universität innsbruck

Diskrete Mathematik

Ralph Bottesch
David Obwaller
Burak Ekici
Vincent van Oostrom
Johannes Koch
Oleksandra Panasiuk
Georg Moser

cbr.uibk.ac.at

Zusammenfassung der letzten LVA

Definition

1. $\exists\, R\colon \Sigma^* \to \Delta^*$
2. R berechenbar in polynomieller Zeit
3. für $A \subseteq \Sigma^*$, $B \subseteq \Delta^*$ gilt $x \in A \Leftrightarrow R(x) \in B$

dann ist A in polynomieller Zeit auf B reduzierbar; kurz: $A \leqslant^{\mathrm{p}} B$

Definition

für eine Sprache B, sei

1. B $\leqslant^{\mathrm{p}}$-hart für $\mathcal{C}$ und
2. $B \in \mathcal{C}$

dann ist B $\leqslant^{\mathrm{p}}$-vollständig für $\mathcal{C}$ oder (kurz) $\mathcal{C}$-vollständig

Inhalte der Lehrveranstaltung (cont'd)

Reguläre Sprachen

deterministische Automaten, nichtdeterministische Automaten, endliche Automaten mit Epsilon-Übergängen, reguläre Ausdrücke, Abgeschossenheit, Schleifenlemma

Berechenbarkeitstheorie

deterministische TM, nichtdeterministische TM, universelle TMs, Äquivalenzen

Komplexitätstheorie

Grundlagen, die Klassen P und NP, polynomielle Reduktion, logspace Reduktion, die Klassen NLOGSPACE und PSPACE

Inhalte der Lehrveranstaltung (cont'd)

Reguläre Sprachen

deterministische Automaten, nichtdeterministische Automaten, endliche Automaten mit Epsilon-Übergängen, reguläre Ausdrücke, Abgeschossenheit, Schleifenlemma

Berechenbarkeitstheorie

deterministische TM, nichtdeterministische TM, universelle TMs, Äquivalenzen

Komplexitätstheorie

Grundlagen, die Klassen P und NP, polynomielle Reduktion, logspace Reduktion, die Klassen NLOGSPACE und PSPACE

Beispiel

Wir zeigen mittels einer polynomiellen Reduktion nach SAT dass 3-Färbbarkeit von Graphen in NP ist

Beispiel

Wir zeigen mittels einer polynomiellen Reduktion nach SAT dass 3-Färbbarkeit von Graphen in NP ist

1. jede Ecke muss genau eine Farbe haben

$$A = \bigwedge_{e \in E} X_{1e} \vee X_{2e} \vee X_{3e}$$

$$B = \bigwedge_{e \in E} (X_{1e} \rightarrow (\neg X_{2e} \wedge \neg X_{3e})) \wedge (X_{2e} \rightarrow (\neg X_{1e} \wedge \neg X_{3e})) \wedge$$
$$\wedge (X_{3e} \rightarrow (\neg X_{1e} \wedge \neg X_{2e}))$$

Beispiel

Wir zeigen mittels einer polynomiellen Reduktion nach SAT dass 3-Färbbarkeit von Graphen in NP ist

1. jede Ecke muss genau eine Farbe haben

$$A = \bigwedge_{e \in E} X_{1e} \vee X_{2e} \vee X_{3e}$$

$$B = \bigwedge_{e \in E} (X_{1e} \rightarrow (\neg X_{2e} \wedge \neg X_{3e})) \wedge (X_{2e} \rightarrow (\neg X_{1e} \wedge \neg X_{3e})) \wedge$$
$$\wedge (X_{3e} \rightarrow (\neg X_{1e} \wedge \neg X_{2e}))$$

2. und die Farben von benachbarten Ecken (also Ecken zwischen denen es eine Kante gibt) müssen unterschiedlich sein

$$C = \bigwedge_{k \in K, x \in r(k), y \in r(k), x \neq y, f \in \{1,2,3\}} X_{fx} \rightarrow \neg X_{fy}$$

