Számítási modellek

5. előadás

Boole függvények

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

X	Y	$\neg X$	$X \wedge Y$	$X \vee Y$	$X \to Y$	$X \otimes Y$
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

$X \to Y$ 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 $f : \{0,1\}^n \to \{0,1\}$ leképezést.

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 $f : \{0,1\}^N \to \{0,1\}$ 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 csúcs 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.

Tétel

Minden Boole-függvény kifejezhető Boole-polinomként.

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

$$\varphi_{\mathbf{z}}(X_1, \ldots, X_n) := \bigwedge_{i=1}^{n} L_i,$$

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

Ekkor $\varphi_{\mathbf{z}}(X_1, \ldots, X_n) = 1$ akkor és csak akkor, ha $X_i = z_i$ minden $1 \leqslant i \leqslant n$ esetén.

Boole-polinomok

Tehát a

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

Boole-polinomra $\psi(X_1, \ldots, X_n) = 1 \iff$ valamely $\mathbf{z} \in f^1$-re $\varphi_{\mathbf{z}}(X_1, \ldots, X_n) = 1 \iff$ valamely $\mathbf{z} \in f^1$-re $X_i = z_i$ minden $1 \leqslant i \leqslant n \iff (X_1, \ldots, X_n) \in f^1 \iff f(X_1, \ldots X_n) = 1.$ □

Példa:

X_1	X_2	X_3	$f(X_1, X_2, X_3)$
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

$$\psi(X_1, X_2, X_3) =$$

$$(X_1 \wedge X_2 \wedge \neg X_3) \vee$$

$$(\neg X_1 \wedge X_2 \wedge \neg X_3) \vee$$

$$(\neg X_1 \wedge \neg X_2 \wedge X_3) \vee$$

$$(\neg X_1 \wedge \neg X_2 \wedge \neg X_3)$$

Topologikus sorrend

Definíció

Legyen $G = (V, E)$ egy irányított gráf. A csúcsaink egy $v_1, \ldots, v_n$ 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(élszám) futási idejű algoritmusok ismeretesek (például a mélységi keresés befejezési száma szerinti csökkenő sorrend jó; egy gráf körmenetes 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.
- 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.

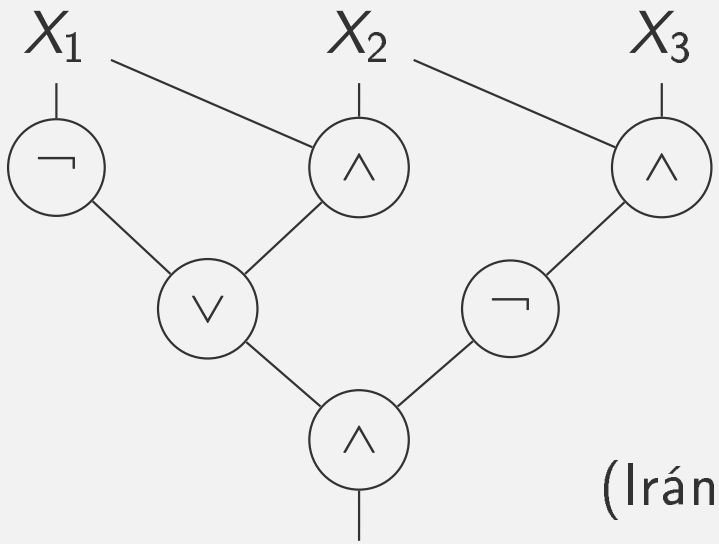
Boole-hálózat

Definíció

Boole-hálózat egy csúcscímkézett, aciklikus irányított gráf, melyre

- 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,
- egyetlen 0 kifokú kapu (nyelő) van, ezt **kimeneti kapunak** (vagy kimeneti csúcsnak) nevezzünk.

