Számításelmélet

Hatodik előadás

PSPACE-teljes problémák

- **Definíció:**
  - **QBF**: Adott egy φ prenex alakú zárt Boole formula
  - **Kérdés:** Igaz-e φ?
- **Példa:**
  - $\exists X \forall Y \exists Z((X \vee Y) \wedge (Y \vee Z) \wedge (\neg Y \vee \neg Z))$ igaz
  - $\exists X \forall Y((X \vee Y) \wedge (\neg X \vee \neg Y))$ nem igaz
- **Tétel:** QBF PSPACE-teljes

QBF PSPACE-teljes

- **Bizonyítás: $\mathbf{QBF} \in \mathbf{PSPACE}$**
  - Az alábbi polinom tárigényű A algoritmus QBF-et dönti el:
    - **Egy φ QBF bemenetre**
      - **Ha φ-ben nincs kvantor**, akkor értékeljük ki
        - Ha φ $igaz$, akkor a kimenetre: $igen$, egyébként a kimenetre: nem
      - **Ha $\varphi = \exists X \psi$,** akkor rekurzívan hívjuk meg A-t ψ-re $X = igaz$ és $X = hamis$ értékekkel is; Ha **valamelyik** esetben $igen$ a kimenet, akkor a kimenetre: $igen$, egyébként a kimenetre: nem
      - **Ha $\varphi = \forall X \psi$,** akkor rekurzívan hívjuk meg A-t ψ-re $X = igaz$ és $X = hamis$ értékekkel is; Ha **mindkét** esetben $igen$ a kimenet, akkor a kimenetre: $igen$, egyébként a kimenetre: nem
  - **Belátható, hogy A QBF-et dönti el**
  - **A tárigénye:**
    - A rekurzió mélysége: változók száma
    - Egy szinten tárolandó adat: egy változó igazságértékei
    - Az összes tárigény lineáris φ változóinak számában

QBF PSPACE-teljes

- **Bizonyítás: QBF PSPACE-nehéz**
  - Legyen $L \in \mathrm{PSPACE}$, M egy $c \cdot n^k$ tárigényű Turing-gép, ami L-et dönti el
  - Legyen w az M egy tetszőleges n hosszú bemenete
  - **Megadunk egy φ QBF-t úgy, hogy $w \in L(M) \Leftrightarrow \varphi$ $igaz$**
  - M-nek w-n legfeljebb $h = 2^{dn^k}$ különböző konfigurációja van egy megfelelő (c-től és M-től függő) d konstansra
  - Ezért M legfeljebb h lépésben elfogadja vagy elutasítja w-t
  - Először tetszőleges c_1, c_2 konfigurációra és $t \geq 1$ számra, megadunk egy $\varphi_{c_1,c_2,t}$ formulát, hogy

$\varphi_{c_1,c_2,t}$ $igaz \Leftrightarrow$ M el tud jutni c_1–ből c_2–be legfeljebb t lépésben

  - Ezután a $\varphi_{c_{kezdő},c_{elfogadó},h}$ felhasználásával megadjuk φ-t (feltehető, hogy M-nek pontosan egy elfogadó konfig.-ja van)

QBF PSPACE-teljes

- M egy c konfigurációjának leírásához egy diszjunkt C váltózóhalmazt (konfiguráció azonosító (KA) halmazt) használunk, melyben egy $c_{j,a}$ változó jelentése: **„A c konfiguráció j-ik betűje a”**
- **Jelölések:**
  - $\exists C$ (rendre $\forall C$) jelöli a C-beli vátozók egzisztenciális (rendre univerzális) kvantifikálását
  - C és D KA-k esetén $C = D$ egy olyan formulát jelöl, ami biztosítja azt, hogy a C és D-beli egymásnak megfelelő változók értékei megegyezzenek
- A $\varphi_{c_1,c_2,1}$ **formula megkonstruálható polinom időben:** azt kell formalizálni, hogy „$c_1 = c_2$ vagy M egy lépésben el tud jutni c_1-ből c_2-be”
- **Megjegyzés:**
  - A második rész formalizálásakor a δ-beli átmeneteket kódoljuk megfelelően
  - Így egy kvantormentes formulát kapunk

