Περιγραφική Πολυπλοκότητα

Πολυπλοκότητα: Βασικοί ορισμοί, παραδείγματα κλπ

Ανδρέας Αβουκάτος, 7115142200001

ΑΛΜΑ

Δευτέρα, 06/03/2023

(Προαπαιτούμενα: μηχανές Turing)

(Προαπαιτούμενα: μηχανές Turing)

- $L(M) = \{w \in \{0,1\}^* \mid M(w)\downarrow\}$, όπου $M(w)\downarrow$ η μηχανή M αποδέχεται τη w

(Προαπαιτούμενα: μηχανές Turing)

- $L(M) = \{w \in \{0,1\}^* \mid M(w)\downarrow\}$, όπου $M(w)\downarrow$ η μηχανή M αποδέχεται τη w
- $T(w)$ = ό,τι έχει η MT στην ταινία της όταν τελειώσει τον υπολογισμό με είσοδο w, $T(w) \in \{0,1\}^*$

(Προαπαιτούμενα: μηχανές Turing)

- $L(M) = \{w \in \{0,1\}^* \mid M(w) \downarrow\}$, όπου $M(w) \downarrow$ η μηχανή M αποδέχεται τη w
- $T(w) =$ ό,τι έχει η MT στην ταινία της όταν τελειώσει τον υπολογισμό με είσοδο $w, T(w) \in \{0,1\}^*$

(Προαπαιτούμενα: μηχανές Turing)

- $L(M) = \{w \in \{0,1\}^* \mid M(w)\downarrow\}$, όπου $M(w)\downarrow$ η μηχανή M αποδέχεται τη w
- $T(w) =$ ό,τι έχει η MT στην ταινία της όταν τελειώσει τον υπολογισμό με είσοδο w, $T(w) \in \{0,1\}^*$

Μπορούμε να δούμε τις MT ως queries:

(Προαπαιτούμενα: μηχανές Turing)

- $L(M) = \{w \in \{0,1\}^* \mid M(w)\downarrow\}$, όπου $M(w)\downarrow$ η μηχανή M αποδέχεται τη w
- $T(w)$ = ό,τι έχει η MT στην ταινία της όταν τελειώσει τον υπολογισμό με είσοδο $w, T(w) \in \{0,1\}^*$

Μπορούμε να δούμε τις MT ως queries: χρειαζόμαστε κωδικοποίηση από δομές (με ό,τι να 'ναι voc) σε δυαδικές συμβολοσειρές

$$\mathrm{bin}_\tau : \mathrm{STRUC}[\tau] \to \mathrm{STRUC}[\tau_S],\ \ \tau_S = \langle S^1 \rangle$$

$$\mathrm{bin}_\tau : \mathrm{STRUC}[\tau] \to \mathrm{STRUC}[\tau_S], \;\; \tau_S = \left\langle S^1 \right\rangle$$

Ας είναι $\mathcal{A} = \left\langle \{0, 1, \ldots, n-1\}, R_1^{\mathcal{A}}, \ldots, R_r^{\mathcal{A}}, c_1, \ldots, c_s \right\rangle$ διατεταγμένη δομή του τ

$$\text{bin}_\tau : \text{STRUC}[\tau] \to \text{STRUC}[\tau_S],\ \ \tau_S = \left\langle S^1 \right\rangle$$

Ας είναι $\mathcal{A} = \left\langle \{0, 1, \ldots, n-1\}, R_1^{\mathcal{A}}, \ldots, R_r^{\mathcal{A}}, c_1, \ldots, c_s \right\rangle$ διατεταγμένη δομή του τ

- Για σχέση $R_i^{\mathcal{A}} \subseteq |\mathcal{A}|^{a_i}$, ορίζουμε $\text{bin}^{\mathcal{A}}(R_i)$ binary string μήκους n^{a_i}, όπου μονάδα στη θέση j σημαίνει ότι το j-οστό tuple ανήκει στη $R_i^{\mathcal{A}}$

$$\mathrm{bin}_\tau : \mathrm{STRUC}[\tau] \to \mathrm{STRUC}[\tau_S],\ \ \tau_S = \left\langle S^1 \right\rangle$$

Ας είναι $\mathcal{A} = \left\langle \{0, 1, \ldots, n-1\}, R_1^{\mathcal{A}}, \ldots, R_r^{\mathcal{A}}, c_1, \ldots, c_s \right\rangle$ διατεταγμένη δομή του τ

- Για σχέση $R_i^{\mathcal{A}} \subseteq |\mathcal{A}|^{a_i}$, ορίζουμε $\mathrm{bin}^{\mathcal{A}}(R_i)$ binary string μήκους n^{a_i}, όπου μονάδα στη θέση j σημαίνει ότι το j-οστό tuple ανήκει στη $R_i^{\mathcal{A}}$
- Για σταθ $c_j^{\mathcal{A}} \in |A|$, ορίζουμε $\mathrm{bin}^{\mathcal{A}}(c_j)$ τη δυαδική αναπαρ. του c_j, μήκους $\lceil \log n \rceil$

$$\text{bin}_\tau : \text{STRUC}[\tau] \to \text{STRUC}[\tau_S],\ \ \tau_S = \left\langle S^1 \right\rangle$$

Ας είναι $\mathcal{A} = \left\langle \{0, 1, \ldots, n-1\}, R_1^{\mathcal{A}}, \ldots, R_r^{\mathcal{A}}, c_1, \ldots, c_s \right\rangle$ διατεταγμένη δομή του τ

- Για σχέση $R_i^{\mathcal{A}} \subseteq |\mathcal{A}|^{a_i}$, ορίζουμε $\text{bin}^{\mathcal{A}}(R_i)$ binary string μήκους n^{a_i}, όπου μονάδα στη θέση j σημαίνει ότι το j-οστό tuple ανήκει στη $R_i^{\mathcal{A}}$
- Για σταθ $c_j^{\mathcal{A}} \in |A|$, ορίζουμε $\text{bin}^{\mathcal{A}}(c_j)$ τη δυαδική αναπαρ. του c_j, μήκους $\lceil \log n \rceil$

$$\text{bin}_\tau : \text{STRUC}[\tau] \to \text{STRUC}[\tau_S],\ \ \tau_S = \left\langle S^1 \right\rangle$$

Ας είναι $\mathcal{A} = \left\langle \{0, 1, \ldots, n-1\}, R_1^{\mathcal{A}}, \ldots, R_r^{\mathcal{A}}, c_1, \ldots, c_s \right\rangle$ διατεταγμένη δομή του τ

- Για σχέση $R_i^{\mathcal{A}} \subseteq |\mathcal{A}|^{a_i}$, ορίζουμε $\text{bin}^{\mathcal{A}}(R_i)$ binary string μήκους n^{a_i}, όπου μονάδα στη θέση j σημαίνει ότι το j-οστό tuple ανήκει στη $R_i^{\mathcal{A}}$
- Για σταθ $c_j^{\mathcal{A}} \in |A|$, ορίζουμε $\text{bin}^{\mathcal{A}}(c_j)$ τη δυαδική αναπαρ. του c_j, μήκους $\lceil \log n \rceil$

Τελικά, $\text{bin}_\tau(\mathcal{A}) = \text{bin}^{\mathcal{A}}(R_1) \cdots \text{bin}^{\mathcal{A}}(R_r)\,\text{bin}^{\mathcal{A}}(c_1) \cdots \text{bin}^{\mathcal{A}}(c_s)$ το concatenation των επιμέρους συνιστωσών.
Μήκος $\hat{n}_\tau = \|\text{bin}\,\tau(\mathcal{A})\| = n^{a_1} + n^{a_2} + \cdots + n^{a_r} + s\lceil \log n \rceil$

