Αναγωγές και Πληρότητα

- Ορισμός των $\leq_l$ αναγωγών
- Ιδιότητες και παραδείγματα αναγωγών
- $\mathcal{C}$-πλήρη προβλήματα για μια κλάση $\mathcal{C}$

Αναγωγές

- $TSP(D)$ ανήκει στην NP
- $REACHABILITY$ ανήκει στην NP

Εισάγουμε την έννοια της αναγωγής ώστε να δείχνουμε ότι ένα πρόβλημα A είναι τουλάχιστον τόσο δύσκολο όσο το B. Ανάγουμε το B στο A και γράφουμε B $\leq_l$ A.

Ορισμός

Μία γλώσσα L_1 ανάγεται σε μία γλώσσα L_2 αν υπάρχει μία συνάρτηση R υπολογίσιμη από μία μηχανή *Turing* λογαριθμικού χώρου, τέτοια ώστε για κάθε είσοδο x να ισχύει: $x \in L_1$ αν και μόνο αν $R(x) \in L_2$. Τότε η συνάρτηση R λέγεται **αναγωγή**.

Αναγωγές

Πρόταση

Αν R είναι μια αναγωγή υπολογίσιμη από μια μηχανή *Turing* M, τότε για κάθε είσοδο x η M σταματάει μετά από πολυωνυμικού πλήθους βήματα.

Απόδειξη: Για κάθε είσοδο x, $|x| = n$, υπάρχουν $\mathcal{O}(nc^{logn})$ δυνατά *configurations*. Επειδή η M είναι ντετερμινιστική, κάθε *configuration* σχηματίζεται το πολύ μία φορά κατά τη διάρκεια κάποιου υπολογισμού. Άρα το μήκος του υπολογισμού είναι το πολύ $\mathcal{O}(n^k)$ για κάποιο k.

HAMILTON PATH $\leq_l$ SAT

Δοθέντος ενός γράφου G κατασκευάζουμε σε λογαριθμικό χώρο έναν τύπο R(G) τέτοιο ώστε ο R(G) είναι ικανοποιήσιμος αν και μόνο αν ο G έχει Hamilton Path.

Η μεταβλητή x_{ij} αντιστοιχεί στο ότι *«η κορυφή j είναι η i-οστή κορυφή στο Hamilton Path»*.

Ο τύπος $R(G)$ είναι η τομή των παρακάτω *clauses*:

1. $(x_{1j} \vee x_{2j} \vee \ldots \vee x_{nj})$, για κάθε j
2. $(\neg x_{ij} \vee \neg x_{kj})$, για κάθε j και $i \neq k$
3. $(\neg x_{ki} \vee \neg x_{k+1,j})$ αν η (i,j) δεν είναι ακμή του G για $k = 0, \ldots, n-1$

$HAMILTON\ PATH \leq_l SAT$

R(G) ικανοποιήσιμος $\implies$ G έχει Hamilton Path:
Λόγω των 1 και 2, για κάθε j υπάρχει μοναδικό i τέτοιο ώστε $T(x_{ij}) = true$.
Παίρνοντας $\pi(i) = j$ ανν $T(x_{ij}) = true$, έχουμε μία μετάθεση των κορυφών $1, \ldots n$.
Λόγω του 3, για κάθε k $(\pi(k), \pi(k+1))$ ανήκει στο G.
G έχει Hamilton Path $\implies$ R(G) ικανοποιήσιμος.
Έστω $(\pi(1), \ldots \pi(n))$ ένα *Hamilton Path* του G. Τότε όλες οι *clauses* ικανοποιούνται από την T με $T(x_{ij}) = true$ αν $\pi(i) = j$ και $T(x_{ij}) = false$ αν $\pi(i) \neq j$.
Παρατηρούμε ότι η κατασκευή του R(x) απαιτεί λογαριθμικό χώρο!

