1 Fagin-tétel

Bizonyos bonyolultságelméleti problémák esetén jó iránymutató a Fagin-tétel, amely arról beszél, hogy egy eldöntési probléma, amelyet logikai formulával fogalmazunk meg, mikor NP-probléma. Nézzünk néhány példát!

Klikk probléma

Keresni egy véges gráfban egy háromszöget a következő formula igazságának ellenőrzését jelenti a gráf által meghatározott "univerzumban"

$$\exists xyz, E(x,y) \wedge E(y,z) \wedge E(z,x)$$

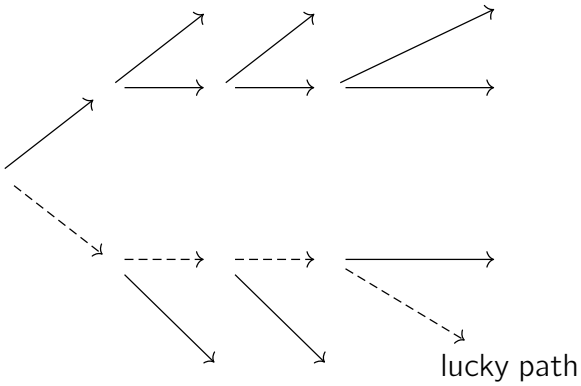
Erre egy Coq példa implementáció itt található:
Keresni egyáltalán egy klikket, azaz olyat, amelynek minden csúcspárja össze van kötve nem más, mint eldöntei, hogy igaz-e ez:

$$\exists X, \forall xy, Xx \wedge Xy \rightarrow E(x,y)$$

itt X egy olyan változó, ami egy egyváltozós logikai függvényt jelöl, ami megmondja, hogy egy csúcs benne van-e az X igazságtartományában vagy sem.

A két formula abban különbözik, hogy az első elsőrendű, csak csúcsokon értelmezett kvantorok vannak benne, míg a második *másodrendű egzisztenciális,* azaz van az elején egy vagy több olyan egzisztenciális kvantor, amely "tulajdonságok" felett is kvantifikál, konkrétan most ez egy egyváltozós tulajdonság, ami azt jelképezi, hogy valamely csúcs egy részgráfban van-e vagy sem és a többi a szokásos elsőrendű, azaz tárgyak felett kvantifikáló kvantor és logikai műveletek.

Fagin tétele: Azok az eldöntési problémák, amelyek NP-ben annak, pontosan azok, amik másodrendű egzisztenciális formulával leírhatók egy véges individuumtartományon.



Az NP problémák olyan nem-determinisztikus Turing-gépen futnak, ami csodálatos képességgel rendelkezik: le tudja "tapogatni" az egész döntési univerzumot és ki tudja választani azt az utat, amelyik gyorsan, polinom időben hozza az igaz döntés eredményét, ha van ilyen (ha nincs, akkor nincs). Voltaképpen ennek a lucky path-nak a létezését állítják a másodrendű egzisztenciális formulák.

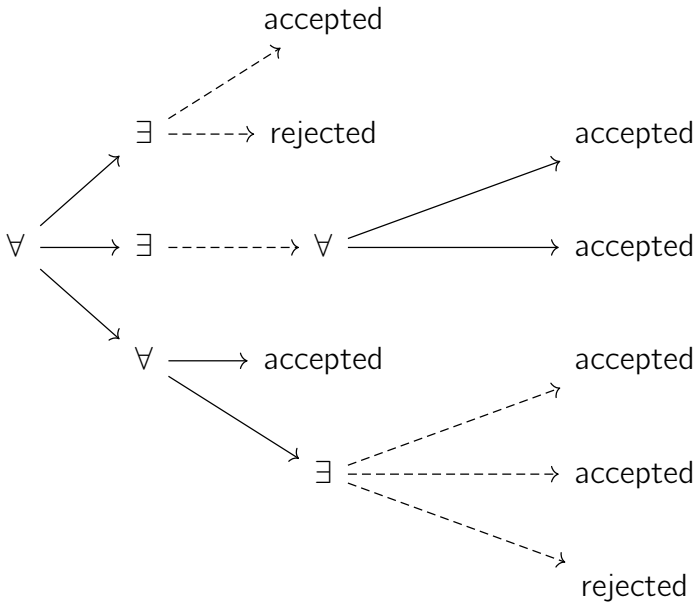
Iyenre Coq példa még a 3-színezés ahol egy háromértékű függvényt kell megadni, a színezést, a csúcsokon.

