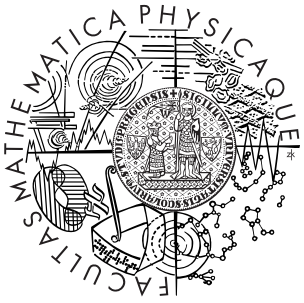
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Tomáš Jirotka

Složitost booleovských funkcí

Katedra algebry

Vedoucí bakalářské práce: Prof. RNDr. Jan Krajíček, DrSc.

Studijní program: Matematika

2009

Rád bych na tomto místě upřímně poděkoval panu profesoru Krajíčkovi za mnoho užitečných rad, které mi během psaní tohoto dokumentu poskytl. Poděkování patří také všem kolegům ze Studentského logického semináře v zimním semestru 2008/2009, kde jsme se rovněž zabývali složitostí booleovských funkcí.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 28. května 2009 Tomáš Jirotka

Obsah

Název práce: Složitost booleovských funkcí
Autor: Tomáš Jirotka
Katedra (ústav): Katedra algebry
Vedoucí bakalářské práce: Prof. RNDr. Jan Krajíček, DrSc.
Email vedoucího práce: krajicek@math.cas.cz

Abstrakt: Práce se zabývá obvodovou složitostí booleovských funkcí. V první části představíme některé obecné výsledky, které však zatím nejsou dostatečně silné. Pro vyřešení P versus NP problému bychom chtěli exponenciální spodní odhady, ale současné nejlepší jsou pouze lineární. V dalších kapitolách zavedeme speciální obvody jako monotónní a obvody s omezenou hloubkou, pro něž známe spodní meze dokonce exponenciální. To však nestačí. V práci neprezentujeme žádné nové výsledky, ale pouze shrnujeme ty nejdůležitější dosud známé.

Klíčová slova: booleovské funkce, booleovské obvody, výpočetní složitost, obvodová složitost, spodní odhady.

Title: Complexity of Boolean Functions
Author: Tomáš Jirotka
Department: Department of Algebra
Supervisor: Prof. RNDr. Jan Krajíček, DrSc.
Supervisor's email address: krajicek@math.cas.cz

Abstract: The paper is concerned with the circuit complexity of boolean functions. In the first part we present some general results, which are still not strong. To solve the P versus NP problem we would like to obtain exponential lower bounds, while we have only linear ones. In the next chapters we introduce some special cases of circuits such as monotone and bounded depth circuits, for which the contemporary lower bounds are even exponential. Unfortunately that is not sufficient. The paper does not contain any new results and we only summarize the most important facts known to us.

Keywords: boolean functions, boolean circuits, computational complexity, circuit complexity, lower bounds.

Kapitola 1

Úvod

Teorie složitosti patří k moderním matematickým disciplinám, velmi úzce souvisejícím s teoretickou informatikou. Jejím ohniskem je zkoumání časové a prostorové náročnosti algoritmů, tj. jak dlouho algoritmus počítá výsledek pro různě velké vstupy, resp. kolik spotřebuje paměti. K odhadům výpočetní náročnosti programů se hojně využívá idealizovaný model tzv. *Turingova stroje*. Tento obecný koncept představil Alan Turing již v roce 1936. Lze si jej představovat jako program pracující s páskami, na kterých má uložená data. Ke každé pásce přiléhá čtecí a zapisovací hlava, která se v každém kroku může na základě určitých pravidel posunout o jedno políčko vpravo či vlevo a také může přepsat jeho obsah.

Definice 1.1. Nechť A je konečná množina znaků, *abeceda*, nechť obsahuje prázdný znak $b \in A$ a nechť $|A| > 2$. Označme $\Sigma = A \setminus \{b\}$ množinu vstupních znaků. Dále nechť Q je konečná množina stavů, obsahující počáteční stav q_0 a dva koncové stavy q_+ a q_- (přijímací a zamítací). Konečně mějme přechodovou funkci

$$\delta : \quad Q \times A \to Q \times A \times \{L, R\}\,,$$

kde písmena L a R značí posuv hlavy vlevo, resp. vpravo. Pak Turingovým strojem M rozumíme uspořádanou šestici $(Q, A, b, \delta, q_0, \{q_+, q_-\})$.

Dobou běhu $T_M(x)$ Turingova stroje M myslíme celkový počet posunutí všech hlav, dokud se stroj nedostane do koncového stavu, má-li na vstupu x. Turingův stroj se však také zastavit nemusí. V takovém případě je doba běhu nekonečná. *Prostorovou náročností* pak rozumíme počet políček „navštívených“ hlavami během výpočtu.

Často se jako abeceda Σ volí množina $\{0,1\}$ a vstupy pak vypadají jako posloupnosti bitů. Množinu všech vektorů (neboli též slov skládajících se pouze z) nul a jedniček libovolné konečné délky značme $\{0,1\}^*$. Podobně budeme značit $\{0,1\}^n$ množinu všech slov délky n. Obvyklou úlohou, kterou mají Turingovy stroje řešit, je rozhodnout, zda-li vstupní vektor patří do jistého *jazyka* $L \subseteq \{0,1\}^*$. Takovým strojům, které po určité době trvání výpočtu skončí s odpovědí ano, nebo ne, říkáme *rozhodovací*. V některých případech však rozhodovací stroj nelze zkonstruovat, a musíme se proto spokojit s poněkud slabším výpočetním modelem – *přijímacím strojem*. Ten se zastaví, právě tehdy když vstupní slovo do jazyka náleží. V opačném případě výpočet nikdy neskončí, ergo jeho výsledek se nedozvíme, protože, pokud jej předčasně ukončíme, nikdy nebudeme mít jistotu, že by se za nějakou dobu stroj přeci jen nezastavil.

Pro lepší představu uveďme krátký příklad. Chceme zjišťovat, zda vstup obsahuje lichý počet jedniček. Pokud ano, přijmeme takový vstup (koncovým stavem je q_+), resp. jej zamítneme v opačném případě (koncovým stavem je q_-). To znamená, že Turingův stroj počítá paritu vstupu neboli $\bigoplus_{i=1}^{n} a_i$, kde $a = (a_1, \ldots, a_n)$ je vstupní slovo. Je zřejmé, že celková doba výpočtu bude záviset na délce vstupu (délku slova a značme $|a|$). Právě od této závislosti se odvíjí klasifikace algoritmu do tzv. *tříd složitosti*.

Definice 1.2. Mějme jazyk $L \subseteq \{0,1\}^*$. Pak $L \in \mathsf{P}$, právě tehdy když existuje Turingův stroj M rozhodující L a

$$\exists k \in \mathbb{N}\, \forall n \in \mathbb{N}\, \forall x \in \{0,1\}^n : \quad T_M(x) = O(n^k)\,.$$

Množinu P nazýváme třídou *polynomiálně rozhodnutelných jazyků*.

Obdobně $L \in \mathsf{EXPTIME}$, právě tehdy když existuje Turingův stroj M rozhodující L a

$$\exists p(n)\, \forall n \in \mathbb{N}\, \forall x \in \{0,1\}^n : \quad T_M(x) = O\left(2^{p(n)}\right)\,,$$

kde $p(n)$ je polynom v proměnné n. Množinu $\mathsf{EXPTIME}$ nazýváme třídou *exponenciálně rozhodnutelných jazyků*.

Poznamenejme, že nás zajímá především asymptotické chování algoritmů, když $n \to \infty$. Zřejmě platí

$$\mathsf{P} \subseteq \mathsf{EXPTIME}\,,$$

neboť pro každý polynom $q(n)$ existuje jisté n_0, že pro všechna $n > n_0$ je shora omezený libovolnou exponenciálou se základem větším než 1.

Aby náš vhled do problému byl úplný, je třeba ještě definovat, co rozumíme *nedeterministickým Turingovým strojem*. Takový stroj N má navíc v množině stavů Q nedeterministický stav $q_?$, do kterého když se dostane, tak se rozhodne zcela libovolně, jak v probíhajícím výpočtu pokračovat. Pochopitelně je možné, že v některém kroku zvolí chybnou možnost, a zavře si tak cestu ke správné odpovědi, čemuž je třeba věnovat pozornost. Totiž pokud výpočet skončí zamítacím stavem, neznamená to nutně, že daný vstup nepatří do příslušného jazyka. Jen se dozvíme, že zvolená posloupnost rozhodnutí byla chybná, a je proto nutné začít výpočet znovu. Když naopak stroj vstupní slovo přijme, je toto rozhodnutí definitivní. Říkáme, že nedeterministický Turingův stroj *přijímá* vstup, právě tehdy když existuje nějaký přijímací výpočet. Možných výpočtů je však velmi mnoho (typicky exponenciálně), a proto je výpočetně velmi náročné je všechny projít.

Dobou běhu $T_N(x)$ nedeterministického Turingova stroje N rozumíme počet kroků nejkratšího přijímacího výpočtu na vstupu x, pokud takový existuje. V opačném případě definujeme $T_N(x) = \infty$.

Analogicky, jako jsme definovali třídu P, můžeme definovat třídu NP.

Definice 1.3. Mějme jazyk $L \subseteq \{0,1\}^*$. Pak $L \in \mathsf{NP}$, právě tehdy když existuje nedeterministický Turingův stroj N přijímající jazyk L a splňující

$$\exists k \in \mathbb{N}\, \forall n \in \mathbb{N}\, \forall x \in \{0,1\}^n \cap L : \quad T_N(x) = O(n^k)\,.$$

Množinu NP nazýváme třídou jazyků, přijímaných nedeterministickým strojem v polynomiálním čase.

Jak jsme již naznačili výše, mnohdy známe efektivnější nedeterministické algoritmy. Avšak to neznamená, že to platí *vždy*. Přesněji řečeno nevíme, zda náhodou neexistuje stejně rychlý deterministický program řešící tutéž úlohu. Formálně vzato nás zejména zajímá, zda platí

$$\mathsf{P} \stackrel{?}{=} \mathsf{NP}\,. \tag{1.1}$$

Odpověď na tuto otázku zatím neznáme a patrně se jí hned tak nedočkáme. Jde o mimořádně obtížný problém, který byl rovněž zařazen na seznam *Millennium Prize Problems* [3]. Všeobecně se soudí, že výše uvedená rovnost neplatí a že platí ostrá inkluze

$$\mathsf{P} \subset \mathsf{NP}\,. \tag{1.2}$$

Důsledky tohoto tvrzení jsou přitom velmi závažné a dotýkají se i takových oblastí jako je kryptografie. Například moderní asymetrická kryptografie, která je využívána i pro tzv. elektronický podpis, je totiž do značné míry založena na tom, že některé matematické problémy mají výpočetně velmi náročné řešení. Konkrétně uvažme problém nalezení rozkladu složeného čísla v součin prvočísel. O něm se ví, že je jistě ve třídě NP, ale přitom nemáme dokázáno, že by neexistoval žádný polynomiální deterministický algoritmus[1]. Jeho existence, která nutně plyne z případné rovnosti $\mathsf{P} = \mathsf{NP}$, by přitom mohla vést (nejen) k velmi snadnému prolomení již zmiňovaného elektronického podpisu.

Jedním z přístupů, kterými se matematici pokoušejí potvrdit platnost inkluze (1.2), je využití dolních odhadů složitosti Turingových strojů počítajících booleovské funkce. V následujícím textu shrneme ty nejzajímavější dosud dokázané odhady a předvedeme metody jejich vytváření a prokazování.

[1]Rozumíme algoritmus pro klasické počítače, nikoli kvantové – pro ně existuje algoritmus se složitostí $O(n^3)$.

Kapitola 2

Booleovské funkce a obvody

2.1 Základní pojmy a definice

Nyní přejdeme k základním pojmům a definicím, pomocí nichž je vybudována poměrně rozsáhlá teorie booleovských funkcí. Budeme přitom postupovat podobně jako v článku Boppana, Sipser [2].

