

Számítási modellek

5. előadás

Boole függvények

Legyenek X és Y logikai változók (0 vagy 1 értékűek). A legfontosabb legfeljebb kétváltozós logikai műveletek:

X	Y	$\neg X$	$X \wedge Y$	$X \vee Y$	$X \rightarrow Y$	$X \otimes Y$
1	1	0	1	1	1	0
1	0	0	0	1	0	1
0	1	1	0	1	1	1
0	0	1	0	0	1	0

Boole függvények

Legyenek X és Y logikai változók (0 vagy 1 értékűek). A legfontosabb legfeljebb kétváltozós logikai műveletek:

X	Y	$\neg X$	$X \wedge Y$	$X \vee Y$	$X \rightarrow Y$	$X \otimes Y$
1	1	0	1	1	1	0
1	0	0	0	1	0	1
0	1	1	0	1	1	1
0	0	1	0	0	1	0

$X \rightarrow Y$ logikailag ekvivalens $\neg X \vee Y$ -nal.

$X \otimes Y$ logikailag ekvivalens $(\neg X \wedge Y) \vee (X \wedge \neg Y)$ -nal.

Boole függvények

Legyenek X és Y logikai változók (0 vagy 1 értékűek). A legfontosabb legfeljebb kétváltozós logikai műveletek:

X	Y	$\neg X$	$X \wedge Y$	$X \vee Y$	$X \rightarrow Y$	$X \otimes Y$
1	1	0	1	1	1	0
1	0	0	0	1	0	1
0	1	1	0	1	1	1
0	0	1	0	0	1	0

$X \rightarrow Y$ logikailag ekvivalens $\neg X \vee Y$ -nal.

$X \otimes Y$ logikailag ekvivalens $(\neg X \wedge Y) \vee (X \wedge \neg Y)$ -nal.

A logikai műveleteket Boole-függvényeknek is szokás nevezni.

Definíció

Boole-függvénynek nevezünk egy $f : \{0, 1\}^n \rightarrow \{0, 1\}$ leképezést.

Boole függvények

Sok algoritmikus probléma valójában megfelel egy Boole-függvény kiszámításának.

Boole függvények

Sok algoritmikus probléma valójában megfelel egy Boole-függvény kiszámításának.

Példa: Ha egy n csúcsú irányítatlan gráfról szeretnénk eldönteni, hogy rendelkezik-e egy tulajdonsággal, akkor az valójában egy $f : \{0, 1\}^N \rightarrow \{0, 1\}$ Boole-függvény kiszámítását jelenti, ahol $N = \binom{n}{2}$.

Boole függvények

Sok algoritmikus probléma valójában megfelel egy Boole-függvény kiszámításának.

Példa: Ha egy n csúcsú irányítatlan gráfról szeretnénk eldönteni, hogy rendelkezik-e egy tulajdonsággal, akkor az valójában egy $f : \{0, 1\}^N \rightarrow \{0, 1\}$ Boole-függvény kiszámítását jelenti, ahol $N = \binom{n}{2}$. A változókat feleltessük meg az N csúcspárnak, és egy változó értéke akkor legyen 1, ha a gráfban a két csúc között van él. Így a változókiértékelések bijekcióba állíthatók a lehetséges gráfokkal.

Boole függvények

Sok algoritmikus probléma valójában megfelel egy Boole-függvény kiszámításának.

Példa: Ha egy n csúcsú irányítatlan gráfról szeretnénk eldönteni, hogy rendelkezik-e egy tulajdonsággal, akkor az valójában egy $f : \{0, 1\}^N \rightarrow \{0, 1\}$ Boole-függvény kiszámítását jelenti, ahol $N = \binom{n}{2}$. A változókat feleltessük meg az N csúcspárnak, és egy változó értéke akkor legyen 1, ha a gráfban a két csúc között van él. Így a változókiértékelések bijekcióba állíthatók a lehetséges gráfokkal.

Akár NP-teljes problémák is leírhatók így, ha például ez a tulajdonság az, hogy van-e a gráfban Hamilton kör.

Boole-polinomok

Definíció

A konjunkció, diszjunkció és negáció műveleteivel felírt kifejezéseket **Boole-polinomoknak** nevezzük.

Boole-polinomok

Definíció

A konjunkció, diszjunkció és negáció műveleteivel felírt kifejezéseket **Boole-polinomoknak** nevezzük.

Tétel

Minden Boole-függvény kifejezhető Boole-polinomként.

Boole-polinomok

Definíció

A konjunkció, diszjunkció és negáció műveleteivel felírt kifejezéseket **Boole-polinomoknak** nevezzük.

Tétel

Minden Boole-függvény kifejezhető Boole-polinomként.

Bizonyítás: Legyen $f(X_1, \dots, X_n) : \{0, 1\}^n \rightarrow \{0, 1\}$ egy tetszőleges Boole-függvény és vezessük be az $f^1 = \{z \in \{0, 1\}^n \mid f(z) = 1\}$ jelölést.

Boole-polinomok

Definíció

A konjunkció, diszjunkció és negáció műveleteivel felírt kifejezéseket **Boole-polinomoknak** nevezzük.

Tétel

Minden Boole-függvény kifejezhető Boole-polinomként.

Bizonyítás: Legyen $f(X_1, \dots, X_n) : \{0, 1\}^n \rightarrow \{0, 1\}$ egy tetszőleges Boole-függvény és vezessük be az $f^1 = \{z \in \{0, 1\}^n \mid f(z) = 1\}$ jelölést. Ha $z \in \{0, 1\}^n$ akkor legyen

$$\varphi_z(X_1, \dots, X_n) := \bigwedge_{i=1}^n L_i,$$

ahol $L_i = X_i$ ha $z_i = 1$ és $L_i = \neg X_i$ ha $z_i = 0$.

Boole-polinomok

Definíció

A konjunkció, diszjunkció és negáció műveleteivel felírt kifejezéseket **Boole-polinomoknak** nevezzük.

Tétel

Minden Boole-függvény kifejezhető Boole-polinomként.

Bizonyítás: Legyen $f(X_1, \dots, X_n) : \{0, 1\}^n \rightarrow \{0, 1\}$ egy tetszőleges Boole-függvény és vezessük be az $f^1 = \{z \in \{0, 1\}^n \mid f(z) = 1\}$ jelölést. Ha $z \in \{0, 1\}^n$ akkor legyen

$$\varphi_z(X_1, \dots, X_n) := \bigwedge_{i=1}^n L_i,$$

ahol $L_i = X_i$ ha $z_i = 1$ és $L_i = \neg X_i$ ha $z_i = 0$.

Ekkor $\varphi_z(X_1, \dots, X_n) = 1$ akkor és csak akkor, ha $X_i = z_i$ minden $1 \leq i \leq n$ esetén.

Boole-polinomok

Tehát a

$$\psi(X_1, \dots, X_n) = \bigvee_{\mathbf{z} \in f^{-1}} \varphi_{\mathbf{z}}(X_1, \dots, X_n)$$

Boole-polinomra

Boole-polinomok

Tehát a

$$\psi(X_1, \dots, X_n) = \bigvee_{z \in f^1} \varphi_z(X_1, \dots, X_n)$$

Boole-polinomra $\psi(X_1, \dots, X_n) = 1 \iff$ valamilyen $z \in f^1$ -re
 $\varphi_z(X_1, \dots, X_n) = 1$

Boole-polinomok

Tehát a

$$\psi(X_1, \dots, X_n) = \bigvee_{z \in f^1} \varphi_z(X_1, \dots, X_n)$$

Boole-polinomra $\psi(X_1, \dots, X_n) = 1 \iff$ valamely $z \in f^1$ -re
 $\varphi_z(X_1, \dots, X_n) = 1 \iff$ valamely $z \in f^1$ -re $X_i = z_i$ minden
 $1 \leq i \leq n$

Boole-polinomok

Tehát a

$$\psi(X_1, \dots, X_n) = \bigvee_{z \in f^1} \varphi_z(X_1, \dots, X_n)$$

Boole-polinomra $\psi(X_1, \dots, X_n) = 1 \iff$ valamely $z \in f^1$ -re
 $\varphi_z(X_1, \dots, X_n) = 1 \iff$ valamely $z \in f^1$ -re $X_i = z_i$ minden
 $1 \leq i \leq n \iff (X_1, \dots, X_n) \in f^1$

Boole-polinomok

Tehát a

$$\psi(X_1, \dots, X_n) = \bigvee_{z \in f^1} \varphi_z(X_1, \dots, X_n)$$

Boole-polinomra $\psi(X_1, \dots, X_n) = 1 \iff$ valamilyen $z \in f^1$ -re
 $\varphi_z(X_1, \dots, X_n) = 1 \iff$ valamilyen $z \in f^1$ -re $X_i = z_i$ minden
 $1 \leq i \leq n \iff (X_1, \dots, X_n) \in f^1 \iff f(X_1, \dots, X_n) = 1.$ □

Boole-polinomok

Tehát a

$$\psi(X_1, \dots, X_n) = \bigvee_{z \in f^1} \varphi_z(X_1, \dots, X_n)$$

Boole-polinomra $\psi(X_1, \dots, X_n) = 1 \iff$ valamely $z \in f^1$ -re
 $\varphi_z(X_1, \dots, X_n) = 1 \iff$ valamely $z \in f^1$ -re $X_i = z_i$ minden
 $1 \leq i \leq n \iff (X_1, \dots, X_n) \in f^1 \iff f(X_1, \dots, X_n) = 1$. □

Példa:

X_1	X_2	X_3	$f(X_1, X_2, X_3)$	$\psi(X_1, X_2, X_3) =$
1	1	1	0	
1	1	0	1	$(X_1 \wedge X_2 \wedge \neg X_3) \vee$
1	0	1	0	
1	0	0	0	
0	1	1	0	
0	1	0	1	$(\neg X_1 \wedge X_2 \wedge \neg X_3) \vee$
0	0	1	1	$(\neg X_1 \wedge \neg X_2 \wedge X_3) \vee$
0	0	0	1	$(\neg X_1 \wedge \neg X_2 \wedge \neg X_3)$

Topologikus sorrend

Definíció

Legyen $G = (V, E)$ egy irányított gráf. A csúcsaink egy v_1, \dots, v_n sorrendjét **topologikus sorrendnek** nevezzük, ha

$$(v_i, v_j) \in E \Rightarrow i < j.$$

Topologikus sorrend

Definíció

Legyen $G = (V, E)$ egy irányított gráf. A csúcsaink egy v_1, \dots, v_n sorrendjét **topologikus sorrendnek** nevezzük, ha $(v_i, v_j) \in E \Rightarrow i < j$.

Tétel

Egy irányított gráf akkor és csak akkor aciklikus (azaz, irányított kört nem tartalmazó), ha a csúcsainak van topologikus sorrendje.

