INFORMÁCIÓS ÁLLAPOTOK ÁBRÁZOLÁSA VÉGES ÁLLAPOTÚ AUTOMATÁKKAL

Dyekiss Emil Gergely

Bevezetés[1]

Dialógusok elemzéséhez természetes megközelítést alkalmaznak a dinamikus szemantikák.[2] Ezekben a dialógusok mondatainak jelentését függvényekként modellezzük, melyek a hallgató információs állapotához (új) információs állapotot rendelnek, vagyis a hallottak jelentése megváltoztatja a hallgató információit.[3]

A modellezés egyik célja, hogy jobban megértsük a modellezett folyamatokat. Másik célja lehet, hogy azokat mi is utánozzuk, mesterségesen megvalósítsuk. Az utóbbi esetben gyakorlati szempontból sem mellékes, hogy a modell milyen eszközöket használ. Vannak struktúrák, melyek egyszerűen megfogalmazhatóak, egy elméletben szépen alkalmazhatóak, de ezek használatával a megvalósítás lehetetlen. Ilyen struktúrák a végtelen halmazok is. Jobb, ha ezeket megpróbáljuk kiküszöbölni már elméleti szinten, hogy a gyakorlatban is használható modelleket kapjunk.

A dinamikus szemantikai elméletek közül a legegyszerűbbek modellhalmazokként definiálják az információs állapotokat.[4] Azok a modellhalmazok tartoznak a hallgató aktuális információs állapotába, melyek szerinte lehetségesek. Ezekből elmélettől függően akár végtelen sok is lehet. Természetesen végtelen sok modellt nem tudunk egyszerűen kezelni. Nem tudjuk mindet felsorolni (legfeljebb módszert tudunk adni a felsorolásukra), nem tudjuk az

---

[1] A cikkem alapjául szolgáló előadásokon főleg az információs állapotok automatával való ábrázolásának azt a módját igyekeztem körüljárni, amely a formulák szintaktikai alakján alapul. Az előadások közben és után, illetve más alkalmakkor több ember tett fel kérdéseket, adott javaslatokat, vagy egyéb módon segített a továbbgondolásban. Szeretném megköszönni nekik: Makrai Márton, Maleczki Márta, Muntag Márton, Rebrus Péter, Szabolcsi Anna, Vásárhelyi Dániel. Elnézést kérek, ha valakit kifelejtettem. Kiemelten köszönöm Kálmán László hozzászólásait a témámhoz, aki a modelleken működtetett automaták vizsgálatára biztatott és javasolta, hogy bővítsem az elképzelésemet az *inquisitive semantics* szemléletmódjával is. Továbbá köszönöm a cikk lektorának alapos munkáját és építő megjegyzéseit, amelyek remélhetőleg pontosabbá és érthetőbbé tették mondanivalómat.

[2] Dinamikus szemantikai rendszerekről összefoglaló: Kálmán – Rádai (2001).

[3] Kálmán – Rádai (2001: 89).

[4] Kálmán – Rádai (2001: 88).

összeset tárolni. Valószínűtlen, hogy az emberek ilyen módszerekkel elemeznék a dialógusokat.

Az információs állapotok automatákkal[5] való ábrázolása szemléleti indokokkal is alátámasztható. Ha elrugaszkodunk a modellhalmazokkal való ábrázolástól annak korlátozott alkalmazhatósága miatt, akkor jobban strukturált információs állapotok, például formulasorozatok jöhetnek szóba[6]. Ezek az ábrázolásmódok annyiban ütköznek a szemléletünkkel, hogy amikor a beszélő megismétli egy korábbi mondatát, vagy olyan állítást tesz, melynek a tartalma következik az előzményekből, akkor úgy érezzük, nem tesz hozzá az elhangzottakhoz, az információnkat nem növeli – az információs állapot mégis változik ezzel az ábrázolásmóddal. Ha automatákkal ábrázolnánk az információs állapotokat, akkor azokat talán lehetne úgy definiálni, hogy a dialógusból eddig kinyert összes információt, a kikövetkeztethetőket is elfogadja, ezek ismétlése ne változtasson az információs állapoton[7].

Cikkemben azt vizsgálom, hogyan lehetne modellhalmazok helyett automatákkal ábrázolni az információs állapotokat, és ennek a fajta ábrázolásnak milyen előnyei vannak. Arra törekszem, hogy minél egyszerűbb automatatípusokkal (lehetőség szerint véges állapotú automatákkal[8]) jelenségek minél szélesebb körét lehessen magyarázni.

2. Szintaktikai jellegű megközelítés

Ha az információs állapotokat formulákra alapozva ábrázoljuk, könnyen eszünkbe juthat az információs állapotok automatákká való alakításával kapcsolatban, hogy talán sikerülhet elérnünk a következőket:

(1) Az automata az általa elfogadott formulahalmaz vagy formulasorozat következményeit is elfogadhatná. Ilyen módon a végtelen formulahalmazok talán véges módon leírható automatákkal is kezelhetőek lennének.

(2) Ha a dinamikus szemantikai elmélet, amelynek információs állapotait automatákkal ábrázolnánk, foglalkozik a visszavonás és a hitfelülvizsgálat (*belief revision*) kérdéseivel, az automatás ábrázolásmód eredményeképpen az információs állapothoz tartozó formulasorozat vagy formulahalmaz következményformuláit is elfogadná, akkor ösz-

[5] Az automatákról lásd Bach (2001) és Babcsányi (2007), valamint Eilenberg (1974).

[6] Dyekiss (2010).

[7] Az automata hétköznapi értelmezése szerint bizonyos feladatokhoz nem kell módosítanunk az automatát, mert az „csak dolgozik”.

[8] A véges állapotú automatákról lásd Bach (2001: 29).

sze lehetne vetni ennek a dinamikus szemantikai elméletnek az eredményeit a hit-felülvizsgálati elméletek eredményeivel – azok ugyanis formulahalmazokra építenek, melyek a következményrelációra nézve zártak[9] (és ennél fogva általában végtelen halmazokról van szó).

Azoknak az automatáknak a megtervezése, amelyek a formulasorozatokra építő információsállapot-ábrázolásból indulnak ki, kihívást jelent. Valószínű, hogy az automata bizonyos állapotai elfogadó állapotok lesznek, melyek egy-egy formula végéhez tartoznak. Ezekből kiindulva még több formula végigolvasása is elfogadó állapotokba visz. Ha az összes következményformula elfogadása is követelmény, akkor az alább leírtakat is figyelembe kell venni:

A klasszikus kijelentéslogika szerint egy A formulának következménye $A \vee B$ is, ahol B tetszőleges formula lehet. Tehát lehet egy tetszőlegesen hosszú, jól zárójelezett formula is. Ha az automatát megpróbáljuk könnyen értelmezhető részekre bontani, akkor lehet egy olyan része, amely az összes lehetséges formulát elfogadja. Ha azonban a formulák tetszőleges hosszúak lehetnek, és jól vannak zárójelezve, akkor ennek az elfogadására véges állapotú automata nem képes, csak veremautomata[10].

Ezek szerint a következményrelációra való zártság a klasszikus következményrelációval nem járható út véges állapotú automatákkal. Ha ragaszkodunk a véges állapotú automatákhoz, akkor vagy le kell mondani a következményrelációra való zártságról (és ezzel a hit-felülvizsgálati elméletekkel való könnyű összevethetőségről), vagy módosítani kell a következményreláció fogalmát, nem a klasszikus értelmezéshez kell ragaszkodni.[11] Sajnos ez utóbbit választva is le kell mondani a hit-felülvizsgálati elméletekkel való könnyű összevethetőségről, azok ugyanis nem a módosított következményrelációval dolgoznak.

