

Számításelmélet

12. előadás

előadó: Kolonits Gábor
kolomax@inf.elte.hu

A tárbonyolultság mérésének problémája

Első megközelítésben a tárigény a működés során felhasznált, pontosabban a fejek által meglátogatott cellák száma.

A tárbonyolultság mérésének problémája

Első megközelítésben a tárigény a működés során felhasznált, pontosabban a fejek által meglátogatott cellák száma.

Probléma: Hiába "takarékoskodik" a felhasznált cellákkal a gép, az input hossza így mindig alsó korlát lesz a tárigényre.

A tárbonyolultság mérésének problémája

Első megközelítésben a tárigény a működés során felhasznált, pontosabban a fejek által meglátogatott cellák száma.

Probléma: Hiába "takarékoskodik" a felhasznált cellákkal a gép, az input hossza így mindig alsó korlát lesz a tárigényre.

Egy megoldási javaslat: Bevezethetjük az többlet tárigény fogalmát, ami az **input tárolására használt cellákon felül** igénybevett cellák száma.

A tárbonyolultság mérésének problémája

Első megközelítésben a tárigény a működés során felhasznált, pontosabban a fejek által meglátogatott cellák száma.

Probléma: Hiába "takarékoskodik" a felhasznált cellákkal a gép, az input hossza így mindig alsó korlát lesz a tárigényre.

Egy megoldási javaslat: Bevezethetjük az többlet tárigény fogalmát, ami az **input tárolására használt cellákon felül** igénybevett cellák száma.

Vannak olyan TG-ek, melyek csak az input területét használják, ám azt akár többször is átírják. Ezt beszámítsuk?

A tárbonyolultság mérésének problémája

Első megközelítésben a tárigény a működés során felhasznált, pontosabban a fejek által meglátogatott cellák száma.

Probléma: Hiába "takarékoskodik" a felhasznált cellákkal a gép, az input hossza így mindig alsó korlát lesz a tárigényre.

Egy megoldási javaslat: Bevezethetjük az többlet tárigény fogalmát, ami az **input tárolására használt cellákon felül** igénybevett cellák száma.

Vannak olyan TG-ek, melyek csak az input területét használják, ám azt akár többször is átírják. Ezt beszámítsuk?

Eldöntési problémáknál beszámítjuk.

Számítási problémáknál viszont ne számítsanak bele a tárigénybe a csak a kimenet előállításához felhasznált cellák.

Az offline Turing gép

Definíció

Az **offline Turing gép** (OTG) egy olyan TG, melynek az első szalagja csak olvasható, a többi írható is. Első szalagját bemeneti szalagnak, további szalagjait munkaszalagoknak nevezzük.

Az offline Turing gép

Definíció

Az **offline Turing gép** (OTG) egy olyan TG, melynek az első szalagja csak olvasható, a többi írható is. Első szalagját bemeneti szalagnak, további szalagjait munkaszalagoknak nevezzük.

Megjegyzés: Egy k munkaszalaggal rendelkező OTG állapotátmenetfüggvénye tehát

$$\delta : (Q \setminus \{q_i, q_n\}) \times \Gamma^{k+1} \rightarrow Q \times \Gamma^k \times \{L, S, R\}^{k+1}.$$

Az offline Turing gép

Definíció

Az **offline Turing gép** (OTG) egy olyan TG, melynek az első szalagja csak olvasható, a többi írható is. Első szalagját bemeneti szalagnak, további szalagjait munkaszalagoknak nevezzük.

Megjegyzés: Egy k munkaszalaggal rendelkező OTG állapotátmenetfüggvénye tehát

$$\delta : (Q \setminus \{q_i, q_n\}) \times \Gamma^{k+1} \rightarrow Q \times \Gamma^k \times \{L, S, R\}^{k+1}.$$

Tétel

Minden TG-hez megadható vele ekvivalens offline TG.

Az offline Turing gép

Definíció

Az **offline Turing gép** (OTG) egy olyan TG, melynek az első szalagja csak olvasható, a többi írható is. Első szalagját bemeneti szalagnak, további szalagjait munkaszalagoknak nevezzük.

Megjegyzés: Egy k munkaszalaggal rendelkező OTG állapotátmenetfüggvénye tehát

$$\delta : (Q \setminus \{q_i, q_n\}) \times \Gamma^{k+1} \rightarrow Q \times \Gamma^k \times \{L, S, R\}^{k+1}.$$

Tétel

Minden TG-hez megadható vele ekvivalens offline TG.

Bizonyítás: Legyen M tetszőleges k szalagos TG. Az M' OTG-nak legyen $k + 1$ szalagja. M' másolja át az inputját a $k + 1$. szalagra és utána működjön úgy a $2 - (k + 1)$. szalagján, mint M . A $k + 1$. szalag felel meg M 1. szalagjának. Ekkor nyilván $L(M') = L(M)$.

Az offline Turing gép

Definíció

Az **offline Turing gép** (OTG) egy olyan TG, melynek az első szalagja csak olvasható, a többi írható is. Első szalagját bemeneti szalagnak, további szalagjait munkaszalagoknak nevezzük.

Megjegyzés: Egy k munkaszalaggal rendelkező OTG állapotátmenetfüggvénye tehát

$$\delta : (Q \setminus \{q_i, q_n\}) \times \Gamma^{k+1} \rightarrow Q \times \Gamma^k \times \{L, S, R\}^{k+1}.$$

Tétel

Minden TG-hez megadható vele ekvivalens offline TG.

Bizonyítás: Legyen M tetszőleges k szalagos TG. Az M' OTG-nak legyen $k + 1$ szalagja. M' másolja át az inputját a $k + 1$. szalagra és utána működjön úgy a $2 - (k + 1)$. szalagján, mint M . A $k + 1$. szalag felel meg M 1. szalagjának. Ekkor nyilván $L(M') = L(M)$.

Megjegyzés: Fordítva is igaz, az offline TG-ek speciális TG-ek.

Offline Turing gép verziók

Definíció

A **nemdeterminisztikus offline Turing gép** (NOTG) egy nemdeterminisztikusan működő offline Turing gép.

Offline Turing gép verziók

Definíció

A **nemdeterminisztikus offline Turing gép** (NOTG) egy nemdeterminisztikusan működő offline Turing gép.

Definíció

A **számító offline Turing gép** olyan legalább 2 szalagos számító Turing gép, amelynek az első szalagja csak olvasható, az utolsó szalagja csak írható. Az első szalagot bemeneti szalagnak, utolsó szalagot kimeneti szalagnak, a többi szalagot munkaszalagnak nevezzük.

Offline Turing gép verziók

Definíció

A **nemdeterminisztikus offline Turing gép** (NOTG) egy nemdeterminisztikusan működő offline Turing gép.

Definíció

A **számító offline Turing gép** olyan legalább 2 szalagos számító Turing gép, amelynek az első szalagja csak olvasható, az utolsó szalagja csak írható. Az első szalagot bemeneti szalagnak, utolsó szalagot kimeneti szalagnak, a többi szalagot munkaszalagnak nevezzük.

Megjegyzés: Egy $k + 2$ szalagos, azaz k munkaszalaggal rendelkező OTG állaptotátmenetfüggvénye tehát

$$\delta : (Q \setminus \{q_i, q_n\}) \times \Gamma^{k+1} \rightarrow Q \times \Gamma^{k+1} \times \{L, S, R\}^{k+2}.$$

A bal oldalon a Γ^{k+1} az $1 - (k + 1)$. szalagoknak, a jobboldalon $2 - (k + 2)$. szalagoknak felel meg.

Az offline Turing gépek tárigénye

Definíció

Egy offline TG **többlet tárigénye** egy adott inputra azon celláknak a száma, amelyeken a működés során valamelyik munkaszalag feje járt.

Az offline Turing gépek tárigénye

Definíció

Egy offline TG **többlet tárigénye** egy adott inputra azon celláknak a száma, amelyeken a működés során valamelyik munkaszalag feje járt.

Egy offline TG $f(n)$ **többlet tárkorlátos**, ha bármely u inputra legfeljebb $f(|u|)$ a többlet tárigénye.

Az offline Turing gépek tárigénye

Definíció

Egy offline TG **többlet tárigénye** egy adott inputra azon celláknak a száma, amelyeken a működés során valamelyik munkaszalag feje járt.

Egy offline TG $f(n)$ **többlet tárkorlátos**, ha bármely u inputra legfeljebb $f(|u|)$ a többlet tárigénye.

Számító OTG-re hasonlóan.

Az offline Turing gépek tárigénye

Definíció

Egy offline TG **többlet tárigénye** egy adott inputra azon celláknak a száma, amelyeken a működés során valamelyik munkaszalag feje járt.

