



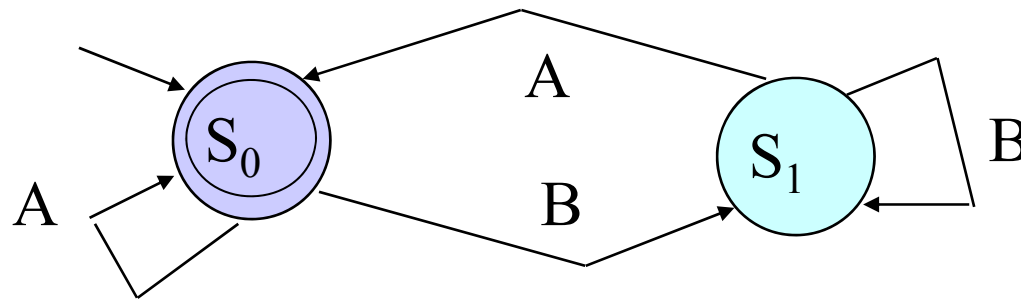
- Ποια η γλώσσα του αυτομάτου;
- Ισοδύναμο ντετερμινιστικό αυτόματο.



Αυτόματα μη-πεπερασμένων λέξεων

- Ίδιος ορισμός, με τη διαφορά ότι τώρα οι λέξεις είναι μη-πεπερασμένου μήκους.
- *Λέξη ενός αλφάβητου* Σ ονομάζουμε μια *μη-πεπερασμένη* ακολουθία γραμμάτων από το Σ .
- Μία ακολουθία από καταστάσεις, π.χ. $S_0S_1S_2\dots$ ονομάζεται *αποδεκτή εκτέλεση της λέξης* $a_1a_2\dots$ αν και μόνο αν
 - η S_0 είναι αρχική κατάσταση,
 - υπάρχει μία αποδεκτή κατάσταση η οποία παρουσιάζεται άπειρες φορές στην εκτέλεση, και
 - για κάθε i , (S_{i-1}, a_i, S_i) είναι μετάβαση.
- Μία λέξη είναι *αποδεκτή* αν υπάρχει εκτέλεση της λέξης από το αυτόματο.

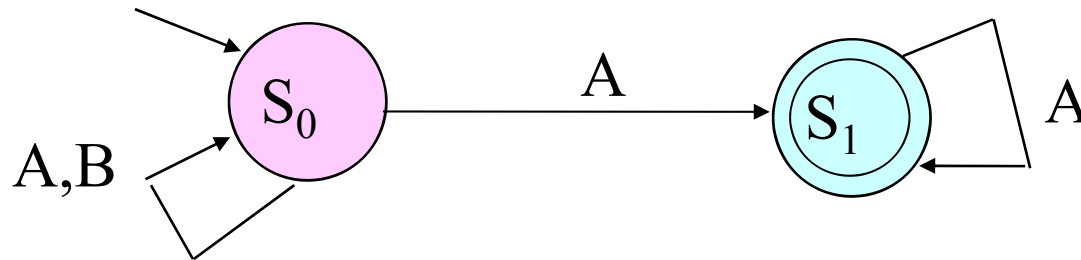
Παράδειγμα



- Έστω η λέξη $A B A B A B A B A B \dots$ και η εκτέλεση της λέξης $S_0 S_1 S_0 S_1 S_0 S_1 \dots$. Η εκτέλεση είναι αποδεκτή αφού η κατάσταση S_0 παρουσιάζεται άπειρες φορές στην εκτέλεση.
- Η λέξη $B B B B B B \dots$ δεν ανήκει στη γλώσσα του αυτομάτου αφού η εκτέλεση της λέξης $S_0 S_1 S_1 S_1 S_1 S_1 \dots$ δεν είναι αποδεκτή εκτέλεση.
- Η λέξη $A B A B B A B B B \dots$;

Μη ντετερμινιστικό αυτόματο μη πεπερασμένων λέξεων

- Ποια η γλώσσα του αυτομάτου;

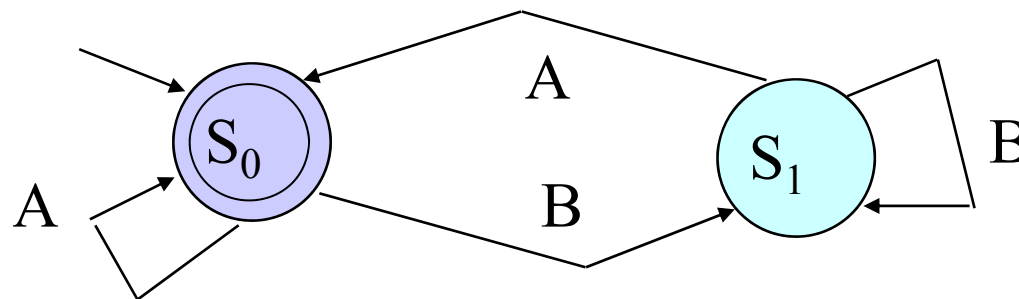


- Έστω
 - $B \equiv PC_0 = CR_0$, και
 - $A \equiv \neg B$

Ποια είναι η LTL προδιαγραφή που αντιστοιχεί στο πιο πάνω αυτόματο;

Προδιαγραφή με αυτόματα

- Αναθέτουμε σε κάθε γράμμα του αλφάβητου κάποια ατομική πρόταση.
- Για παράδειγμα:
A : η διεργασία P_0 βρίσκεται στην κρίσιμη περιοχή, δηλ. $PC_0 = CR_0$
B : η διεργασία P_0 δεν βρίσκεται στην κρίσιμη περιοχή, δηλ. $\neg PC_0 = CR_0$
- Η ιδιότητα $G F (PC_0 = CR_0)$ μεταφράζεται ως το αυτόματο



