Řešení úlohy č. 3
Meziuzelná autorizace

Problém

Nejdřív si řekneme, co je vlastně problém, co se snažíme vyřešit.

Máme n uzlů a každý má jeden z následujících protokolů, který mu dává nějakou vlastnost, při odpovídání na otázky.

SAP na každou otázku odpoví pravdu
MAP na každou otázku odpoví lež

Umí odpovídat pouze ANO nebo NE.

Bud N množina uzlů velikosti n. Dále mějme množiny S a M, $S \cup M = N$. Kde $\forall u \in S:$ u používá SAP a $\forall u \in M : u$ používá MAP. A $|S| \geq 3$ a $|M| \geq 3$

Z množiny N můžeš vybrat tříprvkovou podmnožinu a zeptat se jí na otázku. Odpovědí množiny bude odpověd převažující mezi jejími prvky.

Algoritmy budeme popisovat jako bychom na uzly mohli šahat a fyzicky je přesouvat. Na pochopení principu algoritmu to stačí, pokud bychom je implementovali, tak to samozřejmě bude potřebovat větší abstrakci.

Nyní si ukážeme několik ukázkových algoritmů/protokolů. U každého algoritmu je uvedeno maximální ohodnocení kterého s nimi lze získat. Obecně se bodové zisky odvíjeli následovně:

- 2 body pokud protokol funguje
- 3 body za počet dotazů kolik jich protokol provede
  - 3 body za přesně n dotazů
  - 2 body za $n + c$ dotazů, kde $c > 0$ je konstanta
  - 1 bod za $O(n)$ dotazů
  - 0 bodů ze $\Omega(n^2)$ dotazů
- 3 body pokud je zdůvodněn počet dotazů a proč protokol vždy zaručí správného výsledku
- 1 bod za ukázání, že řešení s méně dotazy neexistuje
- 1 bod za rozšíření na verzi se stydlivými uzly

Algoritmus 1

5 bodů

Vybereme z množiny N dva uzly a a b. Zbytek postavíme do řady a označíme čísly. Postupně je dáváme do trojice s a a b a ptáme se na nějakou tautologii.

1	2	3	4	...	$n-2$
a					
b					

1	2	3	4	...	$n-2$
a					
b					

1	2	3	4	...	$n-2$
a					
b					

Co tím zjistíme? Jsou pouze 3 možnosti, jak vypadá dvojice (a, b). A to

1. $a \in S$, $b \in S$
2. $a \in S$, $b \in M$
3. $a \in M$, $b \in M$

Pokud nastala možnost 2, tak máme prakticky vyřešeno. Pokud byla odpověd `ANO`, tak uzel používá SAP, pokud `NE`, tak používá MAP. Zbývá zjistit, jaký protokol používají a a b. To zjistíme tím, že vezmeme dva uzly, jednoho používajícího $x \in S$ a $y \in M$. Takže vytvoříme jinou skupinku (x, y, a) a (x, y, b). Zeptáme se na tautologii a máme vystaráno.

Pokud nastala možnost 1 nebo 3. Tak na všechny dotazy bude odpověd `ANO` respektive `NE`. Tak máme smůlu a opakujeme znova.

2	3	4	5	...	$n-2$
a					
1					

2	3	4	5	...	$n-2$
a					
1					

2	3	4	5	...	$n-2$
a					
1					

A takto opakujeme dokud nenastane případ 2. Což může být v nejhorším případě n^2 dotazů.

Algoritmus 2

7 bodů

Všimněme si ale, že se nemusíme ptát na tautologii, ale můžeme se zeptat na složení trojice. Takže se zeptáme na následující otázky: Používáte všichni 3 MAP? Používají dva z vás MAP a jeden SAP? Používají dva z vás SAP a jeden MAP? Na čtvrtou možnost se ptát nemusíme, protože to je poslední možnost, když nebudou ani jedna z předchozích. Jak to dopadne v jednotlivých případech?

	Všichni S	Dva S jeden M	Dva M jeden S
S, S, S	NE	ANO	ANO
S, S, M	ANO	NE	ANO
S, M, M	NE	NE	ANO
M, M, M	NE	NE	NE

Tabulka 1: Ve sloupečcích je odpověd odpověd na složení, v řádcích skupinky. Pro přehlednost značíme jenom množiny jejichž jsou prvky.

