

# Κεφάλαιο 17

## *Μελέτη Περιπτώσεων Συστημάτων Σχεδιασμού*

Τεχνητή Νοημοσύνη - Β' Έκδοση

Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας, Η. Σακελλαρίου

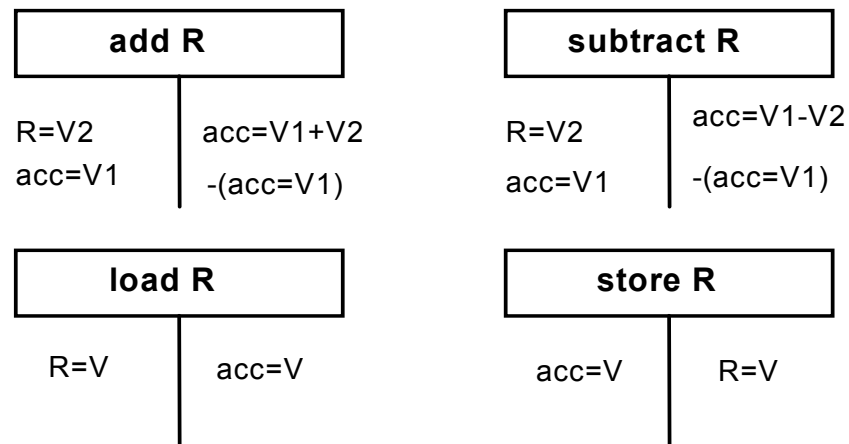
# Γραμμικά Πλάνα με Ανάστροφη Διάσχιση

## STRIPS (1/2)

### ❖ Παράδειγμα:

□ Έστω ένας πολύ απλός υπολογιστής με ένα συσσωρευτή (Accumulator ή acc) και έναν αριθμό καταχωρητών (Registers ή reg1, reg2, reg3). Οι διαθέσιμες εντολές είναι:

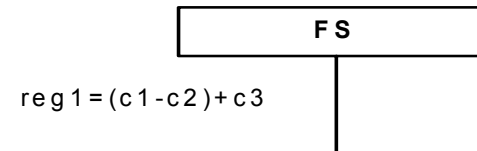
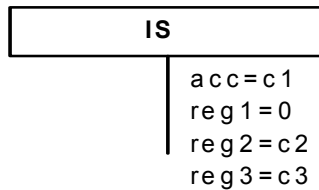
- add R (πρόσθεσε την τιμή του καταχωρητή R στο συσσωρευτή)
- subtract R (αφαίρεσε την τιμή του καταχωρητή R από το συσσωρευτή)
- load R (αντέγραψε την τιμή του καταχωρητή R στο συσσωρευτή)
- store R (αντέγραψε την τιμή του συσσωρευτή στον καταχωρητή R)



# Γραμμικά Πλάνα με Ανάστροφη Διάσχιση

## STRIPS (2/2)

❖ Η αρχική και η τελική κατάσταση:

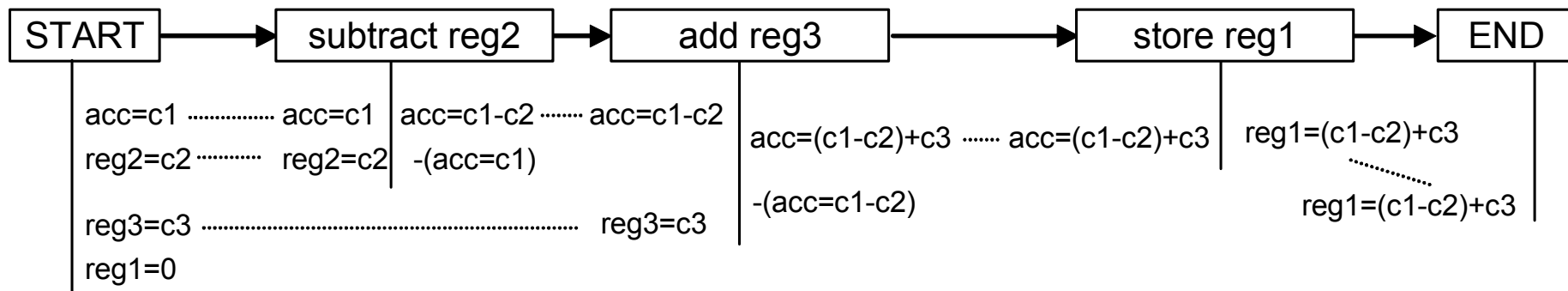


❖ Εφαρμόζοντας τον STRIPS, η πρώτη πρόταση που δεν ικανοποιείται είναι η  $reg1=(c1-c2)+c3$ .

❑ Γίνεται η εισαγωγή στο πλάνο της ενέργειας `store reg1`, η οποία όμως έχει μία προϋπόθεση, την  $acc=(c1-c2)+c3$ , που δεν ικανοποιείται.

❑ Στη συνέχεια γίνεται η εισαγωγή νέας ενέργειας, της `add reg3`, κοκ.

❖ Το τελικό πλάνο:



