Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Modelltranszformációk formális analízise

Szakdolgozat

Készítette
Semeráth Oszkár

Konzulens
dr. Varró Dániel

2011. május 20.

FELADATKIÍRÁS

A feladatkiírást a tanszéki adminisztrációban lehet átvenni, és a leadott munkába eredeti, tanszéki pecséttel ellátott és a tanszékvezető által aláírt lapot kell belefűzni (ezen oldal *helyett*, ez az oldal csak útmutatás). Az elektronikusan feltöltött dolgozatban már nem kell beleszerkeszteni ezt a feladatkiírást.

HALLGATÓI NYILATKOZAT

Alulírott *Semeráth Oszkár*, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2011. május 20.

Semeráth Oszkár
hallgató

Tartalomjegyzék

Kivonat

Modellvezérelt tervezés során egy mérnöki modellből modelltranszformáció segítségével származtatunk egy analízis modellt. A modelltranszformációkban lévő hibák viszont érvénytelenné tehetik a matematikai modell precizitásából származó előnyöket. Így a modelltranszformációk hibamentességének biztosítását célszerű formális verifikációval garantálni.

A gráftranszformációk paradigmája egy intuitív, jól tervezhető, és egyben precíz formális módszert ad a modelltranszformáció megvalósításához. A gráftranszformációs rendszerek potenciálisan végtelen állapottérrel járnak, amely jelentősen gátolja a helyességellenőrzésüket. A shape analízis módszerével akár végtelen állapotterű gráftranszformációs rendszerek is ellenőrizhetőek. Ekkor állapotterünk gráfjait véges számú csoportokba rendszerezzük, a csoportokat címkézett gráfok speciális halmazával, shape gráfokkal ábrázoljuk. A gráftranszformációs lépések absztrakt megfelelőjét végigvezetve a shape gráfokon absztrakt állapotteret kapunk, amely analízisével bizonyíthatjuk gráfnyelvtanunk helyességét.

A szakirodalomban két fő irányvonal található a shape gráfok alkalmazására. Az egyik úgynevezett neighbourhood shape gráfok módszere a benne lévő csúcsok környezete alapján osztályozza gráfjainkat, míg a másik, állapotainkat háromértékű logikai struktúraként ábrázolja. Ezek a módszerek a különböző karakterisztikájú problémákat eltérő hatékonysággal képesek megoldani.

Szakdolgozatomban bemutatom, hogy e módszerek megadhatóak közös, gráfelemek ekvivalenciáján alapuló definícióval, melyben megfelelő paraméterezéssel mindkét absztrakció ábrázolható. A közös leírással elérhető, hogy a feladat ismeretében, a paraméterezés kombinálásával, a módszerek erősségét kiemelve javíthassunk az analízis kivitelezhetőségének és sikerességének esélyén.

A közös módszer VIATRA2 modelltranszformációs rendszer fölé tervezett megvalósíthatóságáról tanulmányt végeztem, lévén, hogy a rendszer alkalmas nagy modellterek kezelésére, illetve az inkrementális gráfmintaillesztés lehetőségét használva hatékonyan hajthatjuk végre a szükséges műveleteket.

Abstract

In model-driven desing an analysis model is derived from an engeenering model by a model transformation. The errors of the model transformations can invalidate the advantages of precision of the mathematical model. So adviseable to garantee the fail-safeness of model transformations by formal verification.

The paradigm of graph transformations offers an intuitive, well designable and as well precise formal method to specify model transformations. Graph transformation systems potentially induce an infinite state space, which significantly hinders formal verification to guarantee their correctness. Shape analysis is one of the techniques which addresses the formal verification of such infinite state systems. In shape analysis the graphs of the state space are categorised into finite number of groups. Applying the abstract equivalent of graph transformation rules on the shapes an abstract state space is got, which on which one can prove the correctness of graph languages.

There are two main trend in the literature for shape analysis. The first, so called neighbourhood shapes categorises the graphs by the context of their nodes, the second represents the states graph by 3-valued logic structures. These methods can solve problems with different characterisation with variant efficiency.

In the current thesis, I propose to unify the definition of these techniques based on the equivalence of of the graph elements, where both abstractions can be represented by a proper parameterisation. With the unified description, in the view of the problem and by the combination of the parameters the the probability of succesful and realizable analysis can be increased.

In the this thesis I made a feasibility study of the realisation of the common method on VIATRA2 model transformation system, because it able to manage big model spaces and with the use of incremental pattern matching the operations are executed effectively.

1. fejezet

Bevezetés

1.1. Modellvezérelt tervezés

Biztonságkritikus rendszerek tervezése során, amelyekben az elkészült programoknak bizonyítottan teljesíteniük kell egy elvárt viselkedést, gyakorta alkalmazzák a modellvezérelt tervezés módszerét [1, 7]. Ennek során a mérnöki modellből származtatunk egy precíz matematikai modellt is, amin már a tervezés korai szakaszában elkezdődhet a leendő program helyességének ellenőrzése. A két modell közti konzisztenciát a modelltranszformációk biztosítják.

A modelltranszformációknak tehát olyan feltételeket kell teljesíteniük, amelyek által az analízis modell helyességéből következik a mérnöki modell helyessége is. Fontos továbbá, hogy a modelltranszformációk implementációja megvalósítsa a modell szinten megfogalmazott specifikációt. Ennek hiányában ugyanis nem lehetne belátni, hogy a konkrét megvalósítás helyessége következik a modellekéből.

A gráftranszformációk alkalmazása egy intuitív, jól tervezhető, és egyben precíz formális módszert ad a modelltranszformációk megvalósításához [14]. Ebből következően ezek verifikációja kutatásra érdemes terület.

1.2. Feladat bemutatása

1.2.1. Gráfnyelvtanok analízise

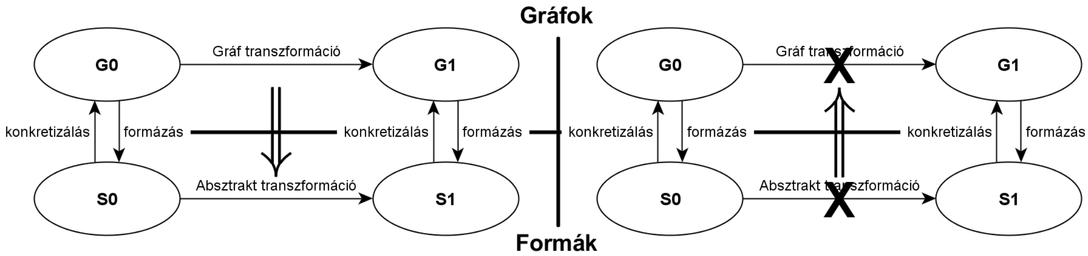
Egy rendszer állapotait modellezhetjük címkézett gráfokkal, a viselkedését gráftranszformációkkal. Egy gráftranszformációs szabályt két gráf ír le: egy bal oldali és egy jobb oldali. A transzformáció egy G gráfon történő végrehajtása során megkeressük G-nek a transzformáció bal oldalára illeszkedő részgráfját, és kicseréljük azt a transzformáció jobb oldalára.

A címkehalmazok, egy G_0 kiindulási állapot és R gráftranszformációs szabályok halmaza meghatároz egy gráfnyelvtant. Egy szolgáltatásbiztos rendszer tervezése során felmerülhet az igény, hogy egy adott gráfnyelvtan kiindulási állapotából annak gráftranszformációs szabályait alkalmazva eljuthatunk-e nemkívánatos állapotba, amelyet definiálhatunk úgy, hogy tartalmaz-e a gráf egy F tiltott gráffal izomorf részgráfot. Esetemben egy ezzel megegyező kifejezőerejű feltételt használok, miszerint tüzelhet-e a $P_F = (F, F)$ transzformáció, ahol a bal oldal és a jobb oldal is a tiltott gráf.

A G_0 kiindulási gráf és a R gráftranszformációs szabályhalmaz felépít egy állapotteret, amiben a G_0-ból elérhető állapotok, és az állapotok közötti R-beli transzformációk szerepelnek. Az állapottér elemzése során nagy problémát jelent, hogy mérete végtelen is lehet, így az összes elérhető gráf ellenőrzése kimerítő bejárással lehetetlen.

1.2.2. Absztrakció alapú analízis

Gráfok véges csoportokba történő rendezésével javítható a végtelen állapottér kezelhetetlensége. A csoportokat címkézett gráfok egy speciális halmazával, úgynevezett *formákkal* (angol szakirodalomban *shape*) ábrázoljuk. Egy gráf egy forma által leírt csoportba történő osztályzását *formázásnak* (angolul *shaping*), egy formához tartozó gráfokat a forma *konkretizáltjának* nevezzük. Fontos megjegyezni, hogy egy formának lehet 0, véges sok, vagy akár végtelen konkretizáltja is.



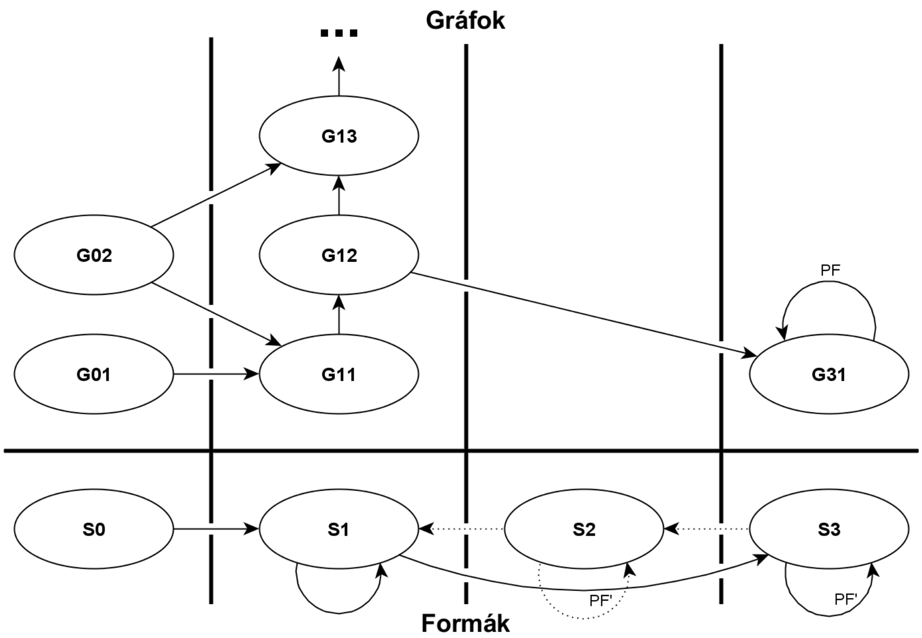
1.1. ábra. *Konkrét és forma gráfokon értelmezett transzformációk*

Formákon is értelmezzünk egy transzformációt, amit *formatranszformációnak* nevezünk. Ahhoz hogy jól kezelhető absztrakciót kapjunk, teljesülnie kell egy feltételnek: ha a G_0-án egy gráftranszformációt alkalmazva G_1-et kapjuk, és G_0-hoz S_0, G_1-hez S_1 forma tartozik, akkor az S_0-án végrehajtva a transzformáció absztrakt megfelelőjét az S_1-et is meg kell hogy kapjuk. Más szavakkal: formák között legalább akkor kell vezetnie formatranszfromációnak, ha léteznek olyan konkretizáltjaik, amelyek között vezet gráftranszformáció. Ezt a megkötést a 1.1 ábra bal oldali diagramja[1] szemlélteti.

A fenti megkötésből következik egy hasznos állítás: ha egy formán nem alkalmazható egy absztrakt transzformáció, akkor a formák konkretizáltjain sem alkalmazható annak gráftranszformáció megfelelője. Ez az állítás figyelhető meg a 1.1 ábra jobb oldali diagramján.

Az állapotterekhez hasonlóan definiálhatunk absztrakt állapottereket is, melyekben egy kiindulási formából absztrakt transzformációkkal elérhető formák szerepelnek. A formatranszformációk definíciója miatt ebben a gráftranszformációkkal elérhető összes állapotnak megfelelő forma szerepel, illetve az absztrakció miatt tipikusan más absztrakt állapotok is megjelennek. A konkrét és absztrakt állapotterek viszonyát a 1.2 ábra szemlélteti. A formák véges számosságából adódóan lehetőség nyílik az absztrakt terünk felépítésére és elemzésére. Abban az esetben, ha tiltott mintát tartalmazó P_F transzformáció absztrakt megfelelője nem szerepel benne, bizonyítottá válik, hogy rendszerünk a kiinduló forma egy konkretizáltjából sem kerülhet nemkívánatos állapotba. Ha szerepel, akkor abból általános

[1] A szakdolgozat során bemutatott ábrákat yEd diagramszerkesztővel készítettem [15].



1.2. ábra. *Konkrét (felső) és absztrakt (alsó) állapotterek viszonya. Egy x csoportba absztrahált G_{xy} gráfok egy oszlopba kerültek, a csoportot az S_x forma reprezentálja. A tiltott P_F átmenetet címke jelöli, a gráftranszformációk során nem létező átmenetet pedig pontozott vonal.*

esetben sajnos semmilyen következtetést nem vonhatunk le.

Az absztrakt állapottereken végzett analízis sikeressége tehát nagyban múlik az alábbi két dologtól:

1. **Absztrakció pontossága**: Az, hogy egy forma milyen pontosan határozza meg konkretizáltjait, fontos kérdés. Ha túl általános, akkor a pontatlanság miatt lehetetlen lesz megcáfolni a tiltott minta előfordulását, ha viszont szükségtelenül precíz, akkor a formák megnövekvő számossága miatt kivitelezhetetlen lesz az absztrakt állapottér felépítése.

2. **Absztrakt transzformáció pontossága**: Az definíció szerint adott, hogy egy absztrakt transzformációt alkalmazva az eredmények halmazában mely formáknak kell legalább szerepelniük. Ajánlatos viszont arra törekednünk, hogy ezen megkötés mellett transzformációnk minél kevesebb olyan formát adjon eredményül, amely konkretizáltjai között nem létezett volna gráftranszformáció. A felesleges absztrakt állapotátmenetek ugyanis egyrészt megnövelik az állapottér méretét, másrészt indokolatlanul olyan állapotba vihetik azt, amellyel meghiúsulhat a helyességellenőrzés.

Ezen problémák kezelhetősége érdekében a formázásnak jól paraméterezhetőnek kell lennie. Szerencsés esetben a kiinduló állapot, az elvárt viselkedés során létrejövő állapotok, a gráftranszformációk és a tiltott minta alapján történő paraméterezés esetén olyan formákat kapunk, amelyekben a felhasználás szempontjából érdekes adatokat ábrázolják formáink, az érdektelenek pedig eltűnnek, így csökkentve a formák számosságát.

1.3. Kontribúció

A szakdolgozatom írásával az alábbi pontokban leírtakkal járultam hozzá a gráftranszformációk helyességellenőrzésének kibővítéséhez:

1. A szakirodalomban eddig használt két fő irányvonalak a szomszédságon alapuló formák, és a háromértékű logikát használó formák. Ezeket közös, gráfelemek ekvivalenciáján alapuló definícióval adtam meg.

2. Megadtam egy olyan absztrakt transzformátor vázát, amelyből a módszerekben használt ekvivalenciákból származó összefüggések felismerése és alkalmazása mellett megkapjuk a szakirodalom főbb irányvonalaiban leírt transzformátorokat.

3. Megvizsgáltam a Viatra2 modelltranszformációs keretrendszer fölé tervezett megvalósíthatóságot.

1.4. Dolgozat felépítése

Dolgozatom a következő fejezetek alapján építettem fel:

A *második fejezetben* a címkézett gráfok, ezek transzformációi, valamint az ezáltal felépíthető állapottér fogalma kerül megadásra.

A *harmadik fejezetben* bemutatom és definiálom a dolgozat során gráfok absztrakciójaként használt formákat.

A *negyedik fejezetben* definiálásra kerülnek a szomszédsági formákban paraméterként használt ekvivalenciák. Ezek a szomszédsági ekvivalencia, amely a csúcsokat hasonló környezetük alapján osztályozza, valamint a csúcs- és élmultiplicitások, amelyekkel a megegyező csoportok számosságát jelölhetjük.

Az *ötödik fejezetben* a háromértékű logikát használó predikátum formák leírása történik, ahol az adott predikátumok igazságértékeinek megegyezése esetén vonunk össze csúcsokat.

A *hatodik fejezetben* az absztrakt gráftranszformáció és transzformátor fogalmai kerülnek tisztázásra. Itt megadok egy olyan transzformátort, amelyet a szakirodalomban használt módszerek alkalmaznak.

A *hetedik fejezetben* a VIATRA2 rendszer fölé tervezett megvalósíthatósági tanulmány szerepel. Példát adok az absztrakt állapottér tárolási lehetőségére, meghatározásra kerülnek a reprezentációk és ekvivalenciák szükséges felelősségei, valamint leírásra kerül az analízis során használt algoritmusok vázai.

2. fejezet

Matematikai definíciók

Ebben a fejezetben a címkézett gráfok, a gráftranszfromációk, az ezek által felépített állapottér, és még több, általálnosan használt fogalom formális definíciói szerepelnek.

2.1. Címkézett gráf

A dolgozat olyan véges gráfokat tárgyal, melyneknek csúcsait és éleit adott véges Lab^N és Lab^E halmazok elemeivel címkézzük. A gráfok élei irányítottak, párhuzamos és hurokélek megengedettek.

A címkézett gráfok matematikai definíciója, amely tartalmazza a végtelen gráfokra történő kiterjeszthetőséget a következő:

1. Definíció (Címkézett gráf). *Egy* címkézett gráf *az alábbi struktúra:*

$$G = (N, Lab^N, lab^N, E, src, trg, Lab^E, lab^E)$$

ahol:

N	–	*A* csúcsok *halmaza.*
Lab^N	–	*A* csúcscímkék *halmaza.*
lab^N	–	*A* csúcsok címkézése. $lab^N : N \mapsto Lab^N$*, azaz minden csúcshoz hozzárendel egy csúcscímkét.*
E	–	*Az* élek *halmaza. N és E diszjunkt.*
src	–	*Az élek* forrása. $src : E \mapsto N$*, azaz minden élhez hozzárendel forrásául egy csúcsot.*
trg	–	*Az élek* célja. $trg : E \mapsto N$*, azaz minden élhez hozzárendel céljául egy csúcsot.*
Lab^E	–	*Az* élcímkék *halmaza.*
lab^E	–	*Az* élek címkézése. $lab^E : E \mapsto Lab^E$*, azaz minden élhez hozzárendel egy élcímkét.*

Maga a G gráf tipikusan véges, ekkor N, Lab^N, E és Lab^E véges halmazok. A Lab^N-nel és Lab^E-vel címkézett gráfok halmazát jelöljük $\mathcal{G}(Lab^N, Lab^E)$-vel, az összes címkézett gráfot tartalmazó halmazt pedig $\mathcal{G}$-vel. Továbbá jelezzük a összes gráf csúcsait $\mathcal{N}$-nel, éleit $\mathcal{E}$-vel.

Egy gráf struktúrájának egy elemét jelöljük a gráf nevével indexelt elemnévvel. Tehát például N_G a G gráf csúcsainak halmaza.

A továbbiakban szükség lesz egy olyan leképezésre, amely egy gráf csúcsait és éleit egy másik gráféhoz tudja rendelni. Ennél a függvénynél kikötjük, hogy csúcsokat csúcsokhoz, éleket élekhez rendeljen, és ha egy csúcs egy élnek forrása vagy célja volt, akkor ez igaz legyen a hozzájuk rendelt gráfelemekre is. Ezt a függvényt morfizmusnak nevezzük.

2. Definíció (Morfizmus). *Legyenek $G, H \in \mathcal{G}$. Ekkor a G és H gráfok közötti morfizmuson azt az $m : N_G \cup E_G \mapsto N_H \cup E_H$ függvényt értjük, amelyre teljesülnek az alábbi feltételek:*

- *m csúcsokhoz csúcsokat, élekhez éleket rendel:*

$$\forall n(n \in N_G \Rightarrow m(n) \in N_H) \wedge \forall e(e \in E_G \Rightarrow m(e) \in E_H)$$

- *ha G-ben egy n csúcs egy e élnek forrása vagy célja volt, ez maradjon igaz $m(n)$-re és $m(e)$-re is:*

$$\forall n \in N_G, \forall e \in E_G \Big(\big(n = src_G(e) \Rightarrow m(n) = src_H(m(e)) \big) \wedge \\ \big(n = trg_G(e) \Rightarrow m(n) = trg_H(m(e)) \big) \Big)$$

Az m-et címkehelyes morfizmusnak nevezzük, ha ezeken kívül az alábbi feltételek is teljesülnek:

- *G és H címkéi közösek: $Lab^N_G = Lab^N_H \wedge Lab^E_G = Lab^E_H$*
- *A csúcsok és élek címkéi ugyanazok maradnak:*

$$\forall n \in N_G \Big(lab^N_G(n) = lab^N_H(m(n)) \Big) \wedge \forall e \in E_G \Big(lab^E_G(e) = lab^E_H(m(e)) \Big)$$

Az m morfizmus esetén m^N jelentse csak a csúcsokon értelmezett függvényt, m^E pedig csak az éleken értelmezettet.

Egy morfizmus lehet injektív, szürjektív vagy bijektív függvény.

3. Definíció (Injektív, szürjektív, bijektív függvény). *Legyen $f : A \mapsto B$ függvény. Azt mondjuk, hogy az f függvény* injektív*, ha minden $x, y \in A$-ra teljesül, hogy $f(x) = f(y)$ esetén $x = y$. Az f függvény* szürjektív*, ha minden $b \in B$-re teljesül, hogy létezik olyan $x \in A$ amelyre $f(x) = b$. Az f függvény* bijektív*, ha injektív és szürjektív egyszerre.*

Két gráf közötti bijektív címkehelyes morfizmus léte meghatároz egy gráfokon értelmezett ekvivalenciarelációt, az izomorfiát.

4. Definíció (Gráf izomorfia). *Két címkézett gráf akkor* izomorf*, ha létezik bijektív címkehelyes morfizmus a két gráf között.*

A későbbiek során két gráfot akkor tekintünk azonosnak, ha izomorfak.