Beispiel

HK ist $\leqslant^{p}$-vollständig für NP

Beispiel

HK ist $\leqslant^p$-vollständig für NP

Folgerung

- *wenn* $\exists$ *polynomieller Algorithmus für HK, dann*
- $\exists$ *polynomieller Algorithmus* $\forall$ *Probleme in* NP

Beispiel

HK ist $\leqslant^p$-vollständig für NP

Folgerung

- *wenn* $\exists$ *polynomieller Algorithmus für HK, dann*
- $\exists$ *polynomieller Algorithmus* $\forall$ *Probleme in* NP

Beispiele

- Maze $\in$ NP, aber Maze ist nicht hart für NP

Beispiel

HK ist $\leqslant^{\mathrm{p}}$-vollständig für NP

Folgerung

- *wenn* $\exists$ *polynomieller Algorithmus für HK, dann*
- $\exists$ *polynomieller Algorithmus* $\forall$ *Probleme in* NP

Beispiele

- Maze $\in$ NP, aber Maze ist nicht hart für NP
- definiere GEO als Sprache

$$\mathrm{GEO} := \{(G, s) \mid \text{gerichteter Graph } G, \text{ Startknoten } s, \exists \text{ Gewinnstrategie für Anna}\}$$

GEO ist hart für NP, aber GEO $\notin$ NP

Komplexitätstheorie, graphisch

“I can't find an efficient algorithm, but neither can all these famous people.”

Definition

sei M eine totale Mehrband-DTM

- die Speicherplatzkomplexität von M ist eine Funktion $S\colon \mathbb{N} \to \mathbb{N}$,
  S misst die maximale Anzahl von Bandfeldern, die M auf allen Eingaben liest
- $S(n)$ ist der Speicherplatz von M, wenn n die Länge der Eingabe
- M heißt S-Platz-Turingmaschine

Definition

sei M eine totale Mehrband-DTM

- die Speicherplatzkomplexität von M ist eine Funktion $S\colon \mathbb{N} \to \mathbb{N}$,
  S misst die maximale Anzahl von Bandfeldern, die M auf allen Eingaben liest
- $S(n)$ ist der Speicherplatz von M, wenn n die Länge der Eingabe
- M heißt S-Platz-Turingmaschine

Definition

sei M eine totale Mehrband-NTM

- die Speicherplatzkomplexität von M ist Funktion $S\colon \mathbb{N} \to \mathbb{N}$,
  S misst die maximale Anzahl von Bandfeldern, die M auf allen Eingaben in jedem möglichen Berechnungspfad liest
- M heißt S-Platz-Turingmaschine

Definition (vorläufig)

sei $S\colon \mathbb{N} \to \mathbb{N}$ eine numerische Funktion

$$\mathrm{DSPACE}(S) := \{\mathrm{L}(M) \mid M \text{ ist eine mehrbändige DTM mit Speicherplatz in } \mathrm{O}(S)\}$$

$$\mathrm{NSPACE}(S) := \{\mathrm{L}(N) \mid N \text{ ist eine mehrbändige NTM mit Speicherplatz in } \mathrm{O}(S)\}$$

Definition (vorläufig)

sei $S\colon \mathbb{N} \to \mathbb{N}$ eine numerische Funktion

$$\mathrm{DSPACE}(S) := \{\mathrm{L}(M) \mid M \text{ ist eine mehrbändige DTM mit Speicherplatz in } \mathrm{O}(S)\ \}$$

$$\mathrm{NSPACE}(S) := \{\mathrm{L}(N) \mid N \text{ ist eine mehrbändige NTM mit Speicherplatz in } \mathrm{O}(S)\ \}$$

Definition (vorläufig)

$$\mathrm{PSPACE} = \bigcup_{k \geqslant 1} \mathrm{DSPACE}(n^k) \qquad \mathrm{NPSPACE} = \bigcup_{k \geqslant 1} \mathrm{NSPACE}(n^k)$$