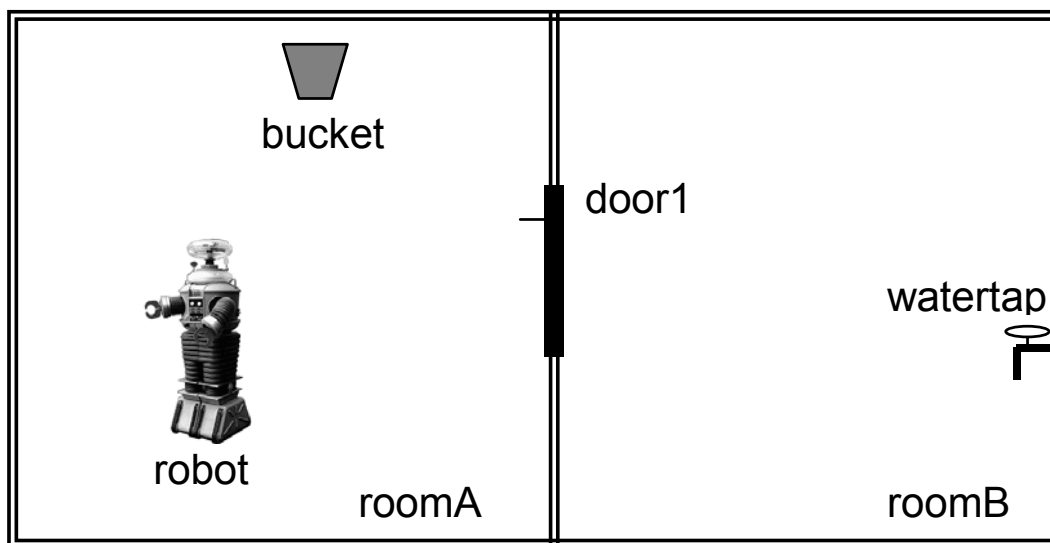
# Ιεραρχικός Σχεδιασμός

## ABSTRIPS (ABstract STRIPS)

- ❖ Σε κάθε κατηγορία αποδίδεται μία τιμή σημαντικότητας (*criticality value*).
- ❖ Ο αλγόριθμος του ABSTRIPS είναι ο ίδιος με αυτόν του STRIPS, μόνον που κάθε φορά επαναλαμβάνεται για το υποπρόβλημα για τα γεγονότα με σημαντικότητα μικρότερη ενός ορίου.
- ❖ Σε κάθε επίπεδο σημαντικότητας ο ABSTRIPS δημιουργεί ένα αφηρημένο πλάνο (*abstract plan*), το οποίο χρησιμοποιείται ως σκελετός για την κατασκευή του αμέσως πιο λεπτομερούς πλάνου.

# Παράδειγμα Ιεραρχικού Σχεδιασμού

## Αρχική - Τελική Κατάσταση



- ❖ Αρχική κατάσταση IS:  
 $connects(door1, roomA, roomB) \wedge closed(door1) \wedge in(robot, roomA) \wedge in(bucket, roomA) \wedge pushable(bucket) \wedge in(watertap, roomB) \wedge closed(watertap) \wedge empty(bucket)$
- ❖ Τελική κατάσταση FS (στόχος):  
 $full(bucket)$

# Παράδειγμα Ιεραρχικού Σχεδιασμού

## Γεγονότα και Τιμές Σημαντικότητας

Γεγονός	Τιμή Σημαντικότητας	Γεγονός	Τιμή Σημαντικότητας
in(X,R)	3	open(X)	2
pushable(X)	3	closed(X)	2
connects(D,R1,R2)	3	near(X,Y)	1
full(bucket)	3	empty(bucket)	1

## Ενέργειες

*PushThroughDoor, GetNear, PushNear, FillBucket, Open, Close*

### PushThroughDoor(Object,Door,R1,R2)

2	open(Door)	- in(robot,R1)	3
1	near(robot,Object)	- in(Object, R1)	3
1	near(robot,door)		
3	pushable(Object)	+ in(robot,R2)	3
3	in(robot,R1)	+ in(Object, R2)	3
3	in(Object, R1)		
3	connects(Door,R1,R2)		

# Παράδειγμα Ιεραρχικού Σχεδιασμού

## Δημιουργία Πλάνων

- ❖ Το πρώτο πλάνο ικανοποιεί τα γεγονότα με την υψηλότερη τιμή σημαντικότητας (3)

*START* ⇒ *PushThroughDoor(bucket, door1, roomA, roomB)*

⇒ *FillBucket* ⇒ *END*

- ❖ Το δεύτερο πλάνο ικανοποιεί τα γεγονότα με τιμή σημαντικότητας (2)

*START* ⇒ *Open(door1)* ⇒ *PushThroughDoor(bucket, door1, roomA, roomB)*

⇒ *Open(watertap)* ⇒ *FillBucket* ⇒ *END*

- ❖ Το τελικό πλάνο ικανοποιεί τα γεγονότα με τη χαμηλότερη τιμή σημαντικότητας (1)

*START* ⇒ *GetNear(bucket)* ⇒ *PushNear(bucket, door1)* ⇒ *Open(door1)* ⇒

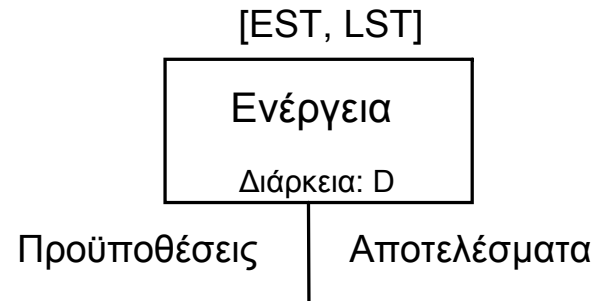
*PushThroughDoor(bucket, door1, roomA, roomB)* ⇒

*PushNear(bucket, watertap)* ⇒ *Open(watertap)* ⇒ *FillBucket* ⇒ *END*

# Σχεδιασμός με Χρονικές Στιγμές

## DEVISER

- ❖ Σε κάθε ενέργεια αποδίδεται και ένα χρονικό παράθυρο [EST, LST] που υποδηλώνει:
  - ❑ Το νωρίτερο που μπορεί να αρχίσει η ενέργεια (*earliest starting time - EST*).
  - ❑ Το αργότερο που μπορεί να αρχίσει η ενέργεια (*latest starting time - LST*).

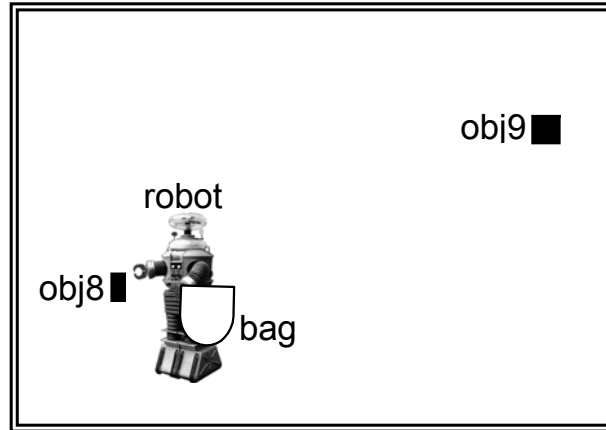


- ❖ Για δύο ενέργειες a και b, τέτοιες ώστε η ενέργεια a να πρέπει να εκτελεστεί πριν από την ενέργεια b, οι τιμές EST και LST ανανεώνονται σύμφωνα με τους τύπους:
  - ❑  $EST(b) = \max( EST(b), EST(a) + DUR(a) )$
  - ❑  $LST(a) = \min( LST(a), LST(b) - DUR(a) )$
- ❖ Σε κάθε βήμα πρέπει να τηρούνται οι εξής περιορισμοί:
  - ❑  $EST(a) + DUR(a) \leq EST(b)$
  - ❑  $LST(a) + DUR(a) \leq LST(b)$



