

Debreceni Egyetem
Informatikai Kar
Információ Technológia Tanszék, Számítógéptudományi Tanszék

Kooperatív játékok információhiányos környezetben
A Hanabi társasjáték digitalizálása

Dr. Aszalós László

Balázs Patrícia

Dr. Kósa Márk

Dr. Vaszil György, tanszékvezető egyetemi docens

Bírálatra való beküldéséhez hozzájárulok:

Debrecen

2022

Tartalomjegyzék

1. Téma felvetése	3
2. Célkitűzéseim	4
3. A téma elméleti háttere	5
3.1 A modális logika	5
3.2 A modális logikai kalkulus és a LoTREC rendszer	7
3.3 <i>Koszorú gyerekek</i> probléma	9
4. A Hanabi társasjáték	12
4.1 Szoftveres megvalósítás	14
4.2 Modális logikai kalkulus a Hanabi társasjátékhoz	14
4.3 Lépésajánláshoz használt algoritmusok.....	18
5. A Hanabi alkalmazás működése.....	22
5.1 Tapasztalatok a játék során elért pontszámokban	22
5.2 Mesterjelölt szint a Hanabi játékban	26
6. Létező implementációk	30
7. Összefoglalás	33
8. Irodalomjegyzék	35
9. Saját munka leírása.....	36

1. Téma felvetése

Napjainkban egyre nagyobb az igény az emberi kapcsolattartásra, hiszen az emberek egyre távolabb kerülnek egymástól, akár a felgyorsult világ, akár a járványhelyzet hatására. Ezáltal felértékelődik a távolsági kapcsolattartás szerepe. Ennek érdekében jó lehetőség, ha olyan programokat hozunk létre, amelyek segítségével az emberek egymástól távol is értékes időt tudnak együtt tölteni, szórakoztatni egymást és önmagukat, akár online, akár offline módon. A társasjátékok pedig, úgy gondolom, egyértelműen időtálló elemei a szórakoztató együttléteknek családi és baráti körökben egyaránt. Napjainkban számtalan fajtájuk – többek között stratégiai, családi, kártya, party, logikai, oktató vagy puzzlejátékok – közül választhatunk, tekintettel a játszani vágyók korára és preferenciáira.

Az elmúlt években egyre népszerűbbé váltak a kooperatív játékok, amelyekben a játékosoknak együtt kell működniük a közös győzelem érdekében. Ezeknek a játékoknak nagy része dedukciós játék, ahol valamilyen információhiányos környezetben kell a játékosoknak helytállniuk. Az ilyen játékokban számos lehetőség adódik különböző mesterséges intelligenciához és logikához kapcsolódó algoritmusok tesztelésére.

A modális logika a logika olyan bővítése, amely a klasszikus logikai kijelentések mellett modális kijelentéseket is tartalmaz. Ezáltal a világ jellemzésén túl, az abban történő változások leírására is képes. Éppen ezért a modális logikát számos tudományterületen alkalmazzák, a filozófiától a közgazdaságtanon át a kriptográfiáig a legkülönbözőbb tudományterületeken dolgozó kutatók mind azt találták, hogy a tudásról való érvelést érintő kérdések nagy jelentőséggel bírnak.

A modális logika sok területen való használhatósága miatt felmerült bennem a kérdés, hogy vajon mennyire alkalmazható egy általam kedvelt tudományterület, a társasjátékok, és azon belül is a kooperatív és dedukciós játékok területén, ahol az episztemikus modalitás eszközeinek használata mellett a komplexitással is meg kell küzdeni.

2. Célkitűzéseim

Kutatásom során azt elemeztem, hogy vajon az olyan kooperatív társasjátékok esetén, amelyek információhiányos környezettel rendelkeznek, a modális logika eszközei mennyire segítik a játék megnyerését. A kooperatív játékok egyre népszerűbbé válása révén a dedukciós környezettel rendelkező változatokból is rendre jelennek meg újabb játékok. Ide sorolható a *Beyond Baker Street: Sherlock Holmes árnyékában* című társasjáték, a *Titkos szavak*, valamint az általam választott és a kutatásom alapjául szolgáló *Hanabi* társasjáték is.

Vizsgálataim során a választott játékkal kapcsolatban egy olyan alkalmazást készítettem el, amelynek segítségével az emberek és a gépi ágensek együttműködve játszhatnak végig egy-egy játszmat. Az ágensek a meglévő ismereteikből következtetnek ki újabb állításokat a modális logika és az automatikus tételbizonyítás felhasználásával, a szóba jöhető lehetőségek közül pedig eltérő valószínűségekkel próbálnak meg választani.

Kutatásom célja az volt, hogy megvizsgáljam a játékelmélet és a mesterséges intelligencia módszereinek felhasználásával, hogy az ágensek milyen eredményeket képesek elérni a játék folyamán különböző szempontok figyelembevételével. Azt a hipotézist szerettem volna megerősíteni vagy cáfolni, hogy az ilyen típusú társasjátékok esetén fontos szerepet játszanak a modális logika kijelentései. Ennek tanulmányozására összevetettem a játék sikerességét azokban az esetekben, amikor a lépéseket az ágensek teljesen véletlenszerűen választják azokkal a változatokkal, amikor valószínűségszámítással kapcsolatos kalkulációkat is figyelembe vesznek, illetve amikor a modális logika eszközei is beépülnek a játék menetébe, ezzel növelve az ágensek tudásbázisát.

Mindezek mellett azt vizsgáltam, hogy az általam létrehozott alkalmazás milyen eredményeket képes elérni más, online felületen elérhető megvalósításokhoz képest, illetve milyen eredményeket képesek elérni ezek a programok közösen játszva. Továbbá azt is elemeztem, hogy a különböző programokkal való játék során a kezdő és a már tapasztaltabb emberi játékosok mennyi pontot tudnak összegyűjteni.

3. A téma elméleti háttere

3.1 A modális logika

Az episztemikus modalitás a klasszikus logikai kijelentéseken túl modális kijelentéseket egyaránt tartalmaz, így a világ jellemzésén túl az abban történő változások leírására is alkalmas. Az episztemikus modalitás a tudásról szóló állítások logikája, ahol az ágensek a világról és más ágensek tudásáról is képesek gondolatokat megfogalmazni. Ez a fajta gondolkodás, amely a Sherlock Holmes rejtvények gyakori szereplője, számos kontextusban bír nagy jelentőséggel.

Az ismeretelméletnek, a tudás tanulmányozásának hosszú és tiszteletreméltó hagyománya van a filozófiában. Olyan kérdések, mint például a „Mit tudunk?“, „Mit lehet tudni?“ és „Mit jelent azt mondani, hogy valaki tud valamit?“ sokszor jelennek meg a filozófiai irodalomban. A tudással kapcsolatos érvelés formális logikai elemzésének gondolata valamivel újabb keletű, egészen *Georg Henrik von Wright* 1950-es évekbeli munkásságáig nyúlik vissza. Az 1960-as években felerősödött az érdeklődés ezen a területen a filozófiai közösségekben. A fő figyelem a tudásban eredendő tulajdonságok megismerésére irányult, a tudás axiómáit magyarázták, támadták és védték.

Az utóbbi időben a közgazdaságtan, a nyelvészet, a mesterséges intelligencia és az elméleti számítástechnika kutatói váltak egyre érdeklődőbbé a tudásról való érvelés iránt. A filozófiai kérdések helyett olyan problémák kerültek középpontba, mint hogy mit kell tudnia egy robotnak ahhoz, hogy kinyisson egy széfet, és honnan tudja, hogy eleget tud-e a kinyitáshoz, vagy hogy mikor tud egy gazdasági szereplő eleget ahhoz, hogy abbahagyja az információgyűjtést és döntést hozzon. A kutatások során ugyanakkor a hangsúly áttolódott olyan esetek vizsgálatára, amelyek egyetlen ágens helyett ágenscsoportok ismereteit foglalják magukba.

Egy csoportban lévő ágensnek nemcsak a világra vonatkozó ismereteket kell figyelembe vennie, hanem a csoport többi szereplőjének tudását is. Például alkuhelyzetben az autó eladójának mérlegelnie kell, hogy a potenciális vásárló mit tud az autó értékéről, a vevőnek pedig azt is figyelembe kell vennie, hogy mit tud az eladó arról, amit a vevő tud az autó értékéről, és így tovább. Az ilyen tudás meglehetősen bonyolult lehet. A legtöbb ember

gyorsan elveszti az ilyen egymásba ágyazott mondatok fonalát, de pontosan az ilyen típusú tudás az, amire szükség van az egy csoportban lévő ágensek tudásának elemzéséhez.

Számos olyan tudás létezik többszereplős helyzetben, amelyek az egy ágenses állapotokban nem jönnek létre. Ezek olyan helyzetek, amikor a csoport minden tagja ismeretében van egy állításnak. Például egy társadalom minden bizonnyal azt akarja, hogy minden járművezető tudja, hogy a piros lámpa azt jelenti, „állj”, a zöld lámpa pedig azt, hogy „mehetsz”. Tegyük fel ugyanakkor, hogy a társadalom minden járművezetője ismeri ezt a tényt, és betartja a szabályokat. A sofőr ekkor sem érezheti magát biztonságban, kivéve, ha azt is tudja, hogy mindenki más is ismeri és betartja a szabályokat. Ellenkező esetben a sofőr elképzelhetőnek tarthatja, hogy bár ő ismeri a szabályokat, néhány másik sofőr nem, és ezért áthajthat a piros lámpán.

Vannak olyan esetek, amikor nem elegendő az a tudás, hogy mindenki tudja, hogy mindenki tud valamit. Bizonyos esetekben figyelembe kell venni azt az állapotot, amikor mindenki egyszerre tud egy φ tényt, mindenki tudja, hogy mindenki tudja φ -t, és így tovább. Ebben az esetben azt mondhatjuk, hogy a csoport számára φ közös tudás. A közös tudás különböző diskurzusok megértésének esetében is szerepet játszik. Tegyük fel, hogy Aliz megkérdezi Balázstól, hogy „Mit gondoltál a filmről?”, utalva egy most látott filmre. Aliznak és Baláznak nemcsak tudnia kell, hogy „a film” kifejezés melyik filmre vonatkozik, hanem Aliznak tudnia kell, hogy Balázs tudja, és Baláznak tudnia kell, hogy Aliz tudja, hogy Balázs tudja, és így tovább. A probléma elemzése megmutatja, hogy a film címének valójában közös tudásnak kell lennie annak érdekében, hogy Balázs megfelelően válaszolhasson a kérdésre.

A közös tudás mellett csoportok esetén beszélhetünk elosztott tudásról is. Egy csoport akkor rendelkezik elosztott tudással, ha a φ állítás ismerete megoszlik a tagok között, és a tudásuk összevonásával a csoport tagjai ki tudják következtetni φ -t, bár egyedül a csoport egyetlen tagja sem ismeri azt. Például, ha Aliz tudja, hogy Balázs szerelmes Cilibe vagy Dórába, és Elemér tudja, hogy Balázs nem szerelmes Cilibe, majd Aliz és Elemér megosztják egymással a tudásukat, akkor közösen rájönnek arra, hogy Balázs szerelmes Dórába, habár külön-külön egyikőjük sem tudta ezt. A közös tudás és az elosztott tudás hasznos eszközök abban, hogy megértsük és elemezzük az ágenscsoportokat érintő bonyolult helyzeteket. A példaként hozott szituációk mellett a (Fagin és társai, 2003) könyvben további helyzetek olvashatók.

