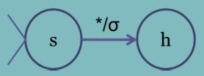
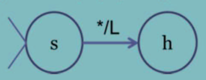
ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΜΗΧΑΝΗΣ TURING

Προκειμένου να κατασκευάσουμε μηχανές Turing που κάνουν μία ουσιαστική δουλειά, ορίζουμε απλές μηχανές που θα χρησιμοποιήσουμε ως δομικά στοιχεία σε πιο περίπλοκες μηχανές.
Οι μηχανές που ορίζει το βιβλίο του ΕΑΠ είναι οι εξής (* σημαίνει «οτιδήποτε»):

M_σ ή σ : «Γράψιμο Συμβόλου σ»

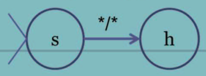


M_L ή L : «Κίνηση Αριστερά»



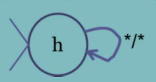
M_R ή R : «Κίνηση Δεξιά»

$M_>$ ή > «Μηχανή αρχή»



(*/* σημαίνει «ότι αφήνουμε την ταινία ανέπαφη»):

M_h ή h «Μηχανή-Τέλος»



R^2: «Δύο θέσεις δεξιά»

$> R \longrightarrow R$

Ομοίως μπορούμε να χρησιμοποιήσουμε την μηχανή L^2

Γενικότερα μπορούμε να χρησιμοποιήσουμε τις μηχανές L^k,R^k για k κάποιον φυσικό αριθμό.

$R_{\#}$: «Δεξιά μέχρι να συναντήσεις μη κενό»

Σχηματικά Διαβάζεται: «Όσο διαβάζεις # πήγαινε δεξιά
Ομοίως ορίζεται η μηχανή $L_{\#}$

$R_{\#}$: «Δεξιά μέχρι να συναντήσεις #»

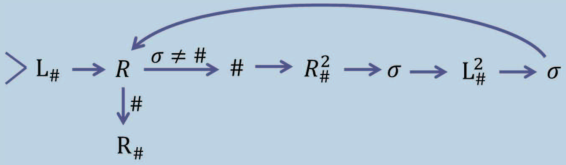
Σχηματικά Διαβάζεται: «Όσο δεν διαβάζεις # πήγαινε δεξιά
Ομοίως ορίζεται η μηχανή $L_{\#}$

$\longrightarrow$ Υποχρεωτική Μετάβαση (την ακολουθούμε υποχρεωτικά)

$\xrightarrow{x}$ Μετάβαση αν η κεφαλή δείχνει στο σύμβολο x

$\xrightarrow{\sigma \neq \#}$ Μετάβαση με αποθήκευση συμβόλου. Η μηχανή θυμάται ότι διάβασε το σύμβολο σ και μπορούμε έπειτα να γράψουμε το σύμβολο σ στην ταινία με την μηχανή **σ**.

Παράδειγμα Διαγράμματος Ροής Μ.Τ. που υπολογίζει τη συνάρτηση (s,#w#)=(h,#w#w^R#)

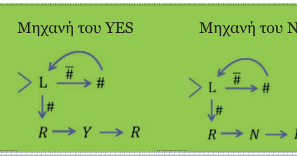


ΙΔΙΟΤΗΤΕΣ ΑΠΟΦΑΣΙΣΙΜΩΝ ΓΛΩΣΣΩΝ

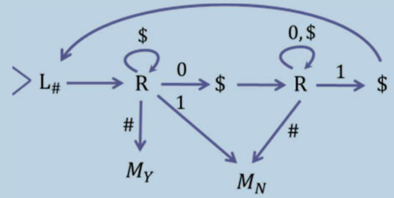
Μία **μηχανή Turing** θα λέμε ότι αποφασίζει μία γλώσσα αν για κάθε συμβολοσειρά εισόδου w:

- Τερματίζει με σχηματισμό (h, #Y#) αν $w \in L$
- Τερματίζει με σχηματισμό (h, #N#) αν $w \notin L$

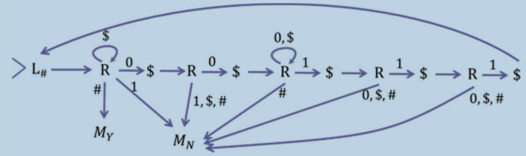
Αν για μία γλώσσα L υπάρχει μηχανή Turing που την αποφασίζει λέγεται Turing-Αποφασίσιμη (ή Αναδρομική ή Επιλύσιμη ή Αποφασίσιμη Γλώσσα)