Ezek a megszorítások már önmagukban elég okot adnak arra, hogy megpróbáljunk elszakadni a szintaktikai megközelítéstől az információs állapotok automatákkal való ábrázolásánál, és megpróbálkozzunk a szemantikai megközelítéssel. Szerencsére ott nagyobb sikerrel járhatunk.

Elképzelésem ismertetése közben törekszem a formalizálással elérhető pontosságra, de nem fogok mindent ténylegesen formalizálva definiálni.

[9] Alchourrón – Gärdenfors – Makinson (1985).

[10] Zárójelezésről, veremautomatákról lásd Bach (2001: 79-126).

[11] Pontosítandó kérdés, hogy milyen következményrelációt érdemes használni – példák nem klasszikus logikákra: fuzzy logika, releváns logika, lineáris logika, nem monoton logikák.

3. Szemantikai megközelítés

Az információs állapotok automatákkal való ábrázolásának szemantikai megközelítése olyan fogalmakra épít, amelyek a klasszikus szemantikai elméletek alapfogalmai, mint például a nyelv modelljei. Ennél pontosabban a részletekből derül ki, hogy mit is értsünk rajta.

3.1. A kiinduló elmélet

Az egyszerűség kedvéért egy kijelentéslogikai rendszerből induljunk ki. Ennek definíciója a következő:[12]

1. Definíció: A nulladrendű kijelentéslogika nyelve
$L_0 =_{\text{def}} \langle \mathbf{LK}, \mathbf{NLK}, \mathbf{F} \rangle$
Elemei a logikai- és a nemlogikai konstansok, valamint a formulák.

2. Definíció: L_0 logikai konstansai
$\mathbf{LK} =_{\text{def}} \langle (,), \neg, \wedge, \vee \rangle$
Ezek a nyitó és csukó zárójel, a negáció, a konjunkció és a diszjunkció jelei.

3. Definíció: L_0 nemlogikai konstansai
$\mathbf{NLK} =_{\text{def}} \{p, q, \ldots\}$
Elemei propozícióknak (kijelentéskonstansok) felelnek meg.

4. Definíció: L_0 formulái
F a legkisebb halmaz, amely eleget tesz a következőknek:

(a) Ha $p \in \mathbf{NLK}$, akkor $p \in \mathbf{F}$. A *kijelentéskonstansok* formulák is egyben.
(b) Ha $A \in \mathbf{F}$, akkor $\neg A \in \mathbf{F}$. A *tagadott formulák* is formulák.
(c) Ha $A, B \in \mathbf{F}$, akkor $(A \wedge B) \in \mathbf{F}$. A *konjunktív formulák* is formulák.
(d) Ha $A, B \in \mathbf{F}$, akkor $(A \vee B) \in \mathbf{F}$. A *diszjunktív formulák* is formulák.

Nézzünk ehhez a nyelvhez egy egyszerű felfrissítő szemantikai elméletet!

5. Definíció: Nulladrendű nyelvek modelljei:

Az $M =_{\text{def}} \langle I, H, IP \rangle$ rendezett hármas akkor és csak akkor modell az L_0 nyelvhez, ha $I \cap H = \emptyset$ és $I \cup H \neq \emptyset$ (az I az igaz, a H a hamis tényállások halmaza; uniójukat U-val jelöljük, ez a modell univerzuma), és az IP interpretációs függvényre pedig a következő teljesül:

Ha $p \in \mathbf{NLK}$ akkor $IP(p) \in U$.

[12] Kálmán-Rádai (2001: 207) nyomán, kicsit módosítva.

6. Definíció: Információs állapotok a nulladrendű felfrissítő szemantikában:[13]

Legyen **M** az L_0 nulladrendű nyelv modelljeinek osztálya. Ekkor a nulladrendű felfrissítő szemantika minden σ információs állapotára igaz, hogy $\sigma \subseteq \mathbf{M}$.

Az egyszerű felfrissítő szemantikában minden formula szemantikai értéke egy függvény, amely **M** hatványhalmazának elemeihez **M** hatványhalmazának elemeit rendeli, vagyis információs állapotokhoz információs állapotokat rendel. Ezt formulatípusonként pontosan definiálni lehet, de ettől most eltekintek, mert a cikkem témája szempontjából érdektelen részleteket tartalmazna csak.

3.2. Könnyen felsorolható nemlogikai konstansok

A nem logikai konstansokat ne különböző betűkkel jelöljük, hanem sorszámmal.

NLK $=_{\text{def}}$ $\{p_i$: $i \in \mathbf{Z}^+$ és $p_1 \in$ **NLK** és nincs olyan $1 < j \in \mathbf{Z}^+$, hogy ha $p_j \in$ **NLK**, akkor $p_{j-1} \notin$ **NLK**$\}$ vagyis **NLK** elemei p_i-k, ahol i 1-től folyamatosan egyesével növekszik (valameddig, akár végtelenségig). $\mathbf{Z}^+$-szal a pozitív egész számok halmazát jelölöm.

A definíció nem zárja ki, hogy végtelen sok lehetséges modell legyen. Ha nem felsoroljuk őket, hanem más módon döntjük el, hogy adott információs állapotban mely modellek elfogadhatóak, akkor a modell megvalósításához sokkal közelebb jutunk. Eldöntésre megfelelő eszközök az automaták.

3.3. Modellhalmazokat felismerő automaták

Építsünk automatákat modellhalmazok felismerésére, mégpedig minél egyszerűbbet. Az automatáknak több osztályát szokták definiálni.[14] Ezek közül az egyik legegyszerűbb a véges állapotú automata. Célom az, hogy ha lehet, ezeknél bonyolultabb automatákat ne kelljen építeni.

Az automaták bemenete legyen a modell kódja. A modell kódja egy 0 és 1 számjegyekből álló számsorozat. A kód i. számjegye 1, ha az adott modell a p_i-vel jelölt nemlogikai konstansot igazra értékeli, különben 0. Tehát az ábécé, amin az automaták működnek: $\{0, 1\}$.

Az automatáknak többféle formális definíciója létezik.

[13] Kálmán-Rádai (2001: 88).

[14] Lásd például Bach (2001: 26–27).

Hagyományosan egy automata elfogad egy karaktersorozatot, ha azt végigolvasva elfogadó állapotba jut[15]. Elméletileg végtelen hosszú karaktersorozatok is létezhetnek (például a fenti modellkódok, ha **NLK** nem véges halmaz), bár értelmezésük problémás. Ez az automaták szempontjából is problémás, mert egy végtelen karaktersorozat végigolvasása nem lehetséges. Ezért jobb, ha **NLK** számossága bármilyen nagy, de véges, nem pedig végtelen.

Elméleti szinten nem szükséges kizárnunk a végtelen hosszú karaktersorozatokat, de ha ezeket is megengedjük, akkor kénytelenek vagyunk kicsit módosítani az automaták definícióját, és speciális esetekben eltekinteni a karaktersorozat végigolvasásától. Ez a speciális eset az lehet, amikor a karaktersorozat egy véges részét végigolvasva az automata olyan elfogadó vagy elutasító állapotba jut, hogy onnét tovább már csak önmagába visz átmenet – bármilyen karakter beolvasása mellett (esetünkben két karakterről van szó: 0 és 1). Nevezzük ezt az állapotot az **automata farká**nak. Ez azt jelenti, hogy ettől az állapottól kezdve mindegy, hogy mit olvas az automata, el fogja fogadni vagy el fogja utasítani. Ezt a módosítást elméleti megfontolásokból meg lehet tenni, de a gyakorlatban nem lesz rá szükség, mert nem találkozunk végtelen hosszú karaktersorozatokkal – ettől függetlenül az automatákban szerepelni fognak a farokállapotok.

