Μη Επιλυσιμότητα

Όλες οι προσπάθειες τις οποίες κάναμε προκειμένου να ισχυροποιήσουμε τις Μ.Τ., έπεσαν στο κενό.

Θέση των Church-Turing: Η μηχανή Turing που τερματίζει για όλες τις εισόδους αποτελεί την αυστηρή τυποποίηση της διαισθητικής έννοιας του ‘αλγορίθμου’.

Η θέση των Church-Turing δεν είναι θεώρημα, και επομένως δεν μπορεί να αποδειχτεί. Είναι (θεωρητικά) δυνατό να βρεθεί στο μέλλον κάποιο ισχυρό μοντέλο υπολογισμού, το οποίο θα καταρρίψει τη θέση των Church-Turing. Κάτι τέτοιο όμως θεωρείται σήμερα απίθανο.

Μη Επιλυσιμότητα, συνέχεια

Μπορούμε να βρούμε προβλήματα τα οποία να μην επιλύονται από κανένα αλγόριθμο;

Τέτοια προβλήματα όπως ξέρουμε υπάρχουν: όλοι οι φορμαλισμοί που έχουμε εξετάσει (κανονικές γλώσσες, γλώσσες χωρίς συμφραζόμενα, μηχανές Turing, κλπ), μπορούν να αναπαραστήσουν μετρήσιμα άπειρες το πλήθος γλώσσες. Γνωρίζουμε όμως ότι το σύνολο όλων των γλωσσών είναι μη-μετρήσιμα άπειρο.

Πως μπορούμε όμως να βρούμε μια γλώσσα η οποία να μη μπορεί να αποφασιστεί από μια μηχανή Turing;

Για να επιτύχουμε το παραπάνω, θα χρειαστεί να ορίσουμε μηχανές Turing που παίρνουν ως είσοδο κωδικοποιήσεις μηχανών Turing.

Καθολικές Μηχανές Turing

Θα ορίσουμε μια μηχανή Turing η οποία όταν παίρνει ως είσοδο την περιγραφή μιας μηχανής M, θα συμπεριφέρεται όπως η M.

Για να μπορούμε να αναπαραστήσουμε μια οποιαδήποτε μηχανή Turing, θα πρέπει να βρούμε τρόπο να κωδικοποιήσουμε τις καταστάσεις και τα σύμβολα ταινίας της χρησιμοποιώντας ένα προκαθορισμένο αλφάβητο.

Αναπαριστούμε καταστάσεις στη μορφή qw, όπου $w \in \{0, 1\}^*$. Ομοίως, τα σύμβολα ταινίας στη μορφή aw, όπου $w \in \{0, 1\}^*$.

Παράδειγμα Αναπαράστασης

$M = (K, \Sigma, \delta, s, \{h\})$
$K = \{s, q, h\}$
$\Sigma = \{a, \sqcup, \triangleright\}$

Κατάσταση q	**Σύμβολο** σ	**Μετάβαση** $\delta(q,\sigma)$
s	a	$(q, \sqcup)$
s	$\sqcup$	$(h, \sqcup)$
s	$\triangleright$	$(s, \rightarrow)$
q	a	(s, a)
q	$\sqcup$	$(s, \rightarrow)$
q	$\triangleright$	$(q, \rightarrow)$

Παράδειγμα Αναπαράστασης, συνέχεια

Αναπαριστούμε τις καταστάσεις και τα στοιχεία του αλφαβήτου της M ως εξής:

Κατάσταση	**Αναπαράσταση**
s	$q00$
q	$q01$
h	$q11$
$\sqcup$	$a000$
$\triangleright$	$a001$
$\leftarrow$	$a010$
$\rightarrow$	$a011$
a	$a100$

Η αναπαράσταση $\ll M \gg$ της μηχανής Turing είναι η συμβολοσειρά:

$$\ll M \gg = (q00, a100, q01, a000), \ldots, (q01, a001, q01, a011)$$

Υπολογισμοί με Καθολικές Μηχανές Turing

Μπορούμε τώρα να ορίσουμε μια καθολική μηχανή Turing U, η οποία έχει δύο ορίσματα: την περιγραφή $\ll M \gg$ μιας μηχανής M και την περιγραφή $\ll w \gg$ μιας συμβολοσειράς w.