A modális logika szemantikáját a lehetséges világok modellje adja meg. Ennek gondolata azon alapul, hogy a dolgok valódi állása mellett számos más lehetséges állapot vagy „világ” létezik. Ennek értelmében egy tudásbázis alapján egy ágens nem feltétlenül képes megmondani, hogy a számára lehetséges világok közül melyik írja le a dolgok tényleges állását. Azt mondhatjuk, hogy az ágens számára akkor lesz egy állítás ismert, ha az igaz minden olyan világban, amit ő lehetségesnek tart. Például lehetséges, hogy egy ágens Budapesten tartózkodik egy napsütéses napon, de nincsen semmilyen információja a debreceni időjárásról. Így minden olyan világban, amit az ágens lehetségesnek tart, Budapesten süt a nap, másrészt, mivel az ágensnek nincsen információja a debreceni időjárásról, vannak olyan lehetséges világok számára, ahol Debrecenben süt a nap, és vannak olyanok, amelyekben esik. Ezek alapján, minél kevesebb a világ, amelyet egy ágens lehetségesnek tart, annál kisebb a bizonytalansága és annál többet tud. Ha például az ágens újabb információhoz jutna, és megtudná, hogy Debrecenben süt a nap, akkor többé nem tartaná lehetségesnek azokat a világokat, amelyekben Debrecenben esik az eső. (Fagin és társai, 2003)

3.2 A modális logikai kalkulus és a LoTREC rendszer

A modális logika atomi állítások nemüres halmaza: $\Phi_0 = \{p_1, p_2, \dots\}$, és számos logikai szimbólumot foglal magába: $\perp, \neg, \wedge, \diamond, \top, \vee, \supset, \equiv, \Box, (,)$, valamint a $\pi(p)$ szimbólum jelöli azon lehetséges világok halmazát, ahol a p állítás igaz. Az $M = (W, R)$ jelölés esetében M a modális logika szemantikáját leíró Kripke struktúra, W a lehetséges világok nemüres halmaza, R pedig egy ezen értelmezett, binér rákövetkező reláció, amely reflexív és euklideszi.

A tudás a modális logika szerint számos tulajdonsággal rendelkezik, amelyek a logikára vonatkozó tételként kerültek megfogalmazásra. A modalitás jellegétől függően más és más tulajdonságokat írhatunk elő. A „szükségszerű” ($\Box$) és a „lehetséges” ($\diamond$) ókori megközelítése, vagy a „tud” vizsgálata egyaránt az S5 axiómarendszerhez vezet:

- a) K: $\Box(\varphi \supset \psi) \rightarrow (\Box\varphi \supset \Box\psi)$: ha szükségszerű, hogy egy állításból következik egy másik állítás, akkor ha szükségszerű az egyik állítás, abból az következik, hogy szükségszerű a másik állítás is. Ezt disztribúciós axiómának nevezzük, amely lehetővé teszi a szükségszerűség implikáción való terjedését, azaz feltételezi, hogy az ágensek képesek következtetések levonására.

- b) T: $\Box\varphi \supset \varphi$: a tudás axióma, amely kimondja, hogy ha egy állítás szükségszerű, akkor igaz.
- c) 4: $\Box\varphi \supset \Box\Box\varphi$ (pozitív önelemzés): ha egy ágens tud egy állítást, akkor azt is tudja, hogy tudja az állítást; ezt pozitív önelemzés axiómának nevezzük.
- d) 5: $\neg \Box\varphi \supset \Box\neg\Box\varphi$ (negatív önelemzés): ha egy ágens nem tud egy állítást, akkor tudja, hogy nem tudja az állítást; ezt negatív önelemzés axiómának nevezzük.

Az S5 axiómarendszert teljesítő szemantikákban a lehetséges világok között értelmezett rákövetkező tulajdonság reflexív és euklideszi. A modális logikai kalkulus segítségével lehetőség van a különböző formulák vagy formulahalmazok tulajdonságainak ellenőrzésére, hogy azok tartalmazznak-e valamilyen ellentmondást, esetleg kielégíthetők-e, vagy pedig hogy egy adott formula érvényes-e.

Az automatikus tételbizonyítás vizsgálatához a *LoTREC* rendszert vettem alapul, amely a *Toulouse-i Informatikai Kutatóintézet* *Andreas Herzig*, *Dominique Longin*, *Mohamad Sahade* és *Olivier Gasquet* által fejlesztett általános tablórendszer modellek vagy ellenmodellek építésére, valamint a modális és ítéletlogikában megjelenő formulák megfelelőségének tesztelésére. (Gasquet és társai, 2005)

Az automatikus tételbizonyítás tanulmányozása során a *LoTREC* rendszerben megtalálható S5 stratégiára támaszkodva, és ezt felhasználva létrehoztam az $S5_n$ stratégiát, amely a korábbi bővített változata, és n szereplő kezelésére alkalmas. Ez a módszer képes olyan formulahalmaz vizsgálatára is, amely különböző ágensekhez tartozó tudás állításait egyaránt tartalmazza.

CPL Strategy : repeat firstRule Stop NotNot NotNec NotPos And NotOr NotImp NotAnd NotEquiv Imp Equiv Or end end	S5 _n Strategy: repeat CPLStrategy MarkFulfilledPos Pos ReflexiveArcs SymmetricArcs TransitiveArcs Nec end
--	--

1. ábra. Az $S5_n$ stratégia

A kalkulus elvégzésére alkalmas stratégia tulajdonképpen ciklusba rendezett kifejtési szabályok összessége. Mindaddig ismétli a lépéseit, amíg vannak kibontandó állítások vagy lehetőség műveletek végrehajtására. Az $S5_n$ stratégiában szereplő ciklus első lépése a CPL stratégia végrehajtása, amely a klasszikus logikai kijelentések kifejtéséért felelős.

A következő jelölésekben M egy Kripke-struktúra, w és u lehetséges világok, A_1 , A_2 és A pedig egy-egy állítás. Ha egy világban igaz két állítás és kapcsolata, akkor igaz mindkét állítás külön-külön is.

$$(M, w) \Vdash A_1 \wedge A_2 \Rightarrow (M, w) \Vdash A_1 \text{ és } (M, w) \Vdash A_2$$

Ha igaz két állítás *vagy* kapcsolata, akkor az adott világban vagy az egyik, vagy a másik állítás igaz.

$$(M, w) \Vdash A_1 \vee A_2 \Rightarrow (M, w) \Vdash A_1 \text{ vagy } (M, w) \Vdash A_2$$

Ha egy világban *lehetséges* egy adott állítás, akkor létre kell hoznunk egy új csúcsot (világot), ami kapcsolatban van az előzővel, és ott teljesülnie kell az állításnak.

$$(M, w) \Vdash \Diamond A \rightarrow +u, wRu, (M, u) \Vdash A$$

Ha a világban *szükségszerű* egy állítás, akkor az adott csúcs minden rákövetkezőjében szerepelnie kell az állításnak.

$$(M, w) \Vdash \Box A \rightarrow wRu, (M, u) \Vdash A$$

(Fagin és társai, 2003)

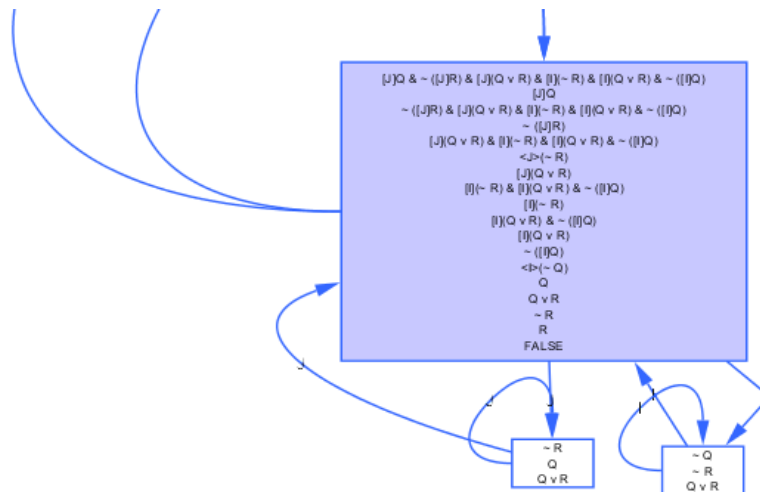
3.3 Koszos gyerekek probléma

Az általam megkonstruált kalkulus tesztelésére és vizsgálatára a modális logika egyik klasszikus problémáját, a *Koszos gyerekek* problémát vettem alapul. A feladat arról szól, hogy n számú gyerek anyukája kiengedi őket játszani a szabadba, arra kérve őket, hogy ne legyenek koszosak. A játék végén az apjuk kijelentést tesz úgy, hogy azt minden gyerek egyszerre hallja, miszerint legalább az egyikőjük homloka koszos, ezáltal az információ közös tudássá válik a gyerekek számára. Ezt követően az apa megkérdezi a gyerekektől, hogy tudják-e, hogy koszos-e a homlokuk. A modellezés során eltérő számú gyerekeket tartalmazó állapotokat vizsgáltam, ezek közül a két gyereket érintő eset az, amelynek az állapotai a legegyszerűbben modellezhetők. Ezek közül három választott helyzetet részletesebben is ismertetnék.

Az első állapot esetében az egyik gyerek homloka koszos, és az apa kijelentése után vagyunk. Jelölje I és J a gyerekeket, Q az I gyerek koszosságát, R pedig a J gyerek koszosságát kijelentő állítást. Mivel J látja, hogy I koszos, ezért tudja is (azaz $[J]Q$), tehát számára szükségszerű lesz a Q állítás, ugyanakkor azt, hogy ő saját maga koszos (R), már nem tudja (tehát $[J]R$ nem igaz, ami miatt $\sim[J]R$ teljesül). I látja — így tudja —, hogy J homloka nem koszos, így számára szükségszerű, hogy az R állítás nem igaz. Emellett mindkét gyerek számára szükségszerű, hogy Q vagy R igaz, hiszen az apjuk kijelentése alapján ezt tudják mindketten, sőt azt is tudják, hogy ezt mindketten tudják. A cél annak a vizsgálata, hogy előfordulhat-e, hogy I nem tudja, hogy ő koszos. Azaz a helyzetet leíró formulahalmazt az $[I]Q$ állítás tagadásával bővítjük, és ellenőrizzük, hogy ez a formulahalmaz kielégíthető, azaz előfordulhat-e? A *LoTREC* ASCII karaktereket használ, ezért a doboz alsó index I helyett a [I] karaktereket használja az I tudja jelölésére:

$$[J]Q \ \& \ \sim ([J]R) \ \& \ [J](Q \vee R) \ \& \ [I](\sim R) \ \& \ [I](Q \vee R) \ \& \ \sim ([I]Q)$$

A modellre lefuttatva a *LoTREC* program $S5_n$ stratégiáját egyértelmű, hogy a rendelkezésre álló információk alapján a koszos gyerek (I) tudja, hogy az ő homloka koszos, mivel a formulahalmazból képezhető lehetséges világok mindegyike ellentmondást tartalmaz.



