

Θέση Church-Turing I

Πολλοί τρόποι περιγραφής αλγορίθμων. Όλοι είναι μηχανιστικά ισοδύναμοι και ειδικά ισοδύναμοι με μερικές αναδρομικές συναρτήσεις

Θέση Church-Turing: Όλες οι υπολογίσιμες συναρτήσεις είναι μερικές αναδρομικές και όλες οι αποκρίσιμες σχέσεις είναι αναδρομικές.

Υπενθύμιση: Μια σχέση R λέγεται αναδρομική όταν η χαρακτηριστική της συνάρτησης χ_R είναι αναδρομική, όπου:

$$\chi_R(x, y) = \begin{cases} 1 & \text{αν } R(x, y) \\ 0 & \text{αλλιώς} \end{cases}$$

Συνέπεια της θέσης του Church: Για να δείξουμε ότι μια συνάρτηση ή σχέση είναι (μερική) αναδρομική, αντί μιας ακολουθίας τυπικών συνεπαγωγών (με πρωταρχικές συναρτήσεις και σχήματα) ή αντί ενός πλήρους προγράμματος ή Τ.Μ., αρκεί να δίνουμε ενορατικά (αλλά ορθά) επιχειρήματα για τον τρόπο υπολογισμού τους.

Θέση Church-Turing II

Έτσι φαίνεται τελικά οτι καταλήξαμε στον τυπικό πια ορισμό της εννοιας **υπολογίσιμος**. Φυσικά υπάρχουν πολλές συναρτήσεις που δεν είναι υπολογίσιμες.

Θεώρημα

Υπάρχουν μη υπολογίσιμες συναρτήσεις.

Απόδειξη.

Απαριθμούμε όλες τις μερικές αναδρομικές συναρτήσεις $\varphi_0, \varphi_1, \varphi_2, \dots$

Άσκηση: Χρησιμοποιώντας κωδικοποίηση, απαρίθμησε μηχανιστικά την κλάση \mathcal{PR} ή όλα τα προγράμματα **while**.

Άρα υπάρχουν άπειρες μεν, αλλά **μόνο** αριθμήσιμες το πλήθος μερικές αναδρομικές συναρτήσεις (\aleph_0). Από την άλλη μεριά όμως, υπάρχουν μη αριθμήσιμες το πλήθος συναρτήσεις γενικώς (2^{\aleph_0} δείχνεται με διαγωνιοποίηση). □

Θέση Church-Turing III

Παράδειγμα μιας μη υπολογίσιμης συνάρτησης:

Το πρόβλημα τερματισμού (*halting problem*).

$$g(x, y) = \begin{cases} 1, & \text{αν } \varphi_x(y) \downarrow \\ 0, & \text{αλλιώς} \end{cases}$$

Universal Machine I

Καθολικό Πρόγραμμα (universal program): ένα πρόγραμμα που παίρνει για είσοδο οποιοδήποτε **πρόγραμμα** με τα **δεδομένα εισόδου** του και **προσομοιάζει** την εκτέλεσή του. Αυτή η έννοια είναι παρόμοια με τη λειτουργία ενός επόπτη (monitor), μεταφραστή (compiler) ή διερμηνέα (interpreter) σ' ένα πραγματικό υπολογιστή.

Μιλήσαμε για την ισοδυναμία των μοντέλων, άρα μπορούμε να μιλάμε είτε για καθολικό While program, είτε για καθολικό Goto program, είτε για καθολική μηχανή Turing (ιστορικά έτσι πρωτοχρησιμοποιήθηκε η έννοια της καθολικότητας), είτε για καθολική μερική αναδρομική συνάρτηση.

Καθολικό While program $\omega^n(x, y)$ με

- είσοδο $x =$ κωδικός ενός Goto προγράμματος π με n μεταβλητές.
- είσοδο $y =$ κωδικός των n τιμών εισόδου $[a_1, a_2, \dots, a_n]$ για τις n μεταβλητές του π και
- έξοδο την ίδια με την έξοδο του $\pi[a_1, a_2, \dots, a_n]$.