Egy offline TG $f(n)$ **többlet tárkorlátos**, ha bármely u inputra legfeljebb $f(|u|)$ a többlet tárigénye.

Számító OTG-re hasonlóan.

Definíció

Egy nemdeterminisztikus offline TG **többlet tárigénye** egy adott inputra a legnagyobb többlet tárigényű számításának az többlet tárigénye.

Az offline Turing gépek tárigénye

Definíció

Egy offline TG **többlet tárigénye** egy adott inputra azon celláknak a száma, amelyeken a működés során valamelyik munkaszalag feje járt.

Egy offline TG $f(n)$ **többlet tárkorlátos**, ha bármely u inputra legfeljebb $f(|u|)$ a többlet tárigénye.

Számító OTG-re hasonlóan.

Definíció

Egy nemdeterminisztikus offline TG **többlet tárigénye** egy adott inputra a legnagyobb többlet tárigényű számításának az többlet tárigénye.

Egy nemdeterminisztikus offline TG $f(n)$ **többlet tárkorlátos**, ha bármely u inputra legfeljebb $f(|u|)$ az többlet tárigénye.

Determinisztikus és nemdeterminisztikus tárbonyolultsági osztályok

- ▶ $\text{SPACE}(f(n)) := \{L \mid L \text{ eldönthető } O(f(n)) \text{ többlet tárkorlátos determinisztikus offline TG-pel}\}$

Determinisztikus és nemdeterminisztikus tárbonyolultsági osztályok

- ▶ $\text{SPACE}(f(n)) := \{L \mid L \text{ eldönthető } O(f(n)) \text{ többlet tárkorlátos determinisztikus offline TG-pel}\}$
- ▶ $\text{NSPACE}(f(n)) := \{L \mid L \text{ eldönthető } O(f(n)) \text{ többlet tárkorlátos nemdeterminisztikus offline TG-pel}\}$

Determinisztikus és nemdeterminisztikus tárbonyolultsági osztályok

- ▶ $\text{SPACE}(f(n)) := \{L \mid L \text{ eldönthető } O(f(n)) \text{ többlet tárkorlátos determinisztikus offline TG-pel}\}$
- ▶ $\text{NSPACE}(f(n)) := \{L \mid L \text{ eldönthető } O(f(n)) \text{ többlet tárkorlátos nemdeterminisztikus offline TG-pel}\}$
- ▶ $\text{PSPACE} := \bigcup_{k \geq 1} \text{SPACE}(n^k).$

Determinisztikus és nemdeterminisztikus tárbonyolultsági osztályok

- ▶ $\text{SPACE}(f(n)) := \{L \mid L \text{ eldönthető } O(f(n)) \text{ többlet tárkorlátos determinisztikus offline TG-pel}\}$
- ▶ $\text{NSPACE}(f(n)) := \{L \mid L \text{ eldönthető } O(f(n)) \text{ többlet tárkorlátos nemdeterminisztikus offline TG-pel}\}$
- ▶ $\text{PSPACE} := \bigcup_{k \geq 1} \text{SPACE}(n^k).$
- ▶ $\text{NPSPACE} := \bigcup_{k \geq 1} \text{NSPACE}(n^k).$

Determinisztikus és nemdeterminisztikus tárbonyolultsági osztályok

- ▶ $\text{SPACE}(f(n)) := \{L \mid L \text{ eldönthető } O(f(n)) \text{ többlet tárkorlátos determinisztikus offline TG-pel}\}$
- ▶ $\text{NSPACE}(f(n)) := \{L \mid L \text{ eldönthető } O(f(n)) \text{ többlet tárkorlátos nemdeterminisztikus offline TG-pel}\}$
- ▶ $\text{PSPACE} := \bigcup_{k \geq 1} \text{SPACE}(n^k)$.
- ▶ $\text{NPSPACE} := \bigcup_{k \geq 1} \text{NSPACE}(n^k)$.
- ▶ $L := \text{SPACE}(\log n)$.

Determinisztikus és nemdeterminisztikus tárbonyolultsági osztályok

- ▶ $\text{SPACE}(f(n)) := \{L \mid L \text{ eldönthető } O(f(n)) \text{ többlet tárkorlátos determinisztikus offline TG-pel}\}$
- ▶ $\text{NSPACE}(f(n)) := \{L \mid L \text{ eldönthető } O(f(n)) \text{ többlet tárkorlátos nemdeterminisztikus offline TG-pel}\}$
- ▶ $\text{PSPACE} := \bigcup_{k \geq 1} \text{SPACE}(n^k)$.
- ▶ $\text{NPSPACE} := \bigcup_{k \geq 1} \text{NSPACE}(n^k)$.
- ▶ $\text{L} := \text{SPACE}(\log n)$.
- ▶ $\text{NL} := \text{NSPACE}(\log n)$.

Determinisztikus és nemdeterminisztikus tárbonyolultsági osztályok

- ▶ $\text{SPACE}(f(n)) := \{L \mid L \text{ eldönthető } O(f(n)) \text{ többlet tárkorlátos determinisztikus offline TG-pel}\}$
- ▶ $\text{NSPACE}(f(n)) := \{L \mid L \text{ eldönthető } O(f(n)) \text{ többlet tárkorlátos nemdeterminisztikus offline TG-pel}\}$
- ▶ $\text{PSPACE} := \bigcup_{k \geq 1} \text{SPACE}(n^k)$.
- ▶ $\text{NPSPACE} := \bigcup_{k \geq 1} \text{NSPACE}(n^k)$.
- ▶ $\text{L} := \text{SPACE}(\log n)$.
- ▶ $\text{NL} := \text{NSPACE}(\log n)$.

Megjegyzés: Így tehát az offline TG-pel **szublineáris** (lineáris alatti) tárbonyolultságot is mérhetünk. Legalább lineáris tárigények esetén nem lenne szükség az offline TG fogalmára, használhattuk volna az eredeti TG fogalmat is.

ELÉR determinisztikus tárbonysolultsága

$\text{ELÉR} = \{\langle G, s, t \rangle \mid \text{A } G \text{ irányított gráfban van } s\text{-ből } t\text{-be út}\}.$

ELÉR determinisztikus tárbonysolultsága

$\text{ELÉR} = \{\langle G, s, t \rangle \mid \text{A } G \text{ irányított gráfban van } s\text{-ből } t\text{-be út}\}.$

Algo 2-ből, tudjuk, hogy ELÉR P-ben van (szélességi bejárás).

ELÉR determinisztikus tárbonysolultsága

$\text{ELÉR} = \{\langle G, s, t \rangle \mid \text{A } G \text{ irányított gráfban van } s\text{-ből } t\text{-be út}\}.$
Algo 2-ből, tudjuk, hogy ELÉR P-ben van (szélességi bejárás).

Tétel

$\text{ELÉR} \in \text{TIME}(n^2).$

ELÉR determinisztikus tárnyolultsága

$\text{ELÉR} = \{\langle G, s, t \rangle \mid \text{A } G \text{ irányított gráfban van } s\text{-ből } t\text{-be út}\}.$
Algo 2-ből, tudjuk, hogy ELÉR P-ben van (szélességi bejárás).

Tétel

$\text{ELÉR} \in \text{TIME}(n^2).$

Tétel

$\text{ELÉR} \in \text{SPACE}(\log^2 n).$

ELÉR determinisztikus tárnyolultsága

$\text{ELÉR} = \{\langle G, s, t \rangle \mid \text{A } G \text{ irányított gráfban van } s\text{-ből } t\text{-be út}\}.$
Algo 2-ből, tudjuk, hogy ELÉR P-ben van (szélességi bejárás).

Tétel

$\text{ELÉR} \in \text{TIME}(n^2).$

Tétel

$\text{ELÉR} \in \text{SPACE}(\log^2 n).$

Bizonyítás:

- ▶ Rögzítsük a csúcsok egy tetszőleges sorrendjét.

ELÉR determinisztikus tárkonyolultsága

$\text{ELÉR} = \{\langle G, s, t \rangle \mid \text{A } G \text{ irányított gráfban van } s\text{-ből } t\text{-be út}\}.$
Algo 2-ből, tudjuk, hogy ELÉR P-ben van (szélességi bejárás).

Tétel

$\text{ELÉR} \in \text{TIME}(n^2).$

Tétel

$\text{ELÉR} \in \text{SPACE}(\log^2 n).$

Bizonyítás:

- ▶ Rögzítsük a csúcsok egy tetszőleges sorrendjét.
- ▶ $\text{ÚT}(x, y, i) := \text{igaz}$, ha \exists x -ből y -ba legfeljebb 2^i hosszú út.

ELÉR determinisztikus tárkonyolultsága

$\text{ELÉR} = \{ \langle G, s, t \rangle \mid \text{A } G \text{ irányított gráfban van } s\text{-ből } t\text{-be út} \}$.
Algo 2-ből, tudjuk, hogy ELÉR P-ben van (szélességi bejárás).