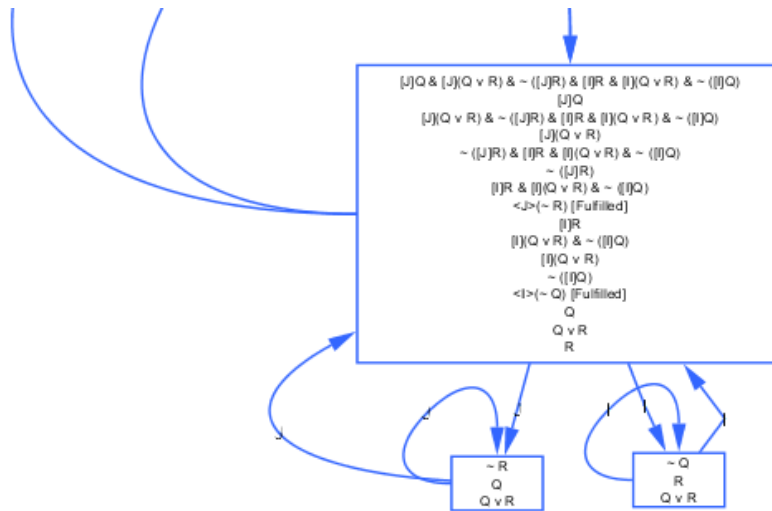
2. ábra. Példa egy lehetséges premodellre

Az ábrán szereplő téglalapok a lehetséges világokat jelölik, míg a nyilak a rákövetkező relációt. Viszont a több szereplő miatt több rákövetkező relációt tartalmaz a szemantika, ezt a nyilakra írt betűk azonosítják.

A második vizsgált állapot esetén mindkét gyerek homloka koszos, és az apa bejelentése után vagyunk. Ezek alapján J szerint szükségszerű Q , és I szerint szükségszerű R , hiszen mindkét gyerek látja a másik koszos homlokát. Emellett az apa kijelentése miatt mindkét gyerek számára szükségszerű, hogy Q vagy R igaz:

$$[J]Q \ \& \ \sim ([J]R) \ \& \ [J](Q \vee R) \ \& \ [I]R \ \& \ [I](Q \vee R) \ \& \ \sim ([I]Q)$$

A vizsgálat célja szintén az, hogy a koszos gyerekek meg tudják-e mondani a rendelkezésre álló információk alapján, hogy a homlokuk koszos-e. Az automatikus tételbizonyítást lefuttatva a *LoTREC* rendszerben, a program talál egy premodell, amely nem tartalmaz ellentmondást, azaz a feltételekből nem következik, hogy a koszos gyerekek rájönnek arra, hogy koszosak.



3. ábra. Ellentmondásmentes premodell

A harmadik vizsgált állapot, az előző helyzet utáni szituációt írja le, azaz az apa egyszer már megkérdezte, hogy a gyerekek tudják-e, hogy koszosak-e. Az előző állapot eredményéből látszik, hogy a gyerekek nem tudtak válaszolni a kérdésre, nem tudták, hogy koszos-e a homlokuk. A formulahalmaz abban változik, hogy a korábbi tudás mellett mindkét gyerek számára szükségszerűvé válik, hogy társuk lehetségesnek tartja, hogy a saját homloka nem koszos, illetve arra is rájönnek, hogy egymás homlokát koszosnak látják, hiszen, ha nem így lenne, az első kérdésre tudták volna a választ:

$$[J]Q \ \& \ [I]R \ \& \ [J]([I](Q \vee R)) \ \& \ [I]([J](Q \vee R)) \ \& \ [J] \langle I \rangle (\sim Q) \ \& \ [I] \langle J \rangle (\sim R) \\ \& \ [J](\sim R \rightarrow [I](\sim R)) \ \& \ [I](\sim Q \rightarrow [J](\sim Q)) \ \& \ \langle I \rangle (\sim Q) \ \& \ \langle J \rangle (\sim R)$$

A cél most is annak a vizsgálata, hogy a koszos gyerekek rájönnek-e, hogy koszosak. Az automatikus tételbizonyítást lefuttatva a keletkezett 17 premodell mindegyike *FALSE*

kimenetet biztosít, azaz a feltételekből következik, hogy a koszos gyerekek képesek megmondani, hogy koszosak. (Herzig és társai, 2005)

4. A Hanabi társasjáték

A *Hanabi* társasjáték a kutatásom során egy eszköz annak a hipotézisnek a vizsgálatára, miszerint a kooperatív és információhiányos környezettel rendelkező játékok esetén a modális logika segítségével a játékosok jobb eredményt képesek elérni, mint anélkül. Az alkalmazás elkészítésével és tesztelésével ennek a feltételezésnek a helyességét szerettem volna ellenőrizni.

A *Hanabi* egy kooperatív játék, ami azt jelenti, hogy a játékosok nem versengenek, hanem együtt, egy csapatként játszanak. A játék célja, hogy a játékosok káprázatos tűzijátékok létrehozásával minél több pontot gyűjtsenek, ezáltal minél nagyobb elismerést kapjanak a bemutató közönségétől. A játékban lévő tűzijáték lapok 5 különböző szín valamelyikével és 1-től 5-ig terjedő skála egy értékével rendelkeznek. Azonos színű 1-es értékkel rendelkező lapból három, 2-es, 3-as és 4-es értékkel rendelkező lapból kettő, míg 5-ös értékű lapból egy darab van a pakliban. A játékosoknak a tűzijáték kártyákat színek szerint csoportosítva, monoton növekvő sorrendben kell kijátszaniuk úgy, hogy minden értékből csak egy szerepelhet egy tűzijátékban. A dedukciós környezetet az adja, hogy senki sem látja a saját lapjait, így kénytelenek a többi játékos tanácsaira hagyatkozni. Minél több lapot sikerül a játékosoknak helyesen kijátszani, annál több pontot kapnak a játék végén.

A játék során három lehetséges lépés közül választhatnak a játékosok. Az egyik lehetőség a segítségnyújtás, amikor a soron következő játékos az egyik társa kézben tartott lapjaira vonatkozóan tesz kijelentést. Ehhez a 8 rendelkezésre álló segítségjelölő közül egyet az elhasznált oldalára kell fordítani. Segítség során kétféle információt lehet adni. Lehet egy adott színről, például „Ez a lap piros” vagy „Két zöld lapod van, ez és ez” vagy „Nincs sárga kártyád”, illetve lehet egy adott értékről, például „Van egy 5-ös lapod, itt” vagy „Két 1-es lapod van, ez és ez” vagy „Nincs 3-as lap a kezében”. A játékosoknak teljes információt kell adniuk, azaz ha valakinek két zöld lapja van, akkor az információt adó játékos nem teheti meg, hogy csak az egyikre mutat rá. Ha már nincsen elérhető segítségjelölő, a soron következő játékos nem választhatja ezt az akciót.

A játékosok egy másik lehetséges lépése egy kézben tartott lap eldobása. A játékos ekkor eldobja az egyik lapját, és a dobott lapok paklijára helyezi, képpel felfelé. Ez után új lapot húz a kezébe úgy, hogy véletlenül se lássa a képes oldalát. Lapeldobást követően a játékosok visszakapnak egy már felhasznált segítségjelölőt. Ezt az akciót nem választhatja a soron következő játékos, ha még az összes segítségjelölő rendelkezésre áll. Ebben az esetben más akciót kell választania.

A harmadik lehetőség egy kézben tartott lap kijátszása. A játékosok ennek a segítségével tudnak tűzijátékokat létrehozni. A soron következő játékos ekkor kiválaszt egy lapot a kezéből, és képpel felfelé az asztal közepére teszi. Ezután két dolog történhet: az egyik, hogy a lap szabályosan lehelyezhető, azaz a játékos a kijátszott lappal képpel felfelé kiegészít egy már meglévő tűzijátékot, vagy elkezd egy újat. A másik lehetőség, hogy a lap nem helyezhető le szabályosan, ekkor a lapot a dobópaklira teszi, és egy viharjelölő kerül felfordításra. Ha a harmadik viharjelölő is felfordításra kerül, a játék egyből véget ér, és a játékosok a kiépített tűzijátékoktól függetlenül elvesztik a játékot. Lapkijátszás után a játékos új lapot húz a húzópakliból, ügyelve arra, hogy azt továbbra se nézze meg.

A társasjáték információhiányos mivolta a játékosok kézben tartott lapjain alapul. Mivel senki nem láthatja közvetlenül a saját lapjait, így azoknak a milyenségére vonatkozó tudás hiánya dedukciós környezetet eredményez. Az információ hiányának mérete a játék előrehaladtával folyamatosan változik, hiszen a segítségnyújtás operátor használatával a játékosok egyre több információval láthatják el társaikat. Ezek alapján a tudás mérete folyamatosan növekszik, ami az információhiány csökkenését okozza. Ugyanakkor vannak olyan esetek a játék során, amikor bizonyos műveletek hatására az ágensek tudása csökken, például lap eldobása vagy kijátszása esetén. Ennek oka, hogy a kézből lehelyezett lapra vonatkozó információ átalakul, és az újonnan felhúzott lapra vonatkozó újabb bizonytalanság következtében az ágens számára újra több lehetséges világ válik elérhetővé. Egy lap kijátszása bizonyos esetekben módosíthat az ágensek lehetséges világainak elképzelésén, hiszen a lap megfordítását követően annak színe és értéke közös tudássá válik, ezáltal csökken annak az esélye, hogy a játékosok valamely kézben tartott lapja megegyezik az éppen kijátszásra került lappal. Az egyes ágensek tudásának rögzítésére, valamint a meglévő információkból újabb

lehetséges állítások kiszámítására jó választás lehet az episztemikus modalitás eszközeinek használata.

4.1 Szoftveres megvalósítás

A *Hanabi* társasjáték számítógépes megvalósítása több elemből épül fel. A játékspecifikus részekhez tartozik a játék állapotterét megvalósító osztály, a játékban megtehető operátorokat leíró osztályok és a játék vezérlését végző szerkezet. A játékot vezérlő programhoz további osztályok kapcsolódnak, mint például a lépésajánlást létrehozó osztály, vagy az ehhez szükséges operátor-szimulátor osztályok. Az állapot és az operátorok leírásához absztrakt osztályokat is alkalmaztam, amelyekben összegyűjtöttem azokat az eszközöket, amelyeket a megvalósító osztályokban felhasználtam, ezáltal egyszerűbbé téve a program szerkezetét. Mindezek mellett a játékban meglévő kártyák kezelésére enumerációkat hoztam létre.

A korábban összeszedett osztályok segítségével az alkalmazás már képes teljes játékok lebonyolítására, azonban az általam feltett hipotézis vizsgálatához a modális logikai kalkulus implementálása is szükséges. Ennek megvalósítására létrehoztam egy kalkulus osztályt, valamint annak egy, a *Hanabi* játékhoz tartozó specifikus alosztályát is, ami a játék sajátos szabályait is figyelembe veszi. A modális logikai állítások létrehozására készítettem formula osztályokat, amelyek egy és két operandusú, illetve atomi formulák leírására képesek, valamint létrehoztam egy enumerációt is a logikai operátorok tárolására.

4.2 Modális logikai kalkulus a Hanabi társasjátékhoz

A *Hanabi* társasjáték esetén a modális logika eszközei felhasználhatók az egyes ágensek tudásainak rögzítéséhez. Mivel az információhiányos környezettel rendelkező játékok esetén fontos tényező a környezetről érkező állítások tárolása, a tényleges világ megismeréséhez az episztemikus modalitás beépítése jó választás lehet.

