Takács Márta

Matematikai logika és formális módszerek

$$\begin{array}{lcr} p & \rightarrow & q \\ p & & \\ \hline & & q \end{array}$$

Kiadó
Újvidéki Egyetem
Magyar Tannyelvű Tanítóképző Kar

Fő- és felelős szerkesztő

Lepes Josip

Recenzensek

Tóthné Laufer Edit
Ősz Rita
Gordana Stankov

Tördelőszerkesztő

Takács Márta

CD kiadvány

ISBN 978-86-87095-74-8

A Tudományos-oktatói Tanács a kiadványt az NNV- 21./2016–17/2 számú határozatban egyetemi tankönyvként ismerte el.

Takács Márta

Matematikai logika és formális módszerek

Szabadka, 2017.

Előszó

A tankönyv részben az Újvidéki Egyetem Magyar Tannyelű Tanítóképző Karának mesterszakos hallgatói számára készült a **Matematikai logika** tárgy kísérő anyagaként, másrészt viszont az Óbudai Egyetem mérnök-informatikus szakán tanuló hallgatók **Formális módszerek és matematikai logika az informatikában** című tárgyához nyújt felkészülési segítséget.

A jegyzet két nagy témakört tárgyal: a matematikai logikai alkalmazásokat az informatikában (és nyilvánvalóan azok elméleti alapjait), és néhány formális módszert az informatikában. A korábban két külön tárgyként szereplő tematika mostanra egy tárgy keretein belül kerül összefoglalásra, ezért volt szükségszerű a jegyzet összeállítása.

A tárgy tematikáját illetően nagyon sok forrás áll a hallgatók rendelkezésére, különböző nehézségi szinteken, eltérő jelölési rendszerrel, ezért a jegyzet inkább összefoglaló jellegű, segítve az eligazodást más forrásokban.

A szerző gyakran hagyatkozik a logika témakörében Bérczesné Novák Ágnes, Pásztorné Varga Katalin, Vartrész Magda, Ésik Zoltán, Fülöp Zoltán és munkatársaik jegyzeteire, diasorozataira, példáira, hiszen hosszú ideje foglalkoznak a logika oktatásával, és letisztult tudásanyagot közvetítenek. Ugyanez a helyzet a formális módszerek tekintetében Pataricza András, Majzik István, Bartha Tamás és munkatársaik anyagaival.

A felsorolt források mellett többek között az olvasó figyelmébe ajánlja a szerző a Mesterséges Intelligencia Almanach-ot, amelyet több felsőoktatási intézmény, többek között az Óbudai Egyetem munkatársai a TAMOP – 4.1.2-08/2/A/KMR projekt keretében készítettek el, és amelyben szinte valamennyi érintett tematikával kapcsolatban találhat az olvasó elméleti leírásokat, példákat, feladatokat, illetve két tankönyv elektronikus változatát.

A jegyzet nem kifejezetten ragaszkodik az axióma - tétel - bizonyítás - definíció vonalon való tárgyalásra, gyakrabban tárgyalja a témákat olvasmányos környezetben, példákon bemutatva és példákkal illusztrálva. Visszajelzéseiket, javaslataikat szívesen fogadja a szerző.

A szerző,

Szabadka, 2015.

1 A matematikai logika

A természet nagy könyve a matematika nyelvén íródott.
Galileo Galilei

1.1 A logika helye és szerepe a matematikában

A matematika mai világában fontos szerepe van a logikának. Nem csak a gondolkodásmód modellezése, hanem a tudás rendszerezése, a matematikai elméletek pontos, egyértelműen kezelhető szerkezetbe való foglalása miatt is fontos.

A halmazelmélet és a matematikai logika biztosítja a különböző matematikai témakörök egységes tárgyalásmódját, az elméletek axiomatikus felépítését, a matematikai problémák megoldásának automatikus módszereit.

A matematikai logika keretrendszerében meghatározott helyes következtetési rendszerek az alapjai a matematikai tételek bizonyításának, és ezáltal a modellalkotás képességégnek is.

Történeti szempontból fontos az Arisztotelész[1] által megfogalmazott logikai gondolkodás szabályaitól indulni. Eukleidész[2] axiomatikus rendezése a geometriában, és a logikai szabályokon alapuló bizonyításai máig alapját képezik a matematikai elméletek rendszerbe foglalásának. Ha továbbra is az európai kultúrkörben maradunk, akkor középkort és a reneszánszt követő korszakban történtek előrelépések. Leibniz[3] kidolgozta a formális bizonyítások elméletét, és megalapozta a formális logikát. Boole[4] a logikai formalizmushoz algebrai hátteret vezetett be, de Morgan[5] pedig mindezt más matematikai területeken is érvényesítette, bizonyítások és a matematikai ágazatok formális rendszerbe foglalása érdekében. Frege[6] elsőként adott teljes értékű szimbolikus nyelvet a logika leírására.