Definice 2.1. *Booleovská funkce n proměnných* je každá funkce f splňující

$$f : \{0,1\}^n \to \{0,1\}\,,$$

přičemž množinu $\{0,1\}$ chápeme jako dvouprvkové těleso $\mathbb{Z}_2$, resp. $\{0,1\}^n$ jako vektorový prostor dimenze n nad tímto tělesem.

Každá booleovská funkce má přesně 2^n možných vstupů, odkud již snadno nahlédneme, že booleovských funkcí n proměnných je 2^{2^n}.

Dále v textu budeme dodržovat následující značení. Pro slovo $x \in \{0,1\}^n$ rozumíme x_i hodnotu na i-té pozici, neboli

$$x = (x_1, \ldots, x_n)\,.$$

Ve výkladu také používáme následující symboly: $a \wedge b$ jako logické „a a zároveň b“, $a \vee b$ nechť označuje spojku „a nebo b“ a $a \oplus b$ chápejme jako „vylučovací nebo“, resp. jako sčítání modulo 2. Negaci proměnné nebo libovolného výrazu budeme značit pruhem nad oním výrazem nebo také značkou $\neg$.

Nyní na vektorovém prostoru $\{0,1\}^n$ zavedeme následující částečné kanonické uspořádání:

$$\forall a, b \in \{0,1\}^n : \quad a \leq b \quad \Leftrightarrow \quad \forall i \in \{1, \ldots, n\} : \quad a_i \leq b_i\,.$$

Poznamenejme, že některá slova porovnat nelze, např. $(0,1)$ a $(1,0)$.

Budeme říkat, že funkce f je *monotónní*, právě když

$$\forall a, b \in \{0,1\}^n, a \leq b \quad \Rightarrow \quad f(a) \leq f(b)\,.$$

Příkladem monotónní funkce je součin, $f(x_1, x_2) = x_1 x_2$. Naopak součet $g(x_1, x_2) = x_1 \oplus x_2$ monotónní není, jak se lze snadno přesvědčit: $g(0,1) = = 1$, ale přitom $g(1,1) = 0$.

Lemma 2.2. *Nechť x, y a z jsou booleovské proměnné. Pak platí:*

(i) $x \vee 0 = x$, $x \vee 1 = 1$, $x \wedge 0 = 0$, $x \wedge 1 = x$, $x \oplus 0 = x$, $x \oplus 1 = \overline{x}$;

(ii) *operace $\vee$, $\wedge$ a $\oplus$ jsou komutativní a asociativní;*

(iii) *(distributivita)* $x \wedge (y \oplus z) = (x \wedge y) \oplus (x \wedge z)$;

(iv) $x \vee x = x$, $x \vee \overline{x} = 1$, $x \wedge x = x$, $x \wedge \overline{x} = 0$, $x \oplus x = 0$, $x \oplus \overline{x} = 1$, $x \vee (x \wedge y) = x$, $x \wedge (x \vee y) = x$;

(v) *(de Morganova pravidla)* $\overline{x \vee y} = \overline{x} \wedge \overline{y}$, $\overline{x \wedge y} = \overline{x} \vee \overline{y}$.

Definice 2.3. Pro booleovskou proměnnou x buď $x^1 = x$ a $x^0 = \neg x$. Dále nechť $x \in \{0,1\}^n$. Pak *minterm* m_a pro $a \in \{0,1\}^n$ definujeme jako

$$m_a(x) = x_1^{a_1} \wedge \ldots \wedge x_n^{a_n}\,.$$

Obdobně definujeme *maxterm*

$$s_a(x) = x_1^{\neg a_1} \vee \ldots \vee x_n^{\neg a_n}\,.$$

Věta 2.4 (o reprezentaci booleovské funkce)**.** *Každou booleovskou funkci f lze reprezentovat jako*

$$f(x) = \bigvee_{a \in \{y; f(y)=1\}} m_a(x) = \bigwedge_{b \in \{y; f(y)=0\}} s_b(x)\,.$$

Těmto vyjádřením říkáme disjunktivní, resp. konjunktivní normální forma, zkráceně DNF, CNF.

Důkaz. Podle definice výše je $m_a(x) = 1$, právě tehdy když $x = a$. Je-li $f(x) = 1$, pak existuje $a \in \{y; f(y) = 1\}$, že $a = x$ a $m_a(x) = 1$, tedy $\bigvee m_a(x) = 1$. Je-li $f(x) = 0$, pak pro všechna $a \in \{y; f(y) = 1\}$ platí $a \neq x$, a tudíž $m_a(x)$ je vždy 0. Proto $\bigvee m_a(x) = 0$.

Naopak, je-li $\bigvee m_a(x) = 1$, tak pro nějaké $a \in \{y; f(y) = 1\}$ musí platit $a = x$, proto i $f(x) = 1$. Je-li $\bigvee m_a(x) = 0$, tak žádné takové a neexistuje, tudíž $f(x) = 0$.

Analogicky se důkaz provede i pro konjunktivní formu. □

Zůstává však otázkou, jakým způsobem k dané funkci najít příslušné mintermy nebo maxtermy. Naštěstí však existují spolehlivé algoritmy, které tento problém řeší. My se však omezíme na pouhé konstatování, že je máme k dispozici, a v tomto textu je popisovat nebudeme. Zájemce lze odkázat na literaturu [17, kapitola 2].

Věta 2.5. *Každou výrokovou formuli φ lze vyjádřit pomocí booleovské funkce f.*

Důkaz. Označme $x_1, \ldots, x_n$ proměnné ve formuli φ. Vytvoříme-li tabulku všech 2^n možných vstupních hodnot a ke každé přiřadíme výslednou hodnotu z formule, lze tuto tabulku použít jako tabulku hodnot funkce f. □

Nyní jsme schopni převádět mezi sebou booleovské funkce a výrokové formule. Naším dalším cílem je zavedení jakéhosi abstraktního výpočetního modelu, pomocí něhož pak budeme provádět odhady složitosti různých booleovských funkcí.

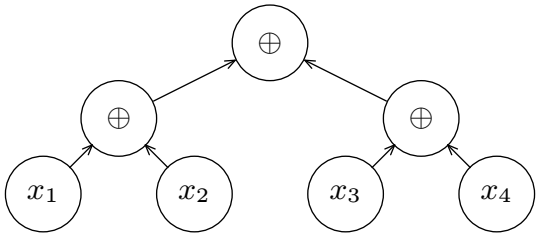
Definice 2.6. Nechť Ω je množina elementárních operací, tzv. *báze*. *Booleovský obvod C v bázi Ω* je acyklický orientovaný graf na množině $\{z_1, \ldots, z_s\}$. Vrcholy, které mají vstupní stupeň (*fanin*) 0, označíme $x_1, \ldots, x_n$ a budeme je nazývat vstupy. Vrcholy, jež mají výstupní stupeň (*fanout*) 0, označíme $y_1, \ldots, y_k$ a budeme jim říkat výstup. Všechny vrcholy kromě vstupních označíme nějakou q-ární funkcí $\omega \in \Omega$, aby její arita odpovídala vstupnímu stupni. Hodnota každého takového vrcholu v je potom rovna funkční hodnotě této funkce aplikované na vrcholy, z nichž vede hrana do v.

Velikostí booleovského obvodu se rozumí počet všech vrcholů, značíme $|C| = s$. *Hloubkou obvodu* myslíme délku nejdelší cesty mezi nějakým vstupním a výstupním vrcholem.

Řekneme, že obvod C *počítá (reprezentuje)* funkci f, jestliže na stejných

vstupech vrací stejné výstupní hodnoty jako f.

Nechť $f : \{0,1\}^n \to \{0,1\}$. Pak $C_\Omega(f)$ je velikost nejmenšího obvodu v bázi Ω reprezentujícího funkci f. Nechť $g : \{0,1\}^* \to \{0,1\}$ a $h : \mathbb{N} \to \to \mathbb{N}$, pak říkáme, že g má *obvodovou složitost* h, jestliže pro každé n platí $C_\Omega(g|_{\{0,1\}^n}) = h(n)$.



Obr. 2.1. Jednoduchý booleovský obvod počítající paritu má velikost 7 a hloubku 2

Obyčejně se jako báze Ω volí de Morganova báze $\{\wedge, \vee, \neg\}$, tedy konjunkce, disjunkce a negace, nebo algebraická báze $\{\oplus, \wedge\}$, odpovídající operacím součet a součin nad tělesem $\mathbb{Z}_2$.

Definice 2.7. Báze Ω je *úplná*, jestliže libovolná booleovská funkce může být počítána booleovským obvodem v bázi Ω.

Věta 2.8 (souvislost mezi formulemi a obvody)**.** *Ať C je obvod v de Morganově bázi mající pouze jeden výstupní vrchol. Pak lze tento obvod převést na výrokovou formuli φ. Naopak libovolnou výrokovou formuli lze chápat jako takový obvod.*

Důkaz. Jednotlivé podformule výrokové formule vytvoříme vždy ze všech vrcholů, které vedou do příslušného vrcholu. Má-li některý vrchol fanout vyšší než jedna, pak je třeba celý term, který jej reprezentuje, použít ve formuli vícekrát.

Z formule budeme postupně vytvářet obvod podle jejího hierarchického uspořádání. Můžeme navíc využít toho, že v obvodu lze jeden výsledek použít vícekrát. Tedy pokud se ve formuli nějaký term opakuje, v obvodu z něj stačí vést více hran a nemusí se počítat celý znovu. □

Předchozí tvrzení společně s větou 2.4 tedy říká, že de Morganova báze

je úplná. Díky de Morganovým pravidlům z lemmatu 2.2 ji můžeme ještě zmenšit a stále bude úplná. Máme na mysli bázi $\{\wedge, \neg\}$ a také $\{\vee, \neg\}$. Navíc lze každý obvod v de Morganově bázi přetransformovat tak, aby negující brány byly již na nejnižší úrovni obvodu, tedy hned nad vstupními proměnnými. Tímto přechodem se nijak nezvýší hloubka obvodu a jeho velikost se přinejhorším zdvojnásobí. Předpokládejme nejprve, že z hradla vychází pouze jedna hrana. Jestliže vede do negace, pak stačí aplikovat výše zmíněná pravidla: nahradit jednu operaci tou druhou ($\wedge$ za $\vee$ a naopak) a negaci přesunout před tento vrchol. Místo jedné negující brány však v obvodu máme dvě, ale o jednu úroveň níže. Ale pokud z takto přestavovaného vrcholu vycházela ještě aspoň jedna hrana bez negace, je třeba celý příslušný podobvod zkopírovat, abychom neovlivnili hodnotu vstupující do těchto hran a uzlů.

Rovněž algebraická báze je úplná. To díky existenci Walsh-Hadamardovy transformace, o níž se zmiňuje například Preneel [11, kapitola 8].

2.2 Obvodová složitost

Nyní se vrátíme opět k třídám složitosti a Turingovým strojům. Jak ukázal J. E. Savage v roce 1972 [14], existuje souvislost mezi výpočetní složitostí Turingova stroje a obvodovou složitostí. *Obvodovou složitostí jazyka* rozumíme obvodovou složitost jeho charakteristické funkce. Na Savageovu práci navázali Pippenger a Fischer, kteří dokázali následující tvrzení [10].

Věta 2.9. *Jestliže jazyk L je rozhodnutelný v čase $T(n)$, neboli $L \in \in \mathsf{TIME}(T(n))$, pak má obvodovou složitost $O(T(n)\log(T(n)))$.*

Důkaz. Dokážeme pouze slabší odhad $O(T^4(n))$. K tomu nejprve předpokládejme, že stroj M rozhoduje jazyk L v čase $T(n)$. Konvertováním na jednopáskový stroj spotřebujeme nejvýše $O(T^2(n))$ kroků, neboť každá páska může mít maximálně délku $T(n)$ (jinak by ji stroj ani nemohl celou přečíst).

