
Χρονικά αυτόματα

Στην ενότητα αυτή θα μελετηθούν τα εξής θέματα:

Συστήματα πραγματικού Χρόνου

Διακριτός και συνεχής χρόνος

Χρονικά αυτόματα

Χρονική CTL

Συστήματα πραγματικού χρόνου

- Η χρονική συμπεριφορά είναι κρίσιμης ασφάλειας σε πολλά συστήματα, όπως:
 - Διασταυρώσεις τραίνων
 - Ελεγκτές ρομπότ
 - Ελεγκτές προσγείωσης αεροπλάνων
 - Πρωτόκολλα επικοινωνίας

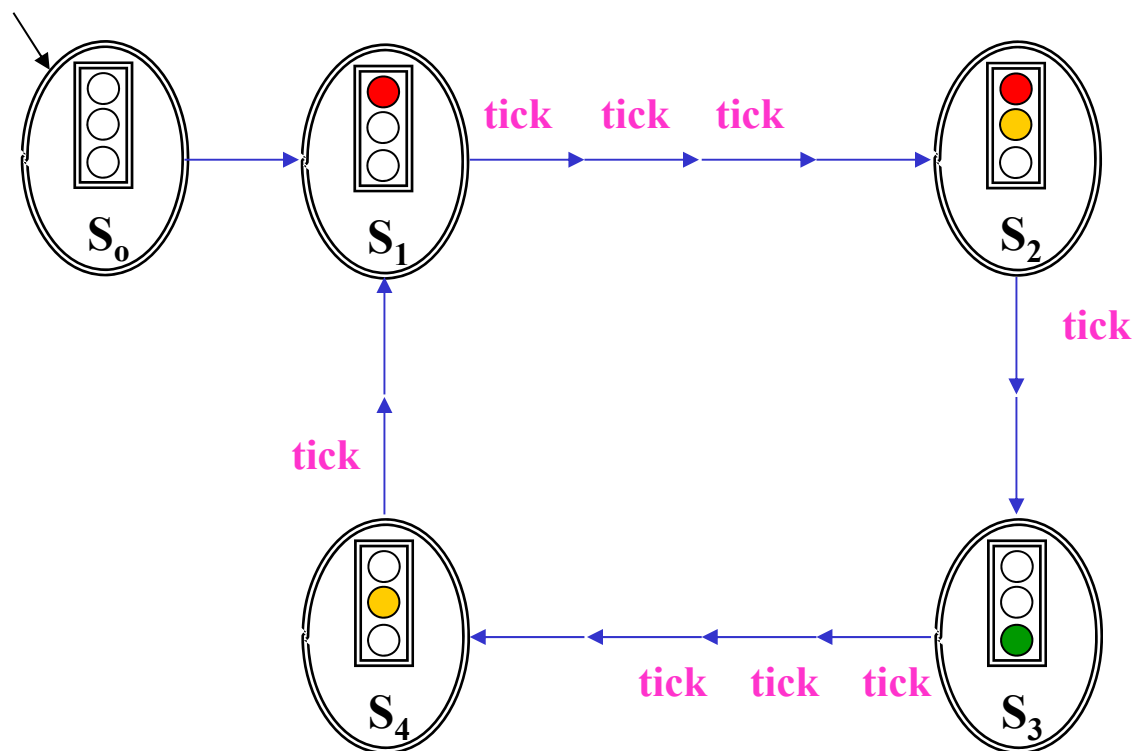
Συστήματα πραγματικού χρόνου ονομάζονται συστήματα των οποίων η ορθότητα καθορίζεται τόσο από τις αλληλεπιδράσεις με το περιβάλλον όσο και από τον χρόνο στον οποίο παρουσιάζονται οι αλληλεπιδράσεις αυτές.

- Θα μελετήσουμε μεθόδους εισαγωγής της έννοιας του χρόνου σε πρότυπα μοντελοποίησης και ανάλυσης συστημάτων.

Ο χρόνος ως ένα διακριτό πεδίο

- *Διακριτός χρόνος*: ο χρόνος μετριέται μέσω διακριτών τιμών και αυξάνεται μέσω διακριτών βημάτων.
- Μοντελοποιείται μέσω θετικών ακεραίων.
- Συγκεκριμένα χρονικά γεγονότα – tick – χρησιμοποιούνται για να μοντελοποιήσουν το πέρασμα μιας μονάδας χρόνου.
- Γεγονότα μπορούν να συμβούν σε ακέραιες χρονικές στιγμές.
- *Η διαφορά χρόνου ανάμεσα σε δύο γεγονότα είναι πάντα πολλαπλάσιο της βασικής μονάδας χρόνου.*

Φώτα της τροχαίας με διακριτό χρόνο



Διακριτός Χρόνος

- Πλεονέκτημα: απλότητα
 - Η επέκταση αφορά μόνο μία καινούρια ενέργεια για να μετρά τον χρόνο στο μοντέλο.
 - Η χρονική λογική μπορεί να μείνει ανεπηρέαστη αφού ο τελεστής X “μετρά” το πέρασμα του χρόνου:

$G(\text{red} \Rightarrow X \ X \ X \text{yellow})$

Πάντα ισχύει ότι 3 μονάδες χρόνου *μετά* από το κόκκινο το φως θα γίνει κίτρινο.

$G(\text{red} \Rightarrow (X \text{green} \vee X \ X \text{green} \vee X \ X \ X \text{green} \vee X \ X \ X \ X \text{green}))$

Πάντα ισχύει ότι *μέσα* σε 4 μονάδες χρόνου από το κόκκινο το φως θα γίνει πράσινο.

- Πολλές εφαρμογές: συστήματα των οποίων οι διεργασίες συγχρονίζονται βάσει ενός καθολικού ρολογιού (π.χ. συγχρονισμένα κυκλώματα υλικού).

Διακριτός Χρόνος

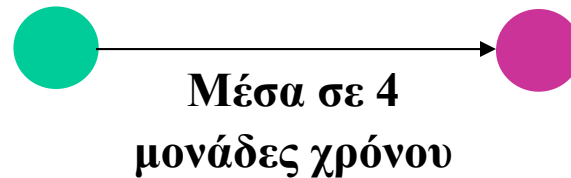
- Περιορισμός: Πως καθορίζεται η διάρκεια της βασικής μονάδας χρόνου;
 - Έτσι ώστε κάθε διαφορά στους χρόνους κάθε ζεύγους γεγονότων να είναι πολλαπλάσιο αυτής.
- Στην πράξη αυτό είναι δύσκολο/αδύνατο να καθοριστεί, και οποιαδήποτε παραδοχή περιορίζει την ακρίβεια του μοντέλου.
- Ακατάλληλο για ασύγχρονα συστήματα, όπως κατανεμημένα συστήματα, πρωτόκολλα επικοινωνίας, κλπ.

Ο χρόνος ως ένα συνεχές πεδίο

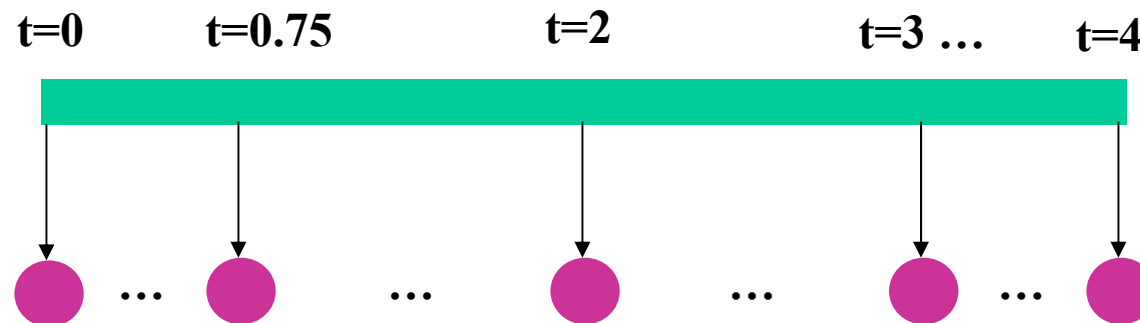
- *Συνεχής χρόνος*: ο χρόνος αυξάνεται συνεχόμενα.
- Μοντελοποιείται μέσω θετικών πραγματικών αριθμών.
- Γεγονότα μπορούν να συμβούν σε οποιαδήποτε χρονική στιγμή, και η διαφορά στους χρόνους των εμφανίσεων δύο γεγονότων μπορεί να είναι αυθαίρετα μικρή.
- Καλύτερη μοντελοποίηση της πραγματικότητας, και κατάλληλη για ασύγχρονα συστήματα.
- Επίσης, πιο πολύπλοκη!

