ΠΛΗ30

ΕΝΟΤΗΤΑ 5: ΑΠΟΦΑΣΙΣΙΜΕΣ και ΑΠΟΔΕΚΤΕΣ ΓΛΩΣΣΕΣ

Μάθημα 5.4:
Αναγωγές

Δημήτρης Ψούνης

www.psounis.gr

ΠΕΡΙΕΧΟΜΕΝΑ

Α. Σκοπός του Μαθήματος

Οι στόχοι του μαθήματος είναι:

Επίπεδο Α
- (-)

Επίπεδο Β
- (-)

Επίπεδο Γ
- Αποδείξη ότι μία γλώσσα δεν είναι επιλύσιμη με αναγωγή.
- Γνώση γλωσσών που δεν είναι επιλύσιμες

Β. Θεωρία

1. Η αποδεικτική διαδικασία της αναγωγής

1. Σκεπτικό: Η γλώσσα Halting

- Όπως είδαμε στο προηγούμενο μάθημα:

Η γλώσσα $\boldsymbol{H} = \{< M, w >|$ Η Μ τερματίζει με είσοδο w$\}$
δεν είναι **αποφασίσιμη** γλώσσα
(ισοδύναμα είναι μια ΜΗ ΕΠΙΛΥΣΙΜΗ γλώσσα)

- Συνεπώς αν βρούμε έναν τρόπο να δίχνουμε για μια άλλη γλώσσα L :
  - «Η γλώσσα L είναι εξίσου δύσκολη με την μη επιλύσιμη γλώσσα»
  - Ή ότι
  - «Η γλώσσα H είναι μικρότερης ή ίσης δυσκολίας από την γλώσσα L»
- Τότε
  - Και η L θα είναι μια μη επιλύσιμη γλώσσα.
- Η διαδικασία αυτή λέγεται **<u>αναγωγή</u>** και αντί να λέμε η L_1 είναι μικρότερης ή ίσης δυσκολίας από την L_2 :
  - Θα λέμε ότι η L_1 ανάγεται στην L_2
  - Και θα συμβολίζουμε $\mathbf{L_1} \leq \boldsymbol{L_2}$
- (φυσικά με τον κατάλληλο μαθηματικό φορμαλισμό)

Β. Θεωρία

1. Η αποδεικτική διαδικασία της αναγωγής

2. Το θεώρημα αναγωγής

- Τα παραπάνω συνοψίζονται στο ακόλουθο θεώρημα:

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

1. Αν η $\mathrm{L_2}$ είναι Turing-Αποφασίσιμη, τότε και η $\mathrm{L_1}$ είναι Turing-Αποφασίσιμη
2. **Αν η $\boldsymbol{L_1}$ δεν είναι Turing-Αποφασίσιμη, τότε και η $\boldsymbol{L_2}$ δεν είναι Turing-Αποφασίσιμη**
3. Αν η $\mathrm{L_2}$ είναι Turing-Αποδεκτή, τότε και η $\mathrm{L_1}$ είναι Turing-Αποδεκτή
4. Αν η $\mathrm{L_1}$ είναι μη Turing-Αποδεκτή, τότε και η $\mathrm{L_2}$ είναι μη Turing-Αποδεκτή

Η χρήση του 2ου σκέλους του θεωρήματος θα χρησιμοποιείται για να δείξουμε ότι μία γλώσσα δεν είναι αποφασίσιμη:

- Θα μας δίνεται στην εκφώνηση ότι μία γλώσσα L_1 δεν είναι αποφασίσιμη (θα είναι για μας η **ΓΝΩΣΤΗ** μη επιλύσιμη γλώσσα)
- Θα μας δίνεται στην εκφώνηση ότι μία γλώσσα L_2 δεν είναι αποφασίσιμη (θα είναι για μας η **ΑΓΝΩΣΤΗ** μη επιλύσιμη γλώσσα)
- Θα κατασκευάζουμε μία αναγωγή της L_1 στην L_2 (δηλαδή μία αναγωγή της γνωστής γλώσσας στην άγνωστη γλώσσα με τρόπο που θα μελετήσουμε στην συνέχεια)

Β. Θεωρία

1. Η αποδεικτική διαδικασία της αναγωγής

3. Το σχήμα της απόδειξης αναγωγής

- Θα μελετήσουμε το σκεπτικό μιας αναγωγικής απόδειξης:

Άσκηση:

Να δείξετε ότι η γλώσσα:

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

δεν είναι επιλύσιμη. Χρησιμοποιείστε το γεγονός ότι:

$H = \{< M, w >|\ H\ \mathrm{M}\ \text{με τερματίζει με είσοδο w}\}$ δεν είναι επιλύσιμη

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:

$$H = \{\overbrace{< M, w >}^{\text{Είσοδος Γνωστής}}|\ \overbrace{H\ \mathrm{M}\ \text{τερματίζει με είσοδο w}}^{\text{Ερώτημα Γνωστής}}\}$$

Αγνωστή μη επιλύσιμη:

$$\mathrm{Q} = \{\overbrace{< M, w, q >}^{\text{Είσοδος Αγνωστής}}|\ \overbrace{H\ \mathrm{M}\ \text{με είσοδο w περνάει από την κατάσταση q}}^{\text{Ερώτημα Αγνωστής}}\}$$