Výpočet jednopáskového stroje si lze představovat jako tabulku, v níž i-tý řádek vyjadřuje stav pásky v čase i. Snadno nahlédneme, že k výpočtu jednoho políčka na pozici (i, j) stačí znát obsah tří polí bezprostředně nad ním, tedy $(i-1, j-1)$, $(i-1, j)$ a $(i-1, j+1)$, neboť v jednom kroku se hlava může posunout maximálně o jedno políčko. Pro každou buňku tabulky proto můžeme zkonstruovat jednoduchý obvůdek konstantní velikosti. Buněk v tabulce je však $O(T^4(n))$. □

Zřejmým důsledkem této věty je, že každý jazyk, který je v P má rovněž polynomiální obvodovou složitost. Opačná implikace bohužel neplatí. To lze nahlédnout následující úvahou. Mějme jazyk L definovaný pomocí nerekurzivní množiny[1] $A \subset \mathbb{N}$ následovně: $x \in L \Leftrightarrow |x| \in A$. Tento jazyk je algoritmicky nerozhodnutelný, ale přitom obvod, který jej počítá, stačí pouze konstantní velikosti. Podle délky vstupu n bude mít na výstupním vrcholu hodnotu 0, nebo 1 v závislosti na tom, zda $n \in A$.

Tyto úvahy vedou k myšlence *uniformního* výpočetního modelu. Zatímco Turingův stroj je velmi univerzální početní stroj, kterému můžeme zadávat různě velké vstupy, tak booleovské obvody tuto vlastnost nemají. Každý obvod musí dostat vstup délky, pro kterou byl postaven. Pokud tedy chceme pomocí obvodů simulovat činnost nějakého Turingova stroje, potřebujeme posloupnost obvodů $\{C_n\}_{n\in\mathbb{N}}$ pro různě dlouhé vstupy. Právě takovéto výpočetní modely nazýváme *neuniformní*.

Na chvíli teď odskočíme od konkrétních jazyků a jejich obvodů a pokusíme se odpovědět na otázku, zda existuje nějaký univerzální odhad na velikost obvodů booleovských funkcí. Roku 1956 D. E. Muller [9] dokázal následující exponenciální odhad.

Věta 2.10. *Téměř každá booleovská funkce n proměnných vyžaduje obvod v de Morganově bázi (navíc takové, že konjunkce i disjunkce mají fanin právě dva) o velikosti $\Omega(2^n/n)$.*

Důkaz. Označme s velikost obvodu a zvolme nějaký vrchol, do něhož vedou dvě vstupní hrany. Každou z nich můžeme zvolit následovně: buďto vede z jiného vrcholu, nebo ze vstupní proměnné, nebo její negace, nebo je konstantní. Celkem máme $(s + 2n + 2)$ možností, jak zvolit jednu hranu. Navíc máme dvě funkce, kterými můžeme vrchol označit. Proto celkem existuje $2(s + 2n + 2)^2$ způsobů, kterak vytvořit jeden vrchol. Celkem je jich s, a proto existuje

$$\left(2(s + 2n + 2)^2\right)^s$$

různých obvodů vyhovujících podmínkám věty.

Konečně si všimneme, že pro $s = 2^n/10n$ je obvodů přibližně $\sqrt[5]{2^{2^n}} \ll 2^{2^n}$, kde na pravé straně nerovnosti je počet booleovských funkcí n proměnných.

[1] Takové množiny skutečně existují, neboť víme, že všech podmnožin přirozených čísel je nespočetně mnoho, ale všech Turingových strojů je pouze spočetně mnoho. Příkladem nerekurzivní množiny je jazyk HALT kódující tzv. halting problem.

Takže skoro všechny takové funkce vyžadují obvody velikosti alespoň $2^n/10n$. □

Bohužel tato věta má k řešení problému P vs. NP daleko. Říká sice, že velmi mnoho funkcí má exponenciální složitost, ale tyto funkce mohou být úplně libovolné. Asi nepřekvapí, že funkce, která je ve třídě EXPTIME bude mí exponenciálně velký obvod, ale proč by jej měla mít i funkce, která je jen NP-těžká?

Je tedy třeba zaměřit se na konkrétní funkce, o nichž víme, jak vypadají, a vytvářet spodní odhady jejich složitosti. Avšak jakkoli jsou tyto odhady významné a důležité, pro obecné obvody zatím máme k dispozici jen lineární omezení. Dle [2] předvedeme konstrukci spodního odhadu pro *prahovou funkci* $\mathrm{TH}_{k,n}$, která vrací hodnotu 1, právě tehdy když alespoň k z n vstupních bitů je rovno jedné.

Stejně jako v předešlém tvrzení i zde budeme pracovat s obvody v de Morganově bázi s binárními operacemi konjunkce a disjunkce a unární negací.

Věta 2.11. *Pro každé $n \geq 2$ prahová funkce $\mathrm{TH}_{2,n}$ potřebuje obvod velikosti nejméně $2n - 4$.*

Důkaz. V důkazu se využívá jednoduchý trik eliminace vrcholů. Postupujeme indukcí dle n. Pro $n = 2$ a $n = 3$ je tvrzení jasné. Obvod pro tři proměnné bude jistě obsahovat více než dva vrcholy, což lze nahlédnout prostým uvážením všech možností.

Nyní buď C optimálním obvodem pro funkci $\mathrm{TH}_{2,n}$. Předpokládejme, že vrchol v je nejníže situovaným (ve smyslu hierarchie) pracujícím s proměnnými x_i a x_j. Ty mohou být ohodnoceny čtyřmi způsoby. V závislosti na tom pak zbytku obvodu stačí počítat funkce $\mathrm{TH}_{0,n-2}$, $\mathrm{TH}_{1,n-2}$, resp. $\mathrm{TH}_{2,n-2}$. Aby však obvod C byl schopný rozlišit, která ze situací nastala, musí alespoň z jedné proměnné x_i nebo x_j vycházet ještě jedna hrana. Ať vychází z x_i. Pokud nastavíme tuto proměnnou konstantně na 0, tak v obvodu můžeme eliminovat nejméně dva vrcholy, které s ní počítají. Zároveň ale obvod pracuje jen s $n-1$ vstupy a řeší tedy funkci $\mathrm{TH}_{2,n-1}$. Dle indukčního předpokladu je tedy jeho velikost $2(n-1)-4$. Ovšem z původního obvodu C jsme nejméně dvě brány odebrali, tudíž jeho velikost byla aspoň $2n - 4$. □

Zatím se však nepodařilo najít podstatně silnější odhady pro obecné

obvody. Všechny dosud dokázané odhady jsou pouze lineárního řádu, tedy $O(n)$. Například Iwama, Lachish, Morizumi a Raz [7] představili spodní odhad $5n - o(n)$ pro (n,k)-*silně*-2-*závislou funkci*. Uveďme nyní její definici. Nechť je dána funkce $F : \{0,1\}^n \to \{0,1\}$ a pro každé $1 \leq i < j \leq n$, $a, b \in \{0,1\}$ je funkce $F[i,j,a,b]$ restrikcí $F|_{\vartheta[i,j,a,b]}$, kde $\vartheta[i,j,a,b] : x_i \mapsto a$ a $x_j \mapsto b$. Řekneme, že funkce F je 2-*závislá*, když pro libovolné i a j, $1 \leq i < < j \leq n$ jsou funkce $F[i,j,0,0]$, $F[i,j,0,1]$, $F[i,j,1,0]$ a $F[i,j,1,1]$ navzájem různé. Buď dále $X_m \subseteq \{x_1, \ldots, x_n\}$ množina velikosti m a $\vartheta_m : X_m \to \{0,1\}$. Konečně funkce F je (n,k)-silně-2-závislá, právě tehdy když $F|_{\vartheta_m}$ je 2-závislá pro každé m, které splňuje $0 \leq m \leq n - k$, pro každou podmnožinu X_m a libovolné ϑ_m. Závěrem poznamenejme, že existují i polynomiální algoritmy, které takovou funkci dokáží explicitně konstruovat.

Kapitola 3

Obvody s omezenou hloubkou

3.1 AC, NC hierarchie

V této kapitole představíme některé základní odhady pro speciální třídu obvodů – obvody s omezenou hloubkou. Jak název napovídá, budeme se zabývat obvody, které řeší určitý problém, ale jejichž hloubku předem explicitně omezíme.

Dále budeme všechny obvody uvažovat v následujícím tvaru. Každá hladina obvodu obsahuje pouze jeden typ logické operace (buď $\wedge$, nebo $\vee$; obojí libovolné arity) a tyto se po hladinách střídají. Další podmínkou je umístění negací – tyto smí být v obvodu jen na nejnižší hladině, tj. přímo u proměnných.

Symbolem Σ_d^S rozumíme obvod mající na výstupu operaci $\vee$, hloubku d a velikost nejvýše S. Podobně znakem $\Sigma_d^{S,t}$ myslíme obvod hloubky $d + 1$ takový, že všechny operace na nejnižší hladině mají fanin nejvýše t. Terminologicky říkáme, že obvod je *t-otevřený*, jestliže je $\Sigma_1^{S,t}$ pro nějaké $S \in \mathbb{N}$, tj. jde o disjunkci $S - n - 1$ konjunkcí arity nejvýše t. Obdobně se definují obvody typu Π_d^S a $\Pi_d^{S,t}$, pokud mají na výstupu operaci $\wedge$. Obvody, které jsou ve tvaru $\Pi_1^{S,t}$, nazýváme *t-uzavřené*.

Na obvody typů Σ a Π budeme v následujícím textu hledět jako na své vzájemné duály.

Když máme vybudovány tyto pojmy, budeme je používat i pro označení booleovských funkcí definovaných pomocí příslušného typu obvodu.

Zaveďme ještě značení

$$\Sigma_d^{\mathrm{poly}} = \bigcup_i \Sigma_d^{n^i} \qquad \text{a} \qquad \Sigma_d^{\mathrm{poly,const}} = \bigcup_i \Sigma_d^{n^i,i}\,.$$

Definice 3.1. Definujeme složitostní třídy

$$\mathsf{NC}^0 = \bigcup_d \Sigma_d^{\mathrm{poly,const}} \qquad \text{a} \qquad \mathsf{AC}^0 = \bigcup_d \Sigma_d^{\mathrm{poly}}$$

a dále induktivně celou hierarchii NC^i a AC^i pro $i \geq 0$. Jde o množiny funkcí vypočítatelných polynomiálně velkými obvody (ve smyslu výše uvedených konvencí) s hloubkou $O(\log^i n)$ a konstantními, resp. neomezenými faniny.

Užitím předchozích definic lze velmi snadno ověřit platnost inkluze

$$\mathsf{AC}^0 \subseteq \mathsf{NC}^1\,.$$

Stačí vzít libovolné hradlo AC^0 obvodu, které nemá konstantou omezený fanin, a přepsat jej pomocí více shodných hradel s konstantním faninem. Tím místo jednoho hradla obdržíme malý obvůdek, který má však zjevně logaritmickou hloubku. Provedeme-li takovou transformaci se všemi hradly, převedeme obvod z typu AC^0 na typ NC^1.

Není však zřejmé, zda je tato inkluze ostrá, či nikoli. Na konci této kapitoly ukážeme, že platí dokonce jako ostrá.

3.2 Håstadovo přepínací lemma

Pro klíčový důkaz budeme potřebovat techniku tzv. *restrikcí*. Jde o jakési částečné ohodnocení vstupních proměnných. Některým přiřadíme hodnotu konstant (0 nebo 1) a ostatní necháme i nadále volné jako proměnné (tento fakt budeme značit symbolem $*$).

Definice 3.2. Nechť $X = \{x_1, \ldots, x_n\}$ jsou vstupní proměnné obvodu C počítajícího booleovskou funkci f. Restrikcí ϱ rozumíme zobrazení z X do $\{0, 1, *\}$. Funkci $f|_\varrho$, resp. obvod $C|_\varrho$ nazýváme *indukované* restrikcí ϱ.