[1] Arisztotelész, (i. e. 384–322), görög filozófus
[2] Eukleidész, (i.e. 300 körül született), görög matematikus
[3] Gottfried Wilhelm Leibniz, (1646-1716), német polihisztor
[4] George Boole, (1815-1864), angol matematikus és filozófus
[5] Augustus de Morgan (1806-1871), angol matematikus
[6] Friedrich Ludwig Gottlob Frege, (1848-1925), német matematikus

Kortársaival a XIX. és XX. század fordulóján a logikai formalizmusát és szabályait alapul véve folytatták a matematikai elméletek teljességének, helyességének vizsgálatát, rendszerezését. Hilbert[7], Gödel[8], Tarski[9] éltek ennek a lehetőségével.

Mindeközben a problémamodellezés tovább vitte magát a logikát is a fejlődés és bővítés útján, Lukasiewicz[10] többértékű logikája, Zadeh[11] fuzzy logikája újabb fejezeteket nyitottak. Ugyanakkor a XX. század nagy tudományos robbanása, az informatikai eszközök és a szoftverek fejlődése elválaszthatatlan a logikai szabályoktól, formalizmustól. Néhány említett alkalmazási területtel kapcsolatosan a későbbi fejezetből többet tudhatunk meg.

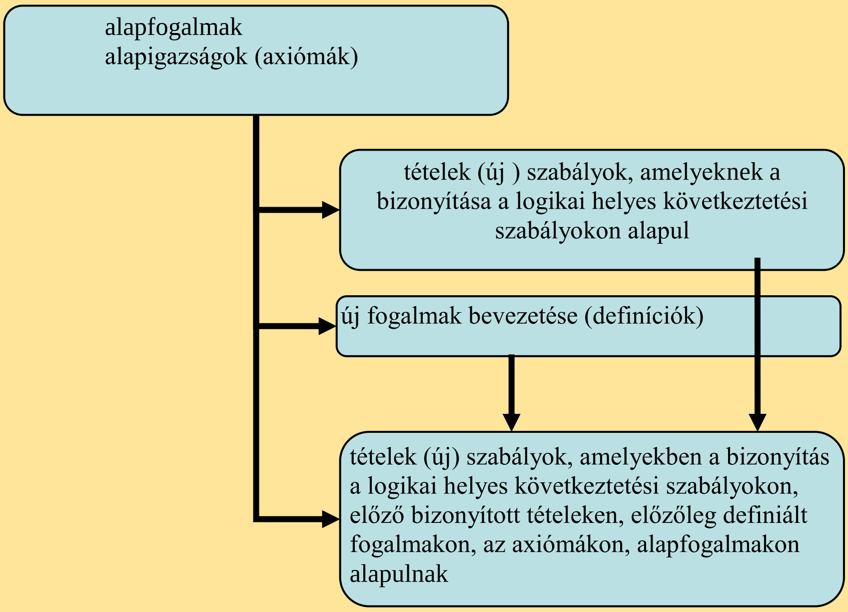
1.2 Predikátum (nulladrendű) logika

A matematika logika az emberi kijelentéseket, következtetéseket hivatott modellezni. A matematikai logikának ma már több ágazata van, az egyszerű kijelentés-logikától kezdődően azon logikákig, amelyek jelzőkkel, időhatározókkal bővített mondatokat képesek modellezni, sőt a bizonytalanság kezelésére is alkalmasak. Mindezek alapja azonban mégis a *predikátum* vagy *kijelentés logika*, amely a legegyszerűbb kijelentések és mondatok modellezésén és igazságértékén alapul.

Ha az alapfogalmakra vonatkozó, alapigazságként elfogadott mondatokból indulunk ki, akkor a predikátumlogika eszköztárával modellezhetjük a helyes következtetési szabályokat, (azokat, amelyek az emberi gondolkodásmód szerint is helyesek), és bizonyíthatunk újabb állításokat az előzőek következményeként. Ha szükség van rá, akkor újabb fogalmak bevezetésével bővíthetjük egy matematikai elmélet eszköztárát, azokra újabb szabályokat láthatunk be. Így épül fel egy matematikai ágazat axiomatikus rendszere.

---

[7] David Hilbert (1862-1943), német matematikus
[8] Kurt Gödel (1906-1978), osztrák matematikus, logikus
[9] Alfred Tarski, lengyel matematikus
[10] Jan Łukasiewicz (1878–1956), lengyel logikus, matematikus
[11] Lofti A. Zadeh (1921-)



1.ábra
Az axiomatikus rendszer felépítése

Ahhoz hogy a kijelentéseink, következtetéseink helyességét megállapítsuk, minden elemi kijelentéshez igazságértéket kell rendelnünk, ami a legegyszerűbb esetben **igaz** vagy **hamis**. A többértékű logika támogatja a bizonytalanság kezelését is, azaz igazságértéket rendel a kijelentéshez akkor is, ha az nem teljesen igaz, vagy ha nem teljesen hamis. Természetesen ezek az igazságértékek az említett két értéktől, az igaztól és a hamistól különböznek.