Β. Θεωρία

1. Η αποδεικτική διαδικασία της αναγωγής

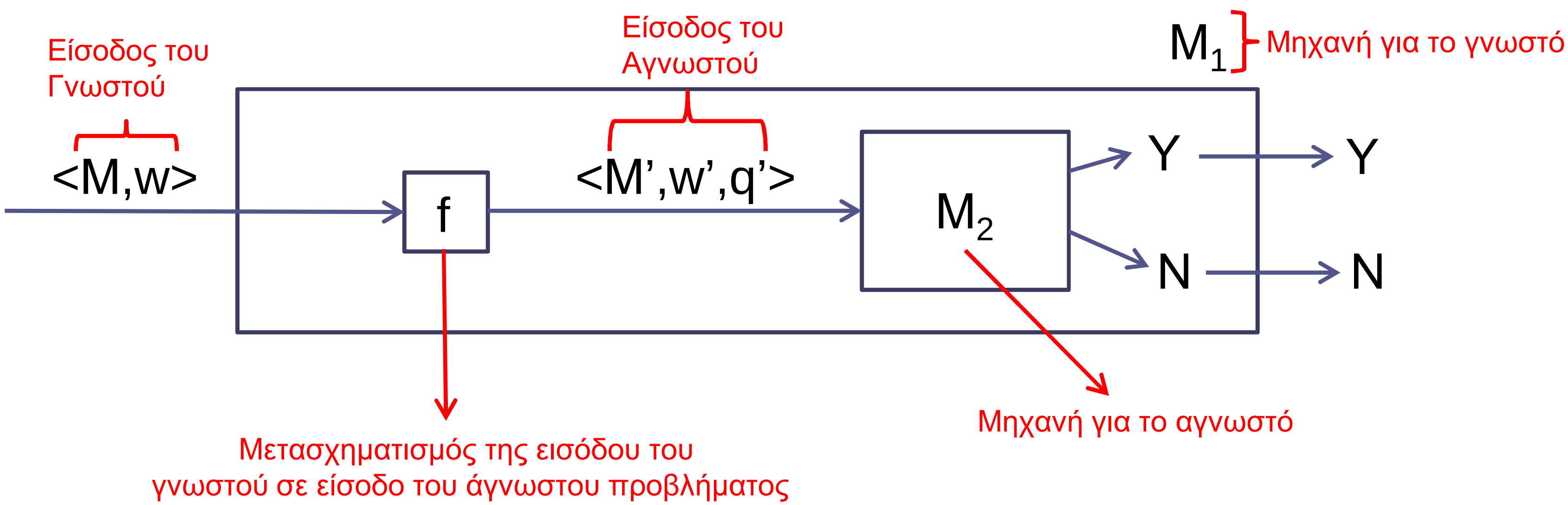
3. Το σχήμα της απόδειξης αναγωγής

Η απόδειξη της αναγωγής είναι μια **απόδειξη με ΑΤΟΠΟ**.
Υποθέτουμε ότι η ΑΓΝΩΣΤΗ ΓΛΩΣΣΑ είναι επιλύσιμη (αποφασίσιμη).
Άρα υπάρχει μία μηχανή Turing M_2 που την αποφασίζει.

Χρησιμοποιούμε την M_2 για να αποφασίσουμε την ΓΝΩΣΤΗ μη επιλύσιμη γλώσσα, δηλαδή για να κατασκευάσουμε μια μηχανή M_1 που αποφασίζει την ΓΝΩΣΤΗ. Αυτό όμως είναι άτοπο!

Απαιτείται λοιπόν ο μετασχηματισμός της εισόδου του γνωστού, σε είσοδο του άγνωστου προβλήματος, έτσι ώστε να αποφασίζεται **ορθά** το γνωστό μη επιλύσιμο πρόβλημα.

- Χρήσιμο θα φανεί το εξής σχήμα (αναγωγής):



Β. Θεωρία

1. Η αποδεικτική διαδικασία της αναγωγής

3. Το σχήμα της απόδειξης αναγωγής

Μετασχηματισμός της εισόδου του γνωστού σε είσοδο του άγνωστου ώστε να αποφασίζεται το γνωστό μη επιλύσιμο πρόβλημα!

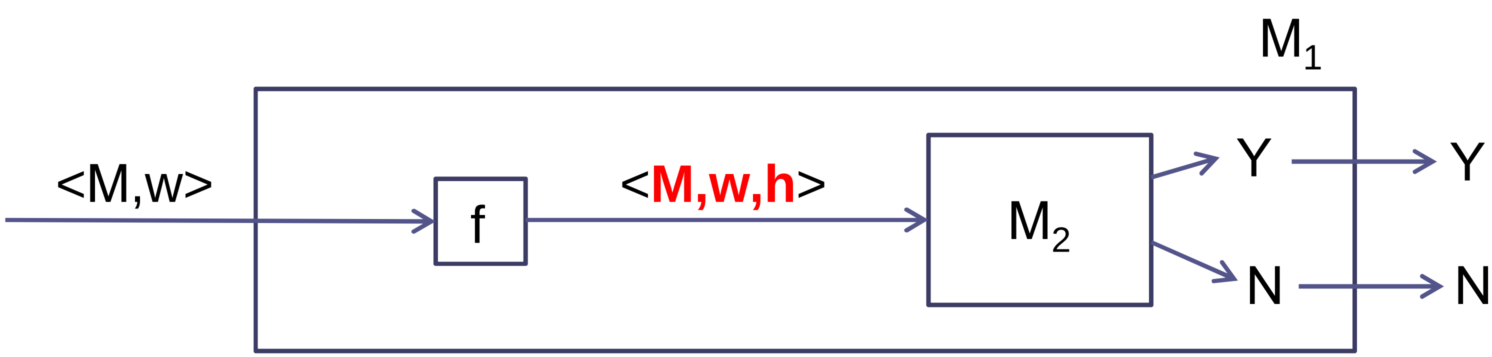
Είναι το πιο δύσκολο κομμάτι της αναγωγής και απαιτεί έμπνευση και μελέτη πολλών παραδειγμάτων από τη βιβλιογραφία!

Στο συγκεκριμένο παράδειγμα ο μετασχηματισμός είναι: M’=M, w’=w, q=τελική κατασταση του M.

Με τον μετασχηματισμό αυτό η ερώτηση της M_1:
- “Τερματίζει η M με είσοδο w?».