Topologikus sorrend

Definíció

Legyen $G = (V, E)$ egy irányított gráf. A csúcsaink egy v_1, \dots, v_n sorrendjét **topologikus sorrendnek** nevezzük, ha $(v_i, v_j) \in E \Rightarrow i < j$.

Tétel

Egy irányított gráf akkor és csak akkor aciklikus (azaz, irányított kört nem tartalmazó), ha a csúcsainak van topologikus sorrendje.

A tétel BSc-s tananyag, itt nem bizonyítjuk.

Topologikus sorrend

Definíció

Legyen $G = (V, E)$ egy irányított gráf. A csúcsaink egy v_1, \dots, v_n sorrendjét **topologikus sorrendnek** nevezzük, ha $(v_i, v_j) \in E \Rightarrow i < j$.

Tétel

Egy irányított gráf akkor és csak akkor aciklikus (azaz, irányított kört nem tartalmazó), ha a csúcsainak van topologikus sorrendje.

A tétel BSc-s tananyag, itt nem bizonyítjuk.

Gráfok körmentességének ellenőrzése és egyúttal aciklikusság esetén a csúcsok topologikus sorrendjének megadására hatékony, $O(\text{élszám})$ futási idejű algoritmusok ismeretesek (például a mélységi keresés befejezési száma szerinti csökkenő sorrend jó; egy gráf körmentes a.cs.a. ha a mélységi keresés nem talál visszaélt).

Boole-hálózatok

- ▶ A logikai hálózat logikai műveleteknek megfelelő kapuk hálózata, speciális esete a Boole-hálózat, ahol a kapuk típusa csak \neg , \wedge és \vee lehet.

Boole-hálózatok

- ▶ A logikai hálózat logikai műveleteknek megfelelő kapuk hálózata, speciális esete a Boole-hálózat, ahol a kapuk típusa csak \neg , \wedge és \vee lehet.
- ▶ A digitális áramkör elméleti megfelelője a Boole-hálózat, a modell számítási erejét tekintve a Turing gépekkel ekvivalens.

Boole-hálózatok

- ▶ A logikai hálózat logikai műveleteknek megfelelő kapuk hálózata, speciális esete a Boole-hálózat, ahol a kapuk típusa csak \neg , \wedge és \vee lehet.
- ▶ A digitális áramkör elméleti megfelelője a Boole-hálózat, a modell számítási erejét tekintve a Turing gépekkel ekvivalens.
- ▶ A logikai hálózatok segítségével közvetlenül, jól áttekinthetően leírható az algoritmusok logikai struktúrája.

Boole-hálózatok

- ▶ A logikai hálózat logikai műveleteknek megfelelő kapuk hálózata, speciális esete a Boole-hálózat, ahol a kapuk típusa csak \neg , \wedge és \vee lehet.
- ▶ A digitális áramkör elméleti megfelelője a Boole-hálózat, a modell számítási erejét tekintve a Turing gépekkel ekvivalens.
- ▶ A logikai hálózatok segítségével közvetlenül, jól áttekinthetően leírható az algoritmusok logikai struktúrája.
- ▶ Az algoritmusok logikai hálózatokkal történő leírása ötleteket adhat párhuzamos algoritmusok készítésére.

Boole-hálózatok

- ▶ A logikai hálózat logikai műveleteknek megfelelő kapuk hálózata, speciális esete a Boole-hálózat, ahol a kapuk típusa csak \neg , \wedge és \vee lehet.
- ▶ A digitális áramkör elméleti megfelelője a Boole-hálózat, a modell számítási erejét tekintve a Turing gépekkel ekvivalens.
- ▶ A logikai hálózatok segítségével közvetlenül, jól áttekinthetően leírható az algoritmusok logikai struktúrája.
- ▶ Az algoritmusok logikai hálózatokkal történő leírása ötleteket adhat párhuzamos algoritmusok készítésére.
- ▶ A $P \neq NP$ sejtés támadására potenciálisan alkalmasnak vélt számítási modell.

Boole-hálózatok

- ▶ A logikai hálózat logikai műveleteknek megfelelő kapuk hálózata, speciális esete a Boole-hálózat, ahol a kapuk típusa csak \neg , \wedge és \vee lehet.
- ▶ A digitális áramkör elméleti megfelelője a Boole-hálózat, a modell számítási erejét tekintve a Turing gépekkel ekvivalens.
- ▶ A logikai hálózatok segítségével közvetlenül, jól áttekinthetően leírható az algoritmusok logikai struktúrája.
- ▶ Az algoritmusok logikai hálózatokkal történő leírása ötleteket adhat párhuzamos algoritmusok készítésére.
- ▶ A $P \neq NP$ sejtés támadására potenciálisan alkalmasnak vélt számítási modell.
- ▶ A Cook-Levin tételre (a SAT nyelv NP-teljes) alternatív bizonyítás adható Boole-hálózatok segítségével.

Boole-hálózat

Definíció

Boole-hálózat egy csúcscímkézett, aciklikus irányított gráf, melyre

Boole-hálózat

Definíció

Boole-hálózat egy csúcscímkézett, aciklikus irányított gráf, melyre

- ▶ a források (bemeneti csúcsok) páronként különböző ítéletváltozókkal vannak címkézve,

Boole-hálózat

Definíció

Boole-hálózat egy csúscímkezett, aciklikus irányított gráf, melyre

- ▶ a források (bemeneti csúcsok) páronként különböző ítéletváltozókkal vannak címkézve,
- ▶ a többi csúcs **logikai kapu**, ezek az \neg , \wedge , \vee logikai műveletek valmelyikével vannak címkézve, a \neg kapuk befoka 1, a \wedge és \vee kapuk befoka 2,

Boole-hálózat

Definíció

Boole-hálózat egy csúscsímkezett, aciklikus irányított gráf, melyre

- ▶ a források (bemeneti csúcsok) páronként különböző ítéletváltozókkal vannak címkézve,
- ▶ a többi csúcs **logikai kapu**, ezek az \neg , \wedge , \vee logikai műveletek valamelyikével vannak címkézve, a \neg kapuk befoka 1, a \wedge és \vee kapuk befoka 2,
- ▶ egyetlen 0 kifokú kapu (nyelő) van, ezt **kimeneti kapunak** (vagy kimeneti csúcsnak) nevezzünk.

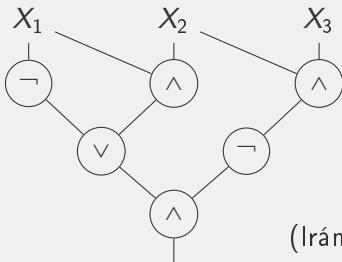
Boole-hálózat

Definíció

Boole-hálózat egy csúcscímkezett, aciklikus irányított gráf, melyre

- ▶ a források (bemeneti csúcsok) páronként különböző ítéletváltozókkal vannak címkézve,
- ▶ a többi csúcs **logikai kapu**, ezek az \neg , \wedge , \vee logikai műveletek valamelyikével vannak címkézve, a \neg kapuk befoka 1, a \wedge és \vee kapuk befoka 2,
- ▶ egyetlen 0 kifokú kapu (nyelő) van, ezt **kimeneti kapunak** (vagy kimeneti csúcsnak) nevezzünk.

Példa:



(Irányítás: ↓)

Boole-hálózatok kiértékelése

Definíció

Legyenek $V(C) = \{X_1, \dots, X_n\}$ a C Boole-hálózat bemeneti csúcsai és $I : V(C) \rightarrow \{0, 1\}$ egy változókiértékelés.

Boole-hálózatok kiértékelése

Definíció

Legyenek $V(C) = \{X_1, \dots, X_n\}$ a C Boole-hálózat bemeneti csúcsai és $I : V(C) \rightarrow \{0, 1\}$ egy változókiértékelés. Tekintsük C egy tetszőleges topologikus sorrendjét.

Boole-hálózatok kiértékelése

Definíció

Legyenek $V(C) = \{X_1, \dots, X_n\}$ a C Boole-hálózat bemeneti csúcsai és $I : V(C) \rightarrow \{0, 1\}$ egy változókiértékelés. Tekintsük C egy tetszőleges topologikus sorrendjét. Ekkor az egyes logikai kapukhoz tartozó Boole-értéket ezen sorrend szerint a kapu típusa alapján a korábbi kapukhoz és bemeneti csúcsok rendelt Boole-értékek alapján meghatározhatjuk.

Boole-hálózatok kiértékelése

Definíció

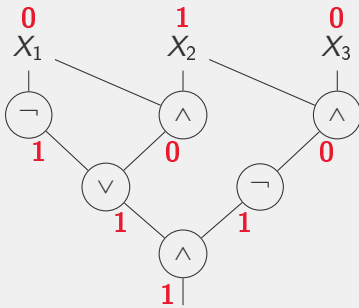
Legyenek $V(C) = \{X_1, \dots, X_n\}$ a C Boole-hálózat bemeneti csúcsai és $I : V(C) \rightarrow \{0, 1\}$ egy változókiértékelés. Tekintsük C egy tetszőleges topologikus sorrendjét. Ekkor az egyes logikai kapukhoz tartozó Boole-értéket ezen sorrend szerint a kapu típusa alapján a korábbi kapukhoz és bemeneti csúcsok rendelt Boole-értékek alapján meghatározhatjuk. A C **Boole-hálózat** $\mathcal{B}_I(C)$ **Boole-értéke** I -ben a kimeneti kapuhoz rendelt Boole-érték.

Boole-hálózatok kiértékelése

Definíció

Legyenek $V(C) = \{X_1, \dots, X_n\}$ a C Boole-hálózat bemeneti csúcsai és $I : V(C) \rightarrow \{0, 1\}$ egy változókiértékelés. Tekintsük C egy tetszőleges topologikus sorrendjét. Ekkor az egyes logikai kapukhoz tartozó Boole-értéket ezen sorrend szerint a kapu típusa alapján a korábbi kapukhoz és bemeneti csúcsok rendelt Boole-értékek alapján meghatározhatjuk. A C **Boole-hálózat** $\mathcal{B}_I(C)$ **Boole-értéke** I -ben a kimeneti kapuhoz rendelt Boole-érték.

Példa:



Boole-hálózat által kiszámított Boole-függvény

Definíció

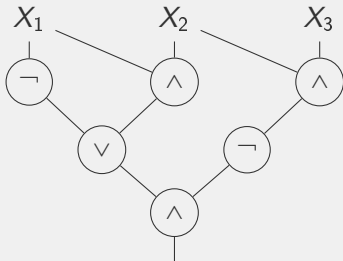
A C Boole-hálózat által kiszámított Boole-függvény alatt az $f : (I(X_1), \dots, I(X_n)) \mapsto \mathcal{B}_I(C)$ függvényt értjük ($I : V(C) \rightarrow \{0, 1\}$ tetszőleges).

