Gliederung

PSPACE

Quelle: `http://commons.wikimedia.org/wiki/File:13_by_13_game_finished.jpg`

Platzkomplexität: PSPACE und LOGSPACE

Bisher: Klassifikation der Berechungsschwere von Problemen anhand von Zeitbedarf

Platzkomplexität: PSPACE und LOGSPACE

Bisher: Klassifikation der Berechungsschwere von Problemen anhand von Zeitbedarf

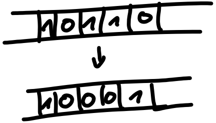
Jetzt: Zweites wichtiges Kriterium – Speicherplatzbedarf

Platzkomplexität: PSPACE und LOGSPACE

Bisher: Klassifikation der Berechungsschwere von Problemen anhand von Zeitbedarf

Jetzt: Zweites wichtiges Kriterium – Speicherplatzbedarf

Zentraler Unterschied: Speicherplatz ist wiederverwendbar!



Platzkomplexität: PSPACE und LOGSPACE

Bisher: Klassifikation der Berechungsschwere von Problemen anhand von Zeitbedarf

Jetzt: Zweites wichtiges Kriterium – Speicherplatzbedarf

Zentraler Unterschied: Speicherplatz ist wiederverwendbar!

Definition (PSPACE, LOGSPACE (informell))

Eine Sprache $L \subseteq \Sigma^*$ liegt in...

...PSPACE gdw. eine DTM M existiert mit $T(M) = L$ und M modifiziert höchstens polynomiell viele Speicherzellen.

Platzkomplexität: PSPACE und LOGSPACE

Bisher: Klassifikation der Berechungsschwere von Problemen anhand von Zeitbedarf

Jetzt: Zweites wichtiges Kriterium – Speicherplatzbedarf

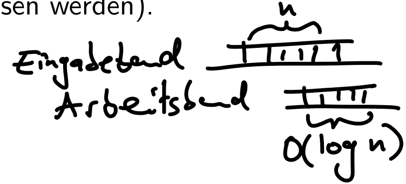
Zentraler Unterschied: Speicherplatz ist wiederverwendbar!

Definition (PSPACE, LOGSPACE (informell))

Eine Sprache $L \subseteq \Sigma^*$ liegt in...
...PSPACE gdw. eine DTM M existiert mit $T(M) = L$ und M modifiziert höchstens polynomiell viele Speicherzellen.
...LOGSPACE gdw. eine DTM M existiert mit $T(M) = L$ und M modifiziert höchstens logarithmisch viele Speicherzellen (Eingabe darf nur gelesen werden).

n $\log n$

$$\begin{array}{r} 17 \\ +\,201 \\ \hline \end{array}$$



Platzkomplexität: PSPACE und LOGSPACE

Bisher: Klassifikation der Berechungsschwere von Problemen anhand von Zeitbedarf

Jetzt: Zweites wichtiges Kriterium – Speicherplatzbedarf

Zentraler Unterschied: Speicherplatz ist wiederverwendbar!

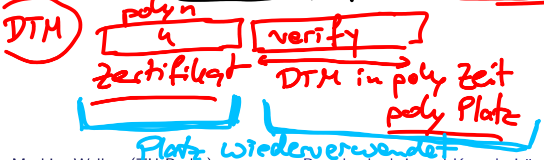
Definition (PSPACE, LOGSPACE (informell))

Eine Sprache $L \subseteq \Sigma^*$ liegt in...

...PSPACE gdw. eine DTM M existiert mit $T(M) = L$ und M modifiziert höchstens polynomiell viele Speicherzellen.

...LOGSPACE gdw. eine DTM M existiert mit $T(M) = L$ und M modifiziert höchstens logarithmisch viele Speicherzellen (Eingabe darf nur gelesen werden).

Mitteilung: LOGSPACE $\subseteq$ P $\subseteq$ NP $\subseteq$ PSPACE.