Συνεχής χρόνος

- Αν ο χρόνος είναι συνεχής, τότε αλλαγή κατάστασης μπορεί να συμβεί σε οποιαδήποτε στιγμή.
- Για παράδειγμα η ενέργεια



μοντελοποιείται ως



Συνεχής χρόνος

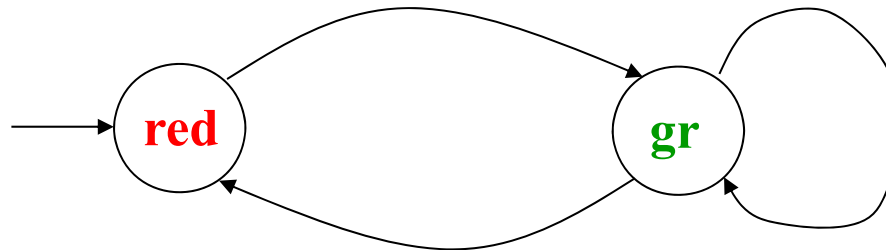
- Πως μπορούμε να ελέγξουμε ιδιότητες όπως:

“όταν το φως είναι κίτρινο τελικά θα γίνει πράσινο μέσα σε 3 μονάδες χρόνου” ;

- Για αντιμετώπιση των προβλημάτων που προκύπτουν από τη χρήση συνεχούς χρόνου έχουν γίνει οι εξής προτάσεις:
 - Περιορισμός της γλώσσας ανάλυσης συστημάτων (χρονικής λογικής)
 - Η μοντελοποίηση συστημάτων πραγματικού χρόνου γίνεται συμβολικά και όχι ρητά [Χρονικά αυτόματα - Alur & Dill, 1989]
 - Υλοποίηση μιας διακριτής μορφής του μη-πεπερασμένου συστήματος καταστάσεων σύμφωνα με το μοντέλο και την ιδιότητα που θέλουμε να ελέγξουμε. [region graphs - Alur & Dill, 1991]

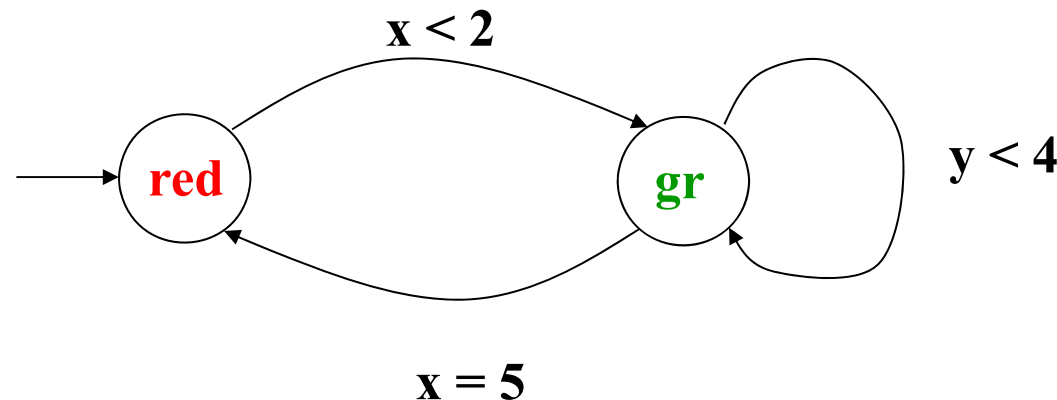
Χρονικά Αυτόματα

- Ένα *χρονικό αυτόματο* αποτελείται από *καταστάσεις* και *ακμές*.
- Κάθε κατάσταση συνοδεύεται από ατομικές προτάσεις που ικανοποιούνται στη συγκεκριμένη κατάσταση.
- Κάθε ακμή μοντελοποιεί μια δυνατή εξέλιξη από μια κατάσταση σε μια άλλη. Οι μεταβάσεις δεν παίρνουν χρόνο.



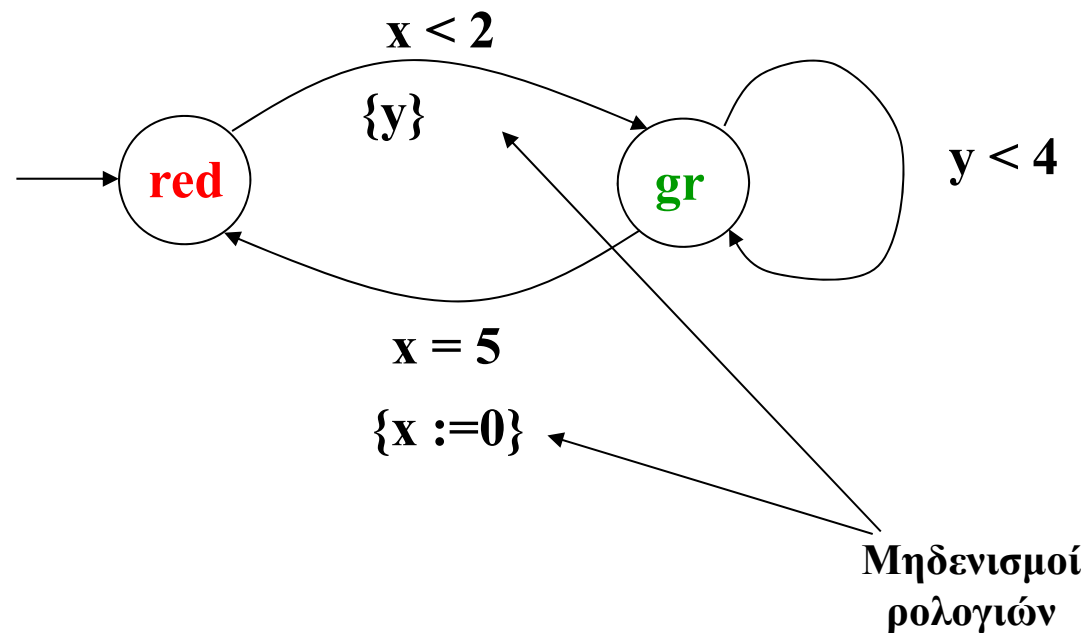
Χρονικά Αυτόματα

- Ένα χρονικό αυτόματο είναι εφοδιασμένο με ένα σύνολο από *ρολόγια*, $x, y, z, \dots$, τα οποία παίρνουν πραγματικές τιμές, και αυξάνουν τις τιμές τους με την ίδια ταχύτητα.
- Οι μεταβάσεις ενός αυτομάτου ελέγχονται από περιορισμούς στις τιμές των ρολογιών, οι οποίοι ονομάζονται *φρουροί των μεταβάσεων*.



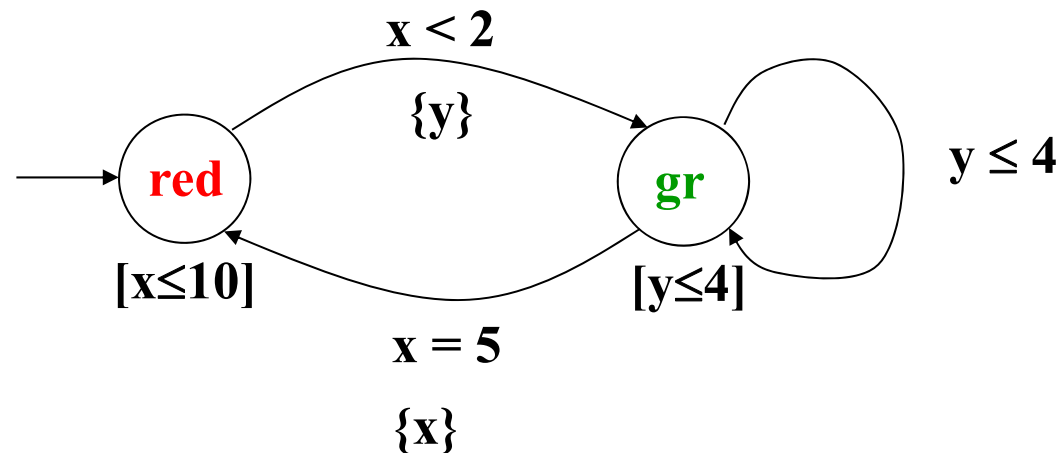
Χρονικά Αυτόματα

- Οι τιμές των ρολογιών μπορούν να **μηδενιστούν** κατά την εκτέλεση μιας μετάβασης.
- Τα ρολόγια έχουν αρχική τιμή 0, και ξεκινούν ταυτόχρονα κατά την είσοδο στην αρχική κατάσταση.