# Παράδειγμα Σχεδιασμού με Χρονικές Στιγμές

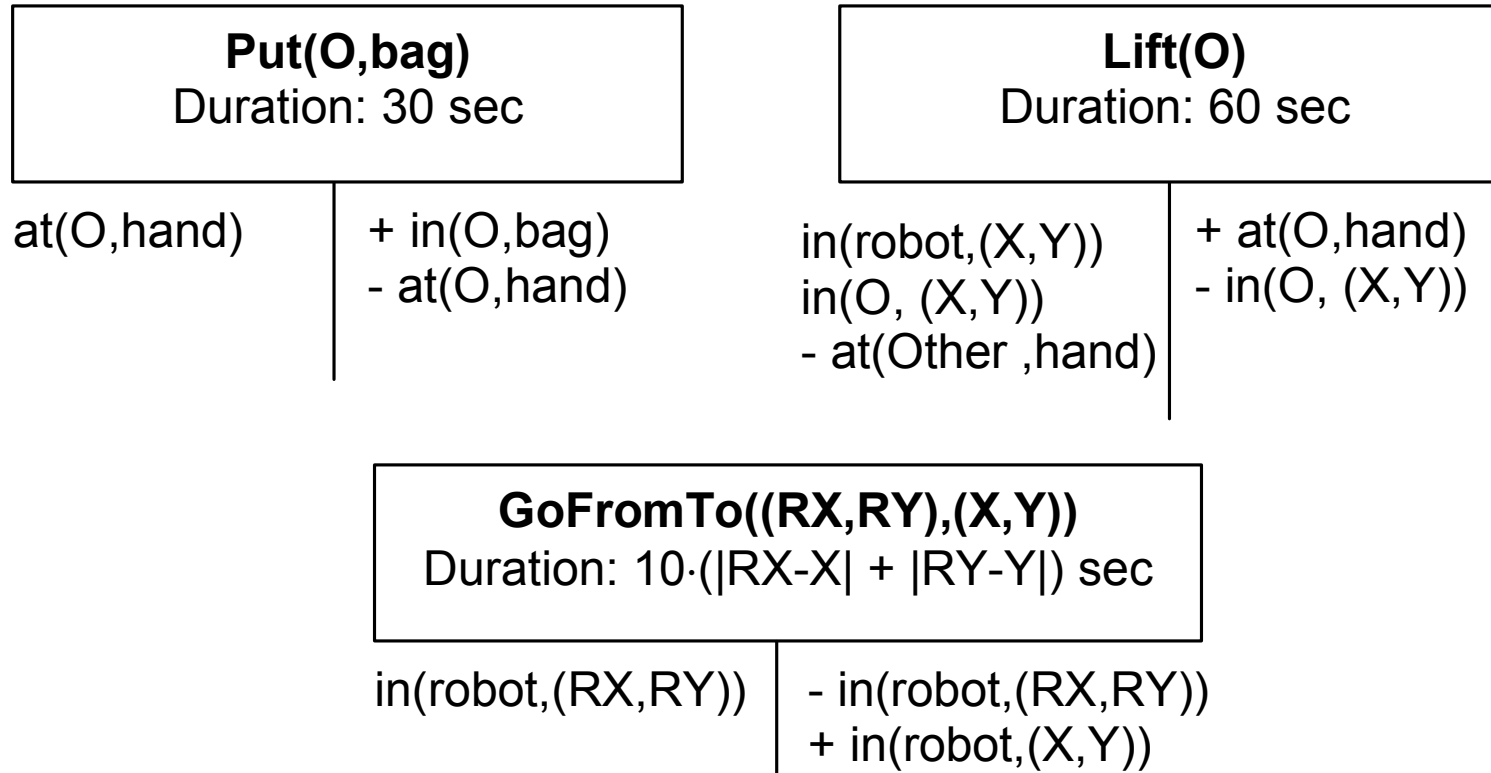
## Περιγραφή Προβλήματος



- ❖ Υπάρχει ένα ρομπότ που μεταφέρει μία τσάντα.
  - ❑ Το ρομπότ βάζει στην τσάντα αντικείμενα τα οποία βρίσκει σκορπισμένα σε ένα χώρο χωρίς εμπόδια.
- ❖ Αρχική κατάσταση *START*:  
 $in(robot, (1,3)) \wedge carry(robot, bag) \wedge in(obj9, (7,5)) \wedge at(obj8, hand)$
- ❖ Τελική κατάσταση *FS*:  
 $in(obj9, bag)$
- ❖ Χρονικός περιορισμός: Η συνολική διάρκεια εκτέλεσης του πλάνου δεν πρέπει να ξεπερνά τα 4 λεπτά (240 δευτερόλεπτα).