Tétel

$\text{ELÉR} \in \text{TIME}(n^2)$.

Tétel

$\text{ELÉR} \in \text{SPACE}(\log^2 n)$.

Bizonyítás:

- ▶ Rögzítsük a csúcsok egy tetszőleges sorrendjét.
- ▶ $\text{ÚT}(x, y, i) := \text{igaz}$, ha \exists x -ből y -ba legfeljebb 2^i hosszú út.
- ▶ s -ből van t -be út G -ben $\iff \text{ÚT}(s, t, \lceil \log_2 n \rceil) = \text{igaz}$.

ELÉR determinisztikus tárkonyolultsága

$\text{ELÉR} = \{ \langle G, s, t \rangle \mid \text{A } G \text{ irányított gráfban van } s\text{-ből } t\text{-be út} \}$.
Algo 2-ből, tudjuk, hogy ELÉR P-ben van (szélességi bejárás).

Tétel

$\text{ELÉR} \in \text{TIME}(n^2)$.

Tétel

$\text{ELÉR} \in \text{SPACE}(\log^2 n)$.

Bizonyítás:

- ▶ Rögzítsük a csúcsok egy tetszőleges sorrendjét.
- ▶ $\text{ÚT}(x, y, i) := \text{igaz}$, ha \exists x -ből y -ba legfeljebb 2^i hosszú út.
- ▶ s -ből van t -be út G -ben $\iff \text{ÚT}(s, t, \lceil \log_2 n \rceil) = \text{igaz}$.
- ▶ $\text{ÚT}(x, y, i) = \text{igaz} \iff \exists z (\text{ÚT}(x, z, i-1) = \text{igaz} \wedge \text{ÚT}(z, y, i-1) = \text{igaz})$.

ELÉR determinisztikus tárkonyolultsága

$\text{ELÉR} = \{ \langle G, s, t \rangle \mid \text{A } G \text{ irányított gráfban van } s\text{-ből } t\text{-be út} \}$.
Algo 2-ből, tudjuk, hogy ELÉR P-ben van (szélességi bejárás).

Tétel

$\text{ELÉR} \in \text{TIME}(n^2)$.

Tétel

$\text{ELÉR} \in \text{SPACE}(\log^2 n)$.

Bizonyítás:

- ▶ Rögzítsük a csúcsok egy tetszőleges sorrendjét.
- ▶ $\text{ÚT}(x, y, i) := \text{igaz}$, ha \exists x -ből y -ba legfeljebb 2^i hosszú út.
- ▶ s -ből van t -be út G -ben $\iff \text{ÚT}(s, t, \lceil \log_2 n \rceil) = \text{igaz}$.
- ▶ $\text{ÚT}(x, y, i) = \text{igaz} \iff \exists z (\text{ÚT}(x, z, i-1) = \text{igaz} \wedge \text{ÚT}(z, y, i-1) = \text{igaz})$.
- ▶ Ez alapján egy rekurzív algoritmust készítünk, melynek persze munkaszalagján tárolnia kell, hogy a felsőbb szinteken milyen (x, y, i) -kre létezik folyamatban lévő hívás.

ELÉR determinisztikus tárbonyolultsága

- ▶ ha $i = 0$, akkor $2^0 = 1$ hosszú út kéne: ez az input alapján megválaszolható

ELÉR determinisztikus tárbonyolultsága

- ▶ ha $i = 0$, akkor $2^0 = 1$ hosszú út kéne: ez az input alapján megválaszolható
- ▶ A munkaszalagon (x, y, i) típusú hármasok egy legfeljebb $\lceil \log_2 n \rceil$ hosszú sorozata áll. A hármasok 3. attribútuma 1-esével csökkenő sorozatot alkot $\lceil \log_2 n \rceil$ -től.

ELÉR determinisztikus tárbonyolultsága

- ▶ ha $i = 0$, akkor $2^0 = 1$ hosszú út kéne: ez az input alapján megválaszolható
- ▶ A munkaszalagon (x, y, i) típusú hármasok egy legfeljebb $\lceil \log_2 n \rceil$ hosszú sorozata áll. A hármasok 3. attribútuma 1-esével csökkenő sorozatot alkot $\lceil \log_2 n \rceil$ -től.
- ▶ Az $\text{ÚT}(x, y, i)$ függvény meghívásakor az utolsó hármas (x, y, i) a munkaszalagon. Az algoritmus felírja az $(x, z, i - 1)$ hármaszt a munkaszalagra az (x, y, i) utáni helyre majd kiszámítja $\text{ÚT}(x, z, i - 1)$ értékét.

ELÉR determinisztikus tárbonyolultsága

- ▶ ha $i = 0$, akkor $2^0 = 1$ hosszú út kéne: ez az input alapján megválaszolható
- ▶ A munkaszalagon (x, y, i) típusú hármasok egy legfeljebb $\lceil \log_2 n \rceil$ hosszú sorozata áll. A hármasok 3. attribútuma 1-esével csökkenő sorozatot alkot $\lceil \log_2 n \rceil$ -től.
- ▶ Az $\text{ÚT}(x, y, i)$ függvény meghívásakor az utolsó hármas (x, y, i) a munkaszalagon. Az algoritmus felírja az $(x, z, i - 1)$ hármaszt a munkaszalagra az (x, y, i) utáni helyre majd kiszámítja $\text{ÚT}(x, z, i - 1)$ értékét.
- ▶ Ha hamis, akkor kitörli $(x, z, i - 1)$ -et és z értékét növeli.

ELÉR determinisztikus tárbonyolultsága

- ▶ ha $i = 0$, akkor $2^0 = 1$ hosszú út kéne: ez az input alapján megválaszolható
- ▶ A munkaszalagon (x, y, i) típusú hármasok egy legfeljebb $\lceil \log_2 n \rceil$ hosszú sorozata áll. A hármasok 3. attribútuma 1-esével csökkenő sorozatot alkot $\lceil \log_2 n \rceil$ -től.
- ▶ Az $\text{ÚT}(x, y, i)$ függvény meghívásakor az utolsó hármas (x, y, i) a munkaszalagon. Az algoritmus felírja az $(x, z, i - 1)$ hármaszt a munkaszalagra az (x, y, i) utáni helyre majd kiszámítja $\text{ÚT}(x, z, i - 1)$ értékét.
- ▶ Ha hamis, akkor kitörli $(x, z, i - 1)$ -et és z értékét növeli.
- ▶ Ha igaz, akkor is kitörli $(x, z, i - 1)$ -et és $(z, y, i - 1)$ -et írja a helyére (y -t tudja az előző (x, y, i) hármasból).
 - Ha $\text{ÚT}(z, y, i - 1)$ igaz, akkor $\text{ÚT}(x, y, i)$ igaz (ezt (x, y, i) és $(z, y, i - 1)$ 2. argumentumának egyezéséből látja)
 - Ha $\text{ÚT}(z, y, i - 1)$ hamis akkor kitörli a $(z, y, i - 1)$ -t és z értékét eggyel növelve $\text{ÚT}(x, z, i - 1)$ -en dolgozik tovább.

ELÉR determinisztikus tárbonyolultsága

- ▶ ha $i = 0$, akkor $2^0 = 1$ hosszú út kéne: ez az input alapján megválaszolható
- ▶ A munkaszalagon (x, y, i) típusú hármasok egy legfeljebb $\lceil \log_2 n \rceil$ hosszú sorozata áll. A hármasok 3. attribútuma 1-esével csökkenő sorozatot alkot $\lceil \log_2 n \rceil$ -től.
- ▶ Az $\text{ÚT}(x, y, i)$ függvény meghívásakor az utolsó hármas (x, y, i) a munkaszalagon. Az algoritmus felírja az $(x, z, i - 1)$ hármaszt a munkaszalagra az (x, y, i) utáni helyre majd kiszámítja $\text{ÚT}(x, z, i - 1)$ értékét.
- ▶ Ha hamis, akkor kitörli $(x, z, i - 1)$ -et és z értékét növeli.
- ▶ Ha igaz, akkor is kitörli $(x, z, i - 1)$ -et és $(z, y, i - 1)$ -et írja a helyére (y -t tudja az előző (x, y, i) hármasból).
 - Ha $\text{ÚT}(z, y, i - 1)$ igaz, akkor $\text{ÚT}(x, y, i)$ igaz (ezt (x, y, i) és $(z, y, i - 1)$ 2. argumentumának egyezéséből látja)
 - Ha $\text{ÚT}(z, y, i - 1)$ hamis akkor kitörli a $(z, y, i - 1)$ -t és z értékét eggyel növelve $\text{ÚT}(x, z, i - 1)$ -en dolgozik tovább.
- ▶ Ha egyik z se volt jó, akkor $\text{ÚT}(x, y, i)$ hamis.