Példa:



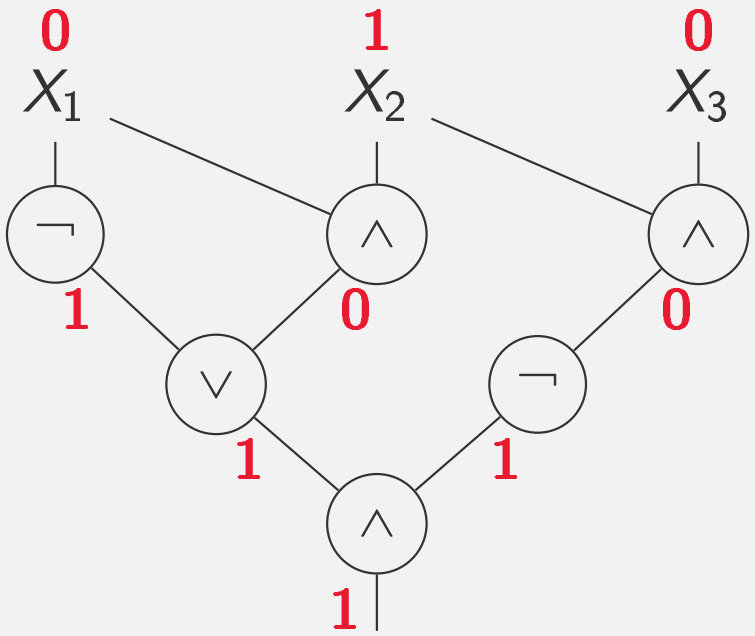
(Irányítás: ↓)

Boole-hálózatok kiértékelése

Definíció

Legyenek $V(C) = \{X_1, \ldots, X_n\}$ a C Boole-hálózat bemeneti csúcsai és $I : V(C) \to \{0, 1\}$ 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:

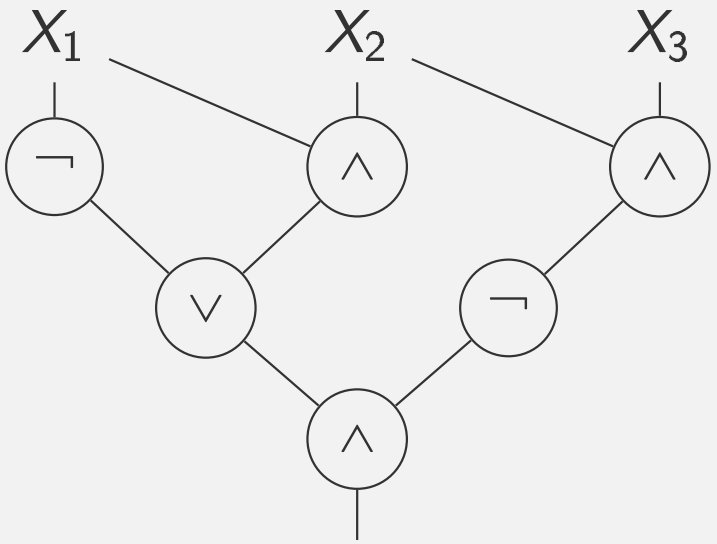


Boole-hálózat által kiszámított Boole-függvény

Definíció

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

Példa:

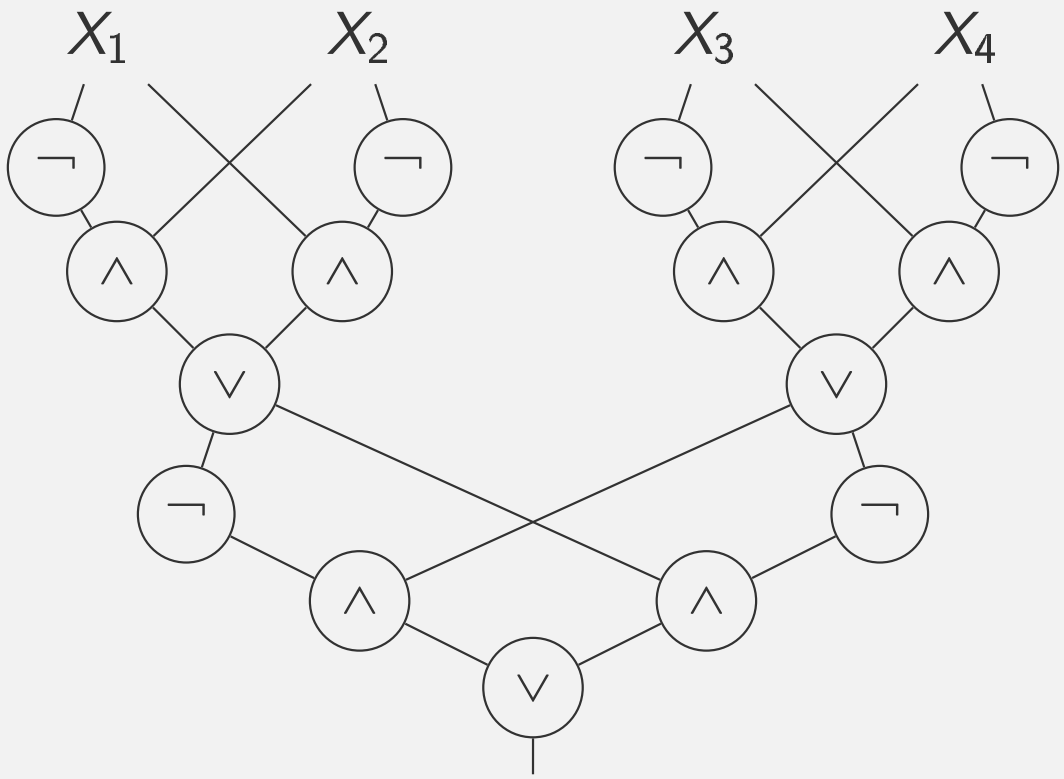


A kiszámított Boole-függvény:

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



$\text{parity}_1(X_1) = X_1, \quad \text{parity}_2(X_1, X_2) = X_1 \otimes X_2,$
$\text{parity}_3(X_1, X_2, X_3) = X_1 \otimes (X_2 \otimes X_3),$
$\text{parity}_4(X_1, X_2, X_3, X_4) = (X_1 \otimes X_2) \otimes (X_3 \otimes X_4),$
$\text{parity}_n(X_1, \ldots, X_n) = \text{parity}_{n-2}(X_1, \ldots, X_{n-2}) \otimes (X_{n-1} \otimes X_n).$

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.

Példa:

Megkonstruálható egy olyan Boole-hálózat, amelynek a bemeneti változói $X_0,\ldots,X_{n-1},Y_0,\ldots,Y_{n-1}$ 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 $R_0,\ldots R_{n-1}$ a maradékok bitjei és $Z_0,\ldots Z_n$ az eredmény bitjei, akkor könnyen látható, hogy $Z_0 = X_0 \otimes Y_0$, $R_0 = X_0 \wedge Y_0$,

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

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

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

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-kapuval, melynek $\binom{n}{2}$ bemenete meghatároz egy gráfot (van-e az egyes pontpárok között él), kimenete 1 $\Longleftrightarrow$ 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) $\neg, \wedge, \vee$ kapuval 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.

Definíció

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

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

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
- 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 $\mathcal{C} = (C_1, \ldots)$ **családja minimális méretű**, ha minden $i \in \mathbb{N}$ esetén a C_i Boole-hálózat minimális méretű.
- **Boole-hálózatok** egy $\mathcal{C} = (C_1, \ldots)$ **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 $\mathcal{C} = (C_1, \ldots)$ családjának **méretbonyolultságán** azt az $f : \mathbb{N} \to \mathbb{N}$ függvényt értjük, melyre $f(n)$ a C_n mérete ($n \in \mathbb{N}$).
- Boole-hálózatok egy $\mathcal{C} = (C_1, \ldots)$ családjának **mélységbonyolultságán** azt az $f : \mathbb{N} \to \mathbb{N}$ 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 $L_{\text{parity}} = \left\{ a_1 \cdots a_n \in \{0,1\}^* \;\middle|\; \sum_{i=1}^{n} a_i \equiv 1 (\text{mod } 2) \right\}$.