# Παράδειγμα Σχεδιασμού με Χρονικές Στιγμές

## Τελεστές



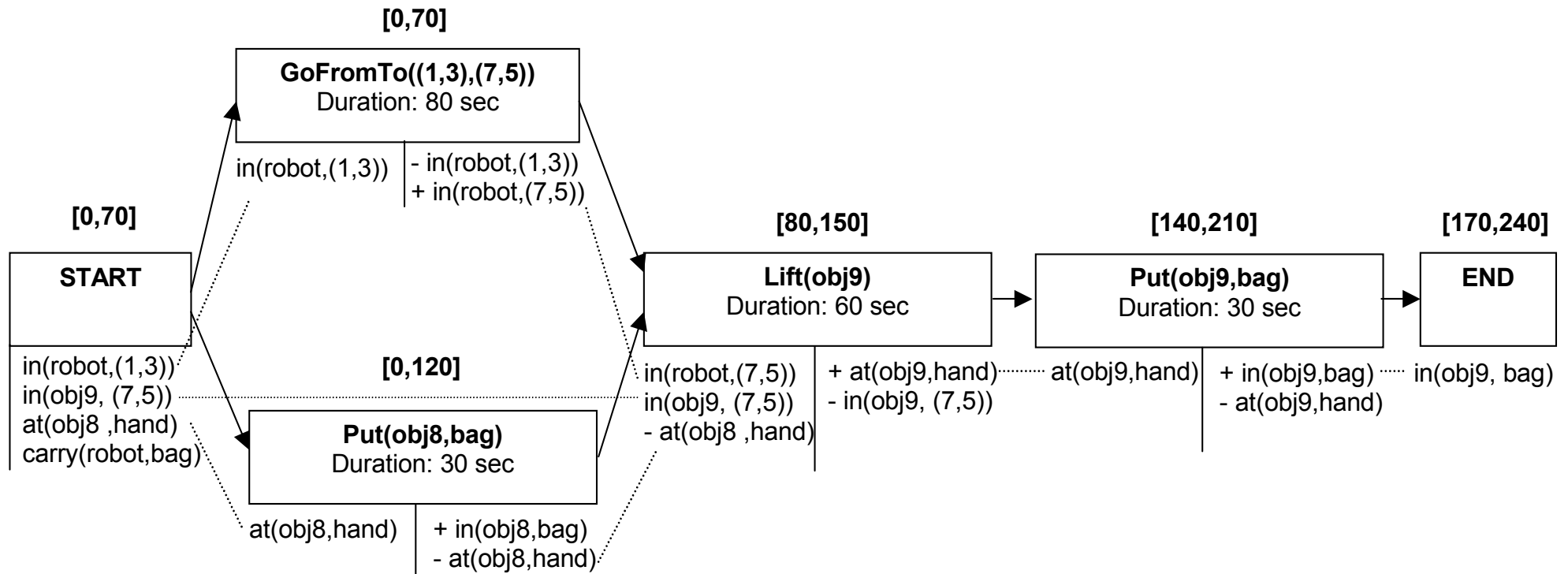
# Παράδειγμα Σχεδιασμού με Χρονικές Στιγμές

## Δημιουργία Πλάνου

- ❖ Αρχικά ισχύουν  $EST(IS)=0$  και  $LST(FS)=240$
- ❖ Η δημιουργία του πλάνου αρχίζει με την εισαγωγή της ενέργειας  $Put(obj9,bag)$  με  $LST(Put(obj9,bag))=210$ ,
  - ❑ Η ενέργεια διαρκεί 30 δευτερόλεπτα και  $EST(Put(obj9,bag))=0$
  - ❑ Ανανεώνεται το  $EST(FS)$  σε 210 και το  $LST(IS)$  σε 210
- ❖ Εισάγεται η ενέργεια  $Lift(obj9)$ 
  - ❑ Ανανεώνονται τα  $EST$  και  $LST$  όλων των ενεργειών στο ημιτελές πλάνο
- ❖ Οι ενέργειες  $GoTo(7,5)$  και  $Put(obj8,bag)$  εισάγονται στο πλάνο παράλληλα λόγω της αρχής της ελάχιστης δέσμευσης και δεν επηρεάζουν η μία την άλλη.
  - ❑ Η διάρκεια της ενέργειας  $GoFromTo((1,3),(7,5))$  είναι  $10 \cdot (|7-1|+|5-3|)=10 \cdot (6+2)=80$  δευτερόλεπτα.

# Παράδειγμα Σχεδιασμού με Χρονικές Στιγμές

## Τελικό πλάνο

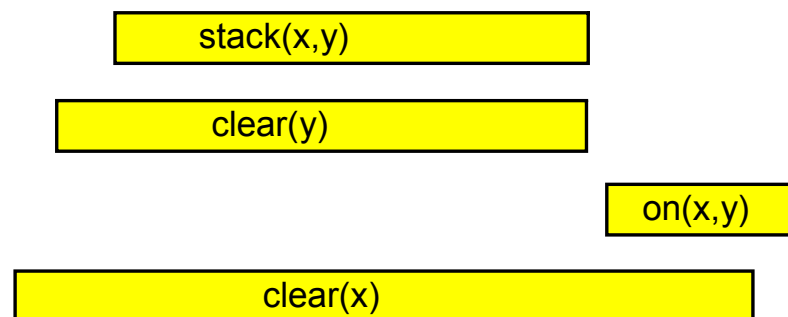


- ❖ Το νωρίτερο που μπορεί να αρχίσει η εκτέλεση του πλάνου είναι "αμέσως" ενώ το αργότερο είναι σε 70 δευτερόλεπτα.

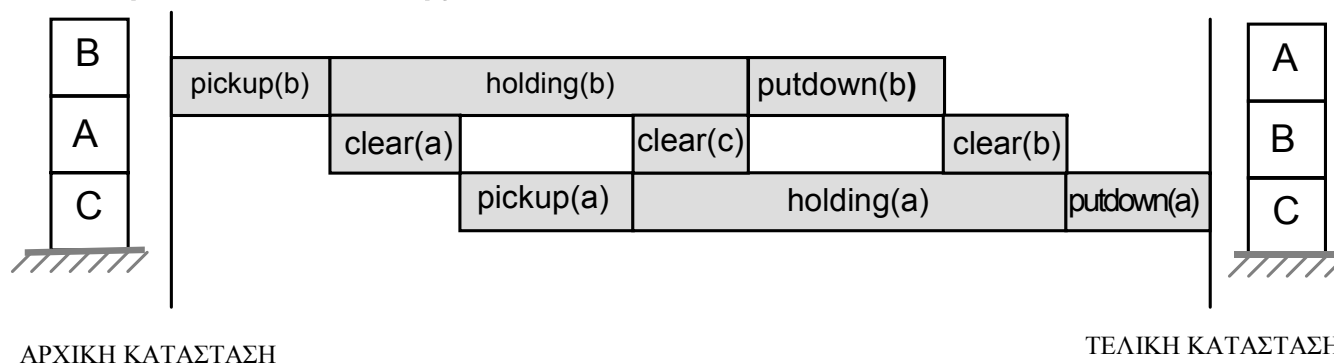
# Σχεδιασμός με Χρονικά Διαστήματα

- ❖ Οι ενέργειες περιέχουν και τους περιορισμούς των χρονικών διαστημάτων κάτω από τους οποίους πρέπει να ισχύουν οι προϋποθέσεις και τα αποτελέσματά τους.