Καθολική αναδρομική συνάρτηση φ_e .

$\forall x, y \varphi_e(x, y) \simeq \varphi_x(y)$

όπου το \simeq σημαίνει: είτε είναι και οι δυο πλευρές ορισμένες και ίσες, είτε καμία πλευρά δεν είναι ορισμένη.

Universal Machine II

πρώτος σκελετός για $\omega^n(x, y)$:

```
label := 0;
while label  $\leq$  maxlab do
  εκτέλεσε εντολή της label;
  βρές επόμενη label
end
```

Δεύτερη προσέγγιση για $\omega^n(x, y)$:

```
label := 0; maxlab := decodesearch(x);
while label  $\leq$  maxlab do
  instruction := decode(x at label);
  label := execnext(instruction,y);
  y:= execvalue(instruction,y)
end
```

Universal Machine III

Εφαρμόζουμε την ακόλουθη κωδικοποίηση:

Κωδικοποίηση για την είσοδο:

$$\langle a_1, a_2, \dots, a_n \rangle \mapsto C_f(a_1, a_2, \dots, a_n)$$

Για τις εντολές του GOTO προγράμματος:

- l_i : assignment **goto** l_j $\mapsto C_f(0, i, C_f(\text{assignment}), j)$ όπου η κωδικοποίηση για τις αναθέσεις, είναι:

Αναθέσεις	Κωδικοποίηση
κενή εντολή	$C_f(0, 0, 0)$
$x_i := 0$	$C_f(1, i, 0)$
$x_i := x_j$	$C_f(2, i, j)$
$x_i := x_j + 1$	$C_f(3, i, j)$
$x_i := x_j - 1$	$C_f(4, i, j)$

- l_i : **if** $x_m <> 0$ **then goto** l_j **else goto** l_k $\mapsto C_f(1, i, m, j, k)$

Έτσι τα προγράμματα goto θα κωδικοποιούνται ως εξής:

$$\text{goto πρόγραμμα με } m \text{ εντολές} \mapsto C_f(C_f(\text{εντολή}_0), \dots, C_f(\text{εντολή}_{m-1}))$$

Universal Machine IV

Τρίτη βελτίωση για $\omega^n(x, y)$:

```

label:=0; maxlab := decodesearch(x);
while label<= maxlab do
  z :=  $D_{\text{label}}^f(x)$ ;
  if  $D_0^f(z) = 0$  then
    ass :=  $D_2^f(z)$ ;
    label :=  $D_3^f(z)$ 
  end
  else
    m :=  $D_2^f(z)$ ;
    if  $m <> 0$  then label :=  $D_3^f(z)$  else label :=  $D_4^f(z)$ 
  end
  y := execvalue(ass, y)
end

```

Η execvalue αποκωδικοποιεί την ass και ανάλογα μεταβάλλει τις τιμές των μεταβλητών του προγράμματος.

Τέταρτη βελτίωση για $\omega^n(x, y)$: κ.ο.κ. Άσκηση.

Θεώρημα Κανονικής μορφής Kleene I

Παρατήρηση

Λόγω της μηχανιστικής ισοδυναμίας των προγραμμάτων **while** και **goto**, έπεται η εξής πρόταση:

Υπάρχει ένα καθολικό πρόγραμμα **while** ω^n , που έχει μόνο ένα “πραγματικό” **while** βρόχο και ίσως πολλούς **for** βρόχους (ή **if** εντολές ...), έτσι ώστε για κάθε πρόγραμμα **while** π και για κάθε είσοδο να ισχύει:

$$\omega^n(\text{κωδικός}(\pi), \text{είσοδος}) \simeq \pi(\text{είσοδος})$$

Θεώρημα Κανονικής μορφής Kleene II

Αν μεταφράσουμε την προηγούμενη πρόταση σε ορολογία μερικών αναδρομικών συναρτήσεων, καταλήγουμε στο εξής:

Θεώρημα κανονικής μορφής Kleene (Kleene normal form theorem)