QBF PSPACE-teljes

- $t > 1$ esetén:
  - Egy rossz megoldás: a $\varphi_{c_1,c_2,t} := \exists N\left(\varphi_{c_1,n,\frac{t}{2}} \wedge \varphi_{n,c_2,\frac{t}{2}}\right)$ rekurzív konstrukció nem jó mert exponenciális méretű formulát eredményez
  - Egy jó megoldás: $\varphi_{c_1,c_2,t} := \exists N \forall C \forall C' \Big[\big((C = C_1 \wedge C' = N) \vee$
    -
-

QBF PSPACE-teljes

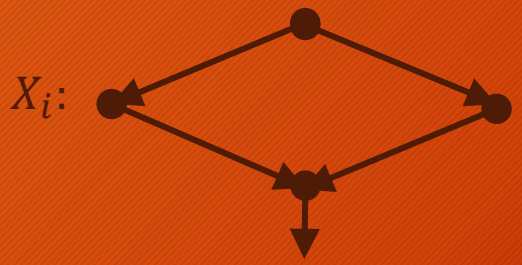
- **A teljes konstrukció időigénye:**
  - A rekurzió mélysége: $O(\log h) = O\left(\log 2^{dn^k}\right) = O(n^k)$
  - A rekurzió minden szintjén $O(n^k)$ időben elkészíthető a „$\varphi_{c,c',\frac{t}{2}}$-n kívüli rész”
- **Megjegyzés:**
  - QBF akkor is PSPACE-teljes ha a bemenetét megszorítjuk:
    - A kvantorok alternálnak, az első (és az utolsó kvantor) $\exists$, a formula magja KNF
    - Ekkor QBF felfogható kétszemélyes játékként:
      - Az első játékos választja a páratlan változók értékét és célja a formula igazzá tétele
      - A második játékos választja a páros változók értékét és célja a formula hamissá tétele

FÖLDRAJZI JÁTÉK PSPACE-teljes

- **Definíció: FÖLDRAJZI JÁTÉK (FJ)**
  - Adott egy $G = (V, E)$ irányított gráf és $p \in V$
  - **Kérdés**: Van-e nyerő stratégiája a kezdő játékosnak az alábbi játékban:
    - Két játékos felváltva jelöli meg p-ből kiindulva G csúcsait úgy, hogy a következő játékos mindig csak egy a legutoljára megjelölt csúcsból elérhető még meg nem jelölt csúcsot választhat
    - Az veszít aki nem tud újabb csúcsot megjelölni
- **Tétel:** FJ PSPACE-teljes
- **Bizonyítás: FJ $\in$ PSPACE**
  - Hasonlóan látható be, mint QBF esetében