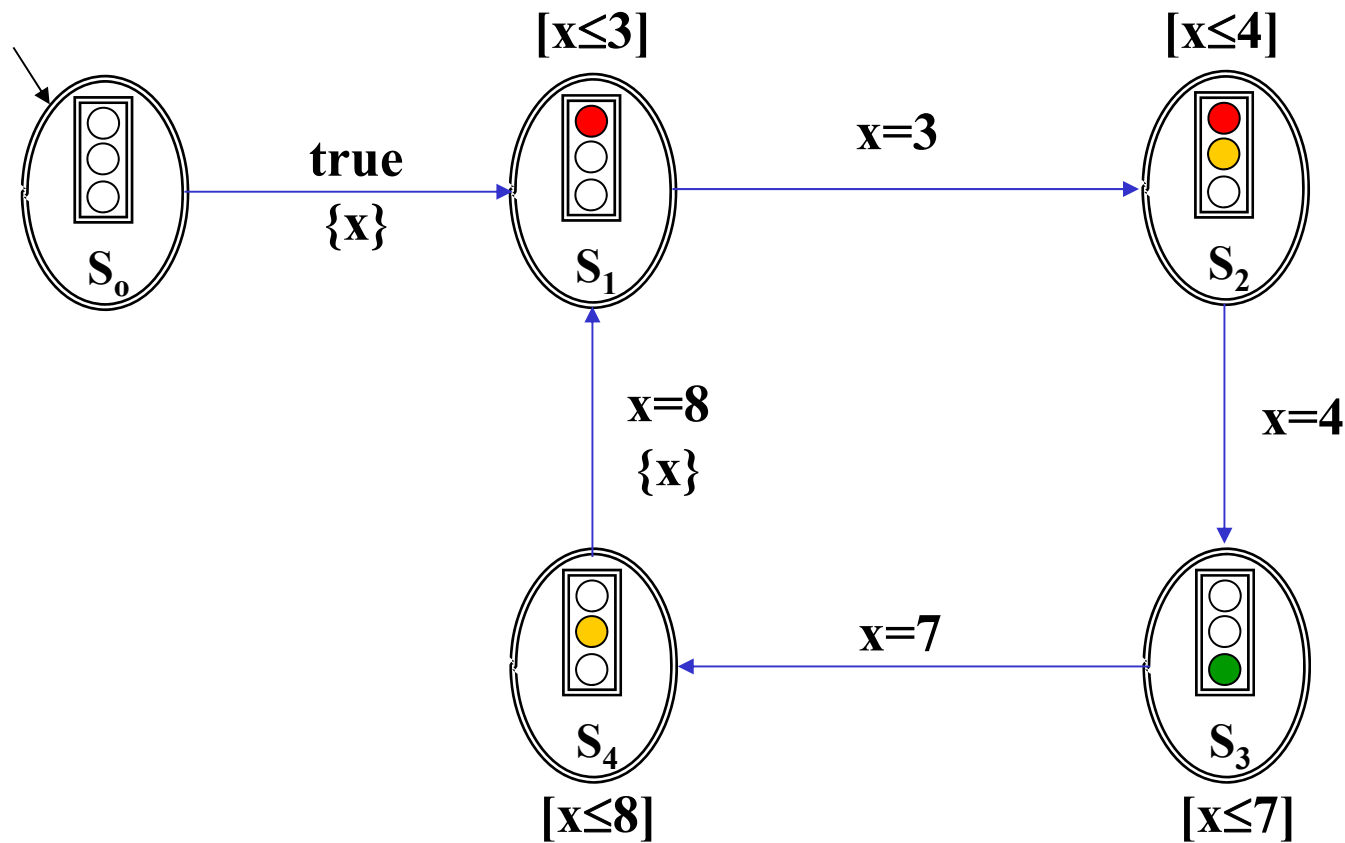


Χρονικά Αυτόματα

- Μία *συνθήκη κατάστασης* είναι ένας περιορισμός ρολογιών που προσδιορίζει τον χρόνο που μπορεί να ξοδευτεί στην κατάσταση.
- Οι φρουροί μεταβάσεων προσδιορίζουν πότε είναι δυνατή μια μετάβαση, ενώ οι συνθήκες καταστάσεων εξαναγκάζουν την εκτέλεση μιας μετάβασης.



Παράδειγμα: Φώτα της τροχαίας



Περιορισμοί Ρολογιών

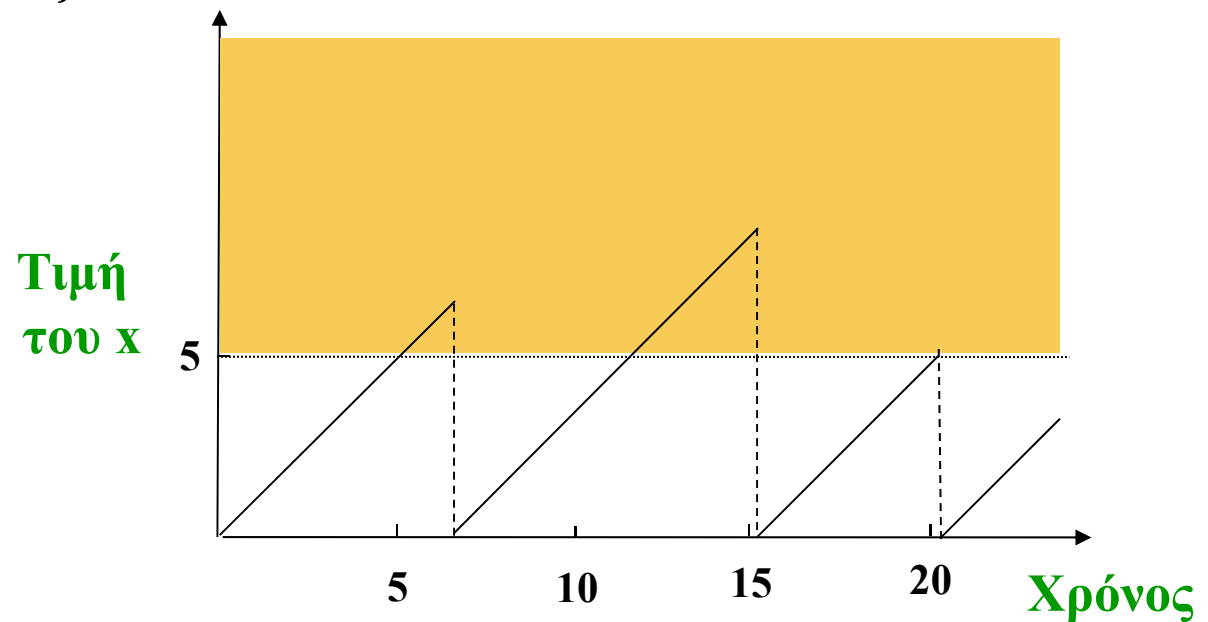
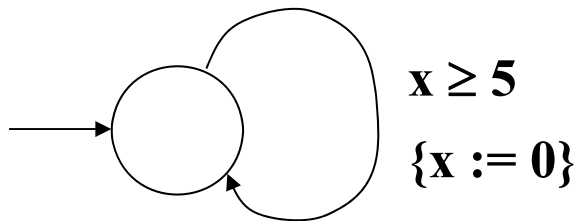
- Το σύνολο *περιορισμών ρολογιών* πάνω στο σύνολο ρολογιών C είναι το μικρότερο σύνολο που ικανοποιεί τους εξής κανόνες:
 - Αν c είναι ένα ρολόι από το σύνολο C και x ένας φυσικός αριθμός, τότε τα $c < x$ και $c \leq x$ είναι περιορισμοί ρολογιών.
 - Αν a είναι περιορισμός ρολογιών τότε και το $\neg a$ είναι περιορισμός ρολογιών.
 - Αν a και b είναι περιορισμοί ρολογιών τότε και το $a \wedge b$ είναι περιορισμός ρολογιών.
- Τα ρολόγια συγκρίνονται μόνο με τους φυσικούς αριθμούς, διαφορετικά ο μοντελο-έλεγχος γίνεται μη-αποφασίσιμος (undecidable).

Χρονικά αυτόματα

- Ένα χρονικό αυτόματο είναι μια πλειάδα ($C, L, l_0, Label, invar, E, clocks, guard$) όπου
 - C είναι ένα σύνολο ρολογιών
 - L είναι ένα σύνολο από καταστάσεις με αρχική κατάσταση την l_0
 - $Label$ είναι μία συνάρτηση που αποδίδει σε κάθε κατάσταση ένα σύνολο από ατομικές προτάσεις (που ικανοποιούνται στην κατάσταση)
 - $invar$ είναι μία συνάρτηση που αποδίδει σε κάθε κατάσταση μια συνθήκη κατάστασης
 - E είναι ένα σύνολο από ακμές ανάμεσα στις καταστάσεις
 - $clocks$ είναι μια συνάρτηση που αποδίδει σε κάθε ακμή το σύνολο των ρολογιών που πρέπει να μηδενιστούν
 - $guard$ είναι μια συνάρτηση που αποδίδει σε κάθε ακμή το φρουρό της ακμής.

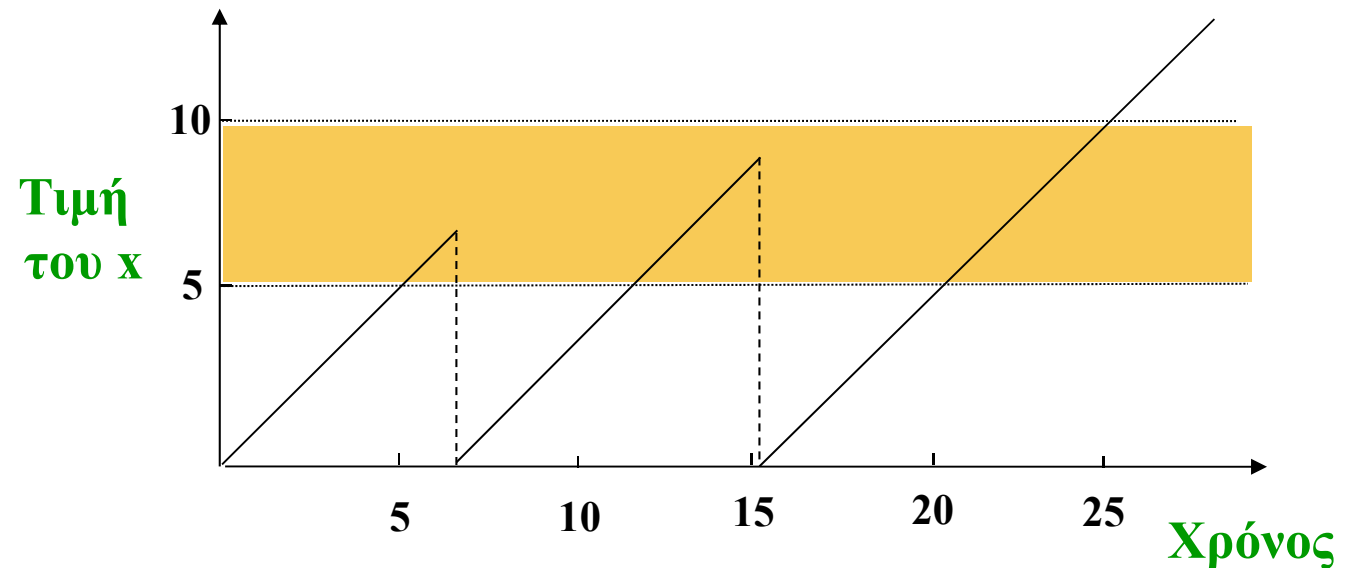
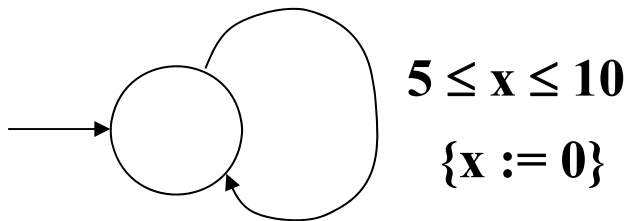
Παράδειγμα – Φρουροί (1)

- Αποτέλεσμα φρουρού με κάτω φράγμα.