Μετασχηματίζεται στο ισοδύναμο ερώτημα:
- «Περνάει η M με είσοδο w από την κατάσταση h?»

Ισχύει λοιπόν ότι αν η M_2 απαντήσει ΝΑΙ, τότε θα θέσουμε την M_1 να απαντήσει ΝΑΙ.
Αν η M_2 απαντήσει ΌΧΙ, τότε η M_1 θα θέσουμε την M_1 να απαντήσει ΌΧΙ.



Β. Θεωρία

1. Η αποδεικτική διαδικασία της αναγωγής

3. Το σχήμα της απόδειξης αναγωγής

Η απόδειξη τυπικά γράφεται ως εξής:

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

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.

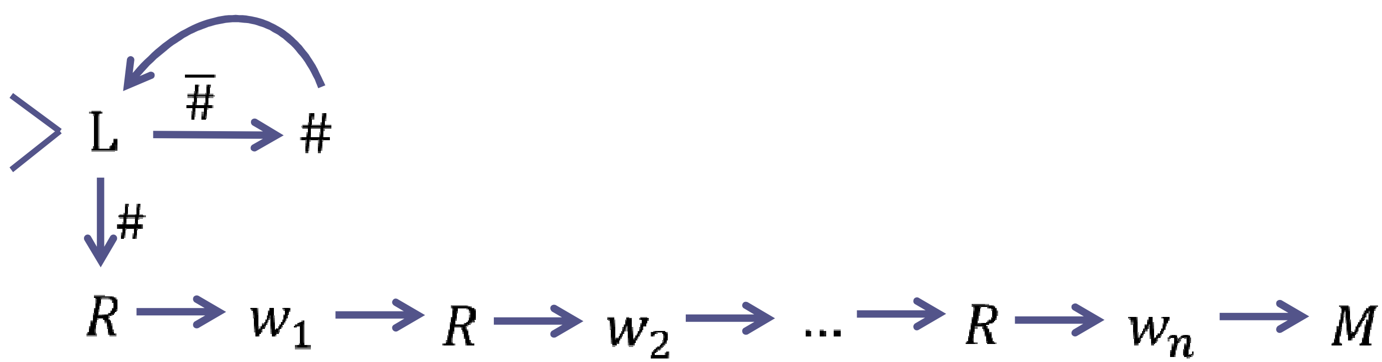
Κατασκευάσαμε μια Μ.Τ. που αποφασίζει την L_1. Άτοπο. Άρα η L_2 δεν είναι αποφασίσιμη — standard

Β. Θεωρία

2. Παραδείγματα Αναγωγών

1. Ένας συχνά χρησιμοποιούμενος μετασχηματισμός

- Ενας από τους πιο συχνά χρησιμοποιούμενους μετασχητισμούς στις αναγωγές, είναι δεδομένης μιας μηχανής M και μιας συμβολοσειράς w, να κατασκευάζουμε μια μηχανή M' η οποία προσομοιώνει την λειτουργία της M με είσοδο w.
- Αυτό γίνεται με μια διαδικασία 3 βημάτων, δηλαδή η M' κάνει 3 ενέργειες:
  - Σβήνει την είσοδό της,
  - Γράφει w στην στην ταινία της
  - Τρέχει την M με είσοδο w
- Αυτή η διαδικασία υλοποιείται εύκολα, π.χ. με την ακόλουθη μηχανή M' σε διάγραμμα ροής (θεωρούμε ότι ξεκινά με σχηματισμό #w$\underline{\#}$):



Β. Θεωρία

2. Παραδείγματα Αναγωγών

1. Ένας συχνά χρησιμοποιούμενος μετασχηματισμός

- Ανάλογα με τη συμπεριφορά της μηχανής Μ με είσοδο w διακρίνουμε δύο περιπτώσεις:

Αν η Μ με είσοδο w τερματίζει.

- Τότε η Μ' τερματίζει για οποιαδήποτε είσοδο.
  - (αφου πρακτικά τρέχει την Μ με είσοδο w)
- Συνεπώς ισχύει ότι $L(M') = \Sigma^*$
  - Δηλαδή η Μ' αποδέχεται οποιαδήποτε είσοδο.

Αν η Μ με είσοδο w δεν τερματίζει.

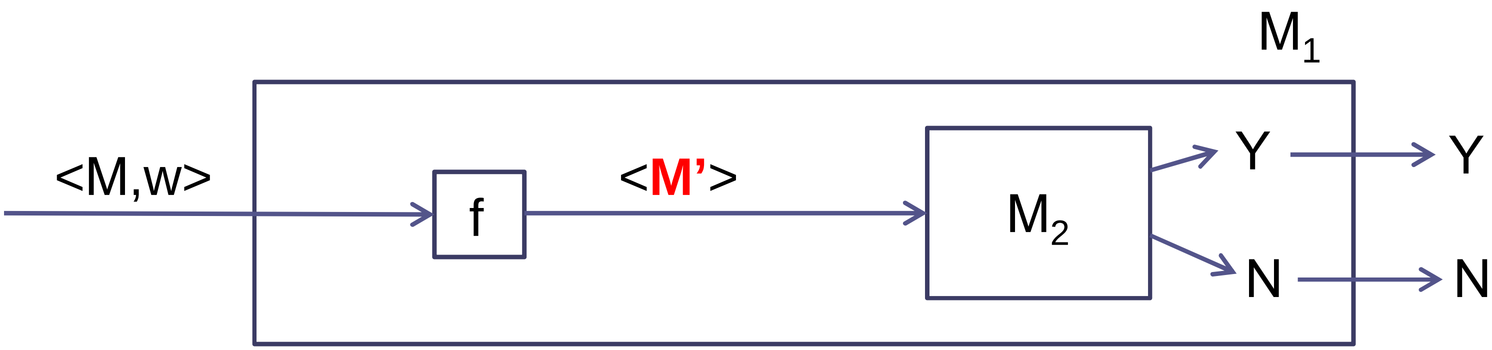
- Τότε η Μ' δεν τερματίζει για καμία είσοδο.
  - (αφου πρακτικά τρέχει την Μ με είσοδο w)
- Συνεπώς ισχύει ότι $L(M') = \emptyset$
  - Δηλαδή η Μ' δεν αποδέχεται καμία είσοδο.