Ha szigorúbban vesszük a definíciót, akkor tudnunk kell, hogy az automata végigolvasott-e egy karaktersorozatot. Ezt úgy tudjuk meg, hogy a karaktersorozat végén van egy speciális karakter, a „szó vége” jel. Jelölhetjük ezt a következő karakterrel: #. Ez is az olvasható elemek közé tartozik, de az ennek hatására definiálandó átmenetet mindig automatikusan megadhatjuk: elfogadó állapotból elfogadó állapotba, nem elfogadóból pedig elutasítóba visz.

Az automaták, amelyeket definiálni fogok, általában egyszerűsíthetőek (átalakíthatók kevesebb állapotot tartalmazó, de ugyanolyan karaktersorozatot elfogadó automatákká), de az áttekinthetőség miatt nem törekszem a legegyszerűbb alakra.

Az automatákban olvasás nélküli átmenetet is megengedek, de ezek az átmenetek szomszéd állapotainak összevonásával megszüntethetőek lennének.

Egyetlen kiinduló állapotot jelölök meg, melyet **S** betűjellel jelölök (*Start*). A következő automatadefiníciókban **S**-ből mindig olvasás nélküli

[15] Lásd Bach (2001: 30).

átmenet vezet a következő állapot(ok)ba, melyekkel elvileg összevonható lenne. Máshol nem fogok olvasás nélküli átmenetet alkalmazni.

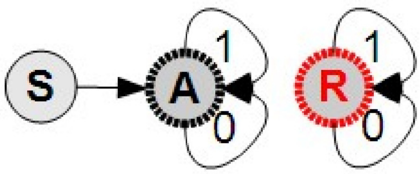
Feltételezek továbbá egy vagy több elutasító állapotot, jelük **R** (*Reject*). Ha többet is tervezek egy automatába, akkor is belátható, hogy ezek mind összevonhatók lennének egyetlen elutasító állapotba – csak a könnyebb áttekinthetőség miatt veszek fel többet, mert így az automatának a kiinduló állapottól eltekintve egymástól független „ágai" lesznek. Amely állapotból valamely karakter olvasása hatására nem vezetne él, ott feltételezem, hogy annak a karakternek a hatására az átmenet az (egyik) elutasító állapotba vezet. Az egyszerűség kedvéért, amikor később egy él törléséről vagy hiányáról fogok beszélni, azt fogom érteni rajta, hogy a hiányzó él az (egyik) elutasító állapotba vezet. **R** egy farokállapot.

A definiált automaták több elfogadó állapotot is tartalmazhatnak majd (általában **A**-val jelölöm őket: *Accept*). Az automatákat úgy definiálom, hogy az elfogadó állapotok is mindig farokállapotok, továbbá tudhatjuk azt is, hogy az elfogadó és az elutasító állapotokon kívül nincsenek az automatának farokállapotai.

Most nézzük, milyen automatákat érdemes definiálni az egyes modellhalmazok felismerésére.

Tegyük fel, hogy kezdetben a hallgatónak nincs előzetes információja, minden modell lehetséges a számára. Ehhez olyan automata tartozik, amely minden modellt elfogad.

A legegyszerűbb automatát (a_0), amely minden modellt elfogad, az 1. ábrán láthatjuk.



1. ábra: Az a_0 automata

A kiinduló állapotból egy átmenete van, az elfogadó állapotba, amely a korábbi definíció szerint egy farok – tehát bármilyen hosszú karaktersorozat következik az ábécé betűinek bármilyen sorrendjével, azt elfogadja. Említettem az elutasító állapotot, amelyet feltüntetek az automatában, de mivel a kiinduló állapotból nem vezet oda él, akár el is hagyhatnánk.

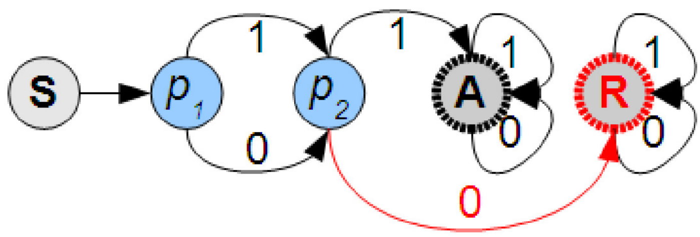
Az automaták további állapotai tulajdonképpen elemi tényállításokhoz rendelhetők. A belőlük kiinduló élek pedig azt határozzák meg, hogy annak az atomi formulának milyen értékelése megengedett. Ha akár 0, akár 1 olvasása után tovább lehet haladni abból az állapotból egy nem elutasító állapotba, ez az állapot nem szűr ki modelleket. Ha csak az egyik átmenet található meg, akkor ennek az atomi formulának csak egy bizonyos értékelését adó modelleket fogja elfogadni. Ha egyetlen él sem vezet belőle tovább, az egyfajta ellentmondásosságra utal. Ezt a képet a diszjunkció modellezése tovább fogja árnyalni.

Annak a folyamatát szeretném modellezni, ahogy egy dialógusban a hallgató információs állapota változik. Mielőtt a formális definíciókkal haladnék, előbb szemléltetni szeretném az elképzelésemet.

Ha a beszélő megemlít, kijelent, állít valamit, ami (elemi) tényállításnak felel meg, a hallgató információt szerez, növekednek az ismeretei – tulajdonképpen megjegyez magának valamit. Ha az automatás megközelítésből indulunk ki a hallgató információs állapotának modellezésénél, eszünkbe juthat, hogy vegyünk fel az újonnan hallott tényállításnak egy új állapotot az automatában, amiről valamiképpen meg kell jegyeznünk, hogy melyik tényállításhoz tartozik. Elnevezhetjük például az atomi tényállításhoz tartozó atomi formuláról (legyen ez most p_2). Sajnos a véges állapotú automaták működését csak az élek és állapotok sorrendjei befolyásolják, az állapotok „címkéi” nem.

Az automatának a modellek kódjait kell (egy darabig) elolvasnia. Ha egy formulát feldolgozunk, akkor az automatát úgy kell módosítanunk, hogy minden ágának tartalmaznia kell minden elemi tényálláshoz tartozó állapotot: sorban, egészen a formulában szereplő legnagyobb sorszámú atomi formuláig, mégpedig p_1-től kezdve. Ilyen módon mindig véges állapotú automatát kapunk eredményül, ha a_0-ból indulunk ki. Ezt a módszert alkalmazva megjelennek olyan elemi tényállításokhoz tartozó automataállapotok, melyekről szó sem volt a dialógusban. Ez ugyan ellentmond az intuíciónknak, de azt a helyzetet tükrözi, hogy míg az emberi elme tetszőleges sorrendben hozzá tud férni az elemi tényállítások értékeléséhez, addig a véges állapotú automatákkal történő modellezés nem. Csak sorban.

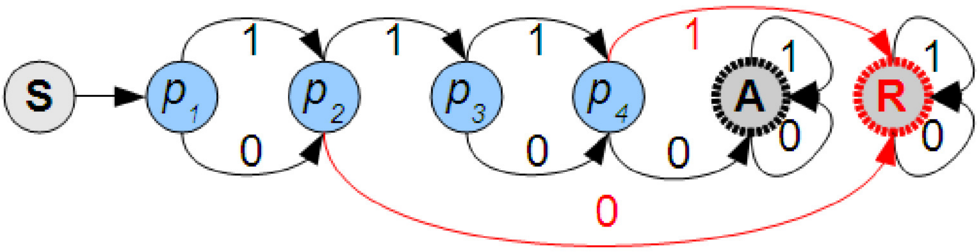
Visszatérve p_2-höz: a tudatlan információs állapothoz tartozó automatát, tehát a_0-t kell módosítani. Eredményét a 2. ábra mutatja.



