First Order Reductions

Κίτσιου Χρυσάνθη Μελίτα,
A.M 7115142100010

Περιγραφική Πολυπλοκότητα
ΑΛΜΑ

Δευτέρα, 13/03/2023

Περιεχόμενα

Περιεχόμενα Παρουσίασης

Κλάσεις Πολυπλοκότητας

- Η κλάση L αποτελείται από τα προβλήματα απόφασης τα οποία μπορούν να επιλυθούν από μια ντετερμινιστική TM σε λογαριθμικό χώρο, δηλ $L = SPACE(\log n)$.

Κλάσεις Πολυπλοκότητας

- Η κλάση L αποτελείται από τα προβλήματα απόφασης τα οποία μπορούν να επιλυθούν από μια ντετερμινιστική TM σε λογαριθμικό χώρο, δηλ $L = SPACE(\log n)$.
- Η κλάση NL αποτελείται από τα προβλήματα απόφασης τα οποία μπορούν να επιλυθούν από μια μη ντετερμινιστική TM σε λογαριθμικό χώρο, δηλ $NL = NSPACE(\log n)$.

Κλάσεις Πολυπλοκότητας

- Η κλάση L αποτελείται από τα προβλήματα απόφασης τα οποία μπορούν να επιλυθούν από μια ντετερμινιστική TM σε λογαριθμικό χώρο, δηλ $L = SPACE(\log n)$.
- Η κλάση NL αποτελείται από τα προβλήματα απόφασης τα οποία μπορούν να επιλυθούν από μια μη ντετερμινιστική TM σε λογαριθμικό χώρο, δηλ $NL = NSPACE(\log n)$.
- Η κλάση P αποτελείται από τα προβλήματα απόφασης τα οποία επιλύονται σε πολυωνυμικό χρόνο από μια ντετερμονιστική TM, δηλ $P = \bigcup_k TIME(n^k)$.

Κλάσεις Πολυπλοκότητας

- Η κλάση L αποτελείται από τα προβλήματα απόφασης τα οποία μπορούν να επιλυθούν από μια ντετερμινιστική TM σε λογαριθμικό χώρο, δηλ $L = SPACE(\log n)$.
- Η κλάση NL αποτελείται από τα προβλήματα απόφασης τα οποία μπορούν να επιλυθούν από μια μη ντετερμινιστική TM σε λογαριθμικό χώρο, δηλ $NL = NSPACE(\log n)$.
- Η κλάση P αποτελείται από τα προβλήματα απόφασης τα οποία επιλύονται σε πολυωνυμικό χρόνο από μια ντετερμονιστική TM, δηλ $P = \bigcup_k TIME(n^k)$.

Ισχύει $L \subseteq NL \subseteq P$.

Logspace Turing Machine

Μια TM λογαριθμικού χώρου αποτελείται από:

Logspace Turing Machine

Μια TM λογαριθμικού χώρου αποτελείται από:

- μια αναγνώσιμη ταινία εισόδου,

Logspace Turing Machine

Μια TM λογαριθμικού χώρου αποτελείται από:

- μια αναγνώσιμη ταινία εισόδου,
- μια εγγράψιμη ταινία εξόδου,

Logspace Turing Machine

Μια TM λογαριθμικού χώρου αποτελείται από:

- μια αναγνώσιμη ταινία εισόδου,
- μια εγγράψιμη ταινία εξόδου,
- μια αναγνώσιμη κι εγγράψιμη ταινία εργασίας που περιέχει $O(\log n)$ bits.

Logspace Turing Machine

Μια TM λογαριθμικού χώρου αποτελείται από:

- μια αναγνώσιμη ταινία εισόδου,
- μια εγγράψιμη ταινία εξόδου,
- μια αναγνώσιμη κι εγγράψιμη ταινία εργασίας που περιέχει $O(\log n)$ bits.

Ειδικότερα, η TM στην ταινία εργασίας μπορεί να

Logspace Turing Machine

Μια TM λογαριθμικού χώρου αποτελείται από:

- μια αναγνώσιμη ταινία εισόδου,
- μια εγγράψιμη ταινία εξόδου,
- μια αναγνώσιμη κι εγγράψιμη ταινία εργασίας που περιέχει $O(\log n)$ bits.

Ειδικότερα, η TM στην ταινία εργασίας μπορεί να

- αποθηκεύσει έναν αριθμό με $\log n$- bits

Θεώρημα: FO $\subseteq$ L

Το σύνολο των boolean queries που μπορούν να περιγραφούν σε λογική πρώτης τάξης μπορεί να υπολογιστεί ντετερμινιστικά σε λογαριθμικό χώρο.

Θεώρημα: FO $\subseteq$ L

Το σύνολο των boolean queries που μπορούν να περιγραφούν σε λογική πρώτης τάξης μπορεί να υπολογιστεί ντετερμινιστικά σε λογαριθμικό χώρο.

Απόδειξη

Έστω voc $\sigma = < R_1^{a_1}, ..., R_r^{a_r}, c_1, ..., c_s >$.

Θεώρημα: FO ⊆ L

Το σύνολο των boolean queries που μπορούν να περιγραφούν σε λογική πρώτης τάξης μπορεί να υπολογιστεί ντετερμινιστικά σε λογαριθμικό χώρο.

Απόδειξη

Έστω voc $\sigma =< R_1^{a_1}, ..., R_r^{a_r}, c_1, ..., c_s >$.
Έστω A πεπερ. δομή, όπου $A \in STRUC[\sigma]$.

Θεώρημα: FO $\subseteq$ L

Το σύνολο των boolean queries που μπορούν να περιγραφούν σε λογική πρώτης τάξης μπορεί να υπολογιστεί ντετερμινιστικά σε λογαριθμικό χώρο.

Απόδειξη

Έστω voc $\sigma =< R_1^{a_1}, ..., R_r^{a_r}, c_1, ..., c_s >$.
Έστω A πεπερ. δομή, όπου $A \in STRUC[\sigma]$.
Ένα boolean FO query αποφασίζεται από πρωτοτάξια πρόταση $\phi \in L(\sigma)$, δηλ $I_\phi(A) = 1 \Leftrightarrow A \models \phi$.

Θεώρημα: FO $\subseteq$ L

Το σύνολο των boolean queries που μπορούν να περιγραφούν σε λογική πρώτης τάξης μπορεί να υπολογιστεί ντετερμινιστικά σε λογαριθμικό χώρο.

Απόδειξη

Έστω voc $\sigma =< R_1^{a_1}, ..., R_r^{a_r}, c_1, ..., c_s >$.
Έστω A πεπερ. δομή, όπου $A \in STRUC[\sigma]$.
Ένα boolean FO query αποφασίζεται από πρωτοτάξια πρόταση $\phi \in L(\sigma)$, ***δηλ*** $I_\phi(A) = 1 \Leftrightarrow A \models \phi$.
Κατασκευάζουμε ντετερμινιστική logspace TM M τ.ω

$$A \models \phi \Leftrightarrow M(bin(A)) \downarrow$$

Απόδειξη

Θέλουμε να υπολογίσουμε το μέγεθος του σύμπαντος $|A|$.

Απόδειξη

Θέλουμε να υπολογίσουμε το μέγεθος του σύμπαντος $|A|$.
Η είσοδος της M θα είναι της μορφής $bin(A)$, όπου

$$bin(A) = bin^A(R_1)...bin^A(R_r)bin^A(c_1)...bin^A(c_s).$$

Απόδειξη

Θέλουμε να υπολογίσουμε το μέγεθος του σύμπαντος $|A|$.
Η είσοδος της M θα είναι της μορφής $bin(A)$, όπου

$$bin(A) = bin^A(R_1)...bin^A(R_r)bin^A(c_1)...bin^A(c_s).$$

Άρα, το μήκος της εισόδου θα είναι

$$f(n) = n^{a_1} + ... + n^{a_r} + s\lceil \log n \rceil, \text{για κάποιο } n$$

Απόδειξη

Θέλουμε να υπολογίσουμε το μέγεθος του σύμπαντος $|A|$.
Η είσοδος της M θα είναι της μορφής $bin(A)$, όπου

$$bin(A) = bin^A(R_1)...bin^A(R_r)bin^A(c_1)...bin^A(c_s).$$

Άρα, το μήκος της εισόδου θα είναι

$$f(n) = n^{a_1} + ... + n^{a_r} + s\lceil \log n \rceil, \text{για κάποιο } n$$

Η M υπολογίζει τα $f(1), f(2), ...$ μέχρι να βρει ένα $f(j)$ που να ισούται με το μήκος εισόδου της M κι άρα $n = j$.