$\forall n \exists \psi, T \in \mathcal{P}$

$\forall f \in \mathcal{PR}^n$ (\mathcal{PR}^n μερικές αναδρομικές συναρτήσεις με n ορίσματα)

\exists δείκτης x :

$$\forall \vec{y} f(\vec{y}) \simeq \psi(\mu z [T(x, \vec{y}, z) = 0], x, \vec{y})$$

όπου:

f : αντιστοιχεί στο πρόγραμμα π

y : αντιστοιχεί στην είσοδο του π

x : αντιστοιχεί στον κώδικο του π (δείκτης της f)

z : αντιστοιχεί στον αριθμό επαναλήψεων του βρόχου **while** στο ω^n

T : πρωταρχική αναδρομική συνάρτηση που αντιστοιχεί στο σώμα του **while** στο ω^n

ψ : πρωταρχική αναδρομική συνάρτηση που αντιστοιχεί στο τμήμα του ω^n έξω από τον βρόχο **while**

Θεώρημα Κανονικής μορφής Kleene III

Παρατηρήσεις:

- 1 T λέγεται συνήθως *κατηγορημα Kleene*.
- 2 Αρκεί μια μόνο εφαρμογή του τελεστή μ για να περιγραφεί οποιαδήποτε μερική αναδρομική συνάρτηση.
- 3 Το παραπάνω θεώρημα (KNF) δίνει μια μηχανιστική απαρίθμηση $\{\varphi_x\}$ όλων των μερικών αναδρομικών συναρτήσεων. Σαν εφαρμογή θα δείξουμε το εξής (περίφημο) αποτέλεσμα αποκρισιμότητας:

Θεώρημα

Το HP (πρόβλημα τερματισμού), είναι μη επιλύσιμο, δηλαδή η συνάρτηση g δεν είναι αναδρομική όπου:

$$g(x, y) = \begin{cases} 1, & \text{αν } \varphi_x(y) \downarrow \\ 0, & \text{αλλιώς} \end{cases}$$

Θεώρημα Κανονικής μορφής Kleene IV

Απόδειξη (διαγωνιοποίηση και εις άτοπον απαγωγή).

Ας υποθέσουμε ότι η g ήταν αναδρομική. Ορίζουμε:

$$h(x) = \begin{cases} \varphi_x(x) + 1, & \text{αν } g(x, x) = 1 \\ 0, & \text{αλλιώς} \end{cases} = \begin{cases} \varphi_x(x) + 1, & \text{αν } \varphi_x(x) \downarrow \\ 0, & \text{αλλιώς} \end{cases}$$

Η συνάρτηση h μπορεί να υπολογιστεί με τη βοήθεια της υπολογίσιμης g . Αρα, λόγω θέσης Church-Turing, η h είναι (ολική) αναδρομική. Αρα, λόγω της κανονικής μορφής Kleene, έχει κάποιο δείκτη (έστω) y , δηλαδή $\varphi_y = h$ (ολική συνάρτηση). Τότε όμως:

$$\varphi_y(y) = h(y) = \begin{cases} \varphi_y(y) + 1, & \text{αν } \varphi_y(y) \downarrow \\ 0, & \text{αλλιώς} \end{cases} = \underline{\varphi_y(y) + 1}, \quad \text{αφού } \forall y : \varphi_y(y) \downarrow$$

Άτοπο. Συνεπώς η g δεν είναι αναδρομική. □

Θεώρημα Κανονικής μορφής Kleene V

Εναλλακτική απόδειξη του ίδιου με προγράμματα:

Έστω μηχανιστική απαρίθμηση προγραμμάτων: $\pi_0, \pi_1, \pi_2, \dots$. Ας υποθέσουμε ότι υπάρχει πρόγραμμα $\text{halt}(x, y)$ που αποκρίνεται αν το x -οστό πρόγραμμα με είσοδο y σταματάει.