Boole-hálózat által kiszámított Boole-függvény

Definíció

A C Boole-hálózat által kiszámított Boole-függvény alatt az $f : (I(X_1), \dots, I(X_n)) \mapsto \mathcal{B}_I(C)$ függvényt értjük ($I : V(C) \rightarrow \{0, 1\}$ tetszőleges).

Példa:



A kiszámított Boole-függvény:

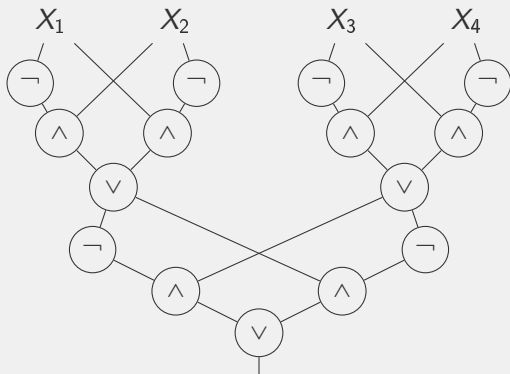
X_1	X_2	X_3	$\mathcal{B}_I(C)$
1	1	1	0
1	1	0	1
1	0	1	0
1	0	0	0
0	1	1	0
0	1	0	1
0	0	1	1
0	0	0	1

Boole-hálózatok – Példa

Példa: A $\text{parity}_n : \{0, 1\}^n \rightarrow \{0, 1\}$ Boole függvény akkor és csak akkor ad 1-et, ha páratlan sok 1-es bemenete van.

Boole-hálózatok – Példa

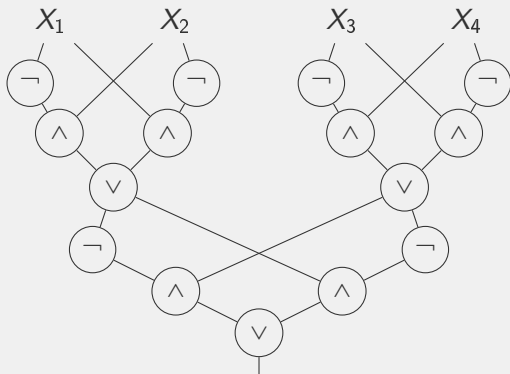
Példa: A $\text{parity}_n : \{0, 1\}^n \rightarrow \{0, 1\}$ Boole függvény akkor és csak akkor ad 1-et, ha páratlan sok 1-es bemenete van. $n = 4$:



$$\text{parity}_1(X_1) = X_1,$$

Boole-hálózatok – Példa

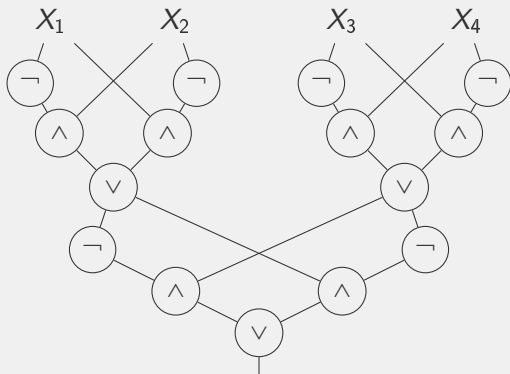
Példa: A $\text{parity}_n : \{0, 1\}^n \rightarrow \{0, 1\}$ Boole függvény akkor és csak akkor ad 1-et, ha páratlan sok 1-es bemenete van. $n = 4$:



$$\text{parity}_1(X_1) = X_1, \quad \text{parity}_2(X_1, X_2) = X_1 \otimes X_2,$$

Boole-hálózatok – Példa

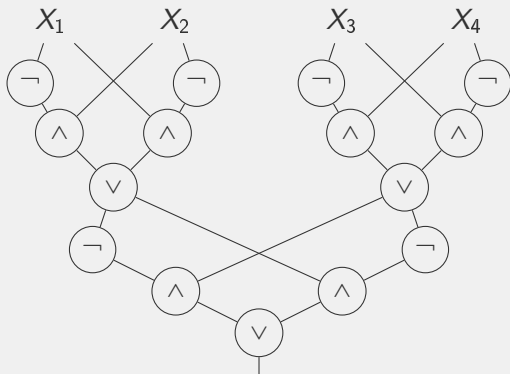
Példa: A $\text{parity}_n : \{0, 1\}^n \rightarrow \{0, 1\}$ Boole függvény akkor és csak akkor ad 1-et, ha páratlan sok 1-es bemenete van. $n = 4$:



$$\begin{aligned}\text{parity}_1(X_1) &= X_1, & \text{parity}_2(X_1, X_2) &= X_1 \otimes X_2, \\ \text{parity}_3(X_1, X_2, X_3) &= X_1 \otimes (X_2 \otimes X_3),\end{aligned}$$

Boole-hálózatok – Példa

Példa: A $\text{parity}_n : \{0, 1\}^n \rightarrow \{0, 1\}$ Boole függvény akkor és csak akkor ad 1-et, ha páratlan sok 1-es bemenete van. $n = 4$:



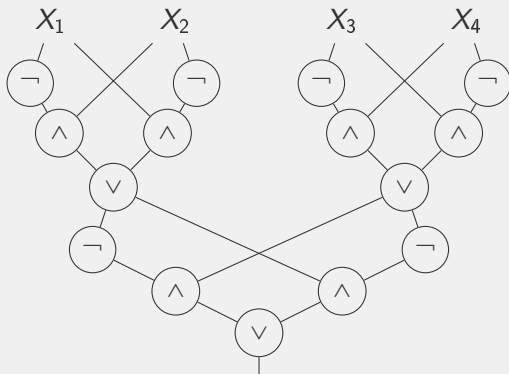
$$\text{parity}_1(X_1) = X_1, \quad \text{parity}_2(X_1, X_2) = X_1 \otimes X_2,$$

$$\text{parity}_3(X_1, X_2, X_3) = X_1 \otimes (X_2 \otimes X_3),$$

$$\text{parity}_4(X_1, X_2, X_3, X_4) = (X_1 \otimes X_2) \otimes (X_3 \otimes X_4),$$

Boole-hálózatok – Példa

Példa: A $\text{parity}_n : \{0, 1\}^n \rightarrow \{0, 1\}$ Boole függvény akkor és csak akkor ad 1-et, ha páratlan sok 1-es bemenete van. $n = 4$:



$$\text{parity}_1(X_1) = X_1, \quad \text{parity}_2(X_1, X_2) = X_1 \otimes X_2,$$

$$\text{parity}_3(X_1, X_2, X_3) = X_1 \otimes (X_2 \otimes X_3),$$

$$\text{parity}_4(X_1, X_2, X_3, X_4) = (X_1 \otimes X_2) \otimes (X_3 \otimes X_4),$$

$$\text{parity}_n(X_1, \dots, X_n) = \text{parity}_{n-2}(X_1, \dots, X_{n-2}) \otimes (X_{n-1} \otimes X_n).$$

Boole-hálózatok – általánosítási lehetőségek

Egy általánosítási lehetőség ha több nyelőt is megengedünk, akkor tetszőleges függvényeket is kiszámíthatunk Boole hálózatok segítségével. Ilyenkor a kimenet egy bitsorozat, melynek a hossza a nyelők száma. Általánosabb függvények kiszámításához kódoljuk át a be- és kimeneteket $\{0, 1\}$ feletti szóvá.

Boole-hálózatok – általánosítási lehetőségek

Egy általánosítási lehetőség ha több nyelőt is megengedünk, akkor tetszőleges függvényeket is kiszámíthatunk Boole hálózatok segítségével. Ilyenkor a kimenet egy bitsorozat, melynek a hossza a nyelők száma. Általánosabb függvények kiszámításához kódoljuk át a be- és kimeneteket $\{0, 1\}$ feletti szóvá.

Példa:

Megkonstruálható egy olyan Boole-hálózat, amelynek a bemeneti változói $X_0, \dots, X_{n-1}, Y_0, \dots, Y_{n-1}$ és az $n + 1$ kimeneti kapuján az $X_{n-1} \cdots X_0$ és $Y_{n-1} \cdots Y_0$ bináris számok összegének a bitjeit számítja ki.

Boole-hálózatok – általánosítási lehetőségek

Egy általánosítási lehetőség ha több nyelőt is megengedünk, akkor tetszőleges függvényeket is kiszámíthatunk Boole hálózatok segítségével. Ilyenkor a kimenet egy bitsorozat, melynek a hossza a nyelők száma. Általánosabb függvények kiszámításához kódoljuk át a be- és kimeneteket $\{0, 1\}$ feletti szóvá.

Példa:

Megkonstruálható egy olyan Boole-hálózat, amelynek a bemeneti változói $X_0, \dots, X_{n-1}, Y_0, \dots, Y_{n-1}$ és az $n + 1$ kimeneti kapuján az $X_{n-1} \cdots X_0$ és $Y_{n-1} \cdots Y_0$ bináris számok összegének a bitjeit számítja ki.

Ha R_0, \dots, R_{n-1} a maradékok bitjei és Z_0, \dots, Z_n az eredmény bitjei, akkor könnyen látható, hogy $Z_0 = X_0 \otimes Y_0$,

Boole-hálózatok – általánosítási lehetőségek

Egy általánosítási lehetőség ha több nyelöt is megengedünk, akkor tetszőleges függvényeket is kiszámíthatunk Boole hálózatok segítségével. Ilyenkor a kimenet egy bitsorozat, melynek a hossza a nyelők száma. Általánosabb függvények kiszámításához kódoljuk át a be- és kimeneteket $\{0, 1\}$ feletti szóvá.

Példa:

Megkonstruálható egy olyan Boole-hálózat, amelynek a bemeneti változói $X_0, \dots, X_{n-1}, Y_0, \dots, Y_{n-1}$ és az $n + 1$ kimeneti kapuján az $X_{n-1} \cdots X_0$ és $Y_{n-1} \cdots Y_0$ bináris számok összegének a bitjeit számítja ki.

Ha R_0, \dots, R_{n-1} a maradékok bitjei és Z_0, \dots, Z_n az eredmény bitjei, akkor könnyen látható, hogy $Z_0 = X_0 \otimes Y_0$, $R_0 = X_0 \wedge Y_0$,

Boole-hálózatok – általánosítási lehetőségek

Egy általánosítási lehetőség ha több nyelőt is megengedünk, akkor tetszőleges függvényeket is kiszámíthatunk Boole hálózatok segítségével. Ilyenkor a kimenet egy bitsorozat, melynek a hossza a nyelők száma. Általánosabb függvények kiszámításához kódoljuk át a be- és kimeneteket $\{0, 1\}$ feletti szóvá.