Β. Θεωρία

2. Παραδείγματα Αναγωγών

2. Τερματίζει ένα πρόγραμμα με οποιαδήποτε είσοδο;

Εκφώνηση: Αποδείξτε ότι η γλώσσα $\mathrm{A} = \{< M >|\ H\ \mathrm{M}\ \text{τερματίζει με κάθε είσοδο}\}$ δεν είναι επιλύσιμη, δεδομένου ότι η γλώσσα $\mathrm{H} = \{< M, w >|\ H\ \mathrm{M}\ \text{τερματίζει με είσοδο w}\}$ δεν είναι επιλύσιμη.



Απόδειξη: Γνωστή μη επιλύσιμη γλώσσα: $L_1 = \{< M, w >|\ H\ \mathrm{M}\ \text{με τερματίζει με είσοδο w}\}$
Αγνωστή μη επιλύσιμη γλώσσα: $L_2 = \{< M >|\ H\ \mathrm{M}\ \text{τερματίζει με κάθε είσοδο}\}$

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:
Η M_1 με είσοδο <M,w> κατασκευάζει την μηχανή M’ η οποία σβήνει την είσοδο της, γράφει w στην ταινία της και τρέχει την M.
Έπειτα περνάει την είσοδο <M’> στη μηχανή M_2

1. Αν η M_2 απαντήσει ΝΑΙ, τότε η M’ τερματίζει με κάθε είσοδο
   άρα η M τερματίζει με είσοδο w, θέτουμε τη M_1 να απαντήσει ΝΑΙ.
2. Αν η M_2 απαντήσει ΟΧΙ, τότε η M΄ δεν τερματίζει με κάθε είσοδο (άρα δεν τερματίζει για καμία είσοδο)
   άρα η M δεν τερματίζει με είσοδο w, θέτουμε τη M_1 να απαντήσει ΟΧΙ.

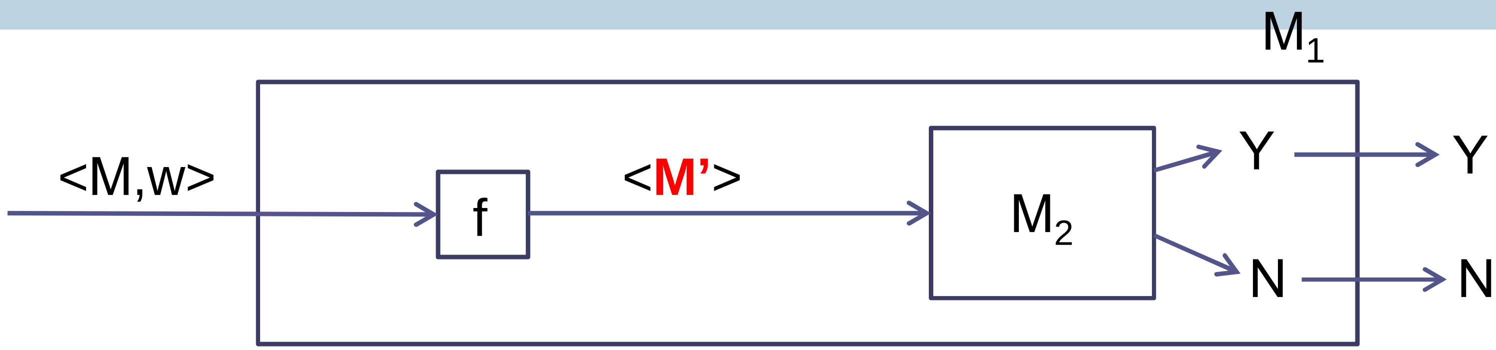
Κατασκευάσαμε μια M.T. που αποφασίζει την L_1. Άτοπο. Άρα η L_2 δεν είναι αποφασίσιμη

Β. Θεωρία

2. Παραδείγματα Αναγωγών

3. Τερματίζει ένα πρόγραμμα με είσοδο την κενή συμβολοσειρά;

Εκφώνηση: Αποδείξτε ότι η γλώσσα $\mathrm{E} = \{< M >| \, H \text{ M τερματίζει με είσοδο την κενή συμβολοσειρά}\}$ δεν είναι επιλύσιμη, δεδομένου ότι η γλώσσα $\mathrm{H} = \{< M, w >| \, H \text{ M τερματίζει με είσοδο w}\}$ δεν είναι επιλύσιμη.



Απόδειξη: Γνωστή μη επιλύσιμη γλώσσα: $L_1 = \{< M, w >| \, H \text{ M τερματίζει με είσοδο w}\}$
Αγνωστή μη επιλύσιμη γλώσσα: $L_2 = \{< M >| \, H \text{ M τερματίζει με είσοδο την κενή συμβολοσειρά}\}$

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:
Η M_1 με είσοδο <M,w> κατασκευάζει την μηχανή M' η οποία σβήνει την είσοδο της, γράφει w στην ταινία της και τρέχει την M.
Έπειτα περνάει την είσοδο <M'> στη μηχανή M_2

1. Αν η M_2 απαντήσει ΝΑΙ, τότε η M' τερματίζει με είσοδο την κενή συμβολοσειρά, άρα η M τερματίζει με είσοδο w, θέτουμε τη M_1 να απαντήσει ΝΑΙ.
2. Αν η M_2 απαντήσει ΟΧΙ, τότε η M΄ δεν τερματίζει με είσοδο την κενή συμβολοσειρά, άρα η M δεν τερματίζει με είσοδο w, θέτουμε τη M_1 να απαντήσει ΟΧΙ.