Προδιαγραφή με αυτόματα

Αμοιβαίος αποκλεισμός

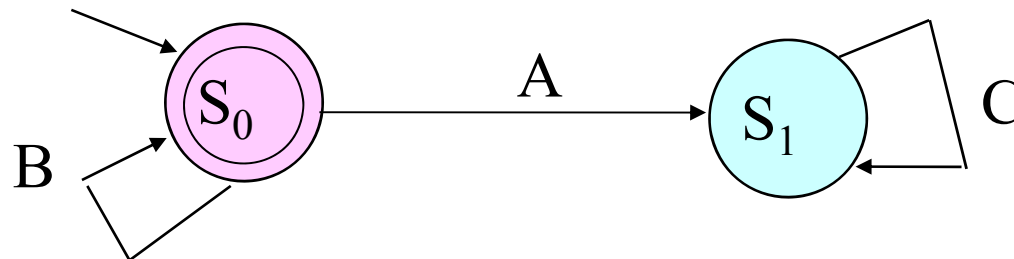
- Έστω

$$A \equiv PC_0=CR_0 \wedge PC_1=CR_1$$

$$B \equiv \neg(PC_0=CR_0 \wedge PC_1=CR_1)$$

$$C \equiv \text{TRUE}$$

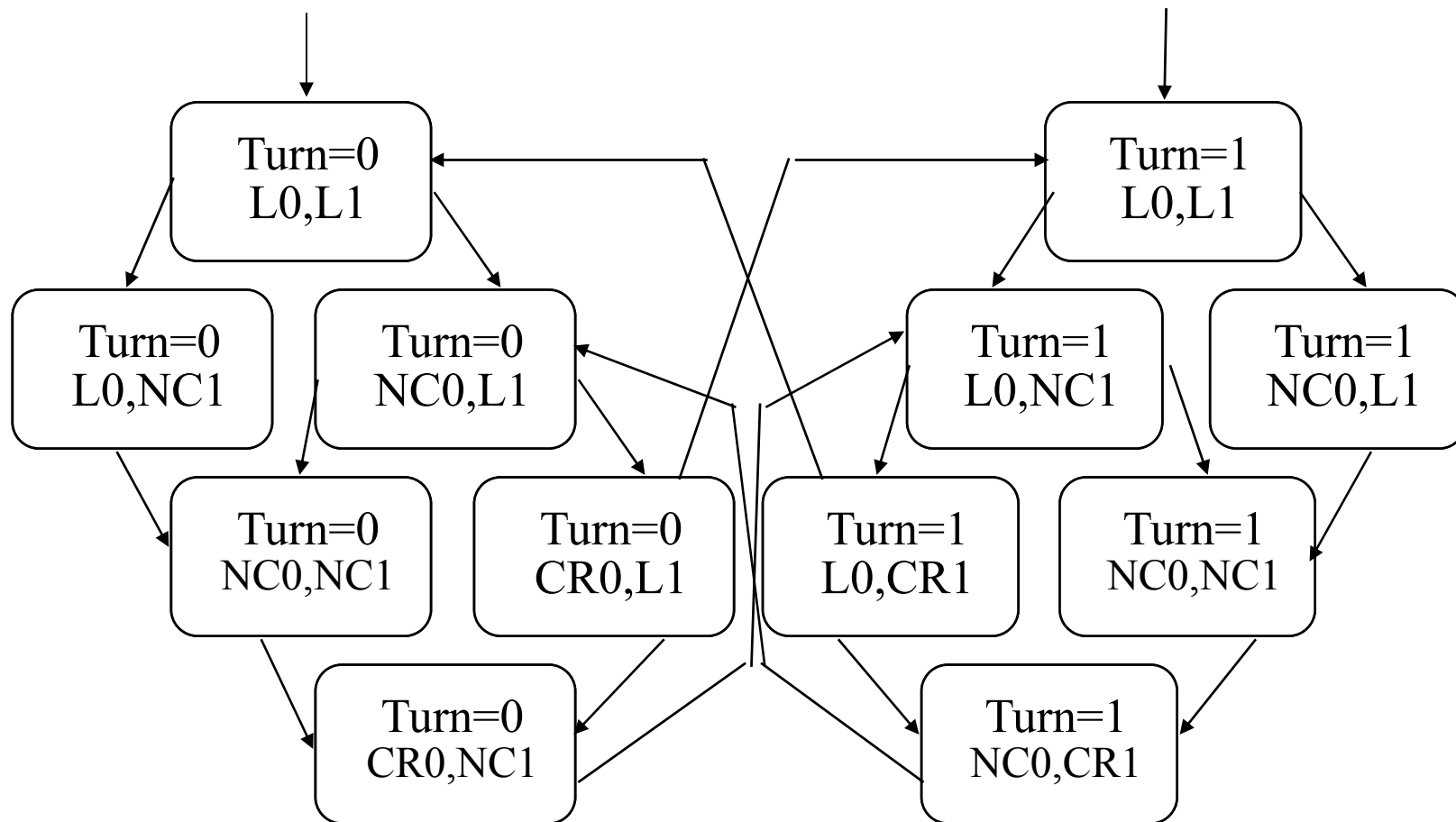
- Το πιο κάτω αυτόματο εκφράζει την ιδιότητα $G\neg(PC_0=CR_0 \wedge PC_1=CR_1)$



