

Κεφάλαιο 16

Εξελιγμένες Τεχνικές Σχεδιασμού

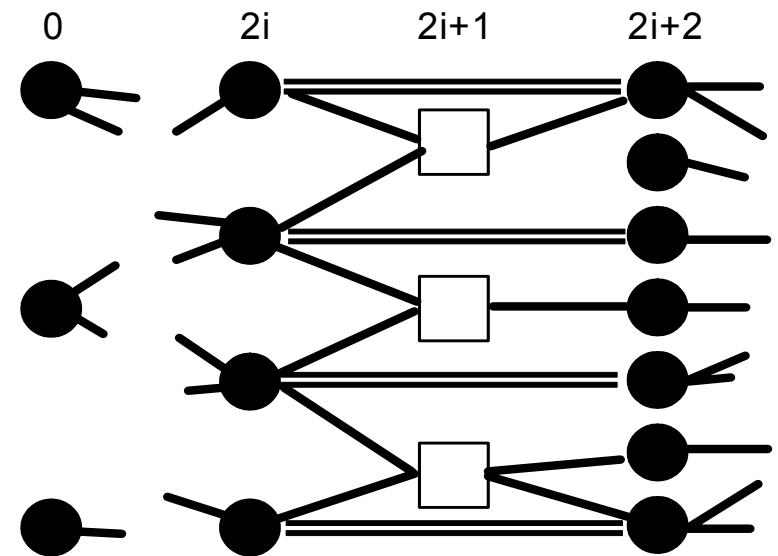
Τεχνητή Νοημοσύνη - Β' Έκδοση

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

Σχεδιασμός Βασισμένος σε Γράφους

Γράφος σχεδιασμού (1/2)

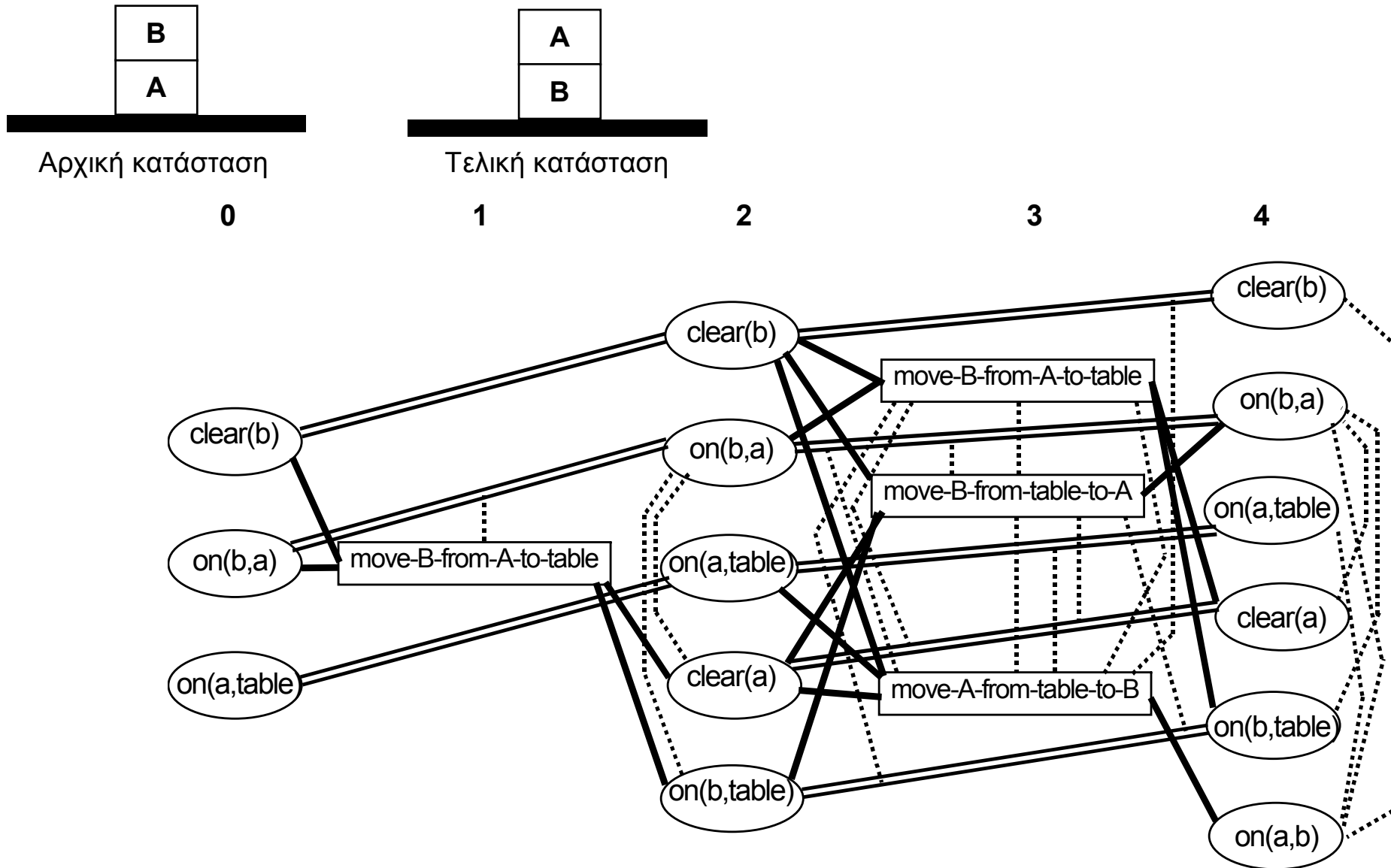
- ❖ Ο γράφος σχεδιασμού αποτελείται από αριθμημένα επίπεδα κόμβων
 - ❑ Κόμβοι των γεγονότων ή προτάσεων (fact nodes ή proposition nodes), στα άρτια επίπεδα.
 - ❑ Κόμβοι των ενεργειών (action nodes), στα περιττά επίπεδα
- ❖ Επαναλαμβανόμενη εναλλαγή δύο φάσεων:
 - ❑ *Επέκταση του γράφου (graph expansion).*
 - ❑ *Εξαγωγή λύσης (solution extraction).*
- ❖ Οι ακμές συνδέουν:
 - ❑ Τα γεγονότα ενός επιπέδου με τις ενέργειες του επόμενου επιπέδου που τα έχουν ως προϋποθέσεις.
 - ❑ Τις ενέργειες ενός επιπέδου με τα γεγονότα των λιστών προσθήκης αυτών στο επόμενο επίπεδο.
- ❖ Ενέργειες διατήρησης
 - ❑ Συμβολίζονται με `noop` (no-operator)