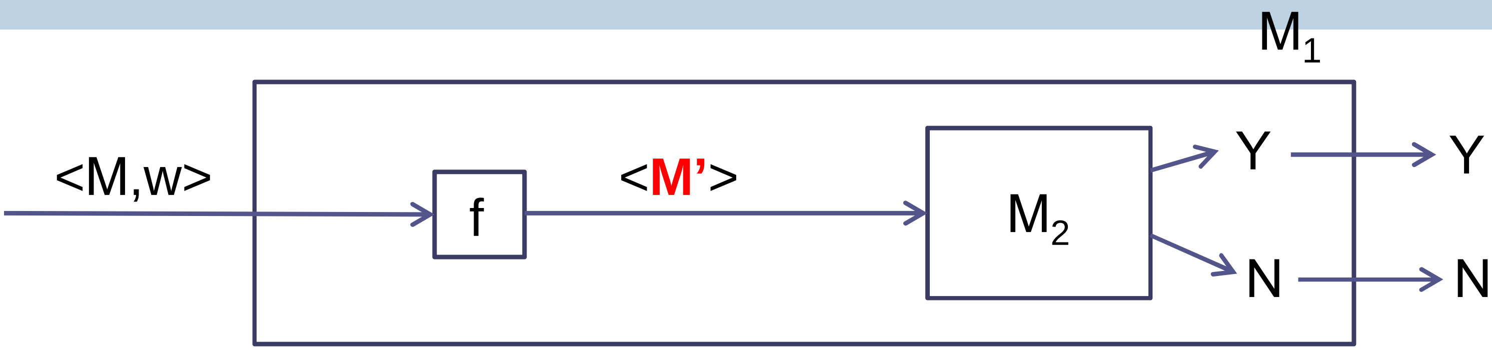
Κατασκευάσαμε μια Μ.Τ. που αποφασίζει την L_1. Άτοπο. Άρα η L_2 δεν είναι αποφασίσιμη

Β. Θεωρία

2. Παραδείγματα Αναγωγών

4. Τερματίζει ένα πρόγραμμα για έστω μία συμβολοσειρά;

Εκφώνηση: Αποδείξτε ότι η γλώσσα $EX = \{< M >|\ \textit{υπαρχει εισοδος με την οποια τερματιζει η}\ \mathrm{M}\}$ δεν είναι επιλύσιμη, δεδομένου ότι η γλώσσα $\mathrm{H} = \{< M, w >|\ H\ \mathrm{M\ τερματίζει\ με\ είσοδο\ w}\}$ δεν είναι επιλύσιμη.



Απόδειξη: Γνωστή μη επιλύσιμη γλώσσα: $L_1 = \{< M, w >|\ H\ \mathrm{M\ τερματίζει\ με\ είσοδο\ w}\}$
Άγνωστη μη επιλύσιμη γλώσσα: $L_2 = \{< M >|\ \textit{υπαρχει εισοδος με την οποια τερματιζει η}\ \mathrm{M}\}$

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:
Η M_1 με είσοδο <M,w> κατασκευάζει την μηχανή M' η οποία σβήνει την είσοδο της, γράφει w στην ταινία της και τρέχει την M.
Έπειτα περνάει την είσοδο <M'> στη μηχανή M_2

1. Αν η M_2 απαντήσει ΝΑΙ, τότε η M' (τερματίζει για κάποια είσοδο (άρα τερματίζει για κάθε είσοδο) άρα η M τερματίζει με είσοδο w, θέτουμε τη M_1 να απαντήσει ΝΑΙ.
2. Αν η M_2 απαντήσει ΟΧΙ, τότε δεν υπάρχει είσοδος με την οποία τερματίζει η M' άρα η M δεν τερματίζει με είσοδο w, θέτουμε τη M_1 να απαντήσει ΟΧΙ.

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

Β. Θεωρία

2. Παραδείγματα Αναγωγών

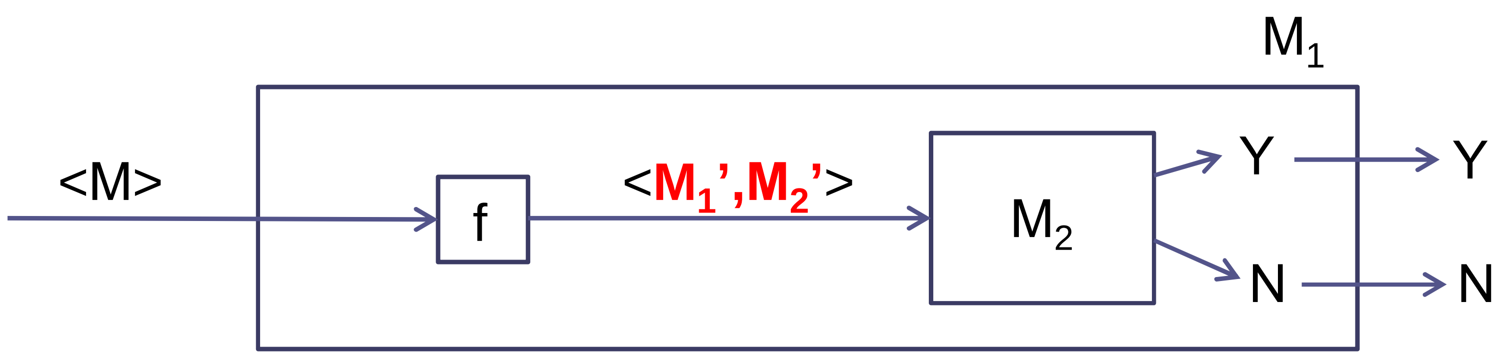
- Αν και η πιο συχνά χρησιμοποιούμενη (γνωστή) γλώσσα είναι η γλώσσα του Halting, μπορεί να μας δίνεται ως εκφώνηση οποιαδήποτε άλλη γλώσσα.
- Στην περίπτωση αυτή η αναγωγή έγκειται στην σωστή απάντηση της ερώτησης:
    - «Πως μπορώ να μεταμορφώσω την είσοδο του γνωστού προβλήματος σε είσοδο του άγνωστου προβλήματος, έτσι ώστε η απάντηση της μηχανής που απαντάει για το άγνωστο πρόβλημα, να απαντάει και στο ερώτημα της γνωστής γλώσσας?»
- Η εύρεση του κατάλληλου μετασχηματισμού μπορεί να είναι ιδιαίτερα επίπονη και εξαρτάται από το γνωστό και το άγνωστο πρόβλημα που έχουμε προς επίλυση σε κάθε περίπτωση.