ELÉR determinisztikus tárnyolultsága

A főprogram, tehát $(s, t, \lceil \log_2 n \rceil)$ feírásából és az $\text{ÚT}(s, t, \lceil \log n \rceil)$ függvény meghívásából áll. Pontosan akkor lesz igaz a kimenet, ha t elérhető s -ből.

Az algoritmus a munkaszalagján végig legfeljebb $\lceil \log_2 n \rceil$ darab rendezett hármast tárol.

Egy szám tárolásához legfeljebb a szám adott számrendszer alapú logaritmusá +1 darab számjegy szükséges.

Így a rendezett hármásokból mindvégig $O(\log n)$ van és egyenként $O(\log n)$ hosszúak, így $\text{ELÉR} \in \text{SPACE}(\log^2 n)$.

Konfigurációs gráf, elérhetőségi módszer

Definíció

Egy M NTG G_M **konfigurációs gráfjának** csúcsai M konfigurációi és $(C, C') \in E(G_M) \Leftrightarrow C \vdash_M C'$.

Konfigurációs gráf, elérhetőségi módszer

Definíció

Egy M NTG G_M **konfigurációs gráfjának** csúcsai M konfigurációi és $(C, C') \in E(G_M) \Leftrightarrow C \vdash_M C'$.

Elérhetőségi módszer: az $\text{ELÉR} \in \text{TIME}(n^2)$ vagy $\text{ELÉR} \in \text{SPACE}(\log^2 n)$ tételek valamelyikét alkalmazva a konfigurációs gráfra (vagy annak egy részgráfjára) bonyolultsági osztályok közötti összefüggéseket lehet bizonyítani.

Lássunk erre egy példát!

Savitch tétele

Savitch tétele

Ha $f(n) \geq \log n$, akkor $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$.

Savitch tétele

Savitch tétele

Ha $f(n) \geq \log n$, akkor $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$.

Bizonyítás: Legyen M egy $f(n)$ tárigényű NOTG és w az M egy n hosszú bemenete.

Savitch tétele

Savitch tétele

Ha $f(n) \geq \log n$, akkor $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$.

Bizonyítás: Legyen M egy $f(n)$ tárigényű NOTG és w az M egy n hosszú bemenete. Kell egy vele ekvivalens, $O(f^2(n))$ táras OTG.

Savitch tétele

Savitch tétele

Ha $f(n) \geq \log n$, akkor $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$.

Bizonyítás: Legyen M egy $f(n)$ tárigényű NOTG és w az M egy n hosszú bemenete. Kell egy vele ekvivalens, $O(f^2(n))$ táras OTG. M egy konfigurációját $O(f(n) + \log n)$ tárral eltárolhatjuk (aktuális állapot, a munkaszalagok tartalma, fejek pozíciója, az első szalag fejének pozíciója n féle lehet, ezért $\geq \log n$ tár kell ennek eltárolásához).

Savitch tétele

Savitch tétele

Ha $f(n) \geq \log n$, akkor $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$.

Bizonyítás: Legyen M egy $f(n)$ tárigényű NOTG és w az M egy n hosszú bemenete. Kell egy vele ekvivalens, $O(f^2(n))$ táras OTG. M egy konfigurációját $O(f(n) + \log n)$ tárral eltárolhatjuk (aktuális állapot, a munkaszalagok tartalma, fejek pozíciója, az első szalag fejének pozíciója n féle lehet, ezért $\geq \log n$ tár kell ennek eltárolásához). Ha $f(n) \geq \log n$, akkor $O(f(n) + \log n) = O(f(n))$.

Savitch tétele

Savitch tétele

Ha $f(n) \geq \log n$, akkor $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$.

Bizonyítás: Legyen M egy $f(n)$ tárigényű NOTG és w az M egy n hosszú bemenete. Kell egy vele ekvivalens, $O(f^2(n))$ táras OTG. M egy konfigurációját $O(f(n) + \log n)$ tárral eltárolhatjuk (aktuális állapot, a munkaszalagok tartalma, fejek pozíciója, az első szalag fejének pozíciója n féle lehet, ezért $\geq \log n$ tár kell ennek eltárolásához). Ha $f(n) \geq \log n$, akkor $O(f(n) + \log n) = O(f(n))$. Feltehető, hogy M -nek csak egyetlen C_{elf} elfogadó konfigurációja van.

Savitch tétele

Savitch tétele

Ha $f(n) \geq \log n$, akkor $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$.

Bizonyítás: Legyen M egy $f(n)$ tárigényű NOTG és w az M egy n hosszú bemenete. Kell egy vele ekvivalens, $O(f^2(n))$ táras OTG. M egy konfigurációját $O(f(n) + \log n)$ tárral eltárolhatjuk (aktuális állapot, a munkaszalagok tartalma, fejek pozíciója, az első szalag fejének pozíciója n féle lehet, ezért $\geq \log n$ tár kell ennek eltárolásához). Ha $f(n) \geq \log n$, akkor $O(f(n) + \log n) = O(f(n))$. Feltehető, hogy M -nek csak egyetlen C_{elf} elfogadó konfigurációja van. (Törölje le a TG a munkaszalagjait, mielőtt q_i -be lép!)

Savitch tétele

Savitch tétele

Ha $f(n) \geq \log n$, akkor $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$.

Bizonyítás: Legyen M egy $f(n)$ tárigényű NOTG és w az M egy n hosszú bemenete. Kell egy vele ekvivalens, $O(f^2(n))$ táras OTG. M egy konfigurációját $O(f(n) + \log n)$ tárral eltárolhatjuk (aktuális állapot, a munkaszalagok tartalma, fejek pozíciója, az első szalag fejének pozíciója n féle lehet, ezért $\geq \log n$ tár kell ennek eltárolásához). Ha $f(n) \geq \log n$, akkor $O(f(n) + \log n) = O(f(n))$. Feltehető, hogy M -nek csak egyetlen C_{elf} elfogadó konfigurációja van. (Törölje le a TG a munkaszalagjait, mielőtt q_i -be lép!) A legfeljebb $O(f(n))$ méretű konfigurációkat tartalmazó konfigurációs gráf mérete $2^{d \cdot f(n)}$ valamely $d > 0$ konstansra.

Savitch tétele

Savitch tétele

Ha $f(n) \geq \log n$, akkor $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$.

Bizonyítás: Legyen M egy $f(n)$ tárigényű NOTG és w az M egy n hosszú bemenete. Kell egy vele ekvivalens, $O(f^2(n))$ táras OTG. M egy konfigurációját $O(f(n) + \log n)$ tárral eltárolhatjuk (aktuális állapot, a munkaszalagok tartalma, fejek pozíciója, az első szalag fejének pozíciója n féle lehet, ezért $\geq \log n$ tár kell ennek eltárolásához). Ha $f(n) \geq \log n$, akkor $O(f(n) + \log n) = O(f(n))$. Feltehető, hogy M -nek csak egyetlen C_{elf} elfogadó konfigurációja van. (Törölje le a TG a munkaszalagjait, mielőtt q_i -be lép!) A legfeljebb $O(f(n))$ méretű konfigurációkat tartalmazó konfigurációs gráf mérete $2^{d \cdot f(n)}$ valamely $d > 0$ konstansra. Így az előző tétel szerint van olyan M' determinisztikus OTG, ami $O(\log^2(2^{df(n)})) = O(f^2(n))$ tárral el tudja dönteni, hogy a kezdőkonfigurációból elérhető-e C_{elf} .

Savitch tétele

Savitch tétele

Ha $f(n) \geq \log n$, akkor $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$.

Bizonyítás: Legyen M egy $f(n)$ tárigényű NOTG és w az M egy n hosszú bemenete. Kell egy vele ekvivalens, $O(f^2(n))$ táras OTG. M egy konfigurációját $O(f(n) + \log n)$ tárral eltárolhatjuk (aktuális állapot, a munkaszalagok tartalma, fejek pozíciója, az első szalag fejének pozíciója n féle lehet, ezért $\geq \log n$ tár kell ennek eltárolásához). Ha $f(n) \geq \log n$, akkor $O(f(n) + \log n) = O(f(n))$. Feltehető, hogy M -nek csak egyetlen C_{elf} elfogadó konfigurációja van. (Törölje le a TG a munkaszalagjait, mielőtt q_i -be lép!) A legfeljebb $O(f(n))$ méretű konfigurációkat tartalmazó konfigurációs gráf mérete $2^{d \cdot f(n)}$ valamely $d > 0$ konstansra. Így az előző tétel szerint van olyan M' determinisztikus OTG, ami $O(\log^2(2^{df(n)})) = O(f^2(n))$ tárral el tudja dönteni, hogy a kezdőkonfigurációból elérhető-e C_{elf} . M' lépjen pontosan ekkor az elfogadó állapotába, így $L(M') = L(M)$.