Το εξής πρόγραμμα (με είσοδο x):

l_0 : **if** $\text{halt}(x, x)$ **then goto** l_0 (* loop forever *) **else goto** l_1 (* stop *)

θα εμφανίζεται κάπου στην απαρίθμηση (έστω) με δείκτη i . Τότε όμως:

$\pi_i(i)$ σταματάει **ανν** $\text{not halt}(i, i)$ **ανν** $\pi_i(i)$ δεν σταματάει. **Άτοπο.** □

Μη επιλυσιμότητα I

Θεώρημα

Το TOT (πρόβλημα ολικής συνάρτησης) είναι μη επιλύσιμο, δηλαδή η f δεν είναι αναδρομική όπου:

$$f(x) = \begin{cases} 1, & \text{αν } \varphi_x \text{ είναι ολική} \\ 0, & \text{αλλιώς} \end{cases}$$

Απόδειξη.

Ορίζουμε

$$h(x) = \begin{cases} \varphi_x(x) + 1, & \text{αν } f(x) = 1 \\ 0, & \text{αλλιώς} \end{cases} = \begin{cases} \varphi_x(x) + 1, & \text{αν } \varphi_x \text{ ολική} \\ 0, & \text{αλλιώς} \end{cases}$$

Όπως στο προηγούμενο θεώρημα, η συνάρτηση h θα ήταν υπολογίσιμη αν υποθέταμε ότι η f ήταν αναδρομική, άρα η h θα ήταν (ολική) αναδρομική, άρα θα είχε κάποιο δείκτη y . Τότε:

$$\underline{\varphi_y(y)} = h(y) = \begin{cases} \varphi_y(y) + 1, & \text{αν } \varphi_y \text{ ολική} \\ 0, & \text{αλλιώς} \end{cases} = \underline{\varphi_y(y) + 1}, \text{ αφού } \varphi_y \text{ ολική}$$

Άτοπο.



Αναγωγές

Δηλαδή: δεν υπάρχει αλγόριθμος που να μπορεί να αποφασίζει αν ένα δεδομένο πρόγραμμα σταματά για όλες τις δυνατές εισόδους.

Παρόμοια αποδεικνύεται ότι: δεν υπάρχει αλγόριθμος που να μπορεί να αποφασίσει αν δυο δεδομένα προγράμματα είναι ισοδύναμα (δηλαδή έχουν την ίδια συμπεριφορά εισόδου - εξόδου).

Αντί να δείχνουμε κατευθείαν (δηλαδή με διαγωνιοποίηση) ότι ένα τέτοιο πρόβλημα είναι μη επιλύσιμο, μπορούμε να χρησιμοποιούμε την ακόλουθη **μέθοδο αναγωγής** (reduction):

Για να δείξουμε ότι η f δεν είναι υπολογίσιμη, ανάγουμε το HP στην f , δηλαδή, δείχνουμε ότι αν ήταν υπολογίσιμη η f τότε θα ήταν επιλύσιμο το HP (πράγμα που δεν αληθεύει!).

Θεώρημα $S-m-n$ I

Θεώρησε ένα πρόγραμμα π με δυο εισόδους a, b για τις μεταβλητές y και z . Τώρα (κρατήστε το y σαν παράμετρο) φτιάξε ένα νέο πρόγραμμα

$$\pi' : y := a; \pi$$

με μόνο μια είσοδο b για την μεταβλητή z . Η έξοδος είναι η ίδια. Έτσι αν ο κωδικός του π είναι x και ο κωδικός του π' είναι s ισχύει:

$$\varphi_x(a, z) \simeq \varphi_s(z)$$

Παρατηρούμε ότι ο κωδικός του π' μπορεί να υπολογιστεί βάσει του κωδικού του $y := a$ και του κωδικού του π , με απλό τρόπο. Άρα η s είναι (LOOP-)υπολογίσιμη συνάρτηση του x και του y , δηλαδή $s(x, y)$ και συνεπώς:

Θεώρημα παραμέτρων. $\exists s \in \mathcal{P}, \forall x, y, z \quad \varphi_x(y, z) \simeq \varphi_{s(x,y)}(z)$

Και γενικά (για πολλές μεταβλητές και πολλές παραμέτρους):

Θεώρημα $S-m-n$.