Παράδειγμα κωδικοποίησης

Ας είναι
$\mathcal{A} = \left\langle \{0,1,2\}, S^1 = \{0,2\}, R^2 = \{(0,1),(1,2)\}, c^{\mathcal{A}} = 2 \right\rangle, n = 3$

- $\left|\text{bin}_\tau(S^1)\right| = n^{a_1} = 3^1$, και $\text{bin}_\tau(S^1) = 1^0 0^1 1^2$

Παράδειγμα κωδικοποίησης

Ας είναι
$\mathcal{A} = \left\langle \{0, 1, 2\}, S^1 = \{0, 2\}, R^2 = \{(0, 1), (1, 2)\}, c^{\mathcal{A}} = 2 \right\rangle, n = 3$

- $\left|\text{bin}_\tau(S^1)\right| = n^{a_1} = 3^1$, και $\text{bin}_\tau(S^1) = 1^0 0^1 1^2$
- $\left|\text{bin}_\tau(R^2)\right| = n^{a_2} = 3^2 = 9$, και
  $\text{bin}_\tau(R^2) = 0^{00} 1^{01} 0^{02} 0^{10} 0^{11} 1^{12} 0^{20} 0^{21} 0^{22}$

Παράδειγμα κωδικοποίησης

Ας είναι

$\mathcal{A} = \left\langle \{0,1,2\}, S^1 = \{0,2\}, R^2 = \{(0,1),(1,2)\}, c^{\mathcal{A}} = 2 \right\rangle, n = 3$

- $\left|\text{bin}_\tau(S^1)\right| = n^{a_1} = 3^1$, και $\text{bin}_\tau(S^1) = 1^0 0^1 1^2$
- $\left|\text{bin}_\tau(R^2)\right| = n^{a_2} = 3^2 = 9$, και $\text{bin}_\tau(R^2) = 0^{00}1^{01}0^{02}0^{10}0^{11}1^{12}0^{20}0^{21}0^{22}$
- $\text{bin}_\tau(c) = 10$

Παράδειγμα κωδικοποίησης

Ας είναι

$\mathcal{A} = \left\langle \{0,1,2\}, S^1 = \{0,2\}, R^2 = \{(0,1),(1,2)\}, c^{\mathcal{A}} = 2 \right\rangle, n = 3$

- $\left|\mathrm{bin}_\tau(S^1)\right| = n^{a_1} = 3^1$, και $\mathrm{bin}_\tau(S^1) = 1^0 0^1 1^2$
- $\left|\mathrm{bin}_\tau(R^2)\right| = n^{a_2} = 3^2 = 9$, και $\mathrm{bin}_\tau(R^2) = 0^{00}1^{01}0^{02}0^{10}0^{11}1^{12}0^{20}0^{21}0^{22}$
- $\mathrm{bin}_\tau(c) = 10$

Παράδειγμα κωδικοποίησης

Ας είναι
$\mathcal{A} = \left\langle \{0,1,2\}, S^1 = \{0,2\}, R^2 = \{(0,1),(1,2)\}, c^{\mathcal{A}} = 2 \right\rangle, n = 3$

- $\left|\text{bin}_\tau(S^1)\right| = n^{a_1} = 3^1$, και $\text{bin}_\tau(S^1) = 1^0 0^1 1^2$
- $\left|\text{bin}_\tau(R^2)\right| = n^{a_2} = 3^2 = 9$, και $\text{bin}_\tau(R^2) = 0^{00}1^{01}0^{02}0^{10}0^{11}1^{12}0^{20}0^{21}0^{22}$
- $\text{bin}_\tau(c) = 10$

Τελικά,

$$\text{bin}(\mathcal{A}) = \underbrace{101}_{R^{\mathcal{A}}}\underbrace{010001000}_{S^{\mathcal{A}}}\underbrace{10}_{c^{\mathcal{A}}}$$

Τυπικά,
$\text{bin}_\tau(\mathcal{A}) = \left\langle \{0,1,2,\ldots,13\}, S^1 = \{0,2,4,8,12\} \right\rangle \in \text{STRUC}[\tau_S]$

Ειδική περ., σχόλια

Αν «$\tau = \emptyset$» ορίζουμε $\text{bin}(\mathcal{A}) = 0^{||A||}$, ώστε $||\text{bin}(\mathcal{A})|| \geq ||\mathcal{A}||$

Ειδική περ., σχόλια

Αν «$\tau = \emptyset$» ορίζουμε $\text{bin}(\mathcal{A}) = 0^{\|A\|}$, ώστε $\|\text{bin}(\mathcal{A})\| \geq \|\mathcal{A}\|$

Διευκρινήσεις

	CT	DC
n	μεγ. εισόδου λέξης w	μεγ. input structure $\|\mathcal{A}\|$

Ειδική περ., σχόλια

Αν «$\tau = \emptyset$» ορίζουμε $\text{bin}(\mathcal{A}) = 0^{||A||}$, ώστε $||\text{bin}(\mathcal{A})|| \geq ||\mathcal{A}||$

Διευκρινήσεις

	CT	DC
n	μεγ. εισόδου λέξης w	μεγ. input structure $\|\|\mathcal{A}\|\|$

Όταν εναπόκειται κάποια δομή να εισαχθεί σε MT, τότε το μήκος της εισόδου είναι $\hat{n}_\tau, \hat{n}$

Ειδική περ., σχόλια

Αν «$\tau = \emptyset$» ορίζουμε $\text{bin}(\mathcal{A}) = 0^{||A||}$, ώστε $||\text{bin}(\mathcal{A})|| \geq ||\mathcal{A}||$

Διευκρινήσεις

	CT	DC
n	μεγ. εισόδου λέξης w	μεγ. input structure $\|\|\mathcal{A}\|\|$

Όταν εναπόκειται κάποια δομή να εισαχθεί σε MT, τότε το μήκος της εισόδου είναι $\hat{n}_\tau, \hat{n}$

Ασκ: $\text{bin}, \text{bin}^{-1}$ είναι FO Queries (ορίζονται με τύπους)

Ειδική περ., σχόλια

Αν «$\tau = \emptyset$» ορίζουμε $\text{bin}(\mathcal{A}) = 0^{||A||}$, ώστε $||\text{bin}(\mathcal{A})|| \geq ||\mathcal{A}||$

Διευκρινήσεις

	CT	DC
n	μεγ. εισόδου λέξης w	μεγ. input structure $\|\|\mathcal{A}\|\|$

Όταν εναπόκειται κάποια δομή να εισαχθεί σε ΜΤ, τότε το μήκος της εισόδου είναι $\hat{n}_\tau, \hat{n}$

Ασκ: $\text{bin}, \text{bin}^{-1}$ είναι FO Queries (ορίζονται με τύπους)

Σχόλια για κωδικοποίηση - αποκωδικοποίηση:

- Πρέπει να είναι υπολογιστικά εύκολη

Ειδική περ., σχόλια

Αν «$\tau = \emptyset$» ορίζουμε $\text{bin}(\mathcal{A}) = 0^{||A||}$, ώστε $||\text{bin}(\mathcal{A})|| \geq ||\mathcal{A}||$

Διευκρινήσεις

	CT	DC
n	μεγ. εισόδου λέξης w	μεγ. input structure $\|\|\mathcal{A}\|\|$

Όταν εναπόκειται κάποια δομή να εισαχθεί σε ΜΤ, τότε το μήκος της εισόδου είναι $\hat{n}_\tau, \hat{n}$

Ασκ: $\text{bin}, \text{bin}^{-1}$ είναι FO Queries (ορίζονται με τύπους)

Σχόλια για κωδικοποίηση - αποκωδικοποίηση:

- Πρέπει να είναι υπολογιστικά εύκολη
- Πρέπει να 'ναι χωρικά αποδοτική

Query calculation

Ας είναι $I : \mathrm{STRUC}[\sigma] \to \mathrm{STRUC}[\tau]$ query και T μηχTur:
$\forall \mathcal{A} \in \mathrm{STRUC}[\sigma] : T(\mathrm{bin}(A)) = \mathrm{bin}(I(\mathcal{A}))$,

Query calculation

Ας είναι $I : \mathrm{STRUC}[\sigma] \to \mathrm{STRUC}[\tau]$ query και T μηχTur:
$\forall \mathcal{A} \in \mathrm{STRUC}[\sigma] : T(\mathrm{bin}(A)) = \mathrm{bin}(I(\mathcal{A}))$, λέμε ότι «η T υπολογίζει το I».