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 ($n \in \mathbb{N}$).

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} hálózatméret-bonyolultsága legfeljebb $5(n-1) = O(n)$.

TG-ek szimulálása Boole-hálózatok családjával

Tétel

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

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

Következmény

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

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

TG-ek szimulálása Boole-hálózatok családjával

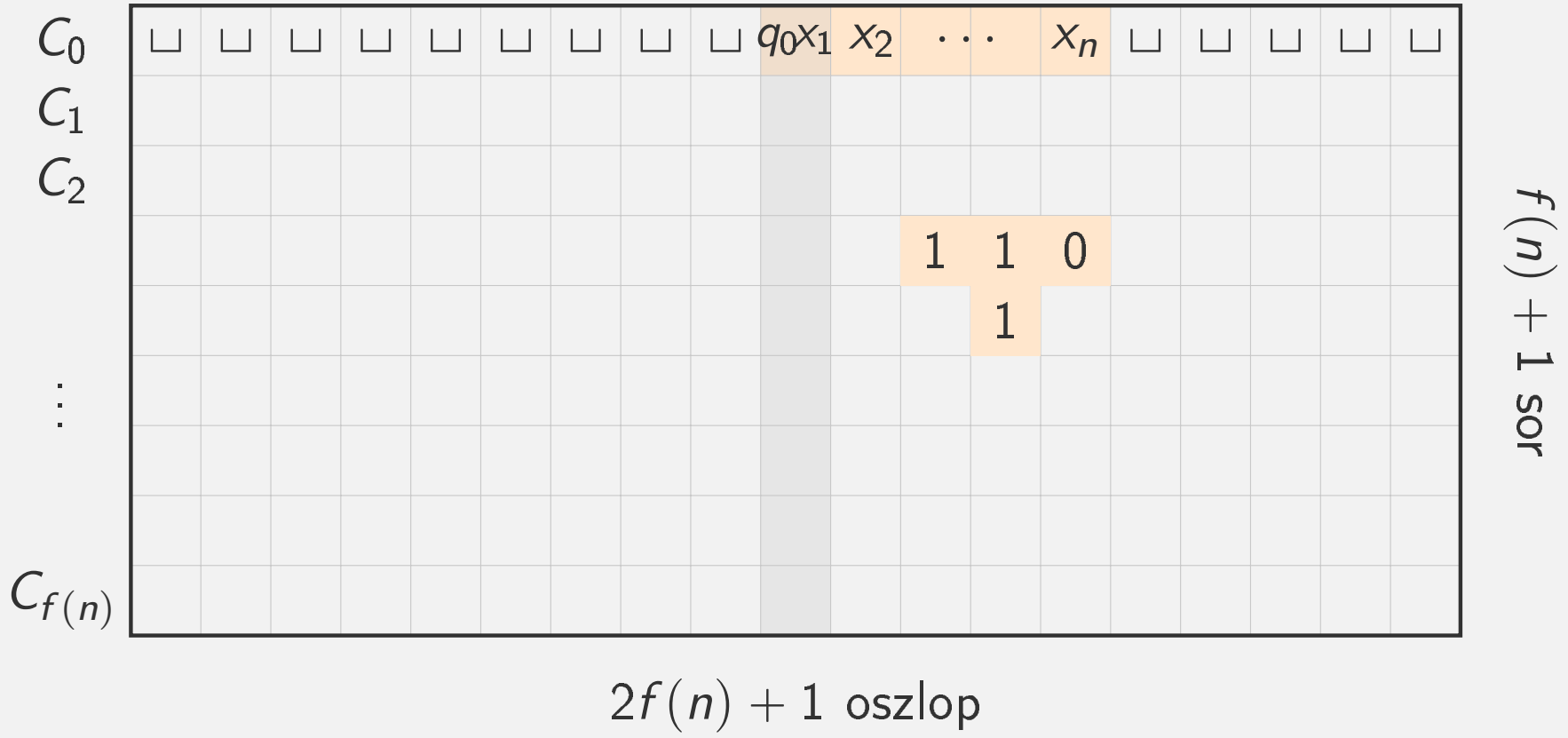
Bizonyítás: Legyen $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_\mathrm{i}, q_\mathrm{n} \rangle$ egy $f(n)$ időkorlátos, L-et eldöntő determinisztikus TG és $w = x_1 \cdots x_n$ egy n hosszú inputja. Azaz egyszerűség kedvéért feltesszük, hogy a TIME$(f(n))$ definíciójában lévő $O(f(n))$ konstans faktora 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. ($f(n) \geqslant n$ 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.