Kijelentésnek vagy állításnak (predikátumnak) nevezzük azt a mondatot, amelynek egyértelműen meghatározhatjuk az igazságértékét, azaz megmondhatjuk róla, hogy ismereteink szerint a kijelentés igaz-e vagy hamis.

1.1. példa.

Az üres halmaz elemeinek a száma nulla.

Ez a kijelentés igaz.

A 9 prímszám.

Ez a mondat kijelentés, de igazságértéke hamis, mert mint tudjuk, a 9 osztható 1-gyel, 9-cel és 3-mal is (azaz nem csak 1-gyel és önmagával, mint a prímszámok). [12]

[12] Ahhoz, hogy a matematikai logika fogalmaival tisztában legyünk, az Önök korábbi matematikai ismereteire támaszkodunk. Olyan matematikai "mondatokat" mondunk ki, amelyeket ismernek, igazságértéküket meg tudják határozni korábbi tanulmányaik alapján. Mindezek a mondatok természetesen részeit képezik az érintett

A kutya a leghűségesebb állat.

Ez a mondat a predikátum-logikában nem kijelentés, mert egy szubjektív megítélés alapján mondjuk ki az igazságértékét, tehát valaki majd azt mondja rá igaz, más azt, hogy hamis. Hasonlóan nem kijelentések például a következő mondatok:

Budapest szép város.

A barátok mindig segítenek egymásnak.

Ez utóbbi mondat időhatározót is tartalmaz (mindig), ami az egyértelmű igazságérték megadását akkor is nehezítené, ha a mondat többi részének igazságértéke egyértelmű lenne.

Nem lehet meghatározni a felkiáltó- és kérdő mondatok igazságértékét sem.

Azaz nem kijelentések a következők:

Holnap jó idő lesz?

Mindenki álljon fel! *[13]

Felmerül kérdés, hogy mi történik az összetett mondtatok igazságértékével?

Ha két egyszerű, ismert igazságértékű kijelentést egy ***és*** szócskával kötünk össze, akkor azt mikor tekintjük igaznak a hétköznapi életben?

Természetesen akkor, ha mindkettő rész-kijelentés igaz. Egyébként ez az ***és*** összekötőt tartalmazó mondat hamis (azaz hamis, ha az egyik kijelentés igaz a másik hamis, vagy ha mindkettő hamis).

Ha két egyszerű, ismert igazságértékű kijelentést egy ***vagy*** szócskával kötünk össze, akkor azt mikor tekintjük igaznak a hétköznapi életben?

Természetesen akkor lesz igaz, ha legalább az egyik rész-kijelentés igaz (azaz ha az egyik kijelentés igaz a másik hamis, vagy ha mindkettő igaz). Ha mindkét részkijelentés hamis, akkor ez a *vagy* összekötőt tartalmazó mondat hamis.

Korábban beszéltünk arról, hogy a matematika egyik fő erőssége, hogy jól érthető formalizálást alkalmaz. Formalizáljuk a következőket!

1.2. *példa*

---

matematikai ágazat axiomatikus felépítésének, néhányat később újra említünk tételként, és alátámasztjuk a megfelelő definíciókkal.

[13] Jelentse a csillag a példa végét a továbbiakban.

Legyen adott a következő összetett mondat:

A háromszögnek három csúcsa van, és a háromszög csúcsaiban mért szögek összege 180°.

Jelölje p "a háromszögnek három csúcsa van" kijelentést, és q a "a háromszög csúcsaiban mért szögek összege 180°" részkijelentést. Jelölje az *és* összekötőt az $\wedge$ jel.

A

A háromszögnek három csúcsa van, és a háromszög csúcsaiban mért szögek összege 180°.

mondat szimbólumokkal formalizálva: $p \wedge q$.

A fenti mondat mindkét részkijelentése igaz (i), jelöljük ezt így: $\tau(p)=i$.

A τ jel a p mondathoz rendelt (ebben az esetben igaz) i gazságértéket jelenti.

Hasonlóképpen: $\tau(q)=i$ és $\tau(p \wedge q)=i$. ∗

1.3. példa

Az n természetes szám páros vagy az n természetes szám páratlan.

Vezessük be a következő jelöléseket:

r: az n természetes szám páros,

s: az n természetes szám páratlan

Legyen a *vagy* jelölése: $\vee$.