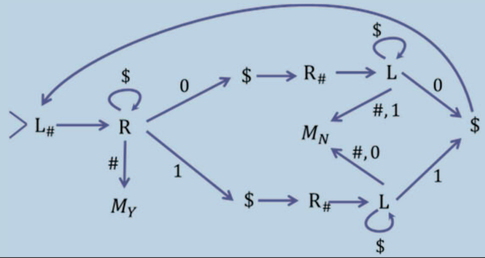
Ισότητα $L = \{0^n 1^n | n \geq 0\}$



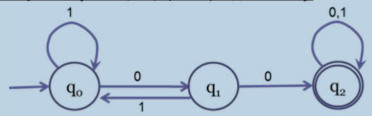
Αναλογία $L = \{0^{2n} 1^{3n} | n \geq 0\}$



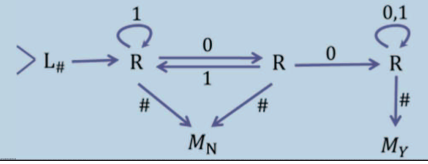
Παλινδρομικότητα $L = \{ww^R | w \in \{0,1\}^*\}$



Κανονικές $L = \{ w \in \{0,1\}^* | w\ \pi\varepsilon\rho\iota\varepsilon\chi\varepsilon\iota\ \tau o\ 00\}$.



Μ.Τ. που προσομοιώνει το ΝΠΑ:



ΑΠΟΔΕΙΞΕΙΣ ΜΗ ΕΠΙΛΥΣΙΜΟΤΗΤΑΣ

ΘΕΩΡΗΜΑ: Έστω $\mathbf{L_1} \leq \boldsymbol{L_2}$ (Υπάρχει αναγωγή από την L_1 στην L_2). Τότε ισχύουν τα εξής:

1. Αν η L_2 είναι Turing-Αποφασίσιμη, τότε και η L_1 είναι Turing-Αποφασίσιμη
2. **Αν η L_1 δεν είναι Turing-Αποφασίσιμη, τότε και η L_2 δεν είναι Turing-Αποφασίσιμη**
3. Αν η L_2 είναι Turing-Αποδεκτή, τότε και η L_1 είναι Turing-Αποδεκτή
4. Αν η L_1 είναι μη Turing-Αποδεκτή, τότε και η L_2 είναι μη Turing-Αποδεκτή

Από την εκφώνηση εντοπίζουμε την **γνωστή** και την **άγνωστη** μη επιλύσιμη γλώσσα. Για κάθε μία από αυτές εντοπίζουμε την είσοδό τους και το ερώτημα το οποίο θέτουν.

Γνωστή μη επιλύσιμη:

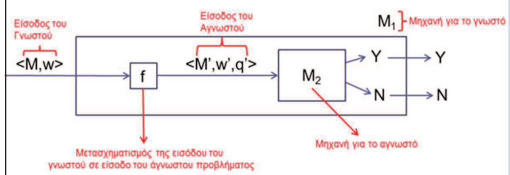
Είσοδος Γνωστής — Ερώτημα Γνωστής

$H = \{< M,w >|\ H\ M\ \mu\varepsilon\ \tau\varepsilon\rho\mu\alpha\tau\acute{\iota}\zeta\varepsilon\iota\ \mu\varepsilon\ \varepsilon\acute{\iota}\sigma o\delta o\ w\}$

Άγνωστη μη επιλύσιμη:

Είσοδος Αγνωστής — Ερώτημα Αγνωστής

$Q = \{< M,w,q >|\ H\ M$ με είσοδο w περνάει από την κατάσταση q$\}$



Απόδειξη: Γνωστή μη επιλύσιμη γλώσσα:
$L_1 = \{< M,w >|\ H\ M$ με τερματίζει με είσοδο w$\}$
Αγνωστή μη επιλύσιμη γλώσσα: $L_2 = \{< M,w,q >|\ H\ M$ με είσοδο w περνάει από την κατάσταση q$\}$

Έστω ότι η γλώσσα L_2 είναι αποφασίσιμη, άρα υπάρχει μια μηχανή Turing που την αποφασίζει, εστω M_2. Κατασκευάζουμε μια μηχανή Turing M_1 που αποφασίζει τη γλώσσα L_1 ως εξής: — standard