Query calculation

Ας είναι $I : \mathrm{STRUC}[\sigma] \to \mathrm{STRUC}[\tau]$ query και T μηχTur: $\forall \mathcal{A} \in \mathrm{STRUC}[\sigma] : T(\mathrm{bin}(A)) = \mathrm{bin}(I(\mathcal{A}))$, λέμε ότι «η T υπολογίζει το I».

Complexity Classes

- $\mathrm{DTIME}(t(n)) =$
  { boolean queries | υπολ. από NTET πολυταιν MT χρόνου $O(t(n))$}

Query calculation

Ας είναι $I : \mathrm{STRUC}[\sigma] \to \mathrm{STRUC}[\tau]$ query και T μηχTur:
$\forall \mathcal{A} \in \mathrm{STRUC}[\sigma] : T(\mathrm{bin}(A)) = \mathrm{bin}(I(\mathcal{A}))$, λέμε ότι «η T υπολογίζει το I».

Complexity Classes

- $\mathrm{DTIME}(t(n)) =$
  $\{$ boolean queries $|$ υπολ. από ΝΤΕΤ πολυταιν ΜΤ χρόνου$\mathcal{O}(t(n))\}$
- NTIME

Query calculation

Ας είναι $I : \mathrm{STRUC}[\sigma] \to \mathrm{STRUC}[\tau]$ query και T μηχTur: $\forall \mathcal{A} \in \mathrm{STRUC}[\sigma] : T(\mathrm{bin}(A)) = \mathrm{bin}(I(\mathcal{A}))$, λέμε ότι «η T υπολογίζει το I».

Complexity Classes

- $\mathrm{DTIME}(t(n)) =$ $\{$ boolean queries $|$ υπολ. από ΝΤΕΤ πολυταιν ΜΤ χρόνου$O(t(n))\}$
- NTIME
- DSPACE, NSPACE

Query calculation

Ας είναι $I : \mathrm{STRUC}[\sigma] \to \mathrm{STRUC}[\tau]$ query και T μηχTur: $\forall \mathcal{A} \in \mathrm{STRUC}[\sigma] : T(\mathrm{bin}(A)) = \mathrm{bin}(I(\mathcal{A}))$, λέμε ότι «η T υπολογίζει το I».

Complexity Classes

- $\mathrm{DTIME}(t(n)) =$
  {boolean queries | υπολ. από NTET πολυταιν MT χρόνου$O(t(n))$}
- NTIME
- DSPACE, NSPACE
- $\mathrm{L} = \mathrm{DSPACE}(\log n), \mathrm{NL} = \mathrm{NSPACE}(\log n)$

Query calculation

Ας είναι $I : \mathrm{STRUC}[\sigma] \to \mathrm{STRUC}[\tau]$ query και T μηχTur: $\forall \mathcal{A} \in \mathrm{STRUC}[\sigma] : T(\mathrm{bin}(A)) = \mathrm{bin}(I(\mathcal{A}))$, λέμε ότι «η T υπολογίζει το I».

Complexity Classes

- $\mathrm{DTIME}(t(n)) =$ $\{$ boolean queries $|$ υπολ. από ΝΤΕΤ πολυταιν ΜΤ χρόνου$O(t(n))\}$
- NTIME
- DSPACE, NSPACE
- $\mathrm{L} = \mathrm{DSPACE}(\log n), \mathrm{NL} = \mathrm{NSPACE}(\log n)$
- $\mathrm{P} = \cup_k \mathrm{DTIME}\, n^k, \mathrm{NP} = \cup_k \mathrm{NTIME}(n^k)$

Ερωτήματα, επεκτάσεις

Γιατί χρησιμοποιούμε boolean queries;

Ερωτήματα, επεκτάσεις

Γιατί χρησιμοποιούμε boolean queries;

Επειδή στους ορισμούς έχουμε προβλήματα απόφασης, τα οποία εκφράζονται ως υπολογισμοί χαρακτηριστικής συνάρτησης - μιας συνάρτησης με εικόνα $\{0, 1\}$, όπως τα boolean queries.

Ερωτήματα, επεκτάσεις

Γιατί χρησιμοποιούμε boolean queries;

Επειδή στους ορισμούς έχουμε προβλήματα απόφασης, τα οποία εκφράζονται ως υπολογισμοί χαρακτηριστικής συνάρτησης - μιας συνάρτησης με εικόνα $\{0, 1\}$, όπως τα boolean queries.

Compl Class & Queries

Για κλάση πολυπλ. C ορίζουμε $Q(C)$ το σύνολο των queries που υπολ. στην κλάση C

Ερωτήματα, επεκτάσεις

Γιατί χρησιμοποιούμε boolean queries;

Επειδή στους ορισμούς έχουμε προβλήματα απόφασης, τα οποία εκφράζονται ως υπολογισμοί χαρακτηριστικής συνάρτησης - μιας συνάρτησης με εικόνα $\{0, 1\}$, όπως τα boolean queries.

Compl Class & Queries

Για κλάση πολυπλ. C ορίζουμε $Q(C)$ το σύνολο των queries που υπολ. στην κλάση C

Αφού η κλάση C έχει μόνο boolean queries, πώς μπορούμε να πούμε ότι ένα general query υπολογίζεται στο C;

Ερωτήματα, επεκτάσεις

Γιατί χρησιμοποιούμε boolean queries;

Επειδή στους ορισμούς έχουμε προβλήματα απόφασης, τα οποία εκφράζονται ως υπολογισμοί χαρακτηριστικής συνάρτησης - μιας συνάρτησης με εικόνα $\{0, 1\}$, όπως τα boolean queries.

Compl Class & Queries

Για κλάση πολυπλ. C ορίζουμε $Q(C)$ το σύνολο των queries που υπολ. στην κλάση C

Αφού η κλάση C έχει μόνο boolean queries, πώς μπορούμε να πούμε ότι ένα general query υπολογίζεται στο C;

Αρκεί με είσοδο $\text{bin}(\mathcal{A})$, κάθε bit του $\text{bin}(I(\mathcal{A}))$ να είναι ομοιόμορφα υπολογίσιμο στο C

Τυπικός ορ.

Ας είναι $I : \mathrm{STRUC}[\sigma] \rightarrow \mathrm{STRUC}[\tau]$ query.

Τυπικός ορ.

Ας είναι $I : \mathrm{STRUC}[\sigma] \to \mathrm{STRUC}[\tau]$ query. Λέμε ότι το I είναι υπολογίσιμο στο C ανν $I_b \in C$, με

$$I_b = \{\, (\mathcal{A}, i, a) \mid \text{ το } i\text{-οστό bit του } \mathrm{bin}(I(\mathcal{A})) \text{ είναι } a\}$$

Τυπικός ορ.

Ας είναι $I : \mathrm{STRUC}[\sigma] \to \mathrm{STRUC}[\tau]$ query. Λέμε ότι το I είναι υπολογίσιμο στο C ανν $I_b \in C$, με

$$I_b = \{ (\mathcal{A}, i, a) \mid \text{ το } i\text{-οστό bit του} \operatorname{bin}(I(\mathcal{A})) \text{ είναι } a \}$$

Πλέον, $Q(C) = C \cup \{ I \mid I_b \in C \}$

Time - space constr func

$s:\mathbb{N}\to\mathbb{N}$ είναι space constructible (resp. time constr.) ανν υπάρχει ντετ ΜΤ χώρου $O(s(n))$ (resp. χρόνου $O(s(n))$) τ.ω.