Θελουμε η U να τερματίζει με είσοδο $\ll M \gg \ll w \gg$ αν και μόνο αν η M τερματίζει με είσοδο w.

Είναι σχετικά απλό να σχεδιάσουμε τη U με τέτοιο τρόπο ώστε να προσομοιώνει τη λειτουργία της M (είναι απλούστερο αν η U χρησιμοποιεί τρείς ταινίες).

Το Πρόβλημα του Τερματισμού (Halting Problem)

Έστω ότι έχουμε καταφέρει να γράψουμε ένα πρόγραμμα `halts(P,X)` το οποίο μας επιστρέφει την απάντηση 'yes' αν το `P` τερματίζει με είσοδο `X` και 'no' διαφορετικά.

Θεωρήστε τώρα το εξής πρόγραμμα:

$$\mathsf{f(X)\{\ l:\ if\ halts(X,X)\ then\ goto\ l\ else\ halt\}}$$

Τερματίζει το `f(f)`;

Τερματίζει αν και μόνο αν το `halts(f,f)` επιστρέφει 'no', δηλαδή αν και μόνο άν το `f` με είσοδο `f` δεν τερματίζει (άτοπο). Συμπέρασμα: δεν υπάρχει τέτοιο πρόγραμμα `halts`.

Το Πρόβλημα του Τερματισμού, συνέχεια

Με οδηγό τον προηγούμενο συλλογισμό, θα βρούμε μια γλώσσα που είναι αναδρομικά απαριθμήσιμη αλλά όχι αναδρομική. Έστω η γλώσσα:

$$\mathsf{H} = \{\ll \mathsf{M} \gg \ll \mathsf{w} \gg\text{: η μηχανή Turing M τερματίζει όταν έχει είσοδο w}\}$$

Η γλώσσα H είναι προφανώς αναδρομικά απαριθμήσιμη καθώς ημιαποφασίζεται από τη μηχανή U. Είναι όμως η H αναδρομική;

Έστω ότι η H είναι αναδρομική. Τότε, το ίδιο θα ισχύει και για τη γλώσσα:

$$\mathsf{H}_1 = \{\ll \mathsf{M} \gg\text{: η μηχανή Turing M τερματίζει όταν έχει είσοδο } \ll \mathsf{M} \gg\}$$

Αφού η H_1 είναι αναδρομική, τότε το ίδιο ισχύει και για το συμπλήρωμά της:

$$\begin{aligned}\overline{\mathsf{H}_1} = \ & \{\mathsf{w} : \text{η w δεν είναι κωδικοποίηση μηχανής Turing} \\ & \text{ή είναι κωδικοποίηση μιας μηχανής Turing M} \\ & \text{που δεν τερματίζει όταν έχει είσοδο } \ll \mathsf{M} \gg\}\end{aligned}$$

Το Πρόβλημα του Τερματισμού, συνέχεια

Όμως, η $\overline{\mathsf{H}_1}$ δεν είναι ούτε καν αναδρομικά απαριθμήσιμη: έστω M^* μια μηχανή Turing που ημιαποφασίζει την $\overline{\mathsf{H}_1}$. Ανήκει η $\ll \mathsf{M}^* \gg$ στην $\overline{\mathsf{H}_1}$;

Από τον ορισμό της $\overline{\mathsf{H}_1}$, $\ll \mathsf{M}^* \gg \in \overline{\mathsf{H}_1}$ αν και μόνο αν η M^* δεν δέχεται τη συμβολοσειρά $\ll \mathsf{M}^* \gg$.