M benutzt $\underline{O(k)}$ Platz

$\underline{2^{O(k)}}$ Konfigurationen

$k = \log(n) \rightsquigarrow 2^{O(\log n)} \underline{= \text{poly}(n)}$

Platzkomplexität: PSPACE und LOGSPACE

Bisher: Klassifikation der Berechungsschwere von Problemen anhand von Zeitbedarf

Jetzt: Zweites wichtiges Kriterium – Speicherplatzbedarf

Zentraler Unterschied: Speicherplatz ist wiederverwendbar!

Definition (PSPACE, LOGSPACE (informell))

Eine Sprache $L \subseteq \Sigma^*$ liegt in...

...PSPACE gdw. eine DTM M existiert mit $T(M) = L$ und M modifiziert höchstens polynomiell viele Speicherzellen.

...LOGSPACE gdw. eine DTM M existiert mit $T(M) = L$ und M modifiziert höchstens logarithmisch viele Speicherzellen (Eingabe darf nur gelesen werden).

Mitteilung: LOGSPACE $\subseteq$ P $\subseteq$ NP $\subseteq$ PSPACE.

Definition

a) Eine Sprache A heißt **PSPACE-schwer**, falls $\forall_{L \in \text{PSPACE}}\ L \leq_m^p A$.

b) A heißt **PSPACE-vollständig**, wenn A PSPACE-schwer ist und $A \in$ PSPACE.

Ein PSPACE-vollständiges Problem

Erfüllbarkeit einer **quantifizierten** aussagenlogischen Formel.

Definition

Eine **quantifizierte** aussagenlogische Formel (in Pränexform) hat die Gestalt $Q_1 x_1 Q_2 x_2 \ldots Q_n x_n F(x_1, \ldots, x_n)$, wobei $Q_i \in \{\exists, \forall\}$ und F eine aussagenlogische Formel mit ungebundenen ("freien") Variablen x_i ist.

Ein PSPACE-vollständiges Problem

Erfüllbarkeit einer **quantifizierten** aussagenlogischen Formel.

Definition

Eine **quantifizierte** aussagenlogische Formel (in Pränexform) hat die Gestalt $Q_1 x_1 Q_2 x_2 \ldots Q_n x_n\ F(x_1, \ldots, x_n)$, wobei $Q_i \in \{\exists, \forall\}$ und F eine aussagenlogische Formel mit ungebundenen ("freien") Variablen x_i ist.

TQBF (True Quantified Boolean Formula) ist die Sprache aller wahren quantifizierten aussagenlogischen Formeln.

Ein PSPACE-vollständiges Problem

Erfüllbarkeit einer **quantifizierten** aussagenlogischen Formel.

Definition

Eine **quantifizierte** aussagenlogische Formel (in Pränexform) hat die Gestalt $Q_1x_1\,Q_2x_2\ldots Q_nx_n\;F(x_1,\ldots,x_n)$, wobei $Q_i \in \{\exists,\forall\}$ und F eine aussagenlogische Formel mit ungebundenen ("freien") Variablen x_i ist.

TQBF (True Quantified Boolean Formula) ist die Sprache aller $\boxed{\underline{\text{wahren}}}$ quantifizierten aussagenlogischen Formeln.

Beispiele:

$$\underline{\forall_x}\,\underline{\exists_y}\,\underline{(x \wedge y) \vee (\bar{x} \wedge \bar{y})}$$

$$\bar y$$

$$\cancel{0 \vee \bar y}$$

$$x=0 \rightsquigarrow \exists_y\, \cancel{(0 \wedge y)} \vee (1 \wedge \bar y) \rightsquigarrow \underbrace{\exists_y\, \bar y}_{\text{wahr}}$$

$$x=1 \rightsquigarrow \exists_y\, (1 \wedge y) \vee \cancel{(0 \wedge \bar y)} \rightsquigarrow \underbrace{\exists_y\, y}_{\text{wahr}}$$

$$\cancel{y \vee 0}$$

$$y$$

Ein PSPACE-vollständiges Problem

Erfüllbarkeit einer **quantifizierten** aussagenlogischen Formel.