Az ágensek tudása a játék esetén több helyről érkezik. Egyrészt vannak azok az információk, amelyek a látványból adódnak. Ezeknek egy része közös tudás az ágensek számára, hiszen minden játékos ugyanazokat a kijátszott lapokat látja, ugyanazt a dobópaklit ismeri, és ugyanazokról a segítség- vagy viharjelölő állapotokról van tudása. Ugyanakkor

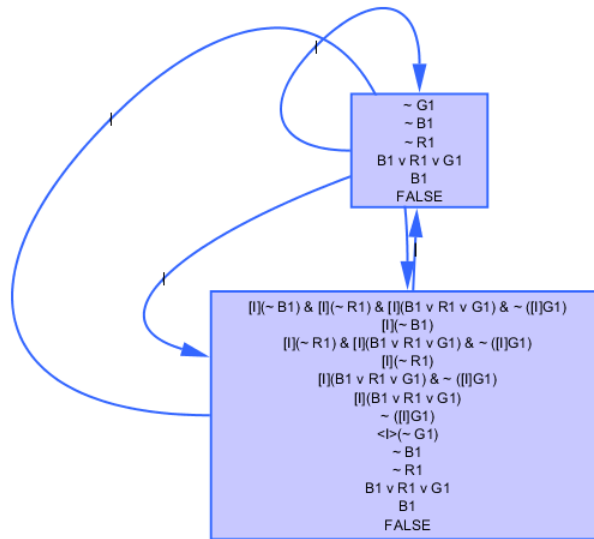
vannak olyan információk, amelyek egy-egy ágenshez tartoznak, és más ágensnek számára nem, vagy nem feltétlenül ismertek. A játék kezdőállapotában például kimondottan kevés információval rendelkezhetnek a játékosok saját lapjukra vonatkozóan, míg játékos társaik tökéletesen ismerik azokat. Ugyanakkor az is lehetséges, hogy néhány információt már a játék kezdőállapotában is ki tudnak következtetni. Például, ha egy játékos a társa kezében lát egy sárga ötös lapot, akkor biztosan tudja, hogy az ő kezében lévő lapok egyike sem lehet sárga ötös, hiszen abból csak egy darab van a pakliban. Mivel a látványból, azaz a már kijátszott lapokból meghatározható tudás a játékosok számára mindig elérhető, azok tárolására külön nincs szükség. Az ágensek tudását tartalmazó listába azok a formulák kerülnek be, amelyek valamilyen segítségnyújtás során merülnek fel, ezek pedig közös tudásként kezelhetők, hiszen a lépés kijátszása után már mindenki tudja az információt ahogy azt is tudják, hogy a többiek is tudják azt.

Az alkalmazásba beépített modális logikai kalkulus megvalósításához a *LoTREC* rendszerben létrehozott $S5_n$ startégiát vettem alapul. Ezt felhasználva készítettem el a *Calculus* osztályt, amelyben a stratégiában szereplő, operátorok kifejtésére vonatkozó szabályokat rekurzív metódusok valósítják meg.

Package::hu.unideb.inf.cooperative.games.knowledge Calculus
#knowledge: List<Formula> #expandedNEC: List<Formula>
+Calculus(knowledge:List<Formula>) +contains(formula:Formula): boolean +canAddTheNewStatement(newStatementToTest:Formula): boolean +isCorrectKnowledge(): boolean +stop(changed:boolean): boolean +extractNotNot(changed:boolean): boolean +extractNotNec(changed:boolean): boolean +extractNotPos(changed:boolean): boolean +extractAnd(changed:boolean): boolean +extractNotOr(changed:boolean): boolean +extractNotImp(changed:boolean): boolean +extractNotAnd(changed:boolean): boolean +extractNotEquiv(changed:boolean): boolean +extractImp(changed:boolean): boolean +extractNotEquiv(changed:boolean): boolean +extractOr(changed:boolean): boolean +extractNec(changed:boolean): boolean +isSyntacticallyCorrect(): boolean +createNewWorld(possibilityToExpand: OneOperandFormula): List<Formula>

4. ábra. A *Calculus* osztály osztálydiagramja

A kalkulus használata során azt tapasztaltam, hogy a formulahalmaz vizsgálata sok időt vett igénybe azokban ez esetekben, amikor a formulák kibontásával keletkező fa elágazási tényezője 1-nél nagyobb volt. Ez a játékban lévő lapokra vonatkozó tudások közül leginkább a *vagy* operátort tartalmazó formulák kibontása miatt jelent meg. Vegyünk példának egy leegyszerűsített játékállapotot, amikor a játékban három szín létezik, kék (*B*), piros (*R*) és zöld (*G*). A lapokra vonatkozó alap tudás alapján mindegyik lapnak van színe, így például az *I*-vel jelzett játékos számára szükségszerű lesz az első lapjára vonatkozóan, hogy az vagy kék (*B1*),



5. ábra. Egy egyszerűsített játékállásra vonatkozó tudás vizsgálata során létrejött premodell

vagy piros (*R1*), vagy pedig zöld (*G1*) színű. Tegyük fel, hogy a játékos korábbi információk alapján szükségszerűnek tartja, hogy a lapja nem kék és nem piros. Ezek alapján az ágensnek ki kell tudnia kalkulálni, hogy a lapja zöld színű.

A vizsgálat arra irányul, hogy annak a feltételezésével, hogy a játékos nem tudja, hogy a lapja zöld színű, létezik-e lehetséges világ, azaz olyan modell, amely nem tartalmaz ellentmondást. A *LoTREC* rendszerben vizuálisan felépítve a problémát már erre az egyszerűsített esetre is 7 premodell jött létre. Ennek oka a *vagy* operátort tartalmazó formula felbontása. A keletkezett premodellek mindegyike *FALSE* kimenettel rendelkezik, ebből következőleg a játékos ki tudja következtetni a lapja színét.

Könnyű belátni, hogy a *Hanabi* társasjáték esetén sok hasonló formula tárolására lenne szükség, hiszen egy játékos mind az öt kézben tartott lapjáról tudja, hogy vagy piros, vagy fehér, vagy kék, vagy zöld, vagy sárga, és azt is tudja, hogy vagy egyes, vagy kettes, vagy hármas, vagy négyes, vagy ötös értékkel rendelkezik. Ezen állítások tárolása és kibontása

lassítja a kalkulus működését. A gyorsítása érdekében hoztam létre a *HanabiCalculus* osztályt, amely a *Calculus* egy alosztálya. Az osztály a játék szabályai alapján a kalkulusban szereplő szintaktikai vizsgálat mellett az adott világok szemantikai vizsgálatát is megvalósítja. Ide tartozik többek között a lapok színére és számára vonatkozó vizsgálat. Ha egy játékos birtokában van olyan információknak, amely az egyik lapja számára négy lehetséges színt, vagy 4 lehetséges értéket kizár, akkor biztos, hogy az adott lap az ötödik színnel vagy értékkel rendelkezik. Például, ha a játékos tudja, hogy a kézben tartott lapjai közül az első biztosan nem piros, nem kék, nem zöld és nem sárga, akkor a kalkulus számításaiban az az állítás, ami azt mondja ki, hogy a lapja fehér színű, a szemantikai vizsgálat révén bekerül a tudásbázisába.



6. ábra. Szemantikai helyesség vizsgálata színre vonatkozóan

A vagy operátorra vezethetők vissza azoknak az állításoknak a kifejtése is, amelyek arra vonatkoznak, hogy egy lap biztosan nem adott színű és számú. Például, ha egy játékos lát már kijátszva vagy a társa kezében három darab kék egyest, akkor számára szükségszerű lesz, hogy egyik lapja sem lehet kék és egyes egyszerre. Az ilyen alakú formulák kibontása vagy operátorokat tartalmazó formulákat eredményez a kalkulus számítása során.

$$[I](\sim (\text{BLUE1} \ \& \ \text{ONE1})) \longrightarrow [I](\sim \text{BLUE1} \vee \sim \text{ONE1})$$

Ezen a formulák számának csökkentése érdekében a tudásbázisban csak azok az információk kerülnek tárolásra, amelyek a látottak alapján nem számíthatók ki. Ennek oka, hogy annak vizsgálatakor, hogy az adott játékos egy lapja milyen lehet, a számára látható lapok mindig elérhetők lesznek, újra végig tudja őket nézni, nem szükségszerű a rájuk vonatkozó információk tárolása.

A kalkulus gyorsítása érdekében a tárolt állítások mind valamilyen szükségszerűséget fejeznek ki a lapokra vonatkozóan, a lehetséges állítások nem kerülnek a listába. Ennek oka, hogy minden, ami nem szükségszerű, és szükségszerű tudásnak nem mond ellent, az lehetséges. Ezeknek a tárolására külön nincs szükség, mert bármikor kikövetkeztethetők a jelenlegi információkból.

4.3 Lépésajánláshoz használt algoritmusok

A kooperatív játékok esetén, a hagyományos kompetitív társasjátékoktól eltérően, a játékosok nem az egyéni győzelem érdekében döntenek, hanem céljuk az együttműködés és a közös haszon maximalizálása. A *Hanabi* társasjátékhoz általam készített lépésajánló algoritmus egy rekurzív módon megvalósított keresés. Az adott játékállás alapján az állapotban végrehajtható összes lehetséges operátorral elérhető új állapotok értékei alapján határozza meg az optimális lépést. Az algoritmus a gépi játékosok konkrét lépéseinek meghatározásakor, valamint az emberi játékos számára ajánlott lépés kiszámítása során kerül alkalmazásra.

A lépésajánlás során figyelembe kellett venni, hogy az alap operátorok használata a legtöbb esetben plusz információt árulhat el a játékosok számára. Tegyük fel, hogy egy kártyakijátszás (*PutCard*) operátor végrehajtásával elérhető eredményt szeretnénk meghatározni. Az operátor egy indexet vár, ami azt határozza meg, hogy a játékos a kezéből hányadik kártyáját szeretné kijátszani. Ha a lépés eredményének meghatározásakor az eredeti operátort használnánk, az így kialakuló új állapotban kiderülne, hogy a kijátszott lap ténylegesen milyen színű és számú, és annak a lépésnek ezáltal konkrétan mekkora a hasznossága, holott valójában a lépés kiválasztásakor a játékos ennek az információnak nem feltétlenül van birtokában. Tegyük fel, hogy egy játékos annyit tud a kijátszandó lapjáról, hogy piros. Ha a lépésajánlás során az eredeti operátort használjuk, akkor a lap kijátszása után a játékos számára a lap értéke is kiderül, mivel az konkrétan felfordításra kerül. További csalást eredményezne az is, hogy a játékos által felhúzott lap is konkrétan meghatározásra kerülne, ezáltal a húzópakli felső lapja a többi játékos számára világossá válna. Ennek kiküszöbölése érdekében operátor-szimulátor osztályokat hoztam létre a lapeldobás és lapkijátszás operátorok számára. Ezen szimulátor osztályok objektumai nem csak a kijátszandó lapok sorszámaival rendelkeznek, hanem azzal is, hogy mit gondol az adott játékos, vajon milyen lap az, amit a kezéből lehelyez, és mit gondol, mit fog húzni a helyére. Ezek alapján egyértelmű, hogy szimulátor operátorból több lehetséges lépést lehet generálni, mint az alap operátorokból a lapok jelenlétének következtében. A játék elején, amikor a játékosok minimális tudással rendelkeznek, bármely lap kijátszását és felhúzását lehetségesnek tarthatják. A játék 25 különböző lapot tartalmaz, így az ágensek számára egy darab, adott

sorszámú lap lehetséges kijátszására vonatkozó szimulátor operátorok (*PutCardSimulator* példányok) száma 620. Mivel ötös értékű lapból egy darab van minden színben, így azt nem lehet egyszerre le is tenni és fel is húzni:

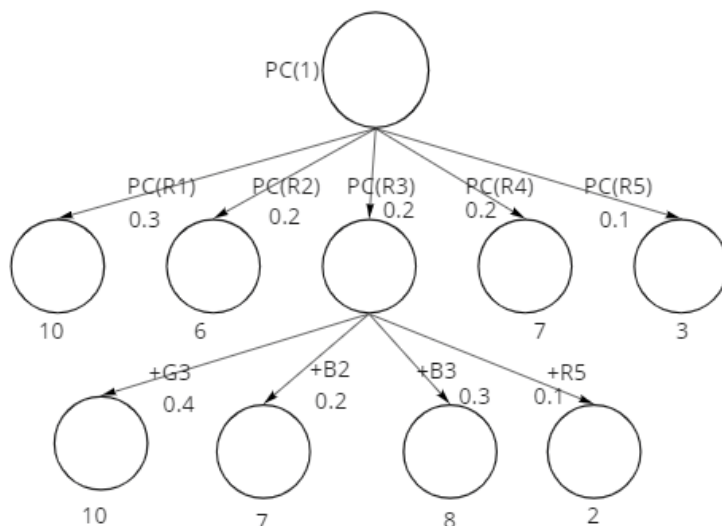
$$20 \cdot 25 + 5 \cdot 24 = 620$$

A szimulátor operátorok segítségével meghatározható egy tényleges operátor hasznosságának várható értéke. Tegyük fel, hogy a soron következő játékos az első lapját szeretné kijátszani a tűzijátékokra. Ehhez az operátorhoz a fenti számítás alapján 620 szimulátor operátor párosítható. A konkrét lépés hasznosságának becslését – több más lehetőséget is végiggondolva – a szimulátor operátorok által elérhető hasznosságok súlyozott összegével határoztam meg. Ez alapján minden lehetséges lapkombinációval lejátszva a lépést, kiszámítottam az így létrejövő állapotok hasznosságait, majd ezt az értéket súlyoztam a lapok előfordulásának valószínűségével.

