CTL – Έλεγχος Μοντέλου
(HR Κεφάλαιο 3.5 και 3.6.1)

Στην ενότητα αυτή θα μελετηθούν τα εξής θέματα:

Έλεγχος μοντέλου για τη CTL

*CTL**

Αλγόριθμος Μοντελο-ελέγχου

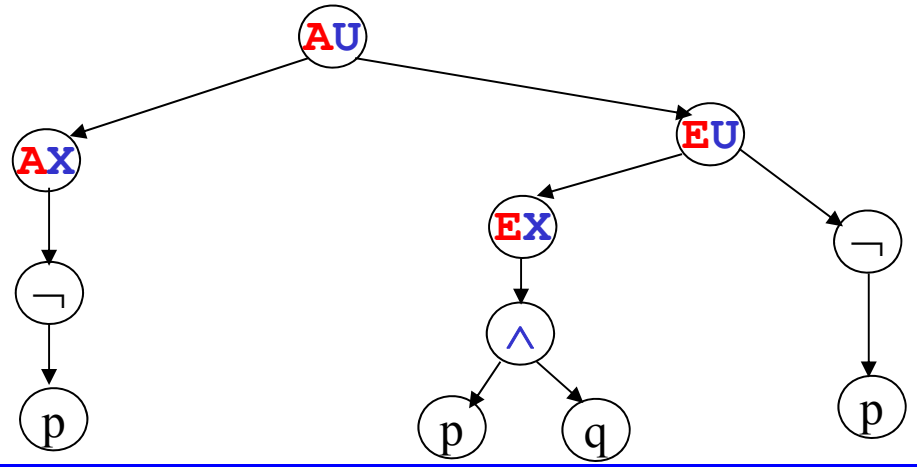
- Πως μπορούμε να ελέγξουμε κατά πόσο μια κατάσταση s ικανοποιεί μια CTL ιδιότητα Φ;
  - Υπολογίζουμε αναδρομικά το σύνολο *Sat*(Φ) των καταστάσεων που ικανοποιούν την Φ.
  - Ελέγχουμε αν η s ανήκει στο *Sat*(Φ).

- Αναδρομικός υπολογισμός
  - προσδιόρισε όλες τις υποιδιότητες της Φ
  - υπολόγισε το σύνολο *Sat*(p) για όλες τις ατομικές προτάσεις p της Φ
  - συνέχισε με τις μικρότερες υποιδιότητες που περιέχουν ατομικές προτάσεις
  - έλεγξε τις υποιδιότητες που περιέχουν αυτές τις ιδιότητες…
  - και ούτω καθεξής μέχρι να φτάσεις στην Φ.

Αλγόριθμος Μοντελο-ελέγχου

- Αναδρομικός υπολογισμός από κάτω προς τα πάνω
  - θεώρησε το δένδρο που αντιστοιχεί στην ιδιότητα Φ
  - υπολόγισε το σύνολο *Sat*(p) για τις ιδιότητες που βρίσκονται στα φύλλα του δένδρου της Φ
  - συνέχισε με τις υποιδιότητες που βρίσκονται σε ύψος 1 στο δένδρο της Φ,
  - στις υποιδιότητες που βρίσκονται σε ύψος 2…
  - και ούτω καθεξής μέχρι να φτάσεις στη ρίζα του δένδρου, δηλαδή στην Φ.

- Το δένδρο που αντιστοιχεί στην ιδιότητα **A**[**AX** ¬p **U** **E**[**EX**(p ∧ q) **U** ¬p]] είναι το



Επαρκή Σύνολα Τελεστών

- Ο αλγόριθμος βασίζεται στο γεγονός ότι το σύνολο των τελεστών $\neg$, $\vee$, T (true), EX, EU, AF είναι επαρκές για τη CTL. Δηλαδή όλοι οι υπόλοιποι τελεστές μπορούν να διατυπωθούν βάσει αυτών:
  - $AX\ \Phi = \neg\ EX\ \neg\Phi$
  - $A(\Phi_1 U \Phi_2) = \neg\ (E[\neg\Phi_2 U(\neg\Phi_1 \wedge \neg\ \Phi_2)] \vee\ EG\ \neg\Phi_2)$
  - $EF\ \Phi = E(T\ U\ \Phi)$
  - $EG\ \Phi = \neg AF\ \neg\ \Phi$
  - $AG\ \Phi_1 = \neg EF\ \neg\Phi_1$