Απόδειξη

Θέλουμε να υπολογίσουμε το μέγεθος του σύμπαντος $|A|$.
Η είσοδος της M θα είναι της μορφής $bin(A)$, όπου

$$bin(A) = bin^A(R_1)...bin^A(R_r)bin^A(c_1)...bin^A(c_s).$$

Άρα, το μήκος της εισόδου θα είναι

$$f(n) = n^{a_1} + ... + n^{a_r} + s\lceil \log n \rceil, \text{για κάποιο } n$$

Η M υπολογίζει τα $f(1), f(2), \ldots$ μέχρι να βρει ένα $f(j)$ που να ισούται με το μήκος εισόδου της M κι άρα $n = j$.
Το $\lceil \log n \rceil$ αποτελεί το μήκος της αναπαράστασης του $(j)_2$.

Απόδειξη

Υποθέτουμε ότι η ϕ είναι της μορφής:

$$\phi \equiv (\exists \boldsymbol{x}_1)(\forall \boldsymbol{x}_2)...(\boldsymbol{Q}_k \boldsymbol{x}_k)\boldsymbol{a}(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_k)$$

όπου Q_i ποσοδείκτες και το a δεν περιέχει ποσοδείκτες.

Απόδειξη

Υποθέτουμε ότι η ϕ είναι της μορφής:

$$\phi \equiv (\exists \boldsymbol{x}_1)(\forall \boldsymbol{x}_2)...(\boldsymbol{Q}_k \boldsymbol{x}_k)\boldsymbol{a}(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_k)$$

όπου Q_i ποσοδείκτες και το a δεν περιέχει ποσοδείκτες. Κατασκευάζουμε την M με επαγωγή στο k.

Απόδειξη

Υποθέτουμε ότι η ϕ είναι της μορφής:

$$\phi \equiv (\exists x_1)(\forall x_2)...(Q_k x_k) a(x_1, x_2, ..., x_k)$$

όπου Q_i ποσοδείκτες και το a δεν περιέχει ποσοδείκτες.
Κατασκευάζουμε την M με επαγωγή στο k.
ΒΑΣΗ: Για $k = 0$ η ϕ δεν περιέχει ποσοδείκτες.

Απόδειξη

Υποθέτουμε ότι η ϕ είναι της μορφής:

$$\phi \equiv (\exists x_1)(\forall x_2)...(Q_k x_k)a(x_1, x_2, ..., x_k)$$

όπου Q_i ποσοδείκτες και το a δεν περιέχει ποσοδείκτες.
Κατασκευάζουμε την M με επαγωγή στο k.
ΒΑΣΗ: Για $k = 0$ η ϕ δεν περιέχει ποσοδείκτες.
Άρα, είναι ένας συνδιασμός από boolean ατομικούς τύπους, οι οποίοι έχουν τη μορφή:

Απόδειξη

Υποθέτουμε ότι η ϕ είναι της μορφής:

$$\phi \equiv (\exists x_1)(\forall x_2)...(Q_k x_k) a(x_1, x_2, ..., x_k)$$

όπου Q_i ποσοδείκτες και το a δεν περιέχει ποσοδείκτες.
Κατασκευάζουμε την M με επαγωγή στο k.
ΒΑΣΗ: Για $k = 0$ η ϕ δεν περιέχει ποσοδείκτες.
Άρα, είναι ένας συνδιασμός από boolean ατομικούς τύπους, οι οποίοι έχουν τη μορφή:

- $R_i(t_1, ..., t_{a_i})$

Απόδειξη

Υποθέτουμε ότι η ϕ είναι της μορφής:

$$\phi \equiv (\exists x_1)(\forall x_2)...(Q_k x_k)a(x_1, x_2, ..., x_k)$$

όπου Q_i ποσοδείκτες και το a δεν περιέχει ποσοδείκτες.
Κατασκευάζουμε την M με επαγωγή στο k.
ΒΑΣΗ: Για $k = 0$ η ϕ δεν περιέχει ποσοδείκτες.
Άρα, είναι ένας συνδιασμός από boolean ατομικούς τύπους, οι οποίοι έχουν τη μορφή:

- $R_i(t_1, ..., t_{a_i})$
- $t_1 = t_2$

Απόδειξη

Υποθέτουμε ότι η ϕ είναι της μορφής:

$$\phi \equiv (\exists x_1)(\forall x_2)...(Q_k x_k) a(x_1, x_2, ..., x_k)$$

όπου Q_i ποσοδείκτες και το a δεν περιέχει ποσοδείκτες.
Κατασκευάζουμε την M με επαγωγή στο k.
ΒΑΣΗ: Για $k = 0$ η ϕ δεν περιέχει ποσοδείκτες.
Άρα, είναι ένας συνδιασμός από boolean ατομικούς τύπους, οι οποίοι έχουν τη μορφή:

- $R_i(t_1, ..., t_{a_i})$
- $t_1 = t_2$
- $t_i \leq t_2$

Απόδειξη

Υποθέτουμε ότι η ϕ είναι της μορφής:

$$\phi \equiv (\exists x_1)(\forall x_2)...(Q_k x_k) a(x_1, x_2, ..., x_k)$$

όπου Q_i ποσοδείκτες και το a δεν περιέχει ποσοδείκτες.
Κατασκευάζουμε την M με επαγωγή στο k.
ΒΑΣΗ: Για $k = 0$ η ϕ δεν περιέχει ποσοδείκτες.
Άρα, είναι ένας συνδιασμός από boolean ατομικούς τύπους, οι οποίοι έχουν τη μορφή:

- $R_i(t_1, ..., t_{a_i})$
- $t_1 = t_2$
- $t_i \leq t_2$
- $BIT(t_1, t_2)$

όπου $t_i \in \{c_1, ..., c_s, 0, 1, max\}$.

Απόδειξη

Η M εξετάζει αν $A \models \phi$ κοιτώντας την είσοδο.

Απόδειξη

Η M εξετάζει αν $A \models \phi$ κοιτώντας την είσοδο.
Πχ, έστω ότι θέλουμε να δούμε αν αληθεύει η φόρμουλα $R_3(c_2, max, c_1)$.

Απόδειξη

Η M εξετάζει αν $A \models \phi$ κοιτώντας την είσοδο.
Πχ, έστω ότι θέλουμε να δούμε αν αληθεύει η φόρμουλα $R_3(c_2, max, c_1)$.

- Η M γράφει στην ταινία εργασίας τις τιμές c_2, max, c_1, όπου $max = n - 1$.

Απόδειξη

Η Μ εξετάζει αν $A \models \phi$ κοιτώντας την είσοδο.
Πχ, έστω ότι θέλουμε να δούμε αν αληθεύει η φόρμουλα $R_3(c_2, max, c_1)$.

- Η Μ γράφει στην ταινία εργασίας τις τιμές c_2, max, c_1, όπου $max = n - 1$.
- Η read only κεφαλή εισόδου μετακινείται δεξιά κατά $n^{a_1} + n^{a_2} + 1$ θέσεις,

Απόδειξη

Η M εξετάζει αν $A \models \phi$ κοιτώντας την είσοδο.
Πχ, έστω ότι θέλουμε να δούμε αν αληθεύει η φόρμουλα $R_3(c_2, max, c_1)$.

- Η M γράφει στην ταινία εργασίας τις τιμές c_2, max, c_1, όπου $max = n - 1$.
- Η read only κεφαλή εισόδου μετακινείται δεξιά κατά $n^{a_1} + n^{a_2} + 1$ θέσεις, $n^{a_1} + n^{a_2}$ για να προσπεράσει τις κωδικ. των R_1 και R_2 και μια θέση για να αρχίσει η κωδικ. της R_3.

Απόδειξη

Η Μ εξετάζει αν $A \models \phi$ κοιτώντας την είσοδο.
Πχ, έστω ότι θέλουμε να δούμε αν αληθεύει η φόρμουλα $R_3(c_2, max, c_1)$.