Formalizálva a fenti mondatot: $r \vee s$.

A korábbi ismereteink alapján elmondhatjuk, hogy egy természetes szám valóban vagy páros (osztható kettővel), vagy páratlan, azaz egy páros szám rákövetkezője. A két állítás egyszerre nem lehet igaz, azaz vagy:

$\tau(r)=i$, $\tau(s)=h$ (ahol h a *hamis* igazságértéket jelöli),

vagy $\tau(r)=h$, $\tau(s)=i$, ugyanakkor $\tau(r \vee s)=i$. ∗

A fenti példákban érintettük a matematikai logika szintaxisát is és szemantikáját is, azaz beszéltünk a kijelentés-logika formális nyelvéről, a jelölésrendszerről is (ez a szintaxis), és a formalizált, keretrendszerbe foglalt mondatok értelmezéséről, a jelentésükhöz kapcsolódó igazságértékéről (ezt nevezzük szemantikának).

A szintaxis további bővítésekkel meghatározza a matematikai logika formális nyelvét, a további kijelentések, összetett mondatok, következtetések igazságértékének megadása pedig teljessé teszi a matematikai logika szemantikáját. Tekintettel arra, hogy a kijelentések nem tartalmaznak további, valamely halmazból való elemekre vonatkozó függvényeket, relációkat, ezt a logikát *nulladrendű logikának* is nevezzük, míg a függvényeket is magába foglaló logikát (a következő fejezetben tárgyaljuk) elsőrendű logikának nevezzük.

1.2.1 A kijelentés-logika formális nyelve, szintaxisa

A kijelentés-logikai formális nyelve a következő elemekből épül fel:

– i és h a logikai állandókból (rendre az igaz és a hamis állandót jelöljük így);

– $p,q,r,s,\ldots$ az elemi kijelentéseket jelölő (helyettesítő) logikai változókból;

– $\neg,\vee,\wedge,\rightarrow,\leftrightarrow$ logikai összekötő jelekből (az összetett kijelentések formalizálására szolgálnak), illetve a logikai összekötő jelekből, logikai állandókból és változókból képezett egyszerű logikai (rész)formulákból (amelyek az összetett kijelentéseket hivatottak formalizálni):

$\neg p$ (így olvassuk: nem p);

$p \vee q$ (így olvassuk: p vagy q);

$p \wedge q$ (így olvassuk: p és q);

$p \rightarrow q$ (így olvassuk: ha p akkor q, vagy így: p-ből következik q);

$p \leftrightarrow q$ (így olvassuk: p akkor és csak akkor ha q, vagy így: p ekvivalens q-val).

Az egyszerű formulák egymás utáni véges sok alkalmazásával nyert összetett formulák is formulák.

Az összetett formulaképzésnél a részformulákat a hagyományos értelmezés szerinti zárójelezéssel különíthetjük el egymástól. Ha a formulákat a fenti szabályok alapján szerkesztjük, azt mondjuk *helyesen* szerkesztettük őket.

Megjegyzés: Az elemi kijelentéseket általában az ábécé kisbetűivel jelöljük, az összetett formulákat általában az ábécé nagybetűivel vagy görög betűjelekkel. Például:

A: $(p \vee q) \to s$, B: $((p \to q) \wedge (q \to p)) \leftrightarrow (p \leftrightarrow q)$

Ez utóbbi mondatot így olvassuk: ha p-ből következik q és q-ból következik p, akkor p és q ekvivalensek. Hogy ez mit jelent, azt a szemantika mondja meg.

Gyakran használjuk a formulákban szereplő logikai összekötő jelek (állítások közötti műveletek) következő elnevezéseit:

- $\neg p$, p negáltja ($\neg$ a negáció jele);
- $p \vee q$, (így olvassuk: p, q diszjunkciója, $\vee$ a diszjunkció jele);
- $p \wedge q$ (így olvassuk: p, q konjunkciója, a $\wedge$ a konjunkció jele);
- $p \to q$ (így olvassuk: p implikálja q-t, azaz $\to$ az implikáció jele). Ezekben a mondatokban p-t feltételnek, q-t pedig következménynek nevezzük.
- $p \leftrightarrow q$ (így olvassuk: p ekvivalens q-val, azaz $\leftrightarrow$ az ekvivalencia).

1.2.2. A predikátumlogika szemantikája

Minden helyesen megszerkesztett nulladrendű (kijelentés) logikai formulához hozzárendelhetjük az összes lehetséges állításváltozó-érték szerinti igazságértékét. Ezt a folyamatot a formula kiértékelésének, interpretációjának nevezzük.

A formula olyan kiértékelései, amelyek igaz igazságértéket eredményeznek, a formula modelljének nevezzük.