$$\exists S_m^n \in \mathcal{P}, \forall x, \vec{y}, \vec{z} \quad \varphi_x(y_1, \dots, y_m, z_1, \dots, z_n) \simeq \varphi_{S_m^n(x, y_1, \dots, y_m)}(z_1, \dots, z_n)$$

Θεώρημα S - m - n II

Θεώρημα

Υπάρχει $g \in \mathcal{P}$ ώστε $\varphi_x(\varphi_y(z)) \simeq \varphi_{g(x,y)}(z)$. (Δηλαδή ο κωδικός της παράθεσης δυο προγραμμάτων, $\pi_1; \pi_2$, είναι υπολογίσιμος βάσει των κωδικών των π_1 και π_2).

Απόδειξη.

Θεώρησε ένα καθολικό πρόγραμμα: $\omega'(x, y, z) \simeq (\pi_1; \pi_2)(z)$, όπου: $y = \text{κωδικός}(\pi_1)$, $x = \text{κωδικός}(\pi_2)$ και $z = \text{είσοδος για } \pi_1$.

Το ω' έχει (έστω) δείκτη u : $\varphi_u(x, y, z) \simeq \varphi_x(\varphi_y(z))$.

Λόγω του S - m - n έχουμε: $\varphi_u(x, y, z) \simeq \varphi_{S_1^2(u,x,y)}(z)$.

'Αρα: $\varphi_x(\varphi_y(z)) \simeq \varphi_{S_1^2(u,x,y)}(z)$, οπότε θέτουμε: $g(x, y) = S_1^2(u, x, y)$ □

Παράδειγμα αναγωγής I

Θεώρημα

Το *CONST* (πρόβλημα σταθερής συνάρτησης) είναι μη επιλύσιμο, δηλαδή η συνάρτηση f δεν είναι αναδρομική, όπου:

$$f(x) = \begin{cases} 1, & \text{αν } \varphi_x \text{ σταθερή συνάρτηση} \\ 0, & \text{αλλιώς} \end{cases}$$

Απόδειξη:

Ορίζουμε:

$$h(x, y) \simeq \begin{cases} 5, & \text{αν } \varphi_x(x) \downarrow \\ \uparrow, & \text{αλλιώς} \end{cases}$$

Η h είναι υπολογίσιμη:

- βρες το x -οστό πρόγραμμα,
- τρέξε το με είσοδο x ,
- αν και όταν σταματήσει δώσε έξοδο 5.

Παράδειγμα αναγωγής II

Λόγω της θέσης Church-Turing η h είναι μερική αναδρομική. Λόγω της κανονικής μορφής Kleene η h έχει δείκτη (έστω) u : $\varphi_u(x, y) \simeq h(x, y)$. Λόγω του S - m - n : $\varphi_{S(u,x)}(y) \simeq h(x, y)$. Θέτουμε $g(x) = S(u, x)$. Έτσι έχουμε $\varphi_{g(x)}(y) \simeq h(x, y)$ και η g είναι υπολογίσιμη.

Τώρα θα χρησιμοποιήσουμε την f :

$$f(g(x)) = \begin{cases} 1, & \text{αν } \varphi_{g(x)} \text{ σταθ. συν.} \\ 0, & \text{αλλιώς} \end{cases} = \begin{cases} 1, & h(x,y)=5, \forall y \\ 0, & \text{αλλιώς} \end{cases} = \begin{cases} 1, & \varphi_x(x) \downarrow \\ 0, & \text{αλλιώς} \end{cases} = \text{HP}$$

Αν η f ήταν υπολογίσιμη τότε και η σύνθεση $f(g(\cdot))$ θα ήταν υπολογίσιμη δηλαδή το HP θα ήταν επιλύσιμο. □

Παράδειγμα αναγωγής III

Άσκηση: Δείξτε με αναγωγή ότι τα εξής προβλήματα δεν είναι επιλύσιμα:

- 1 $\{(x, y) \mid \varphi_x = \varphi_y\}$
- 2 $\{(x, y, z) \mid \varphi_x(y) = z\}$
- 3 $\{x \mid \text{το πεδίο της } \varphi_x \text{ είναι άπειρο}\}$