A valószínűségek kiszámítását metódusok végzik. Annak a valószínűségnek a meghatározására, hogy a játékos kézben tartott kártyája egy adott lap, több szempontot is figyelembe kell venni. A játékban 50 darab kártya szerepel, ebből le kell vonni azokat a lapokat, amiket a játékos lát a kijátszott tűzijátékokon, a dobópakliban vagy a játékosársai kezében. Ezek a kártyák biztosan nem lehetnek a saját kezében, azonban ügyelni kell arra, hogy az ugyanolyan kártyákból is csak annyit vonjunk ki, amennyit lát a játékos. Ezen túl a játékos tudásbázisa is plusz információkat adhat. Ha egy játékos tudja, hogy a lapja piros, akkor minden nem piros lapot kivehetünk a lehetséges lapok közül, ha pont ellenkezőleg azt tudja, hogy nem piros, akkor pedig a piros lapokat hagyhatjuk el. Az így kiszűrt pakliban az 50 kártyából már csak azok maradnak, amelyek lehetségesek, hogy a játékos kezében vannak a megfelelő indexű helyen. A kártyának a valószínűsége a megmaradt lapokban vett előfordulása osztva az összes lehetséges lap darabszámával. Így, ha egy játékos tudja, hogy az adott kártyája piros, annak a valószínűsége, hogy kék hármás van a kezében, 0, míg annak, hogy piros egyes – feltéve, hogy egy piros lapot sem lát a játékos máshol – 0,3, mivel három darab egyes van a tíz piros lap között. A felhúzendó lap valószínűsége is hasonló módon kerül meghatározásra, viszont ebben az esetben a látott lapok kivonása mellett a tudásbázis alapján kiszámítható, biztosan kézben tartott lapok kerülnek levonásra, illetve az a kártya, amit

feltételezünk, hogy kijátszani készülünk. A húzott lap valószínűségének meghatározása feltételes valószínűség felhasználásával kerül kiszámításra.

Tegyük fel, hogy a soron következő játékos első lapjának kijátszásával (*PutCard(1)*) elérhető hasznosságot szeretnénk meghatározni úgy, hogy a játékos annyit tud a lapjáról, hogy piros. Egyszerűség kedvéért feltételezzük, hogy egy piros lapot sem lát még kijátszva, és a kijátszott lap helyére négyféle lapot tud felhúzni adott valószínűséggel.



7. ábra. Példa a várható hasznosság meghatározására

A 7. ábrán látszik, hogy 5 kártya kijátszása lehetséges a piros szín ismeretében, a piros egyes (*R1*), piros kettes (*R2*), piros hármas (*R3*), piros négyes (*R4*), valamint a piros ötös (*R5*). Ezeknek a valószínűségét az előfordulásuk száma határozza meg. A piros hármas kijátszását feltételező esetben a felhúzható lapokat is megjelenítettem a rájuk vonatkozó valószínűséggel. Az operátor végrehajtása utáni állapotot az alsó karikák szemléltetik. Tegyük fel, hogy ezekben az állapotokban az adott hasznosságok érhetők el. Ennek értelmében a piros hármas lap kijátszásával elérhető várható hasznosság ezek súlyozott összegével lesz egyenlő, ami 8. Ennek segítségével meghatározható a *PC(1)*, azaz az első lap kijátszását jelentő operátor becsült hasznossága is, ami a következő:

$$0,3 \cdot 10 + 0,2 \cdot 6 + 0,2 \cdot 8 + 0,2 \cdot 7 + 0,1 \cdot 3 = 7,5$$

Egy operátor becsült hasznossága tehát a lehetséges kártyákkal elérhető új állapotok hasznosságának súlyozott összege, ahol a súlyok a kártyák valószínűségei.

Az állapotok becsült hasznossága mellett a különböző lehetséges kártyák kijátszásával elérhető hasznosságok szórása is fontos tényező lehet. Ha egy operátor valamivel nagyobb

várható hasznossággal rendelkezik, mint egy másik, de az elérhető hasznosságok szórása sokkal nagyobb, biztosabb választás a kicsivel kisebb várható hasznossággal rendelkező, de sokkal kevésbé szóró lépés választása. Lehet, hogy ezzel egy nagyon jó lépés megtételét tartjuk vissza, de az is lehet, hogy egy nagyon rossz döntést előzünk meg vele.

A segítségnyújtás operátor esetén nincs szükség valószínűségek számítására, hiszen azok egy-egy állítás kijelentését tartalmazzák. Ezekben az esetekben a játékosok plusz információhoz jutnak, és esetlegesen a kalkulust felhasználva újabb állításokat következtethetnek ki.

A hasznosságértékek meghatározására több, különböző szempontokat figyelembe vevő hasznosságfüggvényt hoztam létre. Az első változat kizárólag a tűzijáték jelenlegi állapotát veszi figyelembe, azaz a már elért pontok alapján határozza meg az állapotok hasznosságát. A következő függvény emellett a látható tényezőket is figyelembe veszi. Plusz pontot jelent, ha a dobópakli mérete még kicsi, hiszen ekkor nagy esély van még arra, hogy több kártya kerüljön helyesen kijátszásra, ezzel növelve az elért pontokat. További plusz pontot jelent az is, ha olyan lap kerül a dobópakliba, amire már nincsen szükség, mert például már kijátszásra került egy ugyanilyen a tűzijátékra. Ugyanakkor negatív az az eset, amikor egy adott típusú kártyából, például a kék hármashból, az összes a dobópakliban van, hiszen ebben az esetben az adott színű tűzijátékot már biztosan nem tudják befejezni a játékosok. Emellett a hasznosságfüggvény bünteti a viharjelölők számának növekedését is, hiszen ezek a játék elvesztéséhez vezethetnek. A harmadik hasznosságfüggvény az előzőkön túl, a tudásbázist is figyelembe veszi. Az állapotokat az alapján is értékeli, hogy a játékosoknak mennyire van egyenlő méretű tudása a saját lapjukra vonatkozóan, plusz pontot ad azokért az információkért, ha egy játékos tudja, hogy nála van egy, a tűzijátékra soron következő kártya, illetve vizsgálja azt is, hogy van-e lehetősége a játékosoknak további segítségnyújtást adni egymásnak.

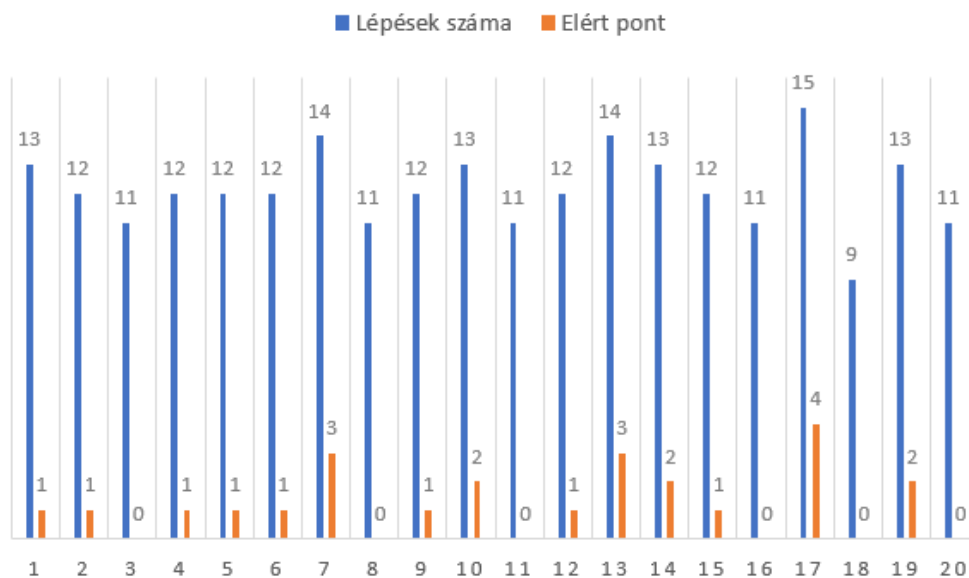
5. A Hanabi alkalmazás működése

Kutatásom elején azt tűztem ki célul, hogy megvizsgáljam a játékelmélet és a mesterséges intelligencia módszereinek felhasználásával, hogy az ágensek milyen eredményeket képesek elérni a játék folyamán különböző szempontok figyelembevételével. Szerettem volna összevetni a játék sikerességét azokban az esetekben, amikor a lépéseket az ágensek teljesen véletlenszerűen választják, azokkal a változatokkal, amikor valószínűségszámítással kapcsolatos kalkulációkat is figyelembe vesznek, illetve mikor a modális logika eszközei is beépülnek a játék menetébe, ezzel növelve az ágensek tudásbázisát.

Ennek vizsgálatára a lehető legegyszerűbb modellt választottam, így az általam készített implementáció jelenleg kétszemélyes játékok lebonyolítására alkalmas. Ezt felhasználva különböző paraméterekkel beállított játékokat teszteltem úgy, hogy a játszmákban két gépi ágens játszott együtt. A kapott eredményeket, a lejátszott játékok sikerességét és az azokban megfigyelt eltéréseket a következőkben részletezem.

5.1 Tapasztalatok a játék során elért pontszámokban

Vizsgálataim első fázisa során az ágensek teljesen véletlenszerűen választottak a lépések közül. Ebben az esetben a hasznosságfüggvény egységesen 0 értéket rendelt minden állapothoz, így a lépésajánlás során vizsgált lehetséges operátorok között nem volt hasznosságbeli különbség, az ágensek ugyanolyan jónak vagy rossznak gondoltak minden műveletet. Ezáltal a gépi játékos a megtehető lépések közül véletlenszerűen, az első lehetséges lépést választotta. Az általam elvárt eredmény az volt, hogy az ilyen véletlenszerűen választott operátorokkal megjátszott lépések hamar a játék végéhez vezetnek. Ebben az esetben az ágensek tulajdonképpen nem is „gondolkodnak”, csak annyit vesznek figyelembe, hogy egy adott állapotra mely lépések tehetők meg, ezek közül pedig rögtön az első meg is lépik. Mivel ebben az esetben egy lapkijátszás operátor ugyanolyan hasznos, mint egy segítségnyújtás vagy egy lapeldobás, függetlenül a lap milyenségétől, a gépi játékosok választásukkal hamar viharjelölő felfordítását érhetik el. Három rossz lapkijátszás pedig a játék végéhez vezet. A kapott eredmények a feltételezés helyességét bizonyítják.