- με είσοδο 0^n

Time - space constr func

$s : \mathbb{N} \to \mathbb{N}$ είναι space constructible (resp. time constr.) ανν υπάρχει ντετ ΜΤ χώρου $O(s(n))$ (resp. χρόνου $O(s(n))$) τ.ω.

- με είσοδο 0^n
- υπολογίζει $s(n)$ σε binary

Time - space constr func

$s : \mathbb{N} \to \mathbb{N}$ είναι space constructible (resp. time constr.) ανν υπάρχει ντετ ΜΤ χώρου $O(s(n))$ (resp. χρόνου $O(s(n))$) τ.ω.

- με είσοδο 0^n
- υπολογίζει $s(n)$ σε binary

Time - space constr func

$s : \mathbb{N} \to \mathbb{N}$ είναι space constructible (resp. time constr.) ανν υπάρχει ντετ MT χώρου $O(s(n))$ (resp. χρόνου $O(s(n))$) τ.ω.

- με είσοδο 0^n
- υπολογίζει $s(n)$ σε binary

Ισοδύναμα, ανν $s' \in Q(\text{DSPACE}[s(n)]))$, resp. $s' \in Q(\text{DTIME}[s(n)])$ όπου s' η συν. που υπολογίζει την $s(n)$ με είσοδο 0^n

Time - space constr func

$s : \mathbb{N} \to \mathbb{N}$ είναι space constructible (resp. time constr.) ανν υπάρχει ντετ MT χώρου $\mathcal{O}(s(n))$ (resp. χρόνου $\mathcal{O}(s(n))$) τ.ω.

- με είσοδο 0^n
- υπολογίζει $s(n)$ σε binary

Ισοδύναμα, ανν $s' \in Q(\mathrm{DSPACE}[s(n)])$), resp.$s' \in Q(\mathrm{DTIME}[s(n)])$ όπου s' η συν. που υπολογίζει την $s(n)$ με είσοδο 0^n

Space Hierarchy Theorem

Για κάθε t, s space constr συναρτήσεις με $(t/s)(n) \xrightarrow{n\to\infty} 0$, τότε $\mathrm{DSPACE}[t(n)] \subsetneq \mathrm{DSPACE}[s(n)]$

Time - space constr func

$s : \mathbb{N} \to \mathbb{N}$ είναι space constructible (resp. time constr.) ανν υπάρχει ντετ MT χώρου $O(s(n))$ (resp. χρόνου $O(s(n))$) τ.ω.

- με είσοδο 0^n
- υπολογίζει $s(n)$ σε binary

Ισοδύναμα, ανν $s' \in Q(\mathrm{DSPACE}[s(n)])$, resp. $s' \in Q(\mathrm{DTIME}[s(n)])$ όπου s' η συν. που υπολογίζει την $s(n)$ με είσοδο 0^n

Space Hierarchy Theorem

Για κάθε t, s space constr συναρτήσεις με $(t/s)(n) \xrightarrow{n\to\infty} 0$, τότε $\mathrm{DSPACE}[t(n)] \subsetneq \mathrm{DSPACE}[s(n)]$

Savitch's Theorem

$\mathrm{DSPACE}[s(n)] \subseteq \mathrm{NSPACE}[s(n)] \subseteq \mathrm{DSPACE}[s^2(n)]$

'query A είναι Turing reducible B αν
A υπολογίζεται εύκολα δοθέντος μαντείου για B'

Oracle Turing Machine

Είναι μια MT M με

- έξτρα ταινία ερώτησης

'query A είναι Turing reducible B αν
A υπολογίζεται εύκολα δοθέντος μαντείου για B'

Oracle Turing Machine

Είναι μια MT M με

- έξτρα ταινία ερώτησης
- ξεχωριστή query state

'query A είναι Turing reducible B αν
A υπολογίζεται εύκολα δοθέντος μαντείου για B'

Oracle Turing Machine

Είναι μια MT M με

- έξτρα ταινία ερώτησης
- ξεχωριστή query state

'query A είναι Turing reducible B αν
A υπολογίζεται εύκολα δοθέντος μαντείου για B'

Oracle Turing Machine

Είναι μια MT M με

- έξτρα ταινία ερώτησης
- ξεχωριστή query state

Δουλεύει όπως κλασική MT, μόνο που όταν μπει στο query state, υπάρχει γραμμένη λέξη $w = \mathrm{bin}(\mathcal{A})$ στην ταινία ερώτησης.

'query A είναι Turing reducible B αν
A υπολογίζεται εύκολα δοθέντος μαντείου για B'

Oracle Turing Machine

Είναι μια MT M με

- έξτρα ταινία ερώτησης
- ξεχωριστή query state

Δουλεύει όπως κλασική MT, μόνο που όταν μπει στο query state, υπάρχει γραμμένη λέξη $w = \mathrm{bin}(\mathcal{A})$ στην ταινία ερώτησης. Αν $\mathcal{A} \in B$, τότε ¨εμφανίζεται' 1 στην ταινία ερωτ., αλλιώς εμφ. 0, υποθετικά σε ένα βήμα.

‘query A είναι Turing reducible B αν
Α υπολογίζεται εύκολα δοθέντος μαντείου για Β’

Oracle Turing Machine

Είναι μια ΜΤ M με

- έξτρα ταινία ερώτησης
- ξεχωριστή query state

Δουλεύει όπως κλασική ΜΤ, μόνο που όταν μπει στο query state, υπάρχει γραμμένη λέξη $w = \text{bin}(\mathcal{A})$ στην ταινία ερώτησης. Αν $\mathcal{A} \in B$, τότε ¨εμφανίζεται’ 1 στην ταινία ερωτ., αλλιώς εμφ. 0, υποθετικά σε ένα βήμα.
Αν το μαντείο είναι για το query B, γράφουμε M^B

Παράδειγμα Turing αναγωγής

$\mathcal{C}$-Turing reducible

Αν A, B queries και C κλάση πολυπλοκότητας, λέμε ότι το A είναι C – Turing reduc. στο B, συμβ $A \leq_C^T B$

Παράδειγμα Turing αναγωγής

C-Turing reducible

Αν A, B queries και C κλάση πολυπλοκότητας, λέμε ότι το A είναι C – Turing reduc. στο B, συμβ $A \leq_C^T B$
ανν υπάρχει OTM M^B τ.ω. $L(M^B) = A$ και M^B τρέχει σε C

Παράδειγμα Turing αναγωγής

C-Turing reducible

Αν A, B queries και C κλάση πολυπλοκότητας, λέμε ότι το A είναι C – Turing reduc. στο B, συμβ $A \leq_C^T B$
ανν υπάρχει OTM M^B τ.ω. $L(M^B) = A$ και M^B τρέχει σε C

Π

Παράδειγμα Turing αναγωγής

C-Turing reducible

Αν A, B queries και C κλάση πολυπλοκότητας, λέμε ότι το A είναι C – Turing reduc. στο B, συμβ $A \leq_C^T B$
ανν υπάρχει OTM M^B τ.ω. $L(M^B) = A$ και M^B τρέχει σε C

ΠΧ

Παράδειγμα Turing αναγωγής

C-Turing reducible

Αν A, B queries και C κλάση πολυπλοκότητας, λέμε ότι το A είναι C – Turing reduc. στο B, συμβ $A \leq_C^T B$
ανν υπάρχει OTM M^B τ.ω. $L(M^B) = A$ και M^B τρέχει σε C

ΠΧ:

Παράδειγμα Turing αναγωγής

C-Turing reducible

Αν A, B queries και C κλάση πολυπλοκότητας, λέμε ότι το A είναι C – Turing reduc. στο B, συμβ $A \leq_C^T B$
ανν υπάρχει OTM M^B τ.ω. $L(M^B) = A$ και M^B τρέχει σε C

ΠΧ: • Ας είναι boolean query CLIQUE πάνω σε δομές του $\tau_{gk} = \left\langle E^2, k \right\rangle$ που είναι αληθής ανν

Παράδειγμα Turing αναγωγής

C-Turing reducible

Αν A, B queries και C κλάση πολυπλοκότητας, λέμε ότι το A είναι C – Turing reduc. στο B, συμβ $A \leq_C^T B$
ανν υπάρχει OTM M^B τ.ω. $L(M^B) = A$ και M^B τρέχει σε C

ΠΧ: • Ας είναι boolean query CLIQUE πάνω σε δομές του $\tau_{gk} = \left\langle E^2, k \right\rangle$ που είναι αληθής ανν το input structure G περιέχει πλήρες υπογράφ. μεγέθους k

Παράδειγμα Turing αναγωγής

C-Turing reducible

Αν A, B queries και C κλάση πολυπλοκότητας, λέμε ότι το A είναι C – Turing reduc. στο B, συμβ $A \leq_C^T B$
ανν υπάρχει OTM M^B τ.ω. $L(M^B) = A$ και M^B τρέχει σε C

ΠΧ: • Ας είναι boolean query CLIQUE πάνω σε δομές του $\tau_{gk} = \langle E^2, k \rangle$ που είναι αληθής ανν το input structure G περιέχει πλήρες υπογράφ. μεγέθους k

• Ας είναι general query MAXCLIQUE πάνω σε τ_g που επιστρέφει το μέγεθος της μέγιστης κλίκας

Δ.ο.: $\text{MAXCLIQUE}_b \leq_P^T \text{CLIQUE}$

Υπενθύμιση

$\text{MAXCLIQUE}_b =$
$\{\, (G, i, a) \mid$ το $i-$οστό bit του $\text{bin}(MC(G))$ είναι $a\}$

Δ.ο.: MAXCLIQUE$_b \leq_P^T$ CLIQUE

Υπενθύμιση

MAXCLIQUE$_b$ =
$\{\, (G, i, a) \mid$ το $i-$οστό bit του $\mathrm{bin}(MC(G))$ είναι $a\}$

Για είσοδο (G, i, a):

Δ.ο.: $\text{MAXCLIQUE}_b \leq_P^T$ CLIQUE

Υπενθύμιση

$\text{MAXCLIQUE}_b =$
$\{ (G, i, a) \mid$ το $i-$οστό bit του $\text{bin}(MC(G))$ είναι $a \}$

Για είσοδο (G, i, a):

- binary search με μαντείο του CLIQUE, για να βρούμε μέγεθος s μέγιστης κλίκας: $(G, n/2) \in$ CLIQUE?

Δ.ο.: $\text{MAXCLIQUE}_b \leq_P^T$ CLIQUE

Υπενθύμιση

$\text{MAXCLIQUE}_b =$
$\{ (G, i, a) \mid$ το $i-$οστό bit του $\text{bin}(MC(G))$ είναι $a\}$

Για είσοδο (G, i, a):

- binary search με μαντείο του CLIQUE, για να βρούμε μέγεθος s μέγιστης κλίκας: $(G, n/2) \in$ CLIQUE?
  - Ναι. $(G, 3n/4) \in$ CLIQUE?

Δ.ο.: MAXCLIQUE$_b \leq_P^T$ CLIQUE

Υπενθύμιση

MAXCLIQUE$_b$ =
$\{ (G, i, a) \mid$ το i – οστό bit του $\mathrm{bin}(MC(G))$ είναι $a\}$

Για είσοδο (G, i, a):

- binary search με μαντείο του CLIQUE, για να βρούμε μέγεθος s μέγιστης κλίκας: $(G, n/2) \in$ CLIQUE?
  - Ναι. $(G, 3n/4) \in$ CLIQUE?
  - Όχι. $(G, n/4) \in$ CLIQUE?

Δ.ο.: MAXCLIQUE$_b \leq_P^T$ CLIQUE

Υπενθύμιση

MAXCLIQUE$_b$ =
$\{ (G, i, a) \mid$ το $i-$οστό bit του $\text{bin}(MC(G))$ είναι $a\}$

Για είσοδο (G, i, a):

- binary search με μαντείο του CLIQUE, για να βρούμε μέγεθος s μέγιστης κλίκας: $(G, n/2) \in$ CLIQUE?
    - Ναι. $(G, 3n/4) \in$ CLIQUE?
    - Όχι. $(G,\ n/4) \in$ CLIQUE?
- (s ο τελευταίος αριθμ. με θετική απάντηση, μετά από $\log n$ ερωτήματα)

Δ.ο.: MAXCLIQUE$_b \leq_P^T$ CLIQUE

Υπενθύμιση

MAXCLIQUE$_b$ =
$\{ (G, i, a) \mid$ το i – οστό bit του bin$(MC(G))$ είναι $a\}$

Για είσοδο (G, i, a):

- binary search με μαντείο του CLIQUE, για να βρούμε μέγεθος s μέγιστης κλίκας: $(G, n/2) \in$ CLIQUE?
  - Ναι. $(G, 3n/4) \in$ CLIQUE?
  - Όχι. $(G,\ n/4) \in$ CLIQUE?
- (s ο τελευταίος αριθμ. με θετική απάντηση, μετά από $\log n$ ερωτήματα)
- Αν το i-οστό bit του s είναι a, αποδεχόμαστε, αλλιώς απορρίπτουμε

Δ.ο.: MAXCLIQUE$_b \leq_P^T$ CLIQUE

Υπενθύμιση

MAXCLIQUE$_b$ =
$\{ (G, i, a) \mid$ το i – οστό bit του bin$(MC(G))$ είναι $a\}$

Για είσοδο (G, i, a):

- binary search με μαντείο του CLIQUE, για να βρούμε μέγεθος s μέγιστης κλίκας: $(G, n/2) \in$ CLIQUE?
  - Ναι. $(G, 3n/4) \in$ CLIQUE?
  - Όχι. $(G, n/4) \in$ CLIQUE?
- (s ο τελευταίος αριθμ. με θετική απάντηση, μετά από $\log n$ ερωτήματα)
- Αν το i-οστό bit του s είναι a, αποδεχόμαστε, αλλιώς απορρίπτουμε

Δ.ο.: $\mathsf{MAXCLIQUE}_b \leq_P^T \mathsf{CLIQUE}$

Υπενθύμιση

$\mathsf{MAXCLIQUE}_b =$
$\{\,(G, i, a) \mid$ το $i-$οστό bit του $\mathrm{bin}(MC(G))$ είναι $a\}$

Για είσοδο (G, i, a):

- binary search με μαντείο του CLIQUE, για να βρούμε μέγεθος s μέγιστης κλίκας: $(G, n/2) \in \mathsf{CLIQUE}$?
  - Ναι. $(G, 3n/4) \in \mathsf{CLIQUE}$?
  - Όχι. $(G, n/4) \in \mathsf{CLIQUE}$?
- (s ο τελευταίος αριθμ. με θετική απάντηση, μετά από $\log n$ ερωτήματα)
- Αν το i-οστό bit του s είναι a, αποδεχόμαστε, αλλιώς απορρίπτουμε

Many-one αναγωγή

Ορ.

Ας είναι C κλάση πολ., $A \subseteq \text{STRUC}[\sigma], B \subseteq \text{STRUC}[\tau]$ boolean queries,

Many-one αναγωγή

Ορ.