2.2. Gráftranszformáció

Egy gráfnak fontos tulajdonsága, hogy megtalálható-e benne egy bizonyos minta. Minta alatt egy címkézett gráfot értünk, ami akkor illeszkedik egy másik gráfra, ha megtalálható benne részgráfként.

5. Definíció (Gráf mintaillesztés). *Legyenek $P, G \in \mathcal{G}(Lab^N, Lab^E)$. Azt mondjuk, hogy a P minta m mentén illeszkedik G-re, és ezt $P \xrightarrow{m} G$-vel jelöljük, ha m egy P és G közötti címkehelyes injektív morfizmus.*

A gráftranszformációk[5] címkézett gráfok feletti műveletek, egy gráfból a transzformációt alkalmazva egy új gráfot kapunk. Egy ilyen műveletet egy gráftranszformációs szabály ír le.

6. Definíció (Gráftranszformációs szabály). Gráftranszformációs szabálynak *nevezzük azt a $P = (L, R)$ párost, ahol L és R (tipikusan nem diszjunkt) ugyanazon címkekészlettel címkézett gráfok. P-nek az L gráfot a* bal oldalának*, az R-t* jobb oldalának *nevezzük.*

Egy $P = (L, R)$ gráftranszformációs szabály esetén különböztessük meg az alábbi, P-vel indexelt halmazokat:

- A törlődő csúcsokat $N_P^{del} = N_L \setminus N_R$, a törlődő éleket $E_P^{del} = E_L \setminus E_R$ jelöli.
- Az új csúcsokat $N_P^{new} = N_R \setminus N_L$, az új éleket $E_P^{new} = E_R \setminus E_L$ jelöli.

A gráftranszformáció végrehajtásakor megkeresünk egy bal oldallal izomorf részgráfot, majd azt kicseréljük a jobb oldali gráffal megegyezőre. A dolgozat során az úgynevezett *double-pushout* (DPO)[2, 5] megközelítést követem, miszerint a transzformáció nem hajtható végre, ha ezzel egy él elveszítené forrását vagy célját.

7. Definíció (Gráftranszformáció). *Legyen G egy címkézett gráf, $P = (L, R)$ egy gráftranszformációs szabály, G elemei diszjunktak L és R elemeitől. Legyen $P \xrightarrow{m} G$ amire igaz, hogy P nem törlődő éle nem illeszkedik törlődő csúcshoz:*

$$\forall e \in E_G \Big(src_G(e) \in m(N_P^{del}) \vee trg_G(e) \in m(N_P^{del}) \Rightarrow e \in m(E_P^{del}) \Big)$$

A P transzformációs szabályt m illeszkedéssel G-n alkalmazva a H címkézett gráfot kapjuk, és ezt $G \xrightarrow{P,m} H$-val jelöljük, ha

$$
\begin{array}{rcl}
N_H & = & (N_G \setminus m(N_P^{del})) \cup N_P^{new} \\
Lab_H^N & = & Lab_G^N \\
lab_H^N(n) & = & \begin{cases} lab_G^N(n) & ha\ n \in N_G \setminus m(N_P^{del}) \\ lab_R^N(n) & ha\ n \in N_P^{new} \end{cases} \\
E_H & = & (E_G \setminus m(E_P^{del})) \cup E_P^{new} \\
src_H(e) & = & \begin{cases} src_H(e) & ha\ e \in E_G \setminus m(E_P^{del}) \\ src_R(e) & ha\ e \in E_P^{new} \wedge src_R(e) \in N_P^{new} \\ m(src_R(e)) & ha\ e \in E_P^{new} \wedge src_R(e) \notin N_P^{new} \end{cases} \\
trg_H(e) & = & \begin{cases} trg_H(e) & ha\ e \in E_G \setminus m(E_P^{del}) \\ trg_R(e) & ha\ e \in E_P^{new} \wedge trg_R(e) \in N_P^{new} \\ m(trg_R(e)) & ha\ e \in E_P^{new} \wedge src_R(e) \notin N_P^{new} \end{cases} \\
Lab_H^E & = & Lab_G^E \\
lab_H^E(e) & = & \begin{cases} lab_G^E(e) & ha\ e \in E_G \setminus m(E_P^{del}) \\ lab_R^E(e) & ha\ e \in E_P^{new} \end{cases}
\end{array}
$$

2.3. Állapottér

A gráftranszformációk elemzésénél fontos, hogy egy adott kiinduló gráfból, adott transzformációkat tartalmazó halmaz elemeit egymás után alkalmazva melyik gráfok kaphatók meg eredményül.

8. Definíció (Elérhető állapot). *Legyen $G_0, H \in \mathcal{G}(Lab^N, Lab^E)$ címkézett gráf, R pedig gráftranszformációs szabályok halmaza. Ekkor G_0-ból R-beli transzformációkkal elérhető H (jelölésben: $G_0 \xrightarrow{R} H$), ha léteznek olyan $P_0, \ldots, P_n \in R$ transzformációk, $m_0, \ldots m_n$ illeszkedések és $G_1, \ldots G_n \in \mathcal{G}(Lab^N, Lab^E)$ címkézett gráfok,hogy:*

$$G_0 \xrightarrow{P_0,m_0} G_1, G_1 \xrightarrow{P_1,m_1} G_2, \ldots, G_n \xrightarrow{P_n,m_n} H$$

Egy gráf önmagából definíció szerint elérhető.

Egy gráfból elérhető összes gráf állapotteret épít. Az állapotteret egy állapotokkal csúcscímkézett és transzformációkkal élcímkézett gráfként ábrázoljuk, amiben akkor vezet egy P transzformációval címkézett él két csúcs között, ha az egyik csúcs gráf címkéjén alkalmazva a P transzformációt megkapjuk a másik címkéje által ábrázolt gráfot.

9. Definíció (Állapottér). *Legyen $G_0 \in \mathcal{G}(Lab^N, Lab^E)$ címkézett kiinduló gráf, R pedig gráftranszformációs szabályok halmaza. Ekkor a $states(G_0, R)$ függvény eredménye egy olyan $STATES \in \mathcal{G}(\mathcal{G}(Lab^N, Lab^E), R)$ gráfokkal csúcscímkézett és transzformációkkal élcímkézett gráf, amelyre az alábbi állítások igazak:*

- *Minden csúcscímke csak egyszer szerepel, más szóval lab^N_{STATES} injektív.*
- *Csúcsai azon G gráfokkal vannak címkézve, amelyekre teljesül, hogy $G_0 \xrightarrow{R} G$.*
- *Minden olyan G és H gráfhoz, m morfizmushoz és $P \in R$ transzformációhoz, amelyre teljesülnek, hogy $G_0 \xrightarrow{R} G$, $G_0 \xrightarrow{R} H$ és $G \xrightarrow{P,m} H$ pontosan egy él tartozik. Ennek az*

élnek a címkéje P, irányát tekintve a G-vel címkézett csúcsból vezet a H-val címkézett csúcsba.

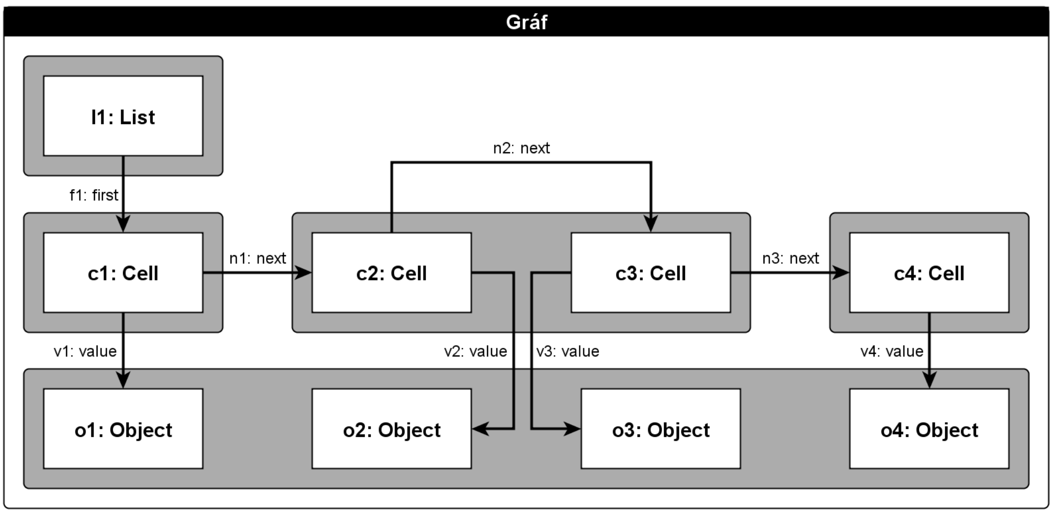
3. fejezet

Ekvivalencia alapú absztrakció

Ebben a fejezetben bemutatom és defininiálom a dolgozat során gráf absztrakcióként használt formákat. A formák ekvivalenciarelációkkal szabadon paraméterezhetően lesznek megadva, így biztosítva a feladatra szabhatóságot. A fejezet tárgyalja a címkézett gráfok formákkal történő osztályozását is, a formázást, és ennek inverz műveletét, a konkretizációt is.

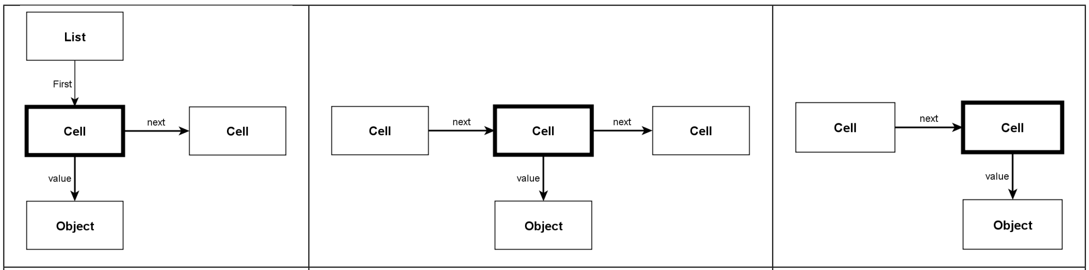
3.1. Gráfelemek csoportosítása közös tulajdonságuk alapján

A formák bemutatása során a 3.1 ábrán látható 4 elemű láncolt listát ábrázoló címkézett gráfra hivatkoznak a példák. A láncolt lista minden elemét egy-egy `Cell` cimkéjű gráfcsomópont reprezentálja, a lista láncolását `next` cimkéjű élek jelölik. A benne tárolt `Object` címkéjű értékeket `value` élek kötik a cellákhoz. Egy `List` címkéjű csúcs `first`-tel jelöli meg a lista első elemét. Az ábrán minden csúcsot és élt egyértelmű azonosító jelöl, melyet kettőspont után a gráfelem címkéje követ.



3.1. ábra. *Egy négyelemű láncolt listát ábrázoló címkézett gráf.*

A forma egy olyan absztrakciója a címkézett gráfoknak, amely leírja a konkretizáltjaiban szereplő gráfelemek tulajdonságait, osztályozza az ugyanolyan tulajdonságokkal rendelkező



Első cella	Középső cella	Utolsó cella
`c1`	`c2` és `c3`	`c4`

3.2. ábra. *A* `Cell` *címkéjű csúcsok csoportjai. Az első sorban a csúcsok néhány csoportjára jellemző szomszédság látható. Az aktuálisan bemutatott csúcs vastag kerettel van rajzolva. A második sorban a szomszédságnak nevet adunk. A harmadikban a csoportba tartozó csúcsok vannak megnevezve.*

csoportokat és meghatározza azok egymás közti viszonyát. A formák szintén címkézett gráfokként vannak ábrázolva.

A formák megismerésének első lépésében csoportosítsuk gráfunk hasonló tulajdonságokkal rendelkező csúcsait. Azok a csúcsok kerüljenek egy kategóriába, amelyek egy paraméterként adott csúcsok felett értelmezett kompatibilitási ekvivalencia szerint megegyeznek.

Példa A 3.1 ábrán szemléltetett példa csúcsait akkor rendeltük egy csoportba, ha az alábbi feltételek teljesülnek:

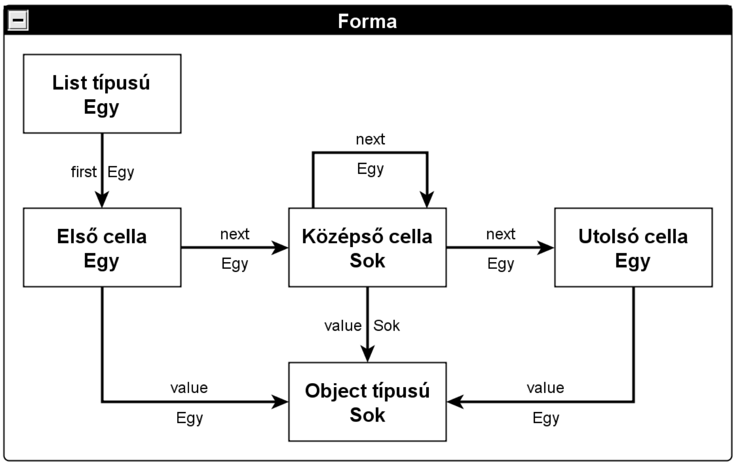
- Címkéik azonosak
- Szomszédságuk hasonló, azaz ugyanolyan címkéjű élekkel ugyanolyan címkéjű csúcsokkal vannak összekötve

A 3.1 ábrán az ezen feltételeknek megfelelően egy kategóriába sorolt csúcsokat bekeretezés jelöli. Megfigyelhető, hogy ezen kategorizálás szerint például különválasztjuk az első `c1` és az utolsó `c4` `Cell`-lel címkézett csúcs, de a két középső `c2` és `c3` cella összemosódnak. Ezen csúcsok kategorizálásában fontos tulajdonságait jegyzi fel a 3.2 ábra. A későbbiekben a csoportosítást meghatározó jellemzőket reprezentációnak fogjuk nevezni. Fontos észrevenni, hogy olyan csoportok, amelyek nem tartalmazzák a lista egyetlen csúcsát sem, nem jelennek meg az ábrán. Másképpen ezt úgy mondhatjuk, hogy az elemekhez a kategóriák szürjektíven lettek összerendelve.

A gráfunk éleit is csoportosítani kell. Az éleknél a csúcsokkal ellentétben nem adunk meg ekvivalenciarelációt, a kompatibilitási ekvivalencia szerint ugyanolyan csúcs-csoportok között futó, azonos címkékkel ellátott éleket rendelünk egy csoportba.

Példa A 3.1 ábrán egyedül a `v2` és a `v3` élek futnak ugyan azon csúcs-csoportok között, így ezeket összerendeljük. A csúcsok csoportjaihoz hasonlóan itt sem jelennek meg olyan csoportjai az éleknek, amelyeknek nincsenek elemeik, tehát az élekhez is szürjektíven vannak csoportok rendelve.

Ha a 3.1 ábrán látható címkézett gráfnak csak a csúcs- és élcsoportjait jelenítjük meg, a 3.3 ábrához hasonló címkézett gráfot kapunk. A csúcsok címkéi közül a felső jelöli a kompatibilitási, az alsó az instrumentális ekvivalencia osztályt. Éleknél hasonlóan az eredeti címke felül/bal oldalt, az instrumentális ekvivalencia osztály lent/jobb oldalt található.



3.3. ábra. *Forma gráf példa.*

Ezt a címkézett gráfot formának nevezzük. A formák csúcsait a kompatibilitási ekvivalencia osztályaival címkézzük, úgy, hogy ez a címke megegyezzen a csoportosítás során létrejövő kategóriával. A formák élei az eredeti gráf éleinek címkéivel címkéződnek.

A jelenleg ekvivalenciaosztályokkal és eredeti élcímkékkel címkézett formák kifejezőereje a következő:

1. Egy formabeli n_S csúcs egy nemüres[1] gráfbeli csúcshalmazt ír le. A csúcshalmaz minden elemére igaz, hogy a kompatibilitási ekvivalencia reláció n_S címkéjébe megadott osztályba esnek.

2. Egy formabeli e_S él olyan élekből álló nemüres halmazt biztosít, melyeknek forrásai e_S forrása által leírt, céljai e_S célja által leírt csúcshalmazban vannak, és címkéjük megegyezik e_S címkéjével.

Példa Vegyük a 3.3 ábrán látható forma azon csúcsát, ami a „középső cella" ekvivalenciaosztályba eső csúcsokat foglalja magába, és nevezzük n_S-nek. A csúcs jelenlétéből következik, hogy ezen forma által leírt gráfok mindegyikében szerepel néhány olyan csúcs, aminek szomszédsága megegyezik n_S által definiáltéval (lásd: 3.2 táblázat). A jólformált láncolt listák halmazából ez a tény már kizárja azokat az elemeket, amelyeknél a tárolt értékek száma kevesebb mint 3.

Vegyük az n_S-hez csatlakozó `next` címkéjű hurokélt. Az a tény, hogy legalább két középső cella osztályú csúcsot legalább egy `next` él köt össze, kizárja a 3 elemű listát is a lehetséges konkrét gráfok közül.

Annak ellenére, hogy a konkrét gráfokban előforduló csúcsok és élek tulajdonságait tetszőlegesen pontosan írhatják le ekvivalenciaoszályaik, szükség lehet az egy ekvivalenciaosz-

[1] A halmaz nemüressége a szürjektív kategorizálásból következik. Ha minden kategóriához rendelünk elemet, akkor minden kategóriához tartozik legalább egy hozzárendelt elem.

tályba eső elemek gyűjteményéről is állításokat feljegyezni a formákban. Például különbséget tehetünk két, egy ekvivalenciaosztályba tartozó csúcshalmaz között, ha a csúcshalmazok számossága eltérő.

A formák ezzel újabb két új tulajdonságukkal lesznek paraméterezhetőek. Először formák csúcsai között különbséget szeretnénk tenni egy csúcshalmaz fölött értelmezett ekvivalencia-relációval. Másodszor az élek jelölésére alkossunk egy hármast az élek halmazából, azok forrásainak- és céljainak halmazából, és ezen hármasok fölött is értelmezzünk egy ekvivalencia-relációt. Ezeket az új paramétereket nevezzük instrumentális ekvivalenciáknak. A formák a csúcsainak és éleinek címkéi ezzel még egy-egy ekvivalenciaosztályt tartalmaznak.

Példa A 3.3 ábrán látható forma a csúcshalmazokat két csoportra osztja: egyeleműekre és többeleműekre. A forma csúcsai jelzik, hogy kompatibilitási ekvivalenciaosztályba csak egyetlen (lásd: `Egy` címke), vagy több (`Sok`) csúcs esik a konkét gráfokban. A forma élei is csak azt mutatják, hogy a beleképzett élek számossága egy vagy több-e.

Miután megtudtuk, hogy a középső cella típusú élek között csupán egyetlen next él vezet, a jólformált láncolt listák közül egyedül a 4 elemmel rendelkező felel meg a formának.

Vegyük észre, hogy gráf absztrakció során a formázás által reprezentált, gráfok fölött értelmezett ekvivalenciával dolgozunk, azokat a gráfokat soroljuk egy csoportba (vagy a csoportot reprezentáló formába), amelyekhez ugyan azok a formák rendelhetőek.

3.2. Ekvivalenciareláció

A formák későbbi precíz definiálásához szükség lesz néhány ekvivalenciarelációval kapcsolatos fogalom bevezetésére. Mindezeket megelőzően az egyértelműség kedvéért lássuk az alapfogalmakat:

10. Definíció (Ekvivalenciareláció). *Egy H halmaz fölött értelmezett $\sim \subseteq H \times H$ reláció* ekvivalenciareláció*, ha $\sim$-ra az alábbi feltételek teljesülnek:*

1. *reflexív, azaz minden $a \in H$ esetén $a \sim a$*
2. *szimmetrikus, azaz minden $a, b \in H$ esetén $a \sim b \Rightarrow b \sim a$*
3. *tranzitív, azaz minden $a, b, c \in H$ esetén $a \sim b \wedge b \sim c \Rightarrow a \sim c$*

11. Definíció (Ekvivalenciaosztály). *Egy H halmaz fölött értelmezett egy $\sim \subseteq H \times H$ ekvivalenciareláció. Ekkor egy $a \in H$ elemhez tartozó ekvivalenciaosztály a következő:*

$$[a]_\sim = \{x | a \sim x\}$$

Ha $K \subseteq H$, a K halmaz elemeit tartalmazó ekvivalenciaosztályok pedig a kövezkezők:

$$K/\sim = \{[x]_\sim | x \in K\}$$

A formák kezelése során szükséges lesz, hogy egy halmaz elemeihez hozzárendeljük a hozzájuk tartozó ekvivalenciaosztályt, illetve ezen osztály szerint csopotosítsuk őket. A formák címkéinek ezeket az ekvivalenciaosztályokat tárolniuk kell, továbbá az osztályok tulajdonságaiból következtetéseket kell levonnunk. Ezek a feladatok matematikai feldolgozása körülményes, számítási és tárolási költségük magas, ábrázolásuk átláthatatlan.

A problémákra megoldást nyújt a reprezentációk bevezetése, melyeknek feladata, hogy egy-egy ekvivalenciaosztályt egyértelmű jelöljenek. Abban az esetben, ha van olyan függvény, ami egy elemhez hozzárendeli annak reprezentációját, az ekvivalenciák kiértékelését visszavezethetjük a jobb és bal oldal reprezentációjának összehasonlítására. Az ilyen leképezést az ekvivalencia reprezentáló függvényének nevezzük, pontos definíciója a következő:

12. Definíció (Reprezentáló függvény). *Legyen H egy halmaz, és legyen a H halmazon értelmezett $\sim \subseteq H \times H$ ekvivalencia. A $rep : H \mapsto C$ függvényt* reprezentáló függvénynek*, C nevezzük*[2] *elemeit* reprezentációknak*, ha az alábbi állítás teljesül:*

$$\forall a, b \in H \Big(a \sim b \Leftrightarrow rep(a) = rep(b) \Big)$$

Egy H halmazon értelmezett tetszőleges f függvény definiálhat egy olyan $a \sim_f b \Leftrightarrow f(a) = f(b)$ ekvivalenciarelációt, amit f reprezentál. Könnyen belátható, hogy $\sim_f$ ekvivalencia, hiszen:

1. minden $a \in H$ esetén $f(a) = f(a)$
2. minden $a, b \in H$ esetén $f(a) = f(b) \Rightarrow f(b) = f(a)$
3. minden $a, b, c \in H$ esetén $f(a) = f(b) \wedge f(b) = f(c) \Rightarrow f(a) = f(c)$

Így a későbbiek folyamán, ha egyszerűbb, akkor reprezentáló függvények fogják definiálni az ekvivalenciákat.