2. ábra: A tudatlan információs állapot atomi formula által bővítve

Ez az automata minden modellt elfogad, amely p_2-t igazra értékeli. Nekem ez a modell azért tetszik, mert jól megragadja azt az intuíciónkat, hogy a hallott állítás növeli az információnkat, a tudás bővül az elménkben. A kezdeti információs állítás pedig egyszerű, tulajdonképpen üresnek mondható valami. Hagyományosan a dinamikus szemantikai elméletek információsállapot-modelljei szembe mentek ezzel az intuícióval. A tudatlan információs állapotot úgy modellezik, hogy abban minden lehetséges modell benne van[16] (tehát ez a „legnagyobb" információs állapot), majd a hallottak ezt a halmazt szűkítik, modelleket „eliminálnak" belőle, hogy egyre „kisebb" információs állapotokat kapjunk. Ez a hozzáállás is magyarázható, hiszen ahogy nő az információnk, úgy tudunk egyre több lehetőséget kiszűrni. Ettől függetlenül a szemlélete (számomra legalábbis) nem intuitív.

Ha tovább folytatjuk az automaták szemléltetését, $p_2 \wedge \neg p_4$ (vagy p_2 után egy következő mondat értelmezése, mint $\neg p_4$) eredménye a 3. ábrán látható automata.



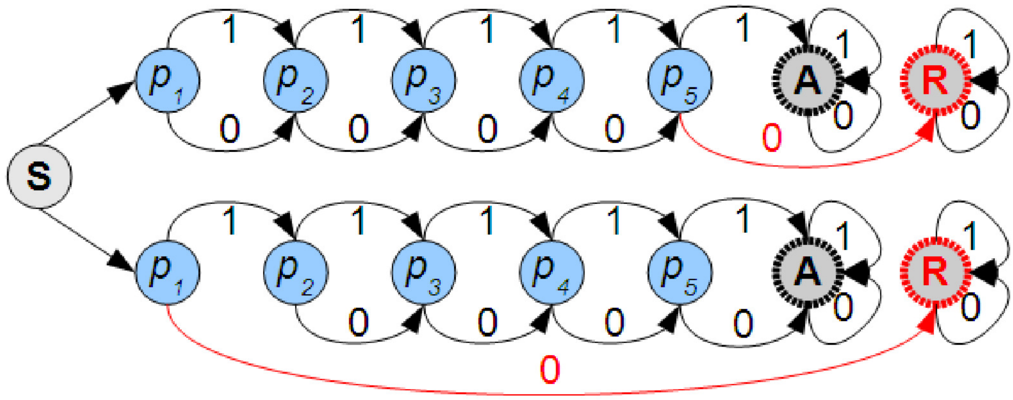
3. ábra: Tudatlan állapot tovább bővítve

A konjunktív formulák hatásának bemutatása után[17] áttérek a diszjunktív formulákra.

A tudatlan információs állapot után a $p_5 \vee p_1$ formula a 4. ábrán látható automatát eredményezhetné.

[16] Lásd Kálmán – Rádai (2001: 87).

[17] Egy dialógusban az egymás után elhangzó állítások kapcsolatát konjunktívnak lehet tekinteni – a sorrend természetesen nem minden esetben mindegy!



4. ábra: Diszjunktív formula hatása

Jól látszik, hogy a kiinduló állapotból két él vezet két ágához az automatának. Ezek gyakorlatilag egymástól függetlenek. Az egyik ág az egyik atomi formulára ad megkötést, a másik pedig a másikra. Megengedi mindkettő megkötés fennállását, de ha mindkettő sérül, akkor az automata mindenképpen elutasító állapotba jut. Érdemes megfigyelni, hogy a diszjunkció hagyományos értelmezése, igazságfeltétele szerint működik az automata, vagyis csak azokat a modelleket utasítja el, amelyeket mindkét ág elutasít – de elfogadja azokat, amelyeket bármelyik ága elfogad – az egyes ágaknak pedig formulák feleltethetők meg. További érdekesség, hogy az automata ágai azoknak az alternatíváknak felelnek meg, amelyek az *inquisitive semantics*[18] szerinti alternatívái a diszjunkciónak.

Következő teendőnk, hogy a_0-ból kiindulva definiáljuk, hogy mely formulák, formulatípusok hogyan változtatják meg az automatát. Így majd tetszőleges formula hatását meg tudjuk határozni.

3.4. Kibővített információt tartalmazó, táblázatos ábrázolás

Ahhoz, hogy általánosságban tudjak beszélni az automatákról, táblázatos módon fogom ábrázolni őket. Nem a hagyományos módon,[19] mert itt egy igen kicsi, kételemű ábécéről van szó. A táblázat sorai az automata „ágait" fogják tartalmazni. Nem ábrázolom a táblázatban a kiinduló állapotot és az elutasító állapotot, továbbá az elfogadó állapotokat sem. Alapvetően az atomi formuláknak megfelelő oszlopok lesznek a táblázatban, p_1-től sorban, sorszám szerint – kivéve az első néhány oszlopot, amelyekben az ágakra vonatkozó adminisztratív adatok lesznek. Ilyen adat lesz például az ág sorszáma, 1-től kezdve, egyesével növekedve. Az egyes nem adminisztratív oszlopok-

[18] Groenendijk – Roelofsen (2009).

[19] Bach (2001: 31), 2.1 ábra.

ban lévő cellák automataállapotok közötti átmeneteket határoznak meg. Két számjegy szerepelhet bennünk: 0 és 1. Amelyik szerepel benne, azt a karaktert olvasva az automata a tőle jobbra lévő állapotba jut (a jobb szélső oszlopok jobb oldali szomszédja egy elfogadó állapot, melyet nem jelölök a táblázatban – ez egy farokállapot). Ha a 0 és 1 számjegyek közül valamelyik nem szerepel a cellában, azt jelenti, hogy az érintett állapotból ezt a karaktert olvasva az (ághoz tartozó) elutasító állapotba jutunk.

Nézzünk egy konkrét példát! A 4. ábrán szereplő automata ($p_5 \vee p_1$) táblázatos leírását láthatjuk az 5. ábrán.

Sorszám	p_1	p_2	p_3	p_4	p_5
1	0, 1	0, 1	0, 1	0, 1	1
2	1	0, 1	0, 1	0, 1	0, 1

5. ábra: $p_5 \vee p_1$ táblázatos ábrázolással

A táblázatban az automata aktuális állapotán kívül a történetét is tárolhatjuk, ami később bizonyos feladatokhoz még jó lehet[20]. A diskurzus története ebben az esetben nem a mondatoknak megfelelő formulákat, hanem inkább azok szemantikai tartalmát tartalmazza[21].

A történet meghatározásához szolgáltat adatot az „*ős*” oszlop (minden ághoz megadja, hogy melyik másik ágból származtattuk, illetve 0, ha nem volt ilyen). Ahhoz, hogy valóban tudjuk az ősöket, szükség van egy új oszlopra is, amely megadja, hogy az illető sorra szükség van-e még az automatában, vagy sem. Legyen a címkéje „*Él?*”, értéke pedig 0 vagy 1, hasonlóan az igazságértékek kódolásához. Ezekkel az új, adminisztratív adatokkal a 6. ábrán látható táblázatot kapjuk a ($p_8 \wedge (p_5 \vee p_1)$) formulasorozat hatására.