$HAMILTON\ PATH \leq_l SAT$

R(G) ικανοποιήσιμος $\implies$ G έχει Hamilton Path:
Λόγω των 1 και 2, για κάθε j υπάρχει μοναδικό i τέτοιο ώστε $T(x_{ij}) = true$.
Παίρνοντας $\pi(i) = j$ αvv $T(x_{ij}) = true$, έχουμε μία μετάθεση των κορυφών $1, \ldots n$.
Λόγω του 3, για κάθε k $(\pi(k), \pi(k+1))$ ανήκει στο G.
G έχει Hamilton Path $\implies$ R(G) ικανοποιήσιμος.
Έστω $(\pi(1), \ldots \pi(n))$ ένα *Hamilton Path* του G. Τότε όλες οι *clauses* ικανοποιούνται από την T με $T(x_{ij}) = true$ αν $\pi(i) = j$ και $T(x_{ij}) = false$ αν $\pi(i) \neq j$.
Παρατηρούμε ότι η κατασκευή του R(x) απαιτεί λογαριθμικό χώρο!

$REACHABILITY \leq_l CIRCUIT\ VALUE$

Δοθέντος ενός γράφου G κατασκευάζουμε σε λογαριθμικό χώρο ένα Boolean Circuit R(G) τέτοιο ώστε η έξοδός του είναι true αν και μόνο αν υπάρχει μονοπάτι από την κορυφή 1 στην κορυφή n του G.

Η πύλη g_{ijk} είναι αληθής ανν *«υπάρχει μονοπάτι από την* i *τσην* j *στο* G *μη χρησιμοποιώντας κορυφές μεγαλύτερες της* k*».*

Το $R(G)$ έχει πύλες g_{ijk} για $1 \leq i,j \leq n$ και $0 \leq k \leq n$, από τις οποίες g_{ij0} είναι πύλες εισόδου και η g_{1nn} η πύλη εξόδου.
Προφανώς, $g_{ij0} = true$ ανν $i = j$ ή (i,j) είναι ακμή του G.

$REACHABILITY \leq_l CIRCUIT\ VALUE$

Η πύλη h_{ijk} είναι αληθής ανν «υπάρχει μονοπάτι από την i τσην j στο G μη χρησιμοποιώντας κορυφές μεγαλύτερες της k, αλλά χρησιμοποιώντας την k».

Το $R(G)$ έχει πύλες h_{ijk} για $1 \leq i, j, k \leq n$.

- Κάθε h_{ijk} είναι AND πύλη με εισερχόμενες ακμές από τις πύλες $g_{i,k,k-1}$ και $g_{k,j,k-1}$
- Κάθε g_{ijk} είναι OR πύλη με εισερχόμενες ακμές από τις πύλες $g_{i,j,k-1}$ και $h_{i,j,k}$

Η κατασκευή βασίζεται στον αλγόριθμο $Floyd-Warshall$ για το $REACHABILITY$!

$REACHABILITY \leq_l CIRCUIT\ VALUE$

Η πύλη h_{ijk} είναι αληθής ανν *«υπάρχει μονοπάτι από την i τσην j στο G μη χρησιμοποιώντας κορυφές μεγαλύτερες της k, αλλά χρησιμοποιώντας την k»*.

Το $R(G)$ έχει πύλες h_{ijk} για $1 \leq i, j, k \leq n$.

- Κάθε h_{ijk} είναι AND πύλη με εισερχόμενες ακμές από τις πύλες $g_{i,k,k-1}$ και $g_{k,j,k-1}$
- Κάθε g_{ijk} είναι OR πύλη με εισερχόμενες ακμές από τις πύλες $g_{i,j,k-1}$ και $h_{i,j,k}$

Η κατασκευή βασίζεται στον αλγόριθμο $Floyd - Warshall$ για το $REACHABILITY$!