Ας είναι C κλάση πολ., $A \subseteq \mathrm{STRUC}[\sigma], B \subseteq \mathrm{STRUC}[\tau]$ boolean queries, και $I : \mathrm{STRUC}[\sigma] \to \mathrm{STRUC}[\tau]$ τ.ω.

$$I \in Q(C) \text{ και } (\forall \mathcal{A} \in \mathrm{STRUC}[\sigma])\, \mathcal{A} \in A \Leftrightarrow I(\mathcal{A}) \in B$$

Many-one αναγωγή

Ορ.

Ας είναι C κλάση πολ., $A \subseteq \mathrm{STRUC}[\sigma], B \subseteq \mathrm{STRUC}[\tau]$ boolean queries, και $I : \mathrm{STRUC}[\sigma] \to \mathrm{STRUC}[\tau]$ τ.ω.

$$I \in Q(C) \text{ και } (\forall \mathcal{A} \in \mathrm{STRUC}[\sigma])\ \mathcal{A} \in A \Leftrightarrow I(\mathcal{A}) \in B$$

Το I είναι C – many-one αναγωγή από το A στο B., συμβ $A \leq_C B$

Many-one αναγωγή

Ορ.

Ας είναι C κλάση πολ., $A \subseteq \mathrm{STRUC}[\sigma], B \subseteq \mathrm{STRUC}[\tau]$ boolean queries, και $I : \mathrm{STRUC}[\sigma] \to \mathrm{STRUC}[\tau]$ τ.ω.

$$I \in Q(C) \text{ και } (\forall \mathcal{A} \in \mathrm{STRUC}[\sigma])\, \mathcal{A} \in A \Leftrightarrow I(\mathcal{A}) \in B$$

Το I είναι C – many-one αναγωγή από το A στο B., συμβ $A \leq_C B$

Ενδιαφέρουσες περιπτώσεις:

Many-one αναγωγή

Ορ.

Ας είναι C κλάση πολ., $A \subseteq \mathrm{STRUC}[\sigma], B \subseteq \mathrm{STRUC}[\tau]$ boolean queries, και $I : \mathrm{STRUC}[\sigma] \to \mathrm{STRUC}[\tau]$ τ.ω.

$$I \in Q(C) \text{ και } (\forall \mathcal{A} \in \mathrm{STRUC}[\sigma])\, \mathcal{A} \in A \Leftrightarrow I(\mathcal{A}) \in B$$

Το I είναι C – many-one αναγωγή από το A στο B., συμβ $A \leq_C B$

Ενδιαφέρουσες περιπτώσεις:

- Το I είναι first order query: fo-reduction, $\leq_{fo}$

Many-one αναγωγή

Ορ.

Ας είναι C κλάση πολ., $A \subseteq \text{STRUC}[\sigma], B \subseteq \text{STRUC}[\tau]$ boolean queries, και $I : \text{STRUC}[\sigma] \to \text{STRUC}[\tau]$ τ.ω.

$$I \in Q(C) \text{ και } (\forall \mathcal{A} \in \text{STRUC}[\sigma])\, \mathcal{A} \in A \Leftrightarrow I(\mathcal{A}) \in B$$

Το I είναι C – many-one αναγωγή από το A στο B., συμβ $A \leq_C B$

Ενδιαφέρουσες περιπτώσεις:

- Το I είναι first order query: fo-reduction, $\leq_{fo}$
- Το I είναι στο $Q(L)$: logspace-reductuon, $\leq_{\log}$

Many-one αναγωγή

Ορ.

Ας είναι C κλάση πολ., $A \subseteq \mathrm{STRUC}[\sigma], B \subseteq \mathrm{STRUC}[\tau]$ boolean queries, και $I : \mathrm{STRUC}[\sigma] \rightarrow \mathrm{STRUC}[\tau]$ τ.ω.

$$I \in Q(C) \text{ και } (\forall \mathcal{A} \in \mathrm{STRUC}[\sigma])\ \mathcal{A} \in A \Leftrightarrow I(\mathcal{A}) \in B$$

Το I είναι C – many-one αναγωγή από το A στο B., συμβ $A \leq_C B$

Ενδιαφέρουσες περιπτώσεις:

- Το I είναι first order query: fo-reduction, $\leq_{fo}$
- Το I είναι στο $Q(L)$: logspace-reductuon, $\leq_{\log}$
- Το I είναι στο $Q(\mathrm{P})$: polynomial-time red., $\leq_{\mathrm{P}}$

Παράδειγμα

Βρείτε ένα fo reduction από PARITY σε MULT_b

Παράδειγμα

Βρείτε ένα fo reduction από PARITY σε MULT_b

Defs

PARITY boolean query, αληθές αν η συμβ. έχει περιττό πλήθος μονάδων

Παράδειγμα

Βρείτε ένα fo reduction από PARITY σε MULT_b

Defs

PARITY boolean query, αληθές αν η συμβ. έχει περιττό πλήθος μονάδων

MULT general query, δέχεται ζεύγη boolean strings μήκους n και τα πάει στο γινόμενό τους μήκους $2n$

$$\tau_{ab} = \left\langle A^1, B^1 \right\rangle, \text{MULT} : \text{STRUC}[\tau_{ab}] \to \text{STRUC}[\tau_s]$$

Παράδειγμα

Βρείτε ένα fo reduction από PARITY σε MULT_b

Defs

PARITY boolean query, αληθές αν η συμβ. έχει περιττό πλήθος μονάδων

MULT general query, δέχεται ζεύγη boolean strings μήκους n και τα πάει στο γινόμενό τους μήκους $2n$

$$\tau_{ab} = \langle A^1, B^1 \rangle, \text{MULT} : \text{STRUC}[\tau_{ab}] \rightarrow \text{STRUC}[\tau_s]$$

MULT_b bool. query στο $\tau_{abcd} = \langle A^1, B^1, c, d \rangle$, αληθές ανν το bit c είναι d στο MULT

Παράδειγμα - cont.

Θεωρούμε $I_{PM} : \mathrm{STRUC}[\tau_s] \to \mathrm{STRUC}[\tau_{abcd}]$ με

$$
\begin{aligned}
\varphi_A(x,y) &\equiv y = \max \wedge S(x) \\
\varphi_B(x,y) &\equiv y = \max \\
I_{PM} &\equiv \lambda_{xy} \langle true, \varphi_A, \varphi_B, \langle 0, \max \rangle, \langle 0, 1 \rangle \rangle
\end{aligned}
$$

Παράδειγμα - cont.

Θεωρούμε $I_{PM} : \mathrm{STRUC}[\tau_s] \to \mathrm{STRUC}[\tau_{abcd}]$ με

$$\varphi_A(x,y) \equiv y = \max \wedge S(x)$$
$$\varphi_B(x,y) \equiv y = \max$$
$$I_{PM} \equiv \lambda_{xy} \langle true, \varphi_A, \varphi_B, \langle 0, \max \rangle, \langle 0, 1 \rangle \rangle$$

Ας είναι $w =$ ”01101”. Είδαμε $\mathcal{A}_w = \langle \{0,1,2,3,4\}, \{1,2,4\} \rangle$.

Παράδειγμα - cont.

Θεωρούμε $I_{PM} : \mathrm{STRUC}[\tau_s] \to \mathrm{STRUC}[\tau_{abcd}]$ με

$$\varphi_A(x,y) \equiv y = \max \wedge S(x)$$
$$\varphi_B(x,y) \equiv y = \max$$
$$I_{PM} \equiv \lambda_{xy} \langle true, \varphi_A, \varphi_B, \langle 0, \max \rangle, \langle 0, 1 \rangle \rangle$$

Ας είναι $w =$ "01101". Είδαμε $\mathcal{A}_w = \langle \{0,1,2,3,4\}, \{1,2,4\} \rangle$.
Σαφώς, $PAR(\mathcal{A}) = 1$

Παράδειγμα - cont.