Nyní se na tento pojem podíváme ještě z pravděpodobnostního hlediska. Volme pevné p, $0 < p < 1$. Označíme R_p systém restrikcí na množině X, které každému $x_i \in X$, $0 \leq i \leq n$ přiřadí nezávisle jednu z hodnot z množiny

$\{0, 1, *\}$ s pravděpodobnostmi $\Pr[\varrho(x_i) = *] = p$ a $\Pr[\varrho(x_i) = 0] = \Pr[\varrho(x_i) = 1] = (1 - p)/2$.

Nyní můžeme vyslovit přepínací lemma, které dokázal roku 1985 Johan Håstad [6] a které říká, že lze „přepínat“ mezi konjunktivními a disjunktivními obvody, aniž bychom je přitom s významnou pravděpodobností nějak zásadně (v asymptotickém smyslu) zvětšili.

Lemma 3.3 (přepínací)**.** *Nechť f je t-uzavřená funkce n proměnných. Ať ϱ je náhodně zvolená restrikce z prostoru R_p. Potom*

$$\Pr\left[f|_\varrho \text{ není } s\text{-otevřená }\right] \leq \left(\frac{4pt}{\log 2}\right)^s . \tag{3.1}$$

Je patrné, že nerovnost (3.1) má smysl pouze tehdy, pokud je pravděpodobnost p dostatečně malá. V opačném případě bychom totiž na pravé straně mohli obdržet číslo větší než jedna, což je ovšem vzhledem k definici pravděpodobnosti splněno vždy a přínos lemmatu by nebyl vůbec žádný. S odvoláním na silný zákon velkých čísel proto můžeme uvažovat pouze ty restrikce, které udělují relativně málo hvězdiček (jejich počet budeme v dalším textu označovat l; tedy $l < n/2$).

Původní Håstadův důkaz lemmatu zásadním způsobem využíval poznatků o podmíněných pravděpodobnostech, ale roku 1993 publikoval Alexander Razborov důkaz odlišný [12, App. E.4], který je možné v teorii booleovských obvodů použít poněkud obecněji. Proto jej v tomto textu upřednostníme. Nejprve definujme množinu $H(r, u)$ všech konečných posloupností $\beta = (\beta_1, \ldots, \beta_k)$, přičemž každé β_i je slovo délky r skládající se ze znaků „$*$“, „0“ a „1“ a vždy obsahující aspoň jednu jedničku. Navíc celkový počet jedniček ve všech β_i dohromady je roven číslu u. Dokažme nyní o velikostech takových množin jednoduché kombinatorické tvrzení.

Lemma 3.4. *Pro libovolná celá čísla $r, u \geq 0$ platí*

$$|H(r, u)| < \left(\frac{r}{\log 2}\right)^u .$$

Důkaz. Definujme číslo γ tak, aby splňovalo $(1 + 1/\gamma)^r = 2$. Indukcí dle u dokážeme, že pro takto zvolenou hodnotu platí $|H(r, u)| \leq \gamma^u$. Pro $u = 0$ je tvrzení zřejmě platné, předpokládejme tedy, že máme $u > 0$.

Podle definice platí $\beta \in H(r, u)$, jestliže existuje „rozklad“ $\beta = (\beta_1, \beta')$, v němž β_1 má $i \leq u$ hvězdiček a $\beta' \in H(r, u - i)$. Využijeme indukčního

předpokladu a jednoduché úvahy, že ve slově β_1 lze i hvězdiček rozmístit $\binom{r}{i}$ způsoby. Proto s použitím binomické věty a definice čísla γ můžeme počítat

$$\begin{aligned} |H(r,u)| &= \sum_{i=1}^{\min\{r,u\}} \binom{r}{i} |H(r,u-i)| \leq \sum_{i=1}^{r} \binom{r}{i} \gamma^{u-i} = \\ &= \gamma^u \sum_{i=1}^{r} \binom{r}{i} \frac{1}{\gamma^i} = \gamma^u \left(\left(1+\frac{1}{\gamma}\right)^r - 1 \right) = \gamma^u \,. \end{aligned}$$

Zbývá použít definici exponenciální funkce a dokončit důkaz. Víme

$$2 = \left(1+\frac{1}{\gamma}\right)^r < \left(\mathrm{e}^{1/\gamma}\right)^r \,,$$

a proto $\gamma < r/\log 2$, což bylo dokázati. □

Nechť symbol $\mathcal{R}_n^l$ značí systém všech restrikcí na n proměnných, které přidělují právě l hvězdiček. Dále označíme $B_l^s(f) \subseteq \mathcal{R}_n^l$ ty „špatné“ restrikce, které funkci f ohodnotí vstupní proměnné tak, že se z ní nestane s-otevřená funkce, a ukážeme, že takových restrikcí je oproti všem relativně málo.

K provedení samotného důkazu najdeme zobrazení mezi $B_l^s(f)$ a nějakou dostatečně malou množinou.

Lemma 3.5. *Nechť f je t-uzavřená funkce a $l > s$ jsou celá kladná čísla, pak existuje prosté zobrazení*

$$F: \quad B_l^s(f) \to \mathcal{R}_n^{l-s} \times H(t,s) \,.$$

Důkaz. Mějme $f = \bigwedge_{i=1}^{h} C_i$, kde C_i jsou klauzule velikosti nejvýše t, a zvolme libovolnou restrikci $\varrho \in B_l^s(f)$ a ohodnocení π funkce $f|_\varrho$, které přiřazuje hodnoty alespoň s zbylým proměnným tak, aby $f|_{\pi\varrho} \equiv 1$.

Rekurzivně vybudujeme posloupnost částečných ohodnocení $\pi_1, \pi_2, \ldots$, která rozděluje π na malé části. Předpokládejme, že již máme $\pi_1, \ldots, \pi_{i-1} \subset$ $\subset \pi$, přičemž jednotlivá π_j mají navzájem různé definiční obory, a jako celek je tato podposloupnost různá od π. Aplikujme restrikci $\pi_{i-1} \ldots \pi_1 \varrho$ na disjunkce $C_1, \ldots, C_h$ a najděme nejmenší index ν_i, který splňuje $C_{\nu_i}|_{\pi_{i-1}\ldots\pi_1\varrho} \not\equiv$ $\not\equiv 1$. Vzhledem k tomu, že $\pi_{i-1} \ldots \pi_1 \neq \pi$, tak takové ν_i musí existovat (v ohodnocení zůstávají „dole“ ještě nějaké volné proměnné). Speciálně také $f|_{\pi_{i-1}\ldots\pi_1\varrho} \not\equiv 1$. Označme množinu proměnných klauzule C_{ν_i} písmenem T_i

a symbolem Y_i její podmnožinu, která obsahuje pouze ty proměnné ohodnocené restrikcí π, ale ne restrikcí $\pi_{i-1} \dots \pi_1$. Zjevně $Y_i \neq \emptyset$, neboť $f|_{\pi\varrho} \equiv 1$, což by pro $Y_i = \emptyset$ vedlo na $C_{\nu_i}|_{\pi\varrho} \equiv 1$, ale my jsme předpokládali opak. Můžeme proto definovat $\pi_i = \pi|_{Y_i}$.

Najděme minimální $k \in \mathbb{N}$, že $\pi_1, \dots, \pi_k$ ohodnocují alespoň s proměnných, a ořízněme z π_k ty poslední proměnné, aby stále platilo $C_{\nu_k}|_{\pi_1,\dots,\pi_k} \equiv 1$, ale přitom $\pi_1, \dots, \pi_k$ obsahovaly dohromady přesně s proměnných. To lze jistě udělat, neboť stačí, aby π_k přiřazovalo právě jedné proměnné hodnotu 1. Ostatní můžeme odebrat.

Nechť $\check{\pi}_i$ je jednoznačně určené ohodnocení, které má stejný definiční obor jako π_i (tj. Y_i), ale nezajišťuje ohodnocení hradla C_{ν_i} hodnotou 1. Definujeme restrikci

$$F_1(\varrho) = \check{\pi}_k \dots \check{\pi}_1 \varrho \,,$$

která je jistě prvkem rodiny $\mathcal{R}_n^{l-s}$, protože všechny dílčí restrikce $\check{\pi}_j$ mají navzájem disjunktní definiční obory.

Dále budeme definovat slova $\sigma^i \in \{0, 1, *\}^t$ pro $1 \leq i \leq k$ následujícím postupem. Prvky množiny T_i označíme $\{x_{\vartheta(i,1)}, \dots, x_{\vartheta(i,t_i)}\}$. Díky vlastnostem funkce f platí $t_i \leq t$. Navíc požadujme, aby platilo $\vartheta(i,1) < \dots < \vartheta(i,t_i)$. Znak na j-té pozici slova σ^i nechť je

$$\sigma^i_j = \begin{cases} \pi_i(x_{\vartheta(i,j)}) \oplus \check{\pi}_i(x_{\vartheta(i,j)}) \,, & \text{když } j \leq t_i \text{ a } x_{\vartheta(i,j)} \text{ je v def. oboru } \pi_i \,, \\ * & \text{jinak} \,. \end{cases}$$

Vzhledem k tomu, jak jsme definovali $\check{\pi}_i$, tak společně s π_i mají tyto restrikce různý efekt na disjunkci C_{ν_i}, a tedy i na alespoň některé proměnné, které v ní vystupují, a proto pro každé i lze nalézt takové j, že platí $\sigma^i_j = 1$. Protože $\pi_1, \dots, \pi_k$ ohodnocují právě s proměnných, vidíme $(\sigma^1, \dots, \sigma^k) \in H(t,s)$. Zaveďme označení

$$F_2(\varrho) = (\sigma^1, \dots, \sigma^k)$$

a hledané zobrazení F definujme předpisem

$$F: \quad \varrho \mapsto (F_1(\varrho), F_2(\varrho)) \,.$$

Abychom dokázali, že jde o injekci, předvedeme, jakým způsobem je možné ze zadané dvojice $(F_1(\varrho), F_2(\varrho))$ získat zpět restrikci ϱ. Najdeme nejprve všechny dvojice $(\pi_j, \check{\pi}_j)$. Se znalostí každé z hodnot π_j a $\check{\pi}_j$ se ϱ obdrží již snadno.

Stejně jako v první části důkazu postupujme indukcí dle i. Předpokládejme, že již známe všechny $(\pi_j, \check{\pi}_j)$ pro $j < i$. Pak ovšem vzhledem k definici symbolu $F_1(\varrho)$ známe rovněž i restrikci $\check{\pi}_k \dots \check{\pi}_i \pi_{i-1} \dots \pi_1 \varrho$. Jenže index ν_i jsme definovali jako nejmenší s vlastností $C_{\nu_i}|_{\check{\pi}_k \dots \check{\pi}_i \pi_{i-1} \dots \pi_1 \varrho} \not\equiv 1$ (restrikce $\check{\pi}_j$, $i \le j \le k$, na hodnotu C_{ν_i} nemají žádný vliv, protože ji neohodnocují jako identicky pravdivou klauzuli, resp. mají jiný definiční obor). To však znamená, že pro všechny indexy $\nu < \nu_i$ je $C_\nu \equiv 1$, neboť ν_i byl první, pro který tato identita neplatila.