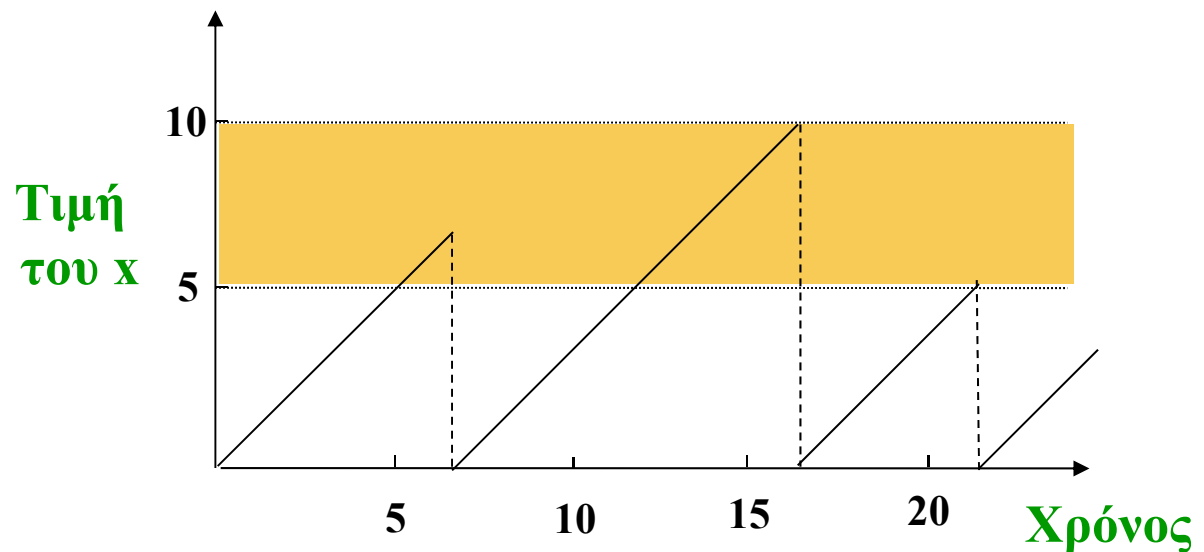
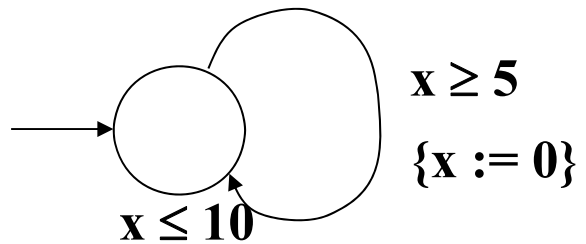
Παράδειγμα – Φρουροί (2)

- Αποτέλεσμα φρουρού με κάτω φράγμα και άνω φράγμα.



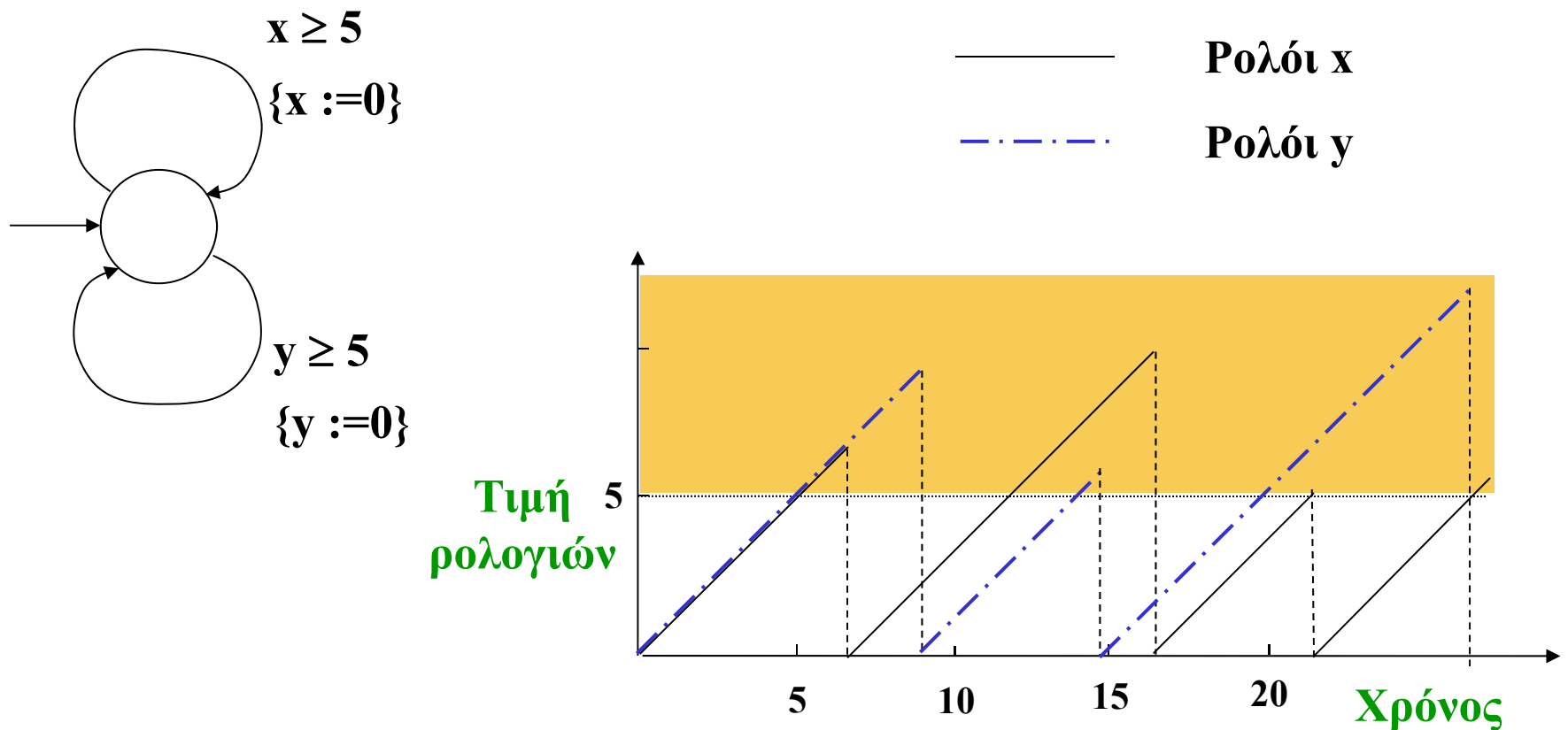
Παράδειγμα – Φρουροί και συνθήκες

- Αποτέλεσμα φρουρού με κάτω φράγμα και συνθήκη.



Πολλαπλά ρολόγια

- Αποτέλεσμα συμπεριφοράς με πολλαπλά ρολόγια. Παρατηρούμε ότι κάτι τέτοιο είναι αδύνατο να μοντελοποιηθεί σε πρότυπα διακριτού χρόνου.



Σύνθεση δύο αυτομάτων

- Για να επιτρέψουμε τη συνθετική δημιουργία συστημάτων χρησιμοποιώντας χρονικά αυτόματα, χρειαζόμαστε ένα τελεστή παράλληλης σύνθεσης.
- Ένας τρόπος για να το πετύχουμε είναι η επέκταση χρονικών αυτομάτων με ετικέτες (που προέρχονται από κάποιο σύνολο Σ) σε κάθε ακμή/μετάβαση και η απαίτηση ότι κατά το συνδυασμό δύο αυτομάτων υπάρχει συγχρονισμός σε κάποιες από αυτές.
- Έστω δύο χρονικά αυτόματα
 $A^1 = \langle \Sigma^1, C^1, L^1, l_0^1, \text{Label}^1, \text{invar}^1, E^1, \text{clocks}^1, \text{guard}^1 \rangle$
και $A^2 = \langle \Sigma^2, C^2, L^2, l_0^2, \text{Label}^2, \text{invar}^2, E^2, \text{clocks}^2, \text{guard}^2 \rangle$, όπου $C^1 \cap C^2 = \emptyset$.