Definition

Eine **quantifizierte** aussagenlogische Formel (in Pränexform) hat die Gestalt $Q_1 x_1 Q_2 x_2 \dots Q_n x_n \; F(x_1, \dots, x_n)$, wobei $Q_i \in \{\exists, \forall\}$ und F eine aussagenlogische Formel mit ungebundenen ("freien") Variablen x_i ist.

TQBF (True Quantified Boolean Formula) ist die Sprache aller wahren quantifizierten aussagenlogischen Formeln.

Beispiele:

$$\forall_x \exists_y \, (x \wedge y) \vee (\bar{x} \wedge \bar{y}) \in \underline{\mathrm{TQBF}}$$

Ein PSPACE-vollständiges Problem

Erfüllbarkeit einer **quantifizierten** aussagenlogischen Formel.

Definition

Eine **quantifizierte** aussagenlogische Formel (in Pränexform) hat die Gestalt $Q_1 x_1 Q_2 x_2 \ldots Q_n x_n\ F(x_1, \ldots, x_n)$, wobei $Q_i \in \{\exists, \forall\}$ und F eine aussagenlogische Formel mit ungebundenen ("freien") Variablen x_i ist.

TQBF (True Quantified Boolean Formula) ist die Sprache aller wahren quantifizierten aussagenlogischen Formeln.

Beispiele:

$$\forall_x \exists_y\ (x \wedge y) \vee (\bar{x} \wedge \bar{y}) \in \mathrm{TQBF}$$

$$\boxed{\underline{\forall_x \forall_y}\ \underline{(x \wedge y) \vee (\bar{x} \wedge \bar{y})}}$$

$(1 \wedge 0) \vee (0 \wedge 1)$

$\underline{x=1,\ y=0}$

Ein PSPACE-vollständiges Problem

Erfüllbarkeit einer **quantifizierten** aussagenlogischen Formel.

Definition

Eine **quantifizierte** aussagenlogische Formel (in Pränexform) hat die Gestalt $Q_1x_1 Q_2x_2 \dots Q_nx_n\ F(x_1, \dots, x_n)$, wobei $Q_i \in \{\exists, \forall\}$ und F eine aussagenlogische Formel mit ungebundenen ("freien") Variablen x_i ist.

TQBF (True Quantified Boolean Formula) ist die Sprache aller wahren quantifizierten aussagenlogischen Formeln.

Beispiele:

$$\forall_x \exists_y\ (x \wedge y) \vee (\bar{x} \wedge \bar{y}) \in \mathrm{TQBF}$$

$$\boxed{\underline{\forall_x} \forall_y\ (x \wedge y) \vee (\bar{x} \wedge \bar{y}) \notin \underline{\mathrm{TQBF}}}$$

Ein PSPACE-vollständiges Problem

$NP \subseteq \underset{\substack{\rotatebox{90}{$\subseteq$} \\ coNP}}{PSPACE}$

Erfüllbarkeit einer **quantifizierten** aussagenlogischen Formel.

Definition

Eine **quantifizierte** aussagenlogische Formel (in Pränexform) hat die Gestalt $Q_1 x_1 Q_2 x_2 \dots Q_n x_n \; F(x_1, \dots, x_n)$, wobei $Q_i \in \{\exists, \forall\}$ und F eine aussagenlogische Formel mit ungebundenen (“freien”) Variablen x_i ist.

TQBF (True Quantified Boolean Formula) ist die Sprache aller wahren quantifizierten aussagenlogischen Formeln.

Beispiele:

$$\forall_x \exists_y \; (x \wedge y) \vee (\bar{x} \wedge \bar{y}) \in \mathrm{TQBF}$$

$$\forall_x \forall_y \; (x \wedge y) \vee (\bar{x} \wedge \bar{y}) \notin \mathrm{TQBF}$$

Spezialfälle:

$\underline{\mathrm{SAT}} \rightsquigarrow \exists_{x_1} \dots \exists_{x_n} F(x_1, \dots, x_n) \in \mathrm{TQBF}$

$\underline{\mathrm{TAUT}} \rightsquigarrow \forall_{x_1} \dots \forall_{x_n} F(x_1, \dots, x_n) \in \mathrm{TQBF}$

Ein PSPACE-vollständiges Problem

Erfüllbarkeit einer **quantifizierten** aussagenlogischen Formel.