Η M_1 με είσοδο <M,w> θέτει M'=M, w'=w και q=τελική κατάσταση του M. Έπειτα περνάει την είσοδο <M',w',q> στη μηχανή M_2 — Περιγραφή του μετασχηματισμού

1. Αν η M_2 απαντήσει ΝΑΙ,
   τότε η M περνάει από την τελική κατάσταση h, με είσοδο w, — YES στο ερώτημα του αγνωστού
   άρα η M τερματίζει με είσοδο w, — YES στο ερώτημα του γνωστού
   θέτουμε τη M_1 να απαντήσει ΝΑΙ.
2. Αν η M_2 απαντήσει ΟΧΙ,
   τότε η M δεν περνάει από την τελική κατάσταση h, με είσοδο w, — NO στο ερώτημα του αγνωστού
   άρα η M δεν τερματίζει με είσοδο w, — NO στο ερώτημα του γνωστού
   θέτουμε τη M_1 να απαντήσει ΟΧΙ.

Κατασκευάσαμε μια Μ.Τ. που αποφασίζει την L_1. Άτοπο. Άρα η L_2 δεν είναι αποφασίσιμη — standard

ΑΠΟΔΕΚΤΕΣ ΓΛΩΣΣΕΣ

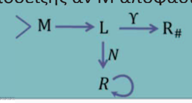
Μία Μηχανή Turing θα λέμε ότι **αποδέχεται** (ή ημι-αποφασίζει ή αναγνωρίζει) μία γλώσσα αν για κάθε συμβολοσειρά εισόδου w:

- Τερματίζει με σχηματισμό (h, #u<u>a</u>v#) αν $w \in L$
- Δεν Τερματίζει αν $w \notin L$ (πέφτει σε βρόχο)

Αν για μία γλώσσα L υπάρχει μηχανή Turing που την ημι-αποφασίζει **λέγεται Turing Αποδεκτή** (ή Αναδρομικά Απαριθμήσιμη ή Turing-Απαριθμήσιμη ή Αναγνωρίσιμη) Γλώσσα

Σχηματικά απεικονίζουμε μια αποδεκτή γλώσσα ως εξής:

ΘΕΩΡΗΜΑ: Κάθε Turing-Αποφασίσιμη γλώσσα είναι Turing-Αποδεκτή γλώσσα.
(Σκιαγράφηση Απόδειξης αν Μ αποφασίσιμη:)



Θεώρημα: Η γλώσσα $\boldsymbol{H} = \{< M,w >|$ H M τερματίζει με είσοδο w$\}$ είναι **αποδεκτή** γλώσσα

Απόδειξη του Θεωρήματος:

Δείχνουμε ότι η H είναι αποδεκτή γλώσσα κατασκευάζοντας μία μηχανή Turing M' η οποία ημι-αποφασίζει την H ως εξής. Η M' με είσοδο <M,w> λειτουργεί όπως η καθολική μηχανή Turing U, δηλαδή προσομοιώνει την λειτουργία της μηχανής Truing M με είσοδο w.

Είναι προφανές ότι:

- Αν η M με είσοδο w τερματίζει, τότε θέτουμε την M' να τερματίζει.
- Αν η M με είσοδο w κρεμάει, μπορούμε να το «πιάσουμε» (π.χ. θέτοντας έναν ειδικο χαρακτήρα στο αριστερό άκρο της ταινίας της M και αν διαβαστεί αυτός ο χαρακτήρας, τότε η M' θα πέφτει σε ατέρμονα βρόχο).
- Αν η M με είσοδο w δεν τερματίζει, τότε και η M' δεν τερματίζει.

Συνεπώς η M' ημι-αποφασίζει την H, άρα η H είναι αποδεκτή γλώσσα.