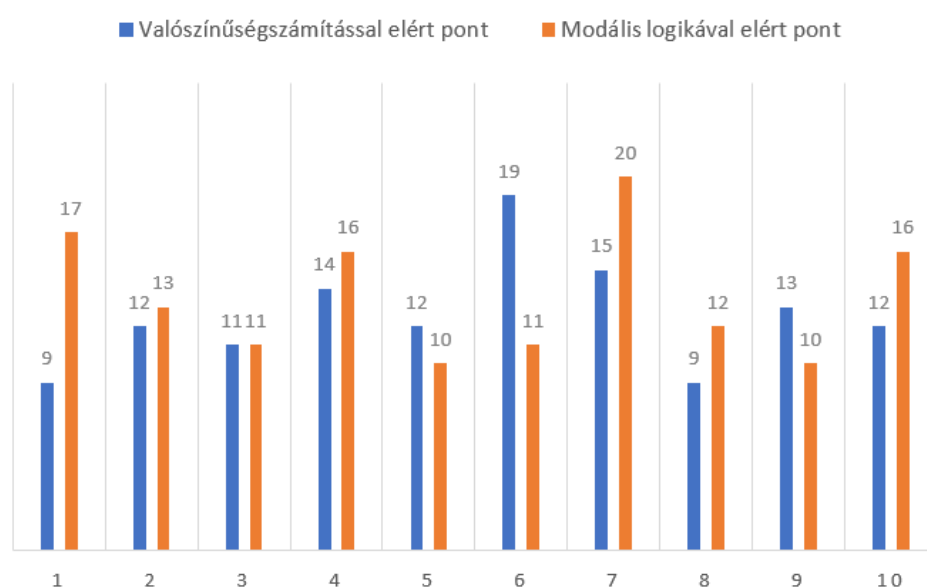
8. ábra. Véletlenszerűen választott operátorokkal elért eredmények

A véletlenszerűen választott lépések alapján végigjátszott játékok mindegyike vesztes kimenetet eredményezett a három viharjelölő megszerzése miatt. Így az elért pontoktól függetlenül a játékosok elvesztették a játékot. A 8. ábrán a vizsgált játékok közül 20-nak az eredménye látható. A kék oszlopok a játékban megtett lépések számát jelölik. Látszik, hogy átlagosan 12 lépést sikerült az ágenseknek a játék vége előtt megtenniük. Mivel a lehetséges lépések között 5 féle lapkijátszás, 5 féle lapeldobás és 10 féle segítségnyújtás operátor szerepelhet, annak a valószínűsége, hogy lapeldobás vagy segítségnyújtás operátor kerül véletlenszerűen kiválasztásra, nagyobb. Ezek a lépések nem eredményezik viharjelölő felfordulását, a játék elvesztéséhez a rosszul meglépett lapkijátszás operátorok vezethetnek. Az ábrán a narancssárga oszlopok a játékban elért pontokat jelölik. Több lépés megtétele esetén a megszerzett pontok is magasabbak, hiszen ebben az esetben a lapkiosztás és a véletlenszerűen megjátszott lapkijátszás operátor szerencsés kombinációja révén helyes lapok kerültek a tűzijátékok megfelelő helyeire.

Vizsgálataim következő szintjén az ágensek a lépéseik meghatározásához a valószínűségszámítás eszközeit vették segítségül. Ennek értelmében a következő lépések meghatározásához a szimulátor operátorokban szereplő lapok előfordulásának valószínűségével súlyozták a lépés végrehajtásával megszerezhető hasznosságot, és ennek segítségével próbáltak dönteni. A valószínűség kiszámításakor a látható információk mellett az ágensek a lapjukra vonatkozó tudást is figyelembe vették. Ennek értelmében, ha tudták,

hogy például az első lapjuk piros színű, a többi színű lap előfordulásának valószínűségét eleve 0-nak tekintették. Az elemzés során több játékot lefuttatva gyűjtöttem össze az ilyen módon elérhető eredményeket, amelyekben azt tapasztaltam, hogy a játék során szerzett pontok számát nagyban befolyásolja a lapkiosztás és a lapok felhúzásának sorrendje. A játszmák végeredményében 6–20 pont megszerzése között szóródtak, azokban az esetekben, amikor előnyös sorrendben következtek a lapok, az ágensek több pontot szereztek, ellenkező esetben pedig rosszabbul teljesítettek. Ezek alapján az mondható, hogy a lapok véletlenszerűsége is befolyásolja az ágensek sikerességét az egyes játszmák során.

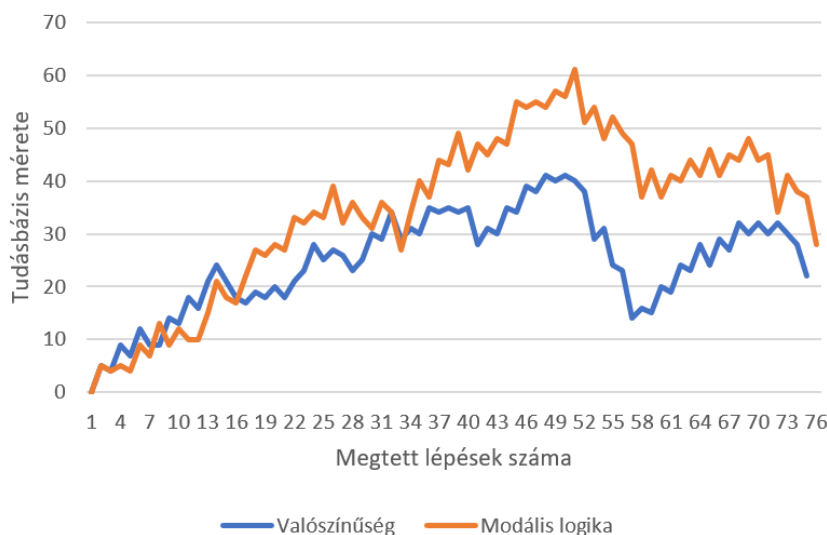
Következő lépésként azt szerettem volna megvizsgálni, hogy a csupán valószínűséget használó, valamint a modális logikai kalkulussal bővített lépésajánlással milyen eredmények érhetők el egy-egy játszma alatt. A 9. ábra ennek a kimenetét szemlélteti. Mindkét lépésajánlás esetén ugyanarra a 10 véletlenszerűen kiválasztott kezdőállapotra futtattam le a játékot, hogy a lapok eltérő sorrendje ne módosítson az eredményeken. Az így elért pontokat ábrázolják az oszlopok. Annak érdekében, hogy a játszmák összehasonlíthatók legyenek, tesztelés során az operátorok kipróbálási sorrendjét fixre állítottam, így azok előfordulásának sorrendje nem befolyásolta az ágensek döntéseit. Ez ugyan jelentősen rontja a játék élvezhetőségét, de a tesztelés miatt fontos tényezőnek számított. A diagramon látszik, hogy voltak olyan esetek, amikor csak a valószínűséget figyelembe véve sokkal jobb eredményt tudtak elérni az ágensek (6. játszma), ugyanakkor voltak olyan kezdőállapotok is, amelyek



9. ábra. Valószínűség számítás és modális logika összehasonlítása

esetén a modális logika alkalmazásával gyűjtöttek több pontot (7. játszma). A különbség a két módszer között a tudásbázis méretében nyilvánult meg.

A modális logikát használó esetben egyes információkból az ágensek újabb állításokat következtethettek ki, így azokkal bővíthették a tudásukat. A tudásbázis méretének változását a 10. ábrán szemléltettem. Látszik, hogy a modális logikai kalkulust alkalmazó lépésajánlásnál a tudásbázis mérete a játék előrehaladtával majdnem folyamatosan nagyobb volt.



10. ábra. Tudásbázis méretének változása

Mivel a hasznosságfüggvény a tudásbázis mérete alapján is értékeli az egyes játékállásokat, így a két módszer esetén az operátorok hasznosságértéke eltérő lehet. Emiatt egy adott állapotban a valószínűségszámítás az egyik, míg a modális logika a másik lépés végrehajtását látta jobbnak. Ugyanakkor a hasznosabb állapotok kiválasztásakor nem a tudásbázis mérete a számottevő, hanem az, hogy mire vonatkozik a tudás, ami az ágensek birtokában van. Értékesebb az a tudás, ami arra vonatkozik, hogy egy ágens tudja, hogy nála van egy olyan lap, ami a tűzijátékra közvetlen következik, vagy valamelyik lapja olyan, amire már nincsen szükség és eldobhatja. Olyan tudás, ami nem viszi előrébb a játékosokat, a tudásbázis méretét ugyan növeli, de a sikerességhez nem feltétlenül járul hozzá.

Ennek a tényezőnek a kizárása érdekében módosítottam a hasznosságfüggvényt, hogy szimplán a tudásbázis mérete ne jelentsen hasznosságbeli eltérést az egyes állapotok között. Az így kialakított hasznosságfüggvény segítségével újabb vizsgálatot tettem ugyanazon kezdőállapotokra futtatott játékok segítségével. Azt tapasztaltam, hogy egy lépésre előretekintve ebben az esetben a valószínűségszámítást alkalmazó és a modális logikát

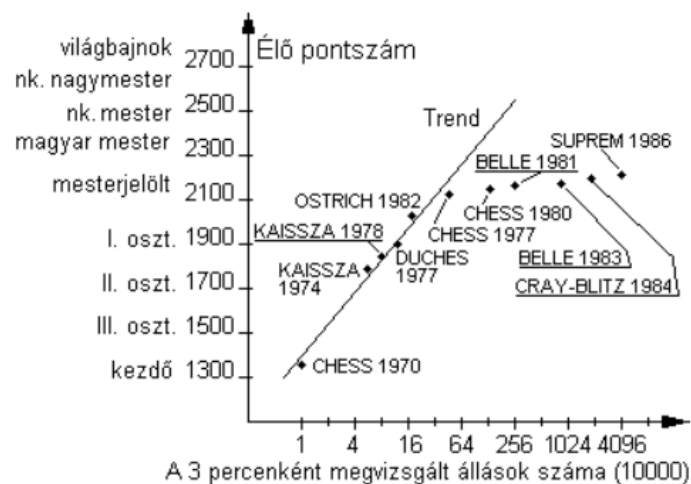
használó módszerekkel futtatott játszmák ugyanazt az eredményt érték el, ugyanazon operátorok végrehajtásával az egyes lépésekben. A hasznosságok abban tértek el, hogy a modális logikát alkalmazó esetben az olyan tudás, ami egy tűzijátékra rákövetkező lap színét és számát is egyszerre tartalmazta, plusz pontot ért. Ez a plusz információ a vizsgált játékok esetén nem jelentett operátorválasztásbeli különbséget. A megegyező eredmények oka, hogy a modális logika által kikövetkeztethető új tudás nem tárol olyan információt, ami a lapok valószínűségeinek kiszámításakor az alap tudásbázis alapján ne érvényesülne. Például, ha egy ágens tudja, hogy az első lapja nem kék, nem sárga, nem piros és nem is zöld, a modális logikával rájön, hogy fehér színű, ugyanakkor a valószínűség meghatározásakor a lehetséges lapok közül az alap tudás miatt ugyanúgy kizárja a nem fehér színű lapokat, így a konkrét állítás tudása nélkül is csak fehér kártyákat tart lehetségesnek.