Definition

Eine **quantifizierte** aussagenlogische Formel (in Pränexform) hat die Gestalt $Q_1 x_1 Q_2 x_2 \dots Q_n x_n \; F(x_1, \dots, x_n)$, wobei $Q_i \in \{\exists, \forall\}$ und F eine aussagenlogische Formel mit ungebundenen ("freien") Variablen x_i ist.

TQBF (True Quantified Boolean Formula) ist die Sprache aller wahren quantifizierten aussagenlogischen Formeln.

$$\underbrace{\exists x_1 \forall x_2 \exists x_3 \forall x_4}\, \underbrace{\varphi(x_1,x_2,x_3,x_4)}$$

Beispiele:

$\boxed{\forall_x \exists_y \, (x \wedge y) \vee (\bar{x} \wedge \bar{y})} \in \underline{\text{TQBF}}$

$\forall_x \forall_y \, (x \wedge y) \vee (\bar{x} \wedge \bar{y}) \notin \text{TQBF}$

Spezialfälle:

SAT $\rightsquigarrow \exists_{x_1} \dots \exists_{x_n} F(x_1, \dots, x_n) \in \text{TQBF}$

TAUT $\rightsquigarrow \forall_{x_1} \dots \forall_{x_n} F(x_1, \dots, x_n) \in \text{TQBF}$

Beachte: Ähnlichkeit zu "Spielproblemen" (kann Spielerin X gewinnen?)

TQBF ist PSPACE-vollständig

Theorem

TQBF ist PSPACE-vollständig.

TQBF ist PSPACE-vollständig

$\boxed{\forall x_1} \exists x_2 \forall x_3 \exists x_4 \varphi(\ldots.)$

$\exists x_2 \forall x_3 \exists x_4 \quad \varphi(1, \ldots)$

$\varphi(0, \ldots.)$

Theorem

TQBF ist PSPACE-vollständig.

Beweis (Skizze)

1. $\mathrm{TQBF} \in \mathrm{PSPACE}$:

Werte die Formel $F(x_1, \ldots, x_n)$ rekursiv für alle Variablenbelegungen β aus ($O(n + |F|)$ Platz)

$2^n \rightsquigarrow$ poly(n) Platz

```
TQBF(i):
  IF $i \leq n$ THEN
    $\beta(x_i) := 0$;   $a :=$ TQBF($i+1$);
```

globale

TQBF ist PSPACE-vollständig

$$\underbrace{\boxed{\exists x_i \, \boxed{\forall x_{i+1} \dots \varphi(0..)}}}_{u} \rightsquigarrow a \qquad \rightsquigarrow b$$

Theorem

TQBF ist PSPACE-vollständig.

Beweis (Skizze)

1. TQBF $\in$ PSPACE:

Werte die Formel $F(x_1, \dots, x_n)$ rekursiv für alle Variablenbelegungen β aus ($O(n + |F|)$ Platz)

```
TQBF($i$):
  IF $i \leq n$ THEN
    $\beta(x_i) := 0$;   $a :=$ TQBF($i + 1$);
    $\beta(x_i) := 1$;   $b :=$ TQBF($i + 1$);
    IF $Q_i = \exists$ THEN
      RETURN $a \vee b$;
```

TQBF ist PSPACE-vollständig

$$\boxed{\forall x_i \boxed{\exists x_{i+1} \cdots \varphi(.0.)}} \to a$$

$$\_\_\_\_ u \_\_\_\_ 1 \_\_\_\_ \to b$$

Theorem

TQBF ist PSPACE-vollständig.