Lássuk, mit nyertünk a reprezentációk bevezetésével[3]:

- A reprezentáló függvény segítségevel triviálissá vált egy elem ekvivalenciaosztályának megállapítása.
- Egy elem adott ekvivalenciaosztályba tartozása eldönthető az elemből számított és az osztályt képviselő reprezentáció összehasonlításával. Így egy halmaz elemeinek adott ekvivalencia szerinti csoportosítására a páronkénti összehasonlításnál jóval hatékonyabb algoritmus is megadható lesz.
- A formák címkéinek az ekvivalenciaosztálynál jóval kezelhetőbb halmaz elemeit kell tárolniuk. Egy jól megkonstruált reprezentáló függvény esetén a reprezentációk az ekvivalencia szempontjából hasznos tulajdonságokat tartalmazó struktúrák. Ezért tárolás tekintetében a címkék mérete kicsi lesz (amennyire csak lehet), számítási hatékonyságot nézve pedig egyértelmű, jól kezelhető adatokat tárolnak.

[2] A C elemei fölött értelmezett $=$ ekvivalenciának nem feltétlenül a logikai egyenlőséget kell jelölnie. A 3.2 ábrán a reprezentációk például gráfok, amelyeket izomorfia szerint hasonlítunk össze.

[3] A módszer előnyeit érdemes lehet összehasonlítani azzal a technikával, amikor ekvivalenciaosztályonként egy elemet, úgynevezett reprezentánst tárolunk. Egy elemnek az osztályba tartozását a reprezentánssal történő összehasonlítás határozza meg. Látható, hogy ez a módszer hatékonysága csak az utolsó pontban említett érveknél különbözik.

3.3. Forma, mint absztrakt gráf

Az ekvivalenciaosztályok kezelhetőbb ábrázolásának bemutatása után eljutottunk a reprezentációkkal paraméterezett forma gráfok matematikai definíciójához:

13. Definíció (Forma). *Jelölje Comp a kompatibilitási reprezentáció, $Instr^N$ az instrumentális csúcs-reprezentációk, Lab^E az eredeti élcímkék és $Instr^E$ instrumentális él-reprezentációk véges halmazát. Legyen továbbá $S \in \mathcal{G}(Comp \times Instr^N, Lab^E \times Instr^E)$. Az S gráf egy $n \in N_S$ csúcsa címkéjének Comp-beli részét megkaphatjuk a $comp_S(n)$ függvénnyel, az $Instr^N$-beli részét pedig az $instr^N_S(n)$ függvénnyel. Hasonló módon egy $e \in E_S$ él címkéjének Lab^E-beli része $olab_S(e)$, az $Instr^E$-beli része pedig $instr^E_S(e)$. Összefoglalva:*

$$\forall n \in N_S\Big(lab^N_S(n) = (comp_S(n), instr^N_S(n))\Big)$$

$$\forall e \in E_S\Big(lab^E_S(e) = (olab_S(e), instr^E_S(e))\Big)$$

Az S gráf forma, *ha az alábbi állítások igazak rá:*

- *„Nincsenek hasonló csúcsok", azaz olyanok amelyeknek ugyanaz a Comp-beli címkéjük:*
$$\forall n_1, n_2 \in N_S\Big(n_1 \neq n_2 \Rightarrow comp_S(n_1) \neq comp_S(n_2)\Big)$$
- *„Nincsenek párhuzamos élek", azaz olyanok, amelyeknek ugyanaz a forrásuk, céljuk és L_E-beli címkéjük:*
$$\begin{aligned}&\forall e_1, e_2 \in E_S\Big(e_1 \neq e_2 \Rightarrow\\ &\neg\Big(olab_S(e_1) = olab_S(e_2) \wedge src_S(e_1) = src_S(e_2) \wedge trg_S(e_1) = trg_S(e_2)\Big)\Big)\end{aligned}$$

A Comp-pal, $Instr^N$-rel, Lab^E-bal és $Instr^E$-rel címkézett formák halmazát jelöljük a következő módon: $\mathcal{S}(Comp, Instr^N, Lab^E, Instr^E)$. Ha a halmazok egyértelműek, használjuk egyszerűen $\mathcal{S}$-t.

Az absztrakt gráfokon történő állapottér felépítésének leglényegesebb feltétele, hogy mérete ne legyen végtelen. Ezt a feltételt a formák úgy tudják garantálni, hogy számosságuk véges. Így állapottérépítés esetén, még ha az összes lehetséges formát számba kell venni, a művelet befejeződik.

A formák számosságát az alábbi tétel mondja ki:

1. Állítás (Formák számossága). *Legyenek Comp, $Instr^N$, Lab^E és $Instr^E$ véges halmazok. Ekkor $|\mathcal{S}(Comp, Instr^N, Lab^E, Instr^E)|$ is véges. (A bizonyítás az alábbi cikkekben tallható: [10, 11].)*

3.4. Formázás, az absztrakció

Egy leképezés esetén az értékkészlet egy adott elemébe képzett elemek egyszerű meghatározására vezessünk be egy egyszerű jelölést:

14. Definíció (Általánositott inverz függvény). *Egy* $f : A \mapsto B$ *függvény* általánosított inverzén *azt az* $\operatorname{inv} f : B \mapsto 2^A$ *függvényt értjük, amire:*

$$\forall b \in B \Big(\operatorname{inv} f(b) = \{a | f(a) = b\} \Big)$$

Többváltozós $f : A, A_1, \ldots, A_n \mapsto B$ *függvény esetén* $\operatorname{inv} f : B, A_1, \ldots, A_n \mapsto 2^A$*, ahol:*

$$\forall b \in B, \forall a_1 \in A_1 \ldots a_n \in A_n \Big(\operatorname{inv} f(b, a_1, \ldots, a_n) = \{a | f(a, a_1, \ldots, a_n) = b\} \Big)$$

A gráfok formákká történő leképezését formázásnak nevezzük. A formázást végző függvény paraméterezésével megadhatóak a kompatibilitási és az instrumentális ekvivalenciák, eredménye pedig a paramétereknek megfelelő címkekészletű forma. Ez a hozzárendelés rendkívül fontos eleme gráf absztrakciónak, mivel ez a rész határozza meg az összeköttetést az absztrakt és a konkrét gráfok között, a formákat ez a kapcsolat ruházza fel jelentéssel. Ebből következően formák tartalmi jelentéséről csak a formázás kontextusának rögzítése esetén van értelme beszélni.

A függvényt akkor használjuk amikor a kiinduló G_0 gráfból szeretnénk egy olyan, a paraméterezésnek megfelelő címkekészletű S_0 formát kapni, amely magába foglalja magát a G_0-át is.

15. Definíció (Formázás). *Legyen* G *egy* Lab^E *élcímkékkel címkézett gráf. Legyenek továbbá:*

$\overset{Comp}{\sim} \subseteq \mathcal{N} \times \mathcal{N}$	–	$f_{Comp} : \mathcal{N} \mapsto Comp$ *által reprezentált csúcsok fölött értelmezett kompatibilitási ekvivalencia*
$\overset{Instr^N}{\sim} \subseteq 2^{\mathcal{N}} \times 2^{\mathcal{N}}$	–	$f_{Instr^N} : 2^{\mathcal{N}} \mapsto Instr^N$ *által reprezentált csúcshalmazok fölött értelmezett instrumentális csúcs-ekvivalencia*
$\overset{Instr^E}{\sim} \subseteq (2^{\mathcal{N}} \times 2^{\mathcal{E}} \times 2^{\mathcal{N}})^2$	–	$f_{Instr^E} : 2^{\mathcal{N}} \times 2^{\mathcal{E}} \times 2^{\mathcal{N}} \mapsto Instr^E$ *által reprezentált instrumentális él-ekvivalencia*

Ekkor $shaping(G, \overset{Comp}{\sim}, \overset{Instr^N}{\sim}, \overset{Instr^E}{\sim}) = S$*, ahol* $S \in \mathcal{S}(Comp, Instr^N, Lab^E_G, Instr^E)$*, és* S*-re az alábbi állítások igazak:*

1. *Létezik* G *és* S *között* m *szürjektív morfizmus. Ezt az* m*-et a formázás morfizmusának fogjuk nevezni.*
2. $\forall n \in N_G \Big(comp_S(m(n)) = f_{Comp}(n) \Big)$
3. $\forall n \in N_S \Big(instr^N_S(n) = f_{Instr^N}(\operatorname{inv} m^N(n)) \Big)$
4. $\forall e \in E_G \Big(olab_S(m(e)) = lab^E_G(e) \Big)$
5. $\forall e \in E_S \Big(instr^E_S(e) = f_{Instr^E}(\ (\operatorname{inv} m^N(src_S(e)), \operatorname{inv} m^E(e), \operatorname{inv} m^N(trg_S(e)))\ ) \Big)$

Ha az ekvivalenciák, azok reprezentáló függvényei és az élcímkék halmaza adottak, az egyszerűbb $shaping(G)$ *jelölést használjuk.*

Az 1. pontban meghatározunk a gráf és a forma között egy morfizmust, amellyel minden gráfelemet egy formabeli gráfelemhez csoportosítunk. A morfizmus szürjektív természete miatt olyan csoport nem jelenik meg, amely A 2. pont kiköti, hogy a formabeli csúcsba csak egy ekvivalenciaosztályba tartozó elemek kerülhetnek. A formák definíciójánál megadott „nincsenek hasonló csúcsok" feltétel miatt azonban több csúcsba sem képződhetnek a konkrét csúcsok. Tehát függvényünk az elvárásainknak megfelelően egy formabeli csúcsba sorolja a $\overset{Comp}{\sim}$-szerint megegyező konkrét csúcsokat. A 4. pont szerint a formázás morfizmusa a gráf éleit olyan formabeli élekhez rendeli, amelyek *olab* értékük megegyezik az él címkéjével. A morfizmus tulajdonságai miatt az élek képei csak olyan formabeli csúcsok között futhatnak, amelyekbe a gráfcsúcsok lettek rendelve. Ezen kívül párhuzamos élek sem lehetnek a formában.

A formázási morfizmus meghatározza, hogy a konkrét gráf melyik csúcs és él melyik formabeli csúcsba és élbe képződik. Ezen morfizmus inverze mentén megkapjuk az egy formacsúcshoz tartozó csúcshalmazokat, valamint az egy formabeli élhez, annak forrásához és céljához tartozó halmazokból álló hármast. Az inverz morfizmus alapján számított értékeket az instrumentális ekvivalenciák osztályozzák a 3-as és 5-ös pont szerint.

Egy adott forma által reprezentált gráfokat a forma konkretizáltjainak nevezzük. A konkretizáltakat megadó függvény a következő:

16. Definíció (Konkretizáció). *Legyenek az alábbiak ekvivalenciarelációk:* $\overset{Comp}{\sim} \subseteq \mathcal{N} \times \mathcal{N}$, $\overset{Instr^N}{\sim} \subseteq 2^{\mathcal{N}} \times 2^{\mathcal{N}}$ *és* $\overset{Instr^E}{\sim} \subseteq 2^{\mathcal{E}} \times 2^{\mathcal{E}}$. *Ekkor egy S forma* konkretizációin *az alábbi címkézett gráfokból álló halmazt értjük:*

$$concr(S, \overset{Comp}{\sim}, \overset{Instr^N}{\sim}, \overset{Instr^E}{\sim}) = \mathrm{inv}\, shape(S, \overset{Comp}{\sim}, \overset{Instr^N}{\sim}, \overset{Instr^E}{\sim})$$

A formázáshoz hasonlóan, ha a többi paraméter egyértelmű, az egyszerűbb $concr(S)$ jelölést követjük.

3.5. Összefoglalás

Mivel a paraméterként megadott ekvivalenciareláció szerint csoportosítottuk a gráfelemeket, így jól kezelhető és tervezhető, hogy mely elemeket kívánunk összevonni. A következő két fejezetből kiderül, hogy a szakirodalomban használt eljárásokhoz is alkothatók olyan ekvivalenciák, amelyekkel az általánosított definíciót felparaméterezve megkapjuk magukat az ott leírt absztrakciókat.

4. fejezet

Szomszédsági formák

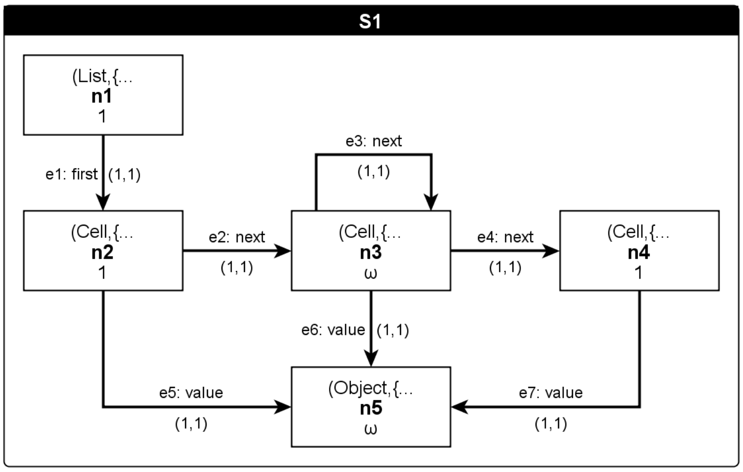
Ebben a fejezetben a formák egy családja, az úgynevezett szomszédsági formák kerülnek tárgyalásra. Gráfok ilyen formájú absztrakciói a [10, 11, 12] és az ezeket összegző [6] cikkekben találhatóak.

4.1. Szomszédság bemutatása

A 3. fejezetben a formák bemutatása során olyan példát használtunk, amelyben a kompatibilitási ekvivalencia szerint akkor egyezett meg két csúcs, ha a környezetük hasonlított. Környezet alatt olyan részgráfra gondoljunk, amely elemei a vizsgált csúcsból indulva meghatározott tulajdonságú gráfbejárásokkal elérhető. A fejezetben ehhez hasonló, jóval precízebben meghatározott, paraméterezhető ekvivalenciák, és az általuk készített formák kerülnek bemutatásra.

Ha megfigyeljük a 4.1 ábrát, példát láthatunk egy szomszédsági formára, ami alakra megegyezik az előző fejezetben bemutatottal. A könnyebb hivatkozás végett minden gráfelemet külön azonosító jelöl (például: `n3`, `e3`). A csúcsok azonosítója felett látható a kompatibilitási, alatta az instrumentális ekvivalenciához tartozó reprezentáció. Ha valamelyik méretéből fakadóan nem fér ki, . . . -tal félbeszakítódik. Élek esetében kettőspont után az eredeti élcímke látható, tőle lent/balra az instrumentális ekvivalencia reprezentációja. A csúcsok kompatibilitás szerinti reprezentációja a csúcsnak a konkrét gráfokban lévő környezetét írja le. Egy ilyen reprezentáció kiszámítását és pontos felépítését a szomszédsági ekvivalencia később tárgyalt definíciója tartalamazza.

Amit még észrevehetünk, hogy instrumentális ekvivalenciák reprezentációi számokból és az ω jelből állnak, ezeket együttesen multiplicitásnak fogjuk nevezni. Az ω egy olyan mennyiséget fog jelölni, amit már nem szeretnénk pontosan számontartani. Multiplicitásokkal a csúcs és élhalmazok különböző mennyiségi jellemzőit fogjuk jelölni, amelyekre példákat és a precíz meghatározásukat a következő szakasz fog adni.



4.1. ábra. *Példa szomszédsági formára. A csúcsok 1 távolságban és 1 párhuzamossággal lettek összehasonlítva, a csúcsok és élek maximális multiplicitása 1.*

4.2. Multiplicitások

4.2.1. Multiplicitás reprezentáció

Mennyiségi jellemzők kerekítése céljából bevezetésre kerül a multiplicitás reprezentáció. Ennek feladata, hogy a pozitív egész számokat véges számú kategóriákba sorolja. Esetünkben ezek a kategóriák az „egy, kettő, sok" elvén alapulnak, amellyel egy adott maximumig pontosan tároljuk az értéket, azon felül pedig egyszerűen annyit állítunk, hogy sok. Ezzel legtöbb esetben a számunkra fontos értékeket pontosan megőrizhetjük.

A multiplicitást felkészítjük, hogy képes legyen annak tárolására is, ha egy elem mennyiségi jellemzője különböző okokból kiértékelhetetlen, vagy nem értelmezett[1]. Ilyen tulajdonság jelzésére használatos az $undef$ érték.

A multiplicitás, és az egész számokból multiplicitásba képző függvény definíciója a következő:

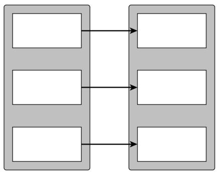
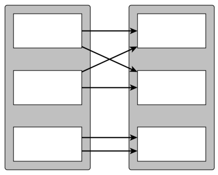
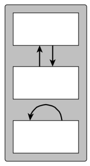
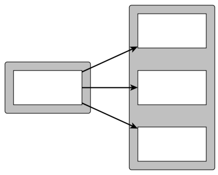
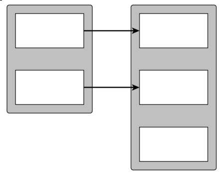
17. Definíció (Multiplicitás reprezentáció). *Egy adott $m \in \mathbb{N}^+$ esetén M_m legyen az alábbi halmaz:*

$$M_m = \{1, \ldots, m, \omega, undef\}$$

Legyen továbbá $mult_m : \mathbb{N} \mapsto M_m$ az alábbi függvény:

$$mult_m(i) = \begin{cases} undef & ha\ i = 0 \\ i & ha\ 0 < i \leq m \\ \omega & m < i \end{cases}$$

[1] Abban az esetben, ha csak ebben a fejezetben használt ekvivalenciákkal paraméterezzük a formáinkat, ez nem fog előfordulni. Viszont ha a módszereket keverni kívánjuk, értelmes jelölésnek bizonyul.

A	B	C	D	E
				
$(1,1)$	$(2,2)$	$(1,1)$	$(1,\omega)$	$(undef,1)$

4.1. táblázat. *Élmultiplicitás példák.*

4.2.2. Egyszerű csúcsmultiplicitás

A szomszédsági formák csúcsain értelmezett instrumentális ekvivalencia azt hivatott jelezni, hogy a konkrét gráfokban a kompatibilitás reprezentációjának hány csúcs felel meg. Más szóval, ha a egy formázás morfizmusa m, akkor a forma n csúcsának címkéjébe az $|\mathrm{inv}\, m(n)|$ értéket szeretnénk valahogy tárolni.

Ezt a számosságot – a véges ekvivalenciaosztály feltételének betartása végett – átalakítjuk multiplicitássá. Ennek következtében két csúcshalmazt akkor tekintünk egyenlőnek, ha számosságuk egyenlő, vagy meghaladnak egy adott határt. Ezt az ekvivalenciarelációt, és az ezt reprezentáló függvényt az alábbi definíció mondja ki formálisan:

18. Definíció (Csúcsmultiplicitás). *Legyen* $\overset{Nm}{\underset{m}{\sim}} \subseteq 2^{\mathcal{N}} \times 2^{\mathcal{N}}$*, ahol* $m \in \mathbb{N}^+$ *a következő ekvivalenciareláció:*

$$\forall U, V \in 2^{\mathcal{N}} \Big(U \overset{Nm}{\underset{m}{\sim}} V \Leftrightarrow |U| = |V| \ \vee \ (m < |U| \wedge m < |V|) \Big)$$

Ekkor a $\overset{Nm}{\underset{m}{\sim}}$ *a* $mult_m^N : 2^{\mathcal{N}} \mapsto M_m$ *függvény által reprezentált ekvivalenciareláció:*

$$mult_m^N(U) = mult_m(|U|)$$

4.2.3. Kimenő és bejövő élmultiplicitás

Ahogy a bevezetőben láthattuk, a szomszédsági formák éleinek instrumentális reprezentációi multiplicitáspárok. Bevezetésként S formában lévő e élen egy $(i, o) \in \mathbb{N}^+ \times \mathbb{N}^+$ párossal szeretnénk jelölni azt, hogy az S konkretizáltjaiban az e forrásának megfelelő csúcsok mindegyikéből i darab e-nek megfelelő él vezet, és hasonló módon e céljainak megfelelő csúcsok mindegyikébe o darab e-nek megfelelő él megy.

Az így kapott (i, o) párokkal két problémánk van: egyrészt a számosságuk végtelen lehet, másrészt nem minden esetben ugyanannyi az élek száma. Ezen problémák elhárítása végett áttérünk a $(m_i, m_o) \in M_m \times M_m$ párosra, ahol egy adott m mellett $m_i = mult_m(i)$ és $m_o = mult_m(o)$. Mivel a multiplicitás párosok számossága véges, így az egyik problémát megoldottuk. Abban az esetben pedig, ha nem minden élszámosság egyezik meg egymással, az $undef \in M_m$ jelet használjuk.

Példa A definíció pontos megadása előtt nézzünk pár példát. A példák során egy $(S, E, T) \in 2^{\mathcal{N}} \times 2^{\mathcal{E}} \times 2^{\mathcal{N}}$ hármashoz az $M_2 \times M_2$ értékkészletből rendelünk élmultiplicitásokat. A példák az 4.1 táblázatban vannak ábrázolva, ahol a csúcshalmazokat bekeretezettség jelöli, egy (S, E, T) esetén, ha $S \neq T$, akkor S a baloldali, T a jobboldali csoportot jeleníti meg. Az ábrán csak az E-beli élek vannak ábrázolva. A példák fölött a példa betűjele, alatta a megoldásaként előálló élmultiplicitás látható.