ΑΠΟΔΕΙΞΕΙΣ ΑΠΑΡΙΘΜΗΣΙΜΟΤΗΤΑΣ

Ορισμός:

- Μία γλώσσα θα λέγεται **λεξικογραφικά Turing-Απαριθμήσιμη** αν και μόνο αν διαθέτει λεξικογραφικό Turing-Απαριθμητή
- **Λεξικογραφικός Turing Απαριθμητής** είναι μία Μ.Τ. που εκτυπώνει μία-μία τις συμβολοσειρές της γλώσσας με λεξικογραφική σειρά

Ορισμός:

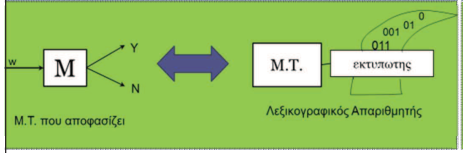
- Μία γλώσσα θα λέγεται **Turing-Απαριθμήσιμη** αν και μόνο αν διαθέτει Turing-Απαριθμητή
- **Turing Απαριθμητής** είναι μία Μ.Τ. που και πάλι εκτυπώνει όλες τις συμβολοσειρές της γλώσσας:
  - Ωστόσο τις εκτυπώνει με τυχαία σειρά και πιθανώς με επαναλήψεις
  - Όμως αν μια συμβολοσειρα ανήκει στην γλώσσα, τότε εγγυημένα σε κάποιο βήμα εκτύπωσης αυτή θα εκτυπωθεί!

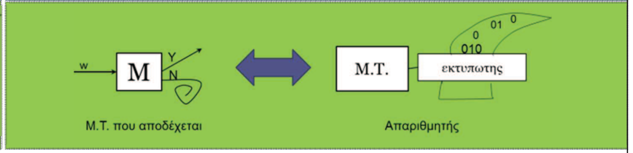
Θεώρημα:

- Μία γλώσσα είναι **λεξικογραφικά Turing-Απαριθμήσιμη** αν και μόνο αν είναι **Turing-Αποφασίσιμη** γλώσσα

Θεώρημα:

- Μία γλώσσα είναι **Turing-Απαριθμήσιμη** αν και μόνο αν είναι **Turing-Αποδεκτή** γλώσσα





Η Γλώσσα L={M | |L(M)|>3} είναι απαριθμήσιμή

Δοθείσης μιας μηχανής Turing M, μπορούμε να κατασκευάσουμε μια μηχανή Turing M' η οποία με τη διαδικασία της χελιδονοούρας απαριθμεί τις λέξεις της L(M). Συγκεκριμένα χρησιμοποιεί τη λεξικογραφική σειρά του αλφαβήτου της M και συγκεκριμένα:

Επαναλαμβάνει σε φάσεις:

- Στην $1^η$ φάση παράγει την πρώτη συμβολοσειρά του Σ*
- Στην $2^η$ φάση παράγει τις 2 πρώτες συμβολοσειρές του Σ*
- Στην $3^η$ φάση παράγει τις 3 πρώτες συμβολοσειρές του Σ*

  κ.οκ.
- Στην n-οστή φάση προσομοιώνουμε την M κατά n βήματα στις n πρώτες συμβολοσειρές.

Κάθε συμβολοσειρά με την οποία η M τερματίζει, τυπώνεται και προχωράμε στην επόμενη φάση.

Τρέχουμε την M' και αν σε κάποια φάση οι λέξεις που απαριθμήσει γίνουν 4, τερματίζει. Αλλιώς δεν τερματίζει.

Κατασκευάσαμε Μ.Τ. η οποία ημιαποφασίζει την L άρα αυτή είναι αποδεκτή, άρα και απαριθμήσιμη.

ΚΛΕΙΣΤΟΤΗΤΕΣ ΑΠΟΦΑΣΙΣΙΜΩΝ

Η L_1 είναι Αποδεκτή Γλώσσα, άρα υπάρχει μία μηχανη Turing που την ημι-αποφασίζει έστω M_1

Η L_2 είναι Αποδεκτή Γλώσσα, άρα υπάρχει μία μηχανη Turing που την ημι-αποφασίζει έστω M_2

Κλειστότητα των Αποφασισίμων Γλωσσών στην **Ένωση**

Κατασκευάζουμε μία μηχανή Turing, έστω M' η οποία με είσοδο w λειτουργεί ως εξής:

1) Τρέχει την M_1 με είσοδο w. Αν η M_1 απαντήσει ΝΑΙ, τότε η M' απαντά ΝΑΙ και τερματίζει. Αν η M_1 απαντήσει ΌΧΙ προχωράει στο βήμα 2:

2) Τρέχει την M_2 με είσοδο w. Αν η η M_2 απαντήσει ΝΑΙ, τότε η M' απαντά ΝΑΙ και τερματίζει. Αν η M_2 απαντήσει ΌΧΙ τότε απαντά ΌΧΙ και τερματίζει.