Determinisztikus/nemdeterminisztikus polinom tár

Következmény

$PSPACE = NPSPACE$

Determinisztikus/nemdeterminisztikus polinom tár

Következmény

PSPACE = NPSPACE

Bizonyítás: $L \in \text{NSPACE}(n^k) \xRightarrow{\text{Savitch}} L \in \text{SPACE}(n^{2k})$.

Determinisztikus/nemdeterminisztikus polinom tár

Következmény

$PSPACE = NPSpace$

Bizonyítás: $L \in NSPACE(n^k) \xRightarrow{\text{Savitch}} L \in SPACE(n^{2k})$.

Tétel

$NL \subseteq P$

Determinisztikus/nemdeterminisztikus polinom tár

Következmény

$PSPACE = NPSPACE$

Bizonyítás: $L \in NSPACE(n^k) \xRightarrow{\text{Savitch}} L \in SPACE(n^{2k})$.

Tétel

$NL \subseteq P$

Bizonyítás

Legyen $L \in NL$ és M L -et $f(n) = O(\log n)$ tárral eldöntő NOTG.

Determinisztikus/nemdeterminisztikus polinom tár

Következmény

$PSPACE = NPSPACE$

Bizonyítás: $L \in NSPACE(n^k) \xRightarrow{\text{Savitch}} L \in SPACE(n^{2k})$.

Tétel

$NL \subseteq P$

Bizonyítás

Legyen $L \in NL$ és M L -et $f(n) = O(\log n)$ tárral eldöntő NOTG. Meggondolható, hogy egy n méretű inputra M legfeljebb $f(n)$ méretű szalagtartalmakat tartalmazó konfigurációinak a száma legfeljebb $cnd^{\log n}$ alkalmas c, d konstansokkal, ami egy $p(n)$ polinommal felülről becsülhető.

Determinisztikus/nemdeterminisztikus polinom tár

Következmény

$PSPACE = NPSPACE$

Bizonyítás: $L \in NSPACE(n^k) \xRightarrow{\text{Savitch}} L \in SPACE(n^{2k})$.

Tétel

$NL \subseteq P$

Bizonyítás

Legyen $L \in NL$ és M L -et $f(n) = O(\log n)$ tárral eldöntő NOTG. Meggondolható, hogy egy n méretű inputra M legfeljebb $f(n)$ méretű szalagtartalmakat tartalmazó konfigurációinak a száma legfeljebb $cnd^{\log n}$ alkalmas c, d konstansokkal, ami egy $p(n)$ polinommal felülről becsülhető. Így a G konfigurációs gráfnak legfeljebb $p(n)$ csúcsa van. G polinom időben megkonstruálható.

Determinisztikus/nemdeterminisztikus polinom tár

Következmény

$PSPACE = NPSPACE$

Bizonyítás: $L \in NSPACE(n^k) \xRightarrow{\text{Savitch}} L \in SPACE(n^{2k})$.

Tétel

$NL \subseteq P$

Bizonyítás

Legyen $L \in NL$ és M L -et $f(n) = O(\log n)$ tárral eldöntő NOTG. Meggondolható, hogy egy n méretű inputra M legfeljebb $f(n)$ méretű szalagtartalmakat tartalmazó konfigurációinak a száma legfeljebb $cnd^{\log n}$ alkalmas c, d konstansokkal, ami egy $p(n)$ polinommal felülről becsülhető. Így a G konfigurációs gráfnak legfeljebb $p(n)$ csúcsa van. G polinom időben megkonstruálható. Feltehető, hogy G -ben egyetlen elfogadó konfiguráció van. G -ben a kezdőkonfigurációból az elfogadó konfiguráció elérhetősége $O(p^2(n))$ idejű determinisztikus TG-pel eldönthető, azaz $L \in P$.

ELÉR eldöntése nemdeterminisztikus log. tárral

ELÉR fontos szerepet tölt be az $L \stackrel{?}{=} NL$ kérdés vizsgálatában is.

Tétel

$ELÉR \in NL$

ELÉR eldöntése nemdeterminisztikus log. tárral

ELÉR fontos szerepet tölt be az $L \stackrel{?}{=} NL$ kérdés vizsgálatában is.

Tétel

$ELÉR \in NL$

Bizonyítás: Az M 3-szalagos NOTG a (G, s, t) inputra $(n = |V(G)|)$ a következőt teszi:

- ▶ ráírja s -t a második szalagra

ELÉR eldöntése nemdeterminisztikus log. tárral

ELÉR fontos szerepet tölt be az $L \stackrel{?}{=} NL$ kérdés vizsgálatában is.

Tétel

$ELÉR \in NL$

Bizonyítás: Az M 3-szalagos NOTG a (G, s, t) inputra $(n = |V(G)|)$ a következőt teszi:

- ▶ ráírja s -t a második szalagra
- ▶ ráírja a 0 -t a harmadik szalagra

ELÉR eldöntése nemdeterminisztikus log. tárral

ELÉR fontos szerepet tölt be az $L \stackrel{?}{=} NL$ kérdés vizsgálatában is.

Tétel

$ELÉR \in NL$

Bizonyítás: Az M 3-szalagos NOTG a (G, s, t) inputra $(n = |V(G)|)$ a következőt teszi:

- ▶ ráírja s -t a második szalagra
- ▶ ráírja a 0 -t a harmadik szalagra
- ▶ Amíg a harmadik szalagon n -nél kisebb szám áll

ELÉR eldöntése nemdeterminisztikus log. tárral

ELÉR fontos szerepet tölt be az $L \stackrel{?}{=} NL$ kérdés vizsgálatában is.

Tétel

$ELÉR \in NL$

Bizonyítás: Az M 3-szalagos NOTG a (G, s, t) inputra ($n = |V(G)|$) a következőt teszi:

- ▶ ráírja s -t a második szalagra
- ▶ ráírja a 0 -t a harmadik szalagra
- ▶ Amíg a harmadik szalagon n -nél kisebb szám áll
 - Legyen u a második szalagon lévő csúc

ELÉR eldöntése nemdeterminisztikus log. tárral

ELÉR fontos szerepet tölt be az $L \stackrel{?}{=} NL$ kérdés vizsgálatában is.

Tétel

$ELÉR \in NL$

Bizonyítás: Az M 3-szalagos NOTG a (G, s, t) inputra ($n = |V(G)|$) a következőt teszi:

- ▶ ráírja s -t a második szalagra
- ▶ ráírja a 0-t a harmadik szalagra
- ▶ Amíg a harmadik szalagon n -nél kisebb szám áll
 - Legyen u a második szalagon lévő csúc
 - Nemdeterminisztikusan kiválasztja v egy ki-szomszédját és felírja u helyére a második szalagra

ELÉR eldöntése nemdeterminisztikus log. tárral

ELÉR fontos szerepet tölt be az $L \stackrel{?}{=} NL$ kérdés vizsgálatában is.

Tétel

$ELÉR \in NL$

Bizonyítás: Az M 3-szalagos NOTG a (G, s, t) inputra $(n = |V(G)|)$ a következőt teszi:

- ▶ ráírja s -t a második szalagra
- ▶ ráírja a 0-t a harmadik szalagra
- ▶ Amíg a harmadik szalagon n -nél kisebb szám áll
 - Legyen u a második szalagon lévő csúc
 - Nemdeterminisztikusan kiválasztja v egy ki-szomszédját és felírja u helyére a második szalagra
 - Ha $v = t$, akkor elfogadja a bemenetet, egyébként növeli a harmadik szalagon lévő számot (binárisan) eggyel

ELÉR eldöntése nemdeterminisztikus log. tárral

ELÉR fontos szerepet tölt be az $L \stackrel{?}{=} NL$ kérdés vizsgálatában is.

Tétel

$ELÉR \in NL$

Bizonyítás: Az M 3-szalagos NOTG a (G, s, t) inputra $(n = |V(G)|)$ a következőt teszi:

- ▶ ráírja s -t a második szalagra
- ▶ ráírja a 0-t a harmadik szalagra
- ▶ Amíg a harmadik szalagon n -nél kisebb szám áll
 - Legyen u a második szalagon lévő csúc
 - Nemdeterminisztikusan kiválasztja v egy ki-szomszédját és felírja u helyére a második szalagra
 - Ha $v = t$, akkor elfogadja a bemenetet, egyébként növeli a harmadik szalagon lévő számot (binárisan) eggyel
- ▶ Ha n -nél nagyobb szám áll a 3. szalagon, akkor elutasítja a bemenetet.

