

Tervezési tér bejárás - DSE

Önálló laboratórium 1 (MSc) - féléves beszámoló

A tervezési tér bejárás (Design Space Exploration) nem más, mint egy keresési algoritmus, amelynek a bemenetei a következők: 1) Egy kezdeti gráf modell, amely lehet például egy rendszer állapota. 2) Gráf transzformációs szabályok, amelyek alkalmazásával a kezdeti állapotból újabb állapotokat érhetünk el. 3) Egy kívánatos cél állapot. 4) Kényszerek, amelyeknek minden a kezdeti és cél állapot közti állapotban teljesülniük kell. Az algoritmus kimenete pedig nem más, mint a bemeneti transzformációs szabályoknak egy olyan sorozata, amely a kezdeti állapotot a cél állapotba juttatja és közben a kényszerek nem sérülnek.

Felhasználási területe széleskörű lehet. Címszavakban néhány példa: 1) Automatikus erőforrás allokáció nagy rendszerekben. 2) Gyorsjavítás (quick-fix) generálás azaz újra tervezés grafikus nyelvek felett.

Magának az algoritmusnak több optimalizációs lehetősége is van. Úgy mint az állapot kódolás és valamiféle heurisztika a célállapot gyors megtalálására. Ez utóbbit különféle statikus analízis során kaphatjuk meg.

A félév során, azon kívül, hogy utána olvastam az algoritmusnak, új technológiákkal ismerkedtem meg, amelyek szükségesek az algoritmus implementálásához a konzulensem által javasolt környezetben. Új volt számomra az Eclipse környezetben való fejlesztés, az Eclipse plugin fejlesztés, az EMF (Eclipse Modeling Framework), illetve a tanszéken fejlesztett EMF-IncQuery. Ezen technológiákkal néhány gyakorló példán keresztül ismerkedtem meg. Először felrajzoltam a Petri háló meta modelljét EMF-ben, majd elkészítettem egy IncQuery mintát, ami megtalálja egy Petri háló tüzelhető tranzícióit. Ezek után elkészítettem egy példány modellt és egy szimulátort, ami a példány modellen addig tüzelte a tüzelhető tranzíciókat, amíg nem jutott holtpontra. Ezt a szimulátort megvalósítottam EVM (Event-driven Virtual Machine) felett is, amely az IncQuery projekt része és egy új réteget képez a modell felett, annak változásairól. Mivel a DSE algoritmusnak is fel kell iratkoznia a modell változásaira, ezért volt célszerű ilyen módon megismerkednem az EVM-mel is.

Utolsó lépésként megvalósítottam egy buta DSE algoritmust egy relaxált problémán. Az algoritmus képes kiszámolni, hogy hogyan milyen tüzelési szekvenciával lehet egy kezdeti Petri hálóból egy kívánt cél állapotba eljutni, amelyet egy token eloszlással lehet megadni. Az algoritmust azért neveztem butának, mert nincsen optimalizálva (például a már említett struktúrális analízissel), illetve a probléma azért relaxált, mert nem jelenik meg minden aspektusa egy lehetséges inputnak. Például a modellben nem jelenhetnek meg új objektumok (a token attribútumban tárolódik), illetve csak egy féle transzformáció létezik.

Összefoglalásképpen, megismerkedtem egy komplex algoritmussal, illetve számomra új, hasznos technológiákat sajátíthattam el. A jövőben célom az algoritmus implementálása egy általános keretrendszer formájában.