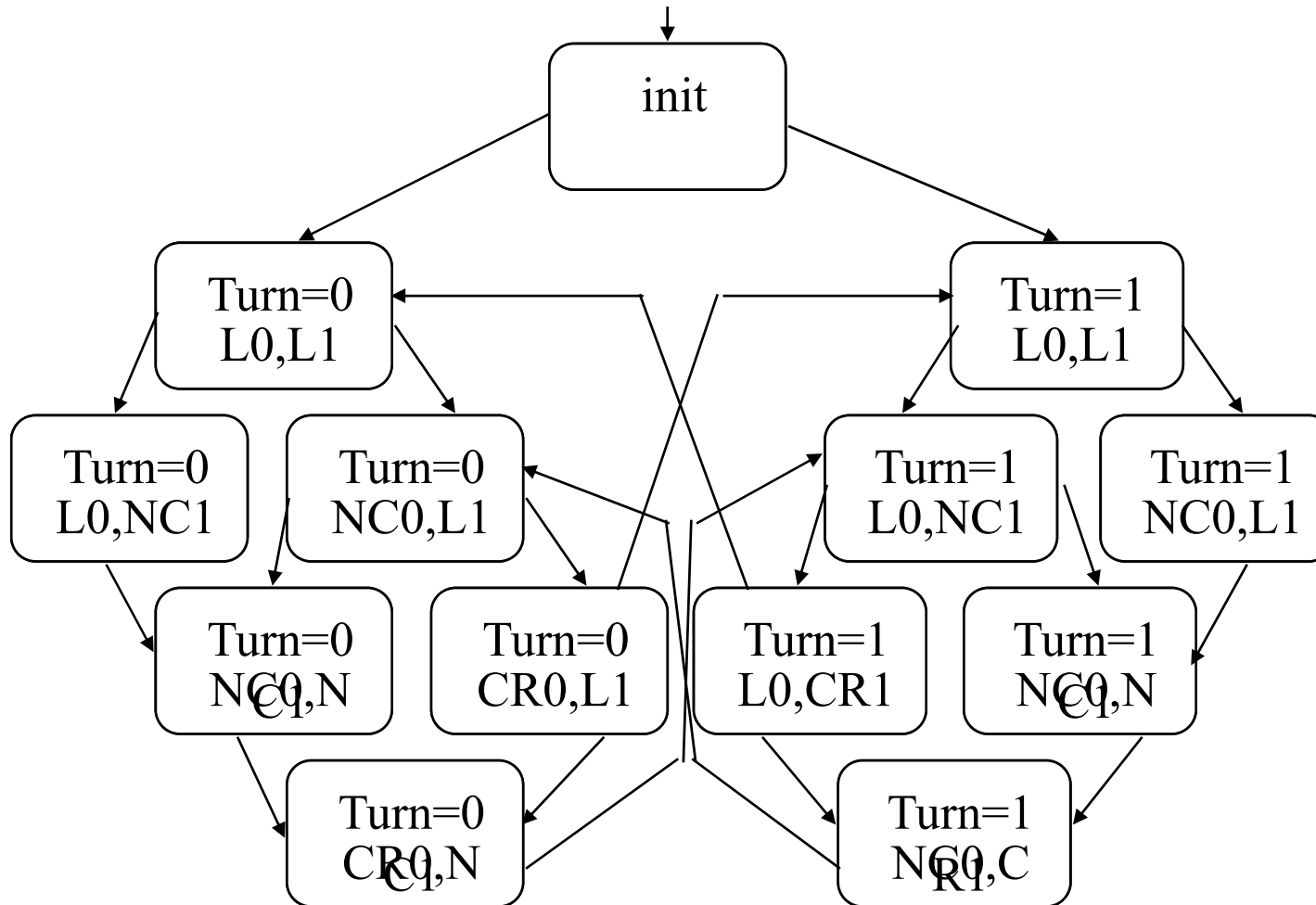
Μοντελοποίηση με αυτόματα

- Από το σύστημα μεταβάσεων
 1. Προσθέτουμε μία νέα αρχική κατάσταση την οποία συνδέουμε με τις παλιές αρχικές καταστάσεις.
 2. Μεταφέρουμε τις ατομικές προτάσεις των καταστάσεων στις ακμές που δείχνουν προς αυτές.
 3. Θέτουμε όλες τις καταστάσεις ως αποδεκτές.