ELÉR eldöntése nemdeterminisztikus log. tárral

ELÉR fontos szerepet tölt be az $L \stackrel{?}{=} NL$ kérdés vizsgálatában is.

Tétel

$ELÉR \in NL$

Bizonyítás: Az M 3-szalagos NOTG a (G, s, t) inputra $(n = |V(G)|)$ a következőt teszi:

- ▶ ráírja s -t a második szalagra
- ▶ ráírja a 0 -t a harmadik szalagra
- ▶ Amíg a harmadik szalagon n -nél kisebb szám áll
 - Legyen u a második szalagon lévő csúc
 - Nemdeterminisztikusan kiválasztja v egy ki-szomszédját és felírja u helyére a második szalagra
 - Ha $v = t$, akkor elfogadja a bemenetet, egyébként növeli a harmadik szalagon lévő számot (binárisan) eggyel
- ▶ Ha n -nél nagyobb szám áll a 3. szalagon, akkor elutasítja a bemenetet.

Mindkét szalag tartalmát $O(\log n)$ bittel kódolhatjuk.

Logaritmiikus táras visszavezetés, NL-teljesség

Definíció

Egy $L_1 \subseteq \Sigma^*$ nyelv **logaritmiikus tárral visszavezethető** egy $L_2 \subseteq \Delta^*$ nyelvre, ha $L_1 \leq L_2$ és a visszavezetéshez használt függvény kiszámítható logaritmiikus többlet tárkorlátos determinisztikus offline Turing géppel. Jelölése: $L_1 \leq_\ell L_2$.

Logaritmiikus táras visszavezetés, NL-teljesség

Definíció

Egy $L_1 \subseteq \Sigma^*$ nyelv **logaritmiikus tárral visszavezethető** egy $L_2 \subseteq \Delta^*$ nyelvre, ha $L_1 \leq L_2$ és a visszavezetéshez használt függvény kiszámítható logaritmiikus többlet tárkorlátos determinisztikus offline Turing géppel. Jelölése: $L_1 \leq_\ell L_2$.

Definíció

Egy L nyelv **NL-nehéz** (a log. táras visszavezetésre nézve), ha minden $L' \in \text{NL}$ nyelvre, $L' \leq_\ell L$. Ha ezen felül $L \in \text{NL}$ is teljesül, akkor L **NL-teljes** (a log. táras visszavezetésre nézve)

Logaritmiikus táras visszavezetés, NL-teljesség

Definíció

Egy $L_1 \subseteq \Sigma^*$ nyelv **logaritmiikus tárral visszavezethető** egy $L_2 \subseteq \Delta^*$ nyelvre, ha $L_1 \leq L_2$ és a visszavezetéshez használt függvény kiszámítható logaritmiikus többlet tárkorlátos determinisztikus offline Turing géppel. Jelölése: $L_1 \leq_\ell L_2$.

Definíció

Egy L nyelv **NL-nehéz** (a log. táras visszavezetésre nézve), ha minden $L' \in \text{NL}$ nyelvre, $L' \leq_\ell L$. Ha ezen felül $L \in \text{NL}$ is teljesül, akkor L **NL-teljes** (a log. táras visszavezetésre nézve)

Tétel

Az L osztály zárt a logaritmiikus tárral való visszavezetésre nézve.

Logaritmikus táras visszavezetés, NL-teljesség

Definíció

Egy $L_1 \subseteq \Sigma^*$ nyelv **logaritmikus tárral visszavezethető** egy $L_2 \subseteq \Delta^*$ nyelvre, ha $L_1 \leq L_2$ és a visszavezetéshez használt függvény kiszámítható logaritmikus többlet tárkorlátos determinisztikus offline Turing géppel. Jelölése: $L_1 \leq_\ell L_2$.

Definíció

Egy L nyelv **NL-nehéz** (a log. táras visszavezetésre nézve), ha minden $L' \in \text{NL}$ nyelvre, $L' \leq_\ell L$. Ha ezen felül $L \in \text{NL}$ is teljesül, akkor L **NL-teljes** (a log. táras visszavezetésre nézve)

Tétel

Az L osztály zárt a logaritmikus tárral való visszavezetésre nézve.

Bizonyítás: Tegyük fel, hogy $L_1 \leq_\ell L_2$ és $L_2 \in L$.

Logaritmikus táras visszavezetés, NL-teljesség

Definíció

Egy $L_1 \subseteq \Sigma^*$ nyelv **logaritmikus tárral visszavezethető** egy $L_2 \subseteq \Delta^*$ nyelvre, ha $L_1 \leq L_2$ és a visszavezetéshez használt függvény kiszámítható logaritmikus többlet tárkorlátos determinisztikus offline Turing géppel. Jelölése: $L_1 \leq_\ell L_2$.

Definíció

Egy L nyelv **NL-nehéz** (a log. táras visszavezetésre nézve), ha minden $L' \in \text{NL}$ nyelvre, $L' \leq_\ell L$. Ha ezen felül $L \in \text{NL}$ is teljesül, akkor L **NL-teljes** (a log. táras visszavezetésre nézve)

Tétel

Az L osztály zárt a logaritmikus tárral való visszavezetésre nézve.

Bizonyítás: Tegyük fel, hogy $L_1 \leq_\ell L_2$ és $L_2 \in L$.

Legyen M_2 az L_2 -t eldöntő, M pedig a visszavezetésben használt f függvényt kiszámoló logaritmikus táras determinisztikus OTG.

L logaritmikus táras visszavezetésre való zártsága

Az M_1 OTG egy tetszőleges u szóra a következőképpen működik

L logaritmikusan társas visszavezetésre való zártsága

Az M_1 OTG egy tetszőleges u szóra a következőképpen működik

- ▶ A második szalagján egy bináris számlálóval nyomon követi, hogy M_2 feje hányadik betűjét olvassa az $f(u)$ szónak; legyen ez a szám i (kezdetben 1)

L logaritmusos táras visszavezetésre való zártsága

Az M_1 OTG egy tetszőleges u szóra a következőképpen működik

- ▶ A második szalagján egy bináris számlálóval nyomon követi, hogy M_2 feje hányadik betűjét olvassa az $f(u)$ szónak; legyen ez a szám i (kezdetben 1)
- ▶ Amikor M_2 lépne egyet, akkor M_1 az M -et szimulálva előállítja a harmadik szalagon $f(u)$ i -ik betűjét (de csak ezt a betűt!!!)

L logaritmikusan társas visszavezetésre való zártsága

Az M_1 OTG egy tetszőleges u szóra a következőképpen működik

- ▶ A második szalagján egy bináris számlálóval nyomon követi, hogy M_2 feje hányadik betűjét olvassa az $f(u)$ szónak; legyen ez a szám i (kezdetben 1)
- ▶ Amikor M_2 lépne egyet, akkor M_1 az M -et szimulálva előállítja a harmadik szalagon $f(u)$ i -ik betűjét (de csak ezt a betűt!!!)
- ▶ Ezután M_1 szimulálja M_2 aktuális lépését a harmadik szalagon lévő betű felhasználásával és aktualizálja a második szalagon M_2 fejének újabb pozícióját

L logaritmikusan társas visszavezetésre való zártsága

Az M_1 OTG egy tetszőleges u szóra a következőképpen működik

- ▶ A második szalagján egy bináris számlálóval nyomon követi, hogy M_2 feje hányadik betűjét olvassa az $f(u)$ szónak; legyen ez a szám i (kezdetben 1)
- ▶ Amikor M_2 lépne egyet, akkor M_1 az M -et szimulálva előállítja a harmadik szalagon $f(u)$ i -ik betűjét (de csak ezt a betűt!!!)
- ▶ Ezután M_1 szimulálja M_2 aktuális lépését a harmadik szalagon lévő betű felhasználásával és aktualizálja a második szalagon M_2 fejének újabb pozícióját
- ▶ Ha M_2 elfogadó vagy elutasító állapotba lép, akkor M_1 lépjen a saját elfogadó vagy elutasító állapotába, egyébként folytassa a szimulációt a következő lépéssel

L logaritmikusan társas visszavezetésre való zártsága

Az M_1 OTG egy tetszőleges u szóra a következőképpen működik

- ▶ A második szalagján egy bináris számlálóval nyomon követi, hogy M_2 feje hányadik betűjét olvassa az $f(u)$ szónak; legyen ez a szám i (kezdetben 1)
- ▶ Amikor M_2 lépne egyet, akkor M_1 az M -et szimulálva előállítja a harmadik szalagon $f(u)$ i -ik betűjét (de csak ezt a betűt!!!)
- ▶ Ezután M_1 szimulálja M_2 aktuális lépését a harmadik szalagon lévő betű felhasználásával és aktualizálja a második szalagon M_2 fejének újabb pozícióját
- ▶ Ha M_2 elfogadó vagy elutasító állapotba lép, akkor M_1 lépjen a saját elfogadó vagy elutasító állapotába, egyébként folytassa a szimulációt a következő lépéssel

L logaritmikus táras visszavezetésre való zártsága

Az M_1 OTG egy tetszőleges u szóra a következőképpen működik

- ▶ A második szalagján egy bináris számlálóval nyomon követi, hogy M_2 feje hányadik betűjét olvassa az $f(u)$ szónak; legyen ez a szám i (kezdetben 1)
- ▶ Amikor M_2 lépne egyet, akkor M_1 az M -et szimulálva előállítja a harmadik szalagon $f(u)$ i -ik betűjét (de csak ezt a betűt!!!)
- ▶ Ezután M_1 szimulálja M_2 aktuális lépését a harmadik szalagon lévő betű felhasználásával és aktualizálja a második szalagon M_2 fejének újabb pozícióját
- ▶ Ha M_2 elfogadó vagy elutasító állapotba lép, akkor M_1 lépjen a saját elfogadó vagy elutasító állapotába, egyébként folytassa a szimulációt a következő lépéssel

Belátható, hogy M_1 L_1 -et dönti el és a működése során csak logaritmikus méretű tárat használ, azaz $L_1 \in L$.