Všimněme si, že pro každý možný případ, jsou odpovědi různé, takže z nich zvládneme dekódovat, jaké složení má skupinka. Když víme, jaké protokoly se ve skupince používají, tak se můžeme zeptat přímo na konkrétní uzly ve skupince. Například: Používá uzel s nejmenším id ve vaší skupince MAP? Takhle zvládneme na dvě otázky zjistit, kdo je kdo ve skupince. Tuto skupinku pak použijeme na zjištění ostatních uzlů. Pomocí otázky: Používá uzel s id = ¡id¿ SAP?

Celkově tento algoritmus potřebuje $n + 2$ otázek. Tři otázky na zjištění složení skupinky, dvě na identifikování uzlů ve skupince a $n - 3$ na identifikování zbytku.

Algoritmus 3

7 bodů

Zjistili jsme, že se můžeme ptát na otázku, jaký mají uzly protokol. Můžeme se zeptat na nějakou tautologii a tím zjistíme, jestli skupinka lže, nebo mluví pravdu. A pak se ptáme přímo na jednotlivé uzly jako v předchozím algoritmu. Tentokrát se ale musíme zeptat všech uzlů na protokol, protože

nevíme, jaké je složení, takže si ten poslední nedopočítáme. Takže máme celkem jednu otázku za zjištění pravdomluvnosti skupinky a n otázek za zjištění protokolu každého uzlu. Celkem $n+1$ otázek.

Optimální algoritmus

8 bodů

Můžeme se nějak zbavit nejistoty ohledně toho, jestli nám skupinka lže? Ano! Můžeme. Třeba otázkou Jak byste odpověděli, kdybych se vás zeptal na X. Koukněme se na možné odpovědi, tentokrát podle toho, jestli ve skupince převažuje SAP, nebo MAP.

X	$\top$	$\bot$
MAP	$\neg\neg\top \models\!\!\models \top$	$\neg\neg\bot \models\!\!\models \bot$
SAP	$\top$	$\bot$

Tabulka 2: Odpovědi na komplexní otázku, $\models\!\!\models$ značí logickou ekvivalenci

Vidíme, že nám pokaždé obě skupinky odpoví stejně, a řeknou nám pravdu. Takže se stačí zeptat: Používá uzel s nejnižším id ve skupince SAP? Používá uzel s prostředním id ve skupince SAP? Používá uzel s nejvyšším id ve skupince SAP?

Takhle se zeptáme všech skupinek a máme to. Celkově potřebuje tento algoritmus n otázek.

Důkaz optimálnosti

Pro dokázání správnosti se podívejme do tabulky 2. Skupinky odpoví na každou otázku pravdivě. Takže když se každé skupinky zeptáme na protokoly jejích prvků, tak odpoví pravdivě a algoritmus bude mít správný výsledek.

Algoritmus je optimální, pokud nelze vymyslet jiný algoritmus takový, který použije méně otázek a zároveň je správný.

Počet různých možností, jaké můžou mít uzly protokoly je 2^n, ale protože od každý protokol mají alespon 3 uzly, tak musíme odečíst $2\cdot(1+n+\frac{n(n-1)}{2})$, po odečtení dá číslo větší než 2^{n-1} pro dost velká n. Každý uzel má bud SAP nebo MAP. Když si používání SAP označíme jako 1 a MAP jako 0. A uzly si seřadíme podle nějakého unikátního id za sebe, tak máme n-bitové binární číslo. Abychom zjistili, jak vypadá číslo pro konkrétní instanci problému musíme zjistit jaký protokol používají všechny uzly. Jednou otázkou získáme maximálně jeden bit tohoto čísla. Proto, musí libovolný algoritmus použít alespon n otázek, jinak nezjistí dostatek bitů a tutíž by nebyl správný, takže ani optimální.

Rozšíření

Všimněmě si, že protože chceme rozlišit jaký protokol používají a je nám jedno, jestli se za něj stydí nebo ne, tak nám stačí použít předchozí algoritmus. Je nám totiž jedno, jestli členové skupinky lžou nebo ne, protože se vždy dozvíme pravdu. Stejně tak je použitelný i algoritmus 3.