Ορίζουμε ως τομή των δύο αυτομάτων το αυτόματο, $A = A^1 \parallel A^2$

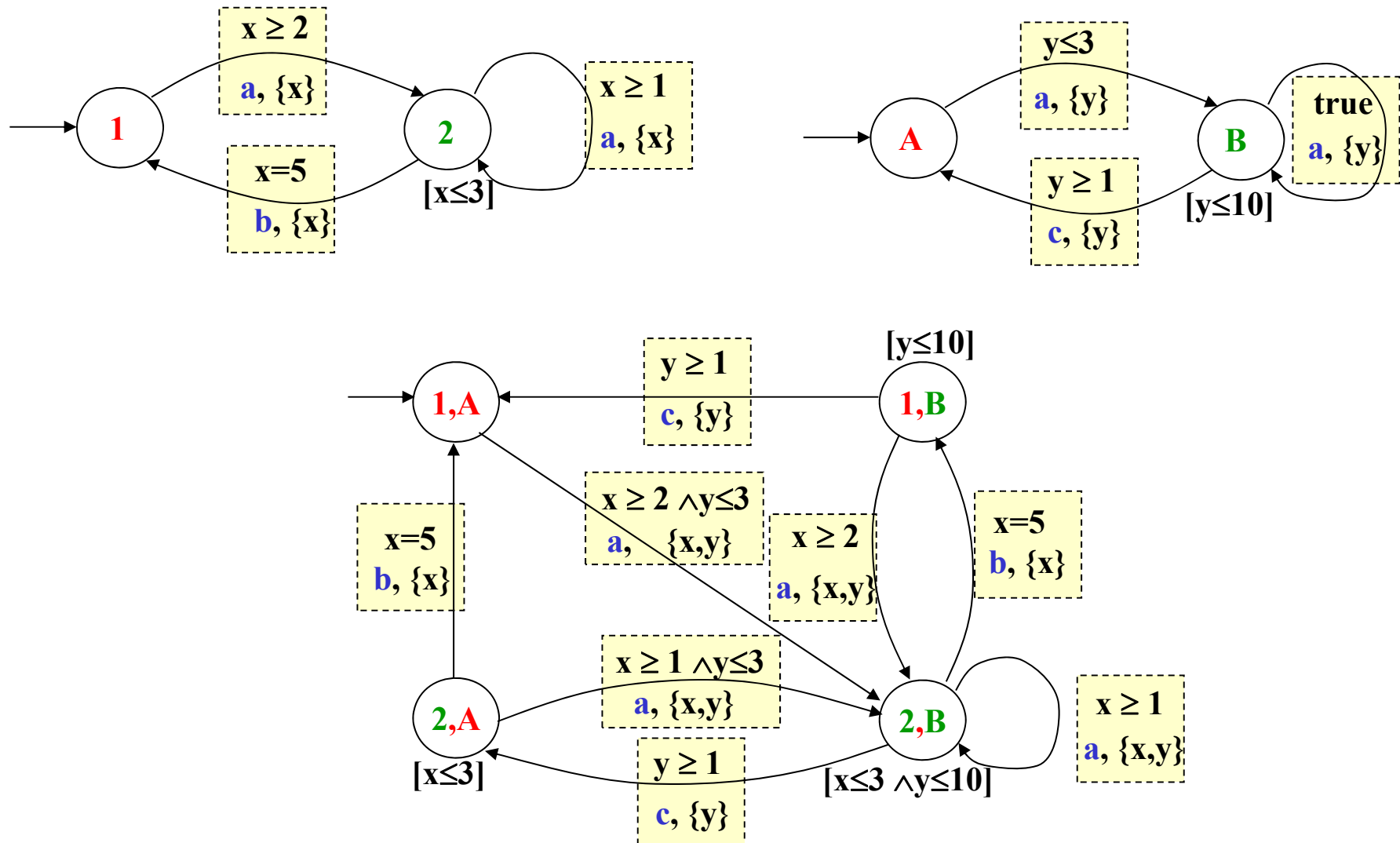
$A = \langle \Sigma^1 \cup \Sigma^2, C^1 \cup C^2, L^1 \times L^2, (l_0^1, l_0^2), \text{Label}, \text{invar}, E, \text{clocks}, \text{guard} \rangle$ όπου

- $\text{Label}(l_1, l_2) = \text{Label}^1(l_1) \cup \text{Label}^2(l_2)$.
- $\text{invar}(l_1, l_2) = \text{invar}^1(l_1) \wedge \text{invar}^2(l_2)$.

Σύνθεση δύο αυτομάτων

- $((l_1, l_2), (l_1', l_2')) \in E$, αν ισχύει ένα από τα ακόλουθα:
 - $(l_1, l_1') \in E_1, \Sigma_1(l_1, l_1') = a$ και $(l_2, l_2') \in E_2, \Sigma_2(l_2, l_2') = a$, οπότε:
 $\Sigma((l_1, l_2), (l_1', l_2')) = a$,
 $\text{clocks}((l_1, l_2), (l_1', l_2')) = \text{clocks}_1(l_1, l_1') \cup \text{clocks}_2(l_2, l_2')$, και
 $\text{guard}((l_1, l_2), (l_1', l_2')) = \text{guard}_1(l_1, l_1') \wedge \text{guard}_2(l_2, l_2')$.
 - $(l_1, l_1') \in E_1, \Sigma_1(l_1, l_1') = a$ και $a \notin \Sigma_2$, οπότε:
 $l_2' = l_2$,
 $\Sigma((l_1, l_2), (l_1', l_2)) = a$,
 $\text{clocks}((l_1, l_2), (l_1', l_2)) = \text{clocks}_1(l_1, l_1')$, και
 $\text{guard}((l_1, l_2), (l_1', l_2)) = \text{guard}_1(l_1, l_1')$.
 - $(l_2, l_2') \in E_2, \Sigma_2(l_2, l_2') = a$ και $a \notin \Sigma_1$, οπότε:
 $l_1' = l_1$,
 $\Sigma((l_1, l_2), (l_1, l_2')) = a$,
 $\text{clocks}((l_1, l_2), (l_1, l_2')) = \text{clocks}_2(l_2, l_2')$, και
 $\text{guard}((l_1, l_2), (l_1, l_2')) = \text{guard}_2(l_2, l_2')$.

Παράδειγμα



Ερμηνεία χρονικών αυτομάτων

- Τα χρονικά συστήματα μεταβάσεων χρησιμοποιούνται για την ερμηνεία χρονικών αυτομάτων ως εξής:
- Οι *καταστάσεις* ενός αυτομάτου δίνονται ως ζεύγη (l, v) όπου l είναι μια κατάσταση του αυτομάτου και v είναι μια συνάρτηση που περιέχει τις τιμές των ρολογιών τα οποία πρέπει να τηρούν τη συνθήκη κατάστασης $\text{invar}(l)$.
- Η *αρχική κατάσταση* είναι η $s_0 = (l_0, v_0)$ και $v_0(x)=0$ για κάθε ρολόι x .

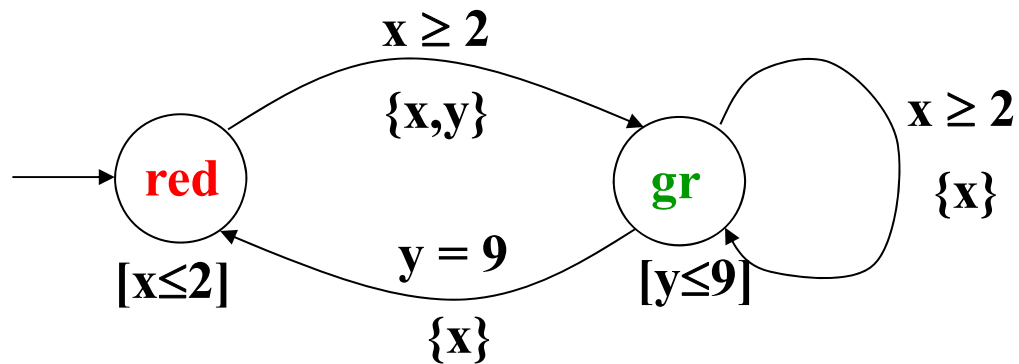
Χρονικά Συστήματα μεταβάσεων

- Ένα *σύστημα μεταβάσεων* ορίζεται ως μια πλειάδα $(S, s_0, \rightarrow)$ όπου
 - S είναι ένα αριθμήσιμο σύνολο από καταστάσεις
 - s_0 είναι η αρχική κατάσταση
 - $\rightarrow$ είναι μία σχέση μεταβάσεων, όπου $s \rightarrow s'$ υπάρχει μετάβαση από την κατάσταση s στην κατάσταση s' .
- Υπάρχουν δύο είδη μεταβάσεων σε ένα χρονικό αυτόματο:
 - *Μετάβαση ακμής*: μπορεί να σημειωθεί πρόοδος με τη διάσχιση μίας ακμής του αυτομάτου.
 - *Μετάβαση καθυστέρησης*: μπορεί να σημειωθεί πρόοδος με την πάροδο χρόνου σε μία κατάσταση.

Ερμηνεία χρονικών αυτομάτων

- Η *σχέση μεταβάσεων* $\rightarrow$ ορίζεται ως εξής:
 - Μετάβαση ακμής $(l, v) \xrightarrow{*} (l', v')$ υπάρχει αν:
 - Υπάρχει ακμή e από την κατάσταση l στην κατάσταση l' .
 - Ο φρουρός $\text{guard}(e)$ ικανοποιείται από τα ρολόγια v .
 - $v'(x) = 0$ αν $x \in \text{clocks}(e)$, διαφορετικά $v'(x) = v(x)$.
 - Μετάβαση καθυστέρησης $(l, v) \xrightarrow{d} (l, v+d)$ υπάρχει αν:
 - Για κάθε $0 \leq d' \leq d$ η συνθήκη κατάστασης της l ικανοποιείται σε κάθε χρόνο $v+d'$.

Παράδειγμα



$$\begin{aligned}
 &(red, v_0) \xrightarrow{2} (red, v_1) \xrightarrow{*} (gr, v_2) \xrightarrow{3} (gr, v_3) \xrightarrow{*} (gr, v_4) \\
 &\xrightarrow{1} (gr, v_5) \xrightarrow{3} (gr, v_6) \xrightarrow{*} (gr, v_7) \xrightarrow{2} (gr, v_8) \\
 &\xrightarrow{*} (red, v_9) \dots
 \end{aligned}$$

	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
x	0	2	0	3	0	1	4	0	2	0
y	0	2	0	3	3	4	7	7	9	9

Σημασιολογία και Χρονικά Παράδοξα

- Η σημασιολογία των χρονικών αυτομάτων αποδίδει σε αυτά ένα σύστημα μεταβάσεων που αποτελείται από ένα μη αριθμήσιμο σύνολο καταστάσεων και μεταβάσεων.
- Αρκετά από τα μονοπάτια δεν αντιπροσωπεύουν ρεαλιστικές συμπεριφορές που σχετίζονται με τα πιο κάτω φαινόμενα
 - Χρονική σύγκλιση/απόκλιση (time convergence/divergence)
 - Timelock
 - Zenoness

Χρονική Απόκλιση

- Έστω κατάσταση l και συνάρτηση ρολογιών v τέτοιες ώστε για κάθε $t < d$, $d \in \text{Real}$, η συνθήκη κατάστασης της l ικανοποιείται για κάθε χρόνο $v + t$. Η **μη-πεπερασμένη ακολουθία μεταβάσεων**

$$(l, v) \xrightarrow{d_1} (l, v + d_1) \xrightarrow{d_2} (l, v + d_1 + d_2) \xrightarrow{d_3} (l, v + d_1 + d_2 + d_3) \\ \xrightarrow{d_4} \dots$$

όπου τα d_i είναι θετικά και $\lim d_1 + d_2 + d_3 + \dots \rightarrow d$ ονομάζεται **χρονικά συγκλίνουσα**.