Definition (vorläufig)

sei $S\colon \mathbb{N} \to \mathbb{N}$ eine numerische Funktion

$$\mathrm{DSPACE}(S) := \{\mathrm{L}(M) \mid M \text{ ist eine mehrbändige DTM mit Speicherplatz in } \mathrm{O}(S)\ \}$$

$$\mathrm{NSPACE}(S) := \{\mathrm{L}(N) \mid N \text{ ist eine mehrbändige NTM mit Speicherplatz in } \mathrm{O}(S)\ \}$$

Definition (vorläufig)

$$\mathrm{PSPACE} = \bigcup_{k \geqslant 1} \mathrm{DSPACE}(n^k) \qquad \mathrm{NPSPACE} = \bigcup_{k \geqslant 1} \mathrm{NSPACE}(n^k)$$

Beispiel

$\mathrm{GEO} \in \mathrm{PSPACE}$

Definition (vorläufig)

sei $S\colon \mathbb{N} \to \mathbb{N}$ eine numerische Funktion

$$\mathrm{DSPACE}(S) := \{\mathrm{L}(M) \mid M \text{ ist eine mehrbändige DTM mit Speicherplatz in } \mathrm{O}(S)\ \}$$

$$\mathrm{NSPACE}(S) := \{\mathrm{L}(N) \mid N \text{ ist eine mehrbändige NTM mit Speicherplatz in } \mathrm{O}(S)\ \}$$

Definition (vorläufig)

$$\mathrm{PSPACE} = \bigcup_{k \geqslant 1} \mathrm{DSPACE}(n^k) \qquad \mathrm{NPSPACE} = \bigcup_{k \geqslant 1} \mathrm{NSPACE}(n^k)$$

Beispiel

GEO $\in$ PSPACE

Satz

$$\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} = \mathrm{NPSPACE}$$

Definition (vorläufig)

sei $S\colon \mathbb{N} \to \mathbb{N}$ eine numerische Funktion

$$\mathrm{DSPACE}(S) := \{\mathrm{L}(M) \mid M \text{ ist eine mehrbändige DTM mit Speicherplatz in } \mathrm{O}(S)\ \}$$

$$\mathrm{NSPACE}(S) := \{\mathrm{L}(N) \mid N \text{ ist eine mehrbändige NTM mit Speicherplatz in } \mathrm{O}(S)\ \}$$

Definition (vorläufig)

$$\mathrm{PSPACE} = \bigcup_{k \geqslant 1} \mathrm{DSPACE}(n^k) \qquad \mathrm{NPSPACE} = \bigcup_{k \geqslant 1} \mathrm{NSPACE}(n^k)$$

Beispiel

GEO $\in$ PSPACE

Satz

$$\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} = \mathrm{NPSPACE}$$

Beispiel

betrachte nichtdeterministischen Algorithmus A für Maze

1. gegeben Graph G, Start- und Endknoten s und t
2. sei b der aktuell betrachtete Knoten; setze $b = s$
3. wähle nichtdeterministisch einen Nachfolgerknoten b' von b
4. wenn $b' = t$, akzeptiere
5. sonst setze $b = b'$ und gehe zu Schritt 3

Beispiel

betrachte nichtdeterministischen Algorithmus A für Maze

1. gegeben Graph G, Start- und Endknoten s und t
2. sei b der aktuell betrachtete Knoten; setze $b = s$
3. wähle nichtdeterministisch einen Nachfolgerknoten b' von b
4. wenn $b' = t$, akzeptiere
5. sonst setze $b = b'$ und gehe zu Schritt 3

Frage

Was ist die Platzkomplexität von A?

Beispiel

betrachte nichtdeterministischen Algorithmus A für Maze

1. gegeben Graph G, Start- und Endknoten s und t
2. sei b der aktuell betrachtete Knoten; setze $b = s$
3. wähle nichtdeterministisch einen Nachfolgerknoten b' von b
4. wenn $b' = t$, akzeptiere
5. sonst setze $b = b'$ und gehe zu Schritt 3

Frage

Was ist die Platzkomplexität von A?

Antwort

Speicherplatzkomplexität einer TM, die A implementiert ist $O(n)$, wobei n die Länge der Codierung der Eingabe!