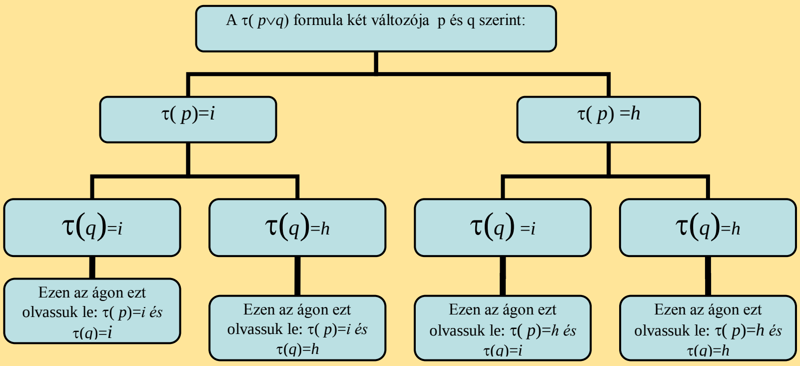
Azt, hogy mely formulához milyen módon rendeljük hozzá az igazságértékét, azaz hogyan értékeljük azt ki, a szintaktikai szabályok alapján megadott formulák szemantikai kiértékelésének nevezzük.

Alapvető szemantikai szabályok:

- Egy kijelentés negáltja a kijelentés igazságértékét az ellenkezőjére váltja.
- Két kijelentés konjunkciója csak akkor igaz, ha mindkét kijelentés igaz.
- Két kijelentés diszjunkciója akkor igaz, ha legalább egyik kijelentés igaz.
- Két kijelentés implikációja csak akkor hamis, ha a feltétel igaz és a következmény hamis.
- Két formula ekvivalenciája akkor igaz, ha a két formula igazságértéke egyenlő (megegyezik).

A kijelentés-logikában egyértelműen meghatározhatjuk, hogy egy formulának hányféle kiértékelése van, hiszen minden logikai ítéletváltozóhoz legfeljebb két igazságérték rendelhető (*i* vagy *h*).

Például a $p \vee q$ formula igazságértékét a kiértékelési fa segítségével így vizsgálhatjuk:



2.ábra
$p \vee q$ formula igazságértékét kiértékelő fa

Láthatjuk, hogy minden egyes *p* szerinti kiértékeléshez további két *q* szerinti kiértékelés tartozik, és minden újabb változóval a kiértékelések száma megkétszereződik, azaz:

Egy *n* ítélet-változót tartalmazó elsőrendű logikai formulának 2^n számú különböző kiértékelése van.

A formulák kiértékelését gyakran adjuk meg táblázatban:

konjunkció

p	*q*	$p \wedge q$
i	*i*	*i*
i	*h*	*h*
h	*i*	*h*
h	*h*	*h*

diszjunkció

p	*q*	$p \vee q$
i	*i*	*i*
i	*h*	*i*
h	*i*	*i*
h	*h*	*h*

negáció

p	$\neg p$
i	*h*
h	*i*

implikáció

p	*q*	$p \rightarrow q$
i	*i*	*i*
i	*h*	*h*
h	*i*	*i*
h	*h*	*i*

ekvivalencia

p	*q*	$p \leftrightarrow q$
i	*i*	*i*
i	*h*	*h*
h	*i*	*h*
h	*h*	*i*

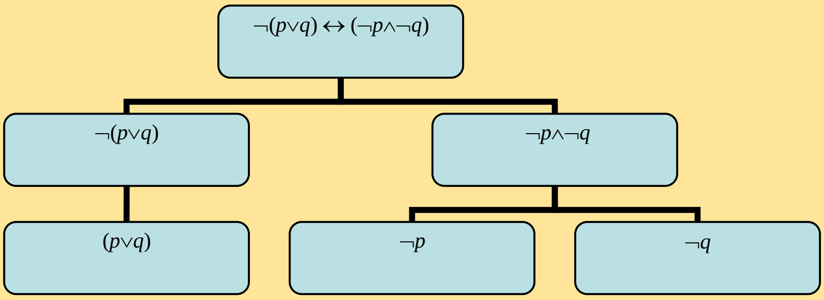
Összetett formulák kiértékelése

Az összetett formulákat részformuláira bontjuk, és lépésről lépésre értékeljük ki.

A részformulára való bontást ugyancsak végezhetjük műveleti fával.

1.4. példa.

Határozzuk meg a $\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$ formula részformuláit, majd ennek alapján a kiértékelési tábláját!