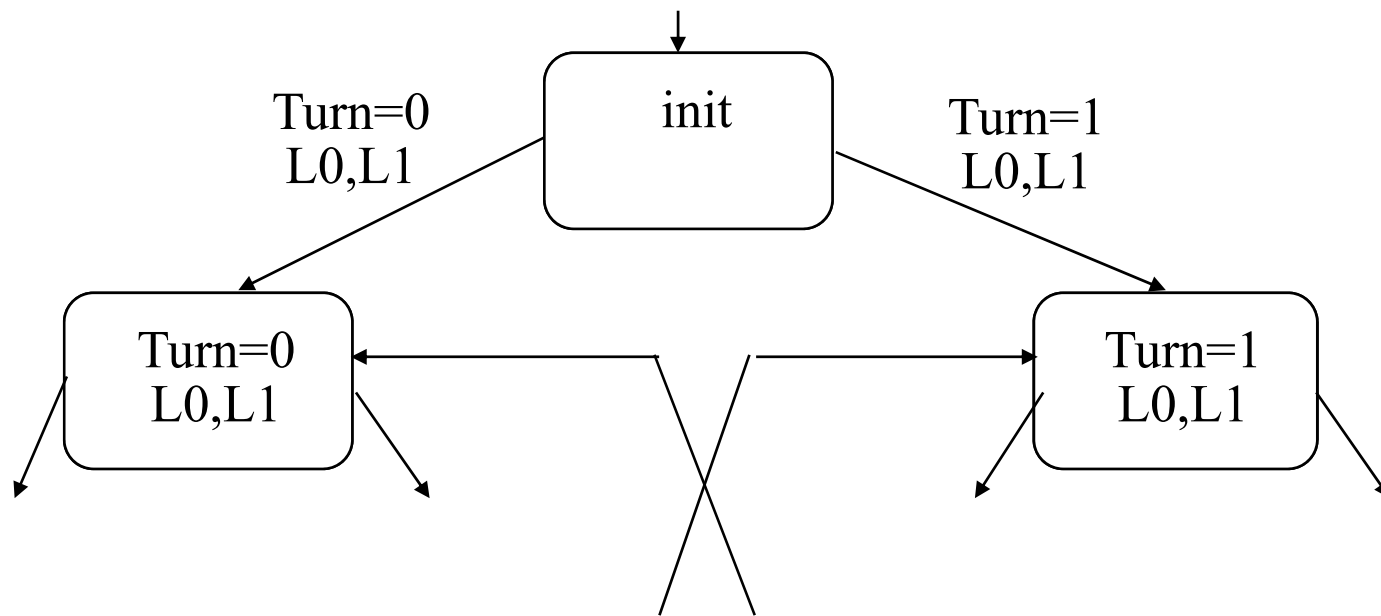
Παράδειγμα



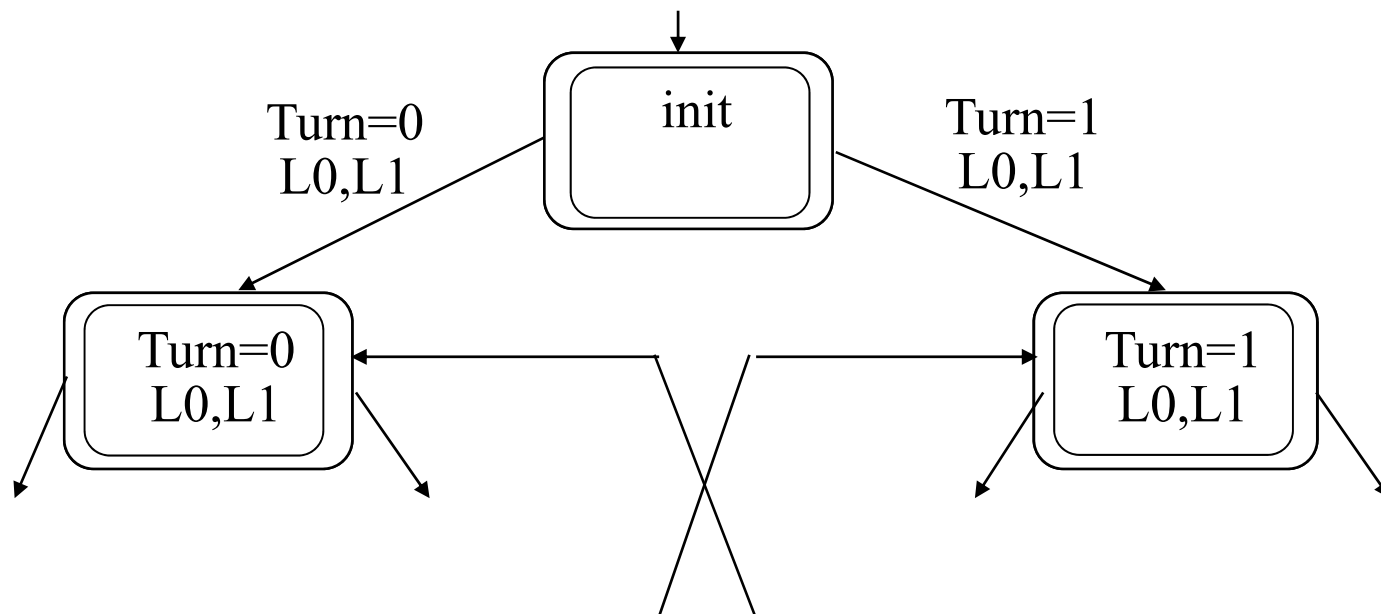
Βήμα 1



Βήμα 2



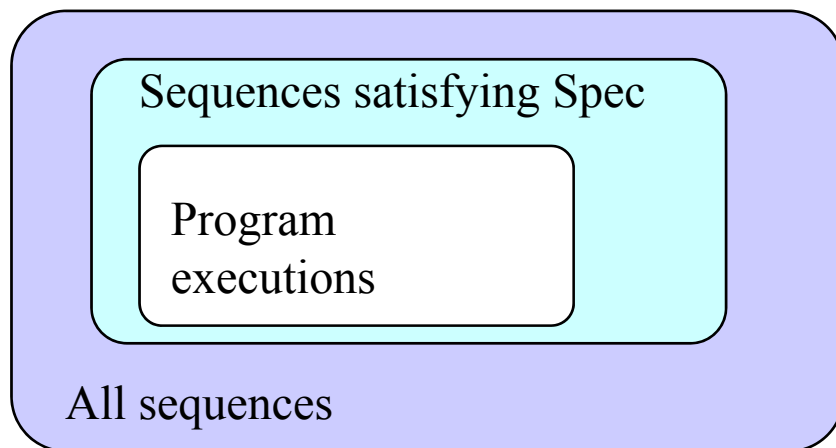
Βήμα 3



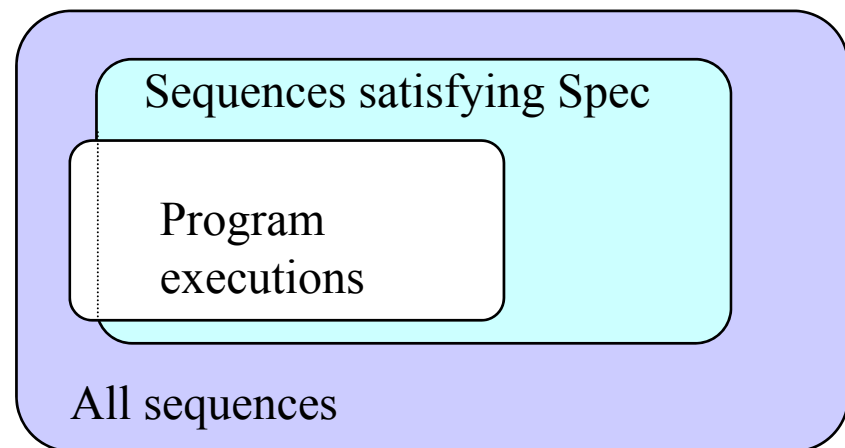
Συνθήκη ορθότητας

- Πότε ένα μοντέλο ικανοποιεί την προδιαγραφή του;
- Έστω $L(\text{Model})$ η γλώσσα του μοντέλου.
- Έστω $L(\text{Spec})$ η γλώσσα της προδιαγραφής
- Πρέπει να ισχύει ότι : $L(\text{Model}) \subseteq L(\text{Spec})$.