Β. Θεωρία

2. Παραδείγματα Αναγωγών

5. Είναι δύο προγράμματα ισοδύναμα;

Εκφώνηση: Αποδείξτε ότι η γλώσσα $EQ = \{< M_1, M_2 >| \, L(M_1) = L(M_2)\}$ δεν είναι επιλύσιμη, δεδομένου ότι η γλώσσα $A = \{< M >| \, H \text{ M με τερματίζει με οποιαδήποτε είσοδο}\}$ δεν είναι επιλύσιμη.



Απόδειξη: Γνωστή μη επιλύσιμη γλώσσα: $L_1 = \{< M >| \, H \text{ M με τερματίζει με οποιαδήποτε είσοδο}\}$
Αγνωστή μη επιλύσιμη γλώσσα $L_2 = \{< M_1, M_2 >| \, L(M_1) = L(M_2)\}$

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:
Η M_1 με είσοδο <M> κατασκευάζει την μηχανή M_1' τέτοια ώστε $L(M_1')=\Sigma^*$ η και $M_2'=M$

Έπειτα περνάει την είσοδο < M_1', M_2'> στη μηχανή M_2

1. Αν η M_2 απαντήσει ΝΑΙ, τότε $L(M_1')=L(M_2')$ άρα $L(M_2')=\Sigma^*$ δηλαδή $L(M)=\Sigma^*$, θέτουμε τη M_1 να απαντήσει ΝΑΙ.
2. Αν η M_2 απαντήσει ΟΧΙ, τότε $L(M_1')\neq L(M_2')$ άρα $L(M_2') \neq \Sigma^*$ δηλαδή $L(M) \neq \Sigma^*$, θέτουμε τη M_2 να απαντήσει ΟΧΙ.

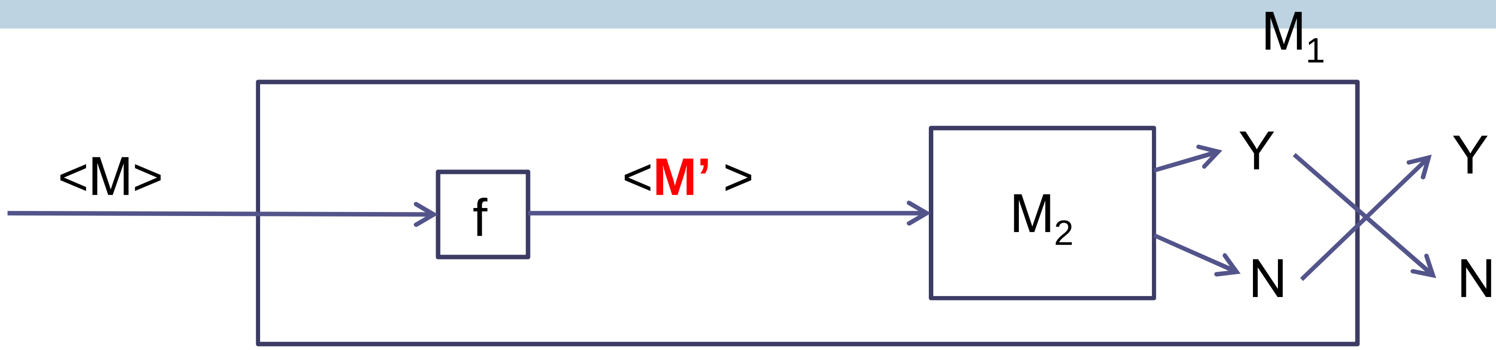
Κατασκευάσαμε μια Μ.Τ. που αποφασίζει την L_1. Άτοπο. Άρα η L_2 δεν είναι αποφασίσιμη

Β. Θεωρία

2. Παραδείγματα Αναγωγών

6. Το πρόγραμμα δεν τερματίζει για καμία είσοδο;

Εκφώνηση: Αποδείξτε ότι η γλώσσα $\mathrm{EM} = \{< \mathrm{M} >|\ \mathrm{L(M)} = \emptyset\}$ δεν είναι επιλύσιμη, δεδομένου ότι η γλώσσα $\mathrm{E} = \{< M >|\ \text{Υπάρχει έστω μία συμβολοσειρά με την οποία η M τερματίζει}\}$ δεν είναι επιλύσιμη.



Απόδειξη: Γνωστή μη επιλύσιμη γλώσσα:
$L_1 = \{< M >|\ \text{Υπάρχει έστω μία συμβολοσειρά με την οποία η M τερματίζει}\}$
Αγνωστή μη επιλύσιμη γλώσσα $L_2 = \{< \mathrm{M} >|\ \mathrm{L(M)} = \emptyset\}$

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:
Η M_1 με είσοδο <M> θέτει M'=M
Έπειτα περνάει την είσοδο < M'> στη μηχανή M_2

1. Αν η M_2 απαντήσει ΝΑΙ, τότε υπάρχει έστω μία συμβολοσειρά με την οποία η M τερματίζει, δηλαδή $L(M) \neq \emptyset$, θέτουμε τη M_1 να απαντήσει ΟΧΙ.
2. Αν η M_2 απαντήσει ΟΧΙ, τότε δεν υπάρχει έστω μία συμβολοσειρά με την οποία η M τερματίζει, δηλαδή $L(M) = \emptyset$, θέτουμε τη M_1 να απαντήσει ΝΑΙ.