A. Az első példában egy egyértelmű esettel állunk szemben. Mivel minden S-beli csúcsból egy E-beli él vezet egy T-beli csúcsba, a multiplicitás $(1, 1)$.

B. A példában látható, az élmultiplicitás nincs tekintettel arra, hogy az élek milyen módon vezetnek a csúcshalmazok között. Sorrendhelyesen a jobboldali és a baloldali csúcsok fokszáma alapján számítódott ki a $(2, 2)$ eredmény.

C. Ebben a példában megfigyelhetünk egy $S = T$ esetén előforduló hurokélt. Az előző pontnál megállapított fokszám szerinti számítást finomítanunk kell „bejövő” és „kimenő” fokszámok megállapítására.

D. A példából számított élmultiplicitás $(1, 3)$ lenne, ha nem korlátoztuk volna le a multiplicitások tartományát 2-re. Ezen megállapítás szerint az élmultiplicitás: $(1, \omega)$.

E. Az utolsó példában egy olyan esetet vizsgálunk, amikor a bejövő fokszámok nem egyeznek meg. Ekkor tehát a multiplicitás: $(undef, 1)$.

A példák bemutatása után lássuk a pontos definíciót:

19. Definíció (Élmultiplicitás). *Adott $m \in \mathbb{N}^+$ mellett legyen $\underset{m}{\overset{Em}{\sim}} \subseteq (2^{\mathcal{N}} \times 2^{\mathcal{E}} \times 2^{\mathcal{N}})^2$ az alábbi $mult_{i,o}^{E} : 2^{\mathcal{E}} \mapsto M_i \times M_o$ függvény által reprezentált ekvivalenciareláció:*

$$mult_{i,o}^{E}((S, E, T)) = (m_i, m_o), \text{ ahol:}$$

$$m_i = \begin{cases} mult_i(x) & \text{ha minden } n \in S\text{-re teljesül, hogy } |\mathrm{inv}\, src(n) \cap E| = x \\ undef & \text{különben} \end{cases}$$

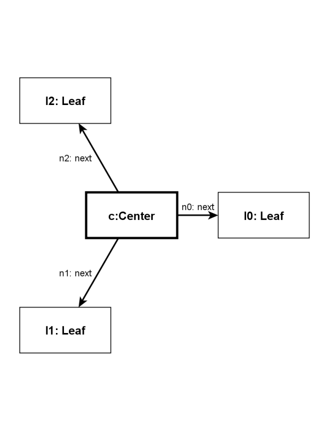
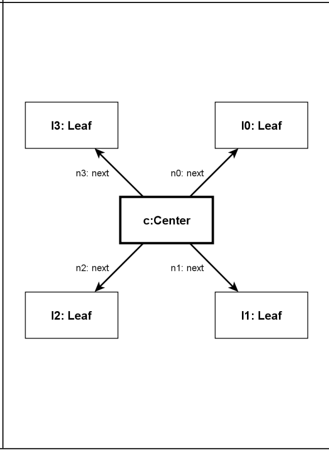
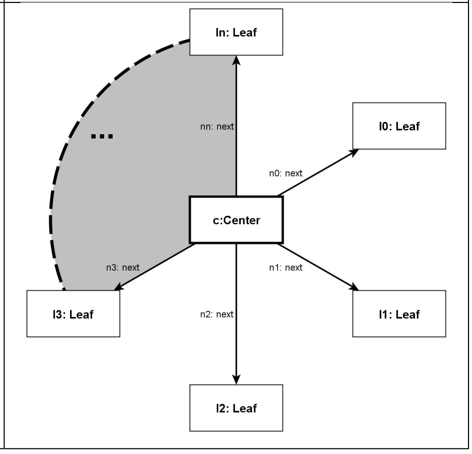
$$m_o = \begin{cases} mult_o(x) & \text{ha minden } n \in T\text{-re teljesül, hogy } |\mathrm{inv}\, trg(n) \cap E| = x \\ undef & \text{különben} \end{cases}$$

4.3. A szomszédsági ekvivalencia

Ebben a szakaszban a szomszédsági formák kompatibilitási ekvivalenciája, a szomszédsági ekvivalencia kerül tárgyalásra. Először az ekvivalencia, majd az azt reprezentáló függvény lesz bemutatva.

4.3.1. Ekvivalencia

Ez az ekvivalencia a gráfok csúcsait környezetük alapján hasonlítja össze. Nem szabad elfelejtenünk, hogy adott címkekészlet mellett véges számú ekvivalenciaosztály létezhet csak. Ezért először vezessünk be egy *távolság* paramétert, amivel meghatározzuk a környezetként vizsgált részgráfot: két csúcs akkor legyen izomorf, ha a két csúcstól a paraméterként megadott távolságon belüli részgráfok izomorfak.

<table>
<tr><th>3 ágú csillag</th><th>4 ágú csillag</th><th>n ágú csillag</th></tr>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
</table>

4.2. táblázat. *Csillag alakú gráfok. Egy* `Center` *címkéjű csúcsból ág darab* `next` *címkéjű él vezet a* `Leaf` *címkéjű csúcsokba.*

Ha megvizsgáljuk a 4.2 táblázatot, láthatjuk, hogy az előbbi definíció véges számú ekvivivalenciaosztály kikötése mellett elégtelennek bizonyul. Az ábrákon látható n ágú csillag gráfok `Center` címkéjű csúcsai 1 távolság esetén ugyanis n különböző osztályba esnek. Ésszerű lenne, ha egy adott határ feletti ágszámú csillagokat már hasonlónak tekintse ekvivalenciánk. Ezt a határt nevezzük *párhuzamosságnak.*

Fogalmazzuk meg precízebben, hogy melyek azok a tulajdonságok, amelyek alapján két környezet távolság és párhuzamosság szempontjából összevonható. Két csúcsot 0 távolság esetén akkor tekintsünk egyenlőnek, ha ugyan az a címkéjük. Ha 1 távolsággal és p párhuzamossággal vizsgáljuk csúcsainkat, akkor hasonlítsuk össze ezek címkéit is először. Megegyezés esetén hasonlítsuk össze közvetlen szomszédságukat, vagyis azt, hogy milyen címkéjű kimenő/bejövő élekkel milyen címkéjű csúcsokkal van összekötve. Ha egy l_E címkéjű éllel ugyanolyan irányítottsággal többször is elérhető l_N címkéjű csúcs, az előfordulások számát kerekítsük $mult_p$-vel.

Vegyük észre, hogy 1 távolság esetén a csúcsokat először 0 távolsággal (azaz címkevizsgálattal) hasonlítottuk össze, és a szomszédos csúcsok között is 0 távolságú szomszédság szerint tettünk különbséget. Így ha r távolság esetén ezeket a vizsgálatokat $r-1$ távolsággal tesszük meg, az ekvivalenciavizsgálat a csúcsoktól r élt tartalmazó sétákkal elérhető részgráfokat fog összehasonlítani, úgy, hogy p darab hasonló szomszéd még megkülönböztethető marad.

Összefoglalva az előbbieket a következő definíciót kapjuk (ne feledjük, hogy $mult_m(|\emptyset|) = undef$):

20. Definíció (Szomszédsági ekvivalencia). *Egy adott $r \in \mathbb{N}$ és $p \in \mathbb{N}^+$ esetén az r távolságú p párhuzamosságú* szomszédsági ekvivalencián *az alábbi $\overset{Neigh}{\underset{r,p}{\sim}} \subseteq \mathcal{N} \times \mathcal{N}$ ekvivalenciát értjük:*

- *$r = 0$ esetén: $u \overset{Neigh}{\underset{0,p}{\sim}} v$ ha ugyanaz a csúcsok címkéje: $lab^N(u) = lab^N(v)$*
- *$r > 0$ esetén: $u \overset{Neigh}{\underset{r,p}{\sim}} v$ ha az alábbi állítások teljesülnek:*

- *$r-1$ sugárban is ekvivalensek:* $u \overset{Neigh}{\underset{r-1,p}{\sim}} v$
- *u-ból és v-ből multiplicitás szerint kerekítve ugyanannyi l címkéjű él vezet a $C \in \mathcal{N}/\overset{Neigh}{\underset{r-1,p}{\sim}}$ osztályú csúcsokba:*
  $\forall C \in \mathcal{N}/\overset{Neigh}{\underset{r-1,p}{\sim}}, \forall l \in Lab^E\Big($
  $mult_p(|\{e|\ src(e) = u \ \wedge \ lab^E(e) = l \ \wedge \ trg(e) \in C\}|) =$
  $mult_p(|\{f|\ src(f) = v \ \wedge \ lab^E(f) = l \ \wedge \ trg(f) \in C\}|)\Big)$
- *u-ba és v-be multiplicitás szerint kerekítve ugyanannyi l címkéjű él vezet a $C \in \mathcal{N}/\overset{Neigh}{\underset{r-1,p}{\sim}}$ osztályú csúcsokból:*
  $\forall C \in \mathcal{N}/\overset{Neigh}{\underset{r-1,p}{\sim}}, \forall l \in Lab^E\Big($
  $mult_p(|\{e|\ trg(e) = u \ \wedge \ lab^E(e) = l \ \wedge \ src(e) \in C\}|) =$
  $mult_p(|\{f|\ trg(f) = v \ \wedge \ lab^E(f) = l \ \wedge \ src(f) \in C\}|)\Big)$

4.3.2. Reprezentáció

A szomszédsági ekvivalencia megismerése után igény támad az azt reprezentáló függvény megadására. Először határozzuk meg ennek a függvénynek értékkészletéül szolgáló szomszédsági reprezentáció halmazt.

A halmaz elemei olyan hármasok ($NRep_{r,p}$), amelyek egyfelől tartalmaznak egy egyel kisebb távolságú szomszédsági reprezentációt, másrészt bejegyzéseket a kimenő és a bejövő éleken keresztül elérhető szomszédokról. A bejegyzéseknek ($NRepEntry_{r-1,p}$) nevezett elemek olyan hármasok, amelyek tartalmaznak egy egyel kisebb távolságú reprezentációt, egy élcímkét, és egy multiplicitást.

A halmaz pontos definíciója a következő:

21. Definíció (Szomszédsági reprezentáció). *Egy adott $r \in \mathbb{N}$ és $p \in \mathbb{N}^+$ esetén az r távolságú p párhuzamosságú* szomszédsági reprezentáció *alatt az alábbi $NRep_{r,p}$ halmaz elemeit értjük:*

- *$r = 0$ esetén:* $NRep_{0,p} = Lab^N$
- *$r > 0$ esetén:* $NRep_{r,p} = NRep_{r-1,p} \times 2^{NRepEntry_{r-1,p}} \times 2^{NRepEntry_{r-1,p}}$*, ahol:*
  $NRepEntry_{r-1,p} = Lab^E \times NRep_{r-1,p} \times M_p$

A reprezentációkba képző függvény definíciója az alábbi módon fogalmazható meg:

22. Definíció (Reprezentáló függvény). *Legyenek $r \in \mathbb{N}$ és $p \in \mathbb{N}^+$. Ekkor a $\overset{Neigh}{\underset{r,p}{\sim}}$ ekvivalenciát reprezentálja az alábbi $nrep_{r,p} : \mathcal{N} \mapsto NRep_{r,m}$ függvény:*

$$nrep_{r,p}(n) \quad = \quad \begin{cases} lab^E(n) & ha\ r = 0 \\ (nrep_{r-1,p}(n), IN, OUT) & különben \end{cases}$$

ahol:

- *IN elemei olyan $(l, c, m) \in NRepEntry_{r-1,p}$ hármasok, amelyekre teljesül, hogy az olyan $(e, u) \in I \subseteq \mathcal{E} \times \mathcal{N}$ párok, amelyeknél $trg(e) = n$, $src(e) = u$, $lab^E(e) = l$ és $nrep_{r-1,p}(u) = c$ esetén $I \neq \emptyset$ és $m = mult_p(|I|)$.*

- *OUT elemei az előző ponthoz hasonlóan olyan* $(l, c, m) \in NRepEntry_{r-1,p}$ *hármasok, amelyekre teljesül, hogy az olyan* $(e, u) \in O \subseteq \mathcal{E} \times \mathcal{N}$ *párok, amelyeknél* $trg(e) = u$, $src(e) = n$, $lab^E(e) = l$ *és* $nrep_{r-1,p}(u) = c$ *esetén* $O \neq \emptyset$ *és* $m = mult_p(|O|)$.

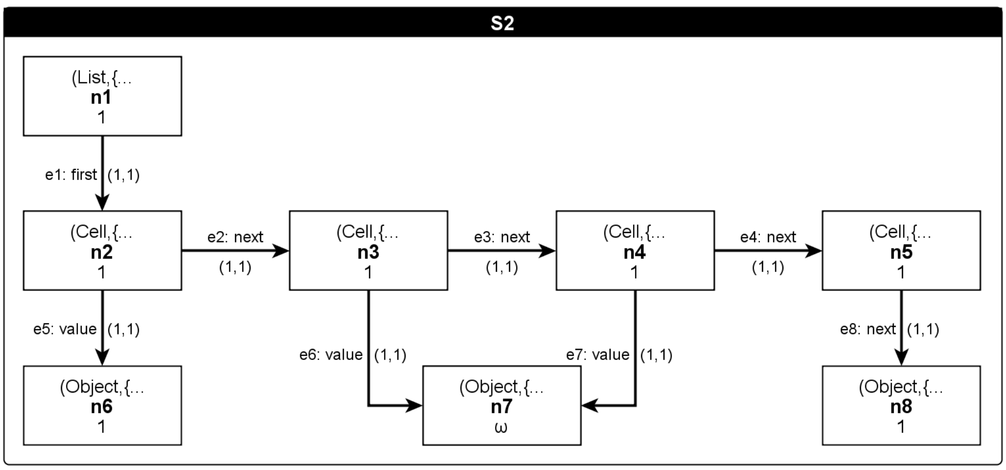
Példa Az 4.2 ábrán látható egy olyan forma, amely a 15. oldalon található 3.1 ábrán bemutatott címkézett gráfból 2 sugarú szomszédsági ekvivalenciával lett képezve. Az `S2` forma `n3` csúcsához tartozó kompatibilitási reprezentációt fogjuk kiszámolni. A számítás részeredményeit a 4.3 táblázat tartalmazza.

Először is számoljunk ki néhány szükséges részeredményt. A formázás morfizmusa szerint az `n3` csúcs egyedül a gráf `c2` csúcsából lett képezve. Ennek a `c2` csúcsnak szomszédai a `c1`, a `c3` és az `o2` csúcsok, tehát ezek 1 távolságú reprezentációira szükség lesz.

Ezek a reprezentációk a táblázat első 4 sorában találhatók meg. A reprezentációk hármasának első részében a csúcs típusa van megadva, mivel ez az $1 - 1 = 0$ távolságú reprezentációja a csúcsoknak. Ezek után a bejövő éleket leíró halmaz következik, amelyben a bejövő élnek és annak forrásának címkéi, valamint előfordulásuk száma szerepelnek. Vegyük észre, hogy elő nem forduló párost nem szerepeltetünk. A hármasok utolsó részei a másodikhoz hasonló módon képződnek, csak kimenő éleket figyelünk.

Az 4.1 ábrán látható `S1` forma egy sugarú szomszédsági ekvivalencia szerint lett képezve, így csúcsainak kompatibilitási reprezentációi közül néhányat kiszámoltunk a feladat részmegoldásaként. Ezeket az eredményeket is feljegyzi táblázatunk.

Az utolsó sorokban látható $nrep_{2,1}$(`c2`) értéke. Érdemes észrevenni, ez $nrep_{1,1}$(`c2`)-től csak annyiban tér el, hogy 1 távolságú reprezentációk szerepelnek a címkék (azaz 0 távolságú reprezentációk) helyett.



4.2. ábra. *Második példa szomszédsági formára. A csúcsok 2 távolságban és 1 párhuzamossággal vannak összehasonlítva, a csúcsok és élek maximális multiplicitása pedig 1.*

4.4. Összefoglalás

A szomszédság alapján történő csoportosítás sajátossága, hogy a csúcsok automatikusan tagolódnak sok szerepkörre. Ennek előnye hogy az analízis során használt ekvivalenciaosz-

1 távolságú	$nrep_{1,1}(\mathtt{c1})$	$=$	$\left\{\begin{array}{l}(\mathtt{Cell},\\ \{(\mathtt{List},\mathtt{first},1)\},\\ \{(\mathtt{Cell},\mathtt{next},1),(\mathtt{Object},\mathtt{value},1)\})\end{array}\right.$
	$nrep_{1,1}(\mathtt{c2})$	$=$	$\left\{\begin{array}{l}(\mathtt{Cell},\\ \{(\mathtt{Cell},\mathtt{next},1)\},\\ \{(\mathtt{Cell},\mathtt{next},1),(\mathtt{Object},\mathtt{value},1)\})\end{array}\right.$
	$nrep_{1,1}(\mathtt{c3})$	$=$	$nrep_{1,1}(\mathtt{c2})$
	$nrep_{1,1}(\mathtt{o1})$	$=$	$\left\{\begin{array}{l}(\mathtt{Object},\\ \{(\mathtt{Cell},\mathtt{value},1)\},\\ \{\})\end{array}\right.$
	$comp_{\mathtt{S1}}(\mathtt{n2})$	$=$	$nrep_{1,1}(\mathtt{c1})$
	$comp_{\mathtt{S1}}(\mathtt{n3})$	$=$	$nrep_{1,1}(\mathtt{c2}) = nrep_{1,1}(\mathtt{c3})$
	$comp_{\mathtt{S1}}(\mathtt{n5})$	$=$	$nrep_{1,1}(\mathtt{o1})$ $(= nrep_{1,1}(\mathtt{o2}) = nrep_{1,1}(\mathtt{o3}) = nrep_{1,1}(\mathtt{o4}))$
2 távolságú	$nrep_{2,1}(\mathtt{c2})$	$=$	$\left\{\begin{array}{l}(nrep_{1,1}(\mathtt{c2}),\\ \{(nrep_{1,1}(\mathtt{c1}),\mathtt{next},1)\},\\ \{(nrep_{1,1}(\mathtt{c3}),\mathtt{next},1),(nrep_{1,1}(\mathtt{o2}),\mathtt{value},1)\})\end{array}\right.$
	$comp_{\mathtt{S2}}(\mathtt{n3})$	$=$	$nrep_{2,1}(\mathtt{c2})$

4.3. táblázat. *Gráfok csúcsaihoz tartozó szomszédsági reprezentációk és formák csúcsaihoz tartozó kompatibilitási reprezentációk.*

tályok könnyen megadhatóak, hátránya, hogy multiplicitások, távolság és párhuzamosság szintjének növelése általánosan javítják formáink kifejezőerejének pontosságát, így feladatra szabásra nincs sok lehetőség.

A szomszédsági ekvivalencia másik jellegzetessége, hogy az ekvivalenciaosztályok bonyolult, nagy méretű struktúrák lehetnek. A gazdag ekvivalenciaosztályok miatt szerény paraméterezés mellett is gondot jelenthet a kompatibilitási ekvivalenciák reprezentációinak külön-külön történő tárolása. Érdemes ezért felismerni, hogy szomszédsági reprezentációinknak sok közös részük van. Ha ezeket sikerül úgy ábrázolni, hogy egymás részeire hivatkoznak, a címkék tárolásának erőforrásigénye nagyban lecsökken.

5. fejezet

Predikátumformák

Ebben a fejezetben megadunk egy módszert, amivel meglévő ekvivalenciáinkat egyszerre használhatjuk. Ezt követően bemutatásra kerülnek a háromértékű logikán alapuló predikátumformák, melyek absztrakciója a [9] cikkben leírtakon alapuló ekvivalenciákat használja. Legvégül definiálunk még két olyan instrumentális ekvivalenciát, amelyek jól használhatóak a predikátumformákban.

5.1. Ekvivalencarelációk kombinálása

Ha van két ekvivalenciánk, és egy halmaz elemeit mindkettő szerint egyszerre hasonlítjuk össze, egy új ekvivalenciát kapunk. Az új relációnk tipikusan finomabb felbontásba osztályozza az elemeket, és ekvivalenciaosztályaink is több jelentést hordozhatnak elemeikről. Használhatósága miatt definiáljuk az egyszerre több ekvivalenciát kombináló műveletet:

23. Definíció (Ekvivalencia szorzat). *Legyen H egy halmaz, és legyenek $\overset{E}{\sim}, \overset{F}{\sim} \subseteq H \times H$ ekvivalenciarelációk. Ekkor $(\overset{E}{\sim} \times \overset{F}{\sim}) \subseteq H \times H$ is egy ekvivalenciareláció, amely a következőképen van definiálva:*

$$\forall a, b \in H\Big(a \overset{E}{\sim} \times \overset{F}{\sim} b \Leftrightarrow a \overset{E}{\sim} b \land a \overset{F}{\sim} b\Big)$$

Továbbá ennek nagyoperátoros megfelelője a következő:

$$\prod_{i=1}^{n} \overset{E_i}{\sim} = \overset{E_1}{\sim} \times \ldots \times \overset{E_n}{\sim}$$

Az ekvivalenciákat reprezentáló függvényeikkel értékeljük ki, úgyhogy fontos kitérnünk az új műveletünkkel kapcsolatos viselkedésükre. Szerencsére nem ütközünk az új függvény definiálásánál komoly problémába, az új reprezentáció az operandusok reprezentációiból álló struktúra lesz.

2. Állítás (Ekvivalencia szorzatot reprezentáló függvény). *Legyen H egy halmaz, legyenek $\overset{E_1}{\sim}, \ldots \overset{E_n}{\sim} \subseteq H \times H$ olyan ekvivalenciarelációk, amelyeket az $f_1 : H \mapsto C_1, \ldots, f_n : H \mapsto C_n$ függvények reprezentálnak. Ha $F = \prod_{i=1}^{n} \overset{E_i}{\sim}$, akkor azt reprezentálja az alábbi*

$f_F : H \mapsto C_1 \times \ldots \times C_n$ *függvény:*

$$f_F(h) = \Big(f_1(h), \ldots, f_n(h)\Big)$$

A végeredmény reprezentációjából következik egy fontos megfigyelés. Ha véges számú osztállyal rendelkező ekvivalenciákat szorzunk össze, a végeredménynek is véges osztálya lesz.

