Logika v Računalništvu: Zapiski Vaj

Blaž Sovdat* Borja Bovcon† Martin Frešer‡

30. april 2014

1 Prevedbe problemov na SAT

V tem poglavju grobo opišemo prevedbe nekaterih odločitvenih problemov na odločitvenih problem SAT, ki sprašuje ali za dano Boolovo formulo obstaja taka prireditev vrednosti spremenljivkam, da bo formula resnična. Preden nadaljujemo vzpostavimo nekaj notacije. Pišemo $[n] := \{1, 2, \ldots, n\}$ za prvih n naravnih števil in $\binom{[n]}{k}$ za družino k-podmnožic množice $[n]$; tako je $\binom{n}{k} = |\binom{[n]}{k}|$.

1.1 Barvanje grafov

Naj bo $G = (V, E)$ graf in naj bo $k > 0$. Graf G je k-obarvljiv, če obstaja $c : V \to \{1, 2, \ldots, k\}$, da za vse $uv \in E$ velja $c(v) \neq c(u)$. Sedaj za dan (G, k) definiramo Boolovo formulo φ, da je φ zadovoljiva (angl. satisfiable) natanko tedaj, ko je $\chi(G) \leq k$.

Najprej za dan graf generiramo pogoj, da so povezana vozlišča različnih barv:

$$\bigwedge_{uv \in E} \bigwedge_{i=1}^{k} \neg(c_{v,i} \wedge c_{u,i}) \tag{1}$$

Dodamo pogoj, da ima vsako vozlišče barvo:

$$\bigwedge_{v \in V} \bigvee_{i \in [k]} c_{v,i}$$

Nazadnje zagotovimo še, da je vsako vozlišče kvečjemu ene barve:

$$\bigwedge_{v \in V} \bigwedge_{(i,j) \in \binom{[k]}{2}} (\neg(c_{v,i} \wedge c_{v,j}))$$

Celotna Boolova formula je potem

$$\left(\bigwedge_{uv \in E} \bigwedge_{i=1}^{k} \neg(c_{v,i} \wedge c_{u,i}) \right) \wedge \left(\bigwedge_{v \in V} \bigvee_{i \in [k]} c_{v,i} \right) \wedge \left(\bigwedge_{v \in V} \bigwedge_{(i,j) \in \binom{[k]}{2}} (\neg(c_{v,i} \wedge c_{v,j})) \right).$$

Formula je velikosti približno $2mk + nk + 2n\binom{k}{2} = O(k(m + nk))$.

*Email: blaz.sovdat@gmail.com.
†Email: gojace@gmail.com
‡Email: martin.freser@gmail.com

1.2 Sudoku

Odločitveni problem SUDOKU sprašuje, če lahko prazna polja 9×9 Sudoku mreže zapolnimo tako, da bo konfiguracija veljavna. Sudoku prevedmo na 9-barvanje grafa G na točkah $s_{11}, s_{12}, \ldots, s_{21}, \ldots, s_{99}$. (Točke ustrezajo poljem 9×9 mreže.) Naj bo s_{ij} polje v i-ti vrstici in j-tem stoplcu. Potem povežemo graf tako, da je $G[\{s_{i1}, \ldots, s_{i9}\}]$ poln graf na 9 točkah $\{s_{i1}, \ldots, s_{i9}\}$ za vsako vrstico i; podobno je $G[\{s_{1j}, \ldots, s_{9j}\}]$ poln graf na 9 točkah za vsak stolpec j. To pomeni $18 \cdot \binom{9}{2}$ povezav. Na koncu dodamo manjkajoče povezave iz 3×3 kvadratkov — dodamo povezave, da točke 3×3 podmrež tvorijo K_9 — kar nam da dodatnih $5 \cdot 9$ povezav, ker imamo 9 takih podmrež. (Primer: Za zgornjo levo 3×3 podmrežo povežemo s_{12} in s_{21}, s_{21} in s_{32}, s_{32} in s_{23}, s_{12} in s_{23}; digaonali s_{11}, s_{22}, s_{33} in s_{31}, s_{22}, s_{31} vsaka zase tvorita K_3.)

1.3 Hadamard

Pri odločitvenem problemu Hadamardove matrike najprej generiramo matriko velikosti $n \times n$ (paramater n poda uporabnik) spremenljivk $v_1 s_1, v_1 s_2, \ldots, v_n s_n$, kjer spremeljivka $v_i s_j$ predstavlja komponento v i-ti vrstici in j-tem solpcu matrike. Nato naredimo XOR vseh možnih parov vrstic za vsak stolpec, nakar z uporabo knjižnice *itertools* generiramo vse možne stolpce, ki imajo $\frac{n}{2}$ elementov 1 (*true*) in $\frac{n}{2}$ elementov -1 (*false*). Za vsak stolpec j povežemo enačbe vseh možnih stolpec, ki ustrezajo stolpcu na mestu j z operacijo OR, nakar dobljene OR enačbe za vsak posamazen stolpec povežemo še z AND operacijo in s tem dobimo ustrezno Boolovo formulo za Hadamardovo matriko.

1.4 Erdősev problem diskrepance

Za generiranje SAT instanc uporabljamo program, ki sta ga napisala Konev in Lisista.