Sorszám	Él?	Ős	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
1	0	0	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	1
2	1	1	0, 1	0, 1	0, 1	0, 1	1	0, 1	0, 1	1
3	1	1	1	0, 1	0, 1	0, 1	0, 1	0, 1	0, 1	1

6. ábra: Az automata ágainak történetét is megjelenítő táblázat

Látható, hogy a táblázat több adatot tartalmaz, mint amennyit a hozzá tartozó automata. Meg fogom adni, hogyan lehet a táblázat adataiból egyértelműen generálni az automatákat.

[20] Az 5. fejezet fogja indokolni ezeknek az adatoknak a tárolását, de a cikk azt megelőző részeiben leírt információsállapot-kezeléshez nem szükségesek.

[21] Többféle formulával meg tudunk fogalmazni hasonló szemantikai tartalmat. Gondoljunk akár csak az olyan egyszerű esetekre, mint például A és ($A \wedge A$).

Most pedig definiálom, hogyan alakítják a különböző formulák a táblázat tartalmát.

7. Definíció: A kiinduló táblázat

Kezdetben üres táblázatunk van, melynek három oszlopa van, a három adminisztratív oszlop, de nincs „tartalmas” sora, csak „fejléce” a táblázatnak. Jelöljük T_0-val. A 7. ábrán látható.

Sorszám	Él?	Ős

7. ábra: Diskurzuskezdő táblázat

Ez az a_0 automatának felel meg, amely az 1. ábrán szerepelt.

8. Definíció: Atomi formula (p_i) hatása a táblázatra

a. Oszlopok módosítása

Ha a táblázatnak már van olyan oszlopa, melynek fejlécében p_i szerepel (korábban már volt p_i-re vagy nála nagyobb indexű elemi tényállításra vonatkozó információ), akkor nincs teendő az oszlopokkal.

Ha a táblázatnak még nincs olyan oszlopa, melynek fejlécében p_i szerepel, akkor ki kell bővíteni a táblázatot úgy, hogy p_1-től kezdve p_i-ig minden elemi tényállításnak legyen oszlopa. Ha a táblázatnak csak adminisztratív oszlopai vannak, akkor a következő oszlop p_1 oszlopa lesz, fejlécébe p_1 lesz írva. Ettől az oszloptól kezdve folyamatosan egyesével növekvő indexű elemi tényállításokhoz tartozó oszlopokat lehet hozzáadni.

b. Sorok módosítása

Ha a táblázatnak még nincsenek tartalmas sorai, csak fejléce, akkor fel kell venni egy új sort. Sorszáma 1 lesz, az „*Él?*” oszlopba 1-et kell írni, az „*Ős*” oszlopba pedig 0-t. A további „tartalmas” oszlopokban minden cellába azt kell írni, hogy „0, 1”, végül a p_i oszlopból ki kell törölni a 0-t (csak az 1-es marad benne).

Ha a táblázatnak vannak tartalmas sorai, akkor azok közül azokat kell lemásolni és új sorként beszúrni a táblázat alján, amelyek élők. Sorszámuk eggyel nagyobb kell, hogy legyen, mint a fölöttük lévő sor sorszáma. Élőnek kell megjelölni őket, őseik pedig azok a sorok lesznek, amelyeknek a másolatai. Az őseiket nem élőnek kell megjelölni.

Az így kapott táblázatban már csak az élő sorokat módosítjuk. Ha bennük p_i oszlopában még nincs adat, akkor a sor összes új, üres celláját kitöltjük „0, 1”-gyel. Végül ha a p_i oszlopban szerepel 0, akkor azt töröljük onnét.

Előfordulhat azonban, hogy ezek után valamelyik élő sorban lesz olyan cella, ami üres. Ez azt jelenti, hogy a táblázathoz tartozó automatának ez az ága több darabra bomlik, lesz egy elérhetetlen darabja – az eleje pedig mindenképpen elutasító állapotba visz. Ez az ágnak az ellentmondásosságát mutatja, aminek a későbbi műveletek végzésekor, vagy a táblázat egészének kiértékeléskor van csak jelentősége.

Elvileg megkülönböztethetnénk az ellentmondás hatására kiürülő és az eleve üres cellákat. Gyakorlatilag erre nincs szükség, mivel üres cellákat élő sorokba csak átmeneti jelleggel szúrunk be, amíg teljesen fel nem dolgoztuk a formulát. Ezek mindig a sorok végén keletkeznek, és a feldolgozás végére tartalmuk is megjelenik – vagy tényleg ellentmondást jeleznek. A nem élő sorokban maradhatnak olyan üres cellák, melyek nem jeleznek ellentmondást, de azok a kiértékelésnél nem játszanak szerepet.

9. Definíció: Tagadott atomi formula ($\neg p_i$) hatása a táblázatra

Ugyanúgy járunk el, mint tagadás nélkül, de ahol p_i oszlopából a 0-t törölnénk, ott most az 1-et töröljük.

10. Definíció: Konjunktív formula ($A \wedge B$) hatása a táblázatra

Először A-t alkalmazzuk, majd B-t.

11. Definíció: Tagadott tagadott formula ($\neg\neg A$) hatása a táblázatra

A dupla negációt elhagyjuk, A hatását vizsgáljuk tovább.

12. Definíció: Diszjunktív formula ($A \vee B$) hatása a táblázatra

a. Oszlopok módosítása

Megnézzük, hogy A-ban és B-ben melyik a legnagyobb indexű atomi formula, jelöljük most ezt az indexet i-vel. Ha a táblázatban még nincs annak megfelelő oszlop, akkor a táblázatot kibővítjük, hogy p_i-ig tartalmazzon oszlopokat.

b. Sorok módosítása

Ha a táblázatnak még nincsenek tartalmas sorai, akkor először A hatására veszünk fel sorokat, azok minden celláját kitöltjük. Utána kitöltünk egy ugyanilyen oszlopokat tartalmazó táblázatot B hatására. Végül a B által meghatározott táblázat sorait bemásoljuk abba a táblázatba, amelyben az A hatására keletkezett sorok vannak, de a bemásolt sorok sorszámát, valamint az ős oszlopban lévő számokat megnöveljük a táblázatban eredetileg benne lévő (A hatására bekerült) sorok számával a folyamatos sorszámozás érdekében (kivéve, ahol az ős 0, azt nem változtatjuk meg).

Ha a táblázat már tartalmazott sorokat, akkor először készítünk egy másolatot a táblázatról, majd az eredeti táblázaton végrehajtjuk azokat a módosításokat, amelyeket az *A* (elemi vagy komplex) formula hatására kell elvégezni, a másolaton pedig azokat, amelyeket a *B* formula hatására végeznénk el. Végül a másolat-táblázatból azokat a sorokat, amelyek nem elemei az eredeti táblázatnak (az *A* formula feldolgozása előtti állapotban), visszamásoljuk az eredeti táblázat meglévő sorai alá úgy, hogy a sorszámok folyamatosak legyenek, és az *Ős* oszlop elemei is megfelelő értékűek legyenek.

13. Definíció: Tagadott diszjunktív formulák ($\neg(A \vee B)$) hatása a táblázatra

Hasonlóan a tagadott konjunktív formulákhoz, itt is alkalmazzuk a *de Morgan* azonosságot, az eredeti helyett a következő formula hatását vizsgáljuk: ($\neg A \wedge \neg B$)

14. Definíció: Tagadott konjunktív formula ($\neg(A \wedge B)$) hatása a táblázatra

A *de Morgan* azonosságot használva átalakítjuk: ($\neg A \vee \neg B$). Most már alternatív formulaként kell kezelnünk.