A vizsgálataim alapján úgy tűnik, hogy a *Hanabi* társasjáték leegyszerűsített modelljében egy lépésre előretekintve nem eredményez magasabb pontokat a modális logikai kalkulus beépítése a lépésajánlásba. Feltételezhető, hogy többszereplős játék esetén már nagyobb szerepe lenne az episztemikus modalitás kijelentéseinek, mivel ott a játékosok tudásának megoszlása összetettebb. Ennek a feltevésnek az ellenőrzésére azonban újabb vizsgálatokra lenne szükség.

5.2 Mesterjelölt szint a Hanabi játékban

Mérő László: Észjárások című könyve alapján a nagymesterré válás folyamata hosszú idő. A nagymesterek – bármiben is azok, akár sakokban, vagy más tudományterületeken – alapvetően nem számolnak ki hosszabb változatokat, mint az amatőrök, sokszor inkább csak jobbkor számolnak és pontosabban. Alapvetően azt mondhatjuk, hogy a sémák ismeretének a mennyisége erősen összefügg a szakértelem szintjével. Mesterjelölt tehát az az ember, akinek ismerete az adott területen eléri a néhány ezer sémát. Gondolkodása analitikus, problémáit az ismert sémák segítségével oldja meg lépésről lépésre. Habár az ismert sémák száma folyamatosan nő, kezdetben sokkal nagyobb mértékben, mint egy bizonyos idő eltelte után, amikor a gyorsuló növekedés akadályba ütközik.

A mesterjelölt szint az egyes mesterséges intelligenciát használó alkalmazások tekintetében is megfigyelhető. Az *Észjárások* könyvben a sakkprogramok fejlődése és eredményessége adja a magyarázatot. A programok a kezdő szintek elérését viszonylag rövid időn belül teljesíteni tudják, a „tanulás” első része gyorsabb folyamat. A percenként kiszámított lépések vagy megvizsgált állapotok számának növelésével a gépi játékosok egyre jobb teljesítmények elérésére képesek. Ugyanakkor a mesterjelölt szint elérésénél törés látható a trendvonalban. A szint elérésétől kezdve a vizsgált lépések, sémák számának növeléséhez mérten nem egyenlő mértékben növekszik a program sikeressége. Azaz van egy pont, ahol az erőfeszítések növelése nem feltétlenül jelent lényegesen jobb eredményt. (Mérő, 1994)



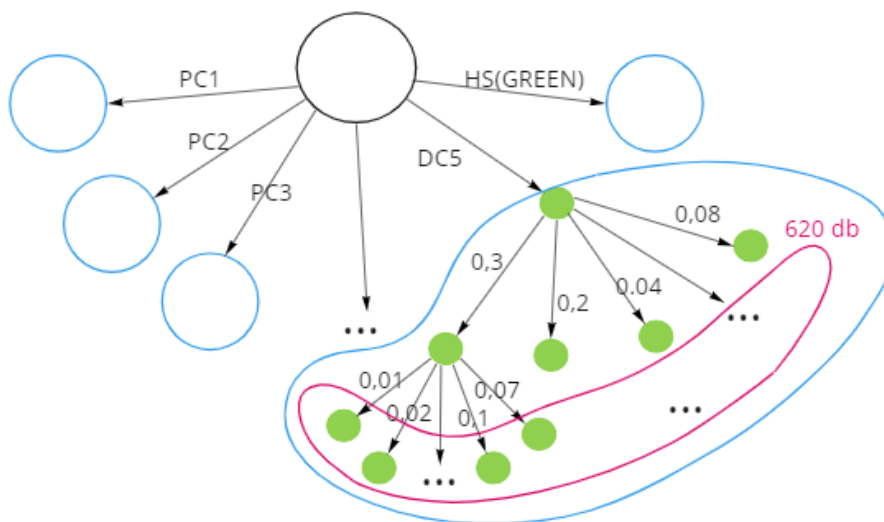
11. ábra. Mérő László: *Észjárások* (1994) – Sakkprogramok fejlődése

Megvizsgáltam, hogy mindezek a *Hanabi* társasjátéknál is megfigyelhetők-e. Az alkalmazásom által szolgáltatott adatok alapján azt feltételezem, hogy a mesterjelölt szint a *Hanabi* játékban a közepes eredmény, azaz a 10-15 pont közötti kategória elérése. Ez az érték viszonylag rövid ideig tartó számításokkal elérhető, akár úgy is, hogy a program mindössze csak egy lépésre előretekintve vizsgálja meg a lehetséges állapotokat.

Egy állapotban legfeljebb 6210 féle szimulátor operátor segítségével megtehető lépés kiértékelésére van szükség. Ezek közül 10 segítségnyújtás lépés, 3100 lapeldobás és 3100 lapkijátszás operátor van. Lapkijátszás és eldobás esetén a lehetséges lapok mindegyikével megvizsgáljuk a létrejött állapotokat.

$$10 + 5 \cdot (20 \cdot 25 + 5 \cdot 24) + 5 \cdot (20 \cdot 25 + 5 \cdot 24) = 6210$$

Lapkijátszás és lapeldobás operátor esetén a kézben tartott és a felhúzendó lapok kombinációjának figyelembevételével 620 féle operátor szimulátor érhető el. A 25 féle lap közül 20 lap helyére bármilyen másik lap felhúzása lehetséges, az 5-ös értékű lapok esetén ugyanolyan színű 5-ös lap nem – mivel abból csak egy van a pakliban –, de 24 másféle lap felhúzása szintén lehetséges. Mind lapkijátszás és mind lapeldobás esetén 5 különböző operátor van, a lapok sorszámától függően.



12. ábra. Példa egy lapeldobás operátor kiszámítására

Egy lépésre előretekintve így viszonylag gyors a számítási művelet. Több, akár csak két lépésre előretekintve is, a kalkuláció lassul, hiszen minden létrejött állapotban újabb 6210 operátor végrehajtásának következményét kell megvizsgálni, ami már 38 564 100 állapotot jelent. Ez azt jelenti, hogy két lépésre előretekintve a következő lépés meghatározása körülbelül 6210-szer annyi időbe telik. Ugyanakkor az így elérhető pontok számában vizsgálataim alapján nem figyelhető meg lényeges növekedés. Átlagosan egy ponttal tudnak az ágensek így jobb eredményt elérni. Visszatekintve a sakkprogramok fejlődését leíró 11. ábrára, a *Hanabi* játék esetében ez jelentheti a trendvonal törését. A számítások gyorsíthatók ugyan, a megvizsgálandó operátorok számát csökkentve, például azoknál a lépéseknél, amelyeknél a játékos ugyanazon információkkal rendelkezik a lapjáról, nem kell mindegyiket kiszámítani. Például ha az első és a harmadik lapjáról is azt tudja, hogy nem pirosak és nem kettesek, akkor a között a két lépés között, hogy leteszi az első vagy leteszi a harmadik lapját, nem tud különbséget tenni, ugyanazon információk alapján ugyanolyan jónak vagy rossznak

gondolja őket. Mivel egyik operátor sem jobb a másiknál, a kettő között véletlenszerűen döntene, így nem szükséges mindkét lépés hasznosságának meghatározása. A játék elején, amikor az ágensnek viszonylag kevés tudással rendelkeznek, ez a fajta egyszerűsítés még sokat jelent, azonban a játék előrehaladtával, ahogy nő a tudásbázis, úgy válik el a lépések azonossága is egymástól. Így a kezdő lépésekben ugyan nem szükséges 6210, csak 1250 lépés kiszámítása, pár segítség után ez a szám hamar növekedésnek indul.

A játék általam elkészített számítógépes megvalósítása jelenleg egy lépésre előretekintve képes olyan eredményeket elérni, mint amelyet a tapasztaltabb emberi játékosok érnek átlagosan egy játék alkalmával. Érdekes kérdés lehet, hogy a lépésajánlást gyorsítva és csökkentve a több lépésre előretekintés erőforrásigényén az alkalmazás milyen pontok elérésére lenne képes.

Az első lehetséges mód a mesterjelölti sémákra vezethető vissza. Ahogy *Mérő László: Észjárások* című könyvében megfogalmazta, a mesterek nem azért jobbak, mert sokat számolnak, hanem mert tudják, mikor kell számolniuk. Ez alapján az alkalmazás lépésajánlása során nem feltétlenül szükséges minden operátor mélyebb szinten való vizsgálata. Elegendő első lépésként az operátorokat egy szintre előretekintve értékelni, majd csak a leginkább ígéretesnek tűnő lépések további szintjeit megvizsgálni. Ezáltal a szintenként kiértékelendő lépések száma nagy mértékben csökkenne. Ennek az elképzelésnek a megerősítésére jó lehet az ilyen módszerek kipróbálása.

Egy másik lehetséges mód a lépésajánló algoritmus megváltoztatása oly módon, hogy az operátorokkal elérhető hasznosság meghatározásakor a hozzájuk tartozó legrosszabb szimulátor operátor(ok) értékét vegye figyelembe, és az így kapott legrosszabb lehetséges lépések közül válassza ki a legjobbnak tűnőt. Ez a megoldás hasonlít a klasszikus versengő kétszemélyes játékok minimax lépésajánló algoritmusára, és emiatt tovább gyorsítható alfa-béta vágással.

Végezetül el lehet gondolkozni a kiértékelt állapotok adatbázisban való tárolásán is, amelynek segítségével ugyanazon játékállásokban nem szükséges újra megvizsgálni a rákövetkező állapotok hasznosságát, mert azok kiolvashatók az adatbázisból. Elég nagy adatbázis felépítése esetén pedig akár több lépésre előretekintve is meglennének a szükséges

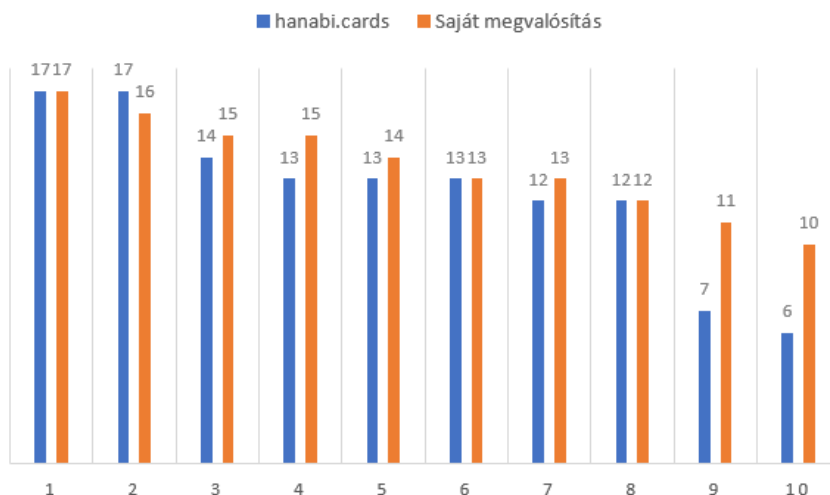
információk és nem lenne szükség azok újra kiszámítására. Ennél a megoldásnál egy-egy állapot reprezentációja fontos lehet a nagy tárigény miatt.

6. Létező implementációk

A *Hanabi* társasjátéknak néhány létező implementációja megtalálható az interneten is, amelyek esetében leggyakrabban emberi játékosok játszhatnak együtt, mint például a *BoardGameGeek* vagy a *hanab.live* oldalán található alkalmazásokban. Vizsgálataim során a *hanabi.cards* oldalon található implementációval foglalkoztam, ahol az emberekkel való együtt játszás mellett, gépi játékkal együtt is meg lehet küzdeni a tűzijáték-készítés nehézségeivel. Elemzésem során olyan kétszemélyes játékmódot teszteltem a felületen, ahol a játékosársam gépi játékos volt. Elsőként azt tanulmányoztam, hogy az itt elérhető alkalmazással, valamint az általam készített megvalósítással játszva milyen pontokat tudok elérni.