Például, ha a konfiguráció $01q_6101$, akkor a $\sqcup$-eket tároló celláktól eltekintve sorra 0,1,$\boxed{q_61}$,0,1 a cellák tartalma.

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 cellákknak. Képzeljük azt, hogy minden (i, j) cellához tartozik k darab villanykörte (light$[i, j, 1]$,. . .,light$[i, j, k]$) a cella minden lehetséges tartalmához pontosan egy. A C_n hálózatban a 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 $(a_1, b_1, c_1), (a_2, b_2, c_2), \ldots, (a_r, b_r, c_r)$ 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 függnek, 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 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 light$[1,\ f(n)+1,\ q_0 0]$ egy $\neg$ kapun keresztül a x_1 inputhoz kapcsolódik.

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

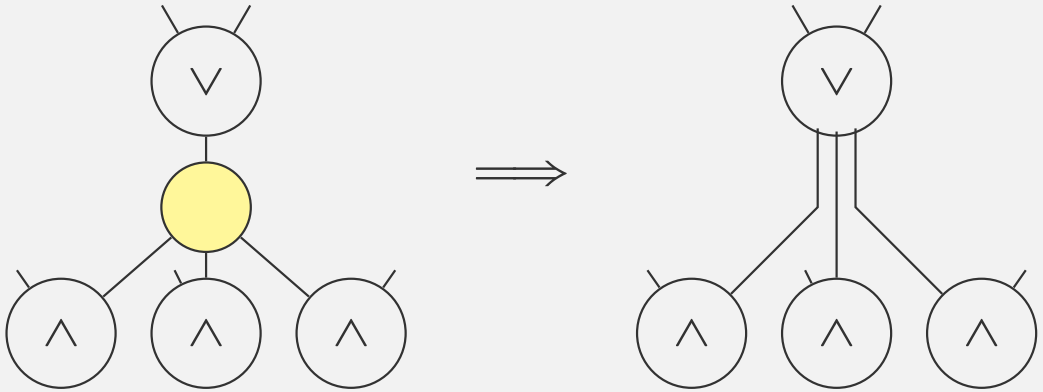
Minden más j-re a light$[1,j,\sqcup]$ villanykörték felkapcsolt állapotban vannak, a többi lámpa az első sorban lekapcsolt állapotban van.

Ez utóbbi megoldható: egy tetszóleges 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).

A villanykörték kiiktatása (1 a be-fokuk):



Meggondoltuk tehát, hogy $(C_0, C_1, \ldots)$ eldönti L-et.

C_n konstukciójában $O(f(n)^2)$ villanykörtét használtunk. Egy villanykörte előző sortól való függőségét egy csak M-től függő konstans darab kapuval szimuláltuk. Az elején és a végén kell még $O(f(n))$ kapu, tehát C_n összesen $O(f(n)^2)$ kaput használ. □

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 $:= \{ \langle C \rangle \mid C$ 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(hálózat méret) időben kiszámítható a kimenet.)

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

Mivel $L \in \mathrm{NP}$ 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.
- 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.
- $|C_w| = O(p(n)^2)$, mely $O(p(n)^2)$ idő alatt el is készíthető, így φ poilnomiális. $\square$

3SAT NP-teljes – bizonyítás Boole-hálózatokkal

Tétel

3SAT NP-teljes.

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

3SAT NP-beli. Elég: CIRCUIT-SAT $\leqslant_p$ 3SAT.

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

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

¬-kapu:

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

∧-kapu:

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

(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.

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. $\square$