Αναδρομική Διαδικασία (1)

```
SAT(Φ){
Case
   Φ = T                 return S
   Φ = ⊥                 return ∅
   Φ = p                 return {s ∈ S | p ∈ Label(s)}
   Φ = ¬Φ1               return   S - SAT (Φ1)
   Φ = Φ1∨Φ2             return SAT(Φ1) ∪ SAT(Φ2)
   Φ = Φ1∧Φ2             return SAT(Φ1) ∩ SAT(Φ2)
   Φ = Φ1→Φ2             return SAT(¬Φ1 ∨ Φ2)
   Φ = AX Φ1             return SAT(¬EX ¬Φ1)
   …
```

το σύνολο όλων των καταστάσεων

Αναδρομική Διαδικασία (2)

```
  …
  Φ = EX Φ1          return SAT_EX(Φ1)
  Φ = A(Φ1UΦ2)       return SAT(¬(E[¬Φ2U(¬Φ1∧¬ Φ2)]∨ EG¬Φ2))
  Φ = E(Φ1 U Φ2)     return SAT_EU(Φ1, Φ2)
  Φ = EF Φ1          return SAT(E(T U Φ1))
  Φ = EG Φ1          return SAT(¬AF ¬Φ1)
  Φ = AF Φ1          return SAT_AF(Φ1)
  Φ = AG Φ1          return SAT(¬EF ¬Φ1)
}
```

Η διαδικασία $SAT_{EX}(\Phi)$

- Υπολογίζει τις καταστάσεις που ικανοποιούν την Φ (SAT(Φ)) και μετά οπισθοδρομεί για να υπολογίσει το σύνολο των καταστάσεων που μπορούν να μεταβούν σ'αυτές.

```
SAT_EX(Φ){
   X = SAT(Φ);
   Y = {s ∈ S | υπάρχει s' τ.ώ. s → s', s'∈ X};
   return Y;
}
```

Η διαδικασία $SAT_{AF}(\Phi)$

- Υπολογίζει τις καταστάσεις X που ικανοποιούν την Φ (SAT(Φ)) και μετά οπισθοδρομεί για να υπολογίσει το σύνολο των καταστάσεων των οποίων κάθε εκτέλεση φθάνει σε μία από τις καταστάσεις X σε ένα βήμα, δύο βήματα, κ.ο.κ. .

```
SAT_AF(Φ){
   X = S;
   Y = SAT(Φ);
   while (X != Y)
        X = Y;
        Y = Y ∪{s ∈ S | για κάθε s' τ.ω. s → s',
                                  τότε s'∈ Y};
   return Y;
}
```

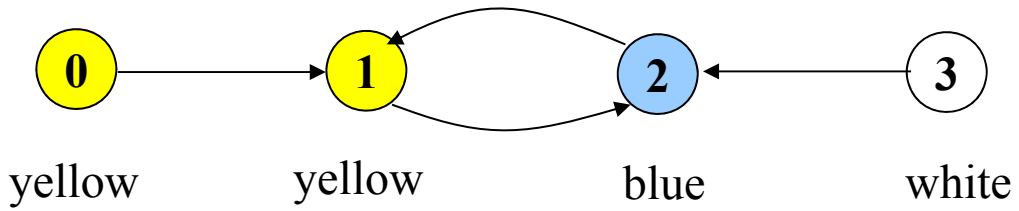
Η διαδικασία $SAT_{EU}(\Phi,\Psi)$

- Υπολογίζει τις καταστάσεις W και Y που ικανοποιούν τις Φ και Ψ αντίστοιχα και μετά οπισθοδρομεί από τις καταστάσεις Y προσθέτοντας στο σύνολο των αποδεκτών καταστάσεων εκείνες που ανήκουν στη W.

```
SAT_EU(Φ, Ψ){
   X = S;
   W = SAT(Φ);
   Y = SAT(Ψ);
   while (X != Y)
        X = Y;
        Y = Y ∪{s ∈ W | υπάρχει s' τ.ώ. s → s'
                                και s'∈ Y};
   return Y;
}
```

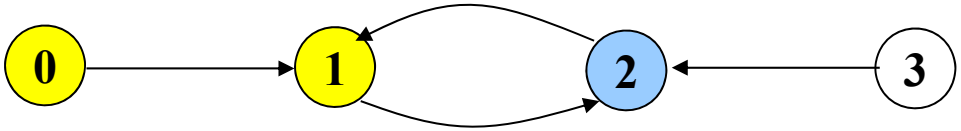
Έλεγχος της E(yellow U blue)

- Θα υπολογίσουμε το σύνολο *Sat*(E(yellow U blue))