ΟΡΘΟΤΗΤΑ



Λανθασμένη συμπεριφορά



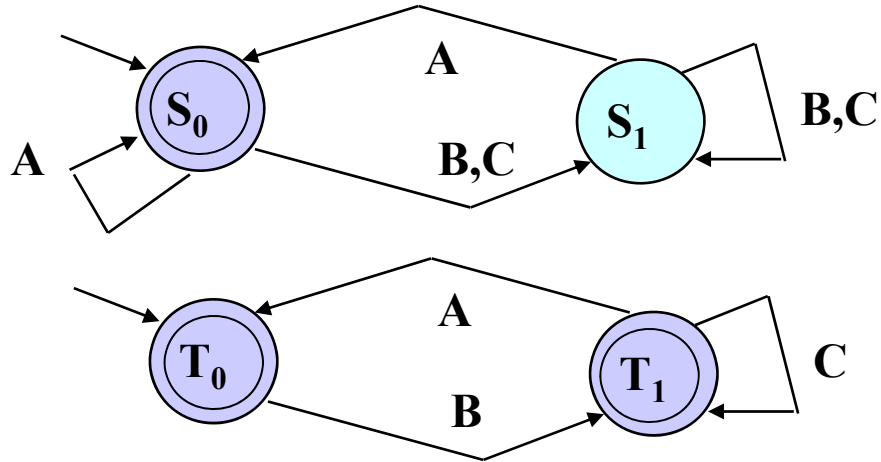
Απόδειξη της ορθότητας

- Είναι αρκετό να δείξουμε ότι $L(\text{Model}) \subseteq L(\text{Spec})$
ή εναλλακτικά, ότι,
 $L(\text{Model}) \cap \overline{L(\text{Spec})} = \emptyset$,
όπου $\overline{L(\text{Spec})}$ είναι το συμπλήρωμα του αυτομάτου Spec, δηλαδή οι λέξεις που δεν λαμβάνονται από αυτό
- Για να το πράξουμε πρέπει να γνωρίζουμε
 - πώς να υπολογίζουμε την **τομή** δύο αυτομάτων, **δηλαδή το αυτόματο που αποδέχεται μόνο τις λέξεις που είναι αποδεκτές και από τα δύο αυτόματα**
 - πώς να υπολογίζουμε το **συμπλήρωμα** ενός αυτομάτου, **δηλαδή το αυτόματο που αποδέχεται μόνο τις λέξεις που δεν είναι αποδεκτές από το αυτόματο**
 - πώς να μεταφράζουμε ιδιότητες της LTL σε αυτόματα

Τομή δύο αυτομάτων

- Έστω δύο αυτόματα
 $A_1 = \langle \Sigma, S_1, \Delta_1, I_1, F_1 \rangle$ και $A_2 = \langle \Sigma, S_2, \Delta_2, I_2, F_2 \rangle$
- Ορίζουμε ως τομή των δύο αυτομάτων το αυτόματο
 $A = \langle \Sigma, S, \Delta, I, F \rangle$ όπου
 - καταστάσεις του A είναι τα ζεύγη (x, y) όπου x είναι κατάσταση του S_1 και y είναι κατάσταση του S_2 .
 - αρχικές καταστάσεις του A είναι τα ζεύγη (x, y) όπου x είναι αρχική κατάσταση στο I_1 και y είναι αρχική κατάσταση στο I_2 .
 - αποδεκτές καταστάσεις του A είναι τα ζεύγη (x, y) όπου x είναι αποδεκτή κατάσταση στο F_1 και y είναι αποδεκτή κατάσταση στο F_2 .
 - $((x, y) \xrightarrow{a} (x', y'))$ είναι μετάβαση του A αν (x, a, x') είναι μετάβαση του A_1 και (y, a, y') είναι μετάβαση του A_2 .

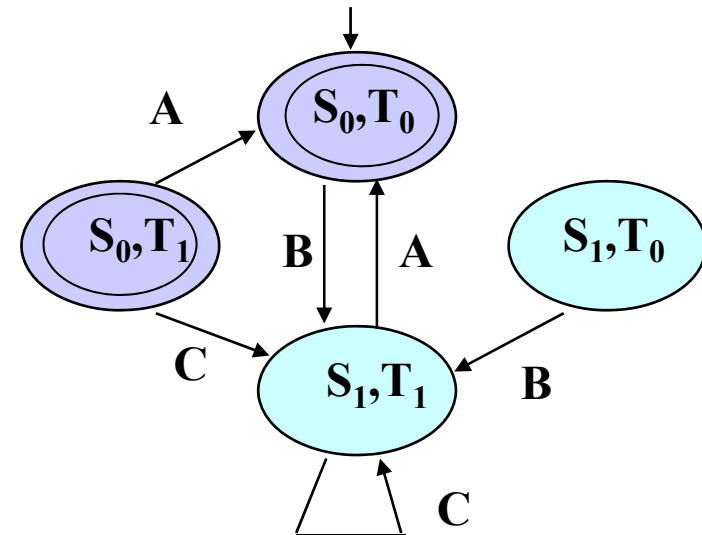
Παράδειγμα



Καταστάσεις: (S_0, T_0) , (S_0, T_1) ,
 (S_1, T_0) , (S_1, T_1) .