Κλειστότητα των Αποφασισίμων Γλωσσών στην **Τομή**

Κατασκευάζουμε μία μηχανή Turing, έστω M' η οποία με είσοδο w λειτουργεί ως εξής:

1) Τρέχει την M_1 με είσοδο w. Αν η M_1 απαντήσει ΟΧΙ, τότε η M' απαντά ΟΧΙ και τερματίζει. Αν η M_1 απαντήσει ΝΑΙ προχωρά στο βήμα 2:

2) Τρέχει την M_2 με είσοδο w. Αν η η M2 απαντήσει ΟΧΙ, τότε η M' απαντά ΟΧΙ και τερματίζει. Αν η M_2 απαντήσει ΝΑΙ τότε η M' απαντά ΝΑΙ και τερματίζει.

Κλειστότητα των Αποφασισίμων Γλωσσών στην **Παράθεση**

Κατασκευάζουμε μία μηχανή Turing, έστω M' η οποία με είσοδο w λειτουργεί ως εξής:

1. Πρώτα μία μηχανή Turing διαχωριστής D παράγει όλους τους δυνατούς διαχωρισμούς της συμβολοσειράς w στην παράθεση δύο συμβολοσειρών w_1 και w_2 (δηλαδή όλους τους δυνατούς διαχωρισμούς της w ως w_1w_2.)
2. Για κάθε δυνατό διαχωρισμό: Τρέχει την M_1 με είσοδο w_1 και την M_2 με είσοδο w_2. Αν και οι δύο μηχανές απαντήσουν ΝΑΙ, τότε η M' τερματίζει απαντώντας ΝΑΙ

Αν όλοι οι δυνατοί διαχωρισμοί απαντηθούν ΌΧΙ, τότε και η M' τερματίζει απαντώντας ΌΧΙ.

Κλειστότητα των Αποφασισίμων Γλωσσών στο **Συμπλήρωμα**

Η L είναι Αποφασίσιμη Γλώσσα, άρα υπάρχει μία μηχανη Turing που την αποφασίζει έστω M

Κατασκευάζουμε μία μηχανή Turing, έστω M' η οποία με είσοδο w λειτουργεί ως εξής:

1) Τρέχει την M με είσοδο w.

- Αν η M απαντήσει ΟΧΙ, τότε η M' απαντά ΝΑΙ και τερματίζει.
- Αν η M απαντήσει ΟΧΙ, τότε η M' απαντάει ΌΧΙ και τερματίζει.

Κλειστότητα των Αποφασισίμων Γλωσσών στο **Αστέρι Kleene**

Κατασκευάζουμε μία μηχανή Turing, έστω M' η οποία με είσοδο w λειτουργεί ως εξής:

1. Πρώτα μία μηχανή Turing διαχωριστής D παράγει όλους τους δυνατούς διαχωρισμούς της συμβολοσειράς w στην παράθεση 1..|w| συμβολοσειρών (δηλαδή όλους τους δυνατούς διαχωρισμούς της w ως $w_1w_2 \ldots w_k$ με k=1,2,…|w|)
2. Για κάθε δυνατό διαχωρισμό: Τρέχει την M διαδοχικά με εισόδους $w_1, w_2, \ldots, w_k$. Αν η M απαντήσει ΝΑΙ για όλες τις συμβολοσειρές τότε η M' τερματίζει απαντώντας ΝΑΙ.

Αν όλοι οι δυνατοί διαχωρισμοί απαντηθούν ΌΧΙ, τότε και η M' τερματίζει απαντώντας ΌΧΙ.

ΚΛΕΙΣΤΟΤΗΤΕΣ ΑΠΟΔΕΚΤΩΝ

Η L_1 είναι Αποδεκτή Γλώσσα, άρα υπάρχει μία μηχανη Turing που την ημι-αποφασίζει έστω M_1

Η L_2 είναι Αποδεκτή Γλώσσα, άρα υπάρχει μία μηχανη Turing που την ημι-αποφασίζει έστω M_2

Κλειστότητα των Αποδεκτών Γλωσσών στην **Ένωση**

Κατασκευάζουμε μία μηχανή Turing, έστω M' η οποία με είσοδο w λειτουργεί ως εξής:

Εκτελεί **εναλλάξ** τις M_1 και M_2, δηλαδή τρέχει εναλλάξ ένα βήμα στην M_1, ένα βήμα στην M_2 κ.ο.κ. Εάν σε κάποιο βήμα μία από τις δύο τερματίσει, τότε θέτουμε την M' να τερματίσει.