5.2. Csúcsokon értelmezett predikátum

Hasznosnak tűnik egy olyan csúcsokon értelmezett predikátummal paraméterezhető ekvivalencia megfogalmazása, amely két osztályba sorolja az elemeket: azokra, amelyekre teljesül a predikátum, és azokra amelyekre nem. A paraméterként megadható predikátumok adott elemi predikátumszimbólumokból és a megszokott logikai összekötőkből ($\neg$, $\wedge$, $\vee$, $\Rightarrow$), logikai egyenlőségvizsgálatból ($=$) és kvantorokból ($\exists$, $\forall$) álló elsőrendű formulák közül kerülnek ki.

Az elemi predikátumszimbólumok készlete a kezelhetőség végett kötött. Példáinkban az alábbiak fognak szerepelni:

$Node(n, l)$	$\Leftrightarrow$	Az n egy l-lel címkézett csúcs.
$Edge(u, v, l)$	$\Leftrightarrow$	Az u csúcsból vezet v-be l címkéjű él.
$Transitive(u, v, l)$	$\Leftrightarrow$	Létezik u-ból v-be vezető l címkéjű éleken irányhelyesen haladó séta.

A lista természetesen bővíthető, de a formákban tudnunk kell kezelni minden predikátum igazságtartalmának jelentését. Vezessük be a csúcspredikátum fogalmát:

24. Definíció (Csúcspredikátum). *Egy P formula csúcspredikátum, ha P-ben egyetlen behelyettesítetlen változó n egy csúcs ($P(n) \Rightarrow n \in \mathcal{N}$).*

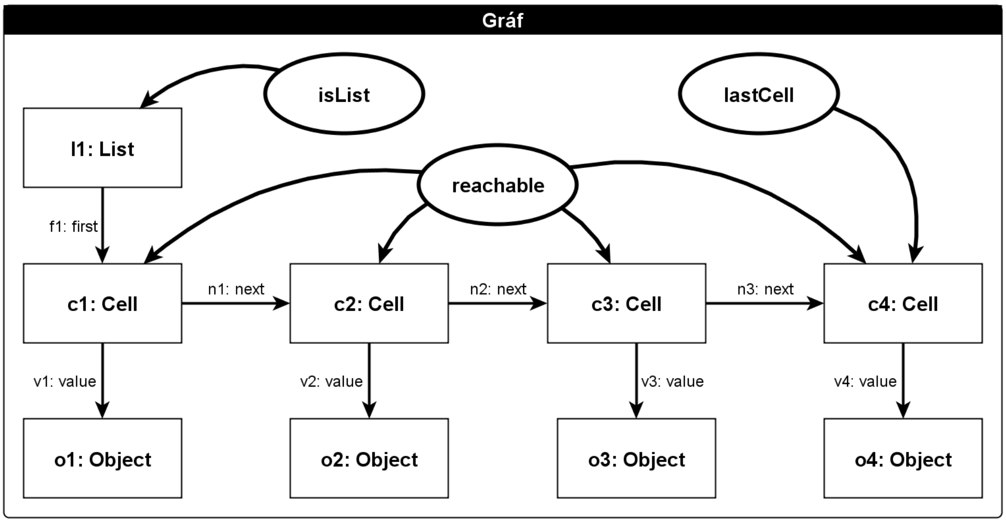
Példa Lássunk csúcspredikátumokra néhány példát:

$isList(n)$	$\Leftrightarrow$	$Node(n, \texttt{List})$, tehát akkor igaz, ha a csúcs címkéje `List`.
$lastCell(n)$	$\Leftrightarrow$	$Node(n, \texttt{Cell}) \wedge \neg\exists u(Edge(n, u, \texttt{next}))$, ami akkor teljesül, ha n az utolsó cella.
$acyclic(n)$	$\Leftrightarrow$	$\neg Transitive(n, n, \texttt{next})$, azaz a csúcs nem része `next` címkéjű élekből álló irányított körnek.
$nonshared(n)$	$\Leftrightarrow$	$\neg(\exists u \exists v(Edge(u, n, \texttt{value}) \wedge Edge(v, n, \texttt{value}) \wedge \neg u = v))$, azaz nem megy két csúcsból `value` él a vizsgált csúcsunkba.
$reachable(n)$	$\Leftrightarrow$	$\exists u(Node(u, \texttt{List}) \wedge (Edge(u, n, \texttt{first}) \vee Edge(u, v, \texttt{first}) \wedge Transitive(v, n, \texttt{next})))$, azaz egy `List` címkéjű elem által elsőnek jelölt, vagy az abból `next` élekkel elérhető csúcs.

$P(n)$ csúcspredikátum értéke 1, ha az állítás igaz az n csúcsra, és 0, ha nem. Ezek alapján csoportosíthatjuk gráfunk csúcsait:

25. Definíció (Predikátum ekvivalencia). *Legyen P egy csúcspredikátum. Ekkor $\overset{Pred}{\underset{P}{\sim}} \subseteq \mathcal{N} \times \mathcal{N}$ egy olyan ekvivalencia, amelyet a P predikátum reprezentál.*

Példa Figyeljük meg az 5.1 ábrán látható példát! A képen minden csúcsra teljesül az *acyclic* és a *nonshared* feltétel. Az ábrán bekarikázva szerepelnek a *isList*, a *reachable* és a *lastCell* predikátumok. Ezek azokra a csúcsokra teljesülnek, amelyekbe a predikátumból nyilak vezetnek.



5.1. ábra. *Csúcspredikátumok teljesülését bemutató címkézett gráf.*

5.3. Csúcshalmazon értelmezett predikátum

A csúcspredikátumok alkalmazásánál nem elégszünk meg elemek egyedi vizsgálatával. Ebben a szakaszban az lesz a célunk, hogy csúcshalmazokat is tudjunk jellemezni predikátumokkal, így egy új instrumentális ekvivalenciát kapjunk.

5.3.1. Háromértékű logika

Predikátumaink csúcson történő kiértékelése révén két féle eredményt kaphatunk, viszont egy halmaz elemeinek jellemzésére ez kevés. Érdemes bevezetni egy új értéket, amivel a predikátumok igazságtartalmának ismeretlenségét jelölhetjük: $1/2$.

Terjesszük ki megszokott logikai összekötőink értelmezését új, harmadik értékünkre is, ezzel megkapjuk a *Kleene logikát* [8]. Ezeknek műveletek igazságtábláját a 5.1 első sora tartalmazza.

A táblázatunk második sorában láthatunk egy új $\cup$ operátort is, amely az igazságértékek összevonására alkalmas. Ha a művelet két operandusa megegyezik, akkor $P \cup Q$ eredménye P (vagy Q is lehetne), ha eltérnek, akkor az eredmény az ismeretlenséget jelző $1/2$. Értelmezzük a művelet nagyoperátoros megfelelőjét is:

$$\bigcup_{i=1}^{n} = P_1 \cup \ldots \cup P_n$$

$\neg$	0	1	1/2
	1	0	1/2

$\wedge$	0	1	1/2
0	0	0	0
1	0	1	1/2
1/2	0	1/2	1/2

$\vee$	0	1	1/2
0	0	1	1/2
1	1	1	1
1/2	1/2	1	1/2

$\Rightarrow$	0	1	1/2
0	1	1	1
1	0	1	1/2
1/2	1/2	1	1/2

$\cup$	0	1	1/2
0	0	1/2	1/2
1	1/2	1	1/2
1/2	1/2	1/2	1/2

5.1. táblázat. *Háromértékű logikán értelmezett operátorok.*

A $\cup$ műveletet egy halmaz osztályozására fogjuk felhasználni. Legyen H egy halmaz, és elemein értelmezzünk egy P predikátumot, és X értéke legyen a következő:

$$X = \bigcup_{\forall h \in H} P(h), \text{ ekkor:}$$

$$X = 1 \quad \Rightarrow \quad \forall h \in H(P(h))$$
$$X = 0 \quad \Rightarrow \quad \forall h \in H(\neg P(h))$$

A művelet ezen tulajdonságát felhasználva egy adott predikátum szerint osztályozhatjuk halmazainkat, Az osztályozás eredményét pedig felhasználhatjuk számításainkhoz.

5.3.2. Csúcshalmazokon értelmezett predikátum

Az előző fejezetben leírt módszer segítségével definiáljuk egy predikátum szerinti instrumentális ekvivalenciát:

26. Definíció (Csúcshalmazon értelmezett predikátum). *Legyen P egy csúcsokon értelmezett predikátum, és $\overset{IPred}{\underset{P}{\sim}} \subseteq 2^{\mathcal{N}} \times 2^{\mathcal{N}}$ ekvivalencia, amit az alábbi $ipred_P : 2^{\mathcal{N}} \mapsto \{0, 1, 1/2\}$ függvény reprezentál:*

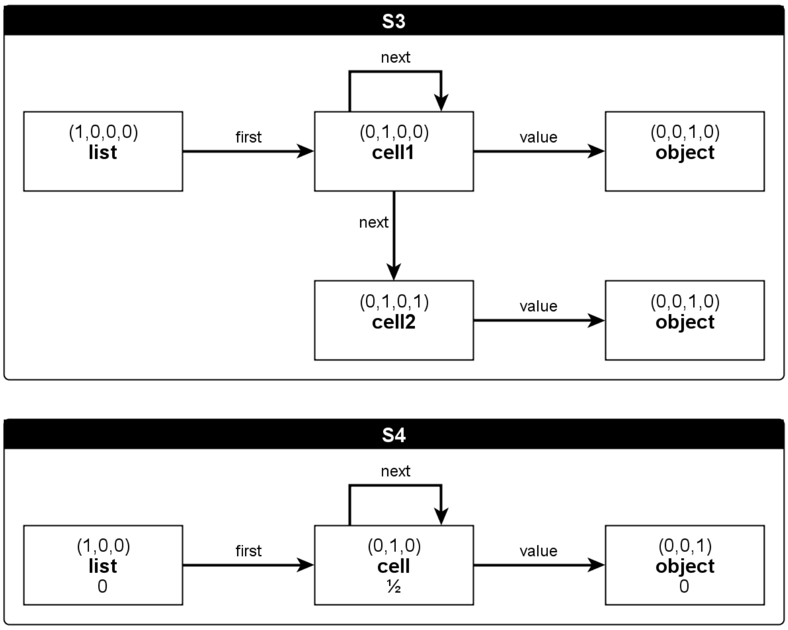
$$ipred_P(U) = \bigcup_{\forall n \in U} P(h)$$

Példa Az 5.2 ábrán két olyan forma látható, amelyet az 5.1 ábrán is megtekinthető listánkból kaptunk. Az első formán kompatibilitási, a másodikon instrumentális relációként használtuk fel a $\overset{IPred}{\underset{lastCell}{\sim}}$ predikátumot.

Az *isList* predikátumunkhoz hasonlóan definiáljuk az *isCell*-t és az *isObject*-et is. Az `S3` formát $\overset{Pred}{\underset{isList}{\sim}} \times \overset{Pred}{\underset{isCell}{\sim}} \times \overset{Pred}{\underset{isObject}{\sim}} \times \overset{Pred}{\underset{lastCell}{\sim}}$ kompatibilitási ekvivalenciával kaptuk meg, míg `S4` forma $\overset{Pred}{\underset{isList}{\sim}} \times \overset{Pred}{\underset{isCell}{\sim}} \times \overset{Pred}{\underset{isObject}{\sim}}$ kompatibilitási és $\overset{IPred}{\underset{lastCell}{\sim}}$ instrumentális ekvivalenciát használó formázás eredményeként állt elő.

5.4. Összegzett csúcsok és bizonytalan élek

A predikátumformák csúcsait szokás még egy instrumentális ekvivalenciával csoportosítani, amely meghatározza, hogy a csúcs kompatibilitási ekvivalenciájának hány elem felelt meg.



5.2. ábra. *Kompatibilitási és instrumentális ekvivalenciaként alkalmazott $lastCell$ predikátum.*

Ezt a relációt összegző ekvivalenciának nevezzük.

27. Definíció (Összegző ekvivalencia). *Legyen* $\overset{Sum}{\sim} \in 2^{\mathcal{N}} \times 2^{\mathcal{N}}$ *ekvivalencia az alábbi* $sum : 2^{\mathcal{N}} \mapsto \{0, 1, 1/2\}$ *függvény által reprezentált függvény:*

$$sum(U) = \begin{cases} 1 & ha\ |U| = 1 \\ 1/2 & különben \end{cases}$$

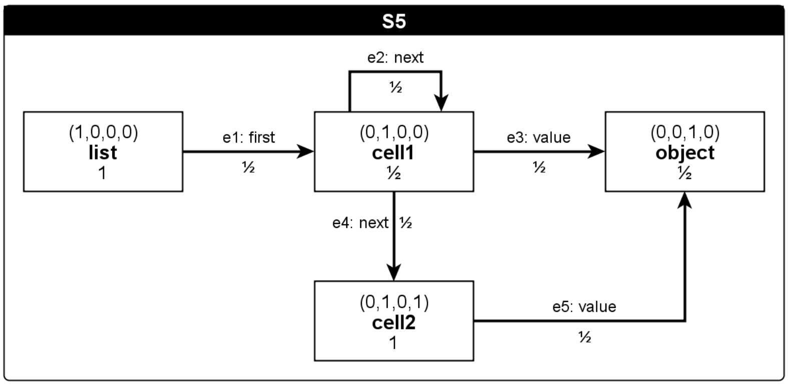
Élek között különbséget tevő ekvivalenciákról eddig még nem esett szó. A formák vizsgálata során hasznos, hogyha tudjuk, mely esetekben vezet egy él a formabeli csúcsok példányai között. Nevezzük azokat a formabeli e éleket bizonytalannak, ahol nem teljesül, hogy ha van olyan csúcs $src(e)$-ben, amiből nem vezet $lab^E(e)$ címkéjű él $trg(e)$ minden konkrét példányába. Ha az e él nem bizonytalan, akkor abból az következik, $src(e)$ és $trg(e)$ kokrét példányai között teljes páros gráf feszül, illetve $src(e) = trg(e)$ esetén olyan teljes gráf amelynek csúcsain még hurokélek is vannak.

A bizonytalanságuk alapján élek közt különbséget tevő reláció definíciója a következő:

28. Definíció (Bizonytalansági ekvivalencia). *Legyen* $\overset{Alw}{\sim} \subseteq (2^{\mathcal{N}} \times 2^{\mathcal{E}} \times 2^{\mathcal{N}})^2$ *ekvivalencia az* $alw : (2^{\mathcal{N}} \times 2^{\mathcal{E}} \times 2^{\mathcal{N}})^2 \mapsto \{0, 1, 1/2\}$ *által reprezentált függvény, amely a következőképpen van definiálva:*

$$sum((S, E, T)) = \begin{cases} 1 & ha\ \forall s \in S, \forall t \in T, \exists e \in E(src(e) = s \wedge trg(e) = t) \\ 1/2 & különben \end{cases}$$

Ha az `S3` forma csúcsait összegző ekvivalenciával, csúcsait bizonytalansági ekvivalenciával osztályozzuk, akkor a 5.3 ábrán látható formát kapjuk.



5.3. ábra. *Összegző csúcsokat és bizonytalan éleket szemléltető forma.*

5.5. Összefoglalás

A fejezetben tárgyalt predikátum alapú ekvivalenciák relatív kevés, de paraméterezés által jól meghatározható csoportra bontják a gráf csúcsait. Ennek előnye, hogy kiemelhetjük az analízis szempontjából lényeges, problémaspecifikus sajátosságokat. Hátránya, hogy a paraméterezés nem, vagy csak rosszul automatizálható.

Ekvivalenciák szorzatával meglévő ekvivalenciarelációink együttes használatára válik lehetségessé. Ezáltal az ekvivalenciaosztályok szaporodása árán elérhető, hogy több szempontból egyszerre osztályozzuk a gráfelemeket, így ötvözhessük módszereink jó tulajdonságait.

6. fejezet

Absztrakt gráftranszformáció

Ebben a fejezetben meghatározásra kerül az absztrakt transzformáció és transzformátor fogalma. Szerepelni fog a témával kapcsolatos szakirodalmakban előforduló absztrakt transzformátor vázának leírása, annak ekvivalenciafüggő megvalósíthatósága. Legvégül megadjuk az absztrakt állapottér fogalmát, és leírjuk kapcsolatát a konkréttal.

6.1. Bevezetés

Az egyszerűbb definíciók végett tisztázzunk előre néhány változót. Legyenek Lab^N és Lab^E véges címkekészletek. $G_0 \in \mathcal{G}(Lab^N, Lab^E)$ kiinduló gráfból R-beli gráftranszformációkat alkalmazva építenénk állapotteret.

Elképzelhető, hogy állapotterünk gráfjaihoz tartozik még C jólformáltsági feltétel is. Ez a feltétel egy olyan predikátum, amelynek állapotterünk minden gráfjára teljesülnie kell. Ha a transzformációk során olyan gráf alakul ki, amely nem felel meg C-nek, eldobjuk és kihagyjuk állapotterünkből.

Legyen továbbá:

$\overset{Comp}{\sim} \subseteq \mathcal{N} \times \mathcal{N}$	–	$f_{Comp} : \mathcal{N} \mapsto Comp$ által reprezentált csúcsok fölött értelmezett ekvivalencia
$\overset{Instr^N}{\sim} \subseteq 2^{\mathcal{N}} \times 2^{\mathcal{N}}$	–	$f_{Instr^N} : 2^{\mathcal{N}} \mapsto Instr^N$ által reprezentált csúcshalmazok fölött értelmezett ekvivalencia
$\overset{Instr^E}{\sim} \subseteq (2^{\mathcal{N}} \times 2^{\mathcal{E}} \times 2^{\mathcal{N}})^2$	–	$f_{Instr^E} : 2^{\mathcal{N}} \times 2^{\mathcal{E}} \times 2^{\mathcal{N}} \mapsto Instr^E$ által reprezentált ekvivalencia

Ahelyett, hogy akár végtelenségig építenénk álapotterünket, áttérünk az absztrakt gráfok $\mathcal{S}(Comp, Instr^N, Lab^E, Instr^E)$ halmazára. Ehhez kiszámoljuk a kezdőállapot absztrakt megfelelőjét:

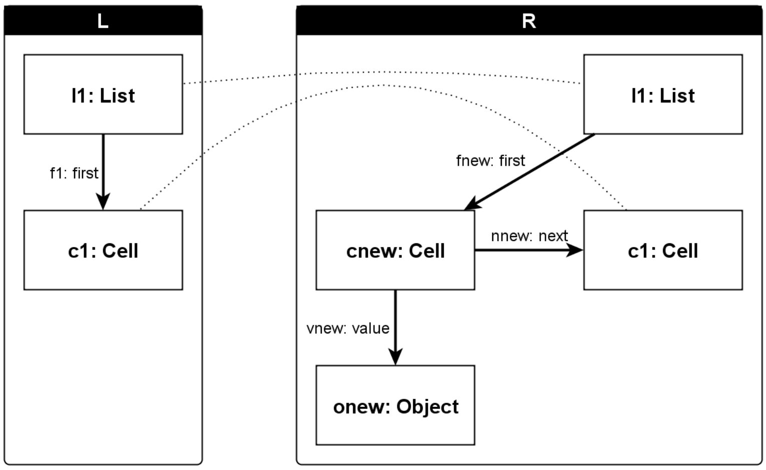
$$S_0 = shaping(G_0, \overset{Comp}{\sim}, \overset{Instr^N}{\sim}, \overset{Instr^E}{\sim})$$

Ezen az S_0 formán fogjuk alkalmazni az R-beli transzformációkat.

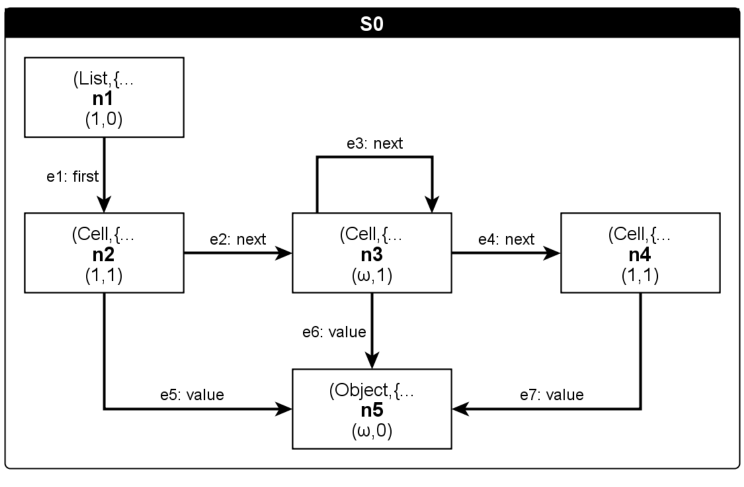
Példa Az általános definíciókat most is példákkal szemléltetjük. Példáinkban a 4 elemű láncolt listát ábrázoló $G_0 \in \mathcal{G}(\{\texttt{List}, \texttt{Cell}, \texttt{Object}\}, \{\texttt{first}, \texttt{next}, \texttt{value}\})$ kezdőállapotból indulunk ki. Egyedül a `Put` transzformációt használjuk, ami a 6.1 ábrán látható. Példánkban előforduló gráfjainkhoz tartozik még egy jólformáltsági feltétel is, mely megtiltja párhuzamos élek létezését.

Gráfunk csúcsait szomszédsági ekvivalencia szerint szeretnénk csoportosítani, amelynél a 1 a távolság és a párhuzamosság is. A csúcsok instrumentális ekvivalenciájaként egyrészt maximálisan 1 értékű multiplicitást, másrészt elérhetőséget vizsgáló predikátumot választunk. Mivel éleinket a jólformáltsági viszony eléggé korlátozza, úgy gondoljuk, nem szükséges élek instrumentális ekvivalenciájával szaporítani formáink számosságát. Ezt egy olyan $\overset{\forall}{\sim}$ ekvivalencia bevezetésével tesszük meg, ami mindig teljesül, így egyetlen ekvivalenciaosztálya van.