$CIRCUIT\ SAT \leq_l SAT$

Δοθέντος ενός Boolean Circuit C κατασκευάζουμε σε λογαριθμικό χώρο έναν τύπο R(C) τέτοιο ώστε ο R(C) είναι ικανοποιήσιμος αν και μόνο αν το C είναι ικανοποιήσιμο.

Οι μεταβλητές του $R(C)$ περιέχουν όλες τις μεταβλητές του C.
Κάθε πύλη g αντιστοιχεί σε μια καινούρια μεταβλητή, έστω g, και στην τομή κάποιων *clauses* στον $R(C)$.

Αν η g είναι:

- πύλη-μεταβλητή x, τότε προσθέτουμε τις *clauses* $\neg g \vee x$ και $g \vee \neg x$ (ισοδύναμα $g \leftrightarrow x$)
- *true*-πύλη, προσθέτουμε την *clause* g
- *false*-πύλη, προσθέτουμε την *clause* $\neg g$

$CIRCUIT\ SAT \leq_l SAT$

- NOT-πύλη με εισερχόμενη ακμή από την πύλη h, προσθέτουμε τις *clauses* $\neg g \vee \neg h$ και $g \vee h$ (ισοδύναμα $g \leftrightarrow \neg h$)
- OR-πύλη με εισερχόμενες ακμές από τις πύλες h και $h^{'}$, προσθέτουμε τις *clauses* $\neg h \vee g$, $\neg h^{'} \vee g$ και $h \vee h^{'} \vee \neg g$ (ισοδύναμα $g \leftrightarrow h \vee h^{'}$)
- AND-πύλη με εισερχόμενες ακμές από τις πύλες h και $h^{'}$, προσθέτουμε τις *clauses* $\neg g \vee h$, $\neg g \vee h^{'}$ και $\neg h \vee \neg h^{'} \vee g$ (ισοδύναμα $g \leftrightarrow h \wedge h^{'}$)
- πύλη εξόδου, προσθέτουμε την *clause* g

Αναγωγές

Μεταβατικότητα της αναγωγιμότητας

Αν R είναι αναγωγή από τη γλώσσα L_1 στη γλώσσα L_2 και R' είναι αναγωγή από την L_2 στην L_3, τότε η σύνθεση $R' \circ R$ είναι αναγωγή από την L_1 στην L_3.

Απόδειξη: Προφανώς ισχύει $x \in L_1$ ανν $R'(R(x)) \in L_3$.
Η $R' \circ R$ μπορεί να υπολογιστεί από μηχανή M λογαριθμικού χώρου!
Η μηχανή M «βασικά» τρέχει τη $\mathbf{M_{R'}}$ με είσοδο $\mathbf{R(x)}$:
Χρησιμοποιεί ένα δείκτη i για να «θυμάται» σε ποια θέση βρίσκεται ο κέρσορας της εισόδου της $M_{R'}$. Σε κάθε βήμα τρέχει τη μηχανή M_R με είσοδο x μέχρι αυτή να παράξει το επόμενο σύμβολο στην έξοδό της. Αυτό το σύμβολο διαβάζεται από τη $M_{R'}$. Αν ο κέρσορας εισόδου της $M_{R'}$ κινηθεί δεξιά το i αυξάνεται κατά ένα και αρχίζει το επόμενο βήμα. Αν ο κέρσορας κινηθεί προς τα αριστερά ξεκινάει τη M_R από την αρχή.

Πληρότητα

- Η αναγωγιμότητα είναι μερική διάταξη.
- Τα μεγιστικά στοιχεία αυτής της διάταξης σε μια κλάση προβλημάτων είναι τα πλήρη προβλήματα αυτής της κλάσης.

Ορισμός

Έστω η κλάση $\mathcal{C}$ και $L \in \mathcal{C}$. Λέμε ότι η L είναι $\mathcal{C}$-πλήρης αν κάθε $L' \in \mathcal{C}$ ανάγεται στην L.