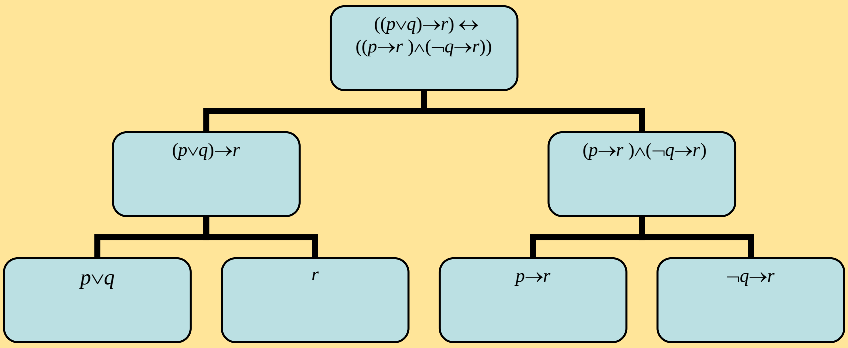


3.ábra
A formula műveleti fája

p	q	$\neg p$	$\neg q$	$\neg p\wedge\neg q$	$p\vee q$	$\neg(p\vee q)$	$\neg(p\vee q) \leftrightarrow (\neg p\wedge\neg q)$
i	*i*	*h*	*h*	*h*	*i*	*h*	*i*
i	*h*	*h*	*i*	*h*	*i*	*h*	*i*
h	*i*	*i*	*h*	*h*	*i*	*h*	*i*
h	*h*	*i*	*i*	*i*	*h*	*i*	*i*

1.5. példa.

Határozzuk meg a $((p \vee q) \rightarrow r) \leftrightarrow ((p \rightarrow r) \wedge (\neg q \rightarrow r))$ formula részformuláit, majd ennek alapján a kiértékelési tábláját!



4.ábra
A formula műveleti fája

p	q	r	$p\vee q$	$(p\vee q)\to r$	$p\to r$	$\neg q$	$\neg q\to r$	$(p\to r)\wedge(\neg q\to r)$	$((p\vee q)\to r)\leftrightarrow((p\to r)\wedge(\neg q\to r))$
i	*i*	*i*	*i*	*i*	*i*	*h*	*i*	*h*	*h*
i	*i*	*h*	*i*	*h*	*h*	*h*	*i*	*h*	*i*
i	*h*	*i*	*i*	*i*	*i*	*i*	*i*	*i*	*i*
i	*h*	*h*	*i*	*h*	*h*	*i*	*h*	*h*	*i*
h	*i*	*i*	*i*	*i*	*i*	*h*	*i*	*h*	*h*
h	*i*	*h*	*i*	*h*	*i*	*h*	*i*	*h*	*i*
h	*h*	*i*	*h*	*i*	*i*	*i*	*i*	*i*	*i*
h	*h*	*h*	*h*	*i*	*i*	*i*	*h*	*h*	*h*

Ha egy formula minden kiértékelése igaz, akkor azonosan igaz formula, azaz **tautológia**.

Ha egy formula minden kiértékelése hamis, akkor azonosan hamis formula, azaz **kontradikció** (ellentmondás).

Ha a formulának van igaz igazságértékű kiértékelése, akkor az adott sorban található igazságértékekre a formulának *modellje van.*

Ha egy formulának van modellje, akkor *kielégíthető.*

Ha egy formulának nincs modellje, akkor *kielégíthetetlen*, azaz sohasem igaz, tehát *kontradikció.*

Az 1.5. példa $\neg(p\vee q)\leftrightarrow(\neg p\wedge\neg q)$ formulája tautológia.

Ha egy formulában vannak igaz és hamis kiértékelések is, akkor az igaz kiértékelésű sorokban olvashatjuk le a formulák **modelljét**.

Az 1.5. példa $((p\vee q)\to r)\leftrightarrow((p\to r)\wedge(\neg q\to r))$ formulájának modelljei a 2., 3., 4., 6. és 7. sorában olvashatók, vagyis például, ha $\tau(p)=i$, $\tau(q)=i$, $\tau(r)=h$ (2. sor).

A logikai műveletek tulajdonságai

Szemantikai kiértékeléssel könnyen bizonyíthatunk néhány alapvető ekvivalenciát a formulák között. Két formula akkor ekvivalens, ha minden egyes kiértékelésükben azonos az igazságértékük.

Az ekvivalens formulák egymással helyettesíthetőek, és ezáltal áttekinthetőbbek, bizonyos helyzetekben könnyebben kezelhetőek lesznek.