Az előbbiekben megfogalmazott paraméterezéssel alkalmazott formázás során megkapjuk az 6.2 ábrán látható $\texttt{S0} = shaping(G_0, \overset{Neigh}{\underset{1,1}{\sim}}, \overset{Nm}{\underset{1}{\sim}} \times \overset{Pred}{\underset{reachable}{\sim}}, \overset{\forall}{\sim})$ formát.



6.1. ábra. *A* $\texttt{Put} = (\texttt{L}, \texttt{R})$ *transzformációt leíró diagram. Az összerendelt gráfelemeket pontozott vonal köti össze. A transzformáció végrehajtásakor egy új* `Cell` *címkéjű csúcs kerül a lista elejére, amelyhez egy új* `Object` *is tartozik.*



6.2. ábra. *Az* `S0` *forma.*

6.2. Absztrakt gráftranszformációk

Absztrakt állapotterünk építése során a célunk, hogy minden konkrét állapotátmenetet rögzítsünk, ezért a formatranszformációknak felül kell becsülniük a gráftranszfromációkat. Ebből következik a formatranszfromációk definíciója:

29. Definíció (Formatranszformáció). *Legyen P egy gráftranszformáció. A $t_P\mathcal{S} \mapsto 2^{\mathcal{S}}$ függvényt formatranszfromációnak nevezzük, ha $t_P(S) = SH$ esetén teljesül az alábbi feltétel:*

$$\forall G \in concr(S), \forall m, \exists F \Big(G \xrightarrow{P,m} F \ \Rightarrow \ shaping(F) \in SH \Big)$$

A transzformációkkal paraméterezhető t neve absztrakt transzformátor.

Amint látható, a transzformáció egyértelmű eredménye nincs meghatározva, ezt a transzformátorok definiálják. Így az absztrakt állapotterünk kifejtése előtt meg kell adnunk egy, a feltételeknek megfelelő transzformátort.

A két transzformátor közül azt tekintjük pontosabbnak, amelynek eredménye kisebb halmaz. Ennek magyarázata az, hogy a kötelező állapotátmenetek mellett általában szerepelnek olyanok is, amelyek az absztrakcióból származó bizonytalanságból származnak. Ezekkel a csak absztrakt szinten létező átmenetekkel olyan formába is eljuthatunk, amelynek konkretizáltjai nem szerepelnek a konkrét állapottérben (esetleg nincsenek is konkretizáltjai), így biztonságos gráfnyelvtan esetén is meghiúsulhat a helyességellenőrzés.

A transzformátorok közül van egy, ami legalább olyan pontos, mint a többi, és ezt tökéletes transzformátornak [3] nevezzük.

30. Definíció (Tökéletes absztrakt transzformátor). *A t transzfromátort tökéletesnek nevezzük, ha minden P gráftranszformáció esetén $t_P(S)$ minimális.*

A tökéletes transzformátor működésének menete úgy képzelhető el, hogy a formánknak kiszámítjuk az összes konkretizáltját, azokon végrehajtjuk a gráftranszformációkat, melyeknek eredményeit újra formázzuk. A módszer kivitelezhetlensége az összes konkretizált megszerzésében leledzik, mivel ezek számossága akár végtelen is lehet.

6.3. Absztrakt transzformátor leírása

Ebben a szakaszban megadásra kerül egy olyan transzformátor váza, amely a formázáshoz használt ekvivalenciákból származó összefüggések felismerése és alkalmazása mellett használható formatraszformációk végrehajtására. A transzformáció menete a következő lesz:

1. Megkeressük, formánk mely csúcsaira és éleire illeszkedhet a transzformáció bal oldala. Ezt nevezzük absztrakt morfizmusnak.

2. Az illeszkedés mentén konkretizáljuk formánk azon részét, amelyre illeszkedhet a bal oldal. A műveletet fókusznak nevezzük.

3. A fókuszált gráfunk ellentmondhat a formánk reprezentációiban tárolt ismeretnek. Ezeket az eseteket ki kell szűrni.

4. Végrehajtjuk a transzformációs lépést a formánk konkrét részén. Ha kell, az absztrakt gráfelemek reprezentációit megváltoztatjuk.

5. A transzformációs lépést után a transzformáció jobb oldalára fókuszált részgráfot kapunk. Ezt a normalizálásnak nevezett folyamat alatt vissza kell alakítani formává.

A transzfromáció lépéseikor több részeredményt is kaphatunk. A kapott részeredmények mindegyikével tovább kell számolnunk, a normalizálás végén megmaradt formák lesznek a transzformáció eredményei.

A következő definíciókban bevezetünk egy új szóhasználatot: A összeegyeztethető B-vel. Ennek jelentése az, hogy A absztrakt gráfelem reprezentációiból nem lehet belátni, hogy B ne lehetne eleme az A által képviselt csoportnak.

6.3.1. Absztrakt morfizmus

Absztrakt morfizmus során megpróbálunk egy konkrét gráfot illeszteni formánkra. Az illesztés során a gráfelemek illeszthetőségét elemenként vizsgáljuk, azaz a gráfunk csúcs és élcímkéit össze kell vetnünk formánk reprezentációival.

31. Definíció (Absztrakt morfizmus). *Egy G gráf és egy S forma közötti* absztrakt morfizmuson *egy olyan m morfizmust értünk, amelyre teljesülnek az alábbi feltételek:*

- $\forall e \in E_G \Big(lab_G^E(e) = olab_S(m(e)) \Big)$
- *minden G-beli l címkéjű csúcs összeegyeztethető $m(n)$-nel.*

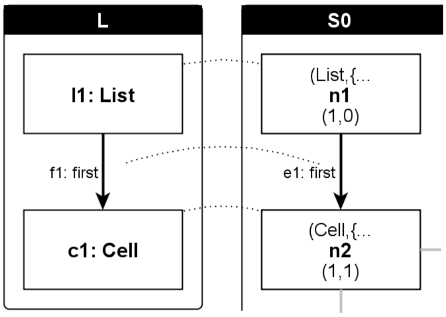
Az absztrakt morfizmus legfontosabb tulajdnonsága, hogy lefed minden konkretizáltakban előforduló morfizmust. Ha létezik G és F gráfunk között egy m morfizmus, és $S = shaping(S)$ esetén m_s a formázás morfizmusa, akkor $m_a = m_s \circ m$ egy G és S közötti absztrakt morfizmus.

Példa A 6.3 ábrán láthatunk egy példát absztrakt morfizmusra, a `Put` transzformáció bal oldalát szeretnénk m_0 morfizmus mentén illeszteni `S0`-ra. Mivel az `S0` formánkban csupán egyetlen `first` címkéjű él van, az ábrán látható illeszkedésen kívül minden más kizárható. A `List` és `Cell` címkéket `n1` és `n2` reprezentációi nem cáfolják meg, így m_0 absztrakt morfizmus.

6.3.2. Fókusz

A tökéletes absztrakt transzfromátor az összes konkretizálton végrehajtaná a transzformációt, ám ez megvalósíthatósági korlátokba ütközik. A probléma áthidalására kompromisszumot kell kötnünk, és gráfunknak csak azon részét konkretizáljuk, amelyen a transzformációt végrehajtanánk. Ezt a műveletet fókuszálásnak nevezzük, eredményét pedig fókuszált formának.

32. Definíció (Fókuszált forma). *Legyen G egy címkézett gráf. Az F gráfot G-re* fókuszált formának *nevezzük, ha részgráfként tartalmazza G-t, G-n kívüli elemei egy formának megfelelően vannak címkézve és G-n kívül megfelel a forma feltételeinek, azaz:*



6.3. ábra. *A* `Put` *transzformáció bal oldalának illesztése az* `S0` *formára. Az* m_0 *absztrakt morfizmust pontozott vonal jelöli.*

- *„Nincsenek G-n kívüli hasonló csúcsok":*

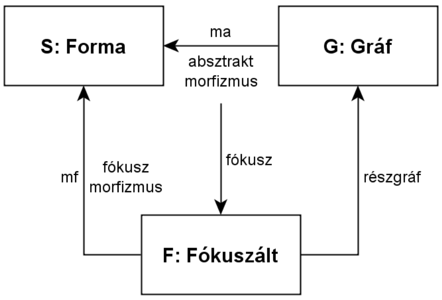
$$\forall n_1, n_2 \in N_F \setminus N_G \Big(n_1 \neq n_2 \Rightarrow comp_F(n_1) \neq comp_F(n_2) \Big)$$

- *„Nincsenek G-n kívüli párhuzamos élek":*

$$\forall e_1, e_2 \in E_F \setminus E_G \Big(e_1 \neq e_2 \Rightarrow \\ \neg \Big(olab_F(e_1) = olab_F(e_2) \land src_F(e_1) = src_F(e_2) \land trg_F(e_1) = trg_F(e_2) \Big) \Big)$$

Egy G-re fókuszált forma G-beli elemeit konkrét elemeknek, G-n kívülieket absztraktnak nevezzük.

A fókusz műveletével elérhetjük formánk egy morfizmus mentén történő kibontását. Egy G és S közötti absztrakt morfizmus mentén történő fókusz során feladatunk egy olyan G-re fókuszált F forma készítése, ami előállhat a formánk részének konkretizálásával.



6.4. ábra. *Fókusz során előforduló morfizmusok szemléltetése*

33. Definíció (Fókusz). *Legyen G egy címkézett gráf, S egy forma, m_a egy G és F közötti absztrakt morfizmus, F egy G-re fókuszált forma. Akkor mondjuk, hogy S-nek m_a szerinti* fókusza *F, ha létezik olyan F és S közötti m_f* fókusz morfizmus *(lásd: 6.4 ábra) amelyre teljesülnek az alábbi állítások:*

- *S minden olyan g_S csúcsára és élére, amely nem szerepelt az absztrakt morfizmusban (azaz $\operatorname{inv} m_a(g_S) = \emptyset$, és él esetén még $\operatorname{inv} m_a(src_s(g_S)) = \emptyset \land \operatorname{inv} m_a(trg_s(g_S)) = \emptyset$),*

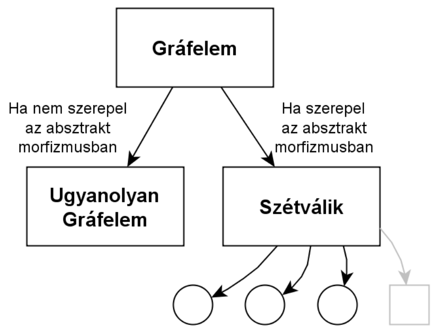
teljesül, hogy $\mathrm{inv}\, m_f(g_S) = \{g_F\}$ *egyelemű halmaz,* g_S *absztrakt gráfelem, kompatibilitási és instrumentális reprezentációja illetve élcímkéje megegyezik* g_F*-ével.*
Szemléletesen: fókuszálás során nem változnak az absztrakt morfizmusban nem érintett elemek.

- *S minden olyan* g_S *csúcsára és élére, amely szerepelt az absztrakt morfizmusban (azaz* $\mathrm{inv}\, m_a(g_S) \neq \emptyset$*, vagy él esetén még* $\mathrm{inv}\, m_a(src_s(g_S)) \neq \emptyset \vee \mathrm{inv}\, m_a(trg_s(g_S)) \neq \emptyset$*)), teljesülnek az alábbi feltételek:*

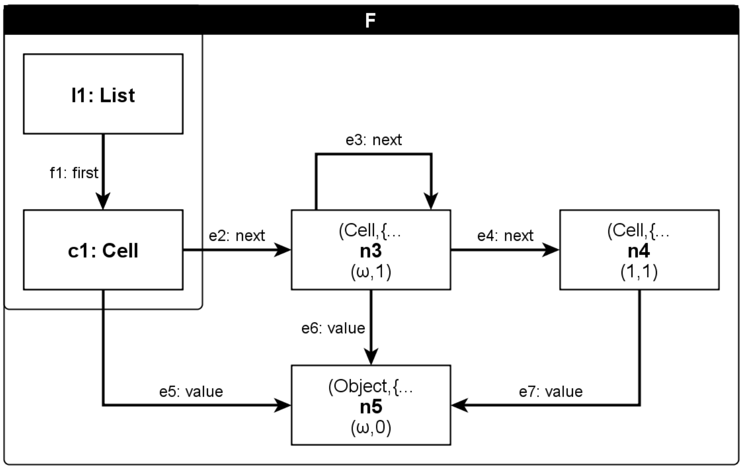
  1. *Ha* g_S *egy él, és* g_F *egy konkrét él, akkor* g_F *címkéje meg kell hogy egyezzen* $olab_S(g_S)$*-bal.*
  2. *Ha* g_S *egy él, és* g_F *is egy absztrakt él, akkor* $olab_F(g_F) = olab_S(g_S)$*.*
  3. *Ha* g_S *egy csúcs, és* g_F *egy konkrét csúcs, akkor* g_F *címkéje csak olyan lehet, amely* $comp_S(g_S)$*-sel összeegyeztethető.*
  4. *Ha* g_S *egy csúcs, és* g_F *is egy absztrakt csúcs, akkor* $comp_F(g_F) = comp_S(g_S)$*.*
  5. *Minden* g_S *instrumentális ekvivalenciája összeegyeztethető a* $\mathrm{inv}\, m_f(g_S)$ *halmazzal.*

  Szemléletesen: az absztrakt morfizmusban érintett elemek szétoszlanak G-nek megfelelő konkrét elemekké, illetve meghagyhatnak még olyan absztrakt elemeket, amelyek a maradékot képzik.

Egy gráfelem fókuszáláskori lehetséges sorsát a 6.5 ábra szemlélteti. Ne felejtsük el, hogy fókuszált formában a konkrét gráfelemeken kívül teljesülnek a formákra tett kikötések, tehát például egy csúcs csak 1 absztrakt gráfelemmé képződhet, mivel különben megsérülne a „nincsenek hasonló csúcsok” feltétel. Éleknek is hasonló feltételeknek kell Vegyük észre, hogy formánk minden konkretizáltjának minden illeszkedése megőrződik a fókuszálás során, így nem hagyunk ki egyetlen konkrét morfizmust sem.



6.5. ábra. *Fókuszáláskor a fókuszmorfizmus inverzének ábrázolása. Az absztrakt morfizmusban résztvevő elemek olyan csoportra bomlanak, amely összegyeztethető az instrumentális ekvivalenciával. A konkrét gráfelemek kis körökkel, a lehetséges absztrakt gráfelem négyzettel van jelölve.*



6.6. ábra. *A 6.3 ábrán látható absztrakt morfizmus szerinti fókusz. A konkrét gráfelemeket bekeretezés jelöli.*

Példa Példánkban végrehajtottuk az absztrakt morfizmus szerinti fókuszálást. A fókuszált függvényünk a 6.6 ábrán látható. Mivel az `S0` gráfban lévő `n1` csúcsnak egy a multiplicitása, és kompatibilitási reprezentációja szerint `List` címkéjű éleket tartalmaz, összeegyeztethető vele a fókuszált `F` gráfban található `l1` csúcs. Hasonlóképpen vezethető le `c1` és `f1` helytállósága is.

6.3.3. Szűrés

Mivel fókuszáláskor csakis azt vizsgáljuk, milyen címkéjű csúcsokat és éleket hozhatunk létre és hányat, ezért előfordul, hogy olyan fókuszált formák jönnek létre, amelyek ellentmondanak a reprezentációikban hordozott információknak. A transzformátor pontosítása végett fontos, hogy minden olyan fókuszált formát kizárjunk, amelyről belátható, hogy nem tartozhat hozzá példány.

Megeshet, hogy bár fókuszált gráfunkhoz tartoznak konkretizáltak, azok egyike sem felel meg az esetleges jólformáltsági feltételünknek. Az ilyen lehetőségeket szintén el kell vetnünk.

Példa Az illeszkedésünk fókuszálásakor előfordulhatott volna az az eset, amikor `f1` élünkkel párhuzamosan megjelenik egy szintén `first` címkéjű absztrakt él. Mivel az absztrakt él is legalább 1 konkrét élt jelöl, ez az eset ellentmond a párhuzamos éleket tiltó jólformáltsági feltételünknek, ezért elvetendő.

6.3.4. Transzformációs lépést

A L-re fókuszált F gráfunkon hajtsuk végre a $P = (L, R)$ gráftranszformációt. Ekkor egy új, R-re fókuszált F_1 gráfot kapunk, amelyben a reprezentációk elavultak lehetnek. A használt ekvivalenciák ismeretében meg kell állapítanunk, mely reprezentációk változhattak meg. Az összes lehetséges változást meg kell állapítanunk, fel kell jegyeznünk és tovább kell számolnunk velük.

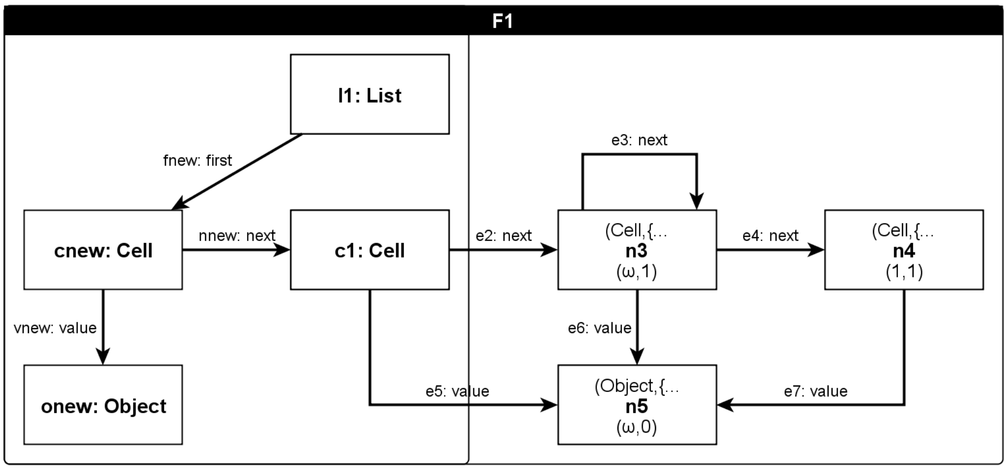
Példa Ha alkalmazzuk fókuszált `F` gráfunkon a `Put` transzformációt, a 6.7 ábrán látható `R`-re fókuszált gráfot kapjuk. Nézzük végig az absztrakt elemek reprezentációit, melyek változhattak.

A szomszédsági reprezentációkat vizsgáljuk először. Az `n3` csúcs a kompatibilitási reprezentációjában azt tárolja, hogy `next` címkéjű bejövő élen `Cell` címkéjű csúccsal 1 alkalommal volt szomszédos. Mivel az átalakítás során sem `c1` nem változott, sem más szomszédja, ezért $comp_{\texttt{F1}}(\texttt{c3})$ is marad a régi. Hasonló a helyzet az `n5` csúccsal is.

Érdemes észrevenni, hogy 1 távolságú szomszédsági ekvivalencia esetén csak akkor módosulhat a reprezentáció, ha a formabeli csúccsal közvetlen szomszédságban lévő elemek módosulnak. Ezek szerint `n4` csúcs kompatibilitási reprezentációja sem változott.

Az instrumentális ekvivalenciáknál a multiplicitások nem változhatnak. Más a helyzet a predikátumainkkal. A `first`-tel címkézett élünk megszűnésével és egy új beszúrásával a *reachable* csúcspredikátumok teljesülését felül kell vizsgálnunk. Tudjuk, hogy amikor még a `c1` csúcsba vezetett az első csúcsot jelző él, akkor teljesült a feltétel `n3`-ra és `n4`-re is. Kikövetkeztethető, hogy ez a tulajdonság akkor is megmarad, ha `c1` a második cella. Mivel az `n5` csúcsba nem megy `next` címkéjű él, a predikátum erre nem teljesülhet.

Az absztrakt élek reprezentációi az egyszerű $\overset{\forall}{\sim}$ ekvivalenciából kifolyólag sosem változnak, így elképzelhető egy olyan transzformációs lépés, amelyben gráfunk címkéi nem módosulnak.



6.7. ábra. *A `Put` transzformáció alkalmazása után létrejövő `R`-re fókuszált forma.*

A transzformációs lépés végrehajtása során vigyáztunk, nehogy megszegjük a formatranszformáció kikötését, és kihagyjunk esetet.

6.3.5. Normalizálás

Normalizálás során inverz fókuszálás műveletét hajtjuk végre, így térve vissza a formák tartományába. A művelet során először határozzuk meg a konkrét csúcsokkal összeegyeztethető kanonikus reprezentációkat. Ez a művelet úgy hajtandó végre, mint a formázás, azzal a nehezítéssel, hogy a már meglévő absztrakt elemeket is bele kell venni a számításba.

A megegyező kanonikus reprezentációjú elemek, és az azok között húzódó párhuzamos élek csoportokat alkotnak. A csoportok instrumentális ekvivalenciaosztályának összeegyeztethetőnek kell lennie a csoport elemeivel. Feladatunk szintén hasonló a formázáshoz, azzal a kivétellel, hogy absztrakt gráfelemek instrumentális ekvivalenciaosztályait is vizsgálni kell.

Példa Határozzuk meg az `F1` `R`-re fókuszált gráfunk konkrét csúcsainak kompatibilitási reprezentációit a szomszédsági ekvivalencia reprezentálófüggvényével!

$$\boxed{\begin{array}{rcl}
nrep_{1,1}(\mathtt{l1}) & = & (\mathtt{List}, \{\}, \{(\mathtt{Cell}, \mathtt{first}, 1)\}) \\
nrep_{1,1}(\mathtt{cnew}) & = & (\mathtt{Cell}, \{(\mathtt{List}, \mathtt{first}, 1)\}, \{(\mathtt{Cell}, \mathtt{next}, 1), (\mathtt{Object}, \mathtt{value}, 1)\}) \\
nrep_{1,1}(\mathtt{c1}) & = & \begin{cases} (\mathtt{Cell}, \{(\mathtt{Cell}, \mathtt{next}, 1)\}, \{(\mathtt{Cell}, \mathtt{next}, 1), (\mathtt{Object}, \mathtt{value}, 1)\}) \\ \text{vagy} \\ (\mathtt{Cell}, \{(\mathtt{Cell}, \mathtt{next}, 1)\}, \{(\mathtt{Cell}, \mathtt{next}, \omega), (\mathtt{Object}, \mathtt{value}, 1)\}) \end{cases} \\
nrep_{1,1}(\mathtt{onew}) & = & (\mathtt{Object}, \{(\mathtt{Cell}, \mathtt{value}, 1)\}, \{\})
\end{array}}$$

Ahogy látható, konkrét szomszédság esetén ugyanazt a műveletet kell végrehajtani, mint formázáskor. Gondot jelentett viszont a `c1` kiszámítása, mert bár tudtuk, hogy kimenő élen szomszédos egy másik `Cell`-lel, nem tudtuk a szomszédság párhuzamosságát megállapítani, mert éleink nem hordoznak semmilyen adatot. Ekkor az összes lehetőséget fel kell venni, így lesz olyan esetünk, amikor a párhuzamosság 1, és lesz, amikor ω. A példában az 1-es a lehetőséget számoljuk végig.