- Η Μ γράφει στην ταινία εργασίας τις τιμές c_2, max, c_1, όπου $max = n - 1$.
- Η read only κεφαλή εισόδου μετακινείται δεξιά κατά $n^{a_1} + n^{a_2} + 1$ θέσεις, $n^{a_1} + n^{a_2}$ για να προσπεράσει τις κωδικ. των R_1 και R_2 και μια θέση για να αρχίσει η κωδικ. της R_3.
- Έπειτα, η κεφαλή μετακινείται δεξιά κατά $n^2 c_2 + n(n-1) + c_1$ θέσεις.

Απόδειξη

Η M εξετάζει αν $A \models \phi$ κοιτώντας την είσοδο.
Πχ, έστω ότι θέλουμε να δούμε αν αληθεύει η φόρμουλα $R_3(c_2, max, c_1)$.

- Η M γράφει στην ταινία εργασίας τις τιμές c_2, max, c_1, όπου max$= n-1$.
- Η read only κεφαλή εισόδου μετακινείται δεξιά κατά $n^{a_1} + n^{a_2} + 1$ θέσεις, $n^{a_1} + n^{a_2}$ για να προσπεράσει τις κωδικ. των R_1 και R_2 και μια θέση για να αρχίσει η κωδικ. της R_3.
- Έπειτα, η κεφαλή μετακινείται δεξιά κατά $n^2 c_2 + n(n-1) + c_1$ θέσεις.
- Σε κάθε tuple αντιστοιχεί ένα bit.

Απόδειξη

Η M εξετάζει αν $A \models \phi$ κοιτώντας την είσοδο.
Πχ, έστω ότι θέλουμε να δούμε αν αληθεύει η φόρμουλα $R_3(c_2, max, c_1)$.

- Η M γράφει στην ταινία εργασίας τις τιμές c_2, max, c_1, όπου $max = n - 1$.
- Η read only κεφαλή εισόδου μετακινείται δεξιά κατά $n^{a_1} + n^{a_2} + 1$ θέσεις, $n^{a_1} + n^{a_2}$ για να προσπεράσει τις κωδικ. των R_1 και R_2 και μια θέση για να αρχίσει η κωδικ. της R_3.
- Έπειτα, η κεφαλή μετακινείται δεξιά κατά $n^2 c_2 + n(n-1) + c_1$ θέσεις.
- Σε κάθε tuple αντιστοιχεί ένα bit.
  Το παραπάνω bit είναι 1 ανν $A \models R_3(c_2, max, c_1)$.

Απόδειξη

ΕΠΑΓΩΓΙΚΗ ΥΠΟΘΕΣΗ : Έστω ότι έχει κατασκευαστεί logspace TM M για κάθε πρόταση ϕ με $k-1$ ποσοδείκτες.

Απόδειξη

ΕΠΑΓΩΓΙΚΗ ΥΠΟΘΕΣΗ : Έστω ότι έχει κατασκευαστεί logspace TM M για κάθε πρόταση ϕ με $k-1$ ποσοδείκτες.
ΕΠΑΓΩΓΙΚΟ ΒΗΜΑ : Θέλουμε να κατασκευάσουμε logspace TM για προτάσεις με k ποσοδείκτες.

Απόδειξη

ΕΠΑΓΩΓΙΚΗ ΥΠΟΘΕΣΗ : Έστω ότι έχει κατασκευαστεί logspace TM M για κάθε πρόταση ϕ με $k-1$ ποσοδείκτες.
ΕΠΑΓΩΓΙΚΟ ΒΗΜΑ : Θέλουμε να κατασκευάσουμε logspace TM για προτάσεις με k ποσοδείκτες.
Θέτουμε $\psi(x_1) \equiv (\forall x_2)...(Q_k x_k) a(x_1, ..., x_k)$.

Απόδειξη

ΕΠΑΓΩΓΙΚΗ ΥΠΟΘΕΣΗ : Έστω ότι έχει κατασκευαστεί logspace TM M για κάθε πρόταση ϕ με $k-1$ ποσοδείκτες.
ΕΠΑΓΩΓΙΚΟ ΒΗΜΑ : Θέλουμε να κατασκευάσουμε logspace TM για προτάσεις με k ποσοδείκτες.
Θέτουμε $\psi(x_1) \equiv (\forall x_2)...(Q_k x_k)a(x_1, ..., x_k)$.
Άρα, $\phi \equiv (\exists x_1)\psi(x_1)$.

Απόδειξη

Κατασκευάζουμε την TM M ως εξής:

Απόδειξη

Κατασκευάζουμε την TM M ως εξής:

- *Εξετάζουμε κάθε πιθανή τιμή της x_1, βάζοντας στη σταθερά c την αντίστοιχη τιμή.*

Απόδειξη

Κατασκευάζουμε την TM M ως εξής:

- *Εξετάζουμε κάθε πιθανή τιμή της x_1, βάζοντας στη σταθερά c την αντίστοιχη τιμή.*
  Η σταθερά καταλαμβάνει χώρο $\log n$ bits.

Απόδειξη

Κατασκευάζουμε την TM M ως εξής:

- Εξετάζουμε κάθε πιθανή τιμή της x_1, βάζοντας στη σταθερά c την αντίστοιχη τιμή.
  Η σταθερά καταλαμβάνει χώρο $\log n$ bits.
- Η πρόταση $\psi(c)$ είναι εκφρασμένη σε λογική πρώτης τάξης και περιέχει ακριβώς $k-1$ ποσοδείκτες.

Απόδειξη

Κατασκευάζουμε την TM M ως εξής:

- Εξετάζουμε κάθε πιθανή τιμή της x_1, βάζοντας στη σταθερά c την αντίστοιχη τιμή.
  Η σταθερά καταλαμβάνει χώρο $\log n$ bits.
- Η πρόταση $\psi(c)$ είναι εκφρασμένη σε λογική πρώτης τάξης και περιέχει ακριβώς $k-1$ ποσοδείκτες.
  Από Υ.Ε, υπάρχει TM M' για την $\psi(c)$.

Απόδειξη

Κατασκευάζουμε την TM M ως εξής:

- Εξετάζουμε κάθε πιθανή τιμή της x_1, βάζοντας στη σταθερά c την αντίστοιχη τιμή.
  Η σταθερά καταλαμβάνει χώρο $\log n$ bits.
- Η πρόταση $\psi(c)$ είναι εκφρασμένη σε λογική πρώτης τάξης και περιέχει ακριβώς $k-1$ ποσοδείκτες.
  Από Υ.Ε, υπάρχει TM M' για την $\psi(c)$.
- Τρέχουμε την M' για όλες τις τιμές που μπορεί να πάρει η c. Αν αποδεχτεί την είσοδο τουλάχιστον μια φορά, τότε υπάρχει x_1 τέτοιο ώστε να ισχύει η $\psi(x_1)$.

Επομένως, η ζητούμενη logspace TM M είναι κατασκευάσιμη.

Σχετικά με την απόδειξη:

Σχετικά με την απόδειξη:

Παρατήρηση

Αν η πρόταση ϕ ήταν της μορφής:

$$\phi \equiv (\forall \boldsymbol{x}_1)(\exists \boldsymbol{x}_2)...(\boldsymbol{Q_k x_k})\boldsymbol{a}(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x_k})$$

Σχετικά με την απόδειξη:

Παρατήρηση

Αν η πρόταση ϕ ήταν της μορφής:

$$\phi \equiv (\forall \boldsymbol{x}_1)(\exists \boldsymbol{x}_2)...(\boldsymbol{Q}_{\boldsymbol{k}} \boldsymbol{x}_{\boldsymbol{k}}) \boldsymbol{a}(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{\boldsymbol{k}})$$

η απόδειξη αλλάζει μόνο στο τελευταίο βήμα!

Σχετικά με την απόδειξη:

Παρατήρηση

Αν η πρόταση ϕ ήταν της μορφής:

$$\phi \equiv (\forall x_1)(\exists x_2)...(Q_k x_k) a(x_1, x_2, ..., x_k)$$

η απόδειξη αλλάζει μόνο στο τελευταίο βήμα!

- Τρέχουμε την M' για όλες τις δυνατές τιμές που μπορεί να πάρει η c.
  Αν αποδεχτεί όλες τις εισόδους, θα έχουμε ότι για κάθε x_1 ικανοποιείται η $\psi(x_1)$.

Συμπέρασμα

- **Από Ιεραρχία των κλάσεων, $L \subseteq NL \subseteq P$.**

Συμπέρασμα

- Από Ιεραρχία των κλάσεων, $L \subseteq NL \subseteq P$.
- Από Θεώρημα, $FO \subseteq L$.