- Παράδειγμα: Τα φώτα της τροχαίας από τη διαφάνεια 7-14 παρουσιάζουν την πιο κάτω χρονικά συγκλίνουσα μετάβαση που ξεκινά από την κατάσταση S_2 .

$$(S_2, \{x \mapsto 3\}) \xrightarrow{2^{-1}} (S_2, \{x \mapsto 4 - 2^{-1}\}) \xrightarrow{2^{-2}} (S_2, \{x \mapsto 4 - 2^{-2}\}) \\ \xrightarrow{2^{-3}} (S_2, \{x \mapsto 4 - 2^{-3}\}) \xrightarrow{2^{-4}} \dots$$

Χρονική Απόκλιση (2)

- Σε αυτή τη μη-πεπερασμένη μετάβαση ο χρόνος δεν προχωρεί μετά από το 4.
- Στην πράξη τα φώτα της τροχαίας δεν θα παρουσιάσουν αυτή τη συμπεριφορά.
- Ως εκ τούτου, θα πρέπει να την αγνοήσουμε κατά το μοντελοέλεγχο του σχετικού συστήματος, μαζί με όλες τις υπόλοιπες χρονικά συγκλίνουσες μεταβάσεις.
- **Ορισμός:** Έστω η μια μη πεπερασμένη ακολουθία μεταβάσεων. Ορίζουμε

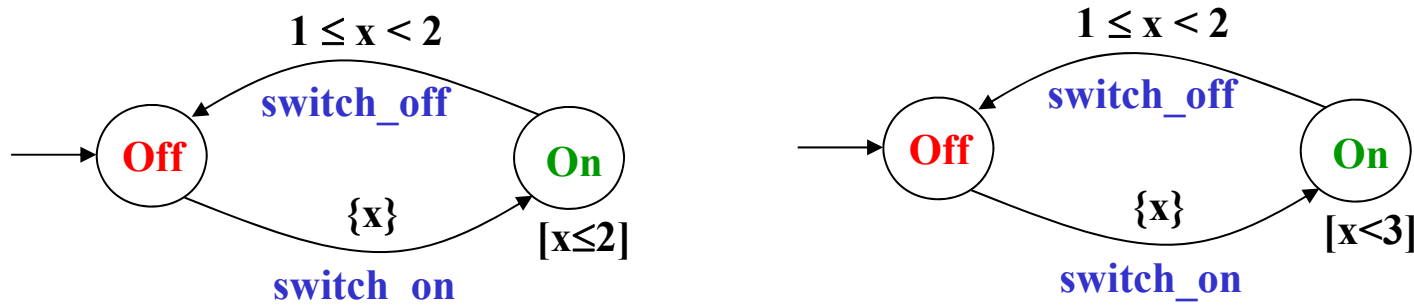
$\text{ExecTime}(\eta)$

ως το άθροισμα των χρόνων που εμφανίζονται στις μεταβάσεις καθυστέρησης της η .

- **Ορισμός:** Μια μη πεπερασμένη μετάβαση η είναι *χρονικά αποκλίνουσα* αν $\text{ExecTime}(\eta) = \infty$. Διαφορετικά ονομάζεται *χρονικά συγκλίνουσα*.

Timelock

- Μια κατάσταση περιέχει *timelock* (κλείδωμα χρόνου) αν δεν παρουσιάζει καμιά χρονικά αποκλίνουσα μετάβαση.
- Παράδειγμα:

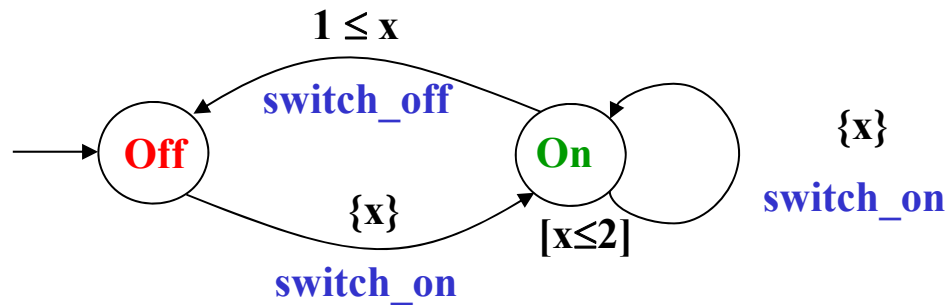


- Και τα δύο αυτόματα πιο πάνω περιέχουν timelock.
- Τέτοιες καταστάσεις δεν είναι ρεαλιστικές και πρέπει να αποφεύγονται κατά τη μοντελοποίηση συστημάτων πραγματικού χρόνου.

Μονοπάτια Zeno

- Ζήνων ο Ελεάτης, 5 αιώνας π.Χ.
- Το μη πεπερασμένο μονοπάτι η είναι *μονοπάτι Zeno* αν είναι χρονικά συγκλίνον και περιέχει μη πεπερασμένο αριθμό μεταβάσεων ακμής.
- Εφόσον είναι αδύνατο να διεξαχθούν άπειρες ενέργειες μέσα σε πεπερασμένη ποσότητα χρόνου τέτοιες συμπεριφορές πρέπει να αποφεύγονται κατά τη μοντελοποίηση συστημάτων πραγματικού χρόνου.
- Ένα χρονικό αυτόματο είναι non-zeno αν και μόνο αν κάθε μονοπάτι του είναι είτε
 - Χρονικά αποκλίνον, είτε
 - Χρονικά συγκλίνον και περιέχει πεπερασμένο αριθμό μεταβάσεων ακμής.

Μονοπάτια Zeno (2)

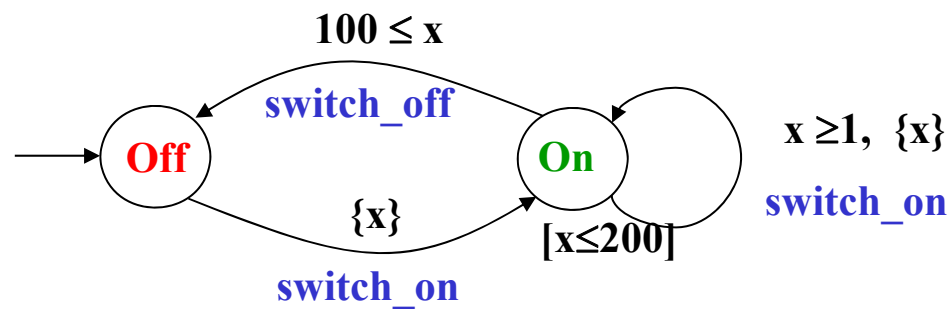


- Το πιο πάνω χρονικό αυτόματο παρουσιάζει διάφορα μονοπάτια zeno, π.χ.

$$\begin{aligned}
 & - (off, 0) \xrightarrow{switch_on} (on, 0) \xrightarrow{switch_on} (on, 0) \xrightarrow{switch_on} \dots \\
 & - (off, 0) \xrightarrow{switch_on} (on, 0) \xrightarrow{0.5} (on, 0.5) \xrightarrow{switch_on} (on, 0) \\
 & \quad \xrightarrow{0.25} (on, 0.25) \xrightarrow{switch_on} \dots
 \end{aligned}$$

Μονοπάτια Zeno (3)

- Το πρόβλημα στο προηγούμενο παράδειγμα μπορεί να αποφευχθεί αν επιβάλουμε μια απαίτηση καθυστέρησης ανάμεσα σε οποιαδήποτε δύο `switch_on`.



- Γενικά είναι δύσκολο να ελέγξουμε αλγοριθμικά κατά πόσο ένα χρονικό αυτόματο περιέχει μονοπάτια zeno. Αντί αυτού, έχουν διατυπωθεί αναγκαίες συνθήκες.
- Λήμμα:** Αν κάθε κύκλος ενός χρονικού αυτόματος έχει χρονική διάρκεια μεγαλύτερη από 0, τότε το χρονικό αυτόματο είναι non-zeno.

Το πρωτόκολλο ABP (1)

- Το πρωτόκολλο εναλλασσόμενου διφύου (Alternating Bit Protocol, ABP) είναι ένα πρωτόκολλο δικτύων για ασφαλή αναμετάδοση δεδομένων διαμέσου καναλιών που δυνατόν να χάσουν ή να αλλοιώσουν μηνύματα.
 - Ο αποστολέας A επιθυμεί να στείλει μια ακολουθία από δεδομένα, $w_1, w_2, \dots$, σε κάποιο παραλήπτη B μέσω ενός καναλιού.
 - Το κανάλι αυτό είναι αναξιόπιστο με την έννοια ότι μηνύματα που στέλλονται μέσω αυτού μπορεί να χαθούν ή να αλλοιωθούν.
 - Έστω ότι το κανάλι μεταξύ A και B αρχικά είναι κενό.
- Τα μηνύματα που στέλλονται είναι τα εξής:
 - Ο A στέλλει στον B μηνύματα που αποτελούνται από ένα τμήμα δεδομένων και έναν αριθμό ακολουθίας (ένα διφύο), δηλαδή, μια τιμή που είναι είτε 0 είτε 1.
 - Ο B μπορεί να στείλει στον A ένα από δύο μηνύματα επιβεβαίωσης (acknowledgment), τα ACK(0) και ACK(1).

