

ΚΕΦΑΛΑΙΟ 15

Βασικές Αρχές και Τεχνικές Σχεδιασμού

Τεχνητή Νοημοσύνη - Β' Έκδοση

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

Σχεδιασμός Ενεργειών (Planning)

- ❖ **Προβλήματα σχεδιασμού ενεργειών** (planning problems) είναι αυτά στα οποία είναι πλήρως γνωστή η τελική κατάσταση και επιδιώκεται η εύρεση μιας ακολουθίας ενεργειών, μέσω της διαδικασίας του σχεδιασμού ενεργειών (planning).
- ❖ Η ακολουθία των ενεργειών που αποτελεί τη λύση ενός προβλήματος σχεδιασμού, ονομάζεται **πλάνο** (plan) ενώ το πρόγραμμα που την παράγει ονομάζεται **σχεδιαστής** (planner).
- ❖ Οι κλασσικοί αλγόριθμοι αναζήτησης είναι ανεπαρκείς για την αντιμετώπισή τους.
- ❖ Σημαντικό θέμα είναι ο ορισμός μίας γλώσσας περιγραφής προβλημάτων, που να υποστηρίζει την εφαρμογή κατάλληλων αλγορίθμων.

Θέματα που θα εξεταστούν

- ❖ Αναπαράσταση προβλημάτων- Το μοντέλο STRIPS
- ❖ Σχεδιασμός με Αναζήτηση στο Χώρο των Καταστάσεων
- ❖ Σχεδιασμός με Αναζήτηση στο Χώρο των Πλάνων
- ❖ Εκτέλεση Πλάνων από Πράκτορες Σχεδιαστές

Αναπαράσταση Προβλημάτων

- ❖ Ένα πρόβλημα σχεδιασμού ορίζεται από τρεις περιγραφές:
 - ❑ Της αρχικής κατάστασης (*Initial*).
 - ❑ Των στόχων (Goals).
 - ❑ Των διαθέσιμων ενεργειών (*Actions*).

- ❖ Η προτασιακή λογική δεν μπορεί να εκφράσει γενικότητα,
- ❖ Η κατηγορηματική λογική πρώτης τάξης δεν μπορεί να περιγράψει ενέργειες με μη προκαθορισμένα αποτελέσματα.

- ❖ Όσο πιο εκφραστική είναι μια αναπαράσταση, τόσο δυσκολότερη είναι η κατασκευή ενός συστήματος σχεδιασμού για την αντιμετώπιση προβλημάτων, ενώ παράλληλα αυξάνεται και ο χρόνος για την επίλυσή τους.

Το Μοντέλο STRIPS

Stanford Research Institute Planning System

- ❖ Προτάθηκε το 1971, (Fikes και Nilsson), για ένα μικρό ρομπότ (Shakey), για την εκτέλεση απλών ενεργειών.
 - ❖ Απλό μοντέλο.
 - ❖ Με στοιχεία προτασιακής λογικής
 - ❖ Κατάλληλο για προβλήματα όπου δεν εμφανίζεται αβεβαιότητα.
-
- ❖ Στην αρχική του μορφή δεν υποστήριζε την αναπαράσταση χρονικών και άλλων περιορισμών.



Μοντέλο STRIPS

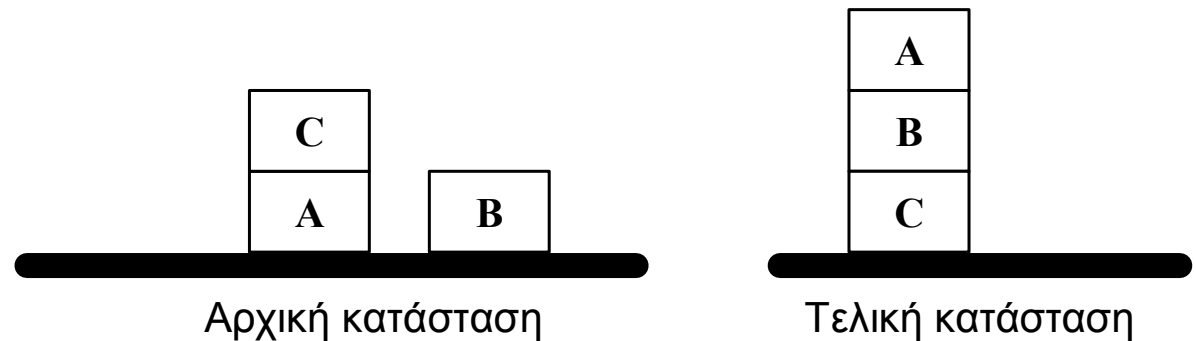
Παραδοχές

- ❖ Στο STRIPS γίνονται οι παρακάτω παραδοχές:
 - ❑ *Αδιαίρετες ενέργειες (indivisible actions)*
 - ❑ *Προκαθορισμένα αποτελέσματα (deterministic effects).*
 - ❑ *Πλήρης γνώση (omniscience).*
 - ❑ *Υπόθεση κλειστού συστήματος (closed world assumption).*
 - ❑ *Στατικός κόσμος (static world).*

Αναπαράσταση Καταστάσεων

❖ Στο STRIPS οι καταστάσεις ορίζονται σαν σύνολα από συγκεκριμένα **γεγονότα** (ή **προτάσεις**) που αληθεύουν.

❖ Παράδειγμα:



❖ Αρχική κατάσταση:

$\text{block}(a) \wedge \text{block}(b) \wedge \text{block}(c) \wedge \text{on}(a, \text{table}) \wedge \text{on}(c, a) \wedge \text{on}(b, \text{table}) \wedge \text{clear}(b) \wedge \text{clear}(c)$

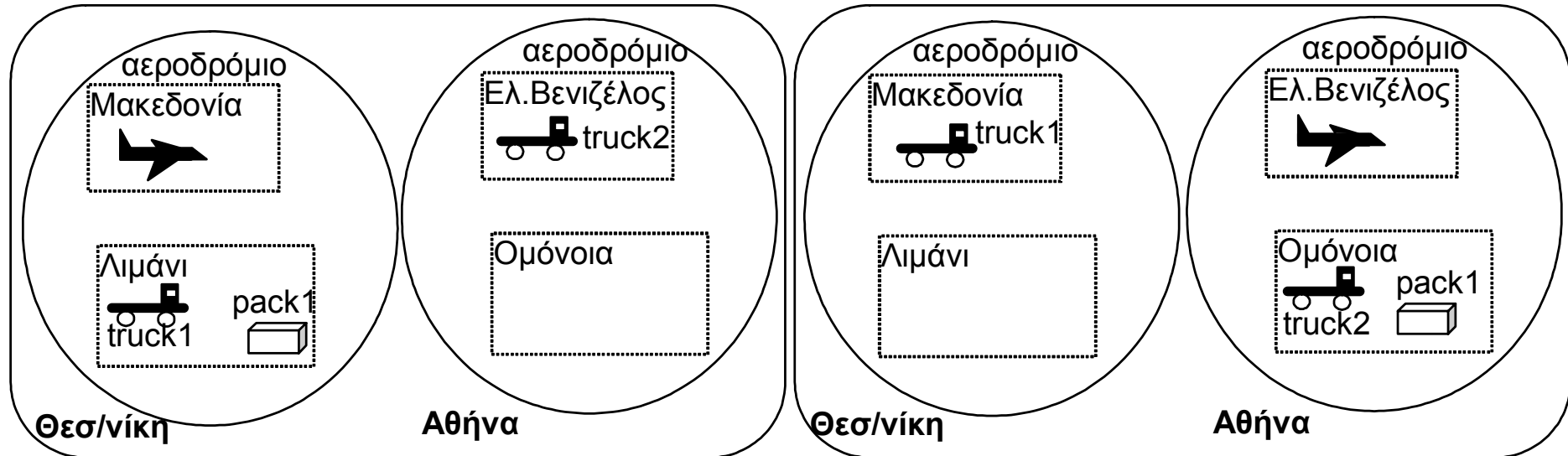
❖ Τελική κατάσταση:

$\text{on}(b, c) \wedge \text{on}(a, b)$

Παράδειγμα: Ένα τυπικό πρόβλημα μεταφοράς φορτίων σαν πρόβλημα εφοδιαστικής (logistics)

Αρχική Κατάσταση

Τελική Κατάσταση



- **Στόχος** είναι η μεταφορά του φορτίου από το κέντρο της Θεσσαλονίκης στο κέντρο των Αθηνών.