Ezzel végeztünk is a formulák áttekintésével. Mivel célunk az, hogy automatákat használjunk, meg kell adnom az algoritmust, amely a táblázatoknak megfelelő automatákat meghatározza. Ez a következő:

15. Definíció: Algoritmus automaták generálására a táblázatok alapján

a. Teendők

A továbbiakban a táblázatnak csak az „élő” soraival foglalkozunk.

- Vegyünk fel egy kezdőállapotot.
- A táblázat élő sorainak minden (nem adminisztratív oszlopában lévő) cellájához vegyünk fel egy automataállapotot.
- A táblázat minden élő sorához vegyünk fel egy elutasító és egy elfogadó állapotot.
- A kiinduló állapotból vigyen olvasás nélküli átmenet minden élő sor első cellájának megfelelő állapothoz.
- A táblázat élő soraiban lévő celláknak megfelelő állapotokból vigyen átmenet a tőlük eggyel jobbra lévő cellához tartozó állapothoz, mégpedig olyan karaktert olvasó átmenetekkel, amelyek a cellában fel vannak sorolva (lehetőségek: 0 és 1)[22]. A fel nem sorolt lehetőségeknek megfelelő átmenetek az elutasító állapotba vezetnek.
- Az elfogadó állapotokból vezessen átmenet önmagukba 0-t vagy 1-et olvasva is.

[22] Egy élő sor utolsó cellájának jobb szomszédja a sor elfogadó állapota.

– Az elutasító állapotokból szintén vezessen átmenet önmagukba 0-t vagy 1-et olvasva.

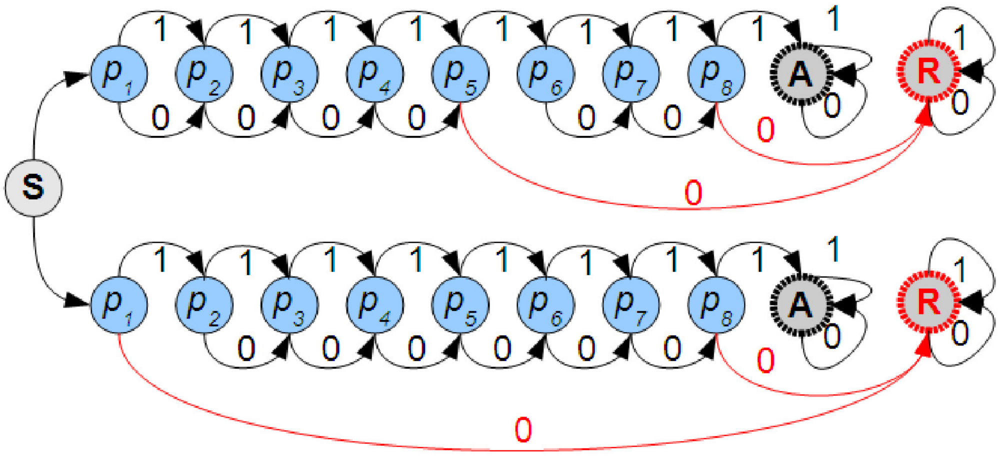
Ezzel kész is a táblázatnak megfelelő automata.

b. Megjegyzések

Továbbra is véges állapotú automatákról lesz szó.

Ha a táblázatnak csak egy élő sora van, akkor determinisztikus véges állapotú automatát kapunk. Ha több, akkor nemdeterminisztikus lesz az automata. Ez tulajdonképpen a diszjunkciónak köszönhető.

A jobb érthetőség kedvéért egy ilyen módon, a 6. ábra táblázata alapján megalkotott automatát mutat be a 8. ábra.



8. ábra: Táblázat alapján készített automata

Látható, hogy az automatát generáló algoritmus nem bonyolult, majdhogynem visszafelé is alkalmazható, vagyis automata alapján is generálható táblázat. Az eredeti és a „visszafejtett” táblázat között az lesz a különbség, hogy ez utóbbi nem fogja tartalmazni az eredeti adminisztratív oszlopainak adatait és a nem élő sorokat sem.

4. Központi szemantikai fogalmak

Néhány fogalmat definiálni kell ahhoz, hogy a szemléleten túl a rendszer logikai tulajdonságairól is tudjunk mondani valamit. A szükséges definíciók a következők (az eddigiekhez hasonlóan inkább szabadszavas definíciók következnek, mint formulák):

16. Definíció: Összeférhetőség

Egy σ információs állapot akkor és csak akkor fér össze az A formulával, ha a formulát alkalmazva az információs állapotra olyan automatát kapunk, amelyben a kezdőállapotból legalább egy elfogadó állapotba el lehet jutni átmenetek sorozatán keresztül.

17. Definíció: Összeférhetetlenség
Az összeférhetőség ellentéte; vagyis egy σ információs állapot akkor és csak akkor összeférhetetlen az A formulával, ha a formulát alkalmazva az információs állapotra olyan automatát kapunk, amelyben a kezdőállapotból egyetlen elfogadó állapotba sem lehet eljutni átmenetek sorozatán keresztül.

18. Definíció: Alátámasztás
A σ információs állapot akkor és csak akkor támasztja alá az A formulát, ha a σ-hoz tartozó automata ugyanazokat a modelleket (modellkódokat) fogadja el, mint az az automata, amely ahhoz az információs állapothoz tartozik, melyet úgy kapunk, hogy A-t alkalmazzuk σ-ra.[23] (Az alátámasztás intuitívan azt jelenti, hogy A nem ad új információt σ-hoz, mert az A formula információtartalmát σ már magában hordozza.)

19. Definíció: Következmény
Az A_1, A_2, …, A_n formuláknak akkor és csak akkor következménye a B formula, ha minden σ információs állapotra igaz, hogy ha σ-ra alkalmazzuk A_1-et, majd ennek eredményére A_2-t és így tovább A_n-ig sorban, az eredményül kapott információs állapot alátámasztja B-t.

5. Kitekintés

Röviden kitérek arra, hogy az információs állapotok ábrázolásának ezt a módját választva milyen kapcsolatok alakulhatnak ki egyéb szemantikai elméletekkel.

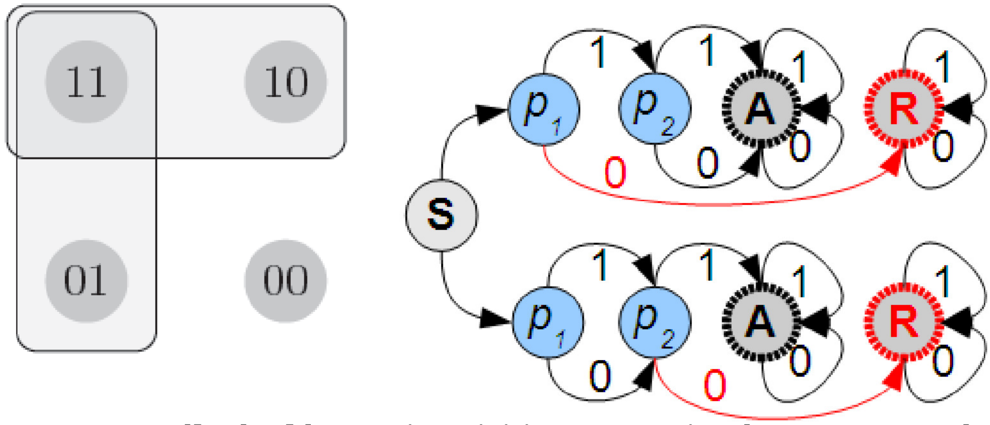
5.1. Inquisitive semantics[24]

A „kíváncsi szemantika” megkülönbözteti a nyelvi kifejezések informatív és kérdő tartalmát.[25] Az informatív tartalom a még lehetségesnek tartott modellek közül kiszűr valamennyit, a kérdő tartalom pedig a lehetséges modelleket

[23] Megjegyzem, hogy az itt közölt definíció számítógépes alkalmazásra ilyen formában használhatatlan. Konkrét megvalósítás előtt érdemes átfogalmazni úgy, hogy az érintett információs állapotok akár táblázatos, akár automatás ábrázolásainak formai tulajdonságait vegye csak figyelembe.