Példa:

Megkonstruálható egy olyan Boole-hálózat, amelynek a bemeneti változói $X_0, \dots, X_{n-1}, Y_0, \dots, Y_{n-1}$ és az $n + 1$ kimeneti kapuján az $X_{n-1} \cdots X_0$ és $Y_{n-1} \cdots Y_0$ bináris számok összegének a bitjeit számítja ki.

Ha R_0, \dots, R_{n-1} a maradékok bitjei és Z_0, \dots, Z_n az eredmény bitjei, akkor könnyen látható, hogy $Z_0 = X_0 \otimes Y_0$, $R_0 = X_0 \wedge Y_0$,

$$Z_i = \text{parity}_3(X_i, Y_i, R_{i-1}) \quad (1 \leq i \leq n-1),$$

Boole-hálózatok – általánosítási lehetőségek

Egy általánosítási lehetőség ha több nyelöt is megengedünk, akkor tetszőleges függvényeket is kiszámíthatunk Boole hálózatok segítségével. Ilyenkor a kimenet egy bitsorozat, melynek a hossza a nyelők száma. Általánosabb függvények kiszámításához kódoljuk át a be- és kimeneteket $\{0, 1\}$ feletti szóvá.

Példa:

Megkonstruálható egy olyan Boole-hálózat, amelynek a bemeneti változói $X_0, \dots, X_{n-1}, Y_0, \dots, Y_{n-1}$ és az $n + 1$ kimeneti kapuján az $X_{n-1} \cdots X_0$ és $Y_{n-1} \cdots Y_0$ bináris számok összegének a bitjeit számítja ki.

Ha R_0, \dots, R_{n-1} a maradékok bitjei és Z_0, \dots, Z_n az eredmény bitjei, akkor könnyen látható, hogy $Z_0 = X_0 \otimes Y_0$, $R_0 = X_0 \wedge Y_0$,

$$Z_i = \text{parity}_3(X_i, Y_i, R_{i-1}) \quad (1 \leq i \leq n-1),$$

$$R_i = \text{majority}_3(X_i, Y_i, R_{i-1}) \quad (1 \leq i \leq n-1),$$

Boole-hálózatok – általánosítási lehetőségek

Egy általánosítási lehetőség ha több nyelőt is megengedünk, akkor tetszőleges függvényeket is kiszámíthatunk Boole hálózatok segítségével. Ilyenkor a kimenet egy bitsorozat, melynek a hossza a nyelők száma. Általánosabb függvények kiszámításához kódoljuk át a be- és kimeneteket $\{0, 1\}$ feletti szóvá.

Példa:

Megkonstruálható egy olyan Boole-hálózat, amelynek a bemeneti változói $X_0, \dots, X_{n-1}, Y_0, \dots, Y_{n-1}$ és az $n + 1$ kimeneti kapuján az $X_{n-1} \cdots X_0$ és $Y_{n-1} \cdots Y_0$ bináris számok összegének a bitjeit számítja ki.

Ha R_0, \dots, R_{n-1} a maradékok bitjei és Z_0, \dots, Z_n az eredmény bitjei, akkor könnyen látható, hogy $Z_0 = X_0 \otimes Y_0$, $R_0 = X_0 \wedge Y_0$,

$$Z_i = \text{parity}_3(X_i, Y_i, R_{i-1}) \quad (1 \leq i \leq n-1),$$

$$R_i = \text{majority}_3(X_i, Y_i, R_{i-1}) \quad (1 \leq i \leq n-1),$$

$$Z_n = R_{n-1},$$

Boole-hálózatok – általánosítási lehetőségek

Egy általánosítási lehetőség ha több nyelöt is megengedünk, akkor tetszőleges függvényeket is kiszámíthatunk Boole hálózatok segítségével. Ilyenkor a kimenet egy bitsorozat, melynek a hossza a nyelők száma. Általánosabb függvények kiszámításához kódoljuk át a be- és kimeneteket $\{0, 1\}$ feletti szóvá.

Példa:

Megkonstruálható egy olyan Boole-hálózat, amelynek a bemeneti változói $X_0, \dots, X_{n-1}, Y_0, \dots, Y_{n-1}$ és az $n + 1$ kimeneti kapuján az $X_{n-1} \cdots X_0$ és $Y_{n-1} \cdots Y_0$ bináris számok összegének a bitjeit számítja ki.

Ha R_0, \dots, R_{n-1} a maradékok bitjei és Z_0, \dots, Z_n az eredmény bitjei, akkor könnyen látható, hogy $Z_0 = X_0 \otimes Y_0$, $R_0 = X_0 \wedge Y_0$,

$$Z_i = \text{parity}_3(X_i, Y_i, R_{i-1}) \quad (1 \leq i \leq n-1),$$

$$R_i = \text{majority}_3(X_i, Y_i, R_{i-1}) \quad (1 \leq i \leq n-1),$$

$Z_n = R_{n-1}$, ahol parity_3 -t lásd fenn, míg majority_3 a 3 bit többségi bitjét adja vissza.

Boole-hálózatok – általánosítási lehetőségek

Felmerülhet egyéb kapuk használata is.

Boole-hálózatok – általánosítási lehetőségek

Felmerülhet egyéb kapuk használata is.

Minden Boole-függvény leírható Boole-polinommal, így további kaputípusok alkalmazása nem növeli a modell számítási erejét.

Boole-hálózatok – általánosítási lehetőségek

Felmerülhet egyéb kapuk használata is.

Minden Boole-függvény leírható Boole-polinommal, így további kaputípusok alkalmazása nem növeli a modell számítási erejét.

Vegyük észre, hogy a kapuk valójában maguk is Boole-függvények. Így 2-nél több bemenetű kapuk használata esetén a kapuk komplexitását is figyelembe kéne venni a hálózat bonyolultságának definiálásakor különben használhatatlanná válik a modellünk.

Boole-hálózatok – általánosítási lehetőségek

Felmerülhet egyéb kapuk használata is.

Minden Boole-függvény leírható Boole-polinommal, így további kaputípusok alkalmazása nem növeli a modell számítási erejét.

Vegyük észre, hogy a kapuk valójában maguk is Boole-függvények. Így 2-nél több bemenetű kapuk használata esetén a kapuk komplexitását is figyelembe kéne venni a hálózat bonyolultságának definiálásakor különben használhatatlanná válik a modellünk.

Extrém példa: az NP-teljes Hamilton kör probléma kiszámítható egyetlen $\binom{n}{2}$ változós Hamilton-kapuvál, melynek $\binom{n}{2}$ bemenete meghatároz egy gráfot (van-e az egyes pontpárok között él), kimenete $1 \iff$ a gráfban van Hamilton kör.

Boole-hálózatok – általánosítási lehetőségek

Felmerülhet egyéb kapuk használata is.

Minden Boole-függvény leírható Boole-polinommal, így további kaputípusok alkalmazása nem növeli a modell számítási erejét.

Vegyük észre, hogy a kapuk valójában maguk is Boole-függvények. Így 2-nél több bemenetű kapuk használata esetén a kapuk komplexitását is figyelembe kéne venni a hálózat bonyolultságának definiálásakor különben használhatatlanná válik a modellünk.

Extrém példa: az NP-teljes Hamilton kör probléma kiszámítható egyetlen $\binom{n}{2}$ változós Hamilton-kapuvál, melynek $\binom{n}{2}$ bemenete meghatároz egy gráfot (van-e az egyes pontpárok között él), kimenete $1 \iff$ a gráfban van Hamilton kör.

Ezért a legfeljebb 2 befokú \neg, \wedge, \vee kapukra korlátozzuk a megengedett kaputípusokat. Meggondolható, hogy az egyéb 2 változós műveleteknek megfelelő kapuk mindegyike néhány (egyszámjegyű) \neg, \wedge, \vee kapuvál helyettesíthető.

Boole-hálózatok – általánosítási lehetőségek

Felmerülhet az is hogy egy korlátot adjunk meg a kapuk ki-fokára.
Néha célszerű lehet feltenni, hogy egy kapu az általa kiszámított bitet nem tudja akárhány helyre „ingyen” szétosztani.

Boole-hálózatok családja

Egyetlen Boole-hálózat önmagában csak adott hosszúságú bemeneteket tud kezelni.

Boole-hálózatok családja

Egyetlen Boole-hálózat önmagában csak adott hosszúságú bemeneteket tud kezelni.

Definíció

Boole-hálózatok egy családja alatt Boole-hálózatok egy végtelen $\mathcal{C} = (C_0, C_1, C_2, \dots)$ sorozatát értjük, ahol C_n -nek n bemeneti változója van.

Boole-hálózatok családja

Egyetlen Boole-hálózat önmagában csak adott hosszúságú bemeneteket tud kezelni.

Definíció

Boole-hálózatok egy családja alatt Boole-hálózatok egy végtelen $\mathcal{C} = (C_0, C_1, C_2, \dots)$ sorozatát értjük, ahol C_n -nek n bemeneti változója van.

Legyen C egy Boole-hálózat az X_1, \dots, X_n bemeneti változókkal és $w = a_1 \cdots a_n \in \{0, 1\}^*$, ekkor $C(w) := \mathcal{B}_I(C)$, ahol $I(X_i) = a_i$ ($1 \leq i \leq n$).

Boole-hálózatok családja

Egyetlen Boole-hálózat önmagában csak adott hosszúságú bemeneteket tud kezelni.

Definíció

Boole-hálózatok egy családja alatt Boole-hálózatok egy végtelen $\mathcal{C} = (C_0, C_1, C_2, \dots)$ sorozatát értjük, ahol C_n -nek n bemeneti változója van.

Legyen C egy Boole-hálózat az X_1, \dots, X_n bemeneti változókkal és $w = a_1 \cdots a_n \in \{0, 1\}^*$, ekkor $C(w) := \mathcal{B}_I(C)$, ahol $I(X_i) = a_i$ ($1 \leq i \leq n$).

Definíció

Boole-hálózatok egy \mathcal{C} családja **eldönti** az $L \subseteq \{0, 1\}^*$ nyelvet, ha minden $w \in \{0, 1\}^*$ szóra, $w \in L$ akkor és csak akkor, ha $C_n(w) = 1$, ahol $n = |w|$.

Boole-hálózatok családja

Egyetlen Boole-hálózat önmagában csak adott hosszúságú bemeneteket tud kezelni.

Definíció

Boole-hálózatok egy családja alatt Boole-hálózatok egy végtelen $\mathcal{C} = (C_0, C_1, C_2, \dots)$ sorozatát értjük, ahol C_n -nek n bemeneti változója van.

Legyen C egy Boole-hálózat az X_1, \dots, X_n bemeneti változókkal és $w = a_1 \cdots a_n \in \{0, 1\}^*$, ekkor $C(w) := \mathcal{B}_I(C)$, ahol $I(X_i) = a_i$ ($1 \leq i \leq n$).