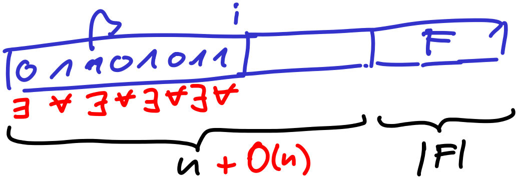
Beweis (Skizze)

1. TQBF $\in$ PSPACE:

Werte die Formel $F(x_1, \ldots, x_n)$ rekursiv für alle Variablenbelegungen β aus ($O(n + |F|)$ Platz)

```
TQBF($i$):
  IF $i \leq n$ THEN
    $\beta(x_i) := 0$;    $a :=$ TQBF($i + 1$);
    $\beta(x_i) := 1$;    $b :=$ TQBF($i + 1$);
    IF $Q_i = \exists$ THEN
      RETURN $a \vee b$;
    ELSE RETURN $a \wedge b$;
```

TQBF ist PSPACE-vollständig



Theorem

TQBF ist PSPACE-vollständig.

Beweis (Skizze)

1. TQBF ∈ PSPACE:

Werte die Formel $F(x_1, \ldots, x_n)$ rekursiv für alle Variablenbelegungen β aus ($\underline{O(n + |F|)}$ Platz)

poly(n+|F|)

```
TQBF($i$):
  IF $i \leq n$ THEN
    $\beta(x_i) := 0$;   $a :=$ TQBF($i+1$);
    $\beta(x_i) := 1$;   $b :=$ TQBF($i+1$);
    IF $Q_i = \exists$ THEN
      RETURN $a \vee b$;
    ELSE RETURN $a \wedge b$;
  ELSE RETURN $\beta(F)$
```

$\text{platz}(i) \in O(\log n)$

$\text{platz}(a) = 1$

$\text{platz}(b) = 1$

$\text{platz}(\beta) = O(n)$

$\text{platz}(F) = O(|F|)$

Σ: poly$(n+|F|)$

TQBF ist PSPACE-vollständig

Theorem

TQBF ist PSPACE-vollständig.

Beweis (Skizze)

1. $\mathrm{TQBF} \in$ PSPACE:

Werte die Formel $F(x_1, \ldots, x_n)$ rekursiv für alle Variablenbelegungen β aus ($O(n + |F|)$ Platz)

```
TQBF($i$):
  IF $i \leq n$ THEN
    $\beta(x_i) := 0$;    $a :=$ TQBF($i + 1$);
    $\beta(x_i) := 1$;    $b :=$ TQBF($i + 1$);
    IF $Q_i = \exists$ THEN
      RETURN $a \vee b$;
    ELSE RETURN $a \wedge b$;
  ELSE RETURN $\beta(F)$
```

2. TQBF ist PSPACE-schwer: Ähnlich wie im Satz von Cook/Levin. ■

PSPACE und Spiele

PSPACE-vollständige Probleme haben oft Bezug zur Frage nach der Existenz von Gewinnstrategien für Spiele mit zwei Spielerinnen (z.B. Schach oder Go).

PSPACE und Spiele

PSPACE-vollständige Probleme haben oft Bezug zur Frage nach der Existenz von Gewinnstrategien für Spiele mit zwei Spielerinnen (z.B. Schach oder Go).

Szenario:

- „$\forall$-Spielerin“ wählt Belegung für $\forall$-quantifizierte Variablen.

PSPACE und Spiele

PSPACE-vollständige Probleme haben oft Bezug zur Frage nach der Existenz von Gewinnstrategien für Spiele mit zwei Spielerinnen (z.B. Schach oder Go).

Szenario:

- „$\forall$-Spielerin“ wählt Belegung für $\forall$-quantifizierte Variablen.
- „$\exists$-Spielerin“ wählt Belegung für $\exists$-quantifizierte Variablen.

PSPACE und Spiele

PSPACE-vollständige Probleme haben oft Bezug zur Frage nach der Existenz von Gewinnstrategien für Spiele mit zwei Spielerinnen (z.B. Schach oder Go).