Již tedy známe i hodnotu ν_i. Jsme proto schopni určit i T_i. Ze znalosti σ^i a T_i odvodíme i definiční obor restrikce $\check{\pi}_i$ (a rovněž π_i) a dle $F_1(\varrho)$ získáme hodnoty $\check{\pi}_i$. S použitím σ^i pak zrekonstruujeme π_i. □

Abychom dokončili Razborovův důkaz Håstadova přepínacího lemmatu, stačí nám provést výpočet

$$\begin{aligned}
\Pr[f|_\varrho \text{ není } s\text{-otevřená}] &= \frac{|B_l^s(f)|}{|\mathcal{R}_n^l|} \le \\
&\overset{\text{(lemma 3.5)}}{\le} \frac{|\mathcal{R}_n^{l-s}| \cdot |H(t,s)|}{\mathcal{R}_n^l} \le \\
&\overset{\text{(lemma 3.4)}}{\le} \frac{\binom{n}{l-s} 2^{n-l+s}}{\binom{n}{l} 2^{n-l}} \left(\frac{t}{\log 2}\right)^s,
\end{aligned}$$

odkud budeme pokračovat pro $l < n/2$, jak jsme vysvětlovali na začátku sekce. Pak ovšem pro velká n intuitivně platí

$$\binom{n}{l-s} \ll \binom{n}{l}.$$

Odhadněme teď poměr těchto binomických koeficientů lépe! Pro lepší orientaci využijeme tzv. sestupných mocnin $x^{(k)} = x(x-1)\dots(x-k+1)$.

$$\begin{aligned}
\frac{\binom{n}{l-s}}{\binom{n}{l}} = \frac{\frac{n!}{(l-s)!\,(n-l+s)!}}{\frac{n!}{(n-l)!\,l!}} &= \frac{(n-l)!\,l!}{(l-s)!\,(n-l+s)!} = \\
&= \frac{l^{(s)}}{(n-l+s)^{(s)}} \le \left(\frac{l}{n-l}\right)^s.
\end{aligned}$$

Ještě před tím, než dosadíme předchozí odhad do hlavního výpočtu, si vzpomeneme, že $l = pn$, a tedy také $p < 1/2$. Dokázali jsme

$$\Pr[f|_\varrho \text{ není } s\text{-otevřená}] \leq \left(\frac{2t}{\log 2}\right)^s \left(\frac{p}{1-p}\right)^s \leq \left(\frac{4pt}{\log 2}\right)^s .$$

3.3 Spodní odhad pro paritu

V této sekci využijeme Håstadovo lemma ke konstrukci exponenciálního dolního odhadu pro velikost obvodu počítajícího paritu. Tato funkce vrací hodnotu 1, pokud je na vstupu lichý počet jedniček, resp. 0 v opačném případě. Budeme ji značit PARITY_n, kde indexem n rozumíme počet vstupujících proměnných (zpravidla jej vynecháváme). Pro klíčový odhad nejprve odvodíme lehký důsledek přepínacího lemmatu.

Důsledek 3.6. *Pro libovolné p, d, $0 \leq k \leq d-1$ a funkci f typu $\Sigma_d^{S,t}$ platí následující: Jestliže $\varrho \in R_{p^k}$, potom*

$$\Pr\left[f|_\varrho \textit{ není } \Sigma_{d-k}^{S,t}\right] < S\left(\frac{4pt}{\log 2}\right)^t .$$

Důkaz. Označme celý obvod počítající funkci f písmenem C. Uvažme prostor $R_{p,k}$ všech k-tic restrikcí, ze kterého vyberme prvek $(\varrho_1, \ldots, \varrho_k)$. Každá složka tohoto vektoru je částečné ohodnocení, které s pravděpodobností p přidělí proměnné symbol hvězdičky. Přitom je nutné, aby restrikce ϱ_{i+1} pracovala pouze s těmi proměnnými, kterým byly předchozími i restrikcemi vždy uděleny hvězdičky. Snadno se lze přesvědčit, že po aplikaci všech ohodnocení z $(\varrho_1, \ldots, \varrho_k)$ zbude $p^k n$ volných proměnných, což je stejné množství jako při užití $\varrho \in R_{p^k}$.

Postupným restringováním obdržíme množinu funkcí $f_1, \ldots, f_{k+1}$, pro které platí $f_{i+1} = f_i|_{\varrho_i}$, přičemž $f_1 = f$. Přitom každá z těchto funkcí zahrnuje spodní podobvody uvažovaného obvodu C, které jsou buďto t-otevřené, nebo t-uzavřené, což plyne z přepínacího lemmatu. Avšak podle uvedeného tvrzení k tomu dochází jen s určitou pravděpodobností. Naopak pravděpodobnost, že proces omezování selže, je nejvýše $(4pt/\log 2)^t$, ale protože dílčích podobvodů je nejvýše $|C| = S$, výsledek ještě přenásobíme touto hodnotou a získáme potřebné tvrzení. □

Důsledek 3.7. *Je-li f typu $\Sigma_d^{S,t}$, $t \geq \log_2 S$, potom existuje restrikce ϱ přiřazující alespoň*

$$\frac{n}{3\,(16t)^{d-1}} - t \tag{3.2}$$

hvězdiček tak, že $f|_\varrho$ je konstantní.

Důkaz. Nejdříve uvažme, pro jak velká n má výraz v (3.2) kladnou hodnotu. To platí pro $n > 3{\cdot}16^{d-1}t^d$. Dosaďme do předešlého důsledku $p = 1/16t$ a $k = d-1$; restrikci tedy hledáme v množině $R_{p^{d-1}}$. Obdržíme nerovnost

$$\Pr\left[f|_\varrho \text{ není } \Sigma_1^{S,t}\right] < S\left(\frac{4pt}{\log 2}\right)^t \leq \left(\frac{1}{2\log 2}\right)^t \leq 0{,}73^t \leq 0{,}73\,,$$

kde jsme využili předpokladu $t \geq \log_2 S$ a výsledek zaokrouhlili nahoru.

Počet udělených hvězdiček můžeme považovat za náhodnou veličinu X_n s binomickým rozdělením s parametry n a p. Střední hodnota konverguje k np a platí centrální limitní věta, která říká, že vzhledem k našemu omezení $n > 3 \cdot 16^{d-1}t^d$ je

$$\Pr\left[\varrho \text{ uděluje méně než } \tfrac{1}{3}\,np^{d-1} \text{ hvězdiček}\right] \leq \Phi(-1) \doteq 0{,}16\,.$$

Zde Φ je distribuční funkce normálního rozdělení a její číselná hodnota je opět zaokrouhlena nahoru.

Vzhledem k tomu, že $0{,}73 + 0{,}16 < 1$, tak pravděpodobnost, že restrikce převede funkci f do tvaru $\Sigma_1^{S,t}$ a zároveň udělí aspoň $np^{d-1}/3$ hvězdiček, je kladná. Jinak řečeno mezi všemi restrikcemi musí existovat nějaká, která vyhovuje této podmínce.

Nyní stačí zajistit, aby vzniklá funkce $f|_\varrho$ byla konstantní. To se však provede snadno, když si uvědomíme, že pro obvody typu $\Sigma_1^{S,t}$ stačí zvolit jednu t-tici vstupních proměnných vstupujících do stejné konjunkce (ta má fanin nejvýše t) a zafixovat je. Přidáním takového mapování k již nalezené restrikci tedy dokážeme tvrzení. □

Věta 3.8. *Pro žádné $n, d > 0$ není funkce* PARITY *typu $\Sigma_d^{S,\log_2 S}$, kde $S < 2^{n^{1/d}/16}$.*

Důkaz. Pro důkaz sporem předpokládejme, že parita je typu $\Sigma_d^{S,\log_2 S}$ a její obvodová složitost je shora omezena výrazem $2^{n^{1/d}/16}$. Potom však dle předchozího důsledku existuje restrikce ϱ, která nechává aspoň jednu proměnnou

volnou, ale přitom zajišťuje, že parita je konstantní. To je ovšem zjevný spor, neboť parita vždy závisí na všech vstupních bitech. □

Důsledek 3.9.

$$\text{PARITY} \notin \mathsf{AC}^0 .$$

Ukázali jsme, že obvody, které by počítali paritu vstupu, musejí být větší nežli obvody typu $\Sigma^{\text{poly}}_{\text{const}}$; podrobněji řečeno paritu se nedá počítat obvodem, který by měl polynomiálně mnoho hradel vzhledem k délce vstupu a přitom měl omezenou hloubku. V další sekci toto pozorování využijeme a ukážeme, že podobnou náročnost má i řada dalších úloh.

Nejdříve ale dokončíme řešení motivačního problému ze začátku kapitoly. Funkci parity lze počítat pomocí obvodu v podobě binárního stromu s hradly $\oplus$. Avšak zjevně platí $a \oplus b = (\neg a \wedge b) \vee (a \wedge \neg b)$. Navíc binární strom pro n listů má hloubku $O(\log n)$. Dokázali jsme PARITY $\in \mathsf{NC}^1$. Dohromady s předchozím důsledkem tedy získáváme

$$\mathsf{AC}^0 \subset \mathsf{NC}^1 .$$

3.4 AC^0 reducibilita

Poměrně silným nástrojem, který umožňuje provádět dolní odhady pro celou škálu problémů, je AC^0 reducibilita. Využívá klasické metody převedení jednoho problému na jiný. Furst, Saxe a Sipser [5] tento postup zavádějí a poskytují několik příkladů.

Definice 3.10. Booleovská funkce f je AC^0*-reducibilní* na booleovskou funkci g (značíme $f \leq_{\mathsf{AC}^0} g$), pokud lze její výpočet realizovat obvodem omezené hloubky a polynomiální velikosti, který se skládá z hradel $\vee$, $\wedge$ a hradla počítajícího funkci g (tuto operaci považujeme za výpočetně stejně náročnou jako zbylé dvě).

Uvedená definice nám ihned umožňuje vyslovit snadné pozorování.

Lemma 3.11. *Pokud je* PARITY AC^0*-reducibilní na funkci* g*, potom* $g \notin \mathsf{AC}^0$.

Důkaz. Uvažme, kdyby $g \in \mathsf{AC}^0$. Protože PARITY je dle předpokladu AC^0-reducibilní na g, tak dostaneme dostaneme pro paritu polynomiálně

velký obvod omezené hloubky, což je spor, neboť z předchozí sekce víme, že PARITY $\notin \mathsf{AC}^0$. □

V následujících odstavcích toto tvrzení šikovně využijeme. Nejprve k důkazu, že jazyk MAJ rovněž nelze počítat polynomiálně velkými obvody s omezenou hloubkou. Nechť máme n proměnných $x_1, \ldots, x_n$. Položíme

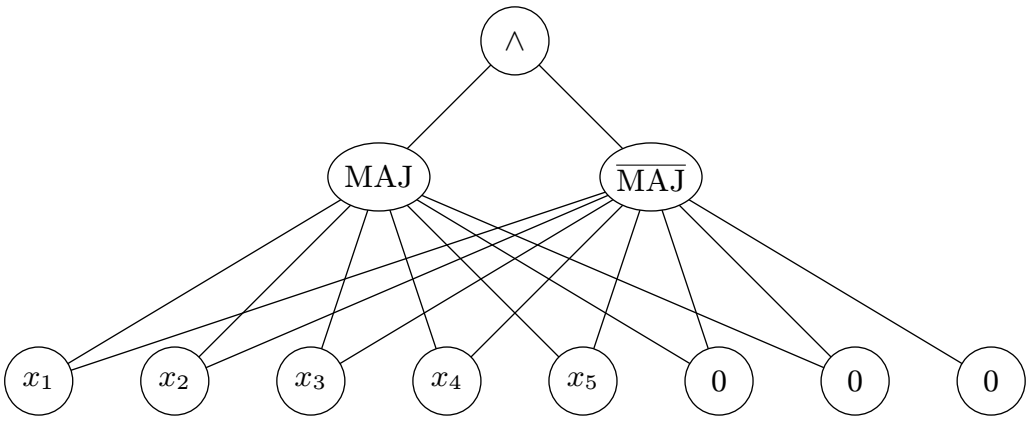
$$\mathrm{MAJ}(x_1, \ldots, x_n) = \begin{cases} 1\,, & \text{pokud } \sum_{i=1}^{n} x_i > n/2\,, \\ 0 & \text{jinak.} \end{cases}$$

Jinými slovy tato funkce vrací „většinový názor“ svých vstupních proměnných[1].

Věta 3.12.

$$\mathrm{PARITY} \leq_{\mathsf{AC}^0} \mathrm{MAJ}\,.$$

Důkaz. Chceme vytvořit obvod, který bude počítat paritu pomoci konjunkcí, disjunkcí a hradel, ve kterých je implementována funkce většiny. Pomocí zmíněných hradel dokážeme vytvořit obvod P_k, který bude vracet hodnotu 1, právě když přesně k vstupních bitů bude rovných jedné. Pak zjevně $\mathrm{PARITY} = \bigvee_{i=1}^{t} P_{2i-1}$, kde t je největší přirozené číslo splňující $2t - 1 \leq n$.