Συμπέρασμα

- Από Ιεραρχία των κλάσεων, $L \subseteq NL \subseteq P$.
- Από Θεώρημα, $FO \subseteq L$.

Επομένως, για να περιγράψουμε προβλήματα της κλάσης P, NP κτλ, ΔΕΝ αρκεί η λογική πρώτης τάξης!

Περιεχόμενα Παρουσίασης

Σημαντικά σημεία ενότητας:

- ***Η κλάση πολυπλοκότητας των boolean queries που μπορούν να εκφραστούν με λογική πρώτης τάξης είναι "χαμηλά" στην ιεραρχία.***

Σημαντικά σημεία ενότητας:

- ***Η κλάση πολυπλοκότητας των boolean queries που μπορούν να εκφραστούν με λογική πρώτης τάξης είναι "χαμηλά" στην ιεραρχία.***
- ***$A \leq_{FO} B$, όπου A, B boolean queries σημαίνει ότι το A δεν είναι "πιο δύσκολο" από το B.***

Σημαντικά σημεία ενότητας:

- *Η κλάση πολυπλοκότητας των boolean queries που μπορούν να εκφραστούν με λογική πρώτης τάξης είναι "χαμηλά" στην ιεραρχία.*
- *$A \leq_{FO} B$, όπου A, B boolean queries σημαίνει ότι το A δεν είναι "πιο δύσκολο" από το B.*

Κλειστότητα FO- Reductions

Ένα σύνολο boolean queries S λέμε ότι είναι κλειστό ως προς FO- reduction ανν όταν υπάρχουν boolean queries A,B τ.ω $B \in S$ και $A \leq_{FO} B$, τότε έχουμε ότι $A \in S$.

Πρόταση

Έστω S σύνολο boolean queries το οποίο είναι κλειστό ως προς logspace- reductions. Τότε, είναι κλειστό και ως προς FO - reductions.

Πρόταση

Έστω S σύνολο boolean queries το οποίο είναι κλειστό ως προς logspace- reductions. Τότε, είναι κλειστό και ως προς FO - reductions.

- Σχεδόν όλες οι κλάσεις πολυπλοκότητας C (εκτός από δυναμικές) και οι γλώσσες L που θα μελετήσουμε είναι κλειστές ως προς FO- reductions.

Πρόταση

Έστω S σύνολο boolean queries το οποίο είναι κλειστό ως προς logspace- reductions. Τότε, είναι κλειστό και ως προς FO - reductions.

- Σχεδόν όλες οι κλάσεις πολυπλοκότητας C (εκτός από δυναμικές) και οι γλώσσες L που θα μελετήσουμε είναι κλειστές ως προς FO- reductions.

Μεθοδολογία

Ας είναι L μια γλώσσα (set boolean queries που εκφράζονται μέσω της L) και C μια κλαση πολυπλοκότητας (set boolean queries υπολογίσιμα στην C).

Μεθοδολογία

Ας είναι L μια γλώσσα (set boolean queries που εκφράζονται μέσω της L) και C μια κλαση πολυπλοκότητας (set boolean queries υπολογίσιμα στην C).
Για νδο ένα query ανήκει στην C ανν εκφράζεται μέσω της L, ακολουθούμε την εξής διαδικασία:

Μεθοδολογία

Ας είναι L μια γλώσσα (set boolean queries που εκφράζονται μέσω της L) και C μια κλαση πολυπλοκότητας (set boolean queries υπολογίσιμα στην C).
Για νδο ένα query ανήκει στην C ανν εκφράζεται μέσω της L, ακολουθούμε την εξής διαδικασία:

Μέθοδος

1. Δημιουργία C− αλγορίθμου που εξετάζει για κάθε ϕ, A της L, αν $A \models \phi$.

Μεθοδολογία

Ας είναι L μια γλώσσα (set boolean queries που εκφράζονται μέσω της L) και C μια κλαση πολυπλοκότητας (set boolean queries υπολογίσιμα στην C).
Για νδο ένα query ανήκει στην C ανν εκφράζεται μέσω της L, ακολουθούμε την εξής διαδικασία:

Μέθοδος

1. Δημιουργία C– αλγορίθμου που εξετάζει για κάθε ϕ, A της L, αν $A \models \phi$.
2. Βρίσκουμε ένα boolean query T που είναι complete στην C ως προς FO- reductions.

Μεθοδολογία

Ας είναι L μια γλώσσα (set boolean queries που εκφράζονται μέσω της L) και C μια κλαση πολυπλοκότητας (set boolean queries υπολογίσιμα στην C).
Για νδο ένα query ανήκει στην C ανν εκφράζεται μέσω της L, ακολουθούμε την εξής διαδικασία:

Μέθοδος

1. Δημιουργία C− αλγορίθμου που εξετάζει για κάθε ϕ, A της L, αν $A \models \phi$.
2. Βρίσκουμε ένα boolean query T που είναι complete στην C ως προς FO- reductions.
3. Δείχνω ότι L κλειστό ως προς FO- reductions.

Μεθοδολογία

Ας είναι L μια γλώσσα (set boolean queries που εκφράζονται μέσω της L) και C μια κλαση πολυπλοκότητας (set boolean queries υπολογίσιμα στην C).
Για νδο ένα query ανήκει στην C ανν εκφράζεται μέσω της L, ακολουθούμε την εξής διαδικασία:

Μέθοδος

1. Δημιουργία C– αλγορίθμου που εξετάζει για κάθε ϕ, A της L, αν $A \models \phi$.
2. Βρίσκουμε ένα boolean query T που είναι complete στην C ως προς FO- reductions.
3. Δείχνω ότι L κλειστό ως προς FO- reductions.
4. Εκφράζω την T μέσω της L.

Μεθοδολογία

Ας είναι L μια γλώσσα (set boolean queries που εκφράζονται μέσω της L) και C μια κλαση πολυπλοκότητας (set boolean queries υπολογίσιμα στην C).
Για νδο ένα query ανήκει στην C ανν εκφράζεται μέσω της L, ακολουθούμε την εξής διαδικασία:

Μέθοδος

1. Δημιουργία C− αλγορίθμου που εξετάζει για κάθε ϕ, A της L, αν $A \models \phi$.
2. Βρίσκουμε ένα boolean query T που είναι complete στην C ως προς FO- reductions.
3. Δείχνω ότι L κλειστό ως προς FO- reductions.
4. Εκφράζω την T μέσω της L.

Παρατήρηση : Από 1. $L \subseteq C$ και από 2.-4. $C \subseteq L$.

ΕΡΩΤΗΜΑ: Γιατί μας ενδιαφέρουν οι FO- reductions;
Θα δούμε ότι κάποια προβλήματα που ορίζονται με φυσικό τρόπο σε κλάσεις πολυπλ. (Reachability Problems) παραμένουν πλήρη ως προς FO - reductions!

Περιεχόμενα Παρουσίασης

Σε προβλήματα απόφασης, χρειαζόμαστε μια TM M που απλώς αποδέχεται ή απορρίπτει την είσοδο της, δηλ δεν μας απασχολεί η ταινία εξόδου.

Σε προβλήματα απόφασης, χρειαζόμαστε μια TM M που απλώς αποδέχεται ή απορρίπτει την είσοδο της, δηλ δεν μας απασχολεί η ταινία εξόδου.
Ένα configuration της M θα έχει τη μορφή:

$$(\boldsymbol{q}, \boldsymbol{i}, \boldsymbol{w}_1, \boldsymbol{w}_2).$$

Σε προβλήματα απόφασης, χρειαζόμαστε μια TM M που απλώς αποδέχεται ή απορρίπτει την είσοδο της, δηλ δεν μας απασχολεί η ταινία εξόδου.
Ένα configuration της M θα έχει τη μορφή:

$$(\boldsymbol{q}, \boldsymbol{i}, \boldsymbol{w}_1, \boldsymbol{w}_2).$$

- *q η κατάσταση στην οποία βρίσκεται η M,*

Σε προβλήματα απόφασης, χρειαζόμαστε μια TM M που απλώς αποδέχεται ή απορρίπτει την είσοδο της, δηλ δεν μας απασχολεί η ταινία εξόδου.
Ένα configuration της M θα έχει τη μορφή:

$$(\boldsymbol{q}, \boldsymbol{i}, \boldsymbol{w}_1, \boldsymbol{w}_2).$$

- ***q η κατάσταση στην οποία βρίσκεται η M,***
- ***i η θέση της κεφαλής εισόδου,***