Κατασκευάσαμε μια M.T. που αποφασίζει την L_1. Άτοπο. Άρα η L_2 δεν είναι αποφασίσιμη

Β. Θεωρία

3. Άλλες Μη Επιλύσιμες Γλώσσες

1. Αναγνώριση Συνόλου Γλώσσας

- Μελετήσαμε τυπικά τις ακόλουθες γλώσσες (και τα αντίστοιχα προβλήματα) και αποδείξαμε ότι δεν είναι επιλύσιμα:
  - $L_1 = \{< M >|\ \text{Η Μ αποδέχεται την κενή συμβολοσειρά}\}$
  - $L_2 = \{< M >|\ \text{Η Μ αποδέχεται τουλάχιστον μία είσοδο}\}$
  - $L_3 = \{< M >|\ \text{Η Μ αποδέχεται όλες τις εισόδους}\}$
  - $L_4 = \{< M >|\ \mathrm{L(M)} = \emptyset\}$
  - $L_5 = \{< M_1, M_2 >|\ \mathrm{L}(M_1) = \mathrm{L}(M_2)\}$
- Στο βιβλίο ΕΑΠ αποδεικνύεται επίσης ότι οι γλώσσες:
  - $L_6 = \{< \mathrm{M} >|\ \mathrm{L(M)}\ \text{ειναι κανονική}\}$
  - $L_7 = \{< \mathrm{M} >|\ \mathrm{L(M)}\ \text{ειναι χωρίς συμφραζόμενα}\}$
  - $L_8 = \{< \mathrm{M} >|\ \mathrm{L(M)}\ \text{ειναι χωρίς αποφασίσιμη}\}$
  - Δηλαδή δεν μπορούμε να αποφανθούμε ούτε καν για το αν η γλώσσα που αποφασίζει μια μηχανή άνήκει σε ένα από τα βασικά σύνολα γλωσσών που μελετήσαμε!
  - Οι παραπάνω αποδεικνύονται μη επιλύσιμες με βάση το γεγονός ότι η L4 δεν είναι επιλύσιμη και με βάση το γεγονός ότι η κενή γλώσσα είναι και κανονική και χωρίς συμφραζόμενα και αποφασίσιμη..

Β. Θεωρία

3. Άλλες Μη Επιλύσιμες Γλώσσες

2. Ασκ.Αυτ.2.2 και Δραστ.2.2.

Στο βιβλίο αποδεικνύεται ότι και οι ακόλουθες γλώσσες δεν είναι επιλύσιμες

- $L_1 = \{< M, w >|$ Η Μ χρησιμοποιει κατά τη διάρκεια εκτέλεσης της πεπερασμένη ταινία$\}$
- $L_2 = \{< M, q, p >|$ Υπάρχει σχηματισμός της Μ με κατάσταση q που παράγει σχηματισμό με κατάσταση p$\}$

  Ανάγοντας τη γλώσσα ύπαρξης συμβολοσειράς, θέτοντας q=s και p=h.
- $L_3 = \{< M, \sigma >|$ Γράφει η Μ κάποτε το σύμβολο σ στην ταινία έχοντας ξεκινήσει με είσοδο την κενή ταινία$\}$

  Ανάγοντας το πρόβλημα του τερματισμού γράφοντας κάποιο ειδικό σύμβολο αμέσως μετά το πέρας της εκτέλεσης της μηχανής
- $L_4 = \{< M, \sigma >|$ Γράφει η Μ κάποτε κάποιο μη κενό σύμβολο στην ταινια έχοντας ξεκινήσει με είσοδο την κενή ταινία$\}$

  Ανάγοντας την γλώσσα αποδοχής της κενής συμβολοσειράς, επιλέγοντας ένα μη κενό σύμβολο α από την περιγραφή της Μ και αποφασίζοντας τη γλώσσα Μα

Β. Θεωρία

3. Άλλες Μη Επιλύσιμες Γλώσσες

2. Ασκ.Αυτ.2.2 και Δραστ.2.2.

Στο βιβλίο αποδεικνύεται ότι και οι ακόλουθες γλώσσες δεν είναι επιλύσιμες

- $L_6 = \{< M >|$ Είναι πεπερασμένη η γλώσσα που αποδέχεται η M$\}$

  Ανάγοντας το πρόβλημα τερματισμού με τον συνήθη μετασχηματισμό

- $L_7 = \{< M_1, M_2 >| \; L(M_1) = \overline{L(M_2)}\}$

  Ανάγοντας τη γλώσσα αποδοχής όλων των συμβολοσειρών θέτοντας $M_1 = \mathrm{M}$, $\mathrm{L}(M_2) = \emptyset$

- $L_8 = \{< M_1, M_2 >|$ Υπάρχει είσοδος που και οι δύο τερματίζουν$\}$

  Ανάγοντας το πρόβλημα του τερματισμού θέτοντας M_1=M_2=M με είσοδο w με τον συνήθη μετασχηματισμό

Ενώ οι ακόλουθες γλώσσες αποδεικνύεται ότι είναι επιλύσιμες:

- $L_1 = \{< M, p >|$ Στην M υπάρχει σχηματισμός που παράγει σχηματισμό με κατάσταση p$\}$