2 ATM (Alternáló Turing-gép)

Egy **Alternáló Turing-gép (ATM)** a nem-determinista Turing gép általánosítása. Az ATM-ben kétféle állapot létezik: univerzális és egztenciális.

- **univerzális állapot:** akkor tekintjük elfogadottnak, ha a számításifa minden levelét elfogadja.
- **egzisztenciális állapot:** akkor tekintjük elfogadottnak, ha a számításfa legalább egy levelét elfogadja.

Itt tehát két dologról beszélünk egyszerre párhuzamos műveletvégzésről is és a lucky path megtalálásáról is. Az ATM futási időtartama a számítási fa magassága, amely az input méretéhez viszonyított időt jelenti. Ha egy ATM kizárólag egztenciális állapotokat tartalmaz, akkor az egy sima nem-determinista Turing gép (NTM), mivel polinomiális idő alatt képes elfogadni, ha elfogad egyáltalán.



3 Quantified Boolean Formula (QBF) probléma

A QBF probléma egy olyan logikai probléma, amelyben egy logikai formula Boole(!) változói kvantifikálva vannak. Ennek a nyelvnek egy Coq-modellje itt található: Prenex normálformára hozás esetén, amikor is elől vannak a kvantorok és utána a nulladrendű operátorok egy ilyen formula a következő formát ölt:

$$Q_1 x_1 Q_2 x_2 \dots Q_n x_n \, \varphi(x_1, x_2, \dots, x_n),$$

Q_i kvantor, $\forall$ (univerzális) vagy $\exists$ (egztenciális), és $\varphi(x_1, x_2, \dots, x_n)$ egy nulladrendű logikai formula olyan változókkal, amiknek az értékei igaz és hamis lehet csak. A QBF eldöntési problélma azt jelenti, hogy algoritmikusan eldöntjük, hogy a formula értéke igaz-e valamire vagy sose igaz.

ATM-en így dől el polinomidőben a QBF:

1. ha $Q_1 = \exists$, akkor egzisztenciális állapotba ugrik és lucky path-t keres a $Q_2x_2 \cdots Q_nx_n\varphi(\top, x_2, \ldots, x_n), Q_2x_2 \cdots Q_nx_n\varphi(\bot, x_2, \ldots, x_n)$ utak közül

2. ha $Q_1 = \forall$, akkor univerzális állapotba ugrik és paralell műveletek végez a $Q_2x_2 \cdots Q_nx_n\varphi(\top, x_2, \ldots, x_n), Q_2x_2 \cdots Q_nx_n\varphi(\bot, x_2, \ldots, x_n)$ lehetőségeken és elfogad, ha mindkettő elfogadó állapot

3. ha már nincs több kvantor, akkor ha nincs benne változó, egyszerűen kiértékeli polinomidőben, ha van, akkor azt nézi meg, hogy van-e igaz értéke, azaz egy SAT-ot old meg.

Világos, hogy(az exponenciális sok lehetőség dacára) a csodálatos ATM polinomidőben célhoz ér. Ha nincs nulladrendű formulában változó, akkor kvantorok száma + műveletek száma időben, ha vannak változók, akkor ehhez még egy SAT NP polinomidő is hozzáadódik.