Το πρωτόκολλο ABP (2)

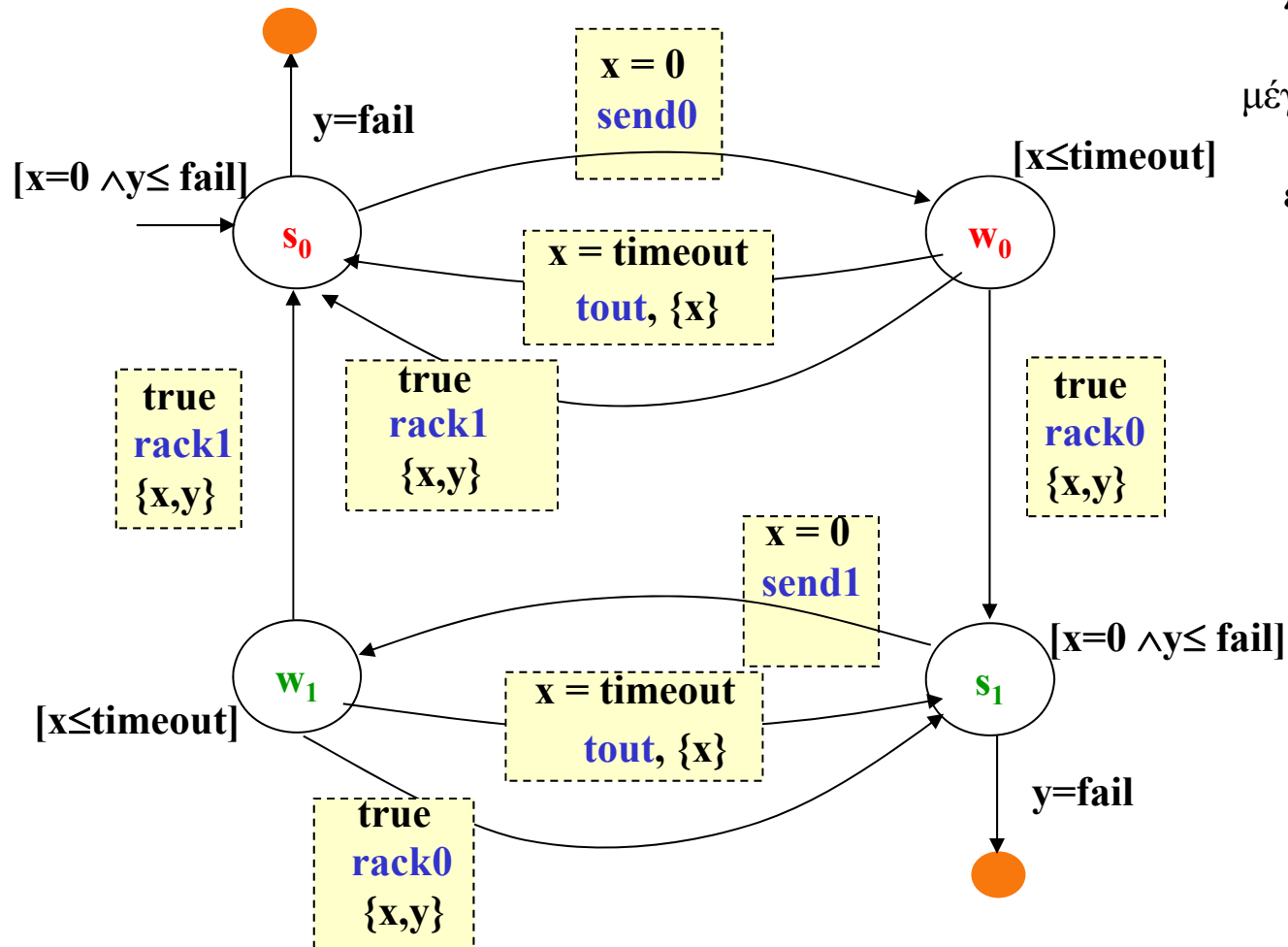
- Λειτουργία του A:
 1. Ο A στέλλει το πρώτο δεδομένο συνοδευόμενο από το δυφίο 0 και αναμένει να λάβει την επιβεβαίωση ACK(0).
 2. Σε τέτοια περίπτωση, συνεχίζει με την αναμετάδοση των δεδομένων στέλλοντας το επόμενο δεδομένο συνοδευόμενο από το δυφίο 1.
 3. Εναλλακτικά, αν λάβει την επιβεβαίωση ACK(1) ή δεν λάβει καμιά απάντηση μέσα σε κάποιο προκαθορισμένο χρόνο, τότε ξαναστέλνει το ίδιο ζεύγος δεδομένου-δυφίου. Μετά από κάποιο συγκεκριμένο αριθμό συνεχόμενων επανειλημμένων προσπαθειών τερματίζει τη λειτουργία του.
 4. Η αποστολή δεδομένων συνεχίζεται με αυτό τον τρόπο εναλλάσσοντας τον αριθμό ακολουθίας από 0 σε 1 και αντίστροφα για κάθε καινούριο δεδομένο.

Το πρωτόκολλο ABP (3)

- Λειτουργία του B:
 1. Κάθε φορά που ο B παραλαμβάνει ένα μήνυμα που δεν είναι αλλοιωμένο και έχει αριθμό ακολουθίας 0, αποστέλλει το μήνυμα ACK(0)
 2. Ξαναστέλνει το μήνυμα αυτό με το πέραςμα κάποιου προκαθορισμένου χρόνου μέχρι να λάβει ένα έγκυρο μήνυμα με αριθμό ακολουθίας 1.
 3. Παρόμοια, στέλλει συνεχόμενα ACK(1) για να επιβεβαιώσει την παραλαβή μηνυμάτων που συνοδεύονται από το δυφίο 1.

Περιγραφή του Αποστολέα ως ΧΑ

- O Sender

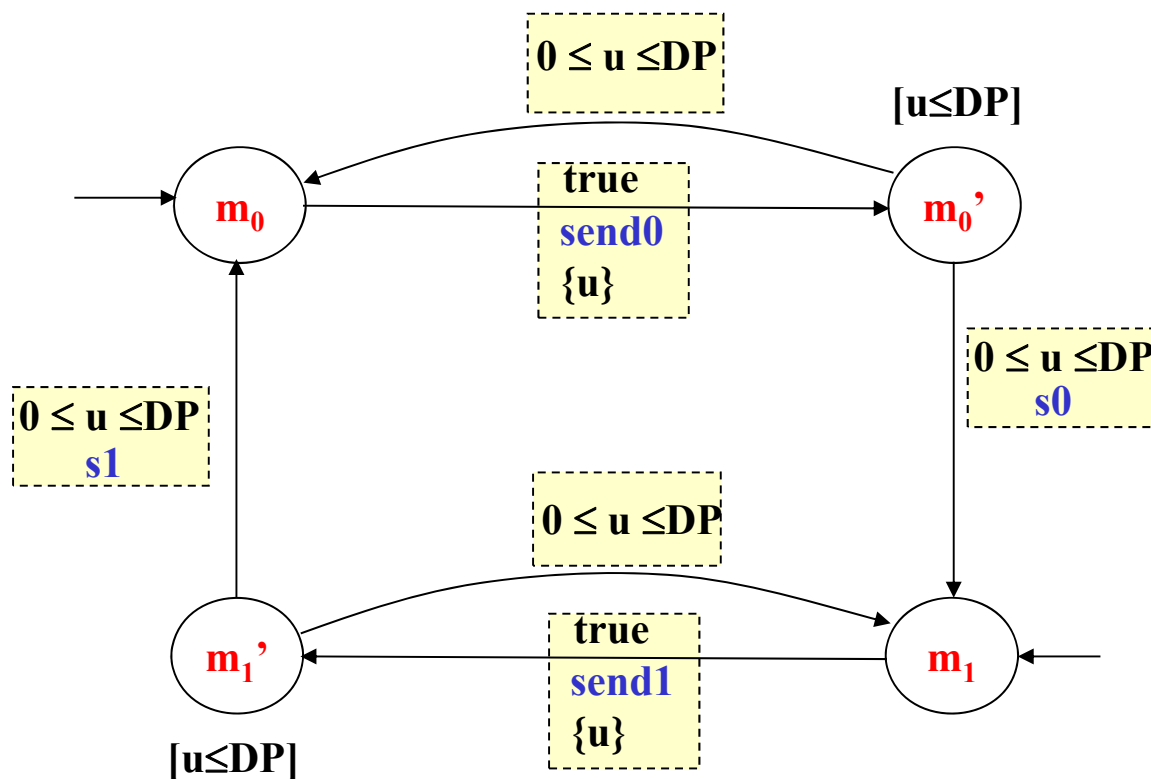


Έστω timeout ο χρόνος πριν ξανασταλθεί ένα μήνυμα και fail ο μέγιστος χρόνος για τον οποίο επαναλαμβάνονται οι δοκιμές.

Περιγραφή του Καναλιού ως ΧΑ

- Το μέσο με απώλειες μηνυμάτων Medium1

Έστω DP η καθυστέρηση του μέσου.



Το πρωτόκολλο ως ΧΑ

- Παρόμοια μπορούν να διατυπωθούν αυτόματα που περιγράφουν τη συμπεριφορά του Receiver και του μέσου Medium2 που είναι υπεύθυνο για την επικοινωνία των acknowledgment.
- Το σύστημα δίνεται από τη σύνθεση των τεσσάρων αυτομάτων, δηλαδή Sender || Medium1 || Receiver || Medium2.