Σε προβλήματα απόφασης, χρειαζόμαστε μια TM M που απλώς αποδέχεται ή απορρίπτει την είσοδο της, δηλ δεν μας απασχολεί η ταινία εξόδου.
Ένα configuration της M θα έχει τη μορφή:

$$(\boldsymbol{q}, \boldsymbol{i}, \boldsymbol{w}_1, \boldsymbol{w}_2).$$

- ***q η κατάσταση στην οποία βρίσκεται η M,***
- ***i η θέση της κεφαλής εισόδου,***
- $\boldsymbol{w}_1$ ***το περιεχόμενο της ταινίας εργασίας μέχρι και το σημείο που δείχνει η κεφαλή,***

Σε προβλήματα απόφασης, χρειαζόμαστε μια TM M που απλώς αποδέχεται ή απορρίπτει την είσοδο της, δηλ δεν μας απασχολεί η ταινία εξόδου.
Ένα configuration της M θα έχει τη μορφή:

$$(\boldsymbol{q}, \boldsymbol{i}, \boldsymbol{w}_1, \boldsymbol{w}_2).$$

- ***q η κατάσταση στην οποία βρίσκεται η M,***
- ***i η θέση της κεφαλής εισόδου,***
- ***$\boldsymbol{w}_1$ το περιεχόμενο της ταινίας εργασίας μέχρι και το σημείο που δείχνει η κεφαλή,***
- ***$\boldsymbol{w}_2$ το υπόλοιπο περιεχόμενο της ταινίας εργασίας.***

Παρατήρηση

Όλες οι configurations της M μπορούν να αντιστοιχηθούν στις κορυφές ενός γραφήματος!

Παρατήρηση

Όλες οι configurations της M μπορούν να αντιστοιχηθούν στις κορυφές ενός γραφήματος!

ΕΡΩΤΗΜΑ: Πότε αποδέχεται η M?

- *Μπορούμε να αναγνωρίσουμε το αρχικό configuration C_s.*

Παρατήρηση

Όλες οι configurations της M μπορούν να αντιστοιχηθούν στις κορυφές ενός γραφήματος!

ΕΡΩΤΗΜΑ: Πότε αποδέχεται η M?

- *Μπορούμε να αναγνωρίσουμε το αρχικό configuration C_s.*
- *Μια ακμή (C, C') αντιστοιχεί σε μία μετάβαση από ένα configuration στο αμέσως επόμενο.*

Παρατήρηση

Όλες οι configurations της M μπορούν να αντιστοιχηθούν στις κορυφές ενός γραφήματος!

ΕΡΩΤΗΜΑ: Πότε αποδέχεται η M?

- Μπορούμε να αναγνωρίσουμε το αρχικό configuration C_s.
- Μια ακμή (C, C') αντιστοιχεί σε μία μετάβαση από ένα configuration στο αμέσως επόμενο.
- Χβτγ, το C_a μπορούμε να πούμε ότι είναι μοναδικό, σβήνωντας τις ταινίες εργασίας.

Παρατήρηση

Όλες οι configurations της M μπορούν να αντιστοιχηθούν στις κορυφές ενός γραφήματος!

ΕΡΩΤΗΜΑ: Πότε αποδέχεται η M?

- Μπορούμε να αναγνωρίσουμε το αρχικό configuration C_s.
- Μια ακμή (C, C') αντιστοιχεί σε μία μετάβαση από ένα configuration στο αμέσως επόμενο.
- Χβτγ, το C_a μπορούμε να πούμε ότι είναι μοναδικό, σβήνωντας τις ταινίες εργασίας.
- Η M αποδέχεται την είσοδό της ανν υπάρχει μονοπάτι από το $C_s \to C_a$.

Παρατήρηση

Όλες οι configurations της M μπορούν να αντιστοιχηθούν στις κορυφές ενός γραφήματος!

ΕΡΩΤΗΜΑ: Πότε αποδέχεται η M?

- Μπορούμε να αναγνωρίσουμε το αρχικό configuration C_s.
- Μια ακμή (C, C') αντιστοιχεί σε μία μετάβαση από ένα configuration στο αμέσως επόμενο.
- Χβτγ, το C_a μπορούμε να πούμε ότι είναι μοναδικό, σβήνωντας τις ταινίες εργασίας.
- Η M αποδέχεται την είσοδό της ανν υπάρχει μονοπάτι από το $C_s \to C_a$.

Οι κορυφές του γραφήματος θα είναι

$$|Q|n|\Sigma|^{2\log n} = O(n^c)$$

REACH Problem

Ορίζουμε το ακόλουθο πρόβλημα:

REACH Problem

Ορίζουμε το ακόλουθο πρόβλημα:

REACH

REACH $= \{(G, s, t)$: *G κατευθυνόμενο γράφημα με μονοπάτι από την κορυφή s στην t*$\}$.

REACH Problem

Ορίζουμε το ακόλουθο πρόβλημα:

REACH

$REACH = \{(G, s, t) :$ G κατευθυνόμενο γράφημα με μονοπάτι από την κορυφή s στην t$\}$.

Θα αποδείξουμε ότι το REACH είναι NL− complete ως προς FO− reductions.

REACH $\in$ NL

Ο ακόλουθος μη ντετερμινιστικός logspace αλγόριθμος λύνει το REACH

REACH $\in$ NL

Ο ακόλουθος μη ντετερμινιστικός logspace αλγόριθμος λύνει το REACH

Αλγόριθμος

1. $b := s$
2. while $((b \neq t)$ do $\{$
3. $a := b$
4. nondeterministically choose new b
5. if (not $E(a, b))$ then reject$\}$
6. accept

REACH ∈ NL

Ο ακόλουθος μη ντετερμινιστικός logspace αλγόριθμος λύνει το REACH

Αλγόριθμος

1. $b := s$
2. while $((b \neq t)$ do $\{$
3. $a := b$
4. nondeterministically choose new b
5. if (not $E(a, b))$ then reject$\}$
6. accept

Ο παραπάνω αλγόριθμος χρειάζεται να αποθηκεύει μόνο τα a, b, τα οποία απαιτούν $\log n$ bits.

Θεώρημα

REACH complete στην NL με FO− reductions.

Απόδειξη

Θεωρούμε $\sigma =< \boldsymbol{R}_1^{\boldsymbol{a}_1}, ..., \boldsymbol{R}_{\boldsymbol{r}}^{\boldsymbol{a}_r}, \boldsymbol{c}_1, ..., \boldsymbol{c}_{\boldsymbol{s}} >$ ***και*** $\tau_{\boldsymbol{g}} =< \boldsymbol{E}^2, \boldsymbol{s}, \boldsymbol{t} >$.

Θεώρημα

REACH complete στην NL με FO− reductions.

Απόδειξη

Θεωρούμε $\sigma =< R_1^{a_1}, ..., R_r^{a_r}, c_1, ..., c_s >$ ***και*** $\tau_g =< E^2, s, t >$.
Έστω N μια logspace μη ντετερμινιστική TM, η οποία αποδέχεται ένα boolean query S, $S \subseteq STRUC[\sigma]$.

Θεώρημα

REACH complete στην NL με FO− reductions.

Απόδειξη

Θεωρούμε $\sigma =< R_1^{a_1}, ..., R_r^{a_r}, c_1, ..., c_s >$ ***και*** $\tau_g =< E^2, s, t >$.
Έστω N μια logspace μη ντετερμινιστική TM, η οποία αποδέχεται ένα boolean query S, $S \subseteq STRUC[\sigma]$.
Θα κάνουμε μια FO- reduction $I : STRUC[\sigma] \to STRUC[\tau_g]$ ***τ.ω***

Θεώρημα

REACH complete στην NL με FO– reductions.

Απόδειξη

Θεωρούμε $\sigma =< R_1^{a_1}, ..., R_r^{a_r}, c_1, ..., c_s >$ *και* $\tau_g =< E^2, s, t >$.
Έστω N μια logspace μη ντετερμινιστική TM, η οποία αποδέχεται ένα boolean query S, $S \subseteq STRUC[\sigma]$.
Θα κάνουμε μια FO- reduction $I : STRUC[\sigma] \rightarrow STRUC[\tau_g]$ *τ.ω*

$$N(bin(A)) \downarrow \Leftrightarrow I(A) \in REACH$$

για κάθε $A \in STRUC[\sigma]$.