Obr. 3.1. Obvod P_4 pro $n = 5$ ($\overline{\mathrm{MAJ}}$ značí majoritu na negacích vstupu)

Předveďme nyní konstrukci podobvodů P_k. Trik je jednoduchý. Předpokládejme nejprve, že $k > n/2$. Pak stačí vstup doplnit konstantami 0 na

[1]Majorita je očividně speciálním případem prahové funkce $\mathrm{TH}_{k,n}$ pro $k = \lfloor n/2 \rfloor + 1$.

délku $2k$. Aplikujme funkci MAJ na prvních $2k-1$ proměnných (tedy jednu konstantu vynecháme) a její hodnotu označme a. Podruhé aplikujeme MAJ na negace všech $2k$ vstupních hodnot. Výsledek pojmenujme b. Snadno nahlédneme, že $P_k = a \wedge b$.

Pro $k \leq n/2$ si stačí uvědomit, že záměnou jedniček za nuly a naopak se úloha převede na předchozí případ. Obvod P_k tudíž obsahuje přesně tři výpočetní hradla, a proto celková velikost obvodu počítajícího paritu je nanejvýš $3\lceil n/2 \rceil + 2n + 1$, tedy jistě polynomiální vzhledem k n. Jeho hloubka je nejvýše 3. □

S použitím lemmatu 3.11 snadno odvodíme následující tvrzení.

Důsledek 3.13.

$$\mathrm{MAJ} \notin \mathsf{AC}^0 .$$

Podobným způsobem je možné dokázat, že například také násobení dvou n-bitových čísel nelze počítat polynomiálně velkými obvody s omezenou hloubkou.

Kapitola 4

Monotónní obvody

4.1 Spodní odhad pro kliku

Úvodem této kapitoly definujeme pojem monotónního obvodu. Následně jej dáme do souvislosti s monotónní funkcí, jak jsme ji definovali na začátku druhé kapitoly. To bude dostatečným základem pro další výklad, jehož vyvrcholením bude předvedení exponenciálního spodního odhadu [1].

Definice 4.1. *Monotónním obvodem* rozumíme každý obvod, který obsahuje pouze binární operace $\vee$ nebo $\wedge$, nikoliv negace.

Nechť $f : \{0,1\}^n \to \{0,1\}$ je monotónní funkce. Zápisem $C^{\mathrm{m}}_{\Omega}(f)$ označujeme velikost nejmenšího monotónního obvodu v bázi Ω reprezentujícího funkci f. Nechť $g : \{0,1\}^* \to \{0,1\}$ a $h : \mathbb{N} \to \mathbb{N}$. Potom říkáme, že g má *monotónní obvodovou složitost* h, jestliže pro každé n platí $C^{\mathrm{m}}_{\Omega}(g|_{\{0,1\}^n}) = h(n)$.

Korektnost poslední definice vyplyne z následující věty.

Věta 4.2. *Booleovská funkce $f : \{0,1\}^n \to \{0,1\}$ je monotónní, právě tehdy když ji lze počítat monotónním obvodem C.*

Důkaz. Bez újmy na obecnosti lze předpokládat, že $f(0,\ldots,0) = 0$ a také $f(1,\ldots,1) = 1$, neb jinak by šlo o konstantní funkci, kterou pochopitelně lze počítat monotónním obvodem, a to dokonce konstantní velikosti.

Představme si *booleovskou hyperkrychli* v dimenzi n, jež má všechny vrcholy očíslované binárními hodnotami od 0 do $2^n - 1$. Její hrany jsou spojnice těch vrcholů, jejichž pojmenování se liší pouze v jedné pozici (tedy

mají Hammingovu vzdálenost 1). Na takový objekt lze nahlížet jako na graf. Uvažme všechny cesty (jejich počet nechť je s) z vrcholu $(0,\ldots,0)$ do vrcholu $(1,\ldots,1)$ a z každé vezměme první vrchol v^j takový, že $f(v^j) = = 1$. Všechny tyto vrcholy použijeme takto: ty proměnné x_i, které mají ve vrcholu v^j hodnotu 1 přidáme jako vstupy do hradla $\wedge$. Tím dostaneme s takových konjunktivních hradel, jejichž výstupy „svedeme“ do disjunktivního hradla $\vee$. Jeho výstup je hodnotou funkce f.

Zpětná implikace je ještě snadnější. Jestliže v obvodu nejsou negace, tak pro vstup $x \geq y$ musí nutně vracet hodnotu stejnou, nebo vyšší, ergo reprezentuje monotónní funkci. □

Plynule navážeme na předchozí důkaz, když se nyní budeme zabývat problémem z teorie grafů. Nechť je dán graf $G = (V, E)$. Říkáme, že v grafu G existuje *klika* velikosti k, pokud v něm existuje úplný podgraf na k vrcholech. Naším úkolem je rozhodnout, zda zadaný graf obsahuje kliku velikosti k. Funkci řešící tento problém nazveme

$$\mathrm{CLIQUE}_k : \quad G \mapsto \{0,1\}\,.$$

Označíme m počet vrcholů grafu. Je zřejmé, že všech možných grafů na m vrcholech je $2^{\binom{m}{2}}$. Proto jako vstup této funkce uvažujme slova $(x_{i,j})$, $1 \leq i < < j \leq m$, délky $\binom{m}{2}$, která budeme interpretovat takto:

$$x_{i,j} = 1 \Leftrightarrow \{i,j\} \in E\,, \qquad x_{i,j} = 0 \Leftrightarrow \{i,j\} \notin E\,.$$

Čili pokud v takovém vstupu změníme jeden bit z nuly na jedničku, tak zvýšíme pravděpodobnost výskytu kliky libovolné velikosti, neboť tím de facto přidáme do grafu jednu hranu. Nahlédli jsme, že funkce CLIQUE_k je monotónní.

Nejen to. *Problém kliky*, definovaný jako množina dvojic $(G, 1^k)$ takových, že graf G obsahuje kliku velikosti alespoň k, je navíc NP-úplný [8]. To znamená, že libovolný jiný problém, který je ve třídě NP, lze na tento převést v polynomiálním čase. Proto, pokud bychom dokázali nějaký silný dolní odhad pro tuto úlohu, měli bychom ihned odhady pro celou řadu dalších situací.

Pro monotónní obvody jsou dnes známy již poměrně silné spodní odhady složitosti. Velkým úspěchem byla Razborovova superpolynomiální spodní mez $n^{\Omega(\log n)}$ kterou publikoval roku 1985 [13]. Její přínos nespočíval pouze

v podstatně lepším odhadu (do té doby nejlepším byl pouze lineární [16]), ale zejména ve zvoleném přístupu. Pomocí vylepšené Razborovovy metody se o dva roky později podařilo Alonovi a Boppanovi dokázat exponenciální spodní odhad pro kliku [1].

Věta 4.3. *Nechť $G = (V, E)$ a označme $m = |V|$. Pokud*

$$3 \leq k \leq \frac{1}{4}\left(\frac{m}{\log m}\right)^{2/3},$$

pak má funkce $\mathrm{CLIQUE}_k(m)$ *monotónní obvodovou složitost alespoň*

$$\frac{1}{8}\left(\frac{m}{4k^{3/2}\log m}\right)^{(\sqrt{k}+1)/2}.$$

Speciálně, pro

$$k = \left\lfloor \frac{1}{4}\left(\frac{m}{\log m}\right)^{2/3} \right\rfloor$$

je monotónní složitost funkce $\mathrm{CLIQUE}_k(m)$ *rovna*

$$\exp\left(\Omega\left(\left(\frac{m}{\log m}\right)^{1/3}\right)\right).$$

Důkaz tohoto tvrzení je poměrně rozsáhlý, a proto jej zde pouze nastíníme. Postupovat přitom budeme jednak dle původního článku [1], v němž autoři používají více algebraickou terminologii, tak také podle práce [2], která zavádí nové pojmy a celkově důkaz zjednodušuje. Pokud jde o pojmy, tak jsou to zjevné analogie, na které povětšinou nebudeme explicitně upozorňovat. Klíčem k úspěchu je užití aproximačních obvodů a ukázání faktu, že tyto se příliš neliší od obvodu počítajícího kliku.

Pozitivním testovacím grafem rozumíme takový graf na m vrcholech, který obsahuje pouze ty hrany, které tvoří úplný podgraf na nějakých k vrcholech a žádné jiné hrany neobsahuje. Všech možných takovýchto grafů je zjevně $\binom{m}{k}$.

Vrcholy grafu lze také „obarvit“. Předpokládejme, že máme k dispozici $k-1$ barvu. Na začátku mějme množinu bodů V a prázdnou množinu hran E. Jestliže každému vrcholu $v \in V$ přiřadíme nějakou barvu $b(v)$ a následně hranami propojíme všechny dvojice vrcholů s různými barvami, aby platilo

$b(v) \neq b(v') \Rightarrow \{v, v'\} \in E$, nikdy nemůžeme dostat graf, který by obsahoval kliku velikosti k. Takto získané grafy nazýváme *negativní testovací grafy*. Všech možných obarvení je $(k-1)^m$, a i přesto, že některá vedou na shodné grafy, budeme mezi nimi rozlišovat.

Hlavní myšlenkou celého důkazu je předvést, že každý malý (ve smyslu odhadu z věty) monotónní obvod by pro většinu pozitivních testovacích grafů vracel hodnotu 0 nebo pro většinu negativních testovacích grafů by vracel hodnotu 1. Jinými slovy chceme ukázat, že pro tyto speciální grafy malé obvody nemohou fungovat správně.

Pro množinu $V = \{1, \ldots, m\}$ označíme písmenem $\mathcal{A}$ systém všech jejích podmnožin. Zřejmě $|\mathcal{A}| = 2^m$. Operace $\lceil A \rceil$ na množině $A \in \mathcal{A}$ dává hodnotu 1, pokud A obsahuje kliku, a 0 jinak (přesněji řečeno graf $(A, E \cap$ $\cap\, (A \times A))$ obsahuje resp. neobsahuje kliku velikosti k). Této funkci říkejme *indikátor kliky*.

Omezíme-li konstantou l kardinalitu množin, které indikátor může zkoumat, můžeme definovat *aproximující obvod* jako disjunkci nejvýše μ omezených klikových indikátorů. Nadále předpokládejme, že $l < k$. Pak aproximátor bude vracet hodnotu 1, pouze tehdy dostane-li na vstupu úplný podgraf na l bodech.

Monotónnímu obvodu C, který rozhoduje problém kliky, tak můžeme přiřadit aproximující obvod $\tilde{C}$. Parametrem tohoto přiřazení je číslo l, které určuje, jak moc velké podmnožiny množiny V má aproximátor prozkoumávat. Je zjevné, že pro velké hodnoty l bude aproximátor poměrně přesný.

Pro důkaz sporem nyní mějme malý obvod C a jemu příslušející aproximátor $\tilde{C}$. V první řadě ukážeme, že pro pozitivní testovací grafy platí

$$C \leq \tilde{C}$$

a pro negativní testovací grafy platí

$$C \geq \tilde{C}\,.$$

V druhém kroku pak zbývá dokázat, že každý aproximátor vrací 0 na většině pozitivních testovacích grafů nebo vrací 1 na většině negativních testovacích grafů.

Odtud snadno plyne, že malé obvody pro simulaci funkce CLIQUE_k nestačí.

Abychom zkonstruovali aproximátor, postupujme „odspodu“. Aproximátor pro graf na dvou vrcholech i a j (lze značit $\lceil\{i,j\}\rceil$) je ekvivalentní proměnné $x_{i,j}$. Zbývá provést indukční krok.

Předpokládejme, že máme aproximátor pro každý vlastní podobvod obvodu C. Pak jsou dvě možnosti. Buďto je poslední operací disjunkce, nebo konjunkce. Potřebujeme tedy vybudovat aproximátory pro tyto dvě elementární operace.