Όμως, η M^* ημιαποφασίζει την $\overline{\mathsf{H}_1}$, και έτσι $\ll \mathsf{M}^* \gg \in \overline{\mathsf{H}_1}$ αν και μόνο αν η M^* δέχεται τη συμβολοσειρά $\ll \mathsf{M}^* \gg$

Καταλήξαμε σε άτοπο, και επομένως η H δεν είναι αναδρομική.

Το Πρόβλημα του Τερματισμού, συνέχεια

Θεώρημα: Η κλάση των αναδρομικών γλωσσών είναι ένα γνήσιο υποσύνολο της κλάσης των αναδρομικά απαριθμήσιμων γλωσσών.

Επιπλέον, όπως είδαμε, η γλώσσα H_1 είναι αναδρομικά απαριθμήσιμη ενώ το συμπλήρωμά της δεν είναι. Κατά συνέπεια:

Θεώρημα: Η κλάση των αναδρομικά απαριθμήσιμων γλωσσών δεν είναι κλειστή ως προς το συμπλήρωμα.

Αφού οι αναδρομικές και οι αναδρομικά απαριθμήσιμες γλώσσες είναι δύο διαφορετικές κατηγορίες τυπικών γλωσσών, ας δούμε με μεγαλύτερη λεπτομέρεια κάποιες ιδιότητές τους.

Ιδιότητες των Αναδρομικών Γλωσσών

Θεώρημα: Μία γλώσσα L είναι αναδρομική αν και μόνο αν και αυτή και το συμπλήρωμά της είναι αναδρομικά απαριθμήσιμες.

Απόδειξη: Η κατεύθυνση ($\Rightarrow$) είναι προφανής (γιατί;)

Για την κατεύθυνση ($\Leftarrow$) θα κατασκευάσουμε μια μηχανή Turing M δύο ταινιών που αποφασίζει την L. Έστω ότι η L ημιαποφασίζεται από μια μηχανή Turing M_1 και η $\overline{L}$ από μία μηχανή Turing M_2.

Δεδομένης μιας συμβολοσειράς w, η M προχωράει στην πρώτη ταινία της όπως θα προχωρούσε η M_1 με είσοδο w, και στη δεύτερη ταινία της όπως η M_2 με είσοδο w.

Κάποια στιγμή ο υπολογισμός σε μία από τις δύο ταινίες θα τερματίσει. Τότε και η M τερματίζει στην κατάσταση **y** ή **n** (ανάλογα με το αν ο υπολογισμός ο οποίος τερμάτισε ήταν αυτός της πρώτης ταινίας ή της δεύτερης ταινίας αντίστοιχα).

Ιδιότητες των Αναδρομικών Γλωσσών, συνέχεια

Ορισμός: Μια μηχανή Turing M απαριθμεί τη γλώσσα L αν και μόνο αν, για κάποια προκαθορισμένη κατάσταση q της M,

$$L = \{w : (s, \triangleright\underline{\sqcup}) \vdash_M^* (q, \triangleright\underline{\sqcup}w)\}$$

Μία γλώσσα είναι απαριθμήσιμη κατά Turing αν και μόνο αν υπάρχει μία μηχανή Turing που την απαριθμεί.

Διαισθητικά: Όταν η M εισέρχεται στην κατάσταση q τότε σηματοδοτεί ότι η συμβολοσειρά που βρίσκεται αυτή τη στιγμή στην ταινία της, ανήκει στην L. Η M ακολούθως μπορεί να αφήσει την q και να επανέλθει αργότερα με ένα άλλο μέλος της L.

Προσοχή: Τα μέλη της L μπορούν να παρατεθούν σε οποιαδήποτε σειρά και με επαναλήψεις.

Ιδιότητες των Αναδρομικών Γλωσσών, συνέχεια

Θεώρημα: Μία γλώσσα L είναι αναδρομικά απαριθμήσιμη αν και μόνο αν είναι απαριθμήσιμη κατά Turing.