Απόδειξη

Υποθέτουμε ότι η N χρησιμοποιεί το πολύ $c\log n$ bits της ταινίας εργασίας, όπου $n = ||A||$.

Απόδειξη

Υποθέτουμε ότι η N χρησιμοποιεί το πολύ $c \log n$ bits της ταινίας εργασίας, όπου $n = ||A||$.
Οι καταστάσεις της N και το πλήθος των σχεσ. συμβόλων και των σταθερών του σ δεν εξαρτάται από το μέγεθος της εισόδου και είναι σταθερές.

Απόδειξη

Υποθέτουμε ότι n N χρησιμοποιεί το πολύ $c \log n$ bits της ταινίας εργασίας, όπου $n = ||A||$.
Οι καταστάσεις της N και το πλήθος των σχεσ. συμβόλων και των σταθερών του σ δεν εξαρτάται από το μέγεθος της εισόδου και είναι σταθερές.
Έστω $a = max\{a_i : 1 \leq i \leq r\}$ και $k = a + c + 1$.

Απόδειξη

Υποθέτουμε ότι η N χρησιμοποιεί το πολύ $c \log n$ *bits της ταινίας εργασίας, όπου* $n = ||A||$.
Οι καταστάσεις της N και το πλήθος των σχεσ. συμβόλων και των σταθερών του σ *δεν εξαρτάται από το μέγεθος της εισόδου και είναι σταθερές.*
Έστω $a = max\{a_i : 1 \leq i \leq r\}$ *και* $k = a + c + 1$.
Το I θα είναι ένα k− FO query, επομένως, το $I(A)$ *θα αποτελείται από tuples μήκους k με στοιχεία από το* $|A|$, *όπου*

$$I(A) =< |I(A)|, R_1^{I(A)}, ..., R_r^{I(A)}, c_1^{I(A)}, ..., c_s^{I(A)} > .$$

Απόδειξη

Υποθέτουμε ότι η N χρησιμοποιεί το πολύ $c\log n$ bits της ταινίας εργασίας, όπου $n = ||A||$.
Οι καταστάσεις της N και το πλήθος των σχεσ. συμβόλων και των σταθερών του σ δεν εξαρτάται από το μέγεθος της εισόδου και είναι σταθερές.
Έστω $a = max\{a_i : 1 \leq i \leq r\}$ και $k = a + c + 1$.
Το I θα είναι ένα $k-$ FO query, επομένως, το $I(A)$ θα αποτελέιται από tuples μήκους k με στοιχεία από το $|A|$, όπου

$$I(A) =< |I(A)|, R_1^{I(A)}, ..., R_r^{I(A)}, c_1^{I(A)}, ..., c_s^{I(A)} > .$$

Θεωρούμε ένα τρέξιμο της N με είσοδο $bin(A)$.

Απόδειξη

Μπορούμε να κωδικοποιήσουμε ένα configuration της N ως:

Απόδειξη

Μπορούμε να κωδικοποιήσουμε ένα configuration της N ως:

$$C = (p, r_1, ..., r_a, w_1, ..., w_c)$$

όπου τα p, r_i, w_j έχουν $\log n$ bits.

Απόδειξη

Μπορούμε να κωδικοποιήσουμε ένα configuration της N ως:

$$C = (p, r_1, ..., r_a, w_1, ..., w_c)$$

όπου τα p, r_i, w_j έχουν $\log n$ bits.

- Τα $r_1, ..., r_a$ κωδικ. το πού βλέπει η κεφαλή της εισόδου, δηλ για κάποιο R_i τι bit δείχνει. Ειδικότερα,

Απόδειξη

Μπορούμε να κωδικοποιήσουμε ένα configuration της N ως:

$$C = (p, r_1, ..., r_a, w_1, ..., w_c)$$

όπου τα p, r_i, w_j έχουν $\log n$ bits.

- Τα $r_1, ..., r_a$ κωδικ. το πού βλέπει η κεφαλή της εισόδου, δηλ για κάποιο R_i τι bit δείχνει. Ειδικότερα, η κεφαλή δείχνει 1 στη δυαδική αναπαράσταση της R_i ανν $A \models R_i(r_1, ..., r_a)$.

Απόδειξη

Μπορούμε να κωδικοποιήσουμε ένα configuration της N ως:

$$C = (p, r_1, ..., r_a, w_1, ..., w_c)$$

όπου τα p, r_i, w_j έχουν $\log n$ bits.

- Τα $r_1, ..., r_a$ κωδικ. το πού βλέπει η κεφαλή της εισόδου, δηλ για κάποιο R_i τι bit δείχνει. Ειδικότερα, η κεφαλή δείχνει 1 στη δυαδική αναπαράσταση της R_i ανν $A \models R_i(r_1, ..., r_a)$.
- Τα $w_1, ..., w_c$ κωδικ. το περιεχόμενο της ταινίας εργασίας.

Απόδειξη

Μπορούμε να κωδικοποιήσουμε ένα configuration της N ως:

$$C = (p, r_1, ..., r_a, w_1, ..., w_c)$$

όπου τα p, r_i, w_j έχουν $\log n$ bits.

- **Τα $r_1, ..., r_a$ κωδικ. το πού βλέπει η κεφαλή της εισόδου, δηλ για κάποιο R_i τι bit δείχνει. Ειδικότερα, η κεφαλή δείχνει 1 στη δυαδική αναπαράσταση της R_i ανν $A \models R_i(r_1, ..., r_a)$.**
- **Τα $w_1, ..., w_c$ κωδικ. το περιεχόμενο της ταινίας εργασίας.**
- **Το p κωδικ. την τωρινή κατάσταση της N, ποιό σχεσ. σύμβολο ή σταθερα δείχνει η κεφαλή εισόδου και τη θέση της κεφαλής της ταινίας εργασίας.**

Απόδειξη

Ειδικότερα, το p απαιτεί $O(\log\log n)$ *bits.*

Απόδειξη

Ειδικότερα, το p απαιτεί $O(\log \log n)$ bits.

1. *Για την κατάσταση της N απαιτούνται $O(1)$ bits, αφού είναι μια σταθερά.*

Απόδειξη

Ειδικότερα, το p απαιτεί $O(\log\log n)$ bits.

1. *Για την κατάσταση της N απαιτούνται $O(1)$ bits, αφού είναι μια σταθερά.*
2. *Για τη θέση της της κεφαλής εισόδου απαιτούνται $O(\log n)$ bits, αφού το πλήθος των σχεσ. συμβόλων και σταθερών είναι σταθερές.*

Απόδειξη

Ειδικότερα, το p απαιτεί $O(\log\log n)$ bits.

1. *Για την κατάσταση της N απαιτούνται $O(1)$ bits, αφού είναι μια σταθερά.*
2. *Για τη θέση της της κεφαλής εισόδου απαιτούνται $O(\log n)$ bits, αφού το πλήθος των σχεσ. συμβόλων και σταθερών είναι σταθερές.*
3. *Η N χρησιμοποιεί το πολύ $c\log n$ bits της ταινίας εργασίας, άρα απαιτούνται $\log(c\log n)$ bits, δηλαδή χώρος $O(\log\log n)$.*

Απόδειξη

Κατασκευάζουμε τη ζητούμενη k− FO query

$$I = \lambda_{C,C'}(\boldsymbol{true}, \phi_{N}, \alpha, \omega)$$

Απόδειξη

Κατασκευάζουμε τη ζητούμενη k− FO query

$$I = \lambda_{C,C'}(\boldsymbol{true}, \phi_{\mathbf{N}}, \alpha, \omega)$$

όπου θέλουμε να ικανοποιεί τη σχέση

$$\boldsymbol{N}(\boldsymbol{bin}(\boldsymbol{A})) \downarrow \Leftrightarrow \boldsymbol{I}(\boldsymbol{A}) \in \boldsymbol{REACH}$$

Απόδειξη

Κατασκευάζουμε τη ζητούμενη k− FO query

$$I = \lambda_{C,C'}(\boldsymbol{true}, \phi_{\mathbf{N}}, \alpha, \omega)$$

όπου θέλουμε να ικανοποιεί τη σχέση

$$\boldsymbol{N}(\boldsymbol{bin}(\boldsymbol{A}))\downarrow \Leftrightarrow \boldsymbol{I}(\boldsymbol{A}) \in \boldsymbol{REACH}$$

- **Το true σημαίνει ότι το σύνολο των κορυφών του $\boldsymbol{I}(\boldsymbol{A})$ ισούται με το σύνολο όλων των δυνατών configurations της N.**

Απόδειξη

Κατασκευάζουμε τη ζητούμενη k− FO query

$$I = \lambda_{C,C'}(\textbf{true}, \phi_N, \alpha, \omega)$$

όπου θέλουμε να ικανοποιεί τη σχέση

$$N(bin(A)) \downarrow \Leftrightarrow I(A) \in REACH$$

- *Το true σημαίνει ότι το σύνολο των κορυφών του $I(A)$ ισούται με το σύνολο όλων των δυνατών configurations της N.*
- *Οι φόρμουλες ϕ_N, α, ω αναπαριστούν το σχεσιακό σύμβολο των ακμών, την αρχική κορυφή s (source node) και την τελική κορυφή t (target node).*

Απόδειξη

- $A \models \phi_N(C, C')$ ανν (C, C') έγκυρη κίνηση της TM με είσοδο $bin(A)$.