- **NP**-πλήρη: *SAT*, *INDEPENDENT SET*, *HAMILTON PATH*
- **P**-πλήρη: *CIRCUIT VALUE*, *HORNSAT*
- **NL**-πλήρη: *2SAT*, *REACHABILITY*

Πληρότητα

- **Πώς δείχνουμε ότι ένα πρόβλημα Α είναι NP-πλήρες;**
  Δείχνουμε ότι το A ανήκει στην κλάση **NP** και ανάγουμε ένα γνωστό **NP**-πλήρες πρόβλημα στο A.

Ορισμός

- Μία γλώσσα L_1 ανάγεται **κατά Karp** στην L_2 ($L_1 \leq_m^P L_2$) αν υπάρχει μία συνάρτηση R υπολογίσιμη από μία μηχανή *Turing* πολυωνυμικού χρόνου, τέτοια ώστε για κάθε είσοδο x να ισχύει $x \in L_1$ αν και μόνα αν $R(x) \in L_2$.
- Μία γλώσσα L_1 ανάγεται **κατά Cook** στην L_2 ($L_1 \leq_T^P L_2$) αν υπάρχει μία μηχανή *Turing* πολυωνυμικού χρόνου που να αποφασίζει την L_1, η οποία έχει μία έξτρα ταινία, τέτοια ώστε όταν γραφεί ένα *string* x στην ταινία, παίρνει την απάντηση στην ερώτηση «$x \in L_2$?» σε ένα υπολογιστικό βήμα.

Πληρότητα

Ορισμός

Μια κλάση $\mathcal{C}$ είναι κλειστή ως προς τις αναγωγές αν όποτε η L ανάγεται στην L' και $L' \in \mathcal{C}$, τότε και $L \in \mathcal{C}$.

Πρόταση

Οι κλάσεις **P**, **NP**, **L**, **NL**, **PSPACE**, **EXP** είναι κλειστές ως προς τις αναγωγές.

Αν ένα $\mathcal{C}$-πλήρες πρόβλημα ανήκει σε μια κλειστή ως προς τις αναγωγές κλάση $\mathcal{C}' \subseteq \mathcal{C}$, τότε $\mathcal{C} = \mathcal{C}'$.

Πληρότητα

Αν ένα $\mathcal{C}$-πλήρες πρόβλημα ανήκει σε μια κλειστή ως προς τις αναγωγές κλάση $\mathcal{C}' \subseteq \mathcal{C}$, τότε $\mathcal{C} = \mathcal{C}'$.

Πρόταση

Αν δύο κλάσεις $\mathcal{C}$ και $\mathcal{C}'$ είναι κλειστές ως προς τις αναγωγές και υπάρχει γλώσσα L η οποία είναι πλήρης και για τις δύο, τότε $\mathcal{C} = \mathcal{C}'$.

Απόδειξη: Όλες οι γλώσσες της $\mathcal{C}$ ανάγονται στην $L \in \mathcal{C}'$. Άρα $\mathcal{C} \subseteq \mathcal{C}'$. Ομοίως $\mathcal{C}' \subseteq \mathcal{C}$.

Το *CIRCUIT VALUE* είναι *P*-πλήρες

Θεώρημα

Το *CIRCUIT VALUE* είναι *P*-πλήρες.

Ιδέα της απόδειξης: Ανάγουμε κάθε $L \in \mathbf{P}$ στο *CIRCUIT VALUE*: Κωδικοποιούμε κάθε σύμβολο που μπορεί να εμφανιστεί στο *computation table* της L με κάποιο *string* μήκους m. Το περιεχόμενο κάθε κελιού του *computation table* εξαρτάται μόνο από τρία προηγούμενα κελιά (ή $3m$ λόγω κωδικοποίησης). Την εξάρτηση των επόμενων κελιών απ'τα προηγούμενα την εκφράζουμε με *Boolean functions* και εναλλακτικά με *Boolean circuits*.