A *hanabi.cards* oldalán megtalálható játék során azt tapasztaltam, hogy a gépi játékos sokkal bátrabban cselekszik, mint az én programom gépi ágense. Addig, amíg van letősege hibázni, mert nincs sok viharjelölő felfordulva, akár már egy adott lapra vonatkozó részleges információ (a szín vagy az érték egyike) alapján megpróbálja lehelyezni a kézben tartott lapjait. Úgy gondolja, hogy ha valamelyik tűzijátékra kettes értékű lap következik, akkor valószínű, hogy azért árultam el neki egy lapjának kettes értékét, mert azt egyből le is tudja tenni. Emiatt minden játék során legalább 2 viharjelölőt biztosan szerez a gépi játékos. Amikor már két jelölő megvan, sokkal biztonságosabb játékmódra vált, és a próbálkozásokat elhagyja. A másik érdekesség, hogy a program elsősorban a saját lapjai által cselekszik, viszonylag kevés segítséget ad, inkább várja, hogy a saját lapjaira vonatkozóan rendelkezzen több információval, valamint dobálja a lapjait, hogy újak kerüljenek a kezébe. Amikor segítséget ad, többnyire elvárja, hogy az emberi játékos is akár „félszavakból” értse az utalást a lapjai milyenségére vonatkozóan, és megpróbáljon kockáztatni, amikor még van erre lehetőség. Eszerint, ha például a sárga tűzijátékra kettes érték következik, és segít nekem azzal, hogy valamely lapjaim sárgák, akkor további segítséget a következő körben sem ad, elvárva, hogy merjek kockáztatni és lépni, mert szerinte ennyi információ elég.

Az elemzés során a *hanabi.cards* oldalán is, valamint a saját megvalósításommal is lejátszottam több játékot. Arra voltam kíváncsi, hogy van-e különbség a megszerzett pontok tekintetében az egyik, vagy a másik alkalmazást használva.



13. ábra. Egyes játékokban elért pontok

A vizsgálataim során azt tapasztaltam, hogy a két megvalósítással játszva közel azonos pontok megszerzésére voltam képes. Mindkét esetben ugyanazt a taktikát alkalmazva játszottam, így a játékokban csak a gépi ágens viselkedése volt eltérő. Mindkét esetben megfigyelhető volt, hogy az eredményességhez a lapok sorrendje is hozzájárult, ezáltal a végkimenetet bizonyos mértékben a véletlen is befolyásolta.

Teljesen kezdő játékosok számára mindkét alkalmazás nehézséget okozott addig, amíg a játékosok rá nem éreztek a játék mechanikájára és szabályrendszerére, esetlegesen meg nem tanulták a gépi játékos gondolatmenetét. Tapasztaltabb játékosok számára a szabályok ismerete gördülékenyebb játékmenetet eredményezett. Az egyes alkalmazásokkal elérhető pontok tekintetében lényegi eltérést nem tapasztaltam.

Kutatásom következő szintjén azt elemeztem, hogy a két megvalósítás együtt játszva vajon milyen eredményt tud elérni, mennyire sikeresen tudnak együtt dolgozni. Ehhez a *hanabi.cards* oldalán indítottam kétszemélyes játékot gépi játékos ellen, ahol a saját lépéseimet az általam elkészített alkalmazás segítségével határoztam meg az egyes játékállapotokat átadva a programnak.

A tesztelés során azt tapasztaltam, hogy a *Hanabi* társasjáték két különböző számítógépes megvalósításának együtt játszása nem eredményezett könnyű játékmenetet.

Mivel a *hanabi.cards* gépi játékos minimális segítséget adva – egy játék során átlagosan három-négy segítség operátort használt – várja el az esetleges lapkijátszást, míg az általam készített megvalósítás gépi játékos ennyi információ alapján nem feltétlenül kockáztatja a lépést. A játék ezáltal egy szinten egyhangúvá vált, miszerint az online ágens várja a segítséget és dobálja a lapjait, valamint az információkból megteszi a lapkijátszásokat, az általam készített program ágense pedig folyamatosan információkat szolgáltat, vagy azok hiányában lapeldobással próbálkozik. A két ágens által játszatott játszmák eredménye átlagosan 10 pont körüli volt, aminek ingadozása erősen függött a lapok felhúzásának sorrendjétől. Ha az online játékos kezében jó lapok voltak jó időpontban, a megszerzett pontok száma magasabb volt, ellenkező esetben a játék nehezebbnek bizonyult. Az online alkalmazás és a saját megvalósításom ugyanakkor eltérő gondolkodásmódot alkalmazva is képesek voltak teljes játékok végig játszására közepes körüli eredménnyel. Az ágensek tulajdonképpen különböző emberi veselkedés mintákat írtak le, különböző játékos típusokat jelenítettek meg. Érdekes lehet az az elgondolás, melyben alpból olyan ágensek játszanak együtt, akik eltérően, külön hasznosságfüggvények alapján viselkednek, az egyikük az információ szolgáltató, míg a másik a lapkijátszó szerepet magára öltve. Felmerülhet a kérdés, hogy vajon ez a megvalósítás milyen hatékonyan tudna működni.

Ami leginkább megkülönbözteti az ágensek általi játékot az emberitől bármely megvalósítás esetén, az az emocionális kijelentések hiánya. Emberek között zajló játék esetén – bármennyire is szabálykövető az együtt játszó csapat – a kijelentések hangsúlya és nyomatékossága, valamint a játékosok arcminikája és testbeszéde, esetleges sóhaja plusz információkat árulhat el a csapattársaknak. Például ha egy játék során valamely kézben tartott lapom kijátszására készülök, ám a játékosársam összehúzott szemöldökkel tekint rám, lehet, hogy mégsem teszem meg a lépést. Ennek a megvalósításához az alkalmazásba egy magasabb szintű mesterséges intelligencia beépítésére lenne szükség.

7. Összefoglalás

Dolgozatomban azt a kérdést vizsgáltam, hogy a modális logika eszközei mennyire segítik a játék megnyerését a kooperatív és információhiányos környezettel rendelkező társasjátékok esetén. Ennek eszközeként készítettem el a *Hanabi* társasjáték kétszereplős számítógépes megvalósítását, amely a játék szabályainak megfelelően képes ember-ember, ember-gép és gép-gép elleni játék lebonyolítására. Az alkalmazás segítségével különféle szempontokat figyelembe vevő lépésajánló módszerekkel elemeztem a játszmák sikerességét és hasonlítottam össze a szerzett pontokat. Az elemzés során elért eredmények alapján azt tapasztaltam, hogy a társasjáték általam megvalósított modelljében az episztemikus modalitás beépítése nem vezetett lényegesen sikeresebb végeredményhez a valószínűséget használó lépésajánláshoz viszonyítva. Kétszereplős esetben a tényleges világról szóló állítások közös tudássá váltak, így a kalkulussal kikövetkeztethető új információk valószínűségszámítást alkalmazva is meghatározhatók voltak. A játék többszereplős formájában az ágensek tudása összetettebb lehet más ágensekre vonatkozóan, így ebben az esetben lehetséges, hogy a modális logika eszközei nagyobb szerepet játszanak.

A kezdetben felállított hipotézis vizsgálatához a *Hanabi* társasjáték legegyszerűbb modelljének tesztelését választottam, így a számítógépes megvalósítás jelenleg kétszemélyes játszmák lebonyolítására alkalmas. Az elkészült implementáció jelentős része ugyanakkor újrafelhasználható többszereplős játék létrehozásához is. Módosításra csupán a szereplők számára vonatkozó esetekben lenne szükség, ami kisebb változtatásokkal megvalósítható. Ezen túl a játékhoz készített implementáció egy része akár más, hasonló típusú játékok számítógépes megvalósításához is alkalmazható lenne, mint a dolgozatban korábban említett *Beyond Baker Street: Sherlock Holmes árnyékában* című társasjáték vagy a *Titkos szavak* játék esetén. Az absztrakt osztályok ezeknek a játékoknak a megvalósításához is keretet adnak, a létrehozott kalkulus pedig teljes egészében újrafelhasználható. A játékok állapotteréhez és az operátorok létrehozásához az absztrakt osztályok mellett a megvalósító osztályok egyes részei is átemelhetők a játékok hasonlóságának köszönhetően. A kalkulus hatékony működéséhez a játékspecifikus *HanabiCalculus* osztály bizonyos részei is újrafelhasználhatók ezen játékok esetén is.

A társasjáték számítógépes megvalósítása mellett a *LoTREC* rendszer alapján elkészítettem egy n szereplőre vonatkozó tudáshalmaz vizsgálatára alkalmas modális logikai kalkulus Java nyelven írt implementációját is. Ez egyrészt a további társasjátékok esetén is tökéletes eszközt ad az episztemikus modalitás használatának vizsgálatához, emellett más problémák elemzéséhez is alkalmas. A dolgozat első felében említett *Kozsos gyerekek* probléma a kalkulus segítségével az általam megvalósított környezetben is tesztelhető, így az más modális logikai kérdések vizsgálatához is alkalmas eszközt ad.

A dolgozat megvalósítása során azt tapasztaltam, hogy a téma több kapcsolódási ponttal is rendelkezik egyéb tudományterületekkel. Például az ágensek viselkedését tanulmányozva és összevetve a játék más implementációival, különböző játékos archetípusokat véltem felfedezni. Ezeknek a tanulmányozása a pszichológia területén is érdekes lehet azt elemezve, hogy az egyes ágensek hogyan, milyen szerepet felvéve játszanak végig egy-egy játékmenetet. Emellett a kooperációs helyzetben történő modális logika használatának elemzésével, valamint a játékelmélet kérdéseinek bevonásával a közgazdaságtan számára is érintkezési pontot jelenthet.

8. Irodalomjegyzék

Könyv:

Brian F. Chellas (1980): *Modal Logic*, Cambridge University Press, pp. 1–24.

Olivier Gasquet, Andreas Herzig, Dominique Longin, Mohamad Sahade (2005): LoTREC: Logical Tableaux Research Engineering Companion, Automated Reasoning with Analytic Tableaux and Related Methods, pp. 318–322.

Ronald Fagin, Joseph Y. Halpern, Yoram Moses, Moshe Y. Vardi (2003): *Reasoning About Knowledge*, The MIT Press, pp. 1–37.

A. Kurucz, F. Wolter, M. Zakharyashev, Dov M. Gabbay (2003): *Many-Dimensional Modal Logics: Theory and Applications*, Elsevier, pp. 1–15.

Mérő László (1994): *Észjárások*, Typotex, Budapest, pp. 161–176, 199–225.

Stuart J. Russell, Peter Norvig (2005): *Mesterséges intelligencia modern megközelítésben*, 2. kiadás, Budapest, Panem.

Weboldal:

Antoine Bauza: Hanabi társasjáték, <https://tarsasjatekok.com/tarsasjatek/hanabi-2010> [letöltve: 2021. november].

Hanabi cards online, <https://hanabi.cards/> [letöltve: 2022. február].

Andreas Herzig, Dominique Longin, Mohamad Sahade, Olivier Gasquet: *LoTREC: possible worlds finally made accessible*, 2005, <https://www.irit.fr/Lotrec/> [letöltve: 2021. október].

9. Saját munka leírása

A dolgozatot a témavezetőim útmutatása mellett egyedül készítettem, a benne bemutatásra került *LoTREC* rendszerben készített $S5_n$ stratégia, a *Hanabi* alkalmazás, az ahhoz kapcsolódó tesztek és azok elemzése teljes egészében a saját munkám. Az általam készített forráskódok letölthetők az alábbi címről:

<https://github.com/deiktdk2022/Hanabi>