4 Típus inhabitáció, bizonyításkeresés

A Curry–Howard-izomorfizmus szerint a típuselmélet és a (konstruktív/intuicionista) logika ugyanaz. Ez főként az egyszerű típuselméletnél (STT) szembeötlő, de bonyibbaknál (DTT, dependens típuselmélet) is megtalálható a párhuzam. Most csak az STT-re koncentrálunk, annak is a legpuritánabb, $\rightarrow$-t tartalmazó változatára. Erre készítünk egy eldöntési eljárást, azaz egy olyan algoritmut, ami eldönti, hogy egy feltételhalmazból egy álltás levezethető-e. Ez persze nem meglepő, mert ilyet a SAT is tud. Ami viszont jó, hogy a SAT nem skálázható, a kavantorokra még csak-csak van félig eldöntő eljárás de összetettebb adattípusokra nincs általánosítása. Szemben az STT-vel, amely az induktív típusokon keresztül messzemenőkig általánosítható. Persze nem ezt mondjuk, hogy létezik olyan általános eljárás, amely minden állításról eldönti, hogy igaz-e, mert ez a Church-tétel miatt nincs. De legalább féligeldöntő van, ami levezetést keres és ha van, megtalálha, ha nincs, fut a míg világ a világ és még három nap. Valójában az eldönthetőséggel úgy sem kezdhetünk sokat, mert az bonyi és annyi időnk úgy sincs, akár eldöntő (R-beli) akár féligeldntő (RE-beli) a probléma.

Lássuk tehát mit kell eldönteni. Van egy álltás az implikációs logika nyelvén, mondjuk $A \rightarrow (B \rightarrow A)$ és az a kérdés, van-e ennek bizonyítása. Mint az közismert két fő levezetésvagy termformer van, az egyik az applikáció $x\$y$, a másik a függvényképzés $\lambda x.P$. És van egy programfutás modell, ami a β-redukció:

$$(\lambda x.P)\$y \rightarrow P[y \rightarrow x]$$

ha ilyen már nincs egy levezetéstermben, akkor az egy normál term és így néz ki:

$$\lambda x.\lambda y.\ldots.(z\$(P\$Q\ldots))$$

ahol $P, Q, \ldots$ már nrmál termek. Na, ilyen alakú termre hajtunk az elábbi eljárásban.

Tétel: Létezik egy olyan APTIME algoritmus, ami eldönti, hogy egy A típusnak (vagy állításnak) van-e a Γ kontextusban lakója (feltételhalmazból levezetése).

Bizonyítás: ATM-en futtatjuk

1. Ha $A = A_1 \to A_2$, akkor bővítsük Γ-t A_1-gyel: A_1, Γ, majd generáljuk kimenetként, hogy $\lambda.x : A_1?$. Meghívjuk $A_1, \Gamma \vdash ? : A_2$ és tárojuk, hogy meghítuk.

2. Ha A atomi típus, akkor *nem-determinisztikus* módon választunk Γ-ban egy elemet, amely
$$A_1 \to A_2 \to \cdots \to A_{n+1}$$
alakú és $A_{n+1} = A$. Ha nincs ilyen típus, akkor elutasítunk. Ha $n = 0$, akkor elfogadjuk. Ha $n > 0$, akkor minden $i < n$-re meghívjuk $\Gamma \vdash ?_i : A_i$-t *paralell módon* és tároljuk ezeket a kérdéseket, majd szépen sorjában, ha van, generáljuk $\Gamma \vdash z\$P_0P_1\ldots$-t, ahol z változó

Ha bármely lépésben a kérdés már szerepel a tárolóban, akkor elutasítjuk, hogy elkerüljük a ciklusokat. A kontextusok csak bővülhetnek, egygyel és a hívások is korlátos halmazból táplálkoztahnak egyért egy idő után minden létezi kérdése sor kerül.

Ezt az eljárást sokat használtuk a korábbiakban. Lásd ilyen példákat. Figyeljünk fel arra, hogy akad el az algoritmus.