Definition

eine TM mit Eingabeband und Arbeitsband

$$M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$$

ist eine TM mit zwei Bändern, sodass

- das 1te Band ein Leseband — Eingabeband
- alle anderen Bänder sind les- und schreibbar — Arbeitsbänder

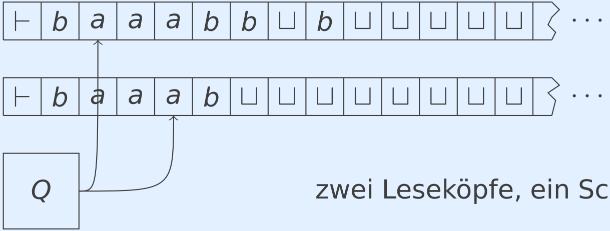
Definition

eine TM mit Eingabeband und Arbeitsband

$$M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$$

ist eine TM mit zwei Bändern, sodass

- das 1te Band ein Leseband — Eingabeband
- alle anderen Bänder sind les- und schreibbar — Arbeitsbänder



zwei Leseköpfe, ein Schreibkopf

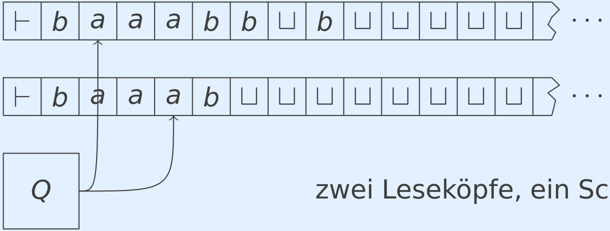
Definition

eine TM mit Eingabeband und Arbeitsband

$$M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$$

ist eine TM mit zwei Bändern, sodass

- das 1te Band ein Leseband — Eingabeband
- alle anderen Bänder sind les- und schreibbar — Arbeitsbänder



zwei Leseköpfe, ein Schreibkopf

Bedingung für δ

$$\delta(p, a, a) = (q, a, c, D_1, D_2) \qquad \forall\, p, q \in Q,\ a, b, c \in \Gamma,\ D_1, D_2 \in \{\mathrm{L}, \mathrm{R}\}$$

Definition

sei $S\colon \mathbb{N} \to \mathbb{N}$ eine numerische Funktion

$$\mathrm{DSPACE}(S) = \{\mathrm{L}(M) \mid M \text{ ist eine mehrbändige DTM mit Speicherplatz auf den Arbeitsbändern in } \mathrm{O}(S)\ \}$$

$$\mathrm{NSPACE}(S) = \{\mathrm{L}(N) \mid N \text{ ist eine mehrbändige NTM mit Speicherplatz auf den Arbeitsbändern in } \mathrm{O}(S)\ \}$$

Definition

sei $S\colon \mathbb{N} \to \mathbb{N}$ eine numerische Funktion

$$\text{DSPACE}(S) = \{\mathrm{L}(M) \mid M \text{ ist eine mehrbändige DTM mit Speicherplatz auf den Arbeitsbändern in } \mathrm{O}(S)\}$$

$$\text{NSPACE}(S) = \{\mathrm{L}(N) \mid N \text{ ist eine mehrbändige NTM mit Speicherplatz auf den Arbeitsbändern in } \mathrm{O}(S)\}$$

Definition

sei $S\colon \mathbb{N} \to \mathbb{N}$ eine numerische Funktion

$$\mathrm{DSPACE}(S) = \{\mathrm{L}(M) \mid M \text{ ist eine mehrbändige DTM mit Speicherplatz auf den Arbeitsbändern in } \mathrm{O}(S)\ \}$$

$$\mathrm{NSPACE}(S) = \{\mathrm{L}(N) \mid N \text{ ist eine mehrbändige NTM mit Speicherplatz auf den Arbeitsbändern in } \mathrm{O}(S)\ \}$$

Definition

$$\mathrm{LOGSPACE} = \mathrm{DSPACE}(\log n) \qquad \mathrm{NLOGSPACE} = \mathrm{NSPACE}(\log n)$$

$$\mathrm{PSPACE} = \bigcup_{k \geqslant 1} \mathrm{DSPACE}(n^k) \qquad \mathrm{NPSPACE} = \bigcup_{k \geqslant 1} \mathrm{NSPACE}(n^k)$$