Definíció

Boole-hálózatok egy \mathcal{C} családja **eldönti** az $L \subseteq \{0, 1\}^*$ nyelvet, ha minden $w \in \{0, 1\}^*$ szóra, $w \in L$ akkor és csak akkor, ha $C_n(w) = 1$, ahol $n = |w|$.

Boole-hálózatok mérete és mélysége

Definíció

- ▶ Egy Boole-hálózat **mérete** a kapuinak száma

Boole-hálózatok mérete és mélysége

Definíció

- ▶ Egy Boole-hálózat **mérete** a kapuinak száma
- ▶ Egy Boole-hálózat **mélysége** a leghosszabb irányított útjának hossza. (Ez nyilván valamelyik bemeneti csúcstól a nyelőig vezet.)

Boole-hálózatok mérete és mélysége

Definíció

- ▶ Egy Boole-hálózat **mérete** a kapuinak száma
- ▶ Egy Boole-hálózat **mélysége** a leghosszabb irányított útjának hossza. (Ez nyilván valamelyik bemeneti csúcstól a nyelőig vezet.)
- ▶ A C és C' Boole-hálózatok **ekvivalensek**, ha megegyezik a bemeneti csúcsaiknak a száma (n) és minden $w \in \{0, 1\}^n$ esetén $C(w) = C'(w)$.

Boole-hálózatok mérete és mélysége

Definíció

- ▶ Egy Boole-hálózat **mérete** a kapuinak száma
- ▶ Egy Boole-hálózat **mélysége** a leghosszabb irányított útjának hossza. (Ez nyilván valamelyik bemeneti csúcstól a nyelőig vezet.)
- ▶ A C és C' Boole-hálózatok **ekvivalensek**, ha megegyezik a bemeneti csúcsaiknak a száma (n) és minden $w \in \{0, 1\}^n$ esetén $C(w) = C'(w)$.

Boole-hálózatok mérete és mélysége

Definíció

- ▶ Egy Boole-hálózat **mérete** a kapuinak száma
- ▶ Egy Boole-hálózat **mélysége** a leghosszabb irányított útjának hossza. (Ez nyilván valamelyik bemeneti csúcstól a nyelőig vezet.)
- ▶ A C és C' Boole-hálózatok **ekvivalensek**, ha megegyezik a bemeneti csúcsaiknak a száma (n) és minden $w \in \{0, 1\}^n$ esetén $C(w) = C'(w)$.

Definíció

- ▶ Egy **Boole-hálózat minimális méretű**, ha nincs vele ekvivalens kisebb méretű hálózat.

Boole-hálózatok mérete és mélysége

Definíció

- ▶ Egy Boole-hálózat **mérete** a kapuinak száma
- ▶ Egy Boole-hálózat **mélysége** a leghosszabb irányított útjának hossza. (Ez nyilván valamelyik bemeneti csúcstól a nyelőig vezet.)
- ▶ A C és C' Boole-hálózatok **ekvivalensek**, ha megegyezik a bemeneti csúcsaiknak a száma (n) és minden $w \in \{0, 1\}^n$ esetén $C(w) = C'(w)$.

Definíció

- ▶ Egy **Boole-hálózat minimális méretű**, ha nincs vele ekvivalens kisebb méretű hálózat.
- ▶ Egy **Boole-hálózat minimális mélységű**, ha nincs vele ekvivalens kisebb mélységű hálózat.

Minimális méretű/mélységű Boole-hálózat család

Definíció

- ▶ **Boole-hálózatok** egy $\mathcal{C} = (C_1, \dots)$ családja **minimális méretű**, ha minden $i \in \mathbb{N}$ esetén a C_i Boole-hálózat minimális méretű.

Minimális méretű/mélységű Boole-hálózat család

Definíció

- ▶ **Boole-hálózatok** egy $\mathcal{C} = (C_1, \dots)$ családja **minimális méretű**, ha minden $i \in \mathbb{N}$ esetén a C_i Boole-hálózat minimális méretű.
- ▶ **Boole-hálózatok** egy $\mathcal{C} = (C_1, \dots)$ családja **minimális mélységű**, ha minden $i \in \mathbb{N}$ esetén a C_i Boole-hálózat minimális mélységű.

Boole-hálózatok méret- és mélységbonyolultsága

Definíció

- ▶ Boole-hálózatok egy $\mathcal{C} = (C_1, \dots)$ családjának **méretbonyolultságán** azt az $f : \mathbb{N} \rightarrow \mathbb{N}$ függvényt értjük, melyre $f(n)$ a C_n mérete ($n \in \mathbb{N}$).

Boole-hálózatok méret- és mélységbonyolultsága

Definíció

- ▶ Boole-hálózatok egy $\mathcal{C} = (C_1, \dots)$ családjának **méretbonyolultságán** azt az $f : \mathbb{N} \rightarrow \mathbb{N}$ függvényt értjük, melyre $f(n)$ a C_n mérete ($n \in \mathbb{N}$).
- ▶ Boole-hálózatok egy $\mathcal{C} = (C_1, \dots)$ családjának **mélységbonyolultságán** azt az $f : \mathbb{N} \rightarrow \mathbb{N}$ függvényt értjük, melyre $f(n)$ a C_n mélysége ($n \in \mathbb{N}$).

Boole-hálózatok méret- és mélységbonyolultsága

Definíció

- ▶ Boole-hálózatok egy $\mathcal{C} = (C_1, \dots)$ családjának **méretbonyolultságán** azt az $f : \mathbb{N} \rightarrow \mathbb{N}$ függvényt értjük, melyre $f(n)$ a C_n mérete ($n \in \mathbb{N}$).
- ▶ Boole-hálózatok egy $\mathcal{C} = (C_1, \dots)$ családjának **mélységbonyolultságán** azt az $f : \mathbb{N} \rightarrow \mathbb{N}$ függvényt értjük, melyre $f(n)$ a C_n mélysége ($n \in \mathbb{N}$).

Boole-hálózatok méret- és mélységbonyolultsága

Definíció

- ▶ Boole-hálózatok egy $\mathcal{C} = (C_1, \dots)$ családjának **méretbonyolultságán** azt az $f : \mathbb{N} \rightarrow \mathbb{N}$ függvényt értjük, melyre $f(n)$ a C_n mérete ($n \in \mathbb{N}$).
- ▶ Boole-hálózatok egy $\mathcal{C} = (C_1, \dots)$ családjának **mélységbonyolultságán** azt az $f : \mathbb{N} \rightarrow \mathbb{N}$ függvényt értjük, melyre $f(n)$ a C_n mélysége ($n \in \mathbb{N}$).

Definíció

- ▶ Egy $L \subseteq \{0, 1\}^*$ **nyelv hálózatméret-bonyolultságán** egy minimális méretű, őt eldöntő Boole-hálózat család méretbonyolultságát értjük.

Boole-hálózatok méret- és mélységbonyolultsága

Definíció

- ▶ Boole-hálózatok egy $\mathcal{C} = (C_1, \dots)$ családjának **méretbonyolultságán** azt az $f : \mathbb{N} \rightarrow \mathbb{N}$ függvényt értjük, melyre $f(n)$ a C_n mérete ($n \in \mathbb{N}$).
- ▶ Boole-hálózatok egy $\mathcal{C} = (C_1, \dots)$ családjának **mélységbonyolultságán** azt az $f : \mathbb{N} \rightarrow \mathbb{N}$ függvényt értjük, melyre $f(n)$ a C_n mélysége ($n \in \mathbb{N}$).

Definíció

- ▶ Egy $L \subseteq \{0, 1\}^*$ **nyelv hálózatméret-bonyolultságán** egy minimális méretű, őt eldöntő Boole-hálózat család méretbonyolultságát értjük.
- ▶ Egy $L \subseteq \{0, 1\}^*$ **nyelv hálózatmélység-bonyolultságán** egy minimális, őt eldöntő Boole-hálózat család mélységbonyolultságát értjük.

Boole-hálózatok méret- és mélységbonyolultsága

Példa:

Legyen $L_{\text{parity}} = \{a_1 \cdots a_n \in \{0, 1\}^* \mid \sum_{i=1}^n a_i \equiv 1 \pmod{2}\}$.

Boole-hálózatok méret- és mélységbonyolultsága

Példa:

Legyen $L_{\text{parity}} = \{a_1 \cdots a_n \in \{0, 1\}^* \mid \sum_{i=1}^n a_i \equiv 1 \pmod{2}\}$.

Könnyen készíthető olyan Boole-hálózat család, amelynek az elemei rendre a

$$\text{parity}_1 = X_1 \quad \text{parity}_2 = X_1 \otimes X_2$$

$$\text{parity}_n = \text{parity}_{n-2} \otimes (X_{n-1} \otimes X_n)$$

Boole-hálózatok méret- és mélységbonyolultsága

Példa:

Legyen $L_{\text{parity}} = \{a_1 \cdots a_n \in \{0, 1\}^* \mid \sum_{i=1}^n a_i \equiv 1(\text{mod } 2)\}$.

Könnyen készíthető olyan Boole-hálózat család, amelynek az elemei rendre a

$$\text{parity}_1 = X_1 \quad \text{parity}_2 = X_1 \otimes X_2$$

$$\text{parity}_n = \text{parity}_{n-2} \otimes (X_{n-1} \otimes X_n)$$

Boole-függvényeket számítják ki ($n \in \mathbb{N}$).

Boole-hálózatok méret- és mélységbonyolultsága

Példa:

Legyen $L_{\text{parity}} = \{a_1 \cdots a_n \in \{0, 1\}^* \mid \sum_{i=1}^n a_i \equiv 1 \pmod{2}\}$.

Könnyen készíthető olyan Boole-hálózat család, amelynek az elemei rendre a

$$\begin{aligned}\text{parity}_1 &= X_1 & \text{parity}_2 &= X_1 \otimes X_2 \\ \text{parity}_n &= \text{parity}_{n-2} \otimes (X_{n-1} \otimes X_n)\end{aligned}$$

Boole-függvényeket számítják ki ($n \in \mathbb{N}$).

Ehhez csak $n - 1$ darab \otimes kapu kell. Az előző példában látott konstrukció alapján egy \otimes kapu összesen 5 darab \neg, \wedge, \vee kapuval szimulálható, így L_{parity} hálózati méret-bonyolultsága legfeljebb $5(n - 1) = O(n)$.

TG-ek szimulálása Boole-hálózatok családjával

Tétel

Ha $L \in \text{TIME}(f(n))$, és $f(n) \geq n$, akkor L hálózatméret-bonyolultsága $O(f(n)^2)$.

Tehát „kis” időbonyolultságú problémák hálózatméret-bonyolultsága is „kicsi”.

TG-ek szimulálása Boole-hálózatok családjával

Tétel

Ha $L \in \text{TIME}(f(n))$, és $f(n) \geq n$, akkor L hálózatméret-bonyolultsága $O(f(n)^2)$.