Απόδειξη: ($\Leftarrow$) Πρώτα αποδεικνύουμε ότι αν έχουμε μια μηχανή Turing E που απαριθμεί την L, τότε υπάρχει μια μηχανή Turing M που ημιαποφασίζει την L. Η M δουλεύει ως εξής:

Για συμβολοσειρά εισόδου w, η M αρχίζει να ‘τρέχει’ τη μηχανή E. Κάθε φορά που η E εισέρχεται στην ειδική κατάστασή της, η M συγκρίνει τη w με τη συμβολοσειρά που μόλις έχει ‘παραχθεί’ από την E. Αν είναι ίδιες, τότε η M αποδέχεται τη w. Διαφορετικά, συνεχίζει την παραπάνω διαδικασία.

Είναι προφανές ότι η M δέχεται ακριβώς εκείνες τις συμβολοσειρές οι οποίες εμφανίζονται στην απαρίθμηση της E.

Ιδιότητες των Αναδρομικών Γλωσσών, συνέχεια

Απόδειξη (συνέχεια): $(\Rightarrow)$ Έστω ότι έχουμε μια μηχανή Turing M που ημιαποφασίζει την L. Μπορούμε να κατασκευάσουμε μια μηχανή Turing E που απαριθμεί την L. Έστω $\mathsf{s}_1, \mathsf{s}_2, \ldots$ μια ακολουθία από όλες τις συμβολοσειρές του Σ^*. Η E λειτουργεί ως εξής:

Επανέλαβε τα ακόλουθα βήματα για $i = 1, 2, 3, \ldots$.

1. 'Τρέξε' την M για i βήματα για τις συμβολοσειρές $\mathsf{s}_1, \mathsf{s}_2, \ldots, \mathsf{s}_i$.
2. Αν κάποια συμβολοσειρά s_j γίνεται δεκτή, η E πηγαίνει στην ειδική κατάσταση q με s_j στην ταινία.

Αν η M αποδέχεται μια συμβολοσειρά w, τότε αυτή θα εμφανιστεί αργά ή γρήγορα στην ακολουθία των συμβολοσειρών που απαριθμεί η E.

Ιδιότητες των Αναδρομικών Γλωσσών, συνέχεια

Ορισμός: Μια μηχανή Turing M απαριθμεί λεξικογραφικά τη γλώσσα L αν, για κάποια προκαθορισμένη κατάσταση q της M, ισχύει το ακόλουθο. Όποτε έχουμε:

$$(q, \triangleright\underline{\sqcup}w) \vdash_M^+ (q, \triangleright\underline{\sqcup}w')$$

τότε η w' βρίσκεται λεξικογραφικά μετά από την w. Μία γλώσσα είναι λεξικογραφικά απαριθμήσιμη κατά Turing αν και μόνο αν υπάρχει μία μηχανή Turing που την απαριθμεί λεξικογραφικά.

Ιδιότητες των Αναδρομικών Γλωσσών, συνέχεια

Θεώρημα: Μία γλώσσα L είναι αναδρομική αν και μόνο αν είναι λεξικογραφικά απαριθμήσιμη κατά Turing.

Απόδειξη: ($\Rightarrow$) Αφού η L είναι αναδρομική, τότε υπάρχει μηχανή Turing M που την αποφασίζει. Κατασκευάζουμε μία μηχανή Turing M′ η οποία απαριθμεί λεξικογραφικά όλες τις συμβολοσειρές του αλφαβήτου της L. Για κάθε μία από τις συμβολοσειρές 'τρέχει' την M. Όποτε η M δέχεται την είσοδο, η M′ παρουσιάζει τη συμβολοσειρά και συνεχίζει στην επόμενη.

($\Leftarrow$) Αφού η L είναι λεξικογραφικά απαριθμήσιμη, μπορούμε να κατασκευάσουμε μια μηχανή Turing M′ που την αποφασίζει. Η M′ με είσοδο w ξεκινάει την απαρίθμηση της L και περιμένει έως ότου είτε η w παρουσιαστεί είτε ξεπεραστεί λεξικογραφικά. Στην πρώτη περίπτωση δέχεται την w ενώ στη δεύτερη την απορρίπτει.