Παρατήρηση

Ας διαφοροποιήσουμε τώρα τις δυο εκδοχές του HP που έχουμε χρησιμοποιήσει για να είμαστε ακριβείς:

- $K_0 = \{(x, y) \mid \varphi_x(y) \downarrow\}$
- $K = \{x \mid \varphi_x(x) \downarrow\}$

Θεώρημα Rice I

Η παραπάνω τεχνική αναγωγής του HP μπορεί να χρησιμοποιηθεί για να δείξουμε ένα πολύ πιο γενικό αποτέλεσμα:

Κάθε μη τετριμμένη ιδιότητα υπολογίσιμων συναρτήσεων αντιπροσωπεύει ένα μη επιλύσιμο πρόβλημα (Θεώρημα Rice).

Πριν την ακριβή διατύπωση του θεωρήματος του Rice θα πρέπει να διευκρινίσουμε κάτι πολύ σημαντικό. Οποιαδήποτε τεχνική απαρίθμησης κι αν εφαρμόσουμε για τις μερικές αναδρομικές συναρτήσεις (είτε μέσω προγραμμάτων **while**, είτε μέσω του μαθηματικά ορισμένου επαγωγικού πεδίου κ.τ.λ.), δεν μπορούμε να πετύχουμε μονοσήμαντη κωδικοποίηση για μια συνάρτηση (αναφερόμαστε στην ουσία της συνάρτησης). Δηλαδή μια συνάρτηση, στις διάφορες ισοδύναμες μορφές της, θα αντιστοιχεί σε πολλούς δείκτες στην απαρίθμηση. Έτσι το σύνολο C που αναφέρεται στο θεώρημα του Rice είναι σύνολο συναρτήσεων και όχι σύνολο δεικτών.

Ορισμός

\mathcal{PR}^1 : Η κλάση των μερικών αναδρομικών συναρτήσεων με ένα όρισμα.

Θεώρημα Rice II

Θεώρημα (Rice)

Έστω $C \subsetneq \mathcal{P}\mathbb{R}^1$, $C \neq \emptyset$. Τότε το $\{x \mid \varphi_x \in C\}$ είναι μη επιλύσιμο, δηλαδή η συνάρτηση f δεν είναι αναδρομική, όπου:

$$f(x) = \begin{cases} 1, & \varphi_x \in C \\ 0, & \text{αλλιώς} \end{cases}$$

Απόδειξη:

Έστω k μια συνάρτηση της κλάσης C . Έστω Ω η πουθενά ορισμένη συνάρτηση, δηλαδή $\Omega(x) = \uparrow, \forall x$.

Περίπτωση 1: $\Omega \notin C$.

$$\text{Ορίζουμε: } h(x, y) \simeq \begin{cases} k(y), & \varphi_x(x) \downarrow \\ \uparrow, & \text{αλλιώς} \end{cases}$$

Θεώρημα Rice III

Όπως στο προηγούμενο θεώρημα έχουμε: $\varphi_{g(x)}(y) \simeq h(x, y)$ και g είναι υπολογίσιμη. Τότε, αφού $\Omega \notin C$:

$$f(g(x)) = \begin{cases} 1, & \varphi_{g(x)} \in C \\ 0, & \text{αλλιώς} \end{cases} = \begin{cases} 1, & \varphi_{g(x)} \simeq k \\ 0, & \text{αλλιώς} \end{cases} = \begin{cases} 1, & \varphi_x(x) \downarrow \\ 0, & \text{αλλιώς} \end{cases} = \text{HP}$$

Συνεπώς η f δεν είναι υπολογίσιμη.

Περίπτωση 2: $\Omega \in C$. Αλλά τότε το Ω δεν ανήκει στο \bar{C} . Δείχνουμε ομοίως ότι: $\{x \mid \varphi_x \in \bar{C}\}$ μη επιλύσιμο. Αλλά τότε και το συμπλήρωμά του δεν είναι επιλύσιμο:

$$\overline{\{x \mid \varphi_x \in \bar{C}\}} = \{x \mid \varphi_x \in C\}$$