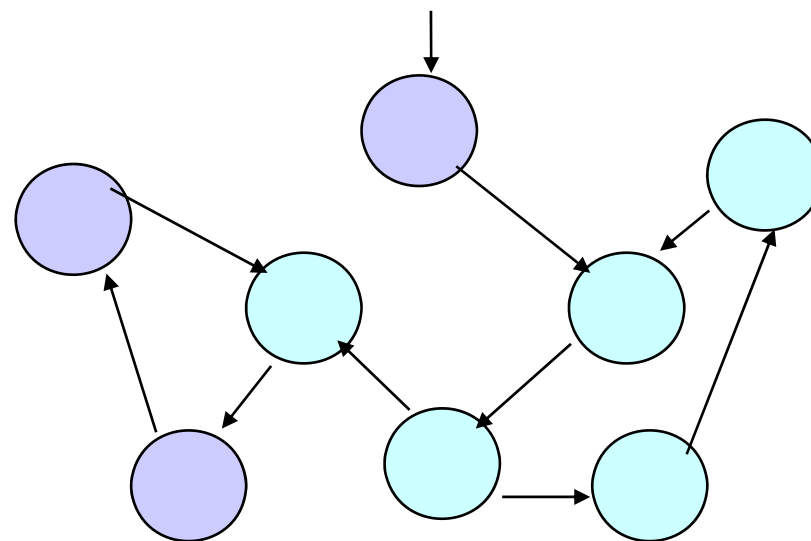
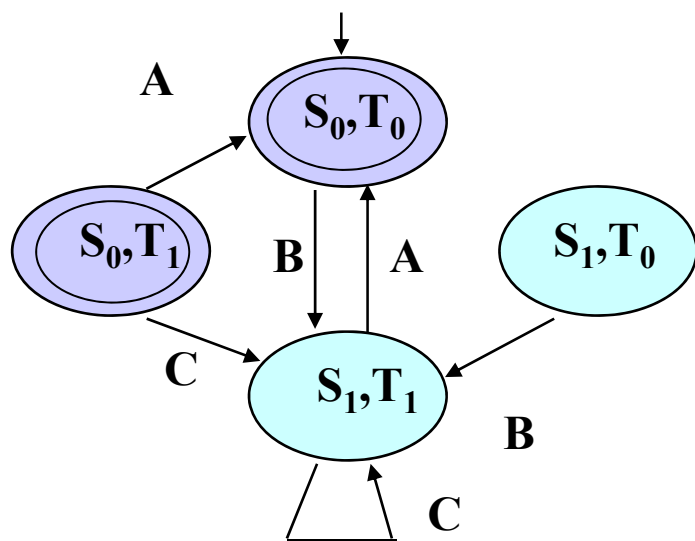
Αποδεκτές: (S_0, T_0) , (S_0, T_1) .

Αρχική: (S_0, T_0) .



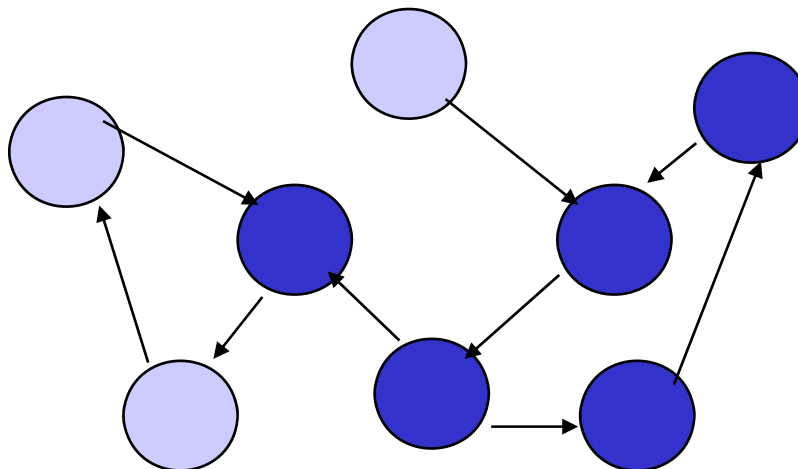
Έλεγχος κενής γλώσσας

- Πρέπει να ελέγξουμε κατά πόσο υπάρχει αποδεκτή εκτέλεση (που περνά από αποδεκτές καταστάσεις άπειρες φορές).



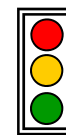
Εύρεση αποδεκτών μονοπατιών

- Αν υπάρχει αποδεκτή εκτέλεση, τότε υπάρχει τουλάχιστον μια αποδεκτή κατάσταση η οποία επαναλαμβάνεται άπειρες φορές. Αυτή η κατάσταση πρέπει να εμφανίζεται σε κύκλο. Επομένως το πρόβλημα μας μεταφράζεται στην εύρεση κύκλου που περιέχει μία (τουλάχιστον) αποδεκτή κατάσταση.
- Ισοδύναμα, θέλουμε να βρούμε ένα εφικτό ισχυρά συνδεδεμένο υπογράφο με αποδεκτή κατάσταση (Ισχυρά συνδεδεμένος υπογράφος είναι ένα σύνολο από κόμβους όπου υπάρχει μονοπάτι από κάθε κόμβο προς κάθε άλλο κόμβο).