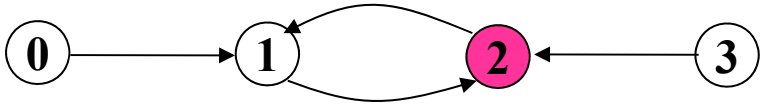


- *Sat*(yellow) = {0, 1}
- *Sat*(blue) = {2}

Έλεγχος της E(yellow U blue)



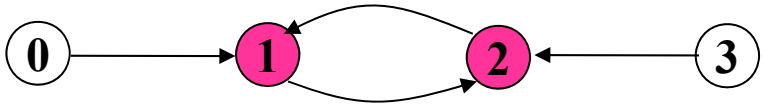
Επανάληψη 1



$Y^0 = \{2\}$ $X^0 = \{0,1,2,3\}$

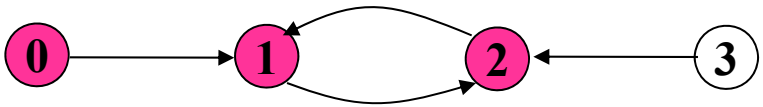
$W = \{0,1\}$

Επανάληψη 2



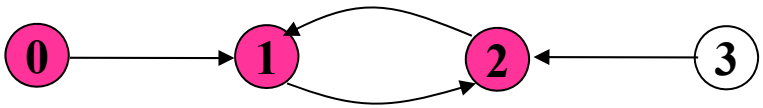
$Y^1 = \{1, 2\}$ $X^1 = \{2\}$

Επανάληψη 3



$Y^2 = \{0,1, 2\}$ $X^2 = \{1,2\}$

Επανάληψη 4



$Y^3 = \{0,1, 2\}$ $X^3 = \{0,1,2\}$

Μοντελο-έλεγχος στη CTL

- Χρόνος εκτέλεσης χείριστης περίπτωσης είναι της τάξης $O(|\Phi| \cdot N^2)$ όπου $|\Phi|$ είναι το μήκος της ιδιότητας Φ και N ο αριθμός καταστάσεων του μοντέλου του συστήματος.

- Υλοποιημένος σε εργαλεία όπως τα UPPAAL, SMV, Cadence, NuSMV…

- Το πρόβλημα της έκρηξης καταστάσεων
  - Τις πλείστες φορές το μέγεθος του μοντέλου είναι εκθετικό ως προς τον αριθμό μεταβλητών και συνιστωσών του συστήματος υπό μελέτη. Έτσι, για παράδειγμα, εισαγωγή μιας μεταβλητής Boolean σε ένα μοντέλο διπλασιάζει το μέγεθος του!
  - Διάφορες προσπάθειες για αντιμετώπιση του προβλήματος, όπως αποδοτικές δομές δεδομένων (OBDDs: Ordered Binary Decision Diagrams), Απόσπαση (abstraction), Partial order reduction, Επαγωγή και Συνθετικές Μέθοδοι.

CTL στην πράξη

- Τυπικές ιδιότητες μπορούν να διατυπωθούν ως ψηλού επιπέδου προδιαγραφές.

- Σε τέτοιες προδιαγραφές ο χρήστης
  - δεν χρειάζεται να γνωρίζει χρονική λογική
  - απλά τοποθετεί τις ατομικές προτάσεις που τον ενδιαφέρουν στις προδιαγραφές που θέλει να ελέγξει

- Τέτοιου είδους προδιαγραφές μπορούν να χωριστούν σε τρεις βασικές κατηγορίες
  - ολικές: αναφέρονται στο σύνολο της εκτέλεσης του συστήματος
  - μετά: αναφέρονται σε εκτελέσεις μετά από κάποια κατάσταση
  - ανάμεσα: αναφέρονται στους υπολογισμούς που λαμβάνουν χώρο ανάμεσα σε δύο καταστάσεις

Τυπικές προδιαγραφές ψηλού επιπέδου

- Μελέτη 555 προδιαγραφών έδειξε τις πιο κάτω συχνότητες για τις δημοφιλέστερες προδιαγραφές

προδιαγραφή	*τύπος*	*CTL ιδιότητα*	*συχνότητα*
ανταπόκριση	ολική	**AG** (p ⇒ **AF** q)	43.3%
καθολικότητα	ολική	**AG** p	19.8%
απουσία	ολική	**AG** ¬p	7.4%
προβάδισμα	ολική	**AG** ¬p ∨ **A**(¬ p **U** q)	4.5%
Απουσία	ανάμεσα	**AG** ((q ∧ ¬r) ⇒ **A**((¬ p ∨ **AG** ¬r) **W** r)	3.2%
απουσία	μετά	**AG** (p ⇒ **AF** ¬ q)	2.1%
ύπαρξη	ολική	**AF** p	2.1%