Θεωρούμε $I_{PM} : \mathrm{STRUC}[\tau_s] \to \mathrm{STRUC}[\tau_{abcd}]$ με

$$\begin{aligned}
\varphi_A(x,y) &\equiv y = \max \wedge S(x) \\
\varphi_B(x,y) &\equiv y = \max \\
I_{PM} &\equiv \lambda_{xy} \langle true, \varphi_A, \varphi_B, \langle 0, \max \rangle, \langle 0, 1 \rangle \rangle
\end{aligned}$$

Ας είναι $w =$ "01101". Είδαμε $\mathcal{A}_w = \langle \{0,1,2,3,4\}, \{1,2,4\} \rangle$.
Σαφώς, $PAR(\mathcal{A}) = 1$
Έχουμε $I_{PM}(\mathcal{A}) =$
$\left\langle \{0,1,2,3,4\}^2, \{(1,4),(2,4),(4,4)\}, \{(0,4),\ldots,(4,4)\}, \langle 0,4 \rangle, \langle 0,1 \rangle \right\rangle$

Def.

Για A bool. query, C κλάση πολυπλοκότητας, $\leq_r$ σχέση αναγωγής - αναγωγισιμότητας.

Def.

Για A bool. query, C κλάση πολυπλοκότητας, $\leq_r$ σχέση αναγωγής - αναγωγισιμότητας. Λέμε ότι το A είναι C-πλήρες μέσω $\leq_r$ ανν

Def.

Για A bool. query, C κλάση πολυπλοκότητας, $\leq_r$ σχέση αναγωγής - αναγωγισιμότητας. Λέμε ότι το A είναι C-πλήρες μέσω $\leq_r$ ανν

- $A \in C$

Def.

Για A bool. query, C κλάση πολυπλοκότητας, $\leq_r$ σχέση αναγωγής - αναγωγισιμότητας. Λέμε ότι το A είναι C-πλήρες μέσω $\leq_r$ ανν

- $A \in C$
- $(\forall B \in C)\ B \leq_r A$

Def.

Για A bool. query, C κλάση πολυπλοκότητας, $\leq_r$ σχέση αναγωγής - αναγωγισιμότητας. Λέμε ότι το A είναι C-πλήρες μέσω $\leq_r$ ανν

- $A \in C$
- $(\forall B \in C)\ B \leq_r A$

Def.

Για A bool. query, C κλάση πολυπλοκότητας, $\leq_r$ σχέση αναγωγής - αναγωγισιμότητας. Λέμε ότι το A είναι C-πλήρες μέσω $\leq_r$ ανν

- $A \in C$
- $(\forall B \in C)\ B \leq_r A$

ΠΧ: $U_{time} = \{\ M\#w\#^r \mid\ M(w)\downarrow$ in r steps$\}$

Complete for L:

- CYCLE: Given an undirected graph, does it contain a cycle?
- REACH_d: Given a directed graph, is there a deterministic path from vertex s to vertex t? (A deterministic path is such that for every edge (u, v) on the path, there is only one edge in the graph from u.)

Complete for NL:

- REACH: Given a directed graph, is there a path from vertex s to vertex t?
- 2-SAT: Given a boolean formula in conjunctive normal form with only two literals per clause, is it satisfiable?

Complete for P:

- CIRCUIT-VALUE-PROBLEM (CVP): Given an acyclic boolean circuit, with inputs specified, does its output gate have value one?
- NETWORK-FLOW: Given a directed graph, with capacities on its edges, and a value V, is it possible to achieve a steady-state flow of value V through the graph?

Complete for NP:

- SAT: Given a boolean formula, is it satisfiable?
- 3-SAT: Given a boolean formula in conjunctive normal form with only three literals per clause, is it satisfiable?
- CLIQUE: Given an undirected graph and a value k, does the graph have a complete subgraph with k vertices?

Complete for PSPACE:

- QSAT: Given a quantified boolean formula, is it satisfiable?
- HEX, GEOGRAPHY, GO: Given a position in the generalized versions of the games hex, geography, or go, is there a forced win for the player whose move it is?

Ορ.

Εναλλασσόμενη ΜΤ είναι μια μη-ντετερμινιστική ΜΤ με διαμοιρασμένο σύνολο μη τερματικών καταστάσεων, σε καθολικές και υπαρξιακές.

Ορ.

Εναλλασσόμενη ΜΤ είναι μια μη-ντετερμινιστική ΜΤ με διαμοιρασμένο σύνολο μη τερματικών καταστάσεων, σε καθολικές και υπαρξιακές. Για την αποδοχή:

- κάθε accepting τελικός φάση είναι αποδεκτικός

Ορ.

Εναλλασσόμενη ΜΤ είναι μια μη-ντετερμινιστική ΜΤ με διαμοιρασμένο σύνολο μη τερματικών καταστάσεων, σε καθολικές και υπαρξιακές. Για την αποδοχή:

- κάθε accepting τελικός φάση είναι αποδεκτικός
- μια υπαρξιακή φάση είναι αποδεκτική αν υπάρχει επόμενη αποδεκτική φάση

Ορ.

Εναλλασσόμενη ΜΤ είναι μια μη-ντετερμινιστική ΜΤ με διαμοιρασμένο σύνολο μη τερματικών καταστάσεων, σε καθολικές και υπαρξιακές. Για την αποδοχή:

- κάθε accepting τελικός φάση είναι αποδεκτικός
- μια υπαρξιακή φάση είναι αποδεκτική αν υπάρχει επόμενη αποδεκτική φάση
- μια καθολική φάση είναι αποδεκτική αν όλες οι επόμενες φάσεις είναι αποδεκτικές

Ορ.

Εναλλασσόμενη ΜΤ είναι μια μη-ντετερμινιστική ΜΤ με διαμοιρασμένο σύνολο μη τερματικών καταστάσεων, σε καθολικές και υπαρξιακές. Για την αποδοχή:

- κάθε accepting τελικός φάση είναι αποδεκτικός
- μια υπαρξιακή φάση είναι αποδεκτική αν υπάρχει επόμενη αποδεκτική φάση
- μια καθολική φάση είναι αποδεκτική αν όλες οι επόμενες φάσεις είναι αποδεκτικές

Ορ.

Εναλλασσόμενη ΜΤ είναι μια μη-ντετερμινιστική ΜΤ με διαμοιρασμένο σύνολο μη τερματικών καταστάσεων, σε καθολικές και υπαρξιακές. Για την αποδοχή:

- κάθε accepting τελικός φάση είναι αποδεκτικός
- μια υπαρξιακή φάση είναι αποδεκτική αν υπάρχει επόμενη αποδεκτική φάση
- μια καθολική φάση είναι αποδεκτική αν όλες οι επόμενες φάσεις είναι αποδεκτικές

Η ΜΤ αποδέχεται αν η ρίζα είναι αποδεκτική

Ορ.