A 6.8 ábrán láthatóak az ugyanolyan kompatibilitási reprezentációval rendelkező csoportok. Az instrumentális ekvivalenciaosztályok megítélése a csak konkrét elemekből álló csoportok esetén ugyanúgy zajlik, mint fókuszáláskor, ezért `n6` és `n7` címkéi egyértelműek. Multiplicitások megítélésénél fel kell fedezzük, hogy egy egyelemű és egy ω elemű halmaz uniójának számossága ω. Ezzel megállapítottuk az összevont csoportok multiplicitását. A csoportokat leíró predikátumok összevonásánál sem kerülünk nagy bajba, hiszen P-vel és Q-val reprezentált csoport uniójának reprezentációja $P \cup Q$. Ebből kifolyólag a középső cellák elérhetősége $1 \cup 1 = 1$, az objektumoké pedig $0 \cup 0 = 0$.

Élek esetén az összevonás az egyosztályú ekvivalencia miatt egyértelmű.

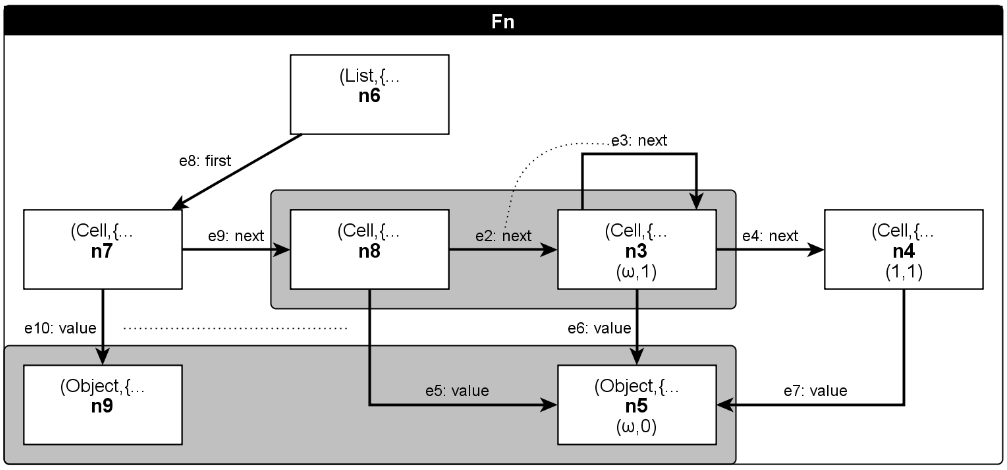
Ha elvégezzük az összevonást, az `S0`-val izomorf formát kapunk.

Mivel minden lépésnél vigyáztunk arra, hogy felülről becsüljük meg a lehetséges átmeneteket, a normalizálás végén minden lehetséges absztrakt állapotátmenetet megkapunk. Mindemellett a formák részleges kifejtésénél sok hamis álapotátmenetet kizártunk.

6.4. Absztrakt állapottér

A kiindulási S_0 formából t absztrakt transzformátor és R által elérhető elérhető összes forma absztrakt állapotteret épít. Ezt ugyanúgy ábrázoljuk, mit a szokványos, gráfokból épülőt, azzal a különbséggel, hogy a csúcscímkék formák.

34. Definíció (Elérhető absztrakt állapot). *Legyen $S_0 \in \mathcal{S}$, R gráftranszformációs szabályok halmaza és t egy absztrakt transzformátor. Ekkor S_0-ból R-beli transzformációkat*



6.8. ábra. *Kompatibilitási reprezentáció megállapítása utáni csoportokat bemutató ábra. Az egynél többelemű csúcscsoportok bekeretezéssel vannak összerendelve, az így kialakuló párhuzamos éleket pontozott vonal köti össze.*

t-vel alkalmazva elérhető T forma (jelölésben: $S \xrightarrow{t,R} T$), ha léteznek olyan $P_0, \ldots, P_n \in R$ transzformációk, és $S_1, \ldots S_n \in \mathcal{S}$ formák, hogy:

$$S_1 \in t_{P_0}(S_0), S_2 \in t_{P_1}(S_1), \ldots, T \in t_{P_n}(S_n)$$

Egy gráf önmagából definíció szerint elérhető.

35. Definíció (Absztrakt állapottér). *Legyen G_0 címkézett kiinduló gráf, R gráftranszformációs szabályok halmaza és t egy absztrakt transzformátor. Ekkor az $absstates(G_0, R, t)$ függvény eredménye egy olyan $ABSSTATES \in \mathcal{G}(\mathcal{S}, R)$ címkézett gráf, amelyre az alábbi állítások igazak:*

- $S_0 = shaping(G_0)$
- *Minden csúcscímke csak egyszer szerepel, más szóval $lab^N_{ABSSTATES}$ injektív.*
- *Csúcsai azon S formákkal vannak címkézve, amelyekre teljesül, hogy $S_0 \xrightarrow{t,R} S$.*
- *Minden olyan S és T formához és $P \in R$ transzformációhoz, amelyre teljesül, hogy $S_0 \xrightarrow{t,R} S$, $S_0 \xrightarrow{t,R} T$ és $T \in t_P(S)$ pontosan egy él tartozik. Ennek az élnek a címkéje P, irányát tekintve a S-sel címkézett csúcsból vezet a T-vel címkézett csúcsba.*

Az absztrakt állapottér vizsgálhatóságát mondja ki az alábbi állítás:

3. Állítás (Absztrakt állapottér mérete). *Mivel az adott címkekészlettel címkézett formák száma és a gráftranszformációk száma véges, állapotterünk címkekészletei **véges** halmazok. Mivel az absztrakt állapottér címkéi injektíven vannak hozzárendelve egy véges halmazhoz, az absztrakt állapotterünk csúcsainak száma is **véges**. Két állapot között maximum $|R|$ állapotátmenet lehet, így állapotterünk éleinek száma is **véges**. Ebből következően az állapotterünket ábrázoló gráf mérete **véges**.*

$$
\begin{array}{ccc}
\exists\ \textbf{Gráftranszformáció} & \Longrightarrow & \exists\ \textbf{Formatranszformáció} \\
\Downarrow & & \Downarrow \\
\textbf{Gráf elérhető} & \Longrightarrow & \textbf{Forma elérhető} \\
\Downarrow & & \Downarrow \\
\textbf{Gráf szerepel az állapottérben} & \Longrightarrow & \textbf{Forma szerepel az absztrakt állapottérben}
\end{array}
$$

6.1. táblázat

A 6.1 táblázatban megfigyelhetjük a gráfokon és a formákon értelmezett fogalmak és műveletek összefüggéseit.

4. Állítás (Absztrakt állapottér viszonya a konkréthoz). *Ha egy forma vagy transzformáció nem szerepel absztrakt állapotterünkben, annak konkretizáltja sem szerepel a normális állapottérben.*

Így ha egy nemkívánatos állapotátmenetet nem találunk meg absztrakt állapotterünkben, gráfnyelvtanunk biztonságossága bizonyítottá válik.

7. fejezet

Viatra2 rendszer fölé tervezett megvalósíthatósági tanulmány

Ez a fejezet a formák analízisének a Viatra2 modelltranszformációs rendszer [13] fölötti megvalósíthatóságát tárgyalja. Ezen Eclipse alapú technológia sajátosságait felhasználva, mint például a modelltér és annak inkrementális mintailleszési technikája úgy tűnik kibővíthető egy olyan alkalmazássá, amellyel hasonló helyességellenőrzési műveleteket végezhetünk.

7.1. Gráfok és formák tárolása

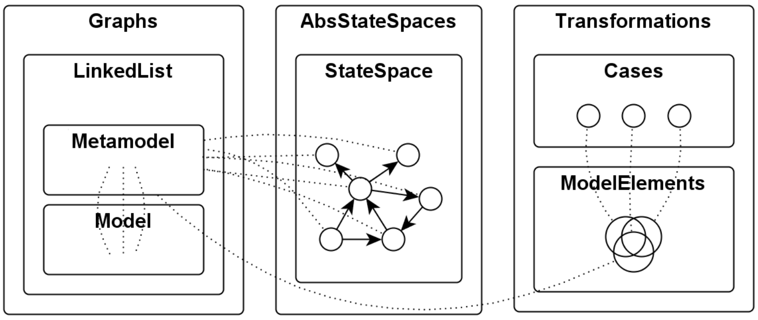
Ez a szakasz tárgyalja a formák analízéséhez szükséges adatok tárolásának egy lehetséges technikáját a Viatra2 modellterében. [4]

7.1.1. A Viatra2 modelltér bemutatása

A Viatra2 rendszer alap tárolási egysége a modelltér. A modellterekben gráfstruktúrába rendeződnek a modellelemek, ahol a csúcsokat entitásoknak, az éleket relációknak nevezzük. Az entitások tartalmazási hiearchiába szerveződnek, amely mentén a lokális névvel rendelkező modellelemek egyedi globális azonosítót kapnak. A modellelemek saját léttel rendelkeznek, tetszőlegesen elérhetőek, létrehozhatóak vagy törölhetőek.

Az elemek között típusosságot és öröklési viszonyt meghatározó kapcsolatok lehetnek, így a modell és a metamodell azonos módon kezelhető és egy helyen tárolható. Egy elemnek több típusa és őse lehet, ezek dinamikusan változhatnak. Egy típus összes példánya könnyen elérhető.

A rendszer magas kifejezőereje mellett alkalmas nagyméretű modellterek kezelésére is. Amíg erőforrásaink engedik, érdemes egy modelltérben dolgoznunk az absztrakt állapottér építése során. A gyökér névtér felépítését a 7.1 ábra szemlélteti, a konkrét elemek szerepét a következő alfejezetek tárgyalják. Az ábrán a névtereket keret jelöli, melyek neve a keret fejrészében található. A névterek tartalmazási hierarchiáját keretek egymásbafoglalása ábrázolja. Körök jelölik az állapotokat és az eseteket, nyilak az állapotátmeneteket. Két névtér között haladó pontozott vonalak típus-példány viszonyt ábrázolnak.

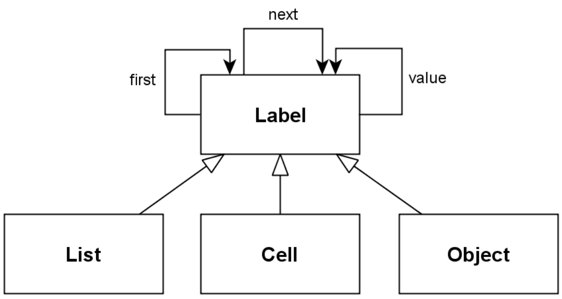


7.1. ábra. *A tárolás felépítését szemléltető ábra.*

7.1.2. Gráfok tárolása

A gyökérben külön névteret alkotó entitás szerepel a gráfok tárolására, amit `Graphs`-nak nevezünk. Ebben a névtérben találhatóak a gráfokat tartalmazó entitások, melyekben kötelezően két elem található: `Metamodel` és `Model`.

`Metamodel`-ben tárolódik gráfunk címkekészlete. Ha az ábrázolt gráfunk a $\mathcal{G}(Lab^N, Lab^E)$ halmazból kerül ki, akkor a Lab^N címkékkel azonos, és egy minden címkétől eltérő `Label` nevű entitás szerepel névterünkben. A címkéket ábrázoló entitások `Label` leszármazottjai. A Lab^E halmaz elemeit olyan relációk ábrázolják, amelyek forrása és célja a kitüntetett szerepben lévő `Label`. Az előző fejezetekben használt lista esetén `Metamodel` tartalma a 7.2 ábrán látható.



7.2. ábra. *A* $\mathcal{G}(\{\mathtt{List}, \mathtt{Cell}, \mathtt{Object}\}, \{\mathtt{first}, \mathtt{next}, \mathtt{value}\})$ *halmazbeli elemek címkekészletének* `Metamodel`*-ben való ábrázolása.*

`Model` névtér entitásai gráfunk csúcsait, két entitás között húzódó reláció pedig az entitásoknak megfelelő csúcsok közötti élt jelöli. Az itt található modellelemek `Metamodel` névtér `Label`-től különböző elemei közül veszik fel egyetlen típusukat, melyekkel a gráfelemek címkéjét adjuk meg. Mivel minden entitás `Label` típusú, tetszőleges Lab^E-nak megfelelő típusú reláció vezethet köztük.

7.1.3. Absztrakt állapottér tárolása

Az állapotterek mérete miatt tartózkodnunk kell az állapotokkal együtt szaporodó objektumoktól, ehelyett modellelemek egyszerű jellemzésére használható annak neve. Ezért a lokális azonosítóban az egyedi név után tetszőleges (rövid) szöveg megadható.

Az `AbsStateSpaces` névtérben foglalnak helyet absztrakt állapottereink. Elképzelhető, hogy egyszerre nem csak eggyel szeretnénk dolgozni, egy régebbi számítást felhasználhatunk heurisztika vagy pontosítás céljából.

Állapottereink névterén belül az állapotokat tároló entitások szerepelnek. Az állapotok közötti átmeneteket olyan relációkkal jelöljük, melynek nevében tároljuk a gráftranszformáció azonosítóját. Ha ezt a feladat megköveteli, a kezdőállapotot jelölhetjük.

Közvetlenül az állapotok névterében helyezkednek el formáink csúcsait ábrázoló entitások. Ezek a kiinduló gráfunk `Metamodel` névteréből veszik fel típusaikat, minden csúcs rendelkezik a vele összeegyeztethető címkéknek megfelelővel. Az entitások közötti relációkkal az éleket jelöljük, ezen kívül az entitások névterét a későbbi használhatóság végett üresen hagyjuk. Az élek *olab* szerinti típust vesznek fel, amely szintén `Metamodel`-ben található. A kompatibilitási és instrumentális reprezentációk tárolása nem a modelltérben történik, azokat `Map` gyűjteményekkel rendeljük a modellelemekhez.

7.1.4. Formatranszformáció részeredményeinek tárolása

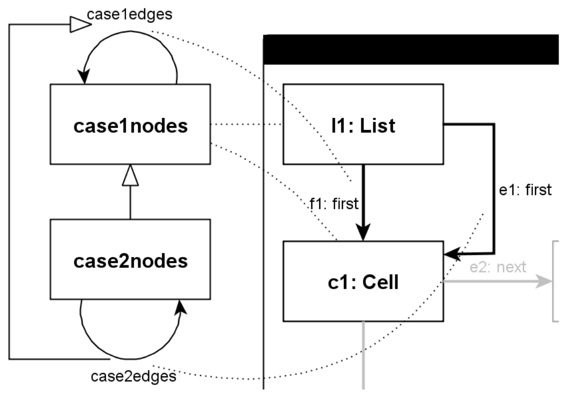
Formatranszformációk során potenciálisan egyre szaporodó esetekre kell bontanunk formánk sorsát, így ezt egy külön erre a célra létrehozott `Transformations` névtérben oldjuk meg. A névtérben több transzformációs részeredményt is tárolhatunk, hasonló megfontolásból, mint amiért több állapotteret is megengedünk. Minden transzformációt jelölő entitásban két névtér található: `Cases` és `ModelElements`.

Esetszétválasztáskor legtöbbször 1-2 modellelemnyi különbség van az esetek között, így érdemes lenne az összes lehetőséget egy modellben ábrázolni, majd megjelölni, melyikhez melyik modellelem tartozik.[1] `Cases` névtérben az eseteket jelölő entitások szerepelnek, amelyek mindegyikéhez egy hurok-reláció tartozik. A vele szomszédos `ModelElements` névtérben található az összes esethez tartozó modellelem, egy modellként ábrázolva. `ModelElements` egy eleme akkor tartozzon egy esethez, ha annak példánya. Relációk hovatartozásának jelölésére szolgálnak az esetekbe kötött hurokélek, esethez tartozás esetén ezek között is vezessen típus-példány viszony.

Az ábrázolás célja az volt, hogy egy esethez tartozó összes modellelemet megkaphassuk úgy, hogy megszerezzük összes példányát. Esetszétválasztásnál egy új esetet leíró entitást és relációt kell létrehozni, majd a példány-viszonyokat megfelelően beállítani. Ha nagy modellekkel dolgozunk, és transzformációink sok esetet dolgoznak fel, lehetséges eseteket egymásból örököltetni is. Erre mutat példát a 7.3 ábra.

Ha analízisünk megkívánja, a számítás menetét is jelölhetjük esetek között húzott relációkkal, melyek nevében tárolhatjuk a számítás lépésének nevét.

[1] Az állapottérben hasonló módon megvalósítható lenne a formák és a hozzájuk tartozó gráfelemek viszonya is. Az ok, amiért nem ezen megoldás mellett döntünk az, hogy gráftranszformációkkal ellentétben a formatranszformációk tetszőlegesen nagy hatást gyakorolhatnak, tehát nem lokális a formatranszformációk hatása. Például a `first` él megszűnésével a *reachable* predikátum tetszőleges mennyiségű reprezentációja átállítódhat.



7.3. ábra. *Esetek öröklését bemutató ábra. Az első eset csúcsai a* `case1nodes`*, élei pedig a* `case1edges` *példányai, a második eset is hasonló módon van ábrázolva. A második az első leszármazottja, és még van egy direkt példánya is:* `e1`*. Így a második eset elemei között szerepel az első összes eleme, és még* `e1` *is.*

7.2. Reprezentációk és ekvivalenciák

Az ekvivalenciák, és azok osztályait megvalósító reprezentációk a formák analízisében kulcsfontosságú szerepet töltenek be. Ezek megvalósíthatóságát tárgyalja ez a szakasz.

7.2.1. Reprezentációk

Formák gráfelemeihez ekvivalenciaosztályok reprezentációit rendeljük. Ezeket a modelltéren kívül tároljuk és kezeljük, a modellelemekhez rendelést egy hatékony `Map` implementációval oldjuk meg. A reprezentációkat a `Representation` absztrakt osztály példányaival ábrázoljuk.

Fontos az állapottér perzisztenciájáról gondoskodnunk. Egyrészt könnyen előfordulhat, hogy a feladat méretéből adódóan nem várhatjuk el az egész állapottér memóriában tartását. Másrészt állapotterünket nem csak gráfnyelvtanok biztonságosságának belátására, hanem annak dokumentálására is használjuk. A modelltér csak állapotok címkék nélküli gráfjának elmenthetőségét biztosítja, így a `Representation` osztály felelőssége biztosítani az egységes (például XML alapú) szerializálást. Nem szabad elfelejtenünk, hogy a címke és modellelem összerendelés hosszútávú tárolását is meg kell ilyenkor oldani.

A reprezentációkat elsősorban ekvivalenciák összehasonlítására alkalmazzuk, ebből fakadóan `Representation` minden leszármazott osztályának felül kell definiálnia az egyenlőségvizsgálat függvényét. Ennek hatékonyságát érdemes megtámogatni egy olyan hash-kód generáló függvénnyel, amelyre teljesül, hogy két objektum egyenlősége esetén megegyezik hash-kóduk értéke is. Ha két reprezentációnak megfelelő hash-kód eltérő, biztosak lehetünk benne, hogy az egyenlőségvizsgálat negatív eredménnyel zárul, így szignifikánsan csökkenthetjük a reprezentációk egyenlőségvizsgálatának számát.

Amint láthatjuk, a reprezentációkra szabott megkötések nem súlyosak, a sorosíthatóság, egyenlőségvizsgálat és hash-kód előállítás szokványos műveletek. Ebből következően `Representation` leszármazottjai majdnem hogy tetszőleges osztályok lehetnek.

7.2.2. Formák hatékony izomorfiavizsgálata

Az absztrakt állapottér építése során szükséges az újonnan létrejövő formákat megvizsgálni, izomorfak-e egy már régebben kiszámítottal. Ez tetszőleges gráfokra rendkívül költséges művelet lenne, ám a formákra teljesülő feltételek megfigyelésével egy gyors módszert adhatunk a problémára.

Azoktól a reprezentációktól, amelyekkel kompatibilitási ekvivalenciaosztályt szeretnénk megvalósítani követeljük meg, hogy rendezhetőek legyenek. Ekkor az S formánkban lévő n csúcsok jellemzőit egyértelműen fel tudjuk sorolni olyan $(comps_S(n), instr_S^N(n))$ párokkal, amely párok sorrendjét a kompatibilitási reprezentációk rendezése adja meg. Mivel a formákban „nincsenek hasonló csúcsok”, a rendezés szigorúan monoton.

Példa Példánkban a 35. oldalon található 5.2 ábrán látható `S4` forma csúcsait rendezzük:

$$\underbrace{((0,0,1,0), 1/2)}_{1.:\texttt{object}}, \underbrace{((0,1,0,0), 1/2)}_{2.:\texttt{cell1}}, \underbrace{((0,1,0,1), 1)}_{3.:\texttt{cell2}}, \underbrace{((1,0,0,0), 1)}_{4.:\texttt{list}}$$

A csúcsok rendezésével minden csúcshoz hozzárendelhetjük a sorban elfoglalt helyének számát. Jelöljük ezt az $order : N_S \mapsto \mathbb{N}^+$ függvénnyel. Mivel a formánkban Lab_S^E már alapból név szerint rendezhető, a csúcsok sorrendezése után a formánk e éleinek jellemzőit is egyértelműen fel tudjuk sorolni $(order(src_S(e)), order(trg_S(e)), olab_S(e), instr_S^E(e))$ ennesekkel. A sorrendet az első három tag lexikális rendezése adja, mivel a formákban „nincsenek párhuzamos élek”, ez szigorúan monoton.