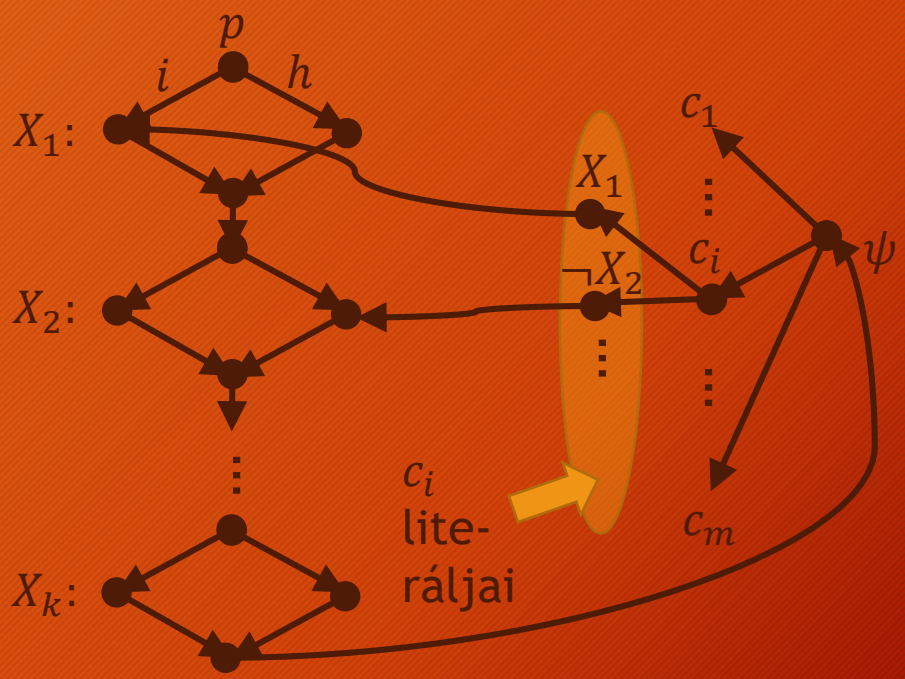
FÖLDRAJZI JÁTÉK PSPACE-teljes

- **Bizonyítás: FJ PSPACE-nehéz**
  - Polinom időben visszavezetjük rá QBF-et
    - Legyen $\varphi = \exists X_1 \forall X_2 \dots \forall X_{k-1} \exists X_k \psi$, ahol $\psi = c_1 \wedge \dots \wedge c_m$ (QBF ilyen alakú bemenetekre megszorítva is PSPACE-teljes)
    - **Konstruáljuk meg G_φ-t a következőképpen**
      - Minden X_i-hez elkészítünk egy részgráfot:



FÖLDRAJZI JÁTÉK PSPACE-teljes

- **Bizonyítás (FJ PSPACE-nehéz)**
  - Legyen G_φ a következő gráf:
  - A konstrukció polinom időben elvégezhető
  - Belátható, hogy φ **igaz** $\Leftrightarrow$ G_φ**-ben van nyerő stratégiája a kezdő játékosnak**
  - **Megjegyzés**: A játékosok által követett út a „rombuszokban” meghatároz egy változóhozzá-rendelést



- **Feladat:**
  - Konstruáljuk meg G_φ-t ha $\varphi = \exists X_1 \forall X_2 \exists X_3 ((X_1 \vee X_2) \wedge (X_2 \vee X_3) \wedge (\neg X_2 \vee \neg X_3))$

Az L és NL osztályok

- **Definíció:** $\mathrm{L} = \mathrm{SPACE}(\log n), \mathrm{NL} = \mathrm{NSPACE}(\log n)$
- Triviálisan $\mathrm{L} \subseteq \mathrm{NL}$, az a sejtés, hogy a tartalmazás valódi
- **Tétel:** $\mathrm{ELÉR} \in \mathrm{NL}$
- **Bizonyítás:**
  - Legyen M egy nemdeterminisztikus Turing-gép ami adott $G = (V, E)$ gráfra és $s, t \in V$ csúcsokra a következőt teszi:
    - Ráírja s-t a második szalagra
    - Ráírja a 0-t a harmadik szalagra
    - Amíg a harmadik szalagon $|V|$-nél kisebb szám van
      - Legyen u a második szalagon lévő csúcs
      - Nemdeterminisztikusan felír u helyére egy u-ból elérhető v csúcsot a második szalagra
      - Ha $v = t$, akkor elfogadja a bemenetet, egyébként növeli a harmadik szalagon lévő számot binárisan eggyel
    - Elutasítja a bemenetet
  - **Belátható, hogy M $O(log\,|V|)$ tárral eldönti, hogy van-e út s-ből t-be**

Az L és NL osztályok

- **Definíció:**
  - Egy $L_1 \subseteq \Sigma^*$ **nyelv logaritmikus tárral visszavezethető** egy $L_2 \subseteq \Delta^*$ nyelvre $(\boldsymbol{L_1 \leq_l L_2})$, ha $L_1 \leq L_2$ és a visszavezetéshez használt függvény kiszámítható logaritmikus táras determinisztikus (off-line) Turing-géppel
  - **Egy L nyelv NL-nehéz** (a log. táras visszavezetésre nézve), ha minden $L' \in \mathrm{NL}$ nyelvre, $L' \leq_l L$; ha ráadásul $L \in \mathrm{NL}$ is teljesül, akkor L **NL-teljes** (a log. táras visszavezetésre nézve)
- **Tétel:** L zárt a logaritmikus tárral való visszavezetésre nézve
- **Bizonyítás:**
  - Tegyük fel, hogy $L_1 \leq_l L_2$ és $L_2 \in \mathrm{L}$
  - Legyen M_2 az L_2-t eldöntő, M pedig a visszavezetésben használt f függvényt kiszámoló logaritmikus táras determinisztikus Turing-gép
  - **Megjegyzés**: Most nem alkalmazhatjuk a polinomidejű visszavezetésre való zártságnál alkalmazott gondolatmenetet