Nevezetes ekvivalenciák:

$$(p \vee q) \leftrightarrow (q \vee p) \tag{0.1}$$

$$(p \wedge q) \leftrightarrow (q \wedge p) \tag{0.2}$$

$$((p \vee q) \vee r) \leftrightarrow (p \vee (q \vee r)) \tag{0.3}$$

$$((p \wedge q) \wedge r) \leftrightarrow (p \wedge (q \wedge r)) \tag{0.4}$$

$$(\neg(\neg p)) \leftrightarrow p \tag{0.5}$$

$$(p \wedge i) \leftrightarrow p \tag{0.6}$$

$$(p \wedge h) \leftrightarrow h \tag{0.7}$$

$$(p \vee i) \leftrightarrow i \tag{0.8}$$

$$(p \vee h) \leftrightarrow p \tag{0.9}$$

$$(p \wedge \neg p) \leftrightarrow h \tag{0.10}$$

$$(p \vee \neg p) \leftrightarrow i \tag{0.11}$$

$$(p \rightarrow q) \leftrightarrow (\neg p \vee q) \tag{0.12}$$

$$((p \rightarrow q) \wedge (q \rightarrow p)) \leftrightarrow (p \leftrightarrow q) \tag{0.13}$$

$$\neg(\neg p \vee \neg q) \leftrightarrow (p \wedge q) \tag{0.14}$$

$$\neg(\neg p \wedge \neg q) \leftrightarrow (p \vee q) \tag{0.15}$$

De Morgan[14] azonosságok

$$\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q) \tag{0.16}$$

$$\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q) \tag{0.17}$$

Önálló feladatként elkészíthető bármely ekvivalencia kiértékelési táblázata! Mindegyik tautológia lesz.

Kimondhatjuk a következő ***Lemmát***:

α és β akkor és csak akkor ekvivalens, ha α↔β tautológia.

Bizonyítása: a.) Ha α ekvivalens β akkor α↔β tautológia.

[14] **Augustus de Morgan** (1806-1871), angol matematikus.

Ha α és β igazságértéke megegyezik, akkor az ekvivalencia definíciója miatt csak igaz lehet, azaz tautológia.

b.) Ha α↔β tautológia, akkor a formula csak igaz lehet, de ez pontosan akkor van, ha α és β igazságértéke ugyanaz, vagyis ha α ekvivalens β-vel.◊

Bizonyítás nélkül fogadjuk el a következőket:

Tétel: Ha α tautológia, akkor az ítéletváltozók helyébe formulákat írva is tautológiát kapunk.

Tétel: Ha α tautológia, akkor bármely részformula helyett azzal ekvivalens formulát írva tautológiát kapunk.

1.3 Helyes következtetési szabályok, alkalmazásaik

A matematikában és a hétköznapi élet minden területén fontos a következő típusú mondatoknak modellezni, és logikai szempontból az igazságértéküket kiértékelni:

Ha *feltétel*, akkor *következmény*.

Az ilyen típusú mondatokat következtetési szabályoknak nevezzük, azokat pedig, amelyek minden *igaz* igazságértékű feltételrendszerre *igaz* következtetést eredményeznek, ***helyes következtetési szabályoknak***, helyes következtetési sémáknak nevezzük.

Modellezzük például a következőt:

Ha elfogy a benzin, akkor megáll az autó. Elfogyott a benzin. Ebből következik, hogy megáll az autó.

Az első dolog, hogy elkülönítsük a feltételeket és a következményt.

Ha elfogy a benzin, akkor megáll az autó. Elfogyott a benzin.	Feltételek
Megáll az autó.	Következmény

Ha bevezetjük a következő jelöléseket:

p: elfogy a benzin,

q: megáll az autó,

akkor a fenti következtetési szabályt így írhatjuk fel sematikusan:

$$\begin{array}{l} p \rightarrow q \\ \underline{p \qquad\quad} \\ \qquad\quad q \end{array}$$

Logikai formulában pedig így hangzik:

$$((p \rightarrow q) \wedge p) \rightarrow q$$

Ezt a következtetési szabályt, a p és q állítások valós tartalmától függetlenül mindig igaz, azaz tautológia, és Modus Ponensnek (rövidítve MP) nevezzük.

Megjegyzés: ha több feltétel is szerepel a következetési szabályban, akkor ezeket a feltételeket ÉS kötéssel kapcsoljuk össze, hiszen egyidejűleg mindegyiknek igaznak kell lennie a feltétel teljesüléséhez (igaz igazságértékéhez).

p	q	$p \rightarrow q$	$(p \rightarrow q) \wedge p$	$((p \rightarrow q) \wedge p) \rightarrow q$
i	*i*	*i*	*i*	*i*
i	*h*	*h*	*h*	*i*
h	*i*	*i*	*h*	*i*
h	*h*	*i*	*h*	*i*

Nevezetes helyes következtetési szabályok:

Modus ponens $((p \rightarrow q) \wedge p) \rightarrow q$

$$\begin{array}{l} p \rightarrow q \\ \underline{p \qquad\quad} \\ \qquad\quad q \end{array}$$

Modus tollens (elvető mód, kontrapozíció): $((p \rightarrow q) \wedge \neg q) \rightarrow \neg p$

$$\begin{array}{l} p \rightarrow q \\ \underline{\qquad\quad \neg q} \\ \neg p \end{array}$$