Nejprve pro disjunkci. Označíme $A = \bigvee_{i=1}^{r} \lceil X_i \rceil$ a $B = \bigvee_{j=1}^{s} \lceil Y_j \rceil$ aproximátory z indukčního předpokladu. Díky konstrukci máme $r, s \leq \mu$. Pokud bychom však jako aproximátor $A \vee B$ vzali $(\bigvee_{i=1}^{r} \lceil X_i \rceil) \vee (\bigvee_{j=1}^{s} \lceil Y_j \rceil)$, tak se může stát $r+s > \mu$, a tedy získaný aproximátor by nebyl korektně definován.

Toto lze ošetřit jednoduchým trikem. Některé části aproximátorů nahradíme jednou částí, která bude pro všechny shodná.

Říkáme, že skupina množin $M_1, M_2, \ldots, M_p$ tvoří *slunečnici*, pokud existuje množina M taková, že pro každou dvojici různých indexů i, j platí $M_i \cap M_j = M$. Množinu M nazýváme *semeník* a může být i prázdná. Množiny M_i tvoří *okvětní lístky*. V našem případě půjde o množiny obsahující vrcholy grafu, tedy o podmnožiny V. Každá z nich má navíc mohutnost nejvýše l.

Vraťme se k našemu problému a podívejme se na systém množin $\mathcal{M} = \{X_1, \ldots, X_r, Y_1, \ldots Y_s\}$. Pokud některé jeho prvky tvoří slunečnici, tak je nahradíme příslušným semeníkem. Této proceduře říkáme *otrhávání* a opakujeme ji, dokud není systém $\mathcal{M}$ dostatečně malý (nejvýše μ prvků).

Lemma 4.4 (Erdös-Rado [4])**.** *Nechť $\mathcal{M}$ je systém množin, z nichž každá má nejvýše l prvků. Pokud $|\mathcal{M}| > (p-1)^l l!$, pak $\mathcal{M}$ obsahuje slunečnici s alespoň p okvětními lístky.*

Zvolíme-li $\mu = (p-1)^l l!$, lze opakovaně použít lemmatu, a dokud bude $|\mathcal{M}| > \mu$, můžeme otrhávat lístky. Na konci pak nutně $|\mathcal{M}| \leq \mu$. Aproximovanou disjunkci budeme značit $\sqcup$.

Nyní pro konjunkci. Opět máme z indukčního předpokladu $A = \bigvee_{i=1}^{r} \lceil X_i \rceil$ a $B = \bigvee_{j=1}^{s} \lceil Y_j \rceil$, kde $r, s \leq \mu$. Pomocí distributivního zákona vypočítáme

$$A \wedge B = \bigvee_{i=1}^{r} \bigvee_{j=1}^{s} \left(\lceil X_i \rceil \wedge \lceil Y_j \rceil \right).$$

Toto ale není aproximátor. Jednak proto, že se může skládat až z μ^2 termů, ale také kvůli tomu, že $(\lceil X_i \rceil \wedge \lceil Y_j \rceil)$ nevytváří indikátor kliky.

Přes takovéto nepříjemnosti se lze přenést ve třech krocích:

($\wedge$1) přepsat $\lceil X_i \rceil \wedge \lceil Y_j \rceil = \lceil X_i \cup Y_j \rceil$,

($\wedge$2) vypustit ty indikátory, které využívají množiny $X_i \cup Y_j$ kardinality větší než l,

($\wedge$3) aplikovat otrhávací algoritmus jako v případě disjunkce.

Získaný aproximátor konjunkce budeme značit $\sqcap$.

Dohromady se nám podařilo zkonstruovat aproximující obvod $\tilde{C}$.

Nyní se podívejme na vlastnosti aproximovaných operací. Uvažme nejprve pozitivní testovací graf. Po jeho otrhání testujeme pouze menší množiny vrcholů. Může se stát, že jsme nějakou testovací množinu tak zmenšili, že z ní zbyl již jen úplný graf, tedy $A \vee B \leq A \sqcup B$.

Pro konjunkci je situace složitější. V kroku ($\wedge$1) se nic neděje, protože na pozitivním testovacím grafu se $\lceil X_i \rceil \wedge \lceil Y_j \rceil$ i $\lceil X_i \cup Y_j \rceil$ chovají stejně. Následující krok je problematický, neboť se při něm vynechávají indikátory mající $|X_i \cup Y_j| \geq l+1$. Tyto indikátory testují, zda na aspoň $l+1$ vrcholu existuje klika velikosti k. Ze zbylých nejvýše $m-(l+1)$ vrcholů můžeme vybrat dalších nejvíce $k-(l+1)$ bodů, které spojíme hranami, ale ořezaný aproximátor to nemůže nijak zjistit. Čili ztrácí informaci o nejvýše $\binom{m-l-1}{k-l-1}$ pozitivních testovacích grafech. Takových zanedbání provádíme nanejvýš μ^2. A konečně krok ($\wedge$3) jsme již rozebrali v minulém odstavci.

Celkem obvod C obsahuje nanejvýš $|C|$ těchto operací, takže počet pozitivních testovacích grafů, pro které $C > \tilde{C}$, je nejvýše

$$|C|\,\mu^2 \binom{m-l-1}{k-l-1}.$$

Všech možných pozitivních testovacích grafů je však $\binom{m}{k}$, což je mnohem více, a proto docházíme k závěru, že pro většinu z nich platí $C \leq \tilde{C}$.

Podobným způsobem, i když poněkud náročněji, se dá ukázat, že nerovnost $C < \tilde{C}$ platí pro nejvýše

$$|C|\,\mu^2 \left(\frac{\binom{l}{2}}{k-1} \right)^p (k-1)^n$$

negativních testovacích grafů. Z těchto výrazů se již poměrně snadno odvodí ona exponenciální dolní hranice monotónní složitosti.

Zbývá ukázat, že aproximátor většinou chybuje. Mějme $A = \bigvee_{i=1}^{r} \lceil X_i \rceil$ a nějaký negativní testovací graf. V úvahu přicházejí dvě možnosti. Aproximátor je buďto identicky nulový, s čímž nic neuděláme, anebo není. Potom ovšem $A \geq \lceil X_1 \rceil$. Negativní testovací graf je tímto indikátorem odmítnut, pokud některé dva vrcholy v X_1 mají tutéž barvu, protože pak nemohou být spojeny, a nejde tudíž o kliku. Mezi všemi $(k-1)^m$ obarveními zvolme náhodně jedno. Pravděpodobnost, že dva vrcholy v X_1 mají shodnou barvu, je nejvýše

$$\frac{\binom{|X_1|}{2}}{k-1} \leq \frac{\binom{l}{2}}{k-1}\,.$$

Takže pravděpodobnost, že se tak nestane, a tedy $\lceil X_1 \rceil = 1$, je nejméně $1 - \binom{l}{2}/(k-1)$. Vzhledem k počtu možných obarvení vidíme, že aproximátor vrací 1 na alespoň

$$\left(1 - \frac{\binom{l}{2}}{k-1}\right)(k-1)^m$$

možných negativních testovacích grafech, nebo je identicky nulový.

Tímto jsme dokázali oba kroky potřebné do „hlavní myšlenky důkazu“. Proto nemůže být žádný monotónní obvod počítající funkci CLIQUE_k menší nežli exponenciální vzhledem k délce vstupu.

Jelikož je tento problém NP-úplný, tak je tento odhad aplikovatelný i na další monotónní NP-úplné úlohy.

4.2 Monotónní projekce

Nyní, podobně jako v předchozí kapitole, využijeme výše uvedený výsledek pro CLIQUE_k k získání spodních odhadů pro další booleovské funkce. Nejprve zaveďme nový termín *monotónní projekce*. Říkáme, že booleovská funkce $f : \{0,1\}^m \to \{0,1\}$ je monotónní projekcí funkce $g : \{0,1\}^n \to \{0,1\}$, právě když existují symboly $\sigma_1, \ldots, \sigma_n \in \{0,1\} \cup \{x_1, \ldots, x_m\}$[1],

[1]Pokud připustíme i negace proměnných x_i, jde pouze o *projekci*.

pro které $f = g(\sigma_1, \ldots, \sigma_m)$. Je očividné, že pokud funkce f je monotónní projekcí g, pak monotónní výpočetní složitost funkce g je alespoň stejně tak velká jako v případě funkce f.

Definice 4.5. SAT(m) je funkce $2m^2$ proměnných $x_{1,1}, \ldots, x_{m,m}, y_{1,1}, \ldots, y_{m,m}$, která vrací hodnotu 1, právě když existuje ohodnocení $z_1, \ldots, z_m \in \in \{0,1\}$, které splňuje formuli

$$\bigwedge_{i=1}^{m} \bigvee_{j=1}^{m} \big((x_{i,j} \wedge z_j) \vee (y_{i,j} \wedge \neg z_j) \big) . \tag{4.1}$$

Lemma 4.6. *Pro $1 \leq k \leq m$ je funkce* CLIQUE$_k(m)$ *monotónní projekcí funkce* SAT($4m^4$).

Důkaz. Vytvořme nejprve CNF formuli A v proměnných $q_{u,i}$, $u \in \lceil k \rceil = \{1, \ldots, k\}$ a $i \in \lceil m \rceil$, a p_t, kde $t \in \binom{\lceil m \rceil}{2}$ odpovídá možným hranám v grafu na m vrcholech, která vyjadřuje, že relace

$$\{(u,i) \mid q_{u,i} = 1\}$$

je grafem prosté funkce $f : \lceil k \rceil \to \lceil m \rceil$, jejíž obor hodnot je klika v grafu s vrcholy $\lceil m \rceil$. Pro úplnost výkladu uveďme, že konjunkce klauzulí

(i) $\{q_{u,1}, \ldots, q_{u,m}\}$ $\forall u \in \lceil k \rceil$,

(ii) $\{\neg q_{u,i}, \neg q_{u,j}\}$ $\forall u \in \lceil k \rceil$, $\forall i, j \in \lceil m \rceil$, $i \neq j$,

(iii) $\{\neg q_{u,i}, \neg q_{v,i}\}$ $\forall u, v \in \lceil k \rceil$, $u \neq v$, $\forall i \in \lceil m \rceil$ a

(iv) $\{\neg q_{u,i}, \neg q_{v,j}, p_{\{i,j\}}\}$ $\forall u, v \in \lceil k \rceil$, $u \neq v$, $\forall i, j \in \lceil m \rceil$, $i \neq j$

je pravdivá, právě když graf obsahuje kliku na $\lceil k \rceil$.

Celkový počet proměnných je tedy $km + \binom{m}{2} \leq 2m^2$ a formule má méně než $k + km^2 + k^2m + k^2m^2 \leq 4m^4$ klauzulí.

Pro redukci zvolme M větší z obou hodnot, tedy $M = 4m^4$. Pro každou konkrétní množinu hran E (tedy každou substituci hodnot 0 a 1 za proměnné p_t) dostaneme formuli A_E v proměnných $q_{u,i}$, jejíž splnitelnost je ekvivalentní existenci kliky velikosti k v grafu $(\lceil m \rceil, E)$.

Pozitivní, resp. negativní literál $q_{u,i}$ v s-té klauzuli reprezentujme užitím nové proměnné $x_{s,r(u,i)}$, resp. $y_{s,r(u,i)}$. Tyto proměnné můžeme přímo použít pro konstrukci formule ve tvaru (4.1), jejíž splnitelnost je pak také ekvivalentní existenci kliky. □

Věta 4.7. *Monotónní obvodová složitost funkce* $\mathrm{SAT}(m)$ *je*

$$\exp\left(\Omega\left(\frac{m^{1/6}}{\log^{1/3} m}\right)\right).$$

Důkaz. Plyne ihned spojením úvahy před definicí, lemmatu a věty 4.3. □

Závěrem poznamenejme, že předvedená metoda monotónních projekcí je koncipována mnohem obecněji a poskytuje užitečnou pomůcku pro spodní odhady složitosti i jiných funkcí.