Παράδειγμα

Αναπαράσταση Καταστάσεων

❖ Αρχική κατάσταση:

- $city(thessalonini) \wedge location(harbor) \wedge location(makedonia) \wedge$
- $at_city(harbor,thessaloniki) \wedge airport(makedonia) \wedge truck(truck1) \wedge$
- $at(truck1, harbor) \wedge at(pack1, harbor) \wedge \dots$

❖ Τελική κατάσταση:

- $city(thessalonini) \wedge location(harbor) \wedge location(makedonia) \wedge$
- $at_city(harbor,thessaloniki) \wedge airport(makedonia) \wedge truck(truck1) \wedge$
- $at(truck1, makedonia) \wedge at(pack1, omonoia) \wedge \dots$

Αναπαράσταση Ενεργειών

❖ Στην STRIPS αναπαράσταση, μια *ενέργεια* (action) a περιγράφεται με τρεις λίστες γεγονότων:

- ❑ Λίστα προϋποθέσεων (Precondition list, $Pre(a)$)
- ❑ Λίστα προσθήκης (Add list, $Add(a)$)
- ❑ Λίστα διαγραφής (Delete list, $Del(a)$)
 - $Del(a) \subseteq Pre(a)$.

Αναπαράσταση Ενεργειών: Παράδειγμα

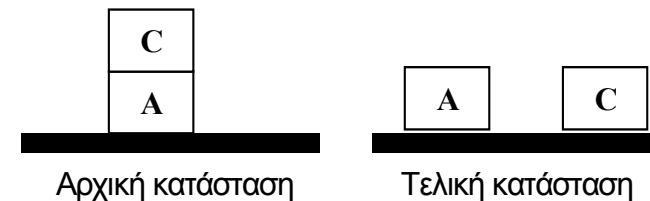
Όνομα ενέργειας	<code>move_C_from_A_to_table</code> μετακίνησε τον κύβο C από τον κύβο A στο τραπέζι
Λίστα προϋποθέσεων	<code>block(a), block(c), clear(c), on(c,a)</code>
Λίστα προσθήκης	<code>clear(a), on(c,table)</code>
Λίστα διαγραφής	<code>on(c,a)</code>

Αρχική Κατάσταση

`block(a)`
`block(c)`
`clear(c)`
`on(c,a)`
:
:

Τελική Κατάσταση

`block(a)`
`block(c)`
`clear(a)`
`on(c,table)`
:
:



Σχήματα Ενέργειών

- ❖ Οι ενέργειες, συνηθίζεται να ομαδοποιούνται σε *σχήματα ενεργειών* (*action schemas*) ή *τελεστές* (*operators*).
- ❖ Με ανάθεση τιμών στις μεταβλητές τους γίνονται κανονικές ενέργειες.

Παράδειγμα

Ενέργειες στον κόσμο των κύβων

Όνομα (σχήματος) ενέργειας	<code>move_X_from_Y_to_table</code> μετακίνησε έναν κύβο X από τον κύβο Y στο τραπέζι
Λίστα προϋποθέσεων	<code>block(X) , block(Y) , clear(X) , on(X,Y)</code>
Λίστα προσθήκης	<code>clear(Y) , on(X,table)</code>
Λίστα διαγραφής	<code>on(X,Y)</code>

Παράδειγμα

Ενέργειες στη μεταφορά φορτίων

❖ Στο πρόβλημα των φορτίων οι επιτρεπτές ενέργειες είναι:

- Φόρτωσε το φορτίο στο φορτηγό.
- Ξεφόρτωσε το φορτίο από το φορτηγό.
- Φόρτωσε το φορτίο στο αεροπλάνο.
- Ξεφόρτωσε το φορτίο από το αεροπλάνο.
- Μετακίνησε το φορτηγό.
- Μετακίνησε το αεροπλάνο.

Όνομα σχήματος ενέργειας	<i>load_truck(T,P,L)</i> (φόρτωσε το φορτίο P στο φορτηγό T στην τοποθεσία L)
Λίστα προϋποθέσεων	<i>package(P), truck(T), location(L), at(T,L), at(P,L)</i>
Λίστα προσθηκών	<i>in(P,T)</i>
Λίστα διαγραφών	<i>at(P,L)</i>

Πρόβλημα σχεδιασμού (Planning problem)

Ορισμός

- ❖ Ένα πρόβλημα σχεδιασμού P αναπαριστάται με $P=(Actions, Initial, Goals)$,
- ❖ Το πρόβλημα συνίσταται στην εύρεση ενός πλάνου a_1, a_2, \dots, a_N .
- ❖ Για να είναι εφαρμόσιμη μια ενέργεια a σε μια κατάσταση S πρέπει να ισχύει:
 - $Pre(a) \subseteq S$
- ❖ Η κατάσταση S' που προκύπτει μετά την εφαρμογή της a ορίζεται ως:
 - $S' = result(S, a) = S - Del(a) \cup Add(a)$
- ❖ Οι ακολουθίες ενεργειών ονομάζονται *πλάνα* (*plans*).
 - Ένα πλάνο το οποίο μπορεί να εφαρμοστεί στην αρχική κατάσταση ονομάζεται *έγκυρο πλάνο* (*valid plan*).
 - Ένα έγκυρο πλάνο το οποίο πετυχαίνει τους στόχους ονομάζεται *λύση* (*solution*) του προβλήματος σχεδιασμού.
- ❖ Ένα πρόβλημα σχεδιασμού μπορεί να έχει μία ή περισσότερες ή καμία λύση.

Διαγραμματική Αναπαράσταση πλάνων (1/2)

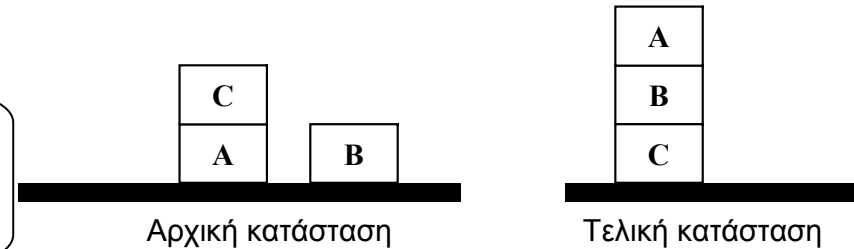
❖ Διαγραμματική αναπαράσταση ενέργειας:

move_C_from_A_to_table

block(c)		- on(c,a)
block(a)		+ on(c,table)
on(c,a)		
clear(c)		

Προϋποθέσεις

Αποτελέσματα
+ Add
- Del



❖ Διαγραμματική αναπαράσταση πλάνου με δίκτυο ενεργειών (*procedural network*).

move_C_from_A_to_table

block(c)		- on(c,a)
block(a)		+ on(c,table)
on(c,a)		+ clear(a)
clear(c)		

move_B_from_table_to_C

block(b)		- on(table,b)
block(c)		- clear(c)
on(table,b)		+ on(b,c)
clear(b)		
clear(c)		