CTL (υπενθύμιση)

Η CTL ορίζεται ως το μικρότερο σύνολο ιδιοτήτων που παράγονται ως εξής:

$$\begin{aligned}\Phi &::= p \mid \neg\Phi \mid \Phi \vee \Psi \mid \mathbf{A} \varphi \mid \mathbf{E} \varphi \\ \varphi &::= \mathbf{X} \Phi \mid \Phi \mathbf{U} \Psi\end{aligned}$$

1. Ιδιότητες κατάστασης – Φ

- κάθε ατομική πρόταση p είναι ιδιότητα *κατάστασης*
- Αν οι Φ και Ψ είναι ιδιότητες *κατάστασης*, τότε και οι $\neg\Phi$ και $\Phi \vee \Psi$ είναι ιδιότητες *κατάστασης*
- Αν η φ είναι μια ιδιότητα *εκτέλεσης*, τότε οι $\mathbf{A} \varphi$ και η $\mathbf{E} \varphi$ είναι ιδιότητες *κατάστασης*

2. Ιδιότητες εκτέλεσης – φ

- Αν οι Φ και Ψ είναι ιδιότητες *κατάστασης*, τότε οι $\mathbf{X} \Phi$ και $\Phi \mathbf{U} \Psi$ είναι ιδιότητες *εκτέλεσης*

Χρονική CTL (TCTL)

Η CTL ορίζεται ως το μικρότερο σύνολο ιδιοτήτων που παράγονται ως εξής:

$$\begin{aligned}\Phi, \Psi &::= p \mid g \mid \neg\Phi \mid \Phi \vee \Psi \mid \mathbf{A} \varphi \mid \mathbf{E} \varphi \\ \varphi &::= \Phi \mathbf{U}^J \Psi\end{aligned}$$

Η χρονική CTL επεκτείνει τη CTL με δύο καινούρια στοιχεία και αποκλείει τον τελεστή **X**:

1. Ιδιότητες κατάστασης – Φ
 - Κάθε **περιορισμός ρολογιού** g είναι ιδιότητα κατάστασης
2. Ιδιότητες εκτέλεσης – φ
 - Αν η Φ είναι ιδιότητα κατάστασης, τότε η $\mathbf{X} \Phi$ δεν είναι ιδιότητα εκτέλεσης.
 - Αν οι Φ, Ψ είναι ιδιότητες κατάστασης, τότε η $\Phi \mathbf{U}^J \Psi$, όπου J διάστημα του οποίου τα άκρα είναι ακέραιοι, $J \subseteq \text{Real}^{\geq 0}$ είναι ιδιότητα εκτέλεσης.

Παραγόμενοι τελεστές

Έστω $J \subseteq \text{Real} \geq 0$. Τότε:

$$\mathbf{EF}^J \Phi \equiv \mathbf{E}(\text{true } \mathbf{U}^J \Phi)$$

$$\mathbf{AG}^J \Phi \equiv \neg \mathbf{EF}^J \neg \Phi$$

$$\mathbf{AF}^J \Phi \equiv \mathbf{A}(\text{true } \mathbf{U}^J \Phi)$$

$$\mathbf{EG}^J \Phi \equiv \neg \mathbf{AF}^J \neg \Phi$$

Παραδείγματα

- **Απαίτηση άμεσης ανταπόκρισης:** προσδιορίζει τη μέγιστη επιτρεπτή καθυστέρηση ανάμεσα σε ένα γεγονός και τη ζητούμενη αντίδραση προς αυτό.

Για παράδειγμα, “μετά από τη μετάδοση κάθε μηνύματος ακολουθεί απάντηση μέσα σε 4 μονάδες χρόνου”:

$$\mathbf{AG} \text{ (send(m) } \rightarrow \mathbf{AF}^{\leq 4} \text{ receive(r}_m\text{))}$$

- **Απαίτηση ακριβούς ανταπόκρισης:** προσδιορίζει την ακριβή καθυστέρηση ανάμεσα σε ένα γεγονός και τη ζητούμενη αντίδραση προς αυτό.

Για παράδειγμα, “είναι δυνατόν η καθυστέρηση ανάμεσα στη μετάδοση ενός μηνύματος και της απάντησης προς αυτό να είναι 11 μονάδες χρόνου”:

$$\mathbf{EF} \text{ (send(m) } \rightarrow \mathbf{EF}^{\leq 11} \text{ receive(r}_m\text{))}$$

Παραδείγματα

- **Απαίτηση ελάχιστης καθυστέρησης:** προσδιορίζει την ελάχιστη καθυστέρηση ανάμεσα σε δύο γεγονότα.

Για παράδειγμα, η διαφορά χρόνου ανάμεσα στις αφίξεις δύο αεροπλάνων στον αεροδιάδρομο πρέπει να είναι τουλάχιστον 100 μονάδες χρόνου.

$$\mathbf{AG} \text{ (arrive}_m \rightarrow \mathbf{AG}^{[0,100]} \neg \text{arrive}_{m'})}$$

- **Απαίτηση διαστήματος καθυστέρησης:** προσδιορίζει ότι ένα γεγονός πρέπει να εμφανίζεται μέσα σε κάποιο διάστημα καθυστέρησης από κάποιο άλλο.

Αεροπλάνα πρέπει να έχουν μέγιστη απόσταση 500 μονάδων χρόνου αλλά η ιδιότητα ασφάλειας πρέπει να διατηρείται.

$$\mathbf{AG} \text{ (arrive}_m \rightarrow (\mathbf{AG}^{[0,100]} \neg \text{arrive}_{m'}) \wedge (\mathbf{AF}^{(100,500]} \text{arrive}_{m'}))$$

Παραδείγματα

- **Απαίτηση περιοδικότητας**: προσδιορίζει ότι κάποιο γεγονός εμφανίζεται σε τακτά διαστήματα.

Για παράδειγμα, το μηχάνημα βγάζει τρύπες τακτικά κάθε 10 μονάδες χρόνου:

AG (**AF**⁼¹⁰ punchhole)

Αυτό όμως επιτρέπει και τρυπήματα ανάμεσα σε κάθε περίοδο. Για να το αποτρέψουμε προσδιορίζουμε ότι

AF^{≤10} punchhole $\wedge$

AG (punchhole $\rightarrow$

A ($\neg$ punchhole **U**⁼¹⁰ punchhole))

Σημασιολογία της TCTL: ιδιότητες κατάστασης

- Για μια ιδιότητα κατάστασης της χρονικής CTL Φ και ένα χρονικό αυτόματο A γράφουμε $A \models \Phi$ αν και μόνο αν η ιδιότητα Φ ικανοποιείται στην κατάσταση (s_0, v_0) του αυτομάτου A . Ορίζουμε $s \models \Phi$, για $s = (l, v)$ ως εξής:

$s \models p$	αν και μόνο αν	$p \in \text{Label}(l)$
$s \models g$	αν και μόνο αν	η g ικανοποιείται στην s
$s \models \neg \Phi$	αν και μόνο αν	δεν ισχύει ότι $s \models \Phi$
$s \models \Phi \vee \Psi$	αν και μόνο αν	$(s \models \Phi)$ ή $(s \models \Psi)$
$s \models E \varphi$	αν και μόνο αν	$w \models \varphi$ για κάποιο χρονικά αποκλίνον μονοπάτι w που ξεκινά από την s
$s \models A \varphi$	αν και μόνο αν	$w \models \varphi$ για κάθε χρονικά αποκλίνον μονοπάτι w που ξεκινά από την s

Σημασιολογία της TCTL: ιδιότητες εκτέλεσης

- Έστω κάποιο χρονικό αυτόματο T με αρχική κατάσταση s_0 . Γράφουμε

$$s_0 \xRightarrow{d_0} s_1 \xRightarrow{d_1} s_2 \xRightarrow{d_2} \dots$$

αν υπάρχει μονοπάτι

$$\begin{aligned} w = s_0 &\xrightarrow{d_0^1} \dots \xrightarrow{d_0^{k_0}} s_0 + d_0 \xrightarrow{a_0} s_1 \xrightarrow{d_1^1} \dots \xrightarrow{d_1^{k_1}} s_1 + d_1 \\ &\xrightarrow{a_1} s_2 \xrightarrow{d_2^1} \dots \xrightarrow{d_2^{k_2}} s_2 + d_2 \xrightarrow{a_2} \dots \end{aligned}$$

όπου $d_i = \sum_{j=1}^{k_i} d_i^j$

- Σε τέτοια περίπτωση, γράφουμε $w \in s_0 \xRightarrow{d_0} s_1 \xRightarrow{d_1} s_2 \xRightarrow{d_2} \dots$

Σημασιολογία της TCTL: ιδ. εκτέλεσης (2)

- Για κάθε χρονικά αποκλίνον μονοπάτι w τέτοιο ώστε

$$w \in \overset{d_0}{s_0} \Rightarrow \overset{d_1}{s_1} \Rightarrow \overset{d_2}{s_2} \Rightarrow \dots$$

ορίζουμε

$w \models \Phi \text{ U}^J \Psi$ αν και μόνο αν υπάρχει $i \geq 0$ τέτοιο ώστε $s_i + d \models \Psi$

για κάποιο $d \in [0, d_i]$,

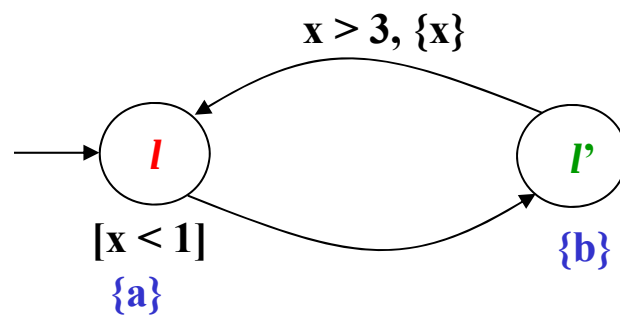
με $\sum_{k=0}^{i-1} d_k + d \in J$ και

για κάθε $j \leq i$, $s_j + d' \models \Phi \vee \Psi$ για

κάθε $d' \in [0, d_j]$ με $\sum_{k=0}^{j-1} d_k + d' \leq \sum_{k=0}^{i-1} d_k + d$

όπου για $s_i = (l_i, v_i)$ και $d \geq 0$ έχουμε ότι $s_i + d = (l_i, v_i + d)$

Παράδειγμα



- Το αυτόματο ικανοποιεί την ιδιότητα $\Phi = A (a U^{>1} b)$;