Definition

sei $S\colon \mathbb{N} \to \mathbb{N}$ eine numerische Funktion

$$\mathrm{DSPACE}(S) = \{\mathrm{L}(M) \mid M \text{ ist eine mehrbändige DTM mit Speicherplatz auf den Arbeitsbändern in } \mathrm{O}(S)\ \}$$

$$\mathrm{NSPACE}(S) = \{\mathrm{L}(N) \mid N \text{ ist eine mehrbändige NTM mit Speicherplatz auf den Arbeitsbändern in } \mathrm{O}(S)\ \}$$

Definition

$$\mathrm{LOGSPACE} = \mathrm{DSPACE}(\log n) \qquad \mathrm{NLOGSPACE} = \mathrm{NSPACE}(\log n)$$

$$\mathrm{PSPACE} = \bigcup_{k \geqslant 1} \mathrm{DSPACE}(n^k) \qquad \mathrm{NPSPACE} = \bigcup_{k \geqslant 1} \mathrm{NSPACE}(n^k)$$

Beispiele

Maze $\in$ NLOGSPACE und GEO $\in$ PSPACE

Frage

Ist Maze NLOGSPACE-vollständig?

Frage

Ist Maze NLOGSPACE-vollständig?

Antwort

die Frage macht keinen Sinn!

Frage

Ist Maze NLOGSPACE-vollständig?

Antwort

die Frage macht keinen Sinn!

Definition

ein logarithmischer Umwandler

$$T = (Q, \Sigma, \Gamma, \Delta, \vdash, \sqcup, \delta, s, t)$$

ist eine totale DTM mit 3 Bändern, sodass

- das 1te Band ist Leseband — Eingabeband
- vom 2te Band mit $O(\log n)$ Zeichen, Eingabelänge n — Arbeitsband
- das 3te Band ist Schreibeband — Ausgabeband

Frage

Ist Maze NLOGSPACE-vollständig?

Antwort

die Frage macht keinen Sinn!

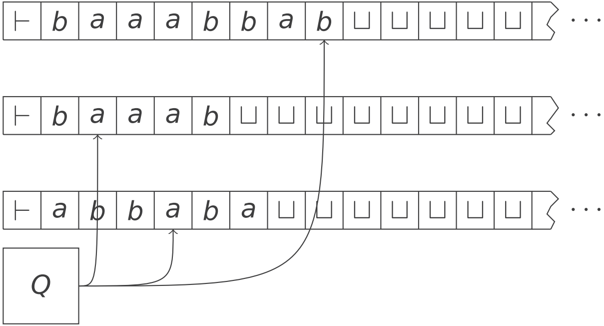
Definition

ein logarithmischer Umwandler

$$T = (Q, \Sigma, \Gamma, \Delta, \vdash, \sqcup, \delta, s, t)$$

ist eine totale DTM mit 3 Bändern, sodass

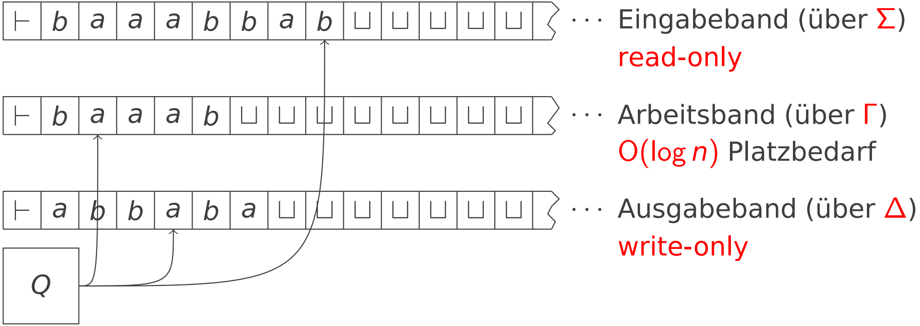
- das 1te Band ist Leseband — Eingabeband
- vom 2te Band mit $O(\log n)$ Zeichen, Eingabelänge n — Arbeitsband
- das 3te Band ist Schreibeband — Ausgabeband