- Ορισμός ενέργειας  $stack(x,y)$



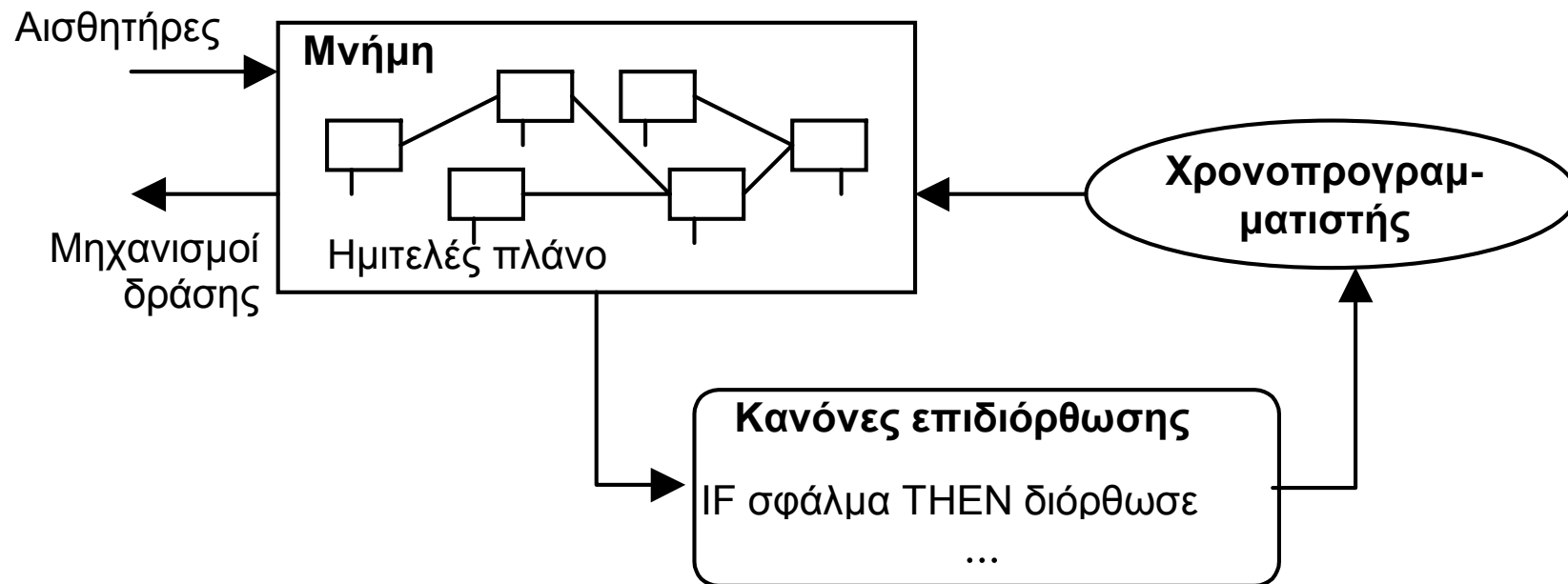
- Αναπαράσταση πλάνου λύσης



# Αλληλεπίδραση Δημιουργίας και Εκτέλεσης Πλάνων

## IPEM (Integrated Plan Execution & Monitoring)

- ❖ Η εκτέλεση ενός πλάνου παρουσιάζει δυσκολίες όταν ο κόσμος κατά την εκτέλεση διαφέρει από το μοντέλο του κόσμου κατά τη δημιουργία του πλάνου.



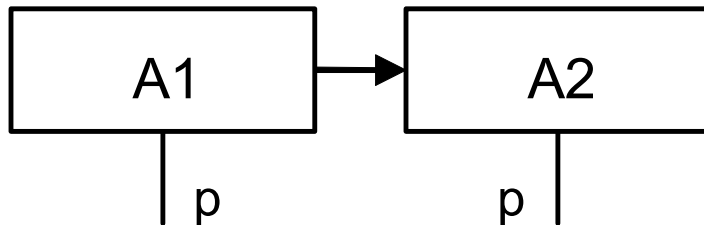
# Κανόνες Επιδιόρθωσης Σχεδιαστή ΙΡΕΜ

## Κανόνες Δημιουργίας Πλάνου (1/4)

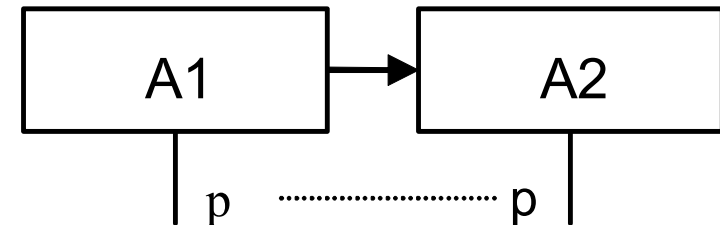
- ❖ **ΚΑΝΟΝΑΣ 1:** 1<sup>η</sup> μορφή: Υποστήριξη από προηγούμενη ενέργεια (reduction prior).