Μη Επιλύσιμα Προβλήματα

Όπως έχουμε δει σε προηγούμενες διαφάνειες, δεν υπάρχει αλγόριθμος ο οποίος για κάθε μηχανή Turing M και συμβολοσειρά εισόδου w, να αποφασίζει αν η M τερματίζει με είσοδο w ή όχι.

Προβλήματα όπως το παραπάνω θα τα ονομάζουμε μη επιλύσιμα.

Προσοχή: Το πρόβλημα του τερματισμού μπορεί να λύνεται σε κάποιες ειδικές περιπτώσεις. Αυτό που έχουμε αποδείξει είναι ότι δεν υπάρχει μια γενική μέθοδος η οποία να αποφασίζει σωστά σε όλες τις περιπτώσεις.

Μη Επιλύσιμα Προβλήματα, συνέχεια

Πως μπορούμε να αποδείξουμε ότι ένα πρόβλημα είναι μη επιλύσιμο (χωρίς να χρησιμοποιήσουμε την τεχνική που ακολουθήσαμε για την H;)

Διαισθητικά, θα δείχνουμε ότι αν μια γλώσσα L_2 ήταν αναδρομική, τότε το ίδιο θα ίσχυε και για τη γλώσσα L_1 η οποία είναι γνωστό ότι δεν είναι αναδρομική. Μπορούμε να περιγράψουμε πιο τυπικά τη διαδικασία αυτή με τον παρακάτω ορισμό:

Ορισμός: Έστω $\mathsf{L}_1, \mathsf{L}_2 \subseteq \Sigma^*$ δύο γλώσσες. Μια αναγωγή από την L_1 στην L_2 είναι μια αναδρομική συνάρτηση $\tau : \Sigma^* \rightarrow \Sigma^*$ τέτοια ώστε $\mathsf{w} \in \mathsf{L}_1$ αν και μόνο αν $\tau(\mathsf{w}) \in \mathsf{L}_2$.

Μη Επιλύσιμα Προβλήματα, συνέχεια

Διαίσθηση: Μια αναγωγή από τη γλώσσα L_1 στη γλώσσα L_2 σημαίνει ότι το να λύσουμε το πρόβλημα που αντιστοιχεί στην L_1 δεν μπορεί να είναι δυσκολότερο από το να λύσουμε το πρόβλημα που αντιστοιχεί στην L_2 (γιατί μια λύση για το L_2 μας δίνει μια λύση για το L_1).

Η παραπάνω παρατήρηση συνεπάγεται ότι αν η ή L_1 δεν είναι αναδρομική και υπάρχει μια αναγωγή από την L_1 στην L_2, τότε και η L_2 δεν είναι αναδρομική.

Μη Επιλύσιμα Προβλήματα, συνέχεια

Θεώρημα: Αν ή L_1 δεν είναι αναδρομική και υπάρχει μια αναγωγή από την L_1 στην L_2, τότε και η L_2 δεν είναι αναδρομική.

Απόδειξη: Έστω ότι η L_2 είναι αναδρομική και έστω ότι αποφασίζεται από μια μηχανή Turing M_2. Έστω T η μηχανή Turing που υπολογίζει τη συνάρτηση τ. Τότε η μηχανή Turing TM_2 θα αποφασίζει την L_1 (γιατί;). Επομένως καταλήξαμε σε άτοπο.

Με τη χρήση αναγωγών μπορούμε να αποδείξουμε ότι πολλά προβλήματα είναι μη επιλύσιμα.

Μη Επιλύσιμα Προβλήματα, συνέχεια

Για να δούμε στην πράξη τη χρήση των αναγωγών, θα χρησιμοποιήσουμε ένα ακόμη πρόβλημα το οποίο είναι γνωστό ότι είναι μη-επιλύσιμο (η απόδειξη του παρακάτω θεωρήματος παραλείπεται):

Θεώρημα: Το πρόβλημα «*Δεδομένης γραμματικής χωρίς συμφραζόμενα* G, *ισχύει ότι* $L(G) = \Sigma^*$;» είναι μη-επιλύσιμο.