CTL (Computation Tree Logic)

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

$$\Phi, \Psi ::= p \mid \neg\Phi \mid \Phi \vee \Psi \mid \mathbf{A}\,\varphi \mid \mathbf{E}\,\varphi$$

$$\varphi ::= \mathbf{X}\,\Phi \mid \Phi\,\mathbf{U}\,\Psi$$

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*

PLTL

Η PLTL μπορεί παρόμοια να οριστεί με βάση την πιο κάτω γραμματική

$$\Phi \quad ::= \quad \mathbf{A}\ \varphi$$

$$\varphi \quad ::= \quad p \quad | \quad \neg\varphi \quad | \quad \varphi \vee \psi \quad | \quad \mathbf{X}\ \varphi \quad | \quad \varphi\ \mathbf{U}\ \psi$$

1. Ιδιότητες κατάστασης – Φ
   - Αν η φ είναι μια ιδιότητα *εκτέλεσης* τότε η $\mathbf{A}\ \varphi$ είναι ιδιότητα *κατάστασης*
2. Ιδιότητες εκτέλεσης – φ
   - κάθε ατομική πρόταση p είναι ιδιότητα *εκτέλεσης*
   - Αν οι φ και ψ είναι ιδιότητες *εκτέλεσης*, τότε και οι $\neg\varphi$ και $\varphi \vee \psi$ είναι ιδιότητες *εκτέλεσης*
   - Αν οι φ και ψ είναι ιδιότητες *εκτέλεσης*, τότε οι $\mathbf{X}\ \varphi$ και $\varphi\ \mathbf{U}\ \psi$ είναι ιδιότητες *εκτέλεσης*

PLTL και CTL

- Οι δύο τύποι λογικής έχουν διαφορετική εκφραστικότητα:
  - υπάρχουν ιδιότητες της CTL που δεν μπορούν να εκφραστούν στην PLTL, π.χ. AG EF p
  - υπάρχουν ιδιότητες που δεν μπορούν να εκφραστούν στην CTL, π.χ. F (p $\wedge$ X p)

    Η ιδιότητα αυτή εκφράζει ότι σε κάθε εκτέλεση η p θα ικανοποιηθεί για δύο συνεχόμενες χρονικές στιγμές. Οι ιδιότητες AF (p $\wedge$ AX p) και AF (p $\wedge$ EX p) εκφράζουν διαφορετικές προτάσεις.

- Η πολυπλοκότητα του μοντελο-ελέγχου για τους δύο τύπους λογικής είναι
  - CTL : $O(|\text{Formula}|\cdot|\text{System}|^2)$
  - PLTL : $O(2^{|\text{Formula}|}.|\text{System}|^2)$

  Συχνά όμως ιδιότητες της CTL είναι μακρύτερες από ιδιότητες της PLTL.

CTL*

- Διακλαδωμένη χρονική λογική με μεγαλύτερη εκφραστικότητα.
- Η σύνταξη της δίνεται ως εξής:

Φ :: = p | $\neg\Phi$ | $\Phi \vee \Psi$ | **A** φ | **E** φ

φ :: = Φ | $\neg\varphi$ | $\varphi \vee \psi$ | **X** ψ | φ **U** ψ

- Έτσι, για παράδειγμα οι ιδιότητες **AXX** p, **EGF** p είναι νόμιμες CTL* ιδιότητες.
- Σημείωση: οι ιδιότητες **AF AF** p (CTL) και **AGF** p (CTL*) αν και συντακτικά διαφορετικές, εκφράζουν την ίδια προδιαγραφή.
- Η πολυπλοκότητα του μοντελο-ελέγχου για τη CTL* είναι PSPACE και $O(2^{|System|} \cdot |Formula|)$. Μέχρι στιγμής δεν έχει διαδοθεί η χρήση εργαλείου για μοντελο-έλεγχο της CTL*.

Εκφραστικότητα χρονικών λογικών

- Δύο ιδιότητες είναι *ισοδύναμες* αν και μόνο αν ικανοποιούνται από ακριβώς τις ίδιες καταστάσεις όλων των δομών Kripke.
- Μία χρονική λογική Λ είναι *τουλάχιστον τόσο εκφραστική* όσο και μια λογική Λ' αν και μόνο αν για κάθε ιδιότητά της Λ' υπάρχει ισοδύναμη ιδιότητα της Λ.
- Η εκφραστικότητα των λογικών PLTL, CTL και CTL* φαίνεται στο πιο κάτω διάγραμμα.