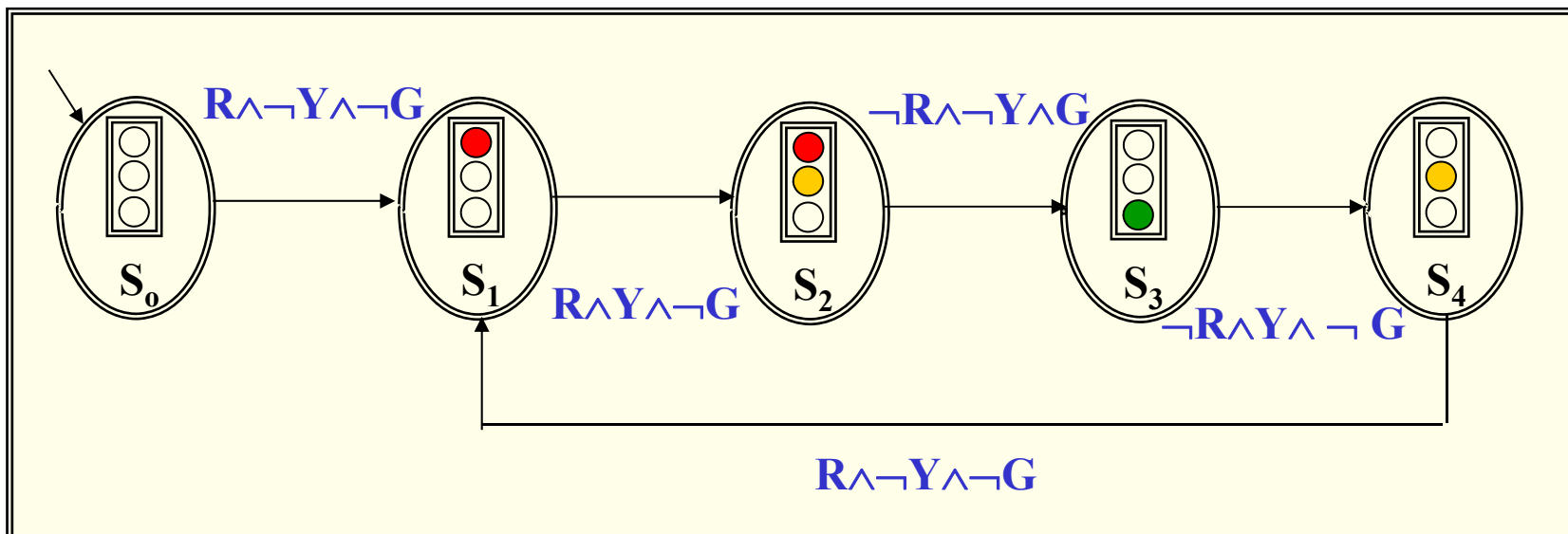
Συμπλήρωμα αυτομάτου

- Η εύρεση του συμπληρώματος ενός αυτομάτου είναι δύσκολη.
- Εναλλακτικά μπορούμε να ξεκινήσουμε με την άρνηση της ζητούμενης ιδιότητας και να τη μετατρέψουμε κατ'ευθείαν στο αυτόματο που μας ενδιαφέρει (το συμπλήρωμα του αυτομάτου που εκφράζει τη γλώσσα με τη ζητούμενη ιδιότητα).

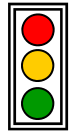


Παράδειγμα – Τα φώτα της τροχαίας

- Έστω το πιο κάτω μοντέλο φώτων της τροχαίας



- Θέλουμε να αποδείξουμε, ότι κάθε φορά που το κίτρινο φως είναι αναμμένο μόνο του, μετά θα ακολουθήσει το κόκκινο φως. Δηλαδή, θέλουμε να αποδείξουμε την ιδιότητα χρονικής λογικής $G (\neg R \wedge Y \Rightarrow X R)$

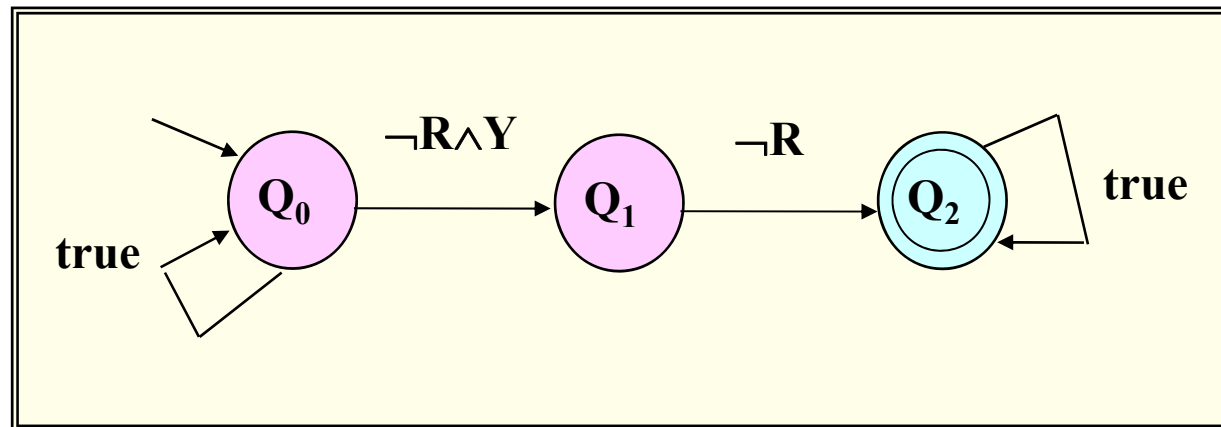


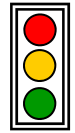
Η άρνηση της ιδιότητας

- Θα κτίσουμε το αυτόματο της *άρνησης* της ζητούμενης ιδιότητα, δηλαδή, της ιδιότητας

$$\begin{aligned}\neg G (\neg R \wedge Y \Rightarrow X R) &\equiv F \neg (\neg R \wedge Y \Rightarrow X R) \\ &\equiv F ((\neg R \wedge Y) \wedge (\neg X R)) \\ &\equiv F ((\neg R \wedge Y) \wedge X (\neg R))\end{aligned}$$

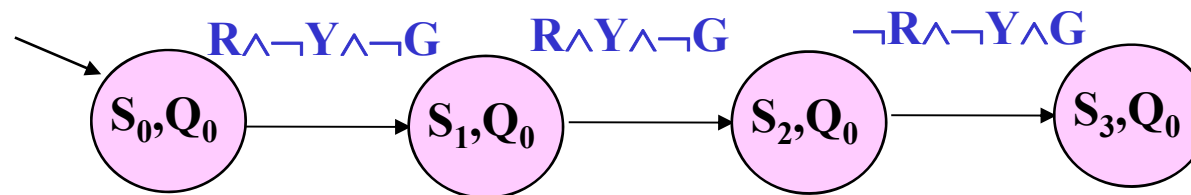
- Η οποία μεταφράζεται ως το πιο κάτω αυτόματο