[24] Az *inquisitive semantics* alapjairól Groenedijk – Roelofsen (2009)-ben lehet olvasni. További olvasmányok elérhetők a következő honlapon: *http://sites.google.com/site/inquisitivesemantics/*

alternatívákra (csoportokra) osztja, melyek át is fedhetik egymást. (Ez a fő újdonsága a kérdések szemantikájával foglalkozó korábbi elméletekkel szemben, melyek diszjunkt partíciókkal magyarázták a kérdésekkel és válaszokkal kapcsolatos jelenségeket.[26]) A kérdések mellett a diszjunkció is ilyen módon viselkedik.



9. ábra: $p_1 \vee p_2$ ábrázolása az *inquisitive semantics* és az automatás megközelítés alapján

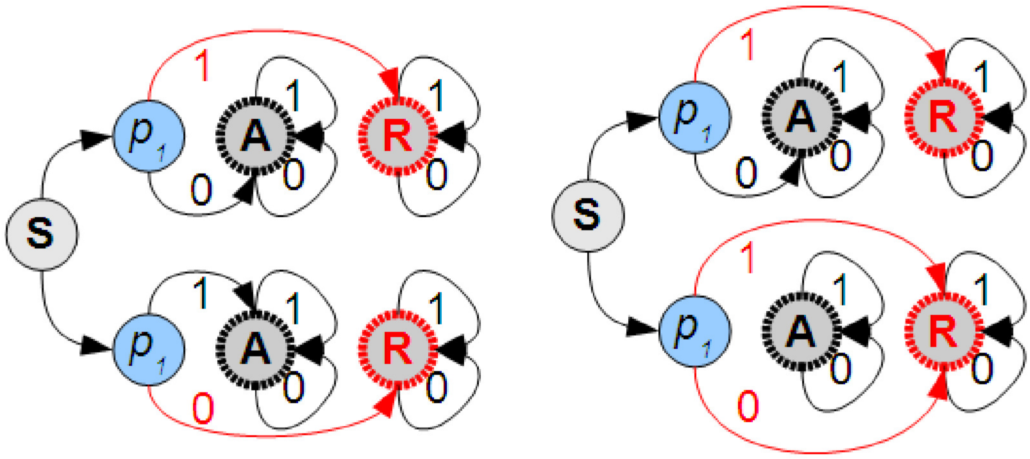
Jól látható, hogy a két alternatíva megfelel az automata két ágának. Mindkettő kizárja azokat a modelleket, amelyek p_1-et és p_2-t is hamisra értékelik – a többit elfogadják, de nem mossák össze a két alternatívát.

A kérdések szemantikája csak a válaszokkal együtt érdekes. Ha valaki feltesz egy kérdést, ezzel megfogalmazza az alternatívákat, amelyek közül a kimerítő válasz egyet hagy csak meg. (Részleges válasz esetén csak csökkenti a számukat.)

Nézzük a következő kérdést: $p_1?$, ahol a kérdés operátort láthatjuk, amelyet úgy is átfogalmazhatunk most, hogy $p_1 \vee \neg p_1$. A 10. ábra mutatja a kérdéshez tartozó, és a kérdésre adott $\neg p_1$ válasznak megfelelő automatát.

[25] Az eredeti angol kifejezések: *informative* és *inquisitive*. Ebben a kijelentéslogikai rendszerben a kérdések mindig csak eldöntendőek.

[26] Többek között példa erre Groenendijk – Stokhof (1997).



10. ábra: p_1? kérdés automatája és ugyanez a $\neg p_1$ válasszal módosítva

Látható, hogy a válasz után az alsó ágban nem vezet egy átmenet sem az elfogadó állapotba (ellentmondásosnak is tekinthető), így csak a felső ág fogad el bármit is, az határozza meg a választ.

Az előző példa nagyon egyszerűen mutatja be a kérdések és a partíciók megfelelőit az automatákban. Valójában egy automatának nem csak két ága lehet. „p_1?” után elhangozhat egy másik kérdés: „p_2?”, és máris négy ága van az automatának. A második kérdés megválaszolása után az első kérdés még nyitva lehet, megengedve több „működő” ágat (olyan ágat, amelyben az ág elfogadó állapotába a kiinduló állapotból vezet út). Láthatjuk, hogy egy alternatívának több ág is lehet a tagja. Ezt az eddig rajzolt automatákkal nem tudjuk megragadni, de segítséget nyújt ebben a táblázatos ábrázolásmód, amely nyilvántartja, hogy mely ágakat (ezek a táblázat egyes sorainak felelnek meg) mely ágakból származtattuk. Amikor ugyanis például diszjunkció (vagy kérdés) hatására új ágakat vezetünk be az automatába, az új ágak egy korábban meglévő ágnak a másolásával és módosításával keletkeznek. Ha több ág is van, amelynek őse ugyanaz a „régi” ág, akkor ezek az ágak magukban jelenítenek meg alternatívákat. Ha vannak ágak, amelyeknek az őse nem ugyanaz, akkor a közös ősű ágak halmazai felelnek meg egy-egy alternatívának. Tehát az „ősök” meghatározzák az alternatívákat.

5.2. A hit-felülvizsgálati elméletek

Az információs állapotokat modellhalmazokként kezelő egyszerű felfrissítő szemantika[27] nem képes arra, hogy ellentmondásos állapot után az ellentmon-

[27] Kálmán – Rádai (2001: 88).

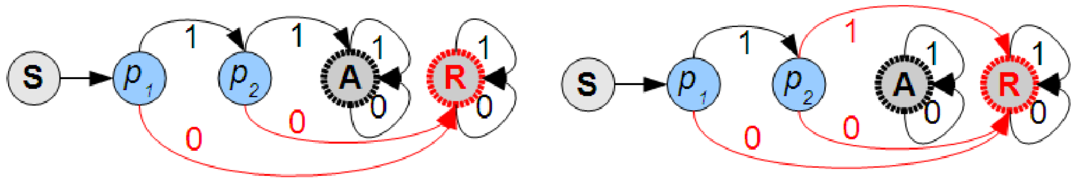
dást okozó állítás visszavonása által kikerüljön az ellentmondásos állapotból[28].

A hit-felülvizsgálati elméletek formulahalmazokon definiálják a visszavonás műveletét. Ez az információs állapotok automatás ábrázolásával dinamikus szemantikai keretben is megtehető. A visszavonás bármikor megtehető, de kiemelt jelentősége van az ellentmondásos állapotból való kikerülésnél, ahogy erről korábbi cikkemben is írtam. Ha az általam leírt módon, automatákkal ábrázoljuk az információs állapotokat, akkor az ellentmondásos állapot arról ismerhető fel, hogy egyik elfogadó állapotba sem visz út a kezdőállapotból, vagyis minden karaktersorozat beolvasásának hatására elutasító állapotba jutunk. Táblázatos ábrázolásnál ez úgy jelentkezik, hogy minden élő sorban lesz legalább egy üres cella.