Απόδειξη

- $A \models \phi_N(C, C')$ ανν (C, C') έγκυρη κίνηση της TM με είσοδο $bin(A)$.
- $A \models \alpha(C)$ ανν το C είναι το μοναδικό αρχικό configuration της N.

Απόδειξη

- $A \models \phi_N(C, C')$ ανν (C, C') έγκυρη κίνηση της TM με είσοδο $bin(A)$.
- $A \models \alpha(C)$ ανν το C είναι το μοναδικό αρχικό configuration της N.
- $A \models \omega(C)$ ανν το C είναι το μοναδικό accept configuration της N.

Απόδειξη

- $A \models \phi_N(C, C')$ ανν (C, C') έγκυρη κίνηση της TM με είσοδο $bin(A)$.
- $A \models \alpha(C)$ ανν το C είναι το μοναδικό αρχικό configuration της N.
- $A \models \omega(C)$ ανν το C είναι το μοναδικό accept configuration της N.

Τα παραπάνω, τα ελέγχουμε από τα σύμβολα $p, r_1, ..., r_a, w_1, ..., w_c$, τα οποία περιγράφουν όλα τα configurations.

Απόδειξη

- $A \models \phi_N(C, C')$ ανν (C, C') έγκυρη κίνηση της TM με είσοδο $bin(A)$.
- $A \models \alpha(C)$ ανν το C είναι το μοναδικό αρχικό configuration της N.
- $A \models \omega(C)$ ανν το C είναι το μοναδικό accept configuration της N.

Τα παραπάνω, τα ελέγχουμε από τα σύμβολα $p, r_1, ..., r_a, w_1, ..., w_c$, τα οποία περιγράφουν όλα τα configurations.
Έτσι, για κάθε $A \in STRUC[\sigma]$, το $I(A)$ είναι το γράφημα που απεικονίζει τα configurations της N με input bin(A).
H N αποδέχεται την bin(a) αν υπάρχει μονοπάτι από την s στην t.

Περιεχόμενα Παρουσίασης

$REACH_d$

Με μια τροποποίηση του REACH, ορίζουμε το εξής πρόβλημα:

$REACH_d$

$REACH_d = \{(G, s, t)$: G κατευθυνόμενο γράφημα και υπάρχει ένα ντετερμινιστικό μονοπάτι από την κορυφή s στην $t\}$.

$REACH_d$

Με μια τροποποίηση του REACH, ορίζουμε το εξής πρόβλημα:

$REACH_d$

$REACH_d = \{(G, s, t)$: *G κατευθυνόμενο γράφημα και υπάρχει ένα ντετερμινιστικό μονοπάτι από την κορυφή s στην t*$\}$.

Ένα μονοπάτι P καλείται ντετερμινιστικό, αν για κάθε ακμή $(x, y) \in P$ *υπάρχει μοναδική ακμή από την κορυφή x προς μία άλλη, έστω y.*

$REACH_d$

Με μια τροποποίηση του REACH, ορίζουμε το εξής πρόβλημα:

$REACH_d$

$REACH_d = \{(G, s, t)$: G κατευθυνόμενο γράφημα και υπάρχει ένα ντετερμινιστικό μονοπάτι από την κορυφή s στην $t\}$.

Ένα μονοπάτι P καλείται ντετερμινιστικό, αν για κάθε ακμή $(x, y) \in P$ υπάρχει μοναδική ακμή από την κορυφή x προς μία άλλη, έστω y.

Παρατήρηση

$REACH_d \subseteq REACH$.

$REACH_d \in L$

Ο ακόλουθος ντετερμινιστικός logspace αλγόριθμος λύνει το $REACH_d$.

Αλγόριθμος

1. b:=s; i:=0; n:=$||G||$
2. while $b \neq t \wedge i < n \wedge (\exists! a)(E((b,a))$ do $\{$
3. $b :=$ the unique a for which $E(b,a)$
4. $i := i + 1\}$
5. if $b = t$ then accept else reject

Θεώρημα

$REACH_d$ complete στην L με FO- reductions.

Θεώρημα

$REACH_d$ complete στην L με FO- reductions.

Σκίτσο Απόδειξης

- *Παρόμοια με αποδ. REACH.*

Θεώρημα

$REACH_d$ complete στην L με FO- reductions.

Σκίτσο Απόδειξης

- Παρόμοια με αποδ. REACH.
- Σημαντική Διαφορά! Η TM M είναι ντετερμινιστική!

Θεώρημα

$REACH_d$ complete στην L με FO- reductions.

Σκίτσο Απόδειξης

- Παρόμοια με αποδ. REACH.
- Σημαντική Διαφορά! Η TM M είναι ντετερμινιστική! Άρα, κάθε configuration έχει μοναδικό επόμενο configuration.

Θεώρημα

$REACH_d$ complete στην L με FO- reductions.

Σκίτσο Απόδειξης

- Παρόμοια με αποδ. REACH.
- Σημαντική Διαφορά! Η TM M είναι ντετερμινιστική!
  Άρα, κάθε configuration έχει μοναδικό επόμενο configuration.
  Άρα, από κάθε κορυφή του γραφήματος $I(A)$ φεύγει μοναδική ακμή.

Θεώρημα

$REACH_d$ complete στην L με FO- reductions.

Σκίτσο Απόδειξης

- Παρόμοια με αποδ. REACH.
- Σημαντική Διαφορά! Η TM M είναι ντετερμινιστική!
  Άρα, κάθε configuration έχει μοναδικό επόμενο configuration.
  Άρα, από κάθε κορυφή του γραφήματος $I(A)$ φεύγει μοναδική ακμή.
- Για κάθε $A \in STRUC[\sigma]$;

$$N(bin(A)) \downarrow \Leftrightarrow I(A) \in REACH_d$$

Περιεχόμενα Παρουσίασης

$REACH_a$ Problem

Εναλασσόμενο Γράφημα

Ένα κατευθυνόμενο γράφημα $G = (V, E, A, s, t)$, όπου οι κορυφές του γραφήματος διακρίνονται σε δύο κατηγορίες:

$REACH_a$ Problem

Εναλασσόμενο Γράφημα

Ένα κατευθυνόμενο γράφημα $G = (V, E, A, s, t)$, όπου οι κορυφές του γραφήματος διακρίνονται σε δύο κατηγορίες:

1. Καθολικές κορυφές (A), όπου οι εξερχόμενες ακμές από τις συγκεκριμένες κορυφές πρέπει να οδηγούν σε μονοπάτι με πέρας την t (προφανώς πρέπει να έχουν μια τουλάχιστον εξερχόμενη ακμή).

$REACH_a$ Problem

Εναλασσόμενο Γράφημα

Ένα κατευθυνόμενο γράφημα $G = (V, E, A, s, t)$, όπου οι κορυφές του γραφήματος διακρίνονται σε δύο κατηγορίες:

1. Καθολικές κορυφές (A), όπου οι εξερχόμενες ακμές από τις συγκεκριμένες κορυφές πρέπει να οδηγούν σε μονοπάτι με πέρας την t (προφανώς πρέπει να έχουν μια τουλάχιστον εξερχόμενη ακμή).
2. Υπαρξιακές κορυφές $(V \setminus A)$, όπου αρκεί μια εξερχόμενη ακμή από τις συγκεκριμένες κορυφές να οδηγεί σε μονοπάτι με πέρας την t.