Σημείωση: Παρατηρούμε ότι ακόμη και προβλήματα που αφορούν απλά μοντέλα υπολογισμού όπως οι γραμματικές χωρίς συμφραζόμενα, μπορεί να είναι μη-επιλύσιμα.

Θα χρησιμοποιήσουμε το παραπάνω θεώρημα για να δείξουμε ότι και άλλα (φαινομενικά απλά) ερωτήματα για γραμματικές, είναι μη-επιλύσιμα.

Μη Επιλύσιμα Προβλήματα, συνέχεια

Θεώρημα: Το πρόβλημα *«Δεδομένων δύο γραμματικών χωρίς συμφραζόμενα G_1, G_2, ισχύει $L(G_1) = L(G_2)$;»* είναι μη-επιλύσιμο.

Απόδειξη: Αν είχαμε ένα αλγόριθμο ο οποίος έλυνε το παραπάνω πρόβλημα, τότε θα μπορούσαμε να λύσουμε και το πρόβλημα της προηγούμενης διαφάνειας αν του δίναμε ως δεύτερη είσοδο την κωδικοποίηση μιας γραμματικής που παράγει το Σ^*.

Η αναγωγή που χρησιμοποιήσαμε έμμεσα στην παραπάνω απόδειξη είναι η ακόλουθη:

$$\tau(\langle\langle G\rangle\rangle) = \langle\langle G\rangle\rangle\langle\langle G_{\Sigma^*}\rangle\rangle$$

όπου G_{Σ^*} είναι η κωδικοποίηση μιας γραμματικής χωρίς συμφραζόμενα που παράγει τη γλώσσα Σ^*.

Μη Επιλύσιμα Προβλήματα, συνέχεια

Θεώρημα: Το πρόβλημα «*Δεδομένων δύο γραμματικών χωρίς συμφραζόμενα* G_1, G_2, *ισχύει* $L(G_1) \subseteq L(G_2)$*;*» είναι μη-επιλύσιμο.

Απόδειξη: Αν είχαμε ένα αλγόριθμο ο οποίος έλυνε το παραπάνω πρόβλημα, τότε θα μπορούσαμε να λύσουμε και το πρόβλημα «Ισχύει ότι $L(G) = \Sigma^*$;» αν του δίναμε ως είσοδο την κωδικοποίηση μιας γραμματικής G_1 που παράγει το Σ^*.

Η αναγωγή που χρησιμοποιήσαμε έμμεσα στην παραπάνω απόδειξη είναι η ακόλουθη:

$$\tau(\langle\langle G\rangle\rangle) = \langle\langle G_{\Sigma^*}\rangle\rangle\langle\langle G\rangle\rangle$$

όπου G_{Σ^*} είναι η κωδικοποίηση μιας γραμματικής χωρίς συμφραζόμενα που παράγει τη γλώσσα Σ^*.

Πως αλλιώς θα μπορούσαμε να δείξουμε ότι το παραπάνω πρόβλημα είναι μη-επιλύσιμο;

Μη Επιλύσιμα Προβλήματα, συνέχεια

Ξεκινώντας από το πρόβλημα του τερματισμού και χρησιμοποιώντας αναγωγές μπορούμε να δείξουμε το ακόλουθο θεώρημα:

Θεώρημα: Τα παρακάτω προβλήματα είναι μη επιλύσιμα:

- *«Δεδομένης μιας μηχανής Turing M, τερματίζει η M με κενή ταινία;»*
- *«Δεδομένης μιας μηχανής Turing M, υπάρχει έστω και μία συμβολοσειρά για την οποία η M τερματίζει;»*
- *«Δεδομένης μιας μηχανής Turing M, τερματίζει η M για κάθε συμβολοσειρά εισόδου;»*
- *«Δεδομένων δύο μηχανών Turing M_1 και M_2, τερματίζουν με τις ίδιες συμβολοσειρές εισόδου;»*