Eingabeband (über Σ)
read-only

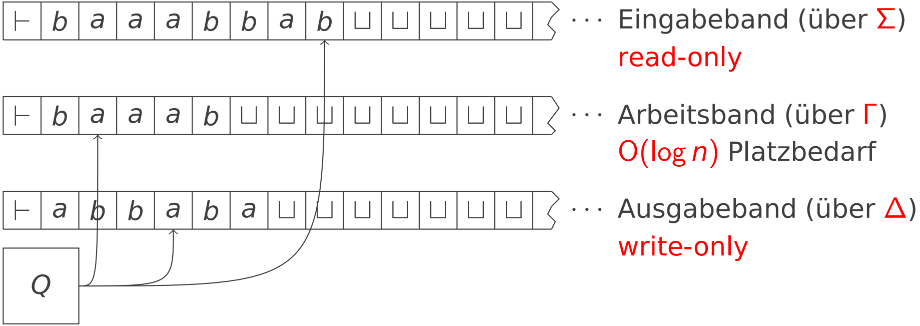
Arbeitsband (über Γ)
$O(\log n)$ Platzbedarf

Ausgabeband (über Δ)
write-only



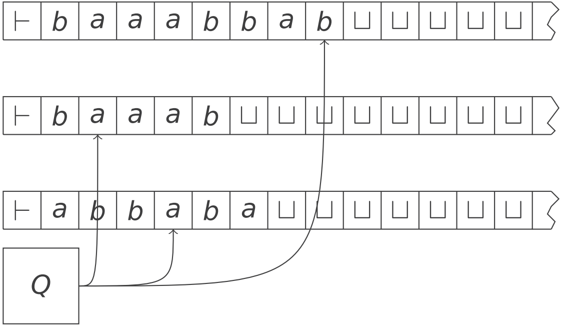
Definition

1 $\exists$ logarithmischer Umwandler M mit Eingabealphabet Σ und Ausgabealphabet Δ



Definition

1. $\exists$ logarithmischer Umwandler M mit Eingabealphabet Σ und Ausgabealphabet Δ
2. bei Eingabe $x \in \Sigma^*$, schreibt M $R(x)$ auf Ausgabeband



$\cdots$ Eingabeband (über Σ)
read-only

$\cdots$ Arbeitsband (über Γ)
$O(\log n)$ Platzbedarf

$\cdots$ Ausgabeband (über Δ)
write-only

Definition

1. $\exists$ logarithmischer Umwandler M mit Eingabealphabet Σ und Ausgabealphabet Δ
2. bei Eingabe $x \in \Sigma^*$, schreibt M $R(x)$ auf Ausgabeband

dann heißt $R\colon \Sigma^* \to \Delta^*$ berechenbar mit logarithmischem Platz

Reduktion mit logarithmischen Platz

Definition

1. $\exists\, R : \Sigma^* \to \Delta^*$
2. R berechenbar mit logarithmischem Platz
3. für $A \subseteq \Sigma^*$, $B \subseteq \Delta^*$ gilt $x \in A \Leftrightarrow R(x) \in B$

Reduktion mit logarithmischen Platz

Definition

1. $\exists\, R : \Sigma^* \to \Delta^*$
2. R berechenbar mit logarithmischem Platz
3. für $A \subseteq \Sigma^*$, $B \subseteq \Delta^*$ gilt $x \in A \Leftrightarrow R(x) \in B$

dann ist A auf B reduzierbar mit logarithmischem Platz ; kurz $A \leqslant^{\log} B$

Reduktion mit logarithmischen Platz

Definition

1. $\exists\, R : \Sigma^* \to \Delta^*$
2. R berechenbar mit logarithmischem Platz
3. für $A \subseteq \Sigma^*$, $B \subseteq \Delta^*$ gilt $x \in A \Leftrightarrow R(x) \in B$

dann ist A auf B reduzierbar mit logarithmischem Platz ; kurz $A \leqslant^{\mathrm{log}} B$