Tehát „kis” időbonyolultságú problémák hálózatméret-bonyolultsága is „kicsi”.

Következmény

Ha egy $L \in \text{NP}$ nyelv hálózatméret-bonyolultsága polinomiálisnál nagyobb aszimptotikus nagyságrendű, akkor $P \neq \text{NP}$.

TG-ek szimulálása Boole-hálózatok családjával

Tétel

Ha $L \in \text{TIME}(f(n))$, és $f(n) \geq n$, akkor L hálózatméret-bonyolultsága $O(f(n)^2)$.

Tehát „kis” időbonyolultságú problémák hálózatméret-bonyolultsága is „kicsi”.

Következmény

Ha egy $L \in \text{NP}$ nyelv hálózatméret-bonyolultsága polinomiálisnál nagyobb aszimptotikus nagyságrendű, akkor $P \neq \text{NP}$.

Így ez a tétel egy lehetséges támadási felületet ad a $P \neq \text{NP}$ sejtés bizonyításához.

TG-ek szimulálása Boole-hálózatok családjával

Bizonyítás: Legyen $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ egy $f(n)$ időkorlátos, L -et eldöntő determinisztikus TG és $w = x_1 \cdots x_n$ egy n hosszú inputja.

TG-ek szimulálása Boole-hálózatok családjával

Bizonyítás: Legyen $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ egy $f(n)$ időkorlátos, L -et eldöntő determinisztikus TG és $w = x_1 \cdots x_n$ egy n hosszú inputja. Azaz egyszerűség kedvéért feltesszük, hogy a $\text{TIME}(f(n))$ definíciójában lévő $O(f(n))$ konstans faktora 1 (különben mindent ezzel a faktorial kéne szorozni). Feltehető továbbá, hogy $\Sigma = \{0, 1\}$.

TG-ek szimulálása Boole-hálózatok családjával

Bizonyítás: Legyen $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ egy $f(n)$ időkorlátos, L -et eldöntő determinisztikus TG és $w = x_1 \cdots x_n$ egy n hosszú inputja. Azaz egyszerűség kedvéért feltesszük, hogy a $\text{TIME}(f(n))$ definíciójában lévő $O(f(n))$ konstans faktora 1 (különben mindent ezzel a faktorral kéne szorozni). Feltehető továbbá, hogy $\Sigma = \{0, 1\}$.

Ekkor M számítása w -re a konfigurációi által leírhatók, melyeket ha egymás alá írunk, akkor ez a számítás egy $(f(n) + 1) \times (2f(n) + 1)$ -es T táblázattal leírható. A kezdőkonfiguráció az első sor közepén, az $f(n) + 1$ -edik cellán kezdődik. ($f(n) \geq n$ esetén w belefér T -be.)

TG-ek szimulálása Boole-hálózatok családjával

Bizonyítás: Legyen $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ egy $f(n)$ időkorlátos, L -et eldöntő determinisztikus TG és $w = x_1 \cdots x_n$ egy n hosszú inputja. Azaz egyszerűség kedvéért feltesszük, hogy a $\text{TIME}(f(n))$ definíciójában lévő $O(f(n))$ konstans faktora 1 (különben mindent ezzel a faktorial kénye szorozni). Feltehető továbbá, hogy $\Sigma = \{0, 1\}$.

Ekkor M számítása w -re a konfigurációi által leírhatók, melyeket ha egymás alá írunk, akkor ez a számítás egy $(f(n) + 1) \times (2f(n) + 1)$ -es T táblázattal leírható. A kezdőkonfiguráció az első sor közepén, az $f(n) + 1$ -edik cellán kezdődik. ($f(n) \geq n$ esetén w belefér T -be.)

A konfigurációk minden betűje saját cellába kerül, kivétel a fej alatt lévő betű, mely kerüljön az állapottal egy cellába az állapot után konkatenálva.

TG-ek szimulálása Boole-hálózatok családjával

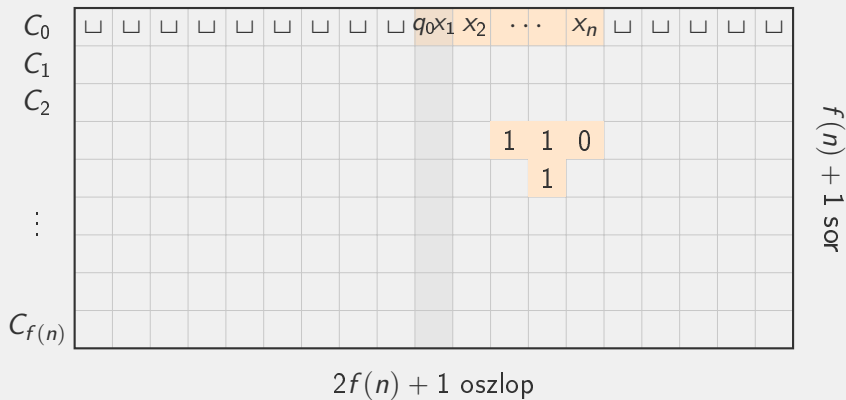
Bizonyítás: Legyen $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ egy $f(n)$ időkorlátos, L -et eldöntő determinisztikus TG és $w = x_1 \cdots x_n$ egy n hosszú inputja. Azaz egyszerűség kedvéért feltesszük, hogy a $\text{TIME}(f(n))$ definíciójában lévő $O(f(n))$ konstans faktora 1 (különben mindent ezzel a faktorral kéne szorozni). Feltehető továbbá, hogy $\Sigma = \{0, 1\}$.

Ekkor M számítása w -re a konfigurációi által leírhatók, melyeket ha egymás alá írunk, akkor ez a számítás egy $(f(n) + 1) \times (2f(n) + 1)$ -es T táblázattal leírható. A kezdőkonfiguráció az első sor közepén, az $f(n) + 1$ -edik cellán kezdődik. ($f(n) \geq n$ esetén w belefér T -be.)

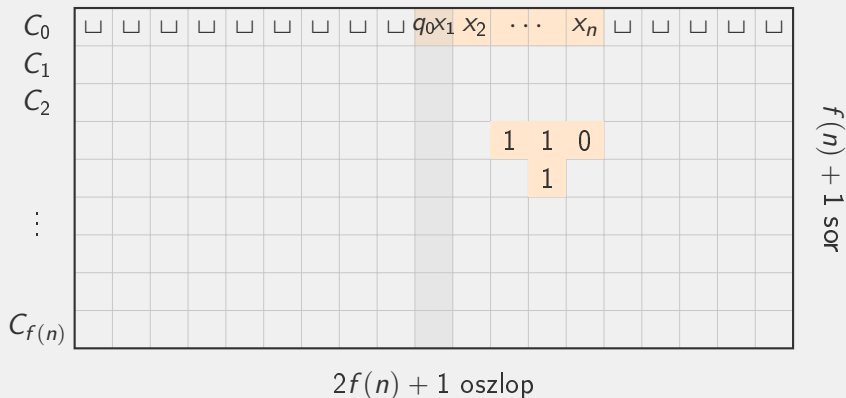
A konfigurációk minden betűje saját cellába kerül, kivétel a fej alatt lévő betű, mely kerüljön az állapottal egy cellába az állapot után konkaténálva.

Például, ha a konfiguráció $01q_6101$, akkor a \sqcup -eket tároló celláktól eltekintve sorra $0, 1, \boxed{q_6 1}, 0, 1$ a cellák tartalma.

TG-ek szimulálása Boole-hálózatok családjával

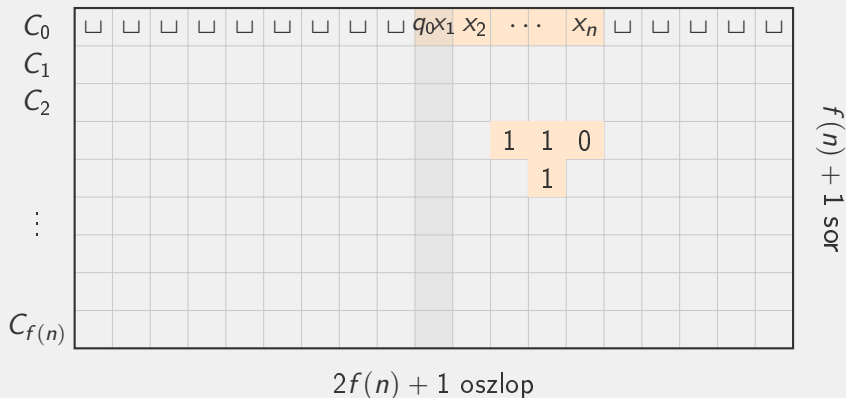


TG-ek szimulálása Boole-hálózatok családjával



Ha esetleg $f(n)$ -nél rövidebb idő alatt jut M megállási konfigurációba, akkor a táblázat utolsó sorait identikusan, ezen utolsó konfigurációval töltjük ki.

TG-ek szimulálása Boole-hálózatok családjával



Ha esetleg $f(n)$ -nél rövidebb idő alatt jut M megállási konfigurációba, akkor a táblázat utolsó sorait identikusan, ezen utolsó konfigurációval töltjük ki.

$T(i, j)$ értéke a konfigurációátmenet definíciója szerint csak a $T(i-1, j-1)$, $T(i-1, j)$, $T(i-1, j+1)$ értékeitől függ.

TG-ek szimulálása Boole-hálózatok családjával

A C_n Boole-hálózat konstrukciója:

$k := |\Gamma \cup Q \times \Gamma|$, ennyi fajta tartalma lehet a celláknak. Képzeljük azt, hogy minden (i, j) cellához tartozik k darab villanykörte ($\text{light}[i, j, 1], \dots, \text{light}[i, j, k]$) a cella minden lehetséges tartalmához pontosan egy. A C_n hálózatban a $\text{light}[i, j, s]$ villanykörte akkor és csak akkor fog égni, ha az (i, j) cella tartalma éppen az s -edik $\Gamma \cup Q \times \Gamma$ -beli szimbólum. (Rögzítjük $\Gamma \cup Q \times \Gamma$ elemeinek egy sorrendjét.)

A villanykörtéknek nincsen hatásuk a kimeneti értékre, a bizonyítás megértését segítik, és a bizonyítás végén ki is iktatjuk őket.

TG-ek szimulálása Boole-hálózatok családjával

Legyenek $(a_1, b_1, c_1), (a_2, b_2, c_2), \dots, (a_r, b_r, c_r)$ azok a $(\text{light}[i-1, j-1, s_1], \text{light}[i-1, j, s_2], \text{light}[i-1, j+1, s_3])$ rendezett 3-asok, melyeknek megfelelő égők, ha világítanak akkor az (i, j) cella tartalma az s -edik $\Gamma \cup Q \times \Gamma$ -beli szimbólum lesz.