*if* μία προϋπόθεση δεν υποστηρίζεται *then* υποστήριξέ την

**if**



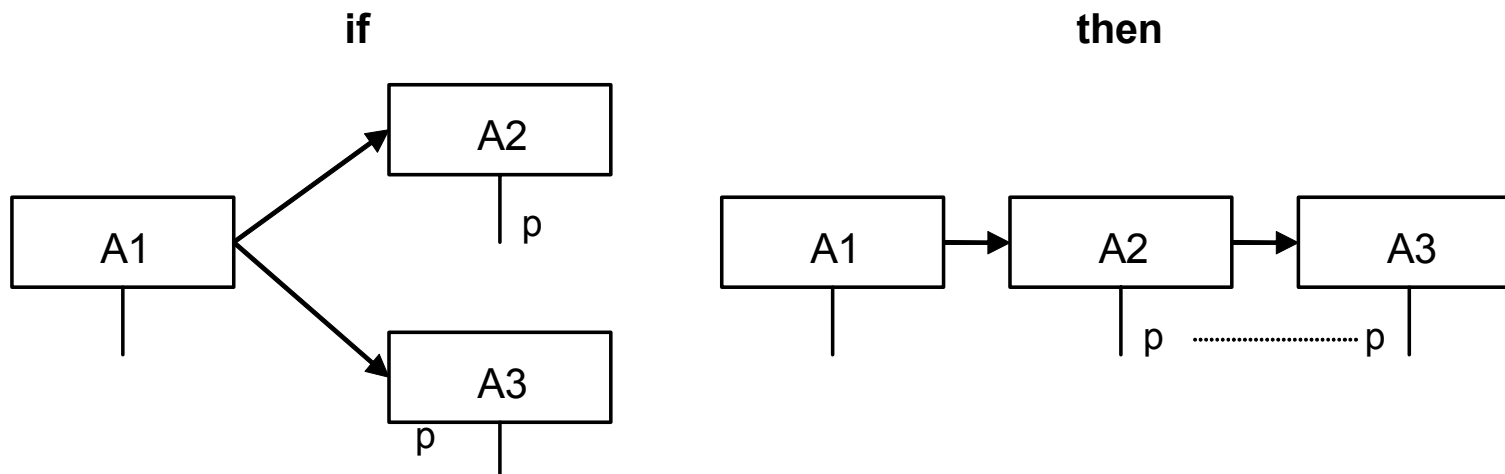
**then**



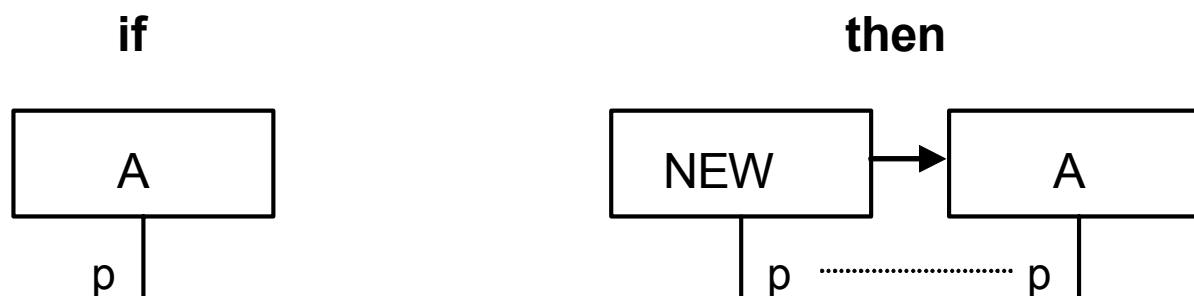
# Κανόνες Επιδιόρθωσης Σχεδιαστή ΙΡΕΜ

## Κανόνες Δημιουργίας Πλάνου (2/4)

- ❖ **ΚΑΝΟΝΑΣ 1, 2<sup>η</sup> μορφή:** Υποστήριξη από παράλληλη ενέργεια (reduction parallel)



- ❖ **ΚΑΝΟΝΑΣ 1, 3<sup>η</sup> μορφή:** Υποστήριξη με εισαγωγή ενέργειας (reduction new)



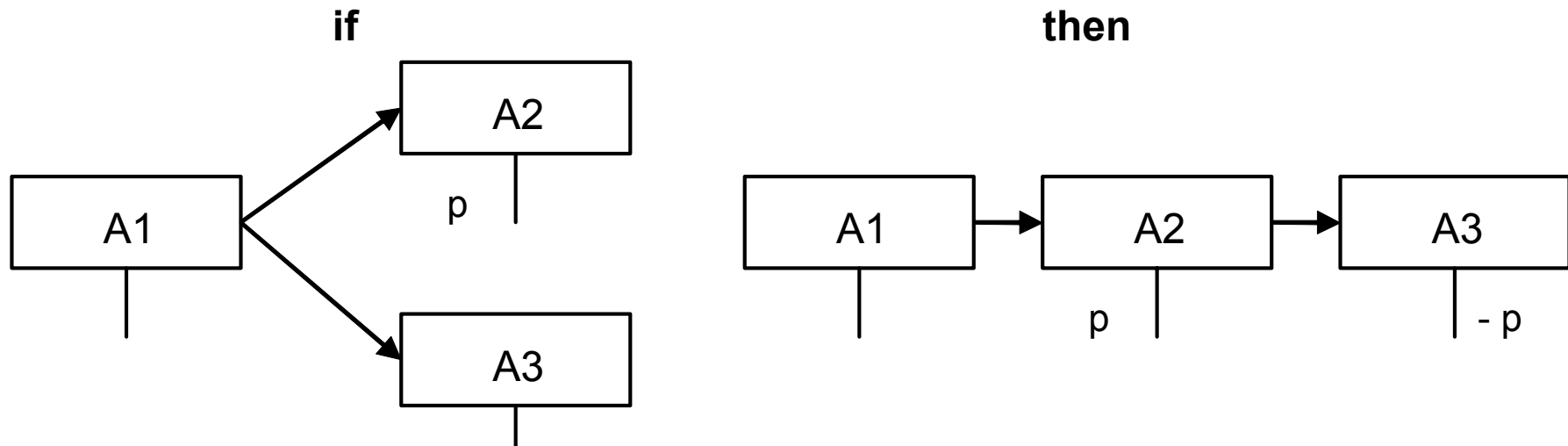


# Κανόνες Επιδιόρθωσης Σχεδιαστή ΙΡΕΜ

## Κανόνες Δημιουργίας Πλάνου (3/4)

❖ **ΚΑΝΟΝΑΣ 2:** Επιλύει συγκρούσεις προϋποθέσεων.