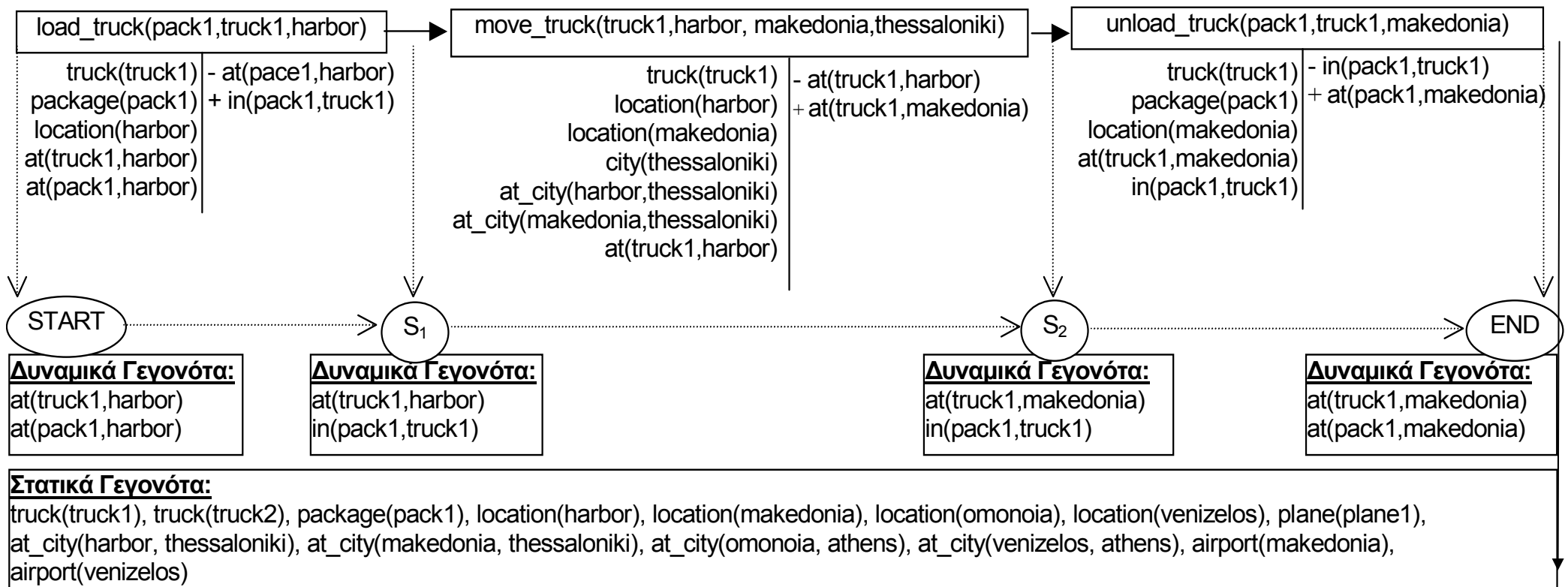
move_A_from_table_to_B

block(a)		- on(table,a)
block(b)		- clear(b)
on(table,a)		+ on(a,b)
clear(a)		
clear(b)		



Παράδειγμα: Πλάνα στη μεταφορά φορτίων

- ❖ Πλάνο τριών ενεργειών, το οποίο μετακινεί το φορτίο *pack1* από τη θέση *harbor* της Θεσσαλονίκης στη θέση αεροδρόμιο *makedonia* της Θεσσαλονίκης, με τη χρήση του φορτηγού *truck1*.



Διαγραμματική Αναπαράσταση πλάνων (2/2)

- ❖ Διαγραμματική αναπαράσταση πλάνου με ραβδόγραμμα (Gantt bar chart).



- ❖ **Γραμμικό πλάνο** (*linear plan*): Υπάρχει αυστηρή διαδοχή των ενεργειών.
- ❖ **Μη-γραμμικό πλάνο** (*non-linear plan*): Δεν υπάρχει αυστηρή διαδοχή ενεργειών, αλλά υπάρχει η δυνατότητα δύο ενέργειες να εκτελούνται παράλληλα.

Αναπαράσταση STRIPS

Μειονεκτήματα

- ❖ Δεν αναφέρει το χρόνο κατά τον οποίο ισχύουν τα γεγονότα.
- ❖ Δεν μπορεί να περιγράψει συνεχείς μεταβολές.
- ❖ Θεωρεί πλήρη βεβαιότητα για την ισχύ των γεγονότων.
- ❖ Δεν είναι πλήρης.

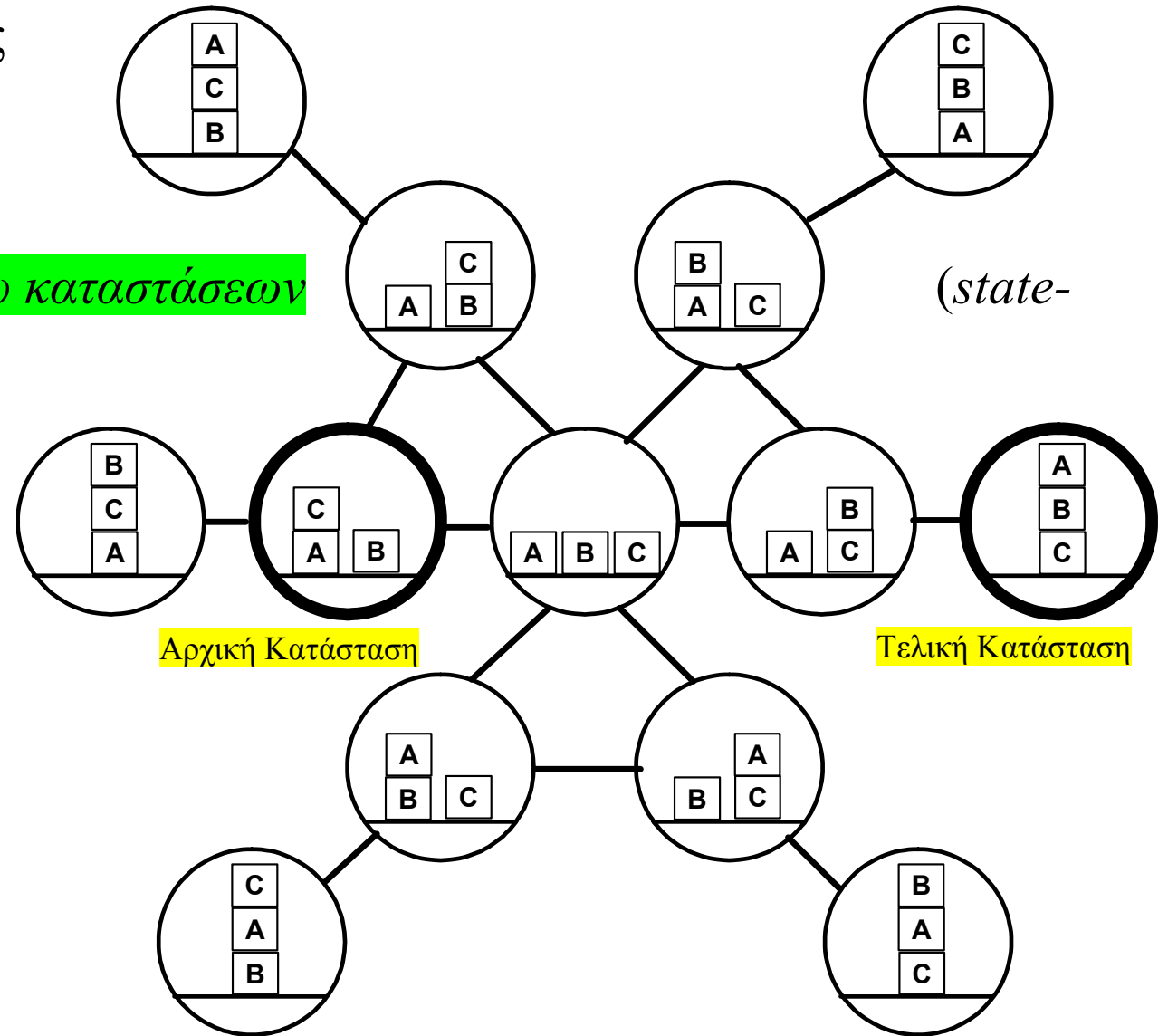
Σχεδίαση με λογισμό καταστάσεων

Situation Calculus

- ❖ Επικρατούσε πριν από το σχεδιαστή STRIPS.
- ❖ Κάθε ενέργεια έπρεπε να ορίζει με σαφήνεια ολόκληρη την κατάσταση που θα έχει ο κόσμος μετά την εκτέλεσή της.
- ❖ Για κάθε ενέργεια γράφονταν πάρα πολλά αξιώματα, τα λεγόμενα αξιώματα του πλαισίου (frame axioms), τα οποία καθόριζαν ποιες από τις προτάσεις του πλαισίου του προβλήματος παρέμεναν ανεπηρέαστες κατά την εκτέλεση της ενέργειας (πρόβλημα του πλαισίου - frame problem).
- ❖ Η αναπαράσταση των ενεργειών στον STRIPS θεωρεί ότι όλες οι υπόλοιπες προτάσεις παραμένουν ανεπηρέαστες.

Σχεδιασμός με Αναζήτηση στο Χώρο των Καταστάσεων

- ❖ Χρησιμοποίηση ενός από τους γνωστούς αλγορίθμους αναζήτησης.
- ❖ Οι αντίστοιχοι σχεδιαστές ονομάζονται **σχεδιαστές χώρου καταστάσεων** (*state-space planners*).



Είδη Διάσχισης

- ❖ Ορθή διάσχιση (progression)
- ❖ Ανάστροφη διάσχιση (regression)
 - ❑ Αναφέρεται και ως: *Τεχνική ανάλυσης των μέσων και των στόχων* (means-ends analysis):
- ❖ Διάσχιση Διπλής Κατεύθυνσης (bi-directional)

Ορθή Διάσχιση

- ❖ Αρχίζοντας από την αρχική κατάσταση εφαρμόζονται όλες οι ενέργειες που μπορούν να εφαρμοστούν και δημιουργούν νέες καταστάσεις.
- ❖ Έστω μια αρχική κατάσταση (I), ένα σύνολο στόχων (G) και ένα σύνολο ενεργειών.
- ❖ Επιλέγεται μια ενέργεια a της οποίας οι προϋποθέσεις της εμπεριέχονται (είναι υποσύνολο) στην αρχική κατάσταση IS.
- ❖ Ύστερα από την εφαρμογή της ενέργειας, προκύπτει μια νέα κατάσταση S:
 - ❑ $S = I - Del(a) \cup Add(a)$
- ❖ Η διαδικασία εφαρμόζεται επαναληπτικά στη νέα κατάσταση S, μέχρις ότου βρεθεί μια κατάσταση που είναι υπερσύνολο (εμπεριέχει) των στόχων.

Ανάστροφη Διάσχιση

- ❖ Στην περίπτωση αυτή, η διάσχιση γίνεται από τους στόχους προς την αρχική κατάσταση.
- ❖ Έστω μια αρχική κατάσταση (I), ένα σύνολο στόχων (G) και ένα σύνολο ενεργειών.
- ❖ Επιλέγεται μια ενέργεια a , τέτοια ώστε κανένα από τα γεγονότα που αυτή διαγράφει να μην εμφανίζεται στην τελική κατάσταση, ενώ πρέπει να εμφανίζεται τουλάχιστον ένα από τα γεγονότα που αυτή προσθέτει.
 - $Del(a) \cap G = \emptyset$ και $Add(a) \cap G \neq \emptyset$
- ❖ Το σύνολο των στόχων αναθεωρείται:
 - $G' = Pre(a) \cup G - Add(a)$
- ❖ Η διαδικασία εφαρμόζεται επαναληπτικά στο νέο σύνολο στόχων (G'), μέχρις ότου βρεθεί ένα σύνολο γεγονότων που να είναι υποσύνολο της αρχικής κατάστασης.
- ❖ Τα σημεία επιλογής των ενεργειών είναι σημεία οπισθοδρόμησης (backtracking points).

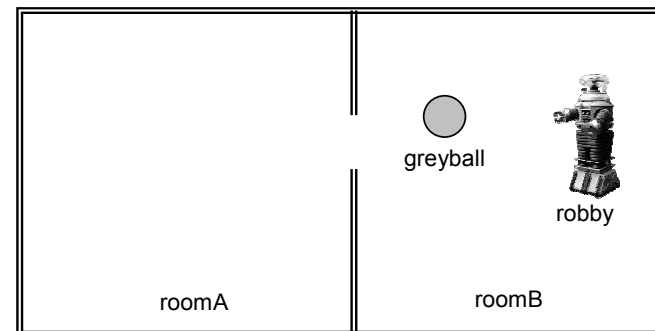
Παράδειγμα

Ένα πρόβλημα κίνησης και λαβής από ρομπότ

:

START={robot(robbby) \wedge room(roomA) \wedge room(roomB) \wedge
ball(greyball) \wedge at(robbby,roomB) \wedge at(greyball,roomB) \wedge
free(robbby) }

END = {at(greyball,roomA) }.



❖ Οι τελεστές είναι:

move (R, X, Y)

robot(R)	- at(R,X)
room(X)	+ at(R,Y)
room(Y)	
at(R,X)	

pick_ball (R,B,X)

robot(R)	- at(B,X)
ball(B)	- free(R)
room(X)	+ has(R,B)
at(R,X)	
at(B,X)	
free(R)	

drop_ball (R,B,X)

robot(R)	- has(R,B)
ball(B)	+ at(B,X)
room(X)	+ free(R)
at(R,X)	
has(R,B)	

Επίλυση με ορθή διάσχιση (1/2)

❖ **Βήμα 1:** Καθώς $prec(A_1) \subseteq START$ και $prec(A_2) \subseteq START$, μπορεί να εφαρμοστούν οι εξής δύο ενέργειες:

- $A_1 = move(robby, roomB, roomA)$
- $A_2 = pick_ball(robby, greyball, roomB)$

□ Έστω ότι ο αλγόριθμος αναζήτησης επιλέγει την ενέργεια A_1 :

- $S_A = START - del(A_1) \cup add(A_1)$
 $= \{ robot(robby) \wedge room(roomA) \wedge room(roomB) \wedge ball(greyball) \wedge at(robby, roomA) \wedge at(greyball, roomB) \wedge free(robby) \}$

❖ **Βήμα 2:** Στην κατάσταση S_A μπορεί να εφαρμοστεί μόνο η ενέργεια:

- $A_3 = move(robby, roomA, roomB)$

□ από την οποία προκύπτει η κατάσταση:

- $S_B = \{ robot(robby) \wedge room(roomA) \wedge room(roomB) \wedge ball(greyball) \wedge at(robby, roomB) \wedge at(greyball, roomB) \wedge free(robby) \}$

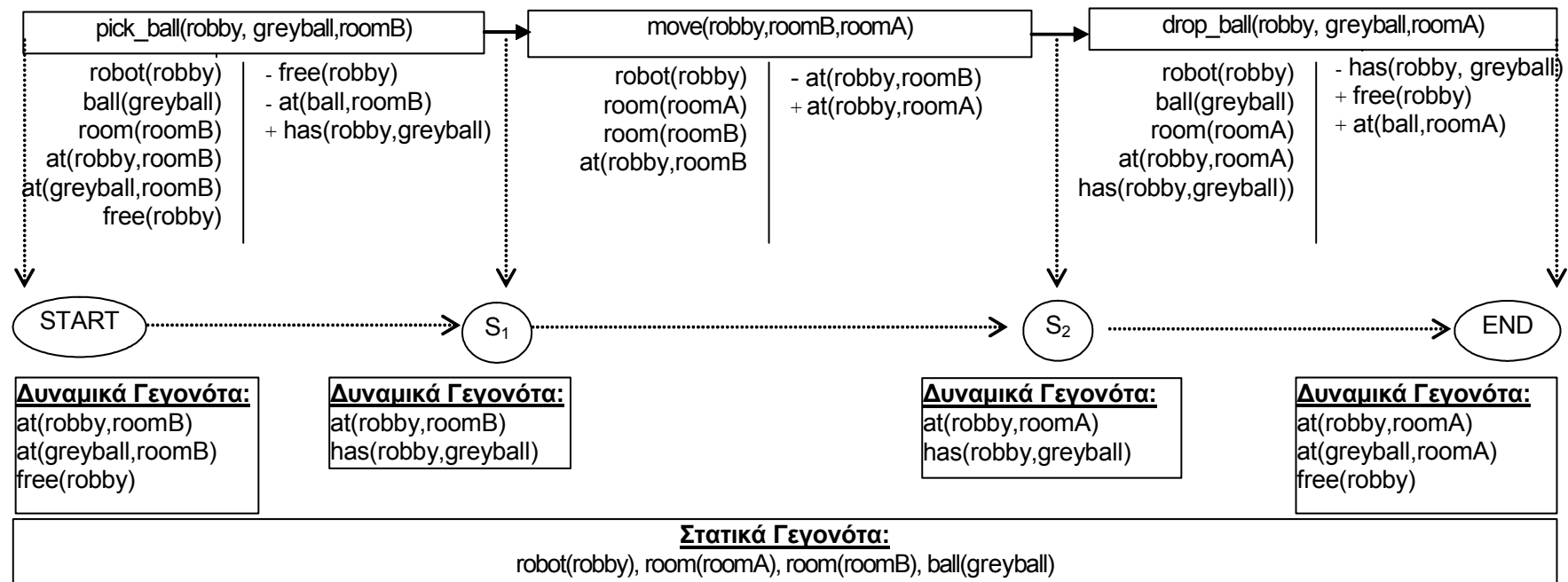
Επίλυση με ορθή διάσχιση (2/2)

❖ **Βήμα 3:** Ο αλγόριθμος επιλέγει την ενέργεια A_2 και προκύπτει η κατάσταση:

- $S_C = START - del(A_2) \cup add(A_2)$
 $= \{ robot(robby) \wedge room(roomA) \wedge room(roomB) \wedge ball(greyball) \wedge at(robby, roomA) \wedge has(robby, greyball) \}$

□ Η ίδια διαδικασία επαναλαμβάνεται μέχρι να βρεθεί κατάσταση S_F για την οποία να ισχύει $END \subseteq S_F$.

❖ Ένα πλάνο είναι:



Επίλυση με ανάστροφη διάσχιση (1/2)

- ❖ **Βήμα 1:** Η μοναδική ενέργεια που προσθέτει το στόχο $at(greyball, RoomA)$ είναι η:
 - $A_1 = drop_ball(robby, greyball, roomA)$
- Προκύπτει το νέο σύνολο στόχων G_A :
 - $G_A = Pre(A_1) \cup END-Add(A_1) = \{ at(robby, roomA), has(robby, greyball) \}$
- ❖ **Βήμα 2:** Στο σύνολο G_A μπορεί να εφαρμοστούν ανάστροφα οι ενέργειες:
 - $A_2 = move(robby, roomB, roomA)$
 - $A_3 = pick_ball(robby, greyball, roomB)$
 - $A_4 = pick_ball(robby, greyball, roomA)$.
- Έστω ότι επιλέγεται η ενέργεια A_3 , οπότε:
 - $G_B = \{ at(robby, roomA), at(robby, roomB), at(greyball, roomB), free(robby) \}$
- Εύκολα όμως προκύπτει ότι το G_B δεν είναι έγκυρο, καθώς τα γεγονότα:
 - $at(robby, roomA)$ και $at(robby, roomB)$
- είναι ασύμβατα μεταξύ τους. Ο αλγόριθμος οπισθοδρομεί στο προηγούμενο βήμα.

Επίλυση με ανάστροφη διάσχιση (2/2)

- ❖ **Βήμα 3:** Έστω ότι επιλέγεται η ενέργεια A_2 , οπότε προκύπτει:
 - $G_C = \{ at(Rrobby, roomB), has(robby, greyball) \}$
- ❖ Η ίδια διαδικασία επαναλαμβάνεται μέχρι να βρεθεί σύνολο στόχων, το οποίο να περιέχει όλα τα δυναμικά γεγονότα της αρχικής κατάστασης.
- ❖ Το πλάνο που τελικά θα προκύψει από την ανάστροφη διάσχιση, ταυτίζεται με αυτό της ορθής.

Σύγκριση Ορθής και Ανάστροφης Διάσχισης

- ❖ Οι δύο κατευθύνσεις εμφανίζουν την ίδια τάξη πολυπλοκότητας.
 - ❑ Οι δύο κατευθύνσεις θα εκτελέσουν τον ίδιο αριθμό επαναλήψεων.
 - ❑ Ωστόσο, σε μια πραγματική υλοποίηση δεν υπάρχει απόλυτη επιτυχία στην επιλογή των σωστών ενεργειών, οπότε αυτό το οποίο επηρεάζει σημαντικά τον όγκο της αναζήτησης είναι ο αριθμός των εφαρμόσιμων ενεργειών παράγοντας διακλάδωσης (branching factor) σε κάθε κατάσταση του χώρου καταστάσεων, οι οποίες και πρέπει να ελεγχθούν κατά την αναζήτηση.
 - ❑ Αν υποθεθεί ότι η μέση τιμή του είναι παράγοντα διακλάδωσης b , τότε η πολυπλοκότητα του προβλήματος της αναζήτησης είναι της τάξης του $O(bn)$, όπου n το μήκος της λύσης.

- ❖ Η ανάστροφη διάσχιση είναι συνήθως αποτελεσματικότερη.

Σχεδιασμός με Αναζήτηση στο Χώρο των Πλάνων

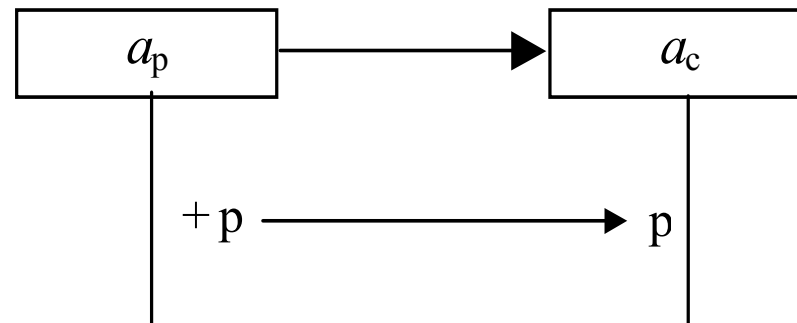
- ❖ Στο σχεδιασμό στο χώρο των καταστάσεων θεωρήθηκε ότι οι ενέργειες των πλάνων είναι πλήρως διατεταγμένες, τα δε πλάνα παράγονται προσθέτοντας νέες ενέργειες.
 - ❑ Τα πλάνα αυτά χαρακτηρίζονται ως γραμμικά πλάνα (linear plans) ή πλάνα πλήρους διάταξης (totally ordered plans).
- ❖ Η αναζήτηση στο χώρο των πλάνων (*plan space*) χρησιμοποιεί τους γνωστούς αλγορίθμους αναζήτησης, ωστόσο:
 - ❑ Το μέτωπο αναζήτησης περιέχει ημιτελή πλάνα.
 - ❑ Τα ημιτελή πλάνα είναι σύνολα από ενέργειες, όχι απαραίτητα συγκεκριμένες και όχι απαραίτητα πλήρως διατεταγμένες στο χρόνο.
 - ❑ Η αναζήτηση ξεκινά από κενό πλάνο ενώ η λύση είναι το πλάνο του τελικού κόμβου.
- ❖ **Αρχή της ελάχιστης δέσμευσης (least commitment principle):** Οι ενέργειες δεν τοποθετούνται σε συγκεκριμένες θέσεις στο χρόνο και οι μεταβλητές τους δε δεσμεύονται σε συγκεκριμένες τιμές αντικειμένων, εφόσον δε συντρέχει λόγος.
- ❖ Οι σχεδιαστές που αναζητούν λύσεις στο χώρο των πλάνων αποκαλούνται:
 - ❑ σχεδιαστές χώρου πλάνων (plan-space planners),
 - ❑ μη-γραμμικοί σχεδιαστές (non-linear planners),
 - ❑ σχεδιαστές μερικής διάταξης (partial order planners) αλλά και
 - ❑ σχεδιαστές ελάχιστης δέσμευσης (least-commitment planners).

Αναπαράσταση Μη-Γραμμικών Πλάνων

- ❖ Ένα μη-γραμμικό πλάνο με συγκεκριμένες ενέργειες ορίζεται ως μια τριάδα, (A, O, L) , όπου:
 - ❑ A είναι ένα σύνολο ενεργειών
 - ❑ O είναι ένα σύνολο περιορισμών διάταξης (ordering constraints)
 - ❑ L ένα σύνολο αιτιολογικών συνδέσεων (causal links)
- ❖ Για παράδειγμα, εάν $A = \{a_1, a_2, a_3\}$, τότε ένα πιθανό σύνολο περιορισμών διάταξης θα ήταν το $O = \{a_1 < a_3, a_2 < a_3\}$.
 - ❑ Το παραπάνω σύνολο περιορισμών διάταξης είναι συμβατό με τις πλήρεις διατάξεις $a_1 < a_2 < a_3$ αλλά και $a_2 < a_1 < a_3$.

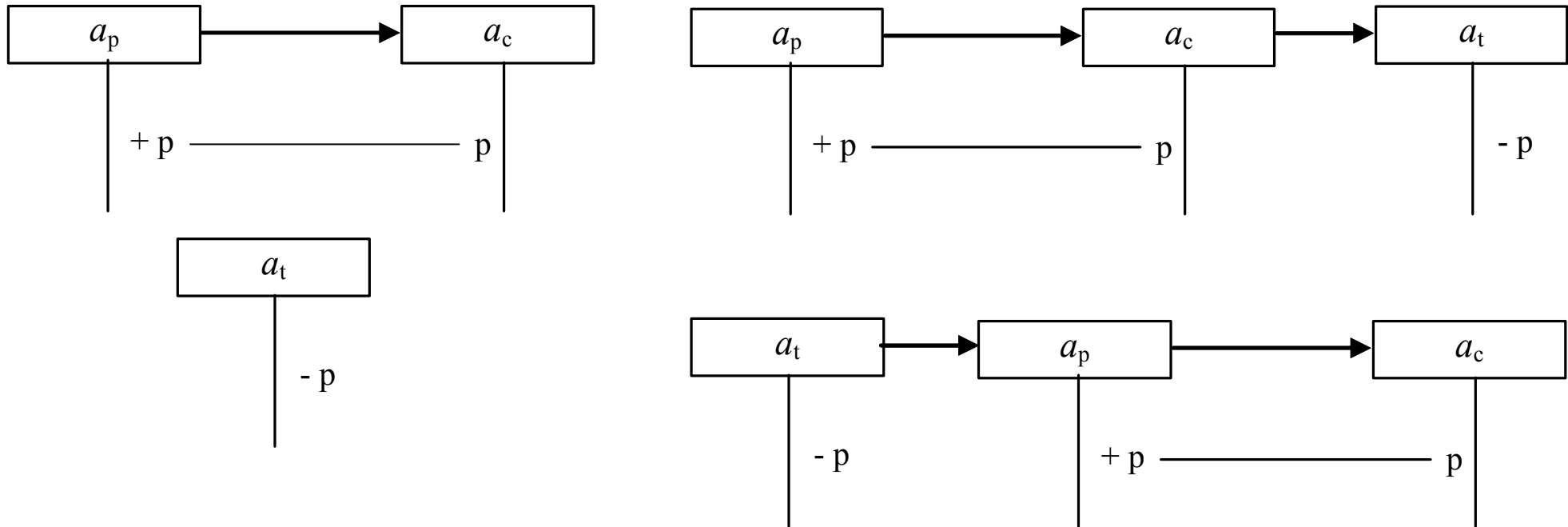
Αιτιολογικές Συνδέσεις

- ❖ Μια αιτιολογική σύνδεση είναι μια δομή με δύο δείκτες σε ενέργειες του πλάνου, έστω a_p και a_c , και ένα γεγονός p .
 - ❑ Μια αιτιολογική σύνδεση συμβολίζεται ως $a_p \xrightarrow{p} a_c$.
 - ❑ Το γεγονός p ανήκει τόσο στη λίστα προσθήκης της ενέργειας a_p , όσο και στη λίστα προϋποθέσεων της ενέργειας a_c .
 - ❑ Η ενέργεια a_p ονομάζεται *παραγωγός* (*producer*) της σύνδεσης, ενώ η ενέργεια a_c ονομάζεται *καταναλωτής* (*consumer*) της σύνδεσης.
 - ❑ Το L είναι το σύνολο όλων των αιτιολογικών συνδέσεων ενός πλάνου.
- ❖ Μια αιτιολογική σύνδεση δηλώνει ότι κατά το χρονικό διάστημα μεταξύ της εκτέλεσης της ενέργειας a_p και της ενέργειας a_c , το γεγονός p αληθεύει συνεχώς.
 - ❑ Για κάθε αιτιολογική σύνδεση της μορφής $a_p \xrightarrow{p} a_c$ του συνόλου L , ο περιορισμός διάταξης $a_p < a_c$ εισάγεται στο σύνολο O



Απειλές (Threats)

- ❖ Έστω μια αιτιολογική σύνδεση $a_p \xrightarrow{p} a_c$ και μια τρίτη ενέργεια $a_t \in A$. Η ενέργεια a_t αποτελεί απειλή για την αιτιολογική σύνδεση $a_p \xrightarrow{p} a_c$, εάν:
 - ❑ το σύνολο $O \cup \{a_p < a_t < a_c\}$ είναι συνεπές, και
 - ❑ $p \in \text{Del}(a_t)$
- ❖ Παράδειγμα απειλής και αντιμετώπισής της με προβιβασμό και υποβιβασμό



Μη Γραμμικά Πλάνα

- ❖ Ένα μη-γραμμικό πλάνο λέγεται *πλήρες (complete)*, όταν:
 - ❑ Κάθε ενέργεια που εμφανίζεται είτε σε κάποια αιτιολογική σύνδεση του συνόλου L ή σε κάποιον περιορισμό διάταξης του συνόλου O , αναφέρεται επίσης στο σύνολο των ενεργειών A .
 - ❑ Για κάθε ενέργεια $a \in A$ και για κάθε προϋπόθεση $p \in Pre(a)$, υπάρχει μια αιτιολογική σύνδεση της μορφής $b \xrightarrow{p} a$ στο σύνολο L , όπου $b \in A$.
 - ❑ Αν το πλάνο περιέχει μία αιτιολογική σύνδεση $b \xrightarrow{p} a$ και μια ενέργεια c που την απειλεί, στο σύνολο O υπάρχει είτε η διάταξη $c < b$ ή η $a < c$.
- ❖ Μία *τοπολογική διάταξη (topological sort)* ενός μη-γραμμικού πλάνου είναι μία γραμμική ακολουθία των ενεργειών του, τέτοια ώστε:
 - ❑ Η πρώτη ενέργεια στην ακολουθία είναι η **START**.
 - ❑ Η τελευταία ενέργεια στην ακολουθία είναι η **FINISH**.
 - ❑ Για κάθε αιτιολογική σύνδεση $b \xrightarrow{p} a$, η ενέργεια b προηγείται της ενέργειας a .
 - ❑ Για κάθε περιορισμό διάταξης $b < a$ του συνόλου O , η ενέργεια b προηγείται της a .

Αλγόριθμος Παραγωγής Μερικώς Διατεταγμένων Πλάνων

Αλγόριθμος POP((A, O, L) , $Agenda$)

1. Εάν $Agenda = \emptyset$, επέστρεψε το πλάνο (A, O, L) .
2. Έστω (q, a_{need}) ένα στοιχείο της $Agenda$ (προφανώς ισχύει $a_{need} \in A$ και $q \in Pre(a_{need})$).
3. Έστω a_{add} μια ενέργεια, τέτοια ώστε $q \in Add(a_{add})$. Η ενέργεια αυτή μπορεί είτε να είναι μια από τις ενέργειες του συνόλου A , τέτοια ώστε να μπορεί να διαταχθεί χρονικά πριν από την a_{need} , ή να είναι μια νέα ενέργεια. Εάν δεν υπάρχει τέτοια ενέργεια, τότε επέστρεψε αποτυχία. Θέσε $L' = L \cup \{a_{add} \xrightarrow{q} a_{need}\}$, $O' = O \cup \{a_{add} < a_{need}\}$. Εάν η ενέργεια a_{add} είναι μια νέα ενέργεια, τότε $A' = A \cup \{a_{add}\}$ και $O' = O \cup \{START < a_{add} < FINISH\}$, ειδάλλως $A' = A$.
4. Θέσε $Agenda' = Agenda - \{(q, a_{need})\}$. Εάν η ενέργεια a_{add} ήταν μια νέα ενέργεια, τότε για κάθε $q_i \in Pre(a_{add})$ πρόσθεσε το στοιχείο $\langle q_i, a_{add} \rangle$ στην $Agenda'$.
5. Για κάθε ενέργεια $a_t \in A'$, η οποία μπορεί να αποτελέσει απειλή για κάποια αιτιολογική σύνδεση $a_p \xrightarrow{q} a_c \in L'$, υπολόγισε το νέο σύνολο περιορισμών διάταξης O' , επιλέγοντας μια από τις παρακάτω δύο σχέσεις, ελέγχοντας ώστε το σύνολο O' να είναι συνεπές:

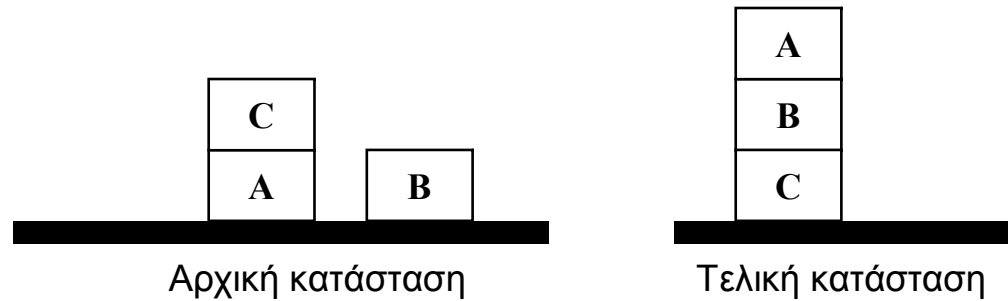
$$O' = O \cup \{a_t < a_p\}$$

$$O' = O \cup \{a_c < a_t\}$$

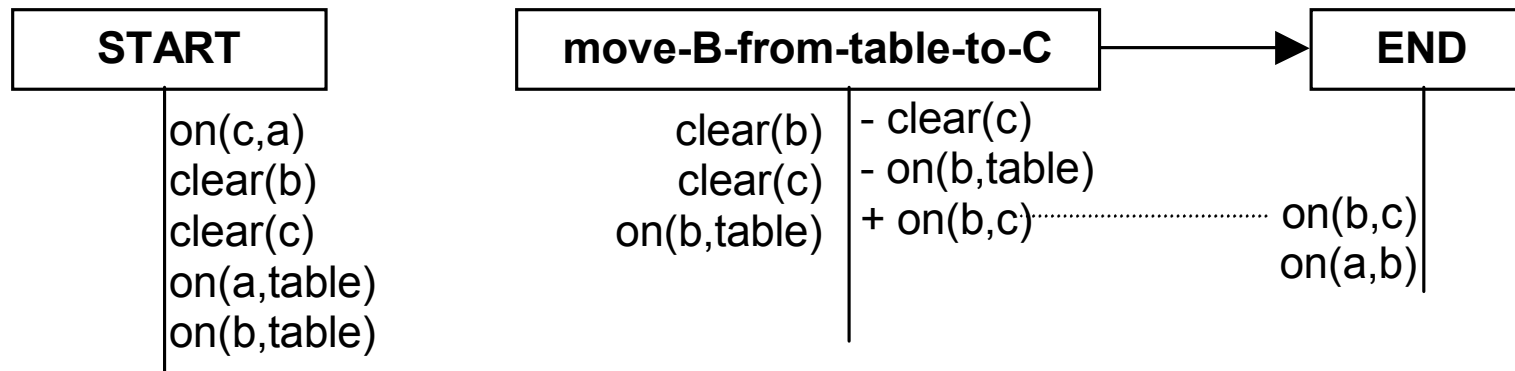
Εάν σε καμία από τις παραπάνω δύο περιπτώσεις το σύνολο O' είναι συνεπές, επέστρεψε αποτυχία.

6. POP((A', O', L') , $Agenda'$)

Παράδειγμα (1/3)

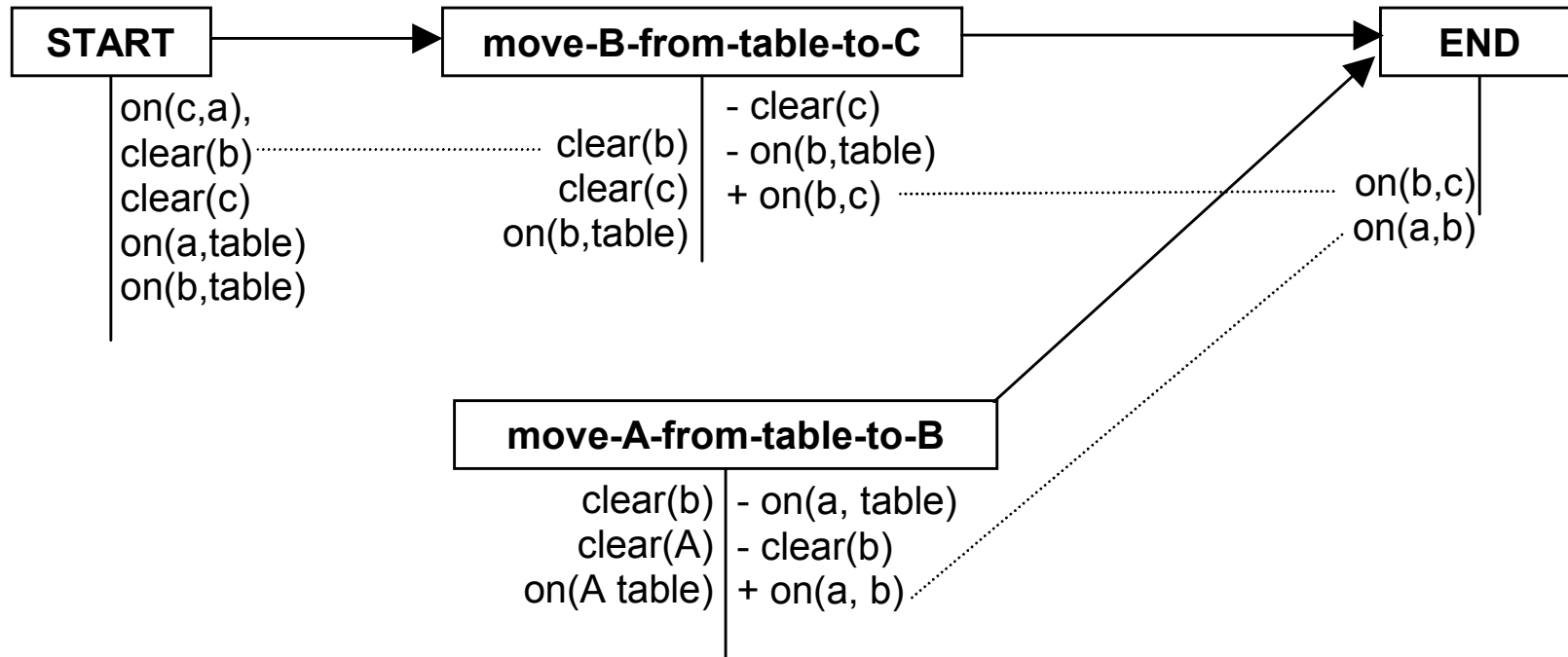


❖ Μερικό πλάνο με μια ενέργεια.



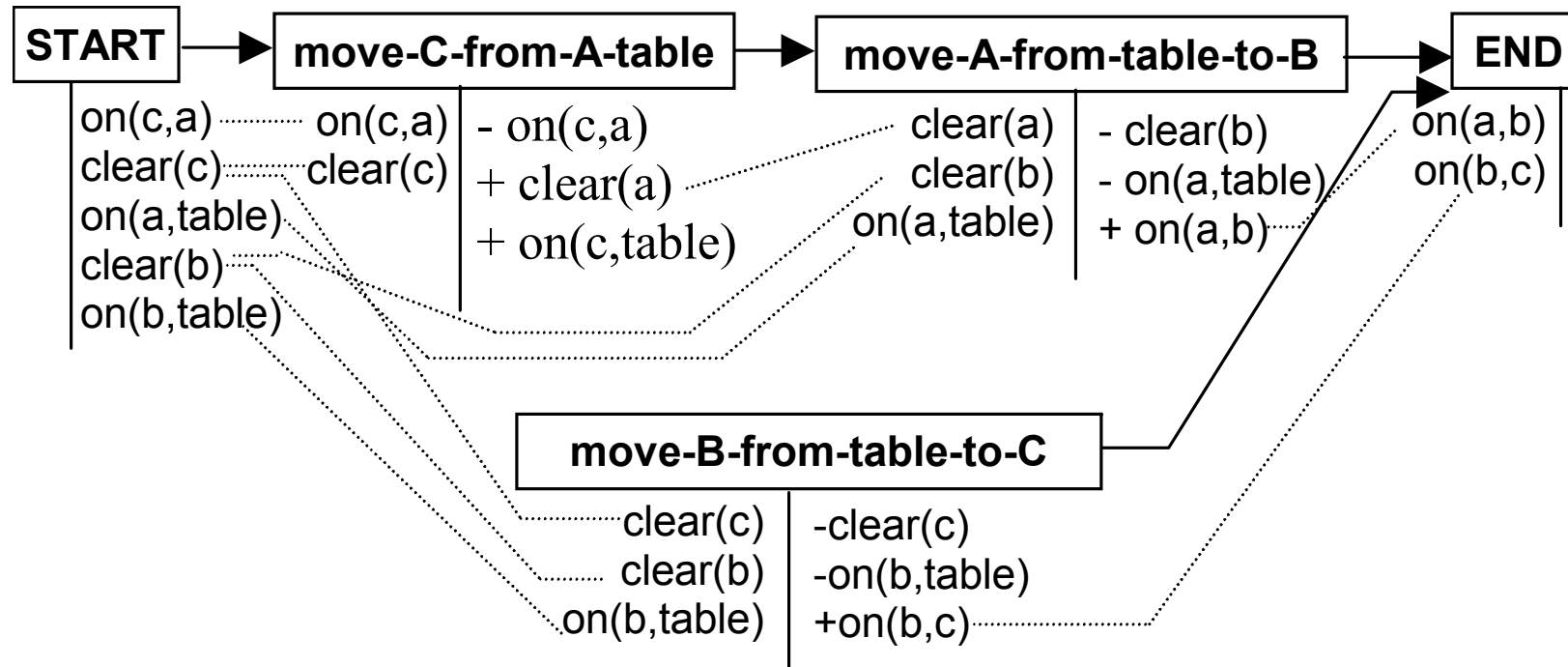
Παράδειγμα (2/3)

❖ Μερικό πλάνο με δύο ενέργειες.



Παράδειγμα (3/3)

❖ Ενδεχόμενο τελικό πλάνο



Χρήση Μη-Συγκεκριμένων Ενεργειών

- ❖ Κατά την επιλογή μιας ενέργειας δεσμεύονται μόνο οι μεταβλητές αυτής που είναι απαραίτητες για την ενοποίηση ενός γεγονότος που η ενέργεια παράγει με την αντίστοιχη μη υποστηριζόμενη προϋπόθεση μιας άλλης ενέργειας.

Εκτέλεση Πλάνων (1/2)

- ❖ Ένας σχεδιαστής αναλαμβάνει τη δημιουργία ενός πλάνου έχοντας σαν βάση την αναπαράσταση του κόσμου σε μία δεδομένη χρονική στιγμή, όπως την έχει δημιουργήσει μέσω κάποιων αισθητήρων (*sensors*).
- ❖ Το πλάνο μετά τη δημιουργία του μεταβιβάζεται στα απαραίτητα όργανα εκτέλεσης (*effectors*).
- ❖ Κατά τη διάρκεια δημιουργίας του πλάνου ο κόσμος δεν παραμένει απαραίτητα ο ίδιος.
- ❖ Για να εκτελεστεί μία ενέργεια πρέπει ο πραγματικός κόσμος να ανταποκρίνεται στις προϋποθέσεις της ενέργειας.
- ❖ **Εκτέλεση** σημαίνει πώς τα αποτελέσματα κάθε ενέργειας εμφανίζονται στον πραγματικό κόσμο.

Εκτέλεση Πλάνων (2/2)

❖ Σε περίπτωση αποτυχίας εκτέλεσης μιας ενέργειας μπορεί:

- ❑ Να επαναληφθεί η δημιουργία του πλάνου με νέα πλέον δεδομένα.
- ❑ Να εκτελεστεί ένα εναλλακτικό πλάνο που είναι εφαρμόσιμο ή το ίδιο πλάνο από διαφορετικό σημείο του.
- ❑ Να γίνει τροποποίηση του πλάνου στα σημεία όπου εμφανίζεται το πρόβλημα

Σχεδιαστής