Εναλλασσόμενη ΜΤ είναι μια μη-ντετερμινιστική ΜΤ με διαμοιρασμένο σύνολο μη τερματικών καταστάσεων, σε καθολικές και υπαρξιακές. Για την αποδοχή:

- κάθε accepting τελικός φάση είναι αποδεκτικός
- μια υπαρξιακή φάση είναι αποδεκτική αν υπάρχει επόμενη αποδεκτική φάση
- μια καθολική φάση είναι αποδεκτική αν όλες οι επόμενες φάσεις είναι αποδεκτικές

Η ΜΤ αποδέχεται αν η ρίζα είναι αποδεκτική

Εφεξής οι MT έχουν random access read-only input.
$\mathrm{ASPACE}[s(n)] =$
set of bool. queries accepted by Alt TM using $\mathcal{O}(s(n))$ cells
$\mathrm{ATIME}[s(n)] =$
set of bool. queries accepted by Alt TM using $\mathcal{O}(s(n))$ steps

Θεωρήματα

Για $s(n) \geq \log n, t(n) \geq n$,

$$\cup_{k=1}^{\infty} \text{ATIME}[t^k(n)] = \cup_{k=1}^{\infty} \text{DSPACE}[t^k(n)]$$

$$\text{ASPACE}[s(n)] = \cup_{k=1}^{\infty} \text{DTIME}[k^{s(n)}]$$

Επιπλέον, $\text{ASPACE}[\log n] = \text{P}, \text{AP} = \text{PSPACE}$

Boolean circuits

boolean circuit = DAG,

$$C = (V, E, G_{\wedge}, G_{\vee}, G_{\neg}, I, r) \in \mathrm{STRUC}[\tau_c]$$

Boolean circuits

boolean circuit = DAG,

$$C = (V, E, G_{\wedge}, G_{\vee}, G_{\neg}, I, r) \in \mathrm{STRUC}[\tau_c]$$

CVP (Circuit Value Problem): αν κύκλωμα με root $r = 1$

Boolean circuits

boolean circuit = DAG,

$$C = (V, E, G_\wedge, G_\vee, G_\neg, I, r) \in \mathrm{STRUC}[\tau_c]$$

CVP (Circuit Value Problem): αν κύκλωμα με root $r = 1$
MCVP (Monotone CVP): υποσύνολο CVP χωρίς negation gates

Boolean circuits

boolean circuit = DAG,

$$C = (V, E, G_\wedge, G_\vee, G_\neg, I, r) \in \mathrm{STRUC}[\tau_c]$$

CVP (Circuit Value Problem): αν κύκλωμα με root $r = 1$

MCVP (Monotone CVP): υποσύνολο CVP χωρίς negation gates

CVP αποφάνσιμο σε γραμμικό χρόνο

CVP αποφάνσιμο σε γραμμικό χρόνο
MCVP αποφάνισμο σε ASPACE[$\log n$]. Αλγόριθμος EVAL(a):

- Αν $I(a)$, accept

CVP αποφάνσιμο σε γραμμικό χρόνο
MCVP αποφάνισμο σε ASPACE[$\log n$]. Αλγόριθμος EVAL(a):

- Αν $I(a)$, accept
- αλλιώς αν a όχι outgoing ακμές, reject

CVP αποφάνσιμο σε γραμμικό χρόνο
MCVP αποφάνισμο σε ASPACE[$\log n$]. Αλγόριθμος EVAL(a):

- Αν $I(a)$, accept
- αλλιώς αν a όχι outgoing ακμές, reject
- Αν $G_\wedge(a)$ τότε επέλεξε καθολικά απόγονο b του a

CVP αποφάνσιμο σε γραμμικό χρόνο
MCVP αποφάνισμο σε ASPACE$[\log n]$. Αλγόριθμος EVAL(a):

- Αν $I(a)$, accept
- αλλιώς αν a όχι outgoing ακμές, reject
- Αν $G_\wedge(a)$ τότε επέλεξε καθολικά απόγονο b του a
- αλλιώς επέλεξε ύπαρξιακά απόγονο b του a

CVP αποφάνσιμο σε γραμμικό χρόνο
MCVP αποφάνισμο σε ASPACE[$\log n$]. Αλγόριθμος EVAL(a):

- Αν $I(a)$, accept
- αλλιώς αν a όχι outgoing ακμές, reject
- Αν $G_\wedge(a)$ τότε επέλεξε καθολικά απόγονο b του a
- αλλιώς επέλεξε ύπαρξιακά απόγονο b του a
- Επέστρεψε EVAL(b)

CVP αποφάνσιμο σε γραμμικό χρόνο
MCVP αποφάνισμο σε ASPACE$[\log n]$. Αλγόριθμος EVAL(a):

- Αν $I(a)$, accept
- αλλιώς αν a όχι outgoing ακμές, reject
- Αν $G_{\wedge}(a)$ τότε επέλεξε καθολικά απόγονο b του a
- αλλιώς επέλεξε ύπαρξιακά απόγονο b του a
- Επέστρεφε EVAL(b)

CVP αποφάνσιμο σε γραμμικό χρόνο
MCVP αποφάνισμο σε ASPACE[$\log n$]. Αλγόριθμος EVAL(a):

- Αν $I(a)$, accept
- αλλιώς αν a όχι outgoing ακμές, reject
- Αν $G_\wedge(a)$ τότε επέλεξε καθολικά απόγονο b του a
- αλλιώς επέλεξε ύπαρξιακά απόγονο b του a
- Επέστρεψε EVAL(b)

Τρέξε EVAL(r)

Quantified SAT

QSAT = set of true formulas of the following form

$$\psi = (Q_1 x_1)(Q_2 x_2) \cdots (Q_r x_r)\varphi$$

όπου φ τύπος πάνω στις $x_1, \ldots, x_r$ και Q_i είναι $\forall$ ή $\exists$

Quantified SAT

QSAT = set of true formulas of the following form

$$\psi = (Q_1 x_1)(Q_2 x_2)\cdots(Q_r x_r)\varphi$$

όπου φ τύπος πάνω στις $x_1, \ldots, x_r$ και Q_i είναι $\forall$ ή $\exists$

Παρατήρηστε για κάθε τύπο φ πάνω στις μεταβλητές $\bar{x}$:

$$\varphi \in \mathrm{SAT} \Leftrightarrow (\exists \bar{x})\varphi \in \mathrm{QSAT}$$

Quantified SAT

QSAT = set of true formulas of the following form

$$\psi = (Q_1 x_1)(Q_2 x_2) \cdots (Q_r x_r)\varphi$$

όπου φ τύπος πάνω στις $x_1, \ldots, x_r$ και Q_i είναι $\forall$ ή $\exists$

Παρατήρηστε για κάθε τύπο φ πάνω στις μεταβλητές $\bar{x}$:

$$\varphi \in \mathrm{SAT} \Leftrightarrow (\exists \bar{x})\varphi \in \mathrm{QSAT}$$

$$\varphi \notin \mathrm{SAT} \Leftrightarrow (\forall \bar{x})\neg\varphi \in \mathrm{QSAT}$$

Quantified SAT

QSAT = set of true formulas of the following form

$$\psi = (Q_1 x_1)(Q_2 x_2)\cdots(Q_r x_r)\varphi$$

όπου φ τύπος πάνω στις $x_1, \ldots, x_r$ και Q_i είναι $\forall$ ή $\exists$

Παρατήρηστε για κάθε τύπο φ πάνω στις μεταβλητές $\bar{x}$:

$$\varphi \in \mathrm{SAT} \Leftrightarrow (\exists \bar{x})\varphi \in \mathrm{QSAT}$$

$$\varphi \notin \mathrm{SAT} \Leftrightarrow (\forall \bar{x})\neg\varphi \in \mathrm{QSAT}$$

Θεωρ. 2.32 σελ 38

Για $s(n) \geq \log n$ space constr.,

$$\text{NSPACE}[s(n)] \subseteq \text{ATIME}[s^2(n)] \subseteq \text{DSPACE}[s^2(n)]$$

$$\text{ASPACE-TIME}[s(n), t(n)]$$

$$\mathrm{ASPACE\text{-}TIME}[s(n), t(n)]$$

$$\mathrm{ATIME\text{-}ALT}[t(n), a(n)]$$

$$\text{ASPACE-TIME}[s(n), t(n)]$$

$$\text{ATIME-ALT}[t(n), a(n)]$$

$$\text{NP} = \text{ATIME-ALT}[n^{O(1)}, 1]$$

$$\text{ASPACE-TIME}[s(n), t(n)]$$

$$\text{ATIME-ALT}[t(n), a(n)]$$

$$\text{NP} = \text{ATIME-ALT}[n^{O(1)}, 1]$$

$$\text{PH} = \cup_{k=1}^{\infty} \text{ATIME-ALT}[n^k, 1]$$