Κλειστότητα των Αποδεκτών Γλωσσών στην **Τομή**

Κατασκευάζουμε μία μηχανή Turing, έστω M' η οποία με είσοδο w λειτουργεί ως εξής:

1) Τρέχει την M_1 με είσοδο w.

Αν η M_1 δεν τερματίσει (άρα η w δεν ανήκει στην L_1), τότε και η M' δεν τερματίζει (όπως θα όφειλε, αφού η w δεν ανήκει στην $L_1 \cap L_2$)

Αν η M_1 τερματίσει (άρα η w ανήκει στην L_1), τότε και η M' προχωρά στο επόμενο βήμα.

2) Τρέχει την M_2 με είσοδο w.

Αν η M_2 δεν τερματίσει (άρα η w δεν ανήκει στην L_2), τότε και η M' δεν τερματίζει (όπως θα όφειλε, αφού η w δεν ανήκει στην $L_1 \cap L_2$)

Αν η M_2 τερματίσει (άρα η w ανήκει στην L_2), τότε και η M' τερματίσει.

Κλειστότητα των Αποδεκτών Γλωσσών στην **Παράθεση**

Κατασκευάζουμε μία μηχανή Turing, έστω M' η οποία με είσοδο w λειτουργεί ως εξής:

1. Πρώτα μία μηχανή Turing διαχωριστής D παράγει όλους τους δυνατούς διαχωρισμούς της συμβολοσειράς w στην παράθεση δύο συμβολοσειρών w_1 και w_2 (δηλαδή όλους τους δυνατούς διαχωρισμούς της w ως w_1w_2.)
2. Για κάθε διαχωρισμό εξετάζεται παράλληλα αν η συμβολοσειρά w ανήκει στην παράθεση ως εξής:
   - Για τον πρώτο διαχωρισμό: Τρέχει ένα βήμα στην M_1 με είσοδο w_1 ένα βήμα της M_2 με είσοδο w_2.
   - Για τον δεύτερο διαχωρισμό: Τρέχει ένα βήμα στην M_1 με είσοδο w_1 ένα βήμα της M_2 με είσοδο w_2.

     …
   - Για τον τελευταίο διαχωρισμό: Τρέχει ένα βήμα στην M_1 με είσοδο w_1 ένα βήμα της M_2 με είσοδο w_2.
3. Αν σε κάποιο βήμα τερματίσουν οι δύο μηχανές που εξετάζουν έναν διαχωρισμό, τότε η M' τερματίζει.

Κλειστότητα των Αποδεκτών Γλωσσών στο **Αστέρι Kleene**

Κατασκευάζουμε μία μηχανή Turing, έστω M' η οποία με είσοδο w λειτουργεί ως εξής:

1. Πρώτα μία μηχανή Turing διαχωριστής D παράγει όλους τους δυνατούς διαχωρισμούς της συμβολοσειράς w στην παράθεση 1..|w| συμβολοσειρών (δηλαδή όλους τους δυνατούς διαχωρισμούς της w ως $w_1w_2 \ldots w_k$ με k=1,2,…|w|)
2. Για κάθε διαχωρισμό εξετάζεται παράλληλα αν η συμβολοσειρά w ανήκει στο αστέρι Kleene:
   - Για τον πρώτο διαχωρισμό, έστω $w_1w_2 \ldots w_i$: Τρέχει ένα βήμα στην M με είσοδο w_1, ένα βήμα της M με είσοδο w_2,…, ένα βήμα της M με είσοδο w_i.

     …
   - Για τον τελευταίο διαχωρισμό $w_1w_2 \ldots w_j$: Τρέχει ένα βήμα στην M_1 με είσοδο w_1 ένα βήμα της M_2 με είσοδο w_2…, ένα βήμα της M με είσοδο w_j.

1. Αν σε κάποιο βήμα τερματίσουν όλες οι μηχανές που εξετάζουν έναν διαχωρισμό, τότε η M' τερματίζει.