Hipotetikus szillogizmus (feltételes szillogizmus, láncszabály):
$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$

$$\begin{array}{l} p \rightarrow q \\ \underline{q \rightarrow r} \\ p \rightarrow r \end{array}$$

Modus tollendo ponens / diszjunktív szillogizmus (elvéve helyező mód):
$((p \vee q) \wedge \neg q) \rightarrow p$

$$\begin{array}{l} p \vee q \\ \underline{\quad\quad \neg q} \\ p \end{array}$$

Indirekt: $((\neg p \to \neg q) \wedge q) \to p$

$$\begin{array}{l} \neg p \to \neg q \\ \underline{\quad\quad\quad q} \\ p \end{array}$$

Ha végiggondoljuk ezeknek a következtetési szabályoknak az értelmét, beláthatjuk, hogy mi is a hétköznapi életben gyakran ezek alapján vonunk le következtetéseket.
A Modus ponensre már láttunk példát. A matematikában azonban nagyon gyakran alkalmazzuk az indirekt bizonyítást vagy a Modus tollenst, az elvető módot. (Ha bebizonyosodik, hogy rosszul feltételeztünk, akkor elvetjük az eredeti elképzelést.)

1.6. példa

Bizonyítsuk be, hogy a $\sqrt{2}$ irracionális szám (azaz nem racionális)!
Alkalmazzuk a

$$\begin{array}{l} p \to q \\ \underline{\quad\quad \neg q} \\ \neg p \end{array}$$

következtetési szabályt!

Bizonyítás. A racionális számok felírhatók $\frac{a}{b}$ alakú tört alakjában, ahol a és b egész számok, és b nem nulla, azaz ha egy szám racionális, akkor felírható $\frac{a}{b}$ tört alakjában.
Ez a mondat modellezhető az Modus tollens következtetési szabály $p \to q$ feltételével.

Az eredeti állítást (a $\sqrt{2}$ nem racionális szám) bizonyítsuk úgy, hogy a feltételek közé felvesszük a következőt: a $\sqrt{2}$ nem írható fel tört alakjában (ezt könnyen bizonyíthatjuk).

Tegyük fel, hogy ezt megtehetjük: $\sqrt{2} = \frac{a}{b}$, ahol a és b relatív prímek, azaz nincs egynél nagyobb közös tényezőjük, tehát a tört már nem egyszerűsíthető tovább.
Ebből:

$2=\frac{a^2}{b^2}$, azaz $2b^2=a^2$.

Ez viszont azt jelenti, hogy a valaminek a kétszerese, tehát osztható kettővel, azaz $2\cdot 2\cdot k=a^2=2b^2$.

Ez utóbbi egyenlőségláncból:

$2\cdot k=b^2$, tehát b is osztható 2-vel, mi pedig azt feltételeztük, hogy a és b relatív prímek, azaz nincs egynél nagyobb közös tényezőjük. (Ez a rész-bizonyítás maga is egy Modus tollensen alapul).

Visszatérve az eredeti bizonyítandó tételhez, és a Modus tollenshez:

p: a szám racionális $\rightarrow$ *q:* a szám felírható $\frac{a}{b}$ tört alakjában

$\neg$ *q:* a $\sqrt{2}$ **nem** felírható $\frac{a}{b}$ tört alakjában

---

$\neg$*p:* a $\sqrt{2}$ **nem** szám racionális

1.4 Modellelméleti vagy szemantikus következményfogalom

Azt mondjuk, hogy az $\{\alpha^1, \alpha^2, \ldots,\alpha^n\}$ formulahalmaz következménye a β formula, ha minden olyan interpretációban, amelyben az α^1, α^2, …α^n formulák igazak, β is igaz.

Más szavakkal: $\{\alpha^1, \alpha^2, \ldots,\alpha^n\}$ formulahalmaz következménye a β formula, ha β legalább akkor igaz, amikor az α^i-k igazak.

Jelölése: $\{\alpha^1, \alpha^2, \ldots,\alpha^n\} \models_0 \beta$

Megjegyzés: Mivel az elsőrendű logika következményfogalma nem teljesen azonos a nulladrendűével, ezért az indexben szokás azt is jelölni, hogy melyik nyelvről van szó: $\models_0$

Az elsőrendű logikában a következményfogalom jele: $\models_1$

Néhány fontos megjegyzés a szemantikus következményfogalommal kapcsolatban:

Ha β tautológia, akkor minden interpretációban igaz, tehát abban is, amelyekben az α^i –k hamisak. Ezért a tautológia lehet bármely formulahalmaz következménye. Ez indokolja a tautológia jelölését $\models_0 \beta$.

α^i-k közös modellje β-nak is modellje (fordítva az állítás nem igaz).