ELÉR NL-teljessége

Következmény

Ha egy L nyelv NL-teljes és $L \in L$, akkor $L = NL$.

ELÉR NL-teljessége

Következmény

Ha egy L nyelv NL-teljes és $L \in \mathcal{L}$, akkor $L = \text{NL}$.

Bizonyítás: Legyen $L' \in \text{NL}$ tetszőleges, ekkor L NL-teljessége miatt $L' \leq_{\ell} L$. $L \in \mathcal{L}$, így L logaritmikus tárral való visszavezetésre való zártsága miatt $L' \in \mathcal{L}$. Tehát $\text{NL} \subseteq \mathcal{L}$. A másik irány a definíciókból következik.

ELÉR NL-teljessége

Következmény

Ha egy L nyelv NL-teljes és $L \in \mathcal{L}$, akkor $L = \text{NL}$.

Bizonyítás: Legyen $L' \in \text{NL}$ tetszőleges, ekkor L NL-teljessége miatt $L' \leq_{\ell} L$. $L \in \mathcal{L}$, így L logaritmikus tárral való visszavezetésre való zártsága miatt $L' \in \mathcal{L}$. Tehát $\text{NL} \subseteq \mathcal{L}$. A másik irány a definíciókból következik.

Tétel

ELÉR NL-teljes a logaritmikus tárral történő visszavezetésre nézve.

ELÉR NL-teljessége

Következmény

Ha egy L nyelv NL-teljes és $L \in L$, akkor $L = NL$.

Bizonyítás: Legyen $L' \in NL$ tetszőleges, ekkor L NL-teljessége miatt $L' \leq_l L$. $L \in L$, így L logaritmikus tárral való visszavezetésre való zártsága miatt $L' \in L$. Tehát $NL \subseteq L$. A másik irány a definíciókból következik.

Tétel

ELÉR NL-teljes a logaritmikus tárral történő visszavezetésre nézve.

Bizonyítás:

- ▶ Korábban láttuk, hogy $ELÉR \in NL$

ELÉR NL-teljessége

Következmény

Ha egy L nyelv NL-teljes és $L \in L$, akkor $L = NL$.

Bizonyítás: Legyen $L' \in NL$ tetszőleges, ekkor L NL-teljessége miatt $L' \leq_\ell L$. $L \in L$, így L logaritmikus tárral való visszavezetésre való zártsága miatt $L' \in L$. Tehát $NL \subseteq L$. A másik irány a definíciókból következik.

Tétel

ELÉR NL-teljes a logaritmikus tárral történő visszavezetésre nézve.

Bizonyítás:

- ▶ Korábban láttuk, hogy $ELÉR \in NL$
- ▶ Legyen $L \in NL$, megmutatjuk, hogy $L \leq_\ell ELÉR$

ELÉR NL-teljessége

Következmény

Ha egy L nyelv NL-teljes és $L \in L$, akkor $L = NL$.

Bizonyítás: Legyen $L' \in NL$ tetszőleges, ekkor L NL-teljessége miatt $L' \leq_\ell L$. $L \in L$, így L logaritmikus tárral való visszavezetésre való zártsága miatt $L' \in L$. Tehát $NL \subseteq L$. A másik irány a definíciókból következik.

Tétel

ELÉR NL-teljes a logaritmikus tárral történő visszavezetésre nézve.

Bizonyítás:

- ▶ Korábban láttuk, hogy $ELÉR \in NL$
- ▶ Legyen $L \in NL$, megmutatjuk, hogy $L \leq_\ell ELÉR$
- ▶ Legyen M egy L -et eldöntő $O(\log n)$ táras NOTG és $|u| = n$

ELÉR NL-teljessége

Következmény

Ha egy L nyelv NL-teljes és $L \in L$, akkor $L = NL$.

Bizonyítás: Legyen $L' \in NL$ tetszőleges, ekkor L NL-teljessége miatt $L' \leq_\ell L$. $L \in L$, így L logaritmikus tárral való visszavezetésre való zártsága miatt $L' \in L$. Tehát $NL \subseteq L$. A másik irány a definíciókból következik.

Tétel

ELÉR NL-teljes a logaritmikus tárral történő visszavezetésre nézve.

Bizonyítás:

- ▶ Korábban láttuk, hogy $ELÉR \in NL$
- ▶ Legyen $L \in NL$, megmutatjuk, hogy $L \leq_\ell ELÉR$
- ▶ Legyen M egy L -et eldöntő $O(\log n)$ táras NOTG és $|u| = n$
- ▶ Az $O(\log n)$ tárat használó konfigurációk $\leq c \cdot \log n$ hosszúak (alkalmas c -re)

ELÉR NL-teljessége; Immerman-Szelepcsényi

- ▶ A G_M konfigurációs gráfban akkor és csak akkor lehet a kezdőkonfigurációból az elfogadóba jutni (feltehető, hogy csak egy ilyen van), ha $u \in L(M)$. Így $L \leq \text{ELÉR}$.

ELÉR NL-teljessége; Immerman-Szelepcsényi

- ▶ A G_M konfigurációs gráfban akkor és csak akkor lehet a kezdőkonfigurációból az elfogadóba jutni (feltehető, hogy csak egy ilyen van), ha $u \in L(M)$. Így $L \leq \text{ELÉR}$.

Kell még, hogy a visszavezetés log. tárat használ, azaz G_M megkonstruálható egy log. táras N determinisztikus OTG-pel:

ELÉR NL-teljessége; Immerman-Szelepcsényi

- ▶ A G_M konfigurációs gráfban akkor és csak akkor lehet a kezdőkonfigurációból az elfogadóba jutni (feltehető, hogy csak egy ilyen van), ha $u \in L(M)$. Így $L \leq \text{ELÉR}$.

Kell még, hogy a visszavezetés log. tárat használ, azaz G_M megkonstruálható egy log. táras N determinisztikus OTG-pel:

- ▶ N sorolja fel a hossz-lexikografikus rendezés szerint az összes legfeljebb $c \cdot \log n$ hosszú szót az egyik szalagján, majd tesztelje, hogy az legális konfigurációja-e M -nek, ha igen, akkor a szót írja ki a kimenetre

ELÉR NL-teljessége; Immerman-Szelepcsényi

- ▶ A G_M konfigurációs gráfban akkor és csak akkor lehet a kezdőkonfigurációból az elfogadóba jutni (feltehető, hogy csak egy ilyen van), ha $u \in L(M)$. Így $L \leq \text{ELÉR}$.

Kell még, hogy a visszavezetés log. tárat használ, azaz G_M megkonstruálható egy log. táras N determinisztikus OTG-pel:

- ▶ N sorolja fel a hossz-lexikografikus rendezés szerint az összes legfeljebb $c \cdot \log n$ hosszú szót az egyik szalagján, majd tesztelje, hogy az legális konfigurációja-e M -nek, ha igen, akkor a szót írja ki a kimenetre
- ▶ Az élek (konfiguráció párok) hasonlóképpen felsorolhatók, tesztelhetők és a kimenetre írhatók

ELÉR NL-teljessége; Immerman-Szelepcsényi

- ▶ A G_M konfigurációs gráfban akkor és csak akkor lehet a kezdőkonfigurációból az elfogadóba jutni (feltehető, hogy csak egy ilyen van), ha $u \in L(M)$. Így $L \leq \text{ELÉR}$.