4.3 Souvislost mezi monotónní a klasickou obvodovou složitostí

Díky dosaženým výsledkům je znalost vztahu mezi monotónní a nemonotónní složitostí velmi zajímavá. Pokud by totiž byla polynomiální, znamenalo by to, že asymptoticky stejné odhady (tj. exponenciální) platí i pro neomezené obvody počítající téže funkce i za pomocí negací.

Bohužel se však ukazuje, že mezi zmíněnými třídami polynomiální vztah není. Éva Tardos ve své poznámce [15] nabízí další zobecnění Razborova přístupu, díky kterému ukáže exponenciální rozdíl mezi monotónní a nemonotónní obvodovou složitostí pro jednu funkci z P.

Naproti tomu zajímavou alternativou jsou tzv. *plátkové funkce* (angl. slice functions), o nichž je známo [17, sekce 6.13], že velmi dobře zachovávají složitost při přechodu z monotónního obvodu do obvodů v úplné bázi.

Definice 4.8. Funkce $f : \{0,1\}^n \to \{0,1\}$ se nazývá k-plátková, jestliže $f(x) = 0$ pro každé $x \in \{0,1\}^n$, které obsahuje méně než k jedniček, $f(x) = 1$ pro každé $x \in \{0,1\}^n$ obsahující více než k jedniček. Pokud vstupní slovo obsahuje právě k jedniček, může být funkce definována zcela libovolně.

Je zřejmé, že plátkové funkce jsou monotónní.

Předpokládejme, že obvodová složitost při použití negací by byla nižší než při jejich zakázání. Uvažme tedy optimální obvod v úplné bázi, který můžeme přestavět tak, aby se všechny negace vyskytovaly v nejnižší úrovni, tj. přímo nad proměnnými. A nyní se pokusíme tyto negované proměnné nahradit monotónním obvodem.

Definice 4.9. Nechť $f : \{0,1\}^n \to \{0,1\}$. Funkce $h_i : \{0,1\}^n \to \{0,1\}$ je *pseudokomplementem* pro x_i vzhledem k f, pokud v libovolném obvodu pro f lze negaci proměnné x_i nahradit funkcí h_i.

Věta 4.10. *Označme $X = (x_1, \ldots, x_n)$ a $X_i = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$. Potom prahová funkce $\mathrm{TH}_k(X_i)$ je pseudokomplementem pro x_i vzhledem k libovolné k-plátkové funkci f.*

Důkaz. Označíme f' funkci, která vznikla z f náhradou $\overline{x}_i$ pomocí prahové funkce $\mathrm{TH}_k(X_i)$. Chceme ukázat $f' = f$.

Mějme nejprve vstup $x \in \{0,1\}^n$ obsahující méně než k jedniček. Potom jistě $f(x) = 0$, ale také $\mathrm{TH}_k(x \cap X_i) = 0 \leq \overline{x}_i$, kde zápisem $x \cap X_i$ rozumíme slovo vzniklé z x vypuštěním znaku x_i. Tedy $f'(x) = f(x)$.

Nyní nechť x obsahuje více než k jedniček. Pak nutně $f(x) = 1$, ale také $\mathrm{TH}_k(x \cap X_i) = 1 \geq \overline{x}_i$. A z monotonie funkce f' máme opět $f'(x) = f(x)$.

Zbývá možnost, že x obsahuje právě k jedniček. V tom případě

$$\mathrm{TH}_k(x \cap X_i) = 1 \Leftrightarrow x_i = 0 \Leftrightarrow \overline{x}_i = 1\,,$$

neboť oněch k jedniček je na jiných pozicích nežli i-té. Z ekvivalence je vidět, že i v tomto případě prahová funkce nahrazuje negaci. □

Navíc je prahová funkce monotónní a má složitost nejvýše lineární [17, sekce 6.2] tvaru

$$2\,(k-1)\,n - k^2\,, \tag{4.2}$$

čili se dobře využije při zbavování se negací v obecných obvodech, ale přitom je příliš nezvětší.

Definice 4.11. *Exaktní funkcí* $\mathrm{E}_k : \{0,1\}^n \to \{0,1\}$ rozumíme takovou funkci, která má na vstupu $x \in \{0,1\}^n$ hodnotu 1, právě když x sestává z k jedniček.

Definice 4.12. Funkci $f^k : \{0,1\}^n \to \{0,1\}$ nazýváme *k-tým plátkem* funkce $f : \{0,1\}^n \to \{0,1\}$ a definujeme ji jako

$$f^k = (f \wedge \mathrm{E}_k) \vee \mathrm{TH}_{k+1} = (f \wedge \mathrm{TH}_k) \vee \mathrm{TH}_{k+1}\,.$$

Lemma 4.13. *Pro libovolnou booleovskou funkci n proměnných platí*

$$f = \bigvee_{0 \leq k \leq n} \left(f^k \wedge \mathrm{E}_k\right)\,.$$

Důkaz. Stačí dosadit do vzorce a použít distributivní zákon

$$\begin{aligned}\bigvee_{0\le k\le n}\left(f^k\wedge \mathrm{E}_k\right) &= \bigvee_{0\le k\le n}\Big(\big((f\wedge \mathrm{E}_k)\vee \mathrm{TH}_{k+1}\big)\wedge \mathrm{E}_k\Big) = \\ &= \bigvee_{0\le k\le n}\big((f\wedge \mathrm{E}_k)\vee(\mathrm{TH}_{k+1}\wedge \mathrm{E}_k)\big) = \\ &= \bigvee_{0\le k\le n}(f\wedge \mathrm{E}_k) = f\,,\end{aligned}$$

neboť prahová funkce pro $k+1$ a exaktní pro k se vzájemně vylučují. □

S použitím lemmatu a horního odhadu (4.2) se snadno dokáží další tvrzení.

Věta 4.14. *Pro složitost C monotónní reprezentace $(f^0,\ldots,f^n)$ funkce f platí*

$$C(f)\le C(f^0,\ldots,f^n)+O(n)\qquad a\qquad C(f^0,\ldots,f^n)\le C(f)+O(n)\,.$$

Pro její monotónní složitost C^{m} obdobně platí

$$C^{\mathrm{m}}(f^0,\ldots,f^n)\le C^{\mathrm{m}}(f)+O(n\log n)\qquad a\qquad C^{\mathrm{m}}(f^k)\le C^{\mathrm{m}}(f)+O(n)\,,$$

je-li f monotónní a k je konstanta.

Důsledek 4.15. *Pro k-tý plátek booleovské funkce f platí*

$$C^{\mathrm{m}}(f^k)\le O(C(f^k))+C^{\mathrm{m}}(\mathrm{TH}_k(X_i))\,,$$

kde $1\le i\le n$.

Čili silné spodní odhady na monotónní složitost nějakého k-tého plátku f^k funkce f zároveň poskytují i dolní meze pro obvodovou složitost této funkce. To je značnou motivací pro hledání různých funkcí, které by poskytly co nejsilnější odhady ve svých plátkových analogiích. Vše ale závisí na tom, jak efektivním způsobem počítat prahovou funkci bez užití negací. Například z osobní korespondence mezi Wegenerem a Patersonem [17] je znám obvod se složitostí

$$C^{\mathrm{m}}(\mathrm{TH}_k)=O\left(n\min\{k,n-k,\log^2 n\}\right)\,,$$

což je v nejhorším případě kvadratický odhad.

Kapitola 5

Závěr

Spodní odhady na velikost obecných obvodů pro konkrétní booleovské funkce se sice na první pohled zdají jako snadné kombinatorické úlohy, ale dosud se jim daří odolávat všem metodám. Jak náš přehled ukazuje, dokonce i nejrůznější třídy speciálních obvodů vyžadují velmi sofistikované postupy a nejednoduché kombinatorické úvahy.

Vývoj za posledních třicet let nám navíc nedodává příliš optimismu, pokud jde o naději na zásadní vylepšení současných výsledků. Vše nasvědčuje tomu, že dosud užívané metody nelze zásadním způsobem vylepšit ani zobecnit. Proto je třeba se pokusit najít lepší přístupy, díky kterým bychom se s našimi znalostmi posunuli blíže řešení těchto otázek.

Literatura

[1] ALLON, Noga, BOPPANA, Ravi. B. The Monotone Circuit Complexity of Boolean Functions. Combinatorica vol. 7, no. 1, s. 1–22. 1987.

[2] BOPPANA, Ravi B., SIPSER, Michael. The Complexity of Finite Functions. In VAN LEEUWEN, Jan. Handbook of Theoretical Computer Science. [s.l.] : Elsevier Science Publishers B.V. and The MIT Press, 1990. 14. s. 757–804. Dostupný z WWW: <http://www.cs.columbia.edu/~rocco/Teaching/S09/6998/Boppana-Sipser-complexity.ps>.

[3] Clay Mathematics Institute : Millenium Prize Problems [online]. 2000 [cit. 2009-03-13]. Dostupný z WWW: <http://www.claymath.org/millennium/P_vs_NP/>.

[4] ERDÖS, Pál, RADO, Richard. Intersection Theorems for Systems of Sets. Journal of the London Math. Soc. 1960, vol. 35, no. 1, s. 85–90.

[5] FURST, Merrick L., SAXE, James B., SIPSER, Michael. Parity, Circuits, and the Polynomial-Time Hierarchy. Mathematical Systems Theory. 1984, vol. 17, no. 1, s. 13–27.

[6] HÅSTAD, Johan. Almost Optimal Lower Bounds for Small Depth Circuits. In Annual ACM Symposium on Theory of Computing. Berkeley, California, United States : ACM New York, NY, USA, 1986. s. 6–20. ISBN 0-89791-193-8.

[7] IWAMA, Kazuo, MORIZUMI, Hiroki. An Explicit Lower Bound of $5n - o(n)$ for Boolean Circuits. In Lecture Notes In Computer Science : Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science. London, UK : Springer-Verlag, 2002. s. 353–364. ISBN 3-540-44040-2.

[8] KARP, Richard M. Reducibility Among Combinatorial Problems. Proceedings of a Symposium on the Complexity of Computer Computations. Plenum Press. 1972.

[9] MULLER, David E. Complexity in Electronic Switching Circuits. IRE Transactions on Electronic Computers. 1956, vol. 5, s. 15–19.

[10] PIPPENGER, Nicholas, FISCHER, Michael J. Relations Among Complexity Measures. Journal of the ACM. 1979, vol. 26, is. 2, s. 361–381.

[11] PRENEEL, Bart. Analysis and Design of Cryptographic Hash Functions. [s.l.], 2003. 338 s. Dizertační práce. Dostupný z WWW: <http://homes.esat.kuleuven.be/~preneel/phd_preneel_feb1993.pdf>.

[12] RAZBOROV, Alexander A. Bounded Arithmetic and Lower Bounds in Boolean Complexity. In Feasible Mathematics II. [s.l.] : Birkhäuser Verlag, 1995. s. 344–386. Dostupný z WWW: <http://www.mi.ras.ru/~razborov/bobo.ps>.

[13] RAZBOROV, Alexander A. Lower Bounds for the Monotone Complexity of some Boolean Functions. Soviet Mathematics Doklady. 1985, vol. 31, no. 2, s. 354–357. Dostupný z WWW: <http://www.mi.ras.ru/~razborov/clique.pdf>.

[14] SAVAGE, John E. Computational Work and Time of Finite Machines. Journal of the ACM. 1972, vol. 19, is. 4, s. 660–674.

[15] TARDOS, Éva. The Gap between Monotone and Non-monotone Circuit Complexity Is Exponential. Combinatorica vol. 7, no. 4, s. 141–142. 1987.

[16] TIEKENHEINRICH J. A $4n$-Lower Bound on the Monotone Network Complexity of a One-Output Boolean Function Information Processing Letters. 1984, vol. 18, no. 4, s. 201–202.

[17] WEGENER, Ingo. The Complexity of Boolean Functions. [s.l.] : Wiley, 1987. 458 s.