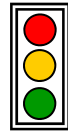




Το γινόμενο ιδιότητας-μοντέλου

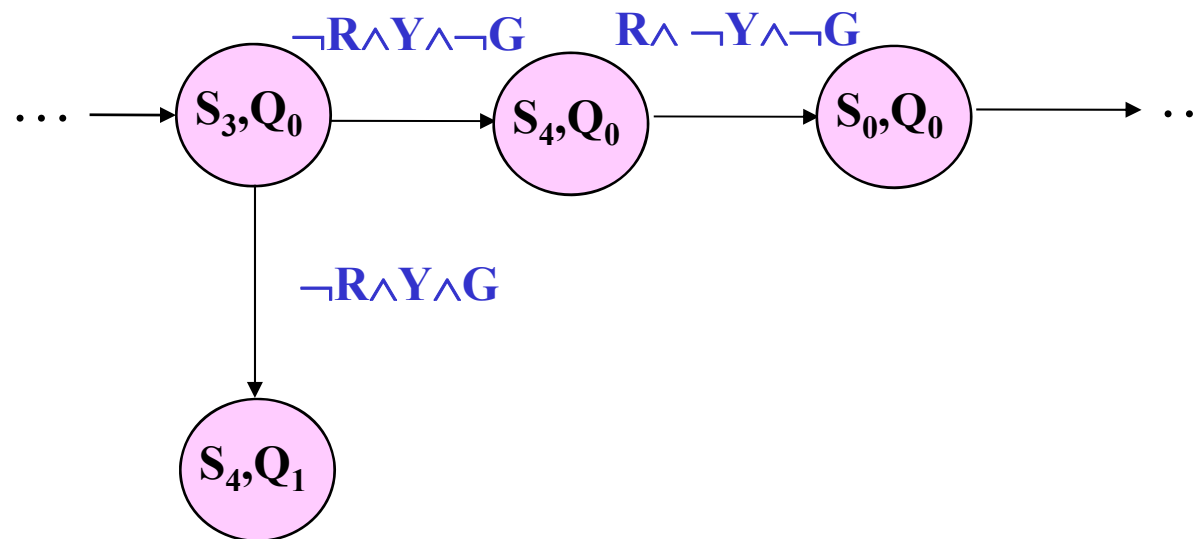
- Σύνολο των καταστάσεων του γινομένου των δύο αυτομάτων είναι όλες οι καταστάσεις (S_i, Q_j) , όπου $0 \leq i \leq 4$, και $0 \leq j \leq 2$.
- Αρχική κατάσταση είναι η (S_0, Q_0) , και αποδεκτές καταστάσεις οι (S_i, Q_2) , για κάθε $0 \leq i \leq 4$.
- Απομένει να υπολογίσουμε τις μεταβάσεις του γινομένου. Ξεκινώντας από την αρχική κατάσταση έχουμε:



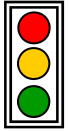


Το γινόμενο ιδιότητας-μοντέλου

- Από την κατάσταση (S_3, Q_0) υπάρχουν δύο εναλλακτικές μεταβάσεις:

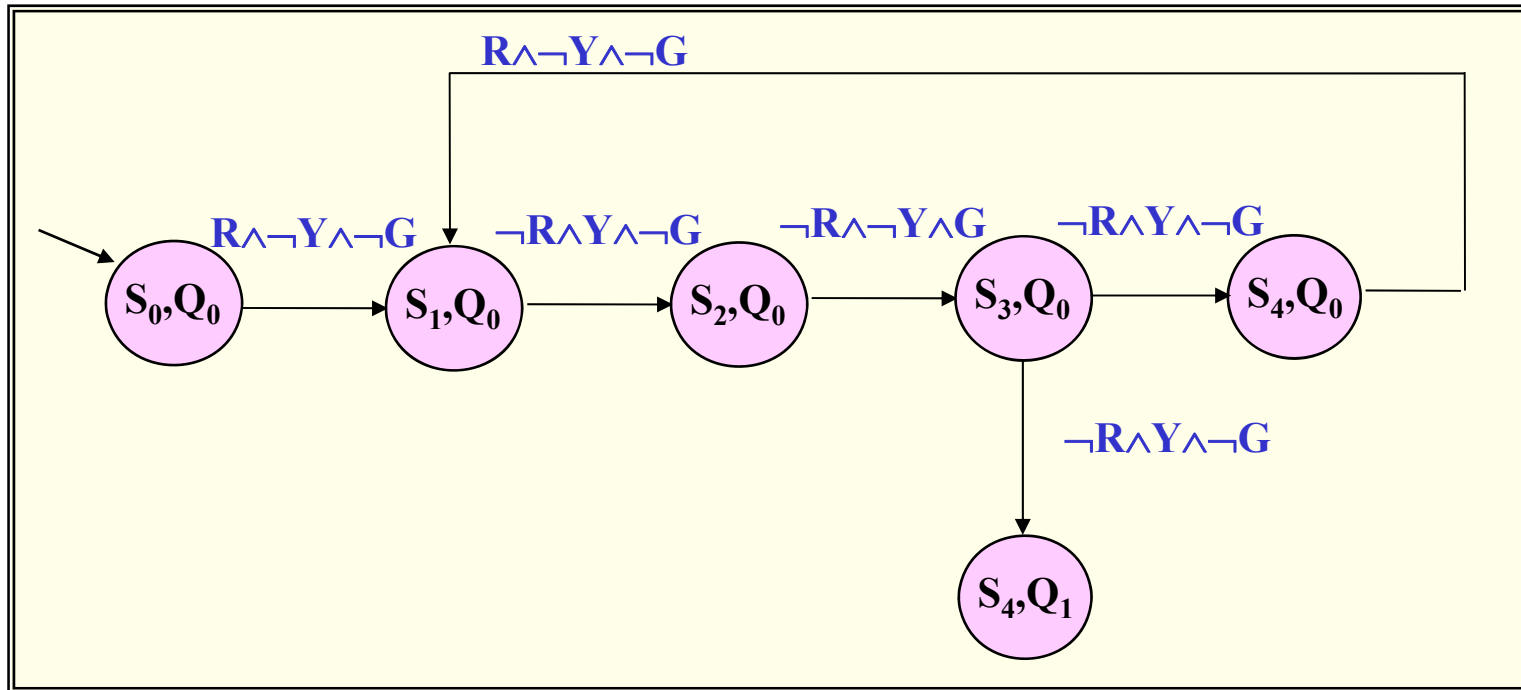


- Η μία επιλογή επιστρέφει στην αρχική κατάσταση (S_0, Q_0) .
- Η άλλη οδηγεί στην κατάσταση (S_4, Q_1) που αποτελεί αδιέξοδο αφού μετάβαση του S_4 απαιτεί την πρόταση R (δηλαδή ύπαρξη κόκκινου φωτός) ενώ μετάβαση από την κατάσταση Q_1 απαιτεί την αντίθετη πρόταση $\neg R$ (δηλαδή την μη ύπαρξη κόκκινου φωτός).



Έλεγχος κενής γλώσσας

- Επομένως το γινόμενο αυτόματο είναι το πιο κάτω



- Αφού το αυτόματο δεν περιέχει αποδεκτή κατάσταση, η γλώσσα του είναι κενή, και, επομένως, η ζητούμενη ιδιότητα ικανοποιείται.