Kell még, hogy a visszavezetés log. tárat használ, azaz G_M megkonstruálható egy log. táras N determinisztikus OTG-pel:

- ▶ N sorolja fel a hossz-lexikografikus rendezés szerint az összes legfeljebb $c \cdot \log n$ hosszú szót az egyik szalagján, majd tesztelje, hogy az legális konfigurációja-e M -nek, ha igen, akkor a szót írja ki a kimenetre
- ▶ Az élek (konfiguráció párok) hasonlóképpen felsorolhatók, tesztelhetők és a kimenetre írhatók

Immerman-Szelepcsényi tétel

$\text{NL} = \text{coNL}$

(bizonyítás nélkül)

Hierarchia tétel

$$\text{EXPTIME} := \bigcup_{k \in \mathbb{N}} \text{TIME}(2^{n^k}).$$

Hierarchia tétel

$$\text{EXPTIME} := \bigcup_{k \in \mathbb{N}} \text{TIME}(2^{n^k}).$$

Hierarchia tétel

(I) $\text{NL} \subset \text{PSPACE}$ és $\text{P} \subset \text{EXPTIME}$.

Hierarchia tétel

$$\text{EXPTIME} := \bigcup_{k \in \mathbb{N}} \text{TIME}(2^{n^k}).$$

Hierarchia tétel

- (I) $\text{NL} \subset \text{PSPACE}$ és $\text{P} \subset \text{EXPTIME}$.
- (II) $\text{L} \subseteq \text{NL} = \text{coNL} \subseteq \text{P} \subseteq \text{NP} \subseteq \text{NPSPACE} = \text{PSPACE} \subseteq \text{EXPTIME}$

Hierarchia tétel

$$\text{EXPTIME} := \bigcup_{k \in \mathbb{N}} \text{TIME}(2^{n^k}).$$

Hierarchia tétel

- (I) $\text{NL} \subset \text{PSPACE}$ és $\text{P} \subset \text{EXPTIME}$.
- (II) $\text{L} \subseteq \text{NL} = \text{coNL} \subseteq \text{P} \subseteq \text{NP} \subseteq \text{NPSPACE} = \text{PSPACE} \subseteq \text{EXPTIME}$

Sejtés: A fenti tartalmazási lánc minden tartalmazása valódi.

Hierarchia tétel

(I)-et nem bizonyítjuk.

Hierarchia tétel

(I)-et nem bizonyítjuk.

(II) bizonyítása:

$$L \overset{(1)}{\subseteq} NL \overset{(2)}{=} \text{coNL} \overset{(3)}{\subseteq} P \overset{(4)}{\subseteq} NP \overset{(5)}{\subseteq} \text{NPSPACE} \overset{(6)}{=} \text{PSPACE} \overset{(7)}{\subseteq} \text{EXPTIME}$$

(1) és (4): a nemdeterminisztikusság definíciójából következik

Hierarchia tétel

(I)-et nem bizonyítjuk.

(II) bizonyítása:

$$L \overset{(1)}{\subseteq} NL \overset{(2)}{=} \text{coNL} \overset{(3)}{\subseteq} P \overset{(4)}{\subseteq} NP \overset{(5)}{\subseteq} \text{NPSPACE} \overset{(6)}{=} \text{PSPACE} \overset{(7)}{\subseteq} \text{EXPTIME}$$

(1) és (4): a nemdeterminisztikusság definíciójából következik

(2): Immerman- Szelepcsényi

Hierarchia tétel

(I)-et nem bizonyítjuk.

(II) bizonyítása:

$$L \overset{(1)}{\subseteq} NL \overset{(2)}{=} \text{coNL} \overset{(3)}{\subseteq} P \overset{(4)}{\subseteq} NP \overset{(5)}{\subseteq} \text{NPSPACE} \overset{(6)}{=} \text{PSPACE} \overset{(7)}{\subseteq} \text{EXPTIME}$$

(1) és (4): a nemdeterminisztikusság definíciójából következik

(2): Immerman- Szelepcsényi

(3),(6): előbb bizonyítottuk

Hierarchia tétel

(I)-et nem bizonyítjuk.

(II) bizonyítása:

$$L \overset{(1)}{\subseteq} NL \overset{(2)}{=} \text{coNL} \overset{(3)}{\subseteq} P \overset{(4)}{\subseteq} NP \overset{(5)}{\subseteq} \text{NPSPACE} \overset{(6)}{=} \text{PSPACE} \overset{(7)}{\subseteq} \text{EXPTIME}$$

(1) és (4): a nemdeterminisztikusság definíciójából következik

(2): Immerman- Szelepcsényi

(3),(6): előbb bizonyítottuk

(5): Ha egy NTG egy számítására adott egy időkorlát, akkor ennél a korlátnál több új cellát nincs ideje egyik fejnek sem felfedezni.

Hierarchia tétel

(I)-et nem bizonyítjuk.

(II) bizonyítása:

$$L \overset{(1)}{\subseteq} NL \overset{(2)}{=} \text{coNL} \overset{(3)}{\subseteq} P \overset{(4)}{\subseteq} NP \overset{(5)}{\subseteq} \text{NPSPACE} \overset{(6)}{=} \text{PSPACE} \overset{(7)}{\subseteq} \text{EXPTIME}$$

(1) és (4): a nemdeterminisztikusság definíciójából következik

(2): Immerman- Szelepcsényi

(3),(6): előbb bizonyítottuk

(5): Ha egy NTG egy számítására adott egy időkorlát, akkor ennél a korlátnál több új cellát nincs ideje egyik fejnek sem felfedezni. Így ez az időkorlát egyben tárkorlát is.

Hierarchia tétel

(I)-et nem bizonyítjuk.

(II) bizonyítása:

$$L \overset{(1)}{\subseteq} NL \overset{(2)}{=} coNL \overset{(3)}{\subseteq} P \overset{(4)}{\subseteq} NP \overset{(5)}{\subseteq} NPSPACE \overset{(6)}{=} PSPACE \overset{(7)}{\subseteq} EXPTIME$$

(1) és (4): a nemdeterminisztikusság definíciójából következik

(2): Immerman- Szelepcsényi

(3),(6): előbb bizonyítottuk

(5): Ha egy NTG egy számítására adott egy időkorlát, akkor ennél a korlátnál több új cellát nincs ideje egyik fejnek sem felfedezni. Így ez az időkorlát egyben tárkorlát is.

(7): Elérhetőségi módszerrel: a használt tár méretének exponenciális függvénye a konfigurációs gráf mérete.

Hierarchia tétel

(I)-et nem bizonyítjuk.

(II) bizonyítása:

$$L \overset{(1)}{\subseteq} NL \overset{(2)}{=} coNL \overset{(3)}{\subseteq} P \overset{(4)}{\subseteq} NP \overset{(5)}{\subseteq} NPSPACE \overset{(6)}{=} PSPACE \overset{(7)}{\subseteq} EXPTIME$$

(1) és (4): a nemdeterminisztikusság definíciójából következik

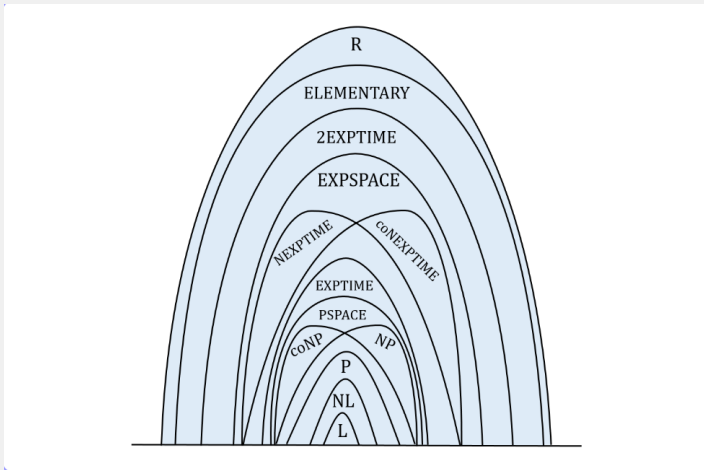
(2): Immerman- Szelepcsényi

(3),(6): előbb bizonyítottuk

(5): Ha egy NTG egy számítására adott egy időkorlát, akkor ennél a korlátnál több új cellát nincs ideje egyik fejnek sem felfedezni. Így ez az időkorlát egyben tárkorlát is.

(7): Elérhetőségi módszerrel: a használt tár méretének exponenciális függvénye a konfigurációs gráf mérete. A konfigurációs gráf méretében négyzetes (azaz összességében a tár méretében exponenciális) időben tudja egy determinisztikus TG az elérhetőséget tesztelni a kezdőkonfigurációból az elfogadó konfigurációba.

R szerkezete



R szerkezete (a tartalmazások valódisága nem mindenütt bizonyított)

[ábra: Gazdag Zs. e-jegyzet]