$REACH_a$ Problem

Εναλασσόμενο Γράφημα

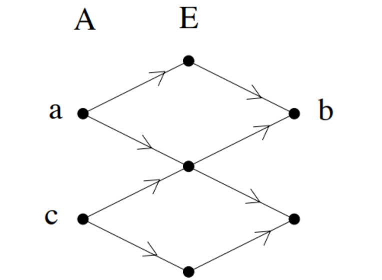
Ένα κατευθυνόμενο γράφημα $G = (V, E, A, s, t)$, όπου οι κορυφές του γραφήματος διακρίνονται σε δύο κατηγορίες:

1. Καθολικές κορυφές (A), όπου οι εξερχόμενες ακμές από τις συγκεκριμένες κορυφές πρέπει να οδηγούν σε μονοπάτι με πέρας την t (προφανώς πρέπει να έχουν μια τουλάχιστον εξερχόμενη ακμή).
2. Υπαρξιακές κορυφές $(V \setminus A)$, όπου αρκεί μια εξερχόμενη ακμή από τις συγκεκριμένες κορυφές να οδηγεί σε μονοπάτι με πέρας την t.

Ορίζεται voc $\tau_{ag} =< E^2, A_1, s, t >$.

Παράδειγμα

Έστω ότι δίνεται το παρακάτω γράφημα:



- Καθολικές κορυφές: a,c.
- Υπαρξιακές κορυφές: όλες οι υπόλοιπες.

$REACH_a$ Problem

P^G

Έστω G ένα εναλασσόμενο γράφημα.

$REACH_a$ Problem

P^G

Έστω G ένα εναλασσόμενο γράφημα.
P^G είναι η μικρότερη δυαδική σχέση που ικανοποιεί τα εξής:

1. $P^G(x, x)$.

$REACH_a$ Problem

P^G

Έστω G ένα εναλασσόμενο γράφημα.
P^G είναι η μικρότερη δυαδική σχέση που ικανοποιεί τα εξής:

1. $P^G(x,x)$.
2. Αν η κορυφή x είναι υπαρξιακή και για ακμή (x,z) ισχύει $P^G(z,y)$, τότε $P^G(x,y)$.

$REACH_a$ Problem

P^G

Έστω G ένα εναλλασσόμενο γράφημα.
P^G είναι η μικρότερη δυαδική σχέση που ικανοποιεί τα εξής:

1. $P^G(x,x)$.
2. Αν η κορυφή x είναι υπαρξιακή και για ακμή (x,z) ισχύει $P^G(z,y)$, τότε $P^G(x,y)$.
3. Αν η κορυφή x είναι καθολική, η x έχει τουλ μια εξερχόμενη ακμή και για όλες τις ακμές (x,z) ισχύει $P^G(z,y)$, τότε $P^G(x,y)$.

$REACH_a$

$REACH_a = \{(G,s,t) : P^G(s,t)\}$

Παράδειγμα

Στο προηγούμενο παράδειγμα:

$$
\left(\begin{array}{c|c|c|c|c|c|c|c}
P^G & \boldsymbol{a} & \boldsymbol{c} & \boldsymbol{e}_1 & \boldsymbol{e}_2 & \boldsymbol{e}_3 & \boldsymbol{b} & \boldsymbol{b}' \\
\hline
\boldsymbol{a} & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hline
\boldsymbol{c} & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
\hline
\boldsymbol{e}_1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
\hline
\boldsymbol{e}_2 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
\hline
\boldsymbol{e}_3 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
\hline
\boldsymbol{b} & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hline
\boldsymbol{b}' & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{array}\right)
$$

$REACH_a$ Problem

Ο ακόλουθος αλγόριθμος υπολογίζει το $REACH_a$ σε γραμμικό χρόνο.

Αλγόριθμος

1. make QUEUE empty; markt (t); insert t into QUEUE
2. while QUEUE not empty do {
3. remove first element, x , from QUEUE
4. for each unmarked vertex y such that E(y,x) do {
5. delete edge (y,x)
6. if y is existential or y has no outgoing edges
7. then { mark (y); insert y into QUEUE}}}
8. if s is marked then accept else reject

Παράδειγμα

Εφαρμόζουμε τον αλγόριθμο στο προηγ. παράδειγμα:

- Η κορυφή b μαρκάρεται, προστίθεται στην ουρά και μετά αφαιρείται.
- Οι ακμές (e_1, b) και (e_2, b) διαγράφονται και οι κορυφές e_1, e_2 μαρκάρονται και προστίθενται στην ουρά.
- Η κορυφή e_1 αφαιρείται από την ουρά και η ακμή (a, e_1) διαγράφεται.
- Η κορυφή e_2 αφαιρείται από την ουρά και οι ακμές $(a, e_1), (c, e_2)$ διαγράφονται.
- Η κορυφή a μαρκάρεται και προστίθεται στην ουρά, άρα αποδεχόμαστε.

Alternating TM

Ορισμός

Μια alternating TM (ATM) είναι μια TM, της οποίας οι καταστάσεις διακρίνονται σε δύο κατηγορίες:

Alternating TM

Ορισμός

Μια alternating TM (ATM) είναι μια TM, της οποίας οι καταστάσεις διακρίνονται σε δύο κατηγορίες:

1. Καθολικές καταστάσεις
2. Υπαρξιακές καταστάσεις

Alternating TM

Ορισμός

Μια alternating TM (ATM) είναι μια TM, της οποίας οι καταστάσεις διακρίνονται σε δύο κατηγορίες:

1. Καθολικές καταστάσεις
2. Υπαρξιακές καταστάσεις

Η ATM δεδομένου ενός configuration c αποδέχεται ανν:

- c βρίσκεται σε τελική κατάσταση αποδοχής,

Alternating TM

Ορισμός

Μια alternating TM (ATM) είναι μια TM, της οποίας οι καταστάσεις διακρίνονται σε δύο κατηγορίες:

1. Καθολικές καταστάσεις
2. Υπαρξιακές καταστάσεις

Η ATM δεδομένου ενός configuration c αποδέχεται ανν:

- c βρίσκεται σε τελική κατάσταση αποδοχής,
- c βρίσκεται σε υπαρξιακή κατάσταση και υπάρχει επόμενο configuration c' που αποδέχεται,

Alternating TM

Ορισμός

Μια alternating TM (ATM) είναι μια TM, της οποίας οι καταστάσεις διακρίνονται σε δύο κατηγορίες:

1. Καθολικές καταστάσεις
2. Υπαρξιακές καταστάσεις

Η ATM δεδομένου ενός configuration c αποδέχεται ανν:

- c βρίσκεται σε τελική κατάσταση αποδοχής,
- c βρίσκεται σε υπαρξιακή κατάσταση και υπάρχει επόμενο configuration c' που αποδέχεται,
- c βρίσκεται σε καθολική κατάσταση και όλες οι επόμενες configurations αποδέχονται

Alternating TM

Ορισμός

Μια alternating TM (ATM) είναι μια TM, της οποίας οι καταστάσεις διακρίνονται σε δύο κατηγορίες:

1. Καθολικές καταστάσεις
2. Υπαρξιακές καταστάσεις

Η ATM δεδομένου ενός configuration c αποδέχεται ανν:

- c βρίσκεται σε τελική κατάσταση αποδοχής,
- c βρίσκεται σε υπαρξιακή κατάσταση και υπάρχει επόμενο configuration c' που αποδέχεται,
- c βρίσκεται σε καθολική κατάσταση και όλες οι επόμενες configurations αποδέχονται

Ισχύει ότι $P = ASPACE[\log n]$.

Θεώρημα

$REACH_a$ complete στην L με FO - reductions.

Θεώρημα

$REACH_a$ complete στην L με FO - reductions.

Σκίτσο Απόδειξης

1. Παρόμοια με αποδ. REACH, $REACH_d$.
2. Σημαντικές Διαφορές! Η TM M είναι μια alternating logspace TM.

Θεώρημα

$REACH_a$ complete στην L με FO - reductions.

Σκίτσο Απόδειξης

1. *Παρόμοια με αποδ. REACH, $REACH_d$.*
2. *Σημαντικές Διαφορές! Η TM M είναι μια alternating logspace TM.*
   Ακόμη, πρέπει οι καθολικές καταστάσεις να αντιστοιχηθούν σε καθολικές κορυφές του $I(A)$.

Ευχαριστώ για την προσοχή σας!