- $L_2 = \{< M, w >|$ H M μετακινεί με είσοδο w την κεφαλή της αριστερά$\}$

  Μελετώντας την συνάρτηση μετάβασης της μηχανής Turing

Β. Θεωρία

3. Άλλες Μη Επιλύσιμες Γλώσσες

3. Μη Επιλύσιμες Γλώσσες για Γλώσσες Χωρίς Συμφραζόμενα

Οι παρακάτω γλώσσες που αφορούν γραμματικές χωρίς συμφραζόμενα δεν είναι επιλύσιμες. Οι αντίστοιχες γραμματικές G που εμφανίζονται είναι χωρίς συμφραζόμενα, ενώ όπου αναφέρεται M εννοείται ότι είναι αυτόματο στοίβας (θεώρημα 2.13, 2.14, 2.15 άσκηση αυτοαξιολόγησης 2.3

- $L_1 = \{< G >|\ \text{Ισχύει } L(G) = \Sigma^*\}$
- $L_2 = \{< G_1, G_2 >|\ \text{Ισχύει } L(G_1) = L(G_2)\}$
- $L_3 = \{< M_1, M_2 >|\ \text{Ισχύει } L(M_1) = L(M_2)\}$
- $L_4 = \{< M >|\ \text{Υπάρχει ισοδύναμο του M με μικρότερο πλήθος καταστάσεων}\}$
- $L_5 = \{< G >|\ \text{Ισχύει } L(G) = \emptyset\}$
- $L_6 = \{< G, L >|\ \text{Η κανονική γλώσσα L παράγεται από την γραμματική G}\}$
- $L_7 = \{< G, L >|\ \text{Είναι η } L(G) \text{ υποσύνολο της κανονικής γλώσσας L}\}$
- $L_8 = \{< G_1, G_2 >|\ \text{Ισχύει } L(G_1) \cap L(G_2) = \emptyset\}$
- $L_9 = \{< G_1, G_2 >|\ \text{Η } L(G_1) \cap L(G_2) \text{ είναι κανονική γλώσσα}\}$
- $L_{10} = \{< G_1, G_2 >|\ \text{Η } L(G_1) \cap L(G_2) \text{ είναι γλώσσα χωρίς συμφραζόμενα}\}$
- $L_{11} = \{< G >|\ \text{Η G είναι διφορούμενη}\}$

Γ. Ασκήσεις
Εφαρμογή 1

Δείξτε ότι η γλώσσα L={<M>| Η Μ τερματίζει με είσοδο 0011} δεν είναι επιλύσιμη χρησιμοποιώντας αναγωγή από το πρόβλημα του τερματισμού.

Γ. Ασκήσεις
Εφαρμογή 2

(2007Β) Έχουμε γνωρίσει γλώσσες που είναι Turing-αποδεκτές και Turing-αποφασίσιμες. Υπάρχουν όμως γλώσσες που δεν είναι ούτε καν Turing-αποδεκτές. Εξετάστε για παράδειγμα τις παρακάτω δύο γλώσσες:

$A = \{ \langle M,w,q \rangle \mid$ η M με είσοδο w μεταβαίνει στην κατάσταση q $\}$ και
$B = \{ \langle M,w,q \rangle \mid$ η M με είσοδο w δεν μεταβαίνει στην κατάσταση q $\}$.

Δείτε ότι το συμπλήρωμα της γλώσσας *A* είναι η ένωση των γλωσσών B και όλων των μη νόμιμων κωδικοποιήσεων $\langle M,w,q \rangle$, δηλαδή $= B \cup \{ x \mid x \neq \langle M,w,q \rangle \}$.

Α) Δείξτε (με σύντομο τρόπο) ότι οι γλώσσες *A*, B δεν είναι Turing-αποφασίσιμες.

Β) Δείξτε (με σύντομο τρόπο) ότι η γλώσσα A είναι Turing-αποδεκτή.

Γ) Δείξτε (με σύντομο τρόπο) ότι η γλώσσα B δεν είναι Turing-αποδεκτή.

Γ. Ασκήσεις
Εφαρμογή 3

(2008Α) *Δίνονται μηχανές Turing* M_1 *και* M_2 *και έστω* $L(M_1)$ *και* $L(M_2)$ *οι γλώσσες που αποδέχονται. Είναι η* $L(M_1)$ *συμπληρωματική της* $L(M_2)$*;* (Υπόδειξη: Αναγωγή από το μη επιλύσιμο πρόβλημα: *"Έστω Μ μια τυχαία Μηχανή Turing. Ισχύει* $L(M) = \emptyset$ *;"*)

Γ. Ασκήσεις

Εφαρμογή 4

(2008Β) Αποδείξτε, αν τα παρακάτω προβλήματα είναι ή όχι επιλύσιμα:

Δίδεται ένα πρόγραμμα στη γλώσσα C και μια εντολή Ε του προγράμματος.

(1) *Θα εκτελέσει το πρόγραμμα 10 φορές την εντολή Ε σε 60 min εκτέλεσής του;*

(2) *Θα εκτελέσει το πρόγραμμα 10 φορές την εντολή Ε κατά τη διάρκεια της εκτέλεσής του;*

Γ. Ασκήσεις
Εφαρμογή 5

(2009Β) Δείξτε ότι η γλώσσα L = {c(M,w,q) | η μηχανή Turing M περνάει από την κατάσταση q όταν ξεκινήσει με είσοδο τη σ/σειρά w} είναι αποδεκτή αλλά μη αποφασίσιμη.

Γ. Ασκήσεις
Εφαρμογή 6

(2011Α) Δίνεται η γλώσσα L = { <M,x,q>} | η μηχανή Turing M με είσοδο την συμβολοσειρά x δεν διέρχεται ποτέ από την κατάσταση q }. Αποδείξτε ότι γλώσσα L δεν είναι Turing αποφασίσιμη. Για την απόδειξη, κάντε αναγωγή από την γλώσσα A = { <M,x> | η μηχανή Turing M αποδέχεται την συμβολοσειρά x } για την οποία γνωρίζουμε ότι δεν είναι Turing αποφασίσιμη. Υποθέστε ότι η γλώσσα L είναι Turing αποφασίσιμη και καταλήξτε σε άτοπο.