

# Statikus programellenőrzés

## Absztrakt interpretáció nagyhatékonyságú logikai programozási technikákkal

A hardver elemek hatalmas fejlődési ütemének köszönhetően a használt szoftverek mérete is robbanásszerűen növekszik. Az ilyen nagy programok tervezése, módosítása és karbantartása jelentős költséggel jár, ezért a szoftver megbízhatóság problémája egyre központibb szerepet tölt majd be a modern, számítógép-függő társadalmakban.

Rengeteg különböző módszert alkalmaznak a programok helyességének ellenőrzésére. Ezeknek egyik csoportja a nyomkövetés (debugging), vagyis a program végrehajtása (vagy szimulációja) sok különböző környezetben. Ezzel a módszerrel azonban manapság csak a valós környezetek töredékét lehet lefedni.

A formális program-ellenőrzési módszerek célja, hogy *bebizonyítsák*, hogy a program helyesen fut *minden* lehetséges környezetben. Ezen módszerek egyike a **statikus programellenőrzés**. Mivel a program-helyesség kérdése általános esetben eldönthetetlen probléma, minden formális módszer egyfajta approximációval él. Az **absztrakt interpretáció** célja ennek az approximációnak a formalizálása.

Az absztrakt interpretáció során egy programhoz hozzárendeljük annak egy absztrakt megfelelőjét, melyben az adatoknak csak egy absztrakcióját ismerjük és a program metódusainak megfelelői az ezen absztrakt értékeken operáló, absztrakt metódusok lesznek. Természetesen ez információvesztéssel jár, de az absztrakció mértékétől függően lehetővé válhat a teljes keresési tér bejárása és így bizonyos megbízhatósági/helyességi kérdések megválaszolása. Különböző jellegű kérdések eldöntésére különböző típusú absztrakciók lehetnek szükségesek. Ebből is látszik, hogy az absztrakt interpretáció egyik kulcsfontosságú kérdése a megfelelő absztrakció(k) megtalálása.

A diplomamunka céljai:

1. Az XSB Prolog logikai programozási nyelv megismerése. Az absztrakt interpretáció során az absztrakt metódusok eredményeinek meghatározásakor az egyik kulcs lépés a fixpont-számítás. Az XSB — *táblázási* képességének következtében — nagy segítséget nyújthat a fixpont-számítás automata-tízálásában.
2. A különböző kényszer-kielégítési logikai programozási módszerek (különösen CLP(FD), CLP(Q) és CLP(R)) alkalmasságának megvizsgálása az absztrakt tartományokon fellépő **kényszerek terjesztésére**.
3. A táblázás és kényszer-terjesztés technikáinak ötvözése az XSB rendszer kiterjesztésével és/vagy az XSB és egy másik CLP környezet integrálásával.
4. Absztrakt interpreterek implementálása, melyek alkalmasak egyszerű imperatív, funkcionális illetve logikai programok statikus ellenőrzésére. Későbbiekben absztrakt interpreterek alkalmazása egyéb vezérlési struktúrákra (például UML statechart-okra vagy tetszőleges vezérlési gráfokra).
5. Ezen absztrakt interpretációs módszerek hatékonyságának vizsgálata, más hasonló módszerekkel való összehasonlítása.