Szenario:

- „$\forall$-Spielerin“ wählt Belegung für $\forall$-quantifizierte Variablen.
- „$\exists$-Spielerin“ wählt Belegung für $\exists$-quantifizierte Variablen.
- $\exists$/$\forall$-Spielerin gewinnt, wenn die Formel F wahr/falsch wird.

PSPACE und Spiele

PSPACE-vollständige Probleme haben oft Bezug zur Frage nach der Existenz von Gewinnstrategien für Spiele mit zwei Spielerinnen (z.B. Schach oder Go).

Szenario:

- „∀-Spielerin“ wählt Belegung für ∀-quantifizierte Variablen.
- „∃-Spielerin“ wählt Belegung für ∃-quantifizierte Variablen.
- ∃/∀-Spielerin gewinnt, wenn die Formel F wahr/falsch wird.

Frage: Existiert eine Gewinnstrategie für ∃-Spielerin?

„existiert ein Zug sodass egal was die Andere tut ich einen Zug habe des Gewinnt“

$$\boxed{\exists_{x_1} \forall_{x_2} \exists_{x_3} \varphi(x_1, x_2, x_3)}$$

Ein Geographiespiel

Eingabe: Menge von Hauptstadtnamen. , Initialer Buchstabe

Spielregeln:

1. zwei Spielerinnen nennen abwechselnd eine Hauptstadt
2. ✱ jede genannte Hauptstadt muss mit dem letzten Buchstaben der Zuvorgenannten beginnen
3. wer als erstes keine Hauptstadt mehr nennen kann, verliert
4. keine Mehrfachnennungen

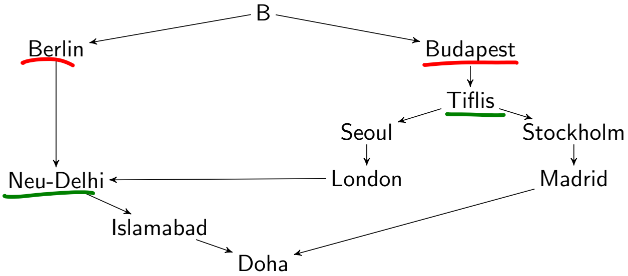
Ein Geographiespiel

Eingabe: Menge von Hauptstadtnamen.

Spielregeln:

1. zwei Spielerinnen nennen abwechselnd eine Hauptstadt
2. jede genannte Hauptstadt muss mit dem letzten Buchstaben der Zuvorgenannten beginnen
3. wer als erstes keine Hauptstadt mehr nennen kann, verliert
4. keine Mehrfachnennungen

Beispiel



Ein Geographiespiel

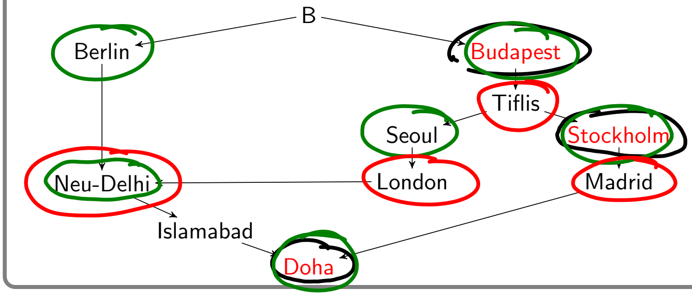
Eingabe: Menge von Hauptstadtnamen.

Spielregeln:

1. zwei Spielerinnen nennen abwechselnd eine Hauptstadt
2. jede genannte Hauptstadt muss mit dem letzten Buchstaben der Zuvorgenannten beginnen
3. wer als erstes keine Hauptstadt mehr nennen kann, verliert
4. keine Mehrfachnennungen

Beispiel

TQBF



Generalisiertes Geographiespiel

Eingabe: Ein gerichteter Graph $\underline{G = (V, E)}$ und ein Startknoten $\underline{v \in V}$.

Generalisiertes Geographiespiel

Eingabe: Ein gerichteter Graph $G = (V, E)$ und ein Startknoten $v \in V$.