*if υπάρχουν συγκρούσεις προϋποθέσεων then γραμμικοποίηση*

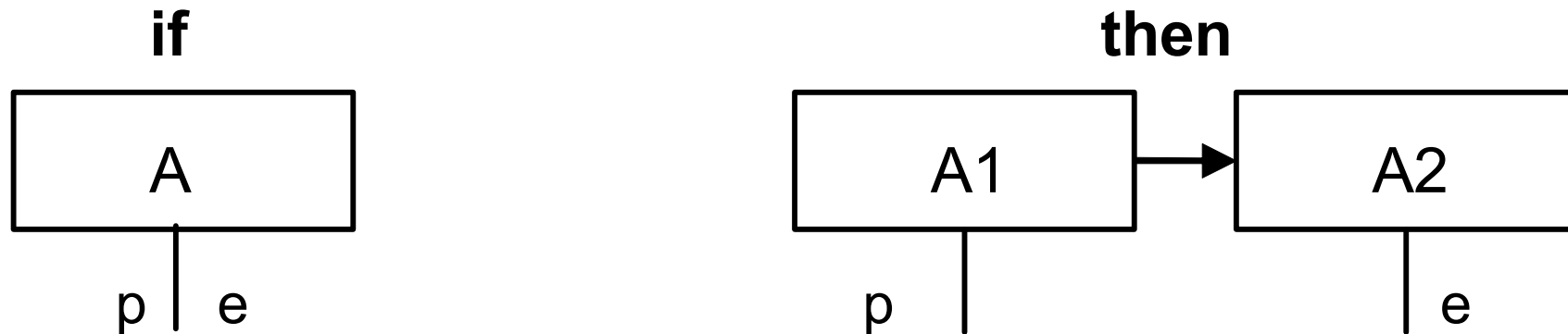


# Κανόνες Επιδιόρθωσης Σχεδιαστή ΙΡΕΜ

## Κανόνες Δημιουργίας Πλάνου (4/4)

### ❖ ΚΑΝΟΝΑΣ 3:

*if* μία ενέργεια μπορεί να επεκταθεί *then* επέκτεινε ενέργεια

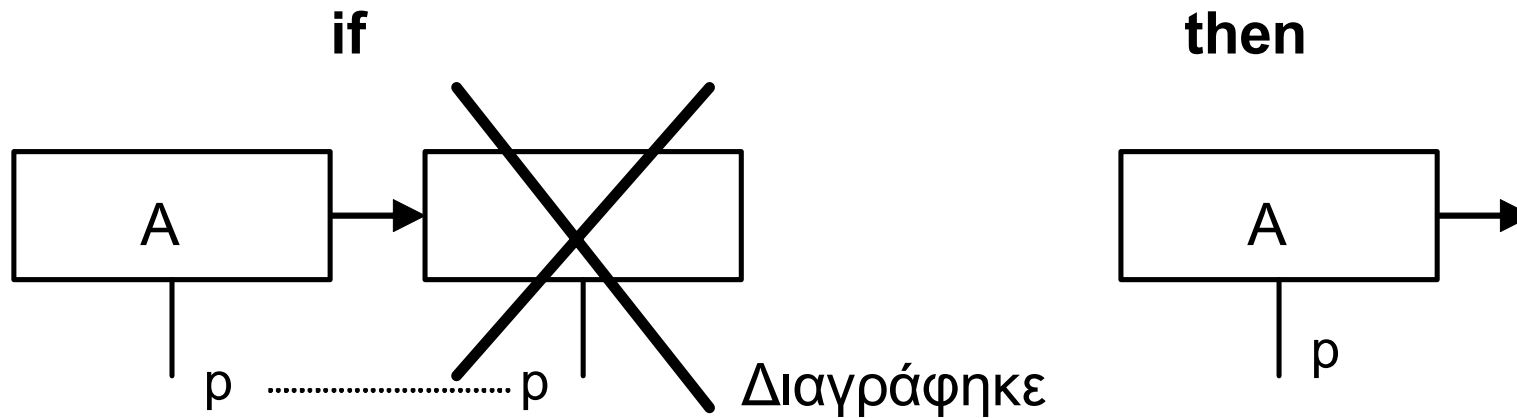


# Κανόνες Επιδιόρθωσης Σχεδιαστή ΙΡΕΜ

Κανόνες για εκτέλεση και έλεγχο πλάνου (1/4)

## ❖ ΚΑΝΟΝΑΣ 4:

*if* μία αιτιολογική σύνδεση δεν υποστηρίζεται *then* σβήσε την

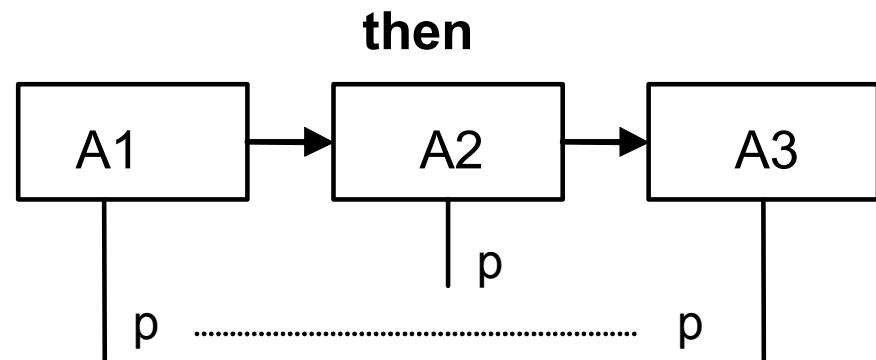
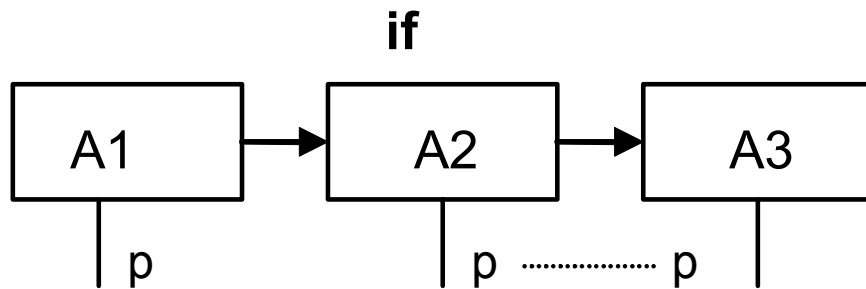


# Κανόνες Επιδιόρθωσης Σχεδιαστή ΙΡΕΜ

Κανόνες για εκτέλεση και έλεγχο πλάνου (2/4)