Satz

Seien A und B Sprachen; wenn $A \leqslant^{log} B$, dann gilt auch $A \leqslant^{p} B$ ■

Definition

sei B eine Sprache, $\mathcal{C}$ eine beliebige Komplexitätsklasse,

Definition

sei B eine Sprache, $\mathcal{C}$ eine beliebige Komplexitätsklasse,

1. wenn $\forall$ Sprachen $A \in \mathcal{C}$ gilt: $A \leqslant^{\log} B$
   dann ist B $\leqslant^{\log}$-hart für $\mathcal{C}$

Definition

sei B eine Sprache, $\mathcal{C}$ eine beliebige Komplexitätsklasse,

1. wenn $\forall$ Sprachen $A \in \mathcal{C}$ gilt: $A \leqslant^{\log} B$
   dann ist B $\leqslant^{\log}$-hart für $\mathcal{C}$
2. wenn B $\leqslant^{\log}$-hart für $\mathcal{C}$ und $B \in \mathcal{C}$
   dann ist B $\leqslant^{\log}$-vollständig für $\mathcal{C}$ oder (kurz) $\mathcal{C}$-vollständig

Definition

sei B eine Sprache, $\mathcal{C}$ eine beliebige Komplexitätsklasse,

1. wenn $\forall$ Sprachen $A \in \mathcal{C}$ gilt: $A \leqslant^{\log} B$
   dann ist B $\leqslant^{\log}$-hart für $\mathcal{C}$
2. wenn B $\leqslant^{\log}$-hart für $\mathcal{C}$ und $B \in \mathcal{C}$
   dann ist B $\leqslant^{\log}$-vollständig für $\mathcal{C}$ oder (kurz) $\mathcal{C}$-vollständig

Beispiele

- GEO ist $\leqslant^{\log}$-vollständig für PSPACE
- HK ist $\leqslant^{\log}$-vollständig für NP
- Maze ist $\leqslant^{\log}$-vollständig für NLOGSPACE

Definition

sei B eine Sprache, $\mathcal{C}$ eine beliebige Komplexitätsklasse,

1. wenn $\forall$ Sprachen $A \in \mathcal{C}$ gilt: $A \leqslant^{\log} B$
   dann ist B $\leqslant^{\log}$-hart für $\mathcal{C}$
2. wenn B $\leqslant^{\log}$-hart für $\mathcal{C}$ und $B \in \mathcal{C}$
   dann ist B $\leqslant^{\log}$-vollständig für $\mathcal{C}$ oder (kurz) $\mathcal{C}$-vollständig

Beispiele

- GEO ist $\leqslant^{\log}$-vollständig für PSPACE
- HK ist $\leqslant^{\log}$-vollständig für NP
- Maze ist $\leqslant^{\log}$-vollständig für NLOGSPACE

Satz

$$\text{LOGSPACE} \subseteq \text{NLOGSPACE} \subseteq \text{P} \subseteq \text{NP} \subseteq \text{PSPACE} = \text{NPSPACE}$$

Definition

sei B eine Sprache, $\mathcal{C}$ eine beliebige Komplexitätsklasse,

1. wenn $\forall$ Sprachen $A \in \mathcal{C}$ gilt: $A \leqslant^{\log} B$
   dann ist B $\leqslant^{\log}$-hart für $\mathcal{C}$
2. wenn B $\leqslant^{\log}$-hart für $\mathcal{C}$ und $B \in \mathcal{C}$
   dann ist B $\leqslant^{\log}$-vollständig für $\mathcal{C}$ oder (kurz) $\mathcal{C}$-vollständig

Beispiele

- GEO ist $\leqslant^{\log}$-vollständig für PSPACE
- HK ist $\leqslant^{\log}$-vollständig für NP
- Maze ist $\leqslant^{\log}$-vollständig für NLOGSPACE

Satz

$$\text{LOGSPACE} \subseteq \text{NLOGSPACE} \subseteq \text{P} \subseteq \text{NP} \subseteq \text{PSPACE} = \text{NPSPACE}$$

Klausurvorbereitung

Fragen aus dem Tutorium

Organisatorische Fragen

Bei der Klausur, Wie sieht die Punkteverteilung bei der Klausur aus, wenn das richtige Ergebnis ausgerechnet wurde, allerdings nicht mit dem gewünschtem Algorithmus?