Spielregeln:

1. Spielerinnen wählen abwechelnd den “nächsten Knoten” unter den Nachfolgern des aktuellen Knotens
2. wer keinen Nachbarknoten mehr auswählen kann verliert

Generalisiertes Geographiespiel

Eingabe: Ein gerichteter Graph $G = (V, E)$ und ein Startknoten $v \in V$.

Spielregeln:

1. Spielerinnen wählen abwechelnd den "nächsten Knoten" unter den Nachfolgern des aktuellen Knotens
2. wer keinen Nachbarknoten mehr auswählen kann verliert

Spielerin 1 hat "Gewinnstrategie" $\rightsquigarrow$ Ähnlichkeit zu TQBF

Generalisiertes Geographiespiel

Eingabe: Ein gerichteter Graph $G = (V, E)$ und ein Startknoten $v \in V$.

Spielregeln:

1. Spielerinnen wählen abwechelnd den “nächsten Knoten” unter den Nachfolgern des aktuellen Knotens
2. wer keinen Nachbarknoten mehr auswählen kann verliert

Spielerin 1 hat “Gewinnstrategie” $\rightsquigarrow$ Ähnlichkeit zu TQBF

Generalized Geography

Eingabe: gerichteter Graph $G = (V, E)$ und ein Knoten $v \in V$

Frage: Hat Spielerin 1 eine Gewinnstrategie, die mit einem Nachbarknoten von v startet?

Theorem

GENERALIZED GEOGRAPHY ist PSPACE-vollständig.

Generalisiertes Geographiespiel

Eingabe: Ein gerichteter Graph $G = (V, E)$ und ein Startknoten $v \in V$.

Spielregeln:

1. Spielerinnen wählen abwechelnd den “nächsten Knoten” unter den Nachfolgern des aktuellen Knotens
2. wer keinen Nachbarknoten mehr auswählen kann verliert

Spielerin 1 hat “Gewinnstrategie” $\rightsquigarrow$ Ähnlichkeit zu TQBF

Generalized Geography

Eingabe: gerichteter Graph $G = (V, E)$ und ein Knoten $v \in V$

Frage: Hat Spielerin 1 eine Gewinnstrategie, die mit einem Nachbarknoten von v startet?

Theorem

GENERALIZED GEOGRAPHY ist PSPACE-vollständig.

1. GENERALIZED GEOGRAPHY $\in$ PSPACE: Einfach.

Generalisiertes Geographiespiel

Eingabe: Ein gerichteter Graph $G = (V, E)$ und ein Startknoten $v \in V$.

Spielregeln:

1. Spielerinnen wählen abwechelnd den “nächsten Knoten” unter den Nachfolgern des aktuellen Knotens
2. wer keinen Nachbarknoten mehr auswählen kann verliert

Spielerin 1 hat “Gewinnstrategie” $\rightsquigarrow$ Ähnlichkeit zu TQBF

Generalized Geography

Eingabe: gerichteter Graph $G = (V, E)$ und ein Knoten $v \in V$

Frage: Hat Spielerin 1 eine Gewinnstrategie, die mit einem Nachbarknoten von v startet?

Theorem

Generalized Geography ist PSPACE-vollständig.

1. Generalized Geography $\in$ PSPACE: Einfach.
2. Generalized Geography ist PSPACE-vollständig:

zeige TQBF $\leq_m^p$ Generalized Geography.

Abschließende Bemerkungen zu PSPACE

Lemmings

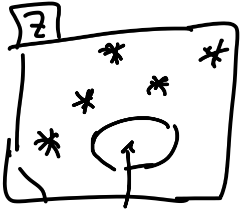
Mitteilung: Für viele Spiele, wie z.B. Schach, Go oder Dame, existieren verallgemeinerte Versionen (auf $n \times n$-Spielbrettern), die PSPACE-schwer sind.

Abschließende Bemerkungen zu PSPACE

Mitteilung: Für viele Spiele, wie z.B. Schach, Go oder Dame, existieren verallgemeinerte Versionen (auf $n \times n$-Spielbrettern), die PSPACE-schwer sind.