TG-ek szimulálása Boole-hálózatok családjával

Legyenek $(a_1, b_1, c_1), (a_2, b_2, c_2), \dots, (a_r, b_r, c_r)$ azok a $(\text{light}[i-1, j-1, s_1], \text{light}[i-1, j, s_2], \text{light}[i-1, j+1, s_3])$ rendezett 3-asok, melyeknek megfelelő égők, ha világítanak akkor az (i, j) cella tartalma az s -edik $\Gamma \cup Q \times \Gamma$ -beli szimbólum lesz.

Vegyük észre, hogy r értéke egy csak M -től függő konstans, n -től (és $f(n)$ -től) nem függ.

TG-ek szimulálása Boole-hálózatok családjával

Legyenek $(a_1, b_1, c_1), (a_2, b_2, c_2), \dots, (a_r, b_r, c_r)$ azok a $(\text{light}[i-1, j-1, s_1], \text{light}[i-1, j, s_2], \text{light}[i-1, j+1, s_3])$ rendezett 3-asok, melyeknek megfelelő égők, ha világítanak akkor az (i, j) cella tartalma az s -edik $\Gamma \cup Q \times \Gamma$ -beli szimbólum lesz.

Vegyük észre, hogy r értéke egy csak M -től függő konstans, n -től (és $f(n)$ -től) nem függ.

Kössük össze ezen 3-asoknak megfelelő villanykörtéket egy-egy \wedge -kapuval, majd ezt az r \wedge -kaput egy \vee -kapuval, amit vezessünk $\text{light}[i, j, s]$ -be. (Mivel bináris kapukat használunk ezt persze több bináris \wedge illetve \vee kapuval kell megoldani.)

TG-ek szimulálása Boole-hálózatok családjával

Legyenek $(a_1, b_1, c_1), (a_2, b_2, c_2), \dots, (a_r, b_r, c_r)$ azok a $(\text{light}[i-1, j-1, s_1], \text{light}[i-1, j, s_2], \text{light}[i-1, j+1, s_3])$ rendezett 3-asok, melyeknek megfelelő égők, ha világítanak akkor az (i, j) cella tartalma az s -edik $\Gamma \cup Q \times \Gamma$ -beli szimbólum lesz.

Vegyük észre, hogy r értéke egy csak M -től függő konstans, n -től (és $f(n)$ -től) nem függ.

Kössük össze ezen 3-asoknak megfelelő villanykörtéket egy-egy \wedge -kapuval, majd ezt az r \wedge -kaput egy \vee -kapuval, amit vezessünk $\text{light}[i, j, s]$ -be. (Mivel bináris kapukat használunk ezt persze több bináris \wedge illetve \vee kapuval kell megoldani.)

A táblázat szélein az értékek persze csak 2 fölötte lévő cella értékétől függenek, ezekre értelemszerűen módosítjuk a konstrukciót.

TG-ek szimulálása Boole-hálózatok családjával

Az első sorban lévő celláknak nincs megelőzője, villanykörtéi a bemeneti változókhoz kapcsolódnak. Tehát a $\text{light}[1, f(n) + 1, q_0 1]$ villanykörte az x_1 inputhoz kapcsolódik, mivel a kezdőkonfiguráció a kezdőállapottal kezdődik, az író-olvasó fej pedig x_1 -re mutat.

TG-ek szimulálása Boole-hálózatok családjával

Az első sorban lévő celláknak nincs megelőzője, villanykörtéi a bemeneti változókhoz kapcsolódnak. Tehát a $\text{light}[1, f(n) + 1, q_0 1]$ villanykörte az x_1 inputhoz kapcsolódik, mivel a kezdőkonfiguráció a kezdőállapottal kezdődik, az író-olvasó fej pedig x_1 -re mutat.

Hasonlóképpen $\text{light}[1, f(n) + 1, q_0 0]$ egy \neg kapun keresztül a x_1 inputhoz kapcsolódik.

TG-ek szimulálása Boole-hálózatok családjával

Az első sorban lévő celláknak nincs megelőzője, villanykörtéi a bemeneti változókhoz kapcsolódnak. Tehát a $\text{light}[1, f(n) + 1, q_0 1]$ villanykörte az x_1 inputhoz kapcsolódik, mivel a kezdőkonfiguráció a kezdőállapottal kezdődik, az író-olvasó fej pedig x_1 -re mutat.

Hasonlóképpen $\text{light}[1, f(n) + 1, q_0 0]$ egy \neg kapun keresztül a x_1 inputhoz kapcsolódik.

$\text{light}[1, f(n) + i, 1]$ az x_i bemenethez, $\text{light}[1, f(n) + i, 0]$ egy \neg kapun keresztül szintén az x_i bemenethez van bedrótözva.

TG-ek szimulálása Boole-hálózatok családjával

Az első sorban lévő celláknak nincs megelőzője, villanykörtéi a bemeneti változókhoz kapcsolódnak. Tehát a $\text{light}[1, f(n) + 1, q_0 1]$ villanykörte az x_1 inputhoz kapcsolódik, mivel a kezdőkonfiguráció a kezdőállapottal kezdődik, az író-olvasó fej pedig x_1 -re mutat.

Hasonlóképpen $\text{light}[1, f(n) + 1, q_0 0]$ egy \neg kapun keresztül a x_1 inputhoz kapcsolódik.

$\text{light}[1, f(n) + i, 1]$ az x_i bemenethez, $\text{light}[1, f(n) + i, 0]$ egy \neg kapun keresztül szintén az x_i bemenethez van bedrótözva.

Minden más j -re a $\text{light}[1, j, \sqcup]$ villanykörték felkapcsolt állapotban vannak, a többi lámpa az első sorban lekapcsolt állapotban van.

TG-ek szimulálása Boole-hálózatok családjával

Az első sorban lévő celláknak nincs megelőzője, villanykörtéi a bemeneti változókhoz kapcsolódnak. Tehát a $\text{light}[1, f(n) + 1, q_0 1]$ villanykörte az x_1 inputhoz kapcsolódik, mivel a kezdőkonfiguráció a kezdőállapottal kezdődik, az író-olvasó fej pedig x_1 -re mutat.

Hasonlóképpen $\text{light}[1, f(n) + 1, q_0 0]$ egy \neg kapun keresztül a x_1 inputhoz kapcsolódik.

$\text{light}[1, f(n) + i, 1]$ az x_i bemenethez, $\text{light}[1, f(n) + i, 0]$ egy \neg kapun keresztül szintén az x_i bemenethez van bedrótózva.

Minden más j -re a $\text{light}[1, j, \sqcup]$ villanykörték felkapcsolt állapotban vannak, a többi lámpa az első sorban lekapcsolt állapotban van.

Ez utóbbi megoldható: egy tetszőleges rögzített x_i bemenethez drótozzuk be simán és \neg -kapun keresztül is a villanykörtét és ezt a két drótot aszerint kapcsoljuk \vee - vagy \wedge -kapun keresztül a villanykörtéhez, hogy melyik konstans Boole-értéket szeretnénk hozzárendelni. (*)

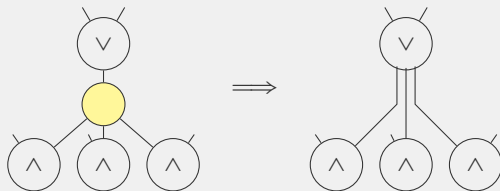
TG-ek szimulálása Boole-hálózatok családjával

M akkor és csak akkor fogadja el w -t, ha T utolsó, $f(n) + 1$ -edik sorában megtalálható q_i . Ez $(2f(n) + 1)|\Gamma|$ darab villanykörtének felel meg, kössük össze őket egy \vee -kapuval, és ez legyen a kimeneti kapu. (ezt persze több bináris \vee kapuval kell megvalósítani).

TG-ek szimulálása Boole-hálózatok családjával

M akkor és csak akkor fogadja el w -t, ha T utolsó, $f(n) + 1$ -edik sorában megtalálható q_i . Ez $(2f(n) + 1)|\Gamma|$ darab villanykörtének felel meg, kössük össze őket egy \vee -kapuval, és ez legyen a kimeneti kapu. (ezt persze több bináris \vee kapuval kell megvalósítani).

A villanykörték kiiktatása (1 a be-fokuk):



CIRCUIT-SAT

Definíció

Egy C n bemenetű Boole-hálózatot **kielégíthetőnek** nevezünk, ha van olyan $w \in \{0, 1\}^n$, melyre $C(w) = 1$.

CIRCUIT-SAT

Definíció

Egy C n bemenetű Boole-hálózatot **kielégíthetőnek** nevezünk, ha van olyan $w \in \{0, 1\}^n$, melyre $C(w) = 1$.

$\text{CIRCUIT-SAT} := \{ \langle C \rangle \mid C \text{ kielégíthető Boole-hálózat} \}$.

(Szokásos módon $\langle C \rangle$ a C valamely legalább bináris ábécé feletti tömör kódját jelöli.)

CIRCUIT-SAT NP-teljes

Tétel

CIRCUIT-SAT NP-teljes.

Bizonyítás: Könnyen meggondolható, hogy CIRCUIT-SAT NP-beli. (Egy konkrét bemenetre $O(\text{hálózat méret})$ időben kiszámítható a kimenet.)

CIRCUIT-SAT NP-teljes

Tétel

CIRCUIT-SAT NP-teljes.

Bizonyítás: Könnyen meggondolható, hogy CIRCUIT-SAT NP-beli. (Egy konkrét bemenetre $O(\text{hálózat méret})$ időben kiszámítható a kimenet.)

Legyen $L \in \text{NP}$, kell, hogy $L \leq_p \text{CIRCUIT-SAT}$, azaz meg kell adnunk egy olyan polinom időben kiszámítható φ függvényt, melyre $w \in L \Leftrightarrow \varphi(w)$ kielégíthető Boole-hálózat.

CIRCUIT-SAT NP-teljes

Tétel

CIRCUIT-SAT NP-teljes.

Bizonyítás: Könnyen meggondolható, hogy CIRCUIT-SAT NP-beli. (Egy konkrét bemenetre $O(\text{hálózat méret})$ időben kiszámítható a kimenet.)

Legyen $L \in \text{NP}$, kell, hogy $L \leq_p \text{CIRCUIT-SAT}$, azaz meg kell adnunk egy olyan polinom időben kiszámítható φ függvényt, melyre $w \in L \Leftrightarrow \varphi(w)$ kielégíthető Boole-hálózat.

Mivel $L \in \text{NP}$ ezért van olyan $p(n)$ polinom időkorlátos M NTG, mely eldönti L -et. Legyen w egy n hosszúságú szó.

CIRCUIT-SAT NP-teljes

Tétel

CIRCUIT-SAT NP-teljes.

Bizonyítás: Könnyen meggondolható, hogy CIRCUIT-SAT NP-beli. (Egy konkrét bemenetre O (hálózat méret) időben kiszámítható a kimenet.)