❖ **ΚΑΝΟΝΑΣ 5:**

*if* μία αιτιολογική σύνδεση μπορεί να επεκταθεί *then* επέκτεινε την

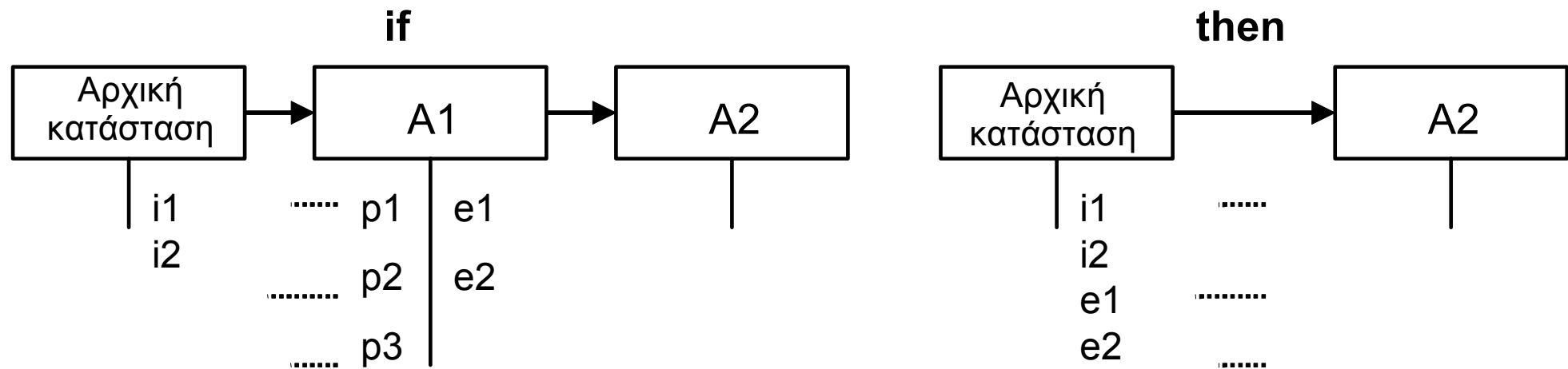


# Κανόνες Επιδιόρθωσης Σχεδιαστή ΙΡΕΜ

Κανόνες για εκτέλεση και έλεγχο πλάνου (3/4)

## □ ΚΑΝΟΝΑΣ 6.

*if* μία αρχέγονη ενέργεια μπορεί να εκτελεστεί *then* εκτέλεσέ την

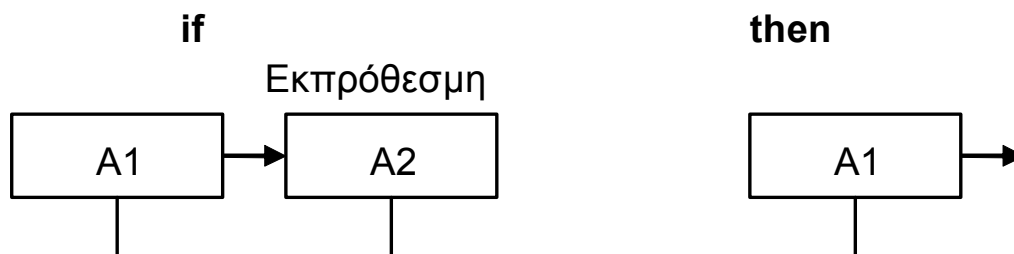


# Κανόνες Επιδιόρθωσης Σχεδιαστή ΙΡΕΜ

Κανόνες για εκτέλεση και έλεγχο πλάνου (4/4)

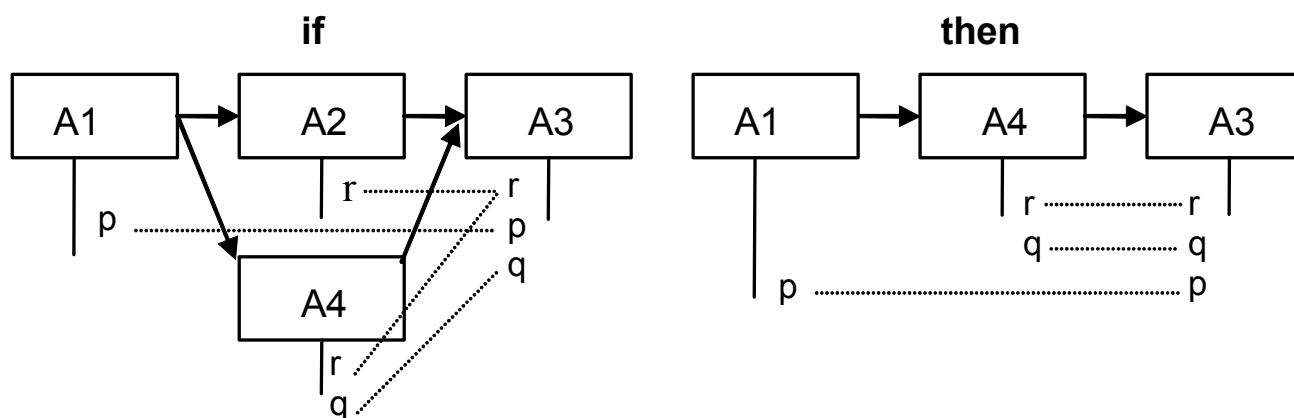
## ❖ ΚΑΝΟΝΑΣ 7:

*if* μια ενέργεια είναι εκπρόθεσμη *then* σβήσε ενέργεια



## ❖ ΚΑΝΟΝΑΣ 8:

*if* υπάρχει περιττή ενέργεια *then* σβήσε την ενέργεια



# Κύκλος Λειτουργίας ΙPEM

- ❖ Επανάλαβε:
  - A. Βρες όλα τα σφάλματα του ημιτελούς πλάνου.
  - B. Διάλεξε το σφάλμα με την υψηλότερη προτεραιότητα.
  - C. Σύμφωνα με τον αντίστοιχο κανόνα κάνε την επιδιόρθωση.
- ❖ Η προτεραιότητα των σφαλμάτων είναι στατική:
  - Ο χρόνος ενέργειας έχει παρέλθει.
  - Η ενέργεια είναι περιττή.
  - Η αιτιολογική σύνδεση δεν έχει επεκταθεί.
  - Υπάρχει απειλή.
  - Υπάρχει προϋπόθεση που δεν υποστηρίζεται.
  - Υπάρχει αρχέγονη ενέργεια που μπορεί να εκτελεστεί.
  - Υπάρχει ενέργεια που μπορεί να επεκταθεί.