Példa Példánkban a 36. oldalon található 5.3 ábrán látható `S5` forma éleit rendezzük:

$$\underbrace{(2,1,\texttt{value}, 1/2)}_{1.:\texttt{e3}}, \underbrace{(2,2,\texttt{next}, 1/2)}_{1.:\texttt{e2}}, \underbrace{(2,3,\texttt{next}, 1/2)}_{1.:\texttt{e4}}, \underbrace{(3,1,\texttt{value}, 1/2)}_{1.:\texttt{e5}}, \underbrace{(4,2,\texttt{first}, 1/2)}_{1.:\texttt{e1}}$$

A csúcsok és élek rendezéséből álló párost nevezzük formareprezentációnak. Vegyük észre, hogy ha két forma izomorf, akkor reprezentációik is megegyeznek, és viszont. Ebből az észrevételből adódóan két forma izomorfiáját ellenőrizhetjük reprezentációjuk kiszámításával és összehasonlításával. A reprezentációk összehasonlítását gyorsíthatjuk hash-kód ellenőrzéssel is. Sőt, két forma összehasonlításához nem szükséges a modelltérben tartani gráfelemeiket, formareprezentációjuk által is végrehajtható az izomorfiavizsgálat.

A formareprezentációkban szereplő struktúrák mentén megoldható a formák sorosítása is. Ezzel javíthatunk a reprezentációk szerializálásával kapcsolatos megkötésünkön, mivel egy formában szereplő összes reprezentáción egyszerre kell végrehajtani a műveletet. Így megvalósíthatóak olyan reprezentációk is, amelyek relációkon keresztül hivatkoznak egymásra (mint például a szomszédsági).

7.2.3. Ekvivalenciák

A formázáskor használt ekvivalenciákat külön osztályokkal ábrázoljuk. Az osztályok alapvető feladata, hogy reprezentációkat tudjon rendelni olyan elemekhez, amiken értelmezni szeretnénk ekvivalenciánkat. Kompatibilitási ekvivalencia esetén tehát gráfcsúcshoz kell kiszámítani egy reprezentációt.

Ez nem tűnik nehéz feladatnak, ám a meghatározott leképezés esetén a további szolgáltatásokat is meg kell valósítania osztályainknak:

1. A kompatibilitási ekvivalenciáknak adott címkekészlet esetén fel kell tudniuk sorolni egy reprezentációval összeegyeztethető címkéket.

2. Nem elég, ha csak gráfelemekhez és azokból képzett halmazokhoz rendeljük a reprezentációt, mert fókuszált formákban megjelennek absztrakt gráfelemek is. Ekkor fel kell tudnunk sorolni az összes lehetséges kompatibilitási ekvivalenciáját a konkrét csúcsoknak. Ezen kívül meg kell tudnunk határozni az összes lehetséges instrumentális reprezentációját az absztrakt és konkrét elemek összevonásából kialakuló absztrakt elemeknek.

3. Az instrumentális reprezentációk összevonását visszafelé is el kell tudnunk végezni. Tehát amikor fókuszáláskor egy absztrakt gráfelemből előállítunk konkrét és absztrakt elemeket, azoknak is meg kell állapítsuk a lehetséges instrumentális reprezentációját.

4. Az transzformációs lépés előtti és utáni fókuszált formából az absztrakt gráfelemek lehetséges reprezentációit meg kell tudjuk állapítani.

5. Az egyes ekvivalenciáknak a hozzájuk tartozó reprezentációkat vizsgálva hatékonyan fel kell fedniük a fókuszált formákban lévő ellentmondások egy részét.

Amint láthatjuk, sok nehéz feladat hárul osztályunkra, predikátumformák esetén az összes ellentmondás felismeréséhez logikai levezetések szükségesek.

Az analízis hatékonyságát nagyban javítja, ha az ekvivalenciák egymás reprezentációiból is képesek következtetéseket levonni.

7.3. Algoritmusok

Ez a szakasz tartalmazza a formák analízise során használt eljárások megvalósítási vázát.

7.3.1. Mintaillesztés Viatra2-ben

A Viatra2 rendszerben adott feltételeknek megfelelő modellelemek lekérdezésére szolgálnak a minták [4]. Ezek a típusos gráfszerű szerkezetek a modell-gráfra automatikusan illeszthetőek, az illeszkedésnek megfelelő modellelemeket a művelet után eredményként megkapjuk. A minták illesztése programozottan is elérhető.

A rendszer inkrementális gráfmintaillesztő technikát használ. Ez annyit tesz, hogy egy minta tárolja és inkrementálisan karbantartja a gráfminták illeszkedéseit, nagyságrendekkel gyorsítva meg a lekérdezéseket.

A mintákban entitásoknak és relációknak megfeleltethető változók szerepelnek, melyek között megadjuk, mely relációk mely élek között futnak. A mintában kiköthetjük, hogy a változók injektíven külön elemeket vegyenek fel értékül, vagy megengedhetjük a szabad változóbehelyettesítést. A mintákban megszabhatjuk, mely változók között szerepel öröklési, típus-példány vagy tartalmazási viszony. Megköthetjük még továbbá, hogy a változóbehelyettesítéskor a modellelemekre milyen egyéb logikai feltételeknek kell még teljesülnie. Ilyenek lehetnek például névre szóló megkötések, két változó értékének kötelező megegyezése, vagy éppen különbözősége.

Példa A 7.4 ábrán látható egy egyszerű minta, melyben egy olyan `List` típusú `l1` csúcsot és egy `Cell` típusú `c1` csúcsot keresünk, amelyek között halad egy `first` típusú `f1` él. A `shareable` kulcsszóval azt fejezzük ki, hogy egy modellelemet több változó is felvehet értékül. Így a minta illeszkedni fog egy olyan esetre is, amikor `l1` = `c1`. [2]

```
shareable pattern putLeft(l1,c1,f1)={
    List(l1);
    Cell(c1);
    first(f1,l1,c1);
}
```

7.4. ábra. *A `Put` transzformáció bal oldalát nem injektíven illesztő `putLeft` minta.*

A minták változói közül kivezethetjük azokat, amelyeknek illeszkedéskori értékét vizsgálni szeretnénk. A kivezetett változóknak illesztés előtt értéket adva azon esetekre korlátozhatjuk a találatokat, mikor a változóbehelyettesítés egybeesik a megadott értékekkel. Egy minta lehetséges illeszkedéseiből a rendszertől kérhetünk egyet (például egy belyettesített értékre teljesül-e a minta), vagy akár az összeset is (mely elemekre teljesülnek az adott feltételek).

Példa A formázás menete során szükséges megtalálnunk azokat az élpárokat, amelyeknek megegyezik típusuk, forrásuk és céljuk. A keresést a 7.5 ábrán látható minta valósítja meg. Paraméterei közül `e1` és `e2` jelöli a párhuzamos éleket. A lekérdezés paraméterezhető még `Metamodel` paraméterrel, amivel a címkekészlet helyét adhatjuk meg, valamint `State`-tel, amiben azt adjuk meg, mely formában keresünk párhuzamos éleket.

A mintában először két csúcsot határozunk meg, amelyek a `State` névterében helyezkednek el. Ezek lesznek a párhuzamos éleink közös forrásai és céljai. Mivel mintánk injektíven képződik a modellre, párhuzamos hurokélekre nem fog illeszkedni. A két csúcs között keresünk két párhuzamos élt: `e1`-et és `e2`-t. Ezek után meghatározunk a `Metamodel` névtérben egy olyan élt, amely `e1`-nek és `e2`-nek is típusa.

---

[2] A jobb olvashatóság érdekében a típusok nem teljes elérési utukkal vannak jelölve.

A megadható feltételek között található egy olyan, amely a változók egy adott mintában meghatározott behelyettesítésével történő illeszkedését vizsgálja. A vizsgálat irányulhat egyéb, vagy akár rekurzívan ugyan azon mintának teljesülésére is, így megvalósítható a minták kompozíciója. Elérhető továbbá az is, hogy hiúsuljon meg az illeszkedés abban az esetben, ha a változók olyan értékeket vennének fel, amire egy negatív minta teljesülne.

Példa A 7.6 ábrán látható egy `c` csúcson a *reachable* predikátum igazságtartalmát vizsgáló minta. A predikátum igaz, ha létezik legalább egy illeszkedés.

A minta kétféleképpen illeszkedhet. Első esetben (2.-5. sor) akkor, ha egy `List` típusú csúcsból vezet `first` típusú él. Második esetben (7.-11. sor), ha vezet `next` típusú él olyan csúcsból, ami elérhető. Így a minta rekurzívan megkeres `c` csúcsunkból egy sétát egy olyan csúcsba, amire teljesül az első esetben történő illeszthetőség.

```
pattern paralleledges(Metamodel,State,e1,e2)={
    //entitások
    entity(State);
    entity(source) in State;
    entity(target) in State;
    //relációk
    relation(e1,source,target);
    relation(e2,source,target);
    //közös címke a metamodellben
    entity(Metamodel);
    entity(Label) in Metamodel;
    relation(edgelabel,Label,Label);
    //mind a kettő példánya edgelabel-nek
    instanceOf(e1,edgelabel);
    instanceOf(e2,edgelabel);
}
```

7.5. ábra. *Párhuzamos, nem hurokélek megtalálására szolgáló* ***`paralleledges`*** *minta.*

```
1 pattern reachable(c)={
2     //c az első elem
3     entity(c);
4     List(l);
5     first(f,l,c);
6 } or {
7     //egy elérhető elemből vezet next él c-be
8     entity(c);
9     entity(r);
10    next(n,r,c);
11    find reachable(r);
12 }
```

7.6. ábra. *A reachable predikátumot vizsgáló rekurzív minta.*

Az előbbiekben körülírt nagy kifejezőerejű lekérdezőnyelvet használjuk a formákkal kapcsolatos számítások alapjául. Egyrészt, a számításainkhoz szükséges modellelemeket ilyen lekérdezésekkel fogjuk elérni (mint például `paralleledges`). Másrészt csúcspredikátumok igazságértékének meghatározására is alkalmazható (`reachable`). Harmadrészt az összes lehetséges morfizmust is megtalálhatjuk vele (`putLeft`).

7.3.2. Formázás és normalizálás folyamatának leírása

Formázáskor gráfunkat átmásoljuk állapotterünk egy új állapotába, és minden gráfelemét konkrétnak nevezzük. Innentől fogva ugyanazt tesszük, mint normalizáláskor.

Normalizálás első lépésében először megkeressük a konkrét csúcsokat. A csúcsokon alkalmazott ekvivalenciával kiszámoltatjuk ezeknek a reprezentációit, és hozzárendeljük ezeket a csúcsokhoz. Abban az esetben, ha egy csúcsra több lehetőség is adódik, esetekre bontjuk, és tovább számolunk mindegyikkel.

Miután már minden csúcsnak van kompatibilitási reprezentációja, rendezzük ezeket eszerint csoportokba. A csoportosítás hatékonyságát hash-kód ellenőrzéssel javítjuk. A kialakuló csoportok között a párhuzamos éleket megkeressük, és azokat is csoportosítjuk. A konkrét és absztrakt elemekből álló csoportoknak kiszámoljuk az instrumentális reprezentációit, és megcímkézzük azokkal a csoportokat alkotó absztrakt elemeket. Több lehetséges eset előfordulásakor mindegyiket eredménynek kell tekinteni.

7.3.3. Absztrakt transzformáció menete

Az absztrakt állapottér építése előtt elkészítjük a transzformációink baloldalához tartozó absztrakt morfizmust leíró mintát. Mivel a formák csúcsainak típusai már rendelkeznek az összeegyeztethető címkékkel, az élek címkéi pedig $olab$-bal, a minta majdnem megegyezik a baloldallal. Az egyetlen különbség csupán annyi, hogy nem követeljük meg az injektív illesztést (mint például `putLeft`-ben).

Ezek után az összes lehetséges illeszkedés mentén esetszétbontás történik, minden esettel tovább számolunk. Az illesztések mentén szétbontjuk formáinkat fókuszált formákká, amelyben adott, milyen absztrakt elemekből mely konkrét és absztrakt elemek. Az instrumentális ekvivalenciáknak feladatuk, hogy a gráfelemek felbontása alapján eldöntsék a fennmaradó absztrakt elemek instrumentális reprezentációit, vagy azt, ha az absztrakt elem megszűnik. Ha valamely instrumentális ekvivalencia nem ad lehetőséget a felbontásra (azaz a fókusz ezen az absztrakt morfizmuson keresztül lehetetlen), az eseteket elvetjük. Azok az esetek is meghiúsulnak, amikor absztrakt csúcsok törlése esetén valamely megmaradó élnek ne lenne forrása vagy célja. A maradék létrejövő esetekkel tovább számolunk.

Ezek után végrehajtjuk a szűrés műveletét, aminek során a formázás alatt használt ekvivalenciákkal megvizsgáltatjuk az összes esetet, találnak-e ellentmondást benne. Ha igen, azt az esetet elvetjük.

A megmaradó eseteken végrehajtjuk a transzformációs lépést. A végrehajtás előtti és azt követő fókuszált formából minden ekvivalenciának meg kell határoznia új reprezentációi lehetséges értékeit. Minden lehetőséget megtartunk, és tovább számolunk azokkal. Előfordulhat, hogy az új reprezentációkkal címkézett formák ellentmondást hordoznak magukban, ezért érdemes még egyszer végrehajtani a szűrést.

Végül elvégezzük a normalizálás folyamatát. Az összes lehetséges eset adja meg a végeredményt.

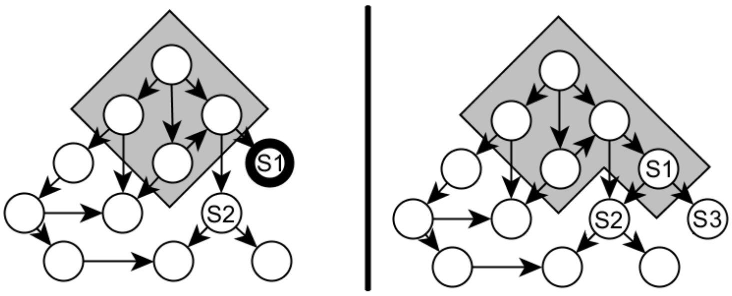
7.3.4. Absztrakt állapottér építése

Állapotterünk építése során két állapothalmazt tartunk számon: kifejtett, és feldolgozandó állapotok.

Kifejtett állapotok közé azok tartoznak, amelyekre végrehajtottuk a feladatban megadott összes transzformációt, és eredményeit elhelyeztük az állapottérben. Ezeknek modellelemeit eldobhatjuk, és ha memóriánk fogyóban van, formareprezentációit kiírhatjuk a háttértárra, majd az objektumokat felszabadíthatjuk. Hash-kódját azonban megőrizzük.

Feldolgozandó állapotok között azok szerepelnek, amelyeket már megkaptunk eredményül, de még nem számítottuk az összes absztrakt transzformációját. Amelyeket már kiszámítottunk, feljegyezzük. Mivel még valószínűleg felhasználjuk a modellelemeket további számításhoz, ezeket a modelltérben tartjuk. A feldolgozandó állapotoknak is kiszámítottuk kanonikus reprezentációit, illetve tároljuk azok hash-kódját.

Állapotterünkben az állapothalmazok viszonyát megfigyelhetjük a 7.7 ábra bal és jobb oldalán, ahol az a kifejtett vagy feldolgozandó állapotokat körök, az állapotátmeneteket nyilak jelzik. A teljesen kifejtett elemeket keret foglalja magába, melyek egymáshoz való viszonya az analízis során már változik.



7.7. ábra. *Absztrakt állapottér építésének bemutatása.*

Kiinduló gráfunkból formázással előállítjuk állapotterünk első elemét, létrehozzuk reprezentációját, eltároljuk hash-kódját, majd a feldolgozandó elemek gyűjteményébe helyezzük. Az állapottér-építés során az alábbi műveletet ismételjük:

1. Kiválasztunk egy elemet a feldolgozandó állapotok halmazából.
2. Végrehajtunk rajta egy még nem kiszámított transzformációt. Ha ez a tiltott mintát képviselő transzformáció, és van eredménye, akkor helyességellenőrzésünk meghiúsul. A transzformációból származó eredményeket elhelyezzük a modelltérben.
3. Ha a transzformációnk végrehajtásával gráfnyelvtanunk minden gráftranszformációjának absztrakt megfelelőjét végrehajtottuk, az állapotot áthelyezzük a kifejtett állapotok halmazába, modellelemeit töröljük.
4. A kapott eredményeknek kiszámoljuk a reprezentációját, majd a hash-kódját. Ha nincs megegyező hash-kód, a feldolgozandó állapotok halmazába kerül. Ha van, akkor megszerezzük a hash-kódhoz tartozó reprezentációkat, és összehasonlítjuk azokat

az új reprezentációval. Ha nincs egyezés, ugyanaz történik, mint ha nem lett volna megegyező hash-kód. Ha van, akkor az új állapototba kerülő állapotátmenetet átirányítjuk a régi állapotba, majd az új állapotot töröljük.

Példa A 7.7 ábra néhány állapotához nevet rendeltünk, amelyet a körön belül található szöveg jelöl. Bal oldalán szereplő állapottérnek válasszuk ki a vastag kerettel ábrázolt még feldolgozandó `S1` állapotát. Végrehajtunk rajta egy még ki nem számított transzformációt, és példánkban két eredményt kapunk: `S2`-t és `S3`-at. Felismerjük, hogy állapotterünkben megtalálható már az `S2` forma, úgyhogy felveszünk egy állapotátmenetet jelző referenciát `S1` és `S2` között. Az `S3` formára viszont még nem akadtunk, ezért új elemként helyezzük el állapotterünkben. Példánkban ezzel kiszámoltuk `S1`-ből az összes gráftranszformáció absztrakt megfelelőjét, ezért áthelyezzük a kifejtett állapotok halmazába. Ezen lépéseket végrehajtva az ábra bal oldali állapotteréből a jobb oldalit kapjuk.

Ha a feldolgozandó állapotainkat tartalmazó halmaz kiürül, helyességellenőrzésünk sikerrel zárult.

8. fejezet

Összegzés

- A végtelen állapotterű rendszerek felett értelmezett shape analízis lényege az **1.2. Feladat bemutatása** című szakaszban található. Ha a **3. Ekvivalencia alapú absztrakció** című fejezetet kibővítjük a **4. Szomszédsági formák** című fejezetben leírtakkal, megkapjuk a szakirodalomban használt szomszédsági formák módszerét. Ha pedig az **5. Predikátumformák** fejezetben leírtakkal vetjük össze, akkor megkapjuk a háromértékű logikán alapuló irodalommal megegyező konstrukciót. A szakirodalomban használt absztrakt transzformációk leírását a **6. Absztrakt gráftranszformáció** fejezet tartalmazza.

- Áttekintettem a Viatra2 modelltranszformációs rendszerben rendelkezésre álló inkrementális gráfmintaillesztési technikákat.

- Az előző pontban leírtakat felhasználva megvizsgáltam, és a **7. Viatra2 rendszer fölé tervezett megvalósíthatósági tanulmány** fejezetben dokumentáltam az egyes shape analízis módszerek átültethetőségét.

Köszönetnyilvánítás

Szeretném köszönetemet kinyilvánítani dr. Varró Dániel konzulensemnek, aki javaslataival, szakirodalomban nyújtott iránymutatásával hatékonyan közreműködött szakdolgozatom sikeres megvalósításában. Köszönet illeti még Horváth Ákost és Ujhelyi Zoltánt, akik a munkám során felmerült problémák megoldásában további segítséget nyújtottak.

Irodalomjegyzék

[1] J. Warmer A. Kleppe and W. Bast. *MDA Explained: The Model Driven Architecture: Practice and Promise.* Addison-Wesley, 2003.

[2] Pataricza András, editor. *Formális módszerek az informatikában.* Typotex, 2006.

[3] Patrick Cousot and Radhia Cousot. Systematic design of program analysis frameworks. Symposium on Principles of Programming Languages, 1979.

[4] István Ráth Dániel Varró Gábor Bergmann, Zoltán Ujhelyi. A Graph Query Language for EMF models. *Transformations, Fourth International Conference, ICMT 2011, Zurich, Switzerland, June 27-28, 2011. Proceedings. Volume 6707 of Lecture Notes in Computer Science, pages 167-182*, 2011.

[5] H.-J. Kreowski H. Ehrig, G. Engels and editor G. Rozenberg. *Handbook on Graph Grammars and Computing by Graph Transformation.* World Scientific, 1999.

[6] Marcos E. Kurbán Iovka Boneva, Arend Rensink and Jörg Bauer. Graph Abstraction and Abstract Graph Transformation. 2007.

[7] J.Sztipanovits and G.Karsai. *Model-integrated computing.* IEEE Computer, 30(4), 1997.

[8] Stephen Cole Kleene. *Introduction to Metamathematics.* Ishi Press International, 1987.

[9] Thomas Reps Mooly Sagiv and Reinhard Wilhelm. Parametric Shape Analysis via 3-Valued Logic. *ACM Transactions on programming Languages and Systems, Vol. 24, No. 3*, 2002.

[10] Arend Rensink. A Logic of Local Graph Shapes. *CTIT Technical Report TR-CTIT-03-35*, 2003.

[11] Arend Rensink. Canonical Graph Shapes. *ESOP 2004, LNCS 2986*, pages 401–415, 2004.

[12] Arend Rensink and Dino Distefano. Abstract Graph Transformation. *International Workshop on Software Verification and Validation (SVV)(2005), Electronic Notes in Theoretical Computer Science*, 2005.

[13] Budapesti Műszaki és Gazdaságtudományi Egyetem. Viatra2 model transformation framework. http://www.eclipse.org/gmt/VIATRA2.

[14] Dániel Varró and András Balogh. The model transformation language of the VIATRA2 framework. *Science of Computer Programming, 68(3)*, 2007.

[15] yWorks. yEd Graph Editor. http://www.yworks.com.