Fragen aus dem Tutorium

Organisatorische Fragen

Bei der Klausur, Wie sieht die Punkteverteilung bei der Klausur aus, wenn das richtige Ergebnis ausgerechnet wurde, allerdings nicht mit dem gewünschtem Algorithmus?

Inhaltliche Fragen

- Was sind die Unterschiede zwischen den einzelnen Induktionsarten: (i) vollständige, (ii) wohlfundierte, oder (iii) strukturelle?

Fragen aus dem Tutorium

Organisatorische Fragen

Bei der Klausur, Wie sieht die Punkteverteilung bei der Klausur aus, wenn das richtige Ergebnis ausgerechnet wurde, allerdings nicht mit dem gewünschtem Algorithmus?

Inhaltliche Fragen

- Was sind die Unterschiede zwischen den einzelnen Induktionsarten: (i) vollständige, (ii) wohlfundierte, oder (iii) strukturelle?
- Warum ist die Reihenfolge (vom Abarbeiten von Kanten mit gleicher Bewertung) beim Algorithmus von Kruskal irrelevant?

Fragen aus dem Tutorium

Organisatorische Fragen

Bei der Klausur, Wie sieht die Punkteverteilung bei der Klausur aus, wenn das richtige Ergebnis ausgerechnet wurde, allerdings nicht mit dem gewünschtem Algorithmus?

Inhaltliche Fragen

- Was sind die Unterschiede zwischen den einzelnen Induktionsarten: (i) vollständige, (ii) wohlfundierte, oder (iii) strukturelle?
- Warum ist die Reihenfolge (vom Abarbeiten von Kanten mit gleicher Bewertung) beim Algorithmus von Kruskal irrelevant?
- Warum sind beim Minimierungsalgorithmus nicht immer alle akzeptierenden Zustände gleich?

Fragen aus dem Tutorium

Organisatorische Fragen

Bei der Klausur, Wie sieht die Punkteverteilung bei der Klausur aus, wenn das richtige Ergebnis ausgerechnet wurde, allerdings nicht mit dem gewünschtem Algorithmus?

Inhaltliche Fragen

- Was sind die Unterschiede zwischen den einzelnen Induktionsarten: (i) vollständige, (ii) wohlfundierte, oder (iii) strukturelle?
- Warum ist die Reihenfolge (vom Abarbeiten von Kanten mit gleicher Bewertung) beim Algorithmus von Kruskal irrelevant?
- Warum sind beim Minimierungsalgorithmus nicht immer alle akzeptierenden Zustände gleich?
- Bei der 1. Klausur (Version G) aus dem SS 2018 warum ist bei Aufgabe 10 die Antwortmöglichkeit B falsch?

Weitere Fragen

Inhaltliche Fragen

- In the Question 12 of version U [of the 1st Exam Summer 2017] it was asked to calculate the shortest paths in a weighted graph using the Algorithm of Floyd-Warshall. The Script (page 39) says that the weights in a graph must be non negative, while in the exam question some of them are negative. Does it mean they can be negative as long as there are no negative cycles in a graph?

Weitere Fragen

Inhaltliche Fragen

- In the Question 12 of version U [of the 1st Exam Summer 2017] it was asked to calculate the shortest paths in a weighted graph using the Algorithm of Floyd-Warshall. The Script (page 39) says that the weights in a graph must be non negative, while in the exam question some of them are negative. Does it mean they can be negative as long as there are no negative cycles in a graph?
- In the Question 15 (same version U) it was asked to convert ϵ-NEA to NEA and then to DEA, and according to the solution, we should not consider ?* before the input 0/1, otherwise there wouldn't be empty sets in NEA in the solution. The question is why shouldn't we consider the epsilon transition before the input.

Vielen Dank für Ihre Aufmerksamkeit!