Atomi formula visszavonása esetén egyszerű dolgunk van, mert ha visszavonunk valamit, akkor annak a tagadását állítjuk. Így tehát egy atomi formulának megfelelő állapothoz hozzá kell adni egy átmenetet. Ezt minden élő ágban meg kell tenni. Ha így lesz az automatának működő ága, akkor kikerültünk az ellentmondásos állapotból.

Az összetett formulák visszavonása már komplikáltabb kérdés. Ha ragaszkodunk ahhoz, hogy csak olyan állítást lehessen visszavonni, amely elhangzott a dialógusban, akkor mindenképpen a táblázatos ábrázolást kell használnunk, mert az tartalmazza a diskurzus történetét, ezért korábbi állapotok is előhívhatók belőle. Mindez komolyabb átgondolást igényel. Részletkérdésnek is gondolhatjuk, de azt mindenképpen érdemes megjegyezni, hogy ez a fajta strukturált információsállapot-ábrázolás egyáltalán lehetővé teszi az ellentmondásos helyzetből való kikerülést és a visszavonást.

A visszavonás működését atomi formula esetén könnyen szemléltetni is tudom ábrákkal. Álljon a dialógus a következő formulasorozatból: <p_1; p_2>. Az ehhez tartozó automata látható a 11. ábra bal oldalán.

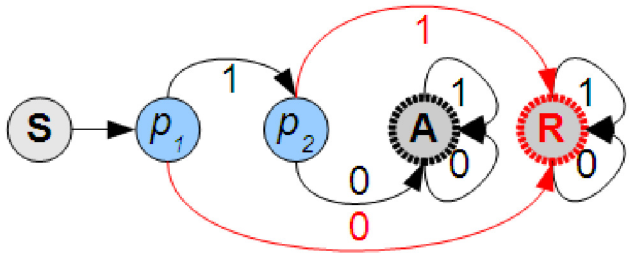


11. ábra: <p_1; p_2>, valamint <p_1; p_2; $\neg p_2$> ábrázolása automatával

Ha hozzáfűzzük, hogy $\neg p_2$, akkor a <p_1; p_2; $\neg p_2$> dialógust és a 11. ábra jobb oldali automatáját kapjuk. Látszik, hogy ellentmondásba ütköztünk, ez az automata nem fogad el semmilyen modellt.

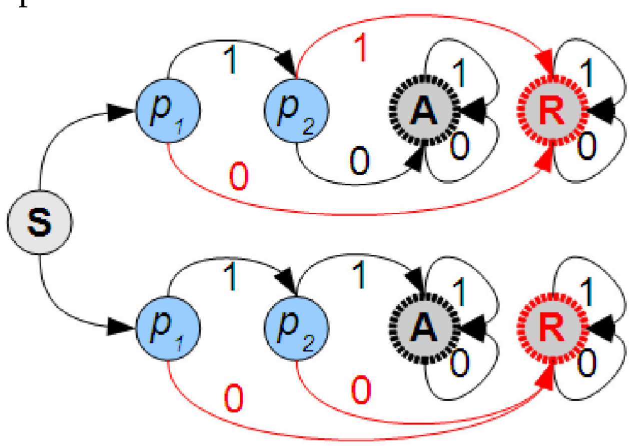
[28] Dyekiss (2010: 15).

Ha a beszélő most észbe kapna, és visszavonná egyik p_2-re vonatkozó állítását, akkor egyszerűen az egyik átmenetet újra be kéne húzni az automatában. Ha $\neg p_2$-t vonná vissza, akkor megkapnánk az eredeti automatát (a $\neg p_2$ állítása előttit), ha pedig p_2-t, akkor a 12. ábrán látható automatát kapnánk.



12. ábra: p_2 visszavonása utáni állapot

Ha pedig nem vonná vissza magától, de a hallgató most visszakérdezne, hogy a beszélő tulajdonképpen most mit gondol p_2 igazságértékéről, akkor a 13. ábra automatáját kapnánk.



13. ábra: Visszavonásra vonatkozó kérdés ábrázolása

Ha pedig elhangzana a válasz, akkor megkapnánk az egyik automatát a nemrég bemutatottak közül – attól függően, hogy p_2 mely értékelése mellett tette le a voksát a beszélő[29].

Összefoglalás

Cikkemben egy olyan kijelentéslogikára alapuló dinamikus szemantikai elmélet körvonalait vázoltam, melyben az információs állapotokat véges táblázatokkal, vagy ezek segítségével egyértelműen meghatározható véges álla-

[29] Az automatának mindkét ágát módosítani kellene a p_2-nek megfelelő állapotokból kiinduló átmeneteknél. Kérdés, hogy a visszavonásra való tekintettel szabad-e csak a működő ágakat megtartani, vagy mindent meg kell tartanunk.

potú automatákkal ábrázoljuk. Indokoltam, hogy miért jobb automatákkal ábrázolni az információs állapotokat, mint modellhalmazokkal.

Bemutattam, miért jobb és hogyan lehetséges az automatákat inkább modellek vizsgálatára használni, mintsem formulák ellenőrzésére. Definiáltam, hogy az egyes formulatípusok hogyan alakítják az információs állapotokat. Központi szemantikai fogalmakat is definiáltam.

Rövid kitekintést adtam az ismertetett elméletnek számomra fontosnak tartott szemantikai elméletekkel (*inquisitive semantics*, *belief revision*) való kapcsolatára.

További lehetőséget jelentenek a következők:

Az itt vázolt elméletet teljesen egzakt módon, formálisan is kellene definiálni, méghozzá a kérdések-válaszok, valamint az ellentmondások és a visszavonás szemantikájával együtt. Ha ez elkészül, akár egy programot is lehetne írni, ami demonstrálja az elmélet működését.

Még nagyobb lépést jelentene, ha mindezt nem kijelentéslogikai, hanem predikátumlogikai keretben tudnám megtenni, hiszen a dinamikus szemantikák legnagyobb előnyei ebben a környezetben jelennek meg.

Hivatkozások

Alchourrón, Carlos Eduardo – Peter Gärdenfors – David Makinson 1985. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic* **50**: 510–530.

Babcsányi István 2007. *Automaták, nyelvek, kódok*. Budapest, BME Matematika Intézet, Algebra Tanszék [http://www.math.bme.hu/~babcs/Automatak.html]

Bach Iván 2001. *Formális nyelvek*. Budapest, TypoTex [http://mek.oszk.hu/05000/05099/]

Dyekiss Emil Gergely 2010. Ellentmondások kiküszöbölése a diskurzusból kérdések segítségével. In Gécseg Zsuzsanna (szerk.) *LingDok 9. Nyelvész-doktoranduszok dolgozatai*. Szeged, JatePress. 9–32. [http://nydi.bibl.u-szeged.hu/SZTE_NYDI/LingDok_kotetek_files/lingdok9.pdf]

Eilenberg, Samuel 1974. *Automata, Languages, and Machines*. Academic Press, New York and London.

Groenendijk, Jeroen – Floris Roelofsen 2009. Inquisitive Semantics and Pragmatics. *Proceedings of the International Workshop on Semantics, Pragmatics and Rhetorics*, Donostia, Spain, May 6–8, 2009. [http://sites.google.com/site/inquisitivesemantics/documents/ISP-Stanford-edition.pdf?attredirects=0]

Groenendijk, Jeroen – Martin Stokhof 1997. Questions. In J. van Benthem – A. ter Meulen (eds.) *Handbook of Logic and Language*, Amsterdam/Cambridge (Mass.), Elsevier/MIT Press, 1055–1124. [http://staff.science.uva.nl/ ~stokhof/q.pdf]

Kálmán László – Rádai Gábor 2001. *Dinamikus szemantika*. Budapest, Osiris kiadó.