Typische Anwendungsgebiete wo PSPACE-vollständige Probleme auftreten:

- Robotik
  (Roboter spielen gegen ihre Umwelt; „Motion Planning“, „Games against Nature“)



Abschließende Bemerkungen zu PSPACE

Mitteilung: Für viele Spiele, wie z.B. Schach, Go oder Dame, existieren verallgemeinerte Versionen (auf $n \times n$-Spielbrettern), die PSPACE-schwer sind.

Typische Anwendungsgebiete wo PSPACE-vollständige Probleme auftreten:

- Robotik
  (Roboter spielen gegen ihre Umwelt; „Motion Planning“, „Games against Nature“)
- Wortproblem für kontextsensitive Sprachen
  (d.h. für Typ 1-Sprachen in der Chomsky-Hierarchie)

⇝ parser bauen

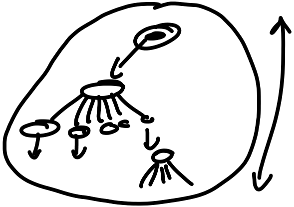
$P \not\subseteq \underline{PSPACE}$

Abschließende Bemerkungen zu PSPACE

Mitteilung: Für viele Spiele, wie z.B. Schach, Go oder Dame, existieren verallgemeinerte Versionen (auf $n \times n$-Spielbrettern), die PSPACE-schwer sind.

Typische Anwendungsgebiete wo PSPACE-vollständige Probleme auftreten:

- Robotik
  (Roboter spielen gegen ihre Umwelt; „Motion Planning“, „Games against Nature“)
- Wortproblem für kontextsensitive Sprachen
  (d.h. für Typ 1-Sprachen in der Chomsky-Hierarchie)



Bemerkungen zu NP und PSPACE:

- NP: kurze Zertifikate
- PSPACE: Zertifikat (Gewinnstrategie) kann exponentiell lang sein

Abschließende Bemerkungen zu PSPACE

Mitteilung: Für viele Spiele, wie z.B. Schach, Go oder Dame, existieren verallgemeinerte Versionen (auf $n \times n$-Spielbrettern), die PSPACE-schwer sind.

Typische Anwendungsgebiete wo PSPACE-vollständige Probleme auftreten:

- Robotik
  (Roboter spielen gegen ihre Umwelt; „Motion Planning“, „Games against Nature“)
- Wortproblem für kontextsensitive Sprachen
  (d.h. für Typ 1-Sprachen in der Chomsky-Hierarchie)

poly (n)

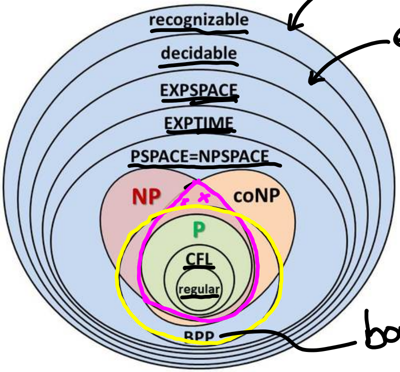
$\boxed{0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 0}$

Bemerkungen zu NP und PSPACE:

- NP: kurze Zertifikate
- PSPACE: Zertifikat (Gewinnstrategie) kann exponentiell lang sein
- $\boxed{\underbrace{\text{PSPACE}}_{\text{DTM}} = \underbrace{\text{NPSPACE}}_{\text{NTM}}}$

mehr als polynomiell viele
nichtdeterministische
Schritte nicht sinnvoll
⇝ kann alle nicht-
deterministischen
Entscheidungen in
poly(n) Platz durch-
probieren

Eine komplexe Welt



semi-entscheidbar

entscheidbar

EXP → exponentiell viel

bounded error probabilistic polynomial time

Quelle: http://cse.psu.edu/~sxr48/cmpsc464/Complexity-classes-diagram.jpg

vielleicht Faktorisierung?