Legyen $L \in \text{NP}$, kell, hogy $L \leq_p \text{CIRCUIT-SAT}$, azaz meg kell adnunk egy olyan polinom időben kiszámítható φ függvényt, melyre $w \in L \Leftrightarrow \varphi(w)$ kielégíthető Boole-hálózat.

Mivel $L \in \text{NP}$ ezért van olyan $p(n)$ polinom időkorlátos M NTG, mely eldönti L -et. Legyen w egy n hosszúságú szó.

Az előző tétel C_n konstrukciója szimulált egy determinisztikus TG-et bármely n hosszú inputra. Vegyük észre a következőket:

CIRCUIT-SAT NP-teljes

- ▶ A konstrukció működik NTG-re is.

CIRCUIT-SAT NP-teljes

- ▶ A konstrukció működik NTG-re is.
- ▶ Az előző bizonyításban (*)-al jelölt ötlet alapján könnyen módosíthatjuk a konstrukciót úgy, hogy minden bemenetre w -t szimulálja (a w bitjeinek megfelelő villanykörtéket konstans igazra, minden mást konstans hamisra állítva), jelölje ezt C_w , így C_w minden bemenetre w működését szimulálja M -en, így a C_w hálózat akkor és csak akkor kielégíthető, ha $w \in L$. Tehát $\varphi(w) := C_w$ visszavezetés.

CIRCUIT-SAT NP-teljes

- ▶ A konstrukció működik NTG-re is.
- ▶ Az előző bizonyításban (*)-al jelölt ötlet alapján könnyen módosíthatjuk a konstrukciót úgy, hogy minden bemenetre w -t szimulálja (a w bitjeinek megfelelő villanykörteket konstans igazra, minden mást konstans hamisra állítva), jelölje ezt C_w , így C_w minden bemenetre w működését szimulálja M -en, így a C_w hálózat akkor és csak akkor kielégíthető, ha $w \in L$. Tehát $\varphi(w) := C_w$ visszavezetés.
- ▶ $|C_w| = O(p(n)^2)$, mely $O(p(n)^2)$ idő alatt el is készíthető, így φ polinomiális. □

3SAT NP-teljes – bizonyítás Boole-hálózatokkal

Tétel

3SAT NP-teljes.

2. Bizonyítás: [1. Bizonyítás : lásd BSc-n]

3SAT NP-beli. Elég: $\text{CIRCUIT-SAT} \leq_p 3\text{SAT}$.

3SAT NP-teljes – bizonyítás Boole-hálózatokkal

Tétel

3SAT NP-teljes.

2. Bizonyítás: [1. Bizonyítás : lásd BSc-n]

3SAT NP-beli. Elég: $\text{CIRCUIT-SAT} \leq_p 3\text{SAT}$.

Legyenek x_1, \dots, x_n a C Boole-hálózat bemeneti változói és rendeljük hozzá az x_{n+1}, \dots, x_m változókat a kapukhoz. Készítünk egy 3KNF-et, melynek x_1, \dots, x_m lesznek a változói.

3SAT NP-teljes – bizonyítás Boole-hálózatokkal

Tétel

3SAT NP-teljes.

2. Bizonyítás: [1. Bizonyítás : lásd BSc-n]

3SAT NP-beli. Elég: $\text{CIRCUIT-SAT} \leq_p 3\text{SAT}$.

Legyenek x_1, \dots, x_n a C Boole-hálózat bemeneti változói és rendeljük hozzá az x_{n+1}, \dots, x_m változókat a kapukhoz. Készítünk egy 3KNF-et, melynek x_1, \dots, x_m lesznek a változói.

Minden kapuhoz tartozik 1 vagy 2 bemeneti változó a be-élek alapján és egy kapuhoz tartozó kimeneti változó.

3SAT NP-teljes – bizonyítás Boole-hálózatokkal

Tétel

3SAT NP-teljes.

2. Bizonyítás: [1. Bizonyítás : lásd BSc-n]

3SAT NP-beli. Elég: $\text{CIRCUIT-SAT} \leq_p 3\text{SAT}$.

Legyenek x_1, \dots, x_n a C Boole-hálózat bemeneti változói és rendeljük hozzá az x_{n+1}, \dots, x_m változókat a kapukhoz. Készítünk egy 3KNF-et, melynek x_1, \dots, x_m lesznek a változói.

Minden kapuhoz tartozik 1 vagy 2 bemeneti változó a be-élek alapján és egy kapuhoz tartozó kimeneti változó.

\neg -kapu:

$$(\bar{x}_i \rightarrow x_j) \wedge (\bar{x}_j \rightarrow x_i)$$

(x_i bemeneti x_j kimeneti változó)

3SAT NP-teljes – bizonyítás Boole-hálózatokkal

Tétel

3SAT NP-teljes.

2. Bizonyítás: [1. Bizonyítás : lásd BSc-n]

3SAT NP-beli. Elég: $\text{CIRCUIT-SAT} \leq_p 3\text{SAT}$.

Legyenek x_1, \dots, x_n a C Boole-hálózat bemeneti változói és rendeljük hozzá az x_{n+1}, \dots, x_m változókat a kapukhoz. Készítünk egy 3KNF-et, melynek x_1, \dots, x_m lesznek a változói.

Minden kapuhoz tartozik 1 vagy 2 bemeneti változó a be-élek alapján és egy kapuhoz tartozó kimeneti változó.

\neg -kapu:

$$(\bar{x}_i \rightarrow x_j) \wedge (\bar{x}_j \rightarrow x_i) \quad (x_i \text{ bemeneti } x_j \text{ kimeneti változó})$$

\wedge -kapu:

$$(\bar{x}_i \wedge \bar{x}_j \rightarrow \bar{x}_k) \wedge (\bar{x}_i \wedge x_j \rightarrow \bar{x}_k) \wedge (x_i \wedge \bar{x}_j \rightarrow \bar{x}_k) \wedge (x_i \wedge x_j \rightarrow x_k) \\ (x_i, x_j \text{ bemeneti } x_k \text{ kimeneti változó})$$

3SAT NP-teljes – bizonyítás Boole-hálózatokkal

\vee -kapu:

$$(\bar{x}_i \wedge \bar{x}_j \rightarrow \bar{x}_k) \wedge (\bar{x}_i \wedge x_j \rightarrow x_k) \wedge (x_i \wedge \bar{x}_j \rightarrow x_k) \wedge (x_i \wedge x_j \rightarrow x_k)$$

(x_i, x_j bemeneti x_k kimeneti változó)

3SAT NP-teljes – bizonyítás Boole-hálózatokkal

\vee -kapu:

$$(\bar{x}_i \wedge \bar{x}_j \rightarrow \bar{x}_k) \wedge (\bar{x}_i \wedge x_j \rightarrow x_k) \wedge (x_i \wedge \bar{x}_j \rightarrow x_k) \wedge (x_i \wedge x_j \rightarrow x_k)$$

(x_i, x_j bemeneti x_k kimeneti változó)

Az \wedge -sel összekötött részformulákat alakítsuk klózokká.

3SAT NP-teljes – bizonyítás Boole-hálózatokkal

\vee -kapu:

$$(\bar{x}_i \wedge \bar{x}_j \rightarrow \bar{x}_k) \wedge (\bar{x}_i \wedge x_j \rightarrow x_k) \wedge (x_i \wedge \bar{x}_j \rightarrow x_k) \wedge (x_i \wedge x_j \rightarrow x_k)$$

(x_i, x_j bemeneti x_k kimeneti változó)

Az \wedge -sel összekötött részformulákat alakítsuk klózokká.

Még egy klóz: x_m önmagában, ha x_m a kimeneti kapu.

3SAT NP-teljes – bizonyítás Boole-hálózatokkal

\vee -kapu:

$$(\bar{x}_i \wedge \bar{x}_j \rightarrow \bar{x}_k) \wedge (\bar{x}_i \wedge x_j \rightarrow x_k) \wedge (x_i \wedge \bar{x}_j \rightarrow x_k) \wedge (x_i \wedge x_j \rightarrow x_k)$$

(x_i, x_j bemeneti x_k kimeneti változó)

Az \wedge -sel összekötött részformulákat alakítsuk klózokká.

Még egy klóz: x_m önmagában, ha x_m a kimeneti kapu.

A hiányos klózokat egészítsük ki 3 hosszúvá.

3SAT NP-teljes – bizonyítás Boole-hálózatokkal

\vee -kapu:

$$(\bar{x}_i \wedge \bar{x}_j \rightarrow \bar{x}_k) \wedge (\bar{x}_i \wedge x_j \rightarrow x_k) \wedge (x_i \wedge \bar{x}_j \rightarrow x_k) \wedge (x_i \wedge x_j \rightarrow x_k)$$

(x_i, x_j bemeneti x_k kimeneti változó)

Az \wedge -sel összekötött részformulákat alakítsuk klózokká.

Még egy klóz: x_m önmagában, ha x_m a kimeneti kapu.

A hiányos klózokat egészítsük ki 3 hosszúvá.

A klózok konjunkciója lesz a C -nek megfelelő φ formula.

3SAT NP-teljes – bizonyítás Boole-hálózatokkal

\vee -kapu:

$$(\bar{x}_i \wedge \bar{x}_j \rightarrow \bar{x}_k) \wedge (\bar{x}_i \wedge x_j \rightarrow x_k) \wedge (x_i \wedge \bar{x}_j \rightarrow x_k) \wedge (x_i \wedge x_j \rightarrow x_k)$$

(x_i, x_j bemeneti x_k kimeneti változó)

Az \wedge -sel összekötött részformulákat alakítsuk klózokká.

Még egy klóz: x_m önmagában, ha x_m a kimeneti kapu.

A hiányos klózokat egészítsük ki 3 hosszúvá.

A klózok konjunkciója lesz a C -nek megfelelő φ formula.

Meggondolható, hogy C kielégíthető akkor és csak akkor, ha φ kielégíthető

3SAT NP-teljes – bizonyítás Boole-hálózatokkal

\vee -kapu:

$$(\bar{x}_i \wedge \bar{x}_j \rightarrow \bar{x}_k) \wedge (\bar{x}_i \wedge x_j \rightarrow x_k) \wedge (x_i \wedge \bar{x}_j \rightarrow x_k) \wedge (x_i \wedge x_j \rightarrow x_k)$$

(x_i, x_j bemeneti x_k kimeneti változó)

Az \wedge -sel összekötött részformulákat alakítsuk klózokká.

Még egy klóz: x_m önmagában, ha x_m a kimeneti kapu.

A hiányos klózokat egészítsük ki 3 hosszúvá.

A klózok konjunkciója lesz a C -nek megfelelő φ formula.

Meggondolható, hogy C kielégíthető akkor és csak akkor, ha φ kielégíthető és hogy a konstrukció polinomiális idejű valamint polinomiális méretű φ -t ad.

□