Az L és NL osztályok

- **Bizonyítás:**
  - Legyen M_1 az a Turing-gép, ami a következőképpen működik
    - M_1 tetszőleges u szóra
      - A második szalagján egy bináris számlálóval nyomon követi, hogy M_2 feje hányadik betűjét olvassa az $f(u)$ szónak; legyen ez a szám i (i kezdetben 1)
      - Amikor M_2 lépne egyet, akkor M_1 az M-et szimulálva előállítja a harmadik szalagon az $f(u)$ i-ik betűjét (de csak ezt a betűt, az $f(u)$ többi betűje nem kerül a harmadik szalagra)
      - 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 eközben M_1 azt látja, hogy M_2 elfogadó vagy elutasító állapotba lép, akkor M_1 is belép a saját elfogadó vagy elutasító állapotába, egyébként folytatja a szimulációt a következő lépéssel
  - **Belátható**, hogy az így vázolt M_1 L_1-et dönti el és a működése során csak logaritmikus méretű tárat használ, azaz $L_1 \in \mathrm{L}$
- **Következmény:** Ha L NL-teljes és $L \in \mathrm{L}$, akkor $\mathrm{L} = \mathrm{NL}$

Az L és NL osztályok

- **Tétel:** $\mathrm{ELÉR}$ **NL-teljes**
- **Bizonyítás:**
  - Korábban láttuk, hogy $\mathbf{ELÉR} \in \mathbf{NL}$
  - Legyen $L \in \mathrm{NL}$, megmutatjuk, hogy $\boldsymbol{L \leq_l \mathrm{ELÉR}}$
    - Legyen M egy L-et eldöntő $O(\log n)$ táras nemdeterminisztikus Turing-gép és legyen u az M egy n hosszú bemenete
    - Akkor M egy konfigurációja $c \cdot \log n$ méretű valamely alkalmas c konstansra
    - Legyen G az M konfigurációs gráfja az u-n
    - Legyen s és t rendre az M kezdő- és elfogadó konfigurációja az u-n (feltehetjük, hogy M-nek pontosan egy elfogadó konfigurációja van)
  - Ekkor $\boldsymbol{u \in L(M) \Leftrightarrow u \in L \Leftrightarrow}$ **G-ben van út s-ből t-be és**
  - **G megkonstruálható egy log. táras N determinisztikus Turing-géppel:**
    - N felsorolja az összes $c \cdot \log n$ hosszú szót az egyik szalagján, majd teszteli, hogy az legális konfigurációja-e M-nek, ha igen, akkor a szót kiírja a kimenetre
    - Az élek (konfiguráció párok) hasonlóképpen felsorolhatók, tesztelhetők és a kimenetre írhatók

Az L és NL osztályok

- **Következmény:** $\mathrm{NL} \subseteq \mathrm{P}$
- **Bizonyítás:**
  - Az előző bizonyításban használt $L \in \mathrm{NL}$ nyelv konfigurációs gráfja polinom méretű $(2^{c \cdot \log n})$ és polinom időben megkonstruálható
  - Erre a gráfra, a kezdő és az elfogadó konfigurációra kell megoldani az $\mathrm{ELÉR}$ problémát
  - Mivel $\mathrm{ELÉR} \in \mathrm{P}$ következik, hogy $L \in \mathrm{P}$
- **Tétel (Immerman-Szelepcsényi):** $\mathrm{NL} = \mathrm{coNL}$
- **Tétel:** $\mathrm{L} \subseteq \mathrm{NL} = \mathrm{coNL} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXPTIME}$, ahol

$$\mathrm{EXPTIME} = \cup_{k=1}^{\infty} \mathrm{TIME}(2^{n^k})$$

- Ismert, hogy $\mathrm{NL} \subset \mathrm{PSPACE}$ és $\mathrm{P} \subset \mathrm{EXPTIME}$
- Az a sejtés, hogy minden tartalmazás valódi

Bonyolultsági osztályok hierarchiája