Σχέσεις Αμοιβαίου Αποκλεισμού

- ❖ Μια σχέση αμοιβαίου αποκλεισμού αναφέρεται πάντα σε δύο κόμβους του ίδιου επιπέδου και δηλώνει ότι αυτοί δεν μπορούν να βρισκονται ταυτόχρονα στο ίδιο έγκυρο πλάνο.
- ❖ Δύο γεγονότα στο επίπεδο i είναι αμοιβαία αποκλειόμενα, εάν όλες οι ενέργειες στο επίπεδο $i-1$, συμπεριλαμβανομένων των ενεργειών `noop`, που επιτυγχάνουν αυτά τα γεγονότα είναι μεταξύ τους αμοιβαίως αποκλειόμενες (ασύμβατη υποστήριξη – inconsistent support).

Παράδειγμα

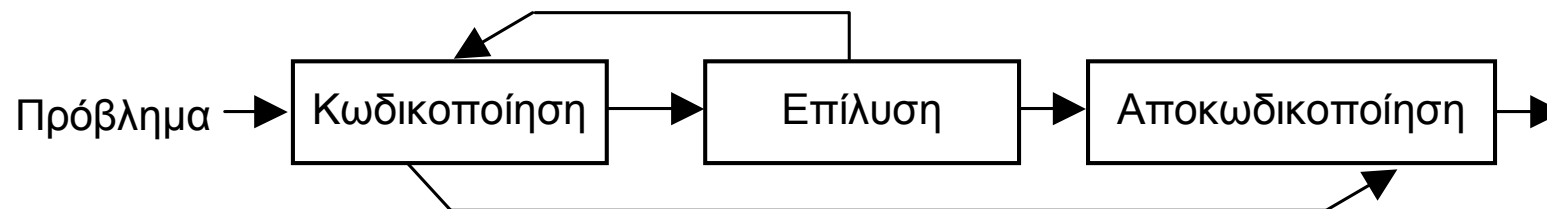


Σχεδιασμός Βασισμένος σε Γράφους

Εξαγωγή λύσης

- ❖ Ξεκινά μόλις σε κάποιο επίπεδο γεγονότων i εμφανιστούν όλα τα γεγονότα των στόχων, χωρίς καμιά σχέση αμοιβαίου αποκλεισμού μεταξύ τους.
- ❖ Τα γεγονότα των στόχων πρέπει να υποστηριχθούν από μη αμοιβαία αποκλειόμενες ενέργειες του προηγούμενου επιπέδου.
- ❖ Αναδρομικά, οι προϋποθέσεις των ενεργειών αυτών πρέπει να υποστηριχθούν από μη αμοιβαία αποκλειόμενες ενέργειες του προηγούμενού τους επιπέδου, μέχρι να φθάσουμε στο πρώτο επίπεδο.
- ❖ Εάν δεν βρεθεί τέτοιο πλάνο, ο γράφος επεκτείνεται κατά 2 ακόμη επίπεδα και η διαδικασία επαναλαμβάνεται.
- ❖ Συνθήκη τερματισμού είναι η εύρεση δύο εντελώς ίδιων επιπέδων γεγονότων.

Σχεδιασμός με Ικανοποίηση Προτάσεων



- ❖ Υπόθεση σχετικά με τον αριθμό των βημάτων του πλάνου-λύσης.
- ❖ Κωδικοποίηση σαν πρόβλημα ικανοποίησης προτάσεων σε μορφή *σύζευξης διαζεύξεων* (*conjunctive normal form*, CNF).
- ❖ Επίλυση με στοχαστικές ή συστηματικές μεθόδους.
- ❖ Εάν δεν βρεθεί λύση, επαναλαμβάνεται η διαδικασία για μεγαλύτερο αριθμό βημάτων.

Σχεδιασμός με Ικανοποίηση Προτάσεων

Κωδικοποίηση (1/2)

- ❖ Η σύζευξη των γεγονότων της αρχικής κατάστασης πρέπει να αληθεύει.

$$\text{on}(b, a)_0 \wedge \text{on}(a, \text{table})_0 \wedge \text{clear}(b)_0$$

- ❖ Η σύζευξη των γεγονότων των στόχων πρέπει επίσης να αληθεύει.

$$\text{on}(a, b)_4 \wedge \text{on}(b, \text{table})_4 \wedge \text{clear}(a)_4$$

- ❖ Οι ενέργειες συνεπάγονται τις προϋποθέσεις τους και τα αποτελέσματά τους.

$$\text{move-A-from-table-to-B}_3 \Rightarrow \text{on}(a, \text{table})_2 \wedge \text{clear}(a)_2 \wedge \text{clear}(b)_2 \wedge \text{on}(a, b)_4 \wedge \\ \text{clear}(a)_4 \wedge \neg \text{on}(a, \text{table})_4 \wedge \neg \text{clear}(b)_4$$

- ή ισοδύναμα σε μορφή CNF

$$\begin{aligned} & (\neg \text{move-A-from-table-to-B}_3 \vee \text{on}(a, \text{table})_2) \wedge \\ & (\neg \text{move-A-from-table-to-B}_3 \vee \text{clear}(a)_2) \wedge \\ & (\neg \text{move-A-from-table-to-B}_3 \vee \text{clear}(b)_2) \wedge \\ & (\neg \text{move-A-from-table-to-B}_3 \vee \text{on}(a, b)_4) \wedge \\ & (\neg \text{move-A-from-table-to-B}_3 \vee \text{clear}(a)_4) \wedge \\ & (\neg \text{move-A-from-table-to-B}_3 \vee \neg \text{on}(a, \text{table})_4) \wedge \\ & (\neg \text{move-A-from-table-to-B}_3 \vee \neg \text{clear}(b)_4) \end{aligned}$$

Σχεδιασμός με Ικανοποίηση Προτάσεων

Κωδικοποίηση (2/2)

- ❖ Ενέργειες ενός επιπέδου που είναι αμοιβαία αποκλειόμενες μεταξύ τους δεν μπορούν να εκτελεστούν ταυτόχρονα.

$$\neg \text{move-A-from-table-to-B}_3 \vee \neg \text{move-B-from-table-to-A}_3$$

- ❖ Κάθε γεγονός ενός επιπέδου (εκτός του επιπέδου 0) συνεπάγεται τη διάζευξη όλων των ενεργειών του προηγούμενου επιπέδου που το επιτυγχάνουν (συμπεριλαμβανομένων των ενεργειών διατήρησης).

$$\text{on}(b, a)_4 \Rightarrow \text{move-B-from-table-to-A}_3 \vee (\text{noop on}(b, a))_3$$

- ή ισοδύναμα σε μορφή CNF

$$\neg \text{on}(b, a)_4 \vee \text{move-B-from-table-to-A}_3 \vee (\text{noop on}(b, a))_3$$

Σχεδιασμός με Ικανοποίηση Προτάσεων

Συστηματική Επίλυση Προβλημάτων

Αλγόριθμος DPLL (CNF έκφραση φ)

Εάν η φ είναι κενή, επέστρεψε αληθές,

αλλιώς εάν υπάρχει πρόταση στη φ που να αποτιμάται ψευδής, επέστρεψε ψευδές,

αλλιώς εάν υπάρχει μια καθαρή μεταβλητή X στη φ , επέστρεψε $\text{DPLL}(\varphi(X))$,

αλλιώς εάν υπάρχει μια μοναδιαία πρόταση $\{X\}$ στη φ , επέστρεψε $\text{DPLL}(\varphi(X))$,

αλλιώς

επέλεξε μια μεταβλητή X που εμφανίζεται στη φ ,

Εάν $\text{DPLL}(\varphi(X)) = \text{αληθές}$, επέστρεψε αληθές,

αλλιώς επέστρεψε $\text{DPLL}(\varphi(\neg X))$.

Σχεδιασμός με Ικανοποίηση Προτάσεων

Στοχαστική Επίλυση Προβλημάτων

Αλγόριθμος GSAT (CNF έκφραση φ , integer: N_{restarts} , N_{flips})

Από $i=1$ μέχρι $i=N_{\text{restarts}}$

Έστω A μια τυχαία ανάθεση τιμών σε όλες τις μεταβλητές της φ .

Από $j=1$ μέχρι $j=N_{\text{flips}}$

Εάν η ανάθεση A ικανοποιεί την φ , επέστρεψε αληθές

Αλλιώς

Έστω X η μεταβλητή εκείνη της φ , της οποίας η αντιστροφή της τιμής δίνει το μεγαλύτερο αριθμό ικανοποιημένων προτάσεων στην πρόταση φ (σε περίπτωση ύπαρξης πολλών τέτοιων μεταβλητών, επέλεξε μια τυχαία)

Τροποποίησε την A , αντιστρέφοντας την τιμή της μεταβλητής X .

Επέστρεψε ψευδές.

Άλλες Εξελιγμένες Τεχνικές Σχεδιασμού

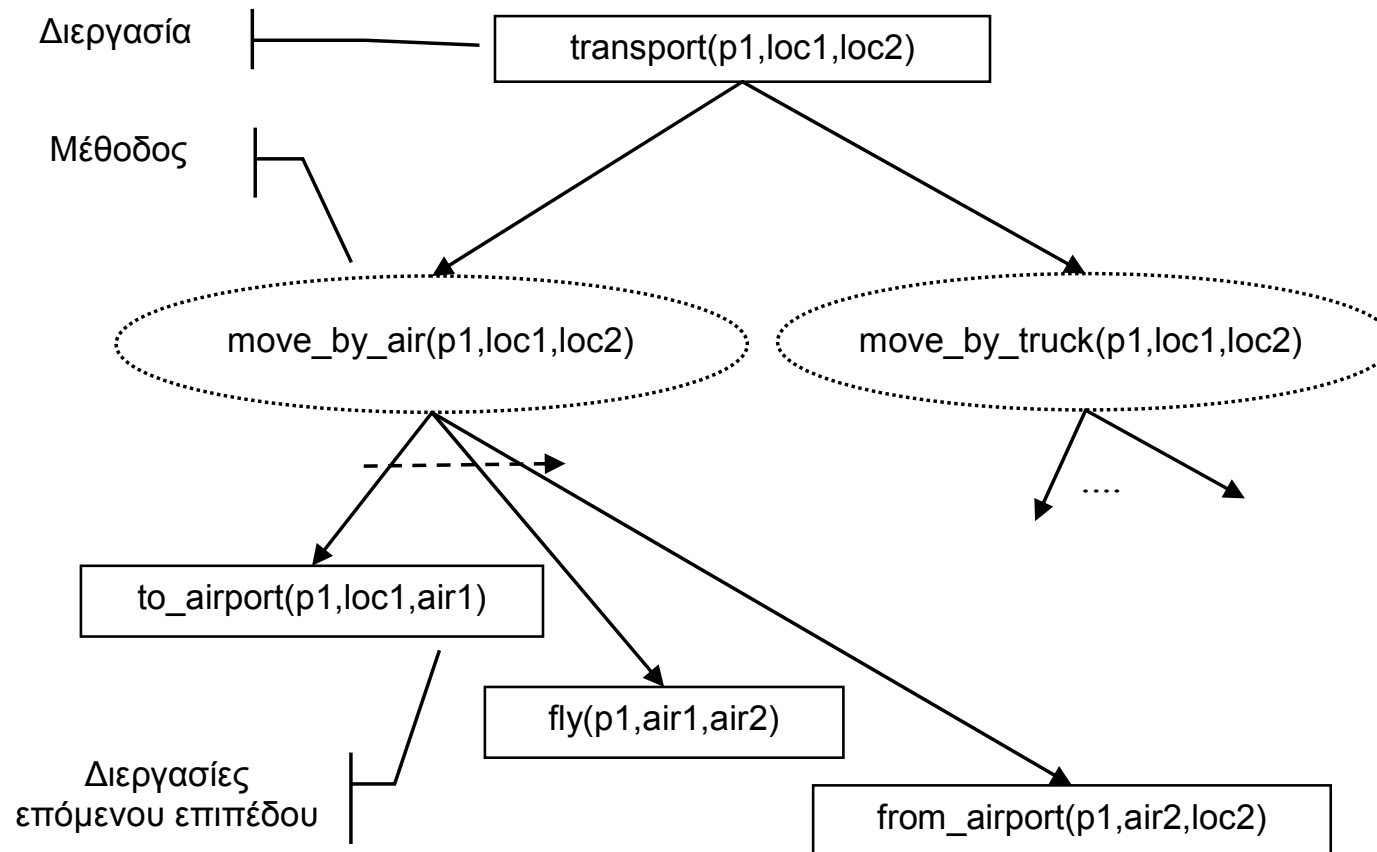
- ❖ Εφαρμογή Ικανοποίησης Περιορισμών στο σχεδιασμό
- ❖ Αναπαράσταση προβλημάτων σχεδιασμού ως προβλήματα
 - ❑ *Ελέγχου Μοντέλων*
 - ❑ *Μαρκοβιανές Διαδικασίες Απόφασης*
 - ❑ Επίλυση προβλημάτων σχεδιασμού με αβεβαιότητα (uncertainty)

Σχεδιασμός σε Ιεραρχικά Δίκτυα Διεργασιών

Hierarchical Task Network Planning-HTN

- ❖ Κατηγορία σχεδιασμού με τις περισσότερες πρακτικές εφαρμογές.
- ❖ Επιχειρείται η εύρεση πλάνου με ζητούμενο την επίτευξη μιας ανώτερου επιπέδου διεργασίας (high level task)
- ❖ *Βασική ιδέα:* Ιεραρχική αποδόμηση της αρχικής διεργασίας στόχου σε απλούστερες μέχρις ότου ο σχεδιαστής να καταλήξει σε άμεσα εκτελέσιμες διεργασίες.
- ❖ *Μέθοδος σχεδιασμού:* Αποδομεί μια διεργασία σε ένα σύνολο από απλούστερες του αμέσως κατώτερου επιπέδου, οι οποίες είναι μερικώς διατεταγμένες.
- ❖ Επιτυγχάνεται η κωδικοποίηση γνώσης για την εύρεση πλάνου που χρησιμοποιείται από ανθρώπους ειδικούς
- ❖ *Πλεονέκτημα:* Σημαντικά αποδοτικότερη λύση του προβλήματος σχεδιασμού
- ❖ *Μειονέκτημα:* Ανάγκη για κωδικοποίηση των μεθόδων
- ❖ *Ιεραρχικοί σχεδιαστές:* SHOP, JSHOP, SHOP2

Παράδειγμα Αποδόμησης Διεργασιών



- ❖ Πρόβλημα μετακίνησης φορτίου $p1$ από μια τοποθεσία $loc1$ σε μια τοποθεσία $loc2$
 - ❑ Το ζητούμενο είναι μια διεργασία $transport(p1,loc1,loc2)$
 - ❑ Η επίτευξη της $transport(p1,loc1,loc2)$ μπορεί να πραγματοποιηθεί με τη μεταφορά του φορτίου αεροπορικώς $move_by_air(p1,loc1,loc2)$ ή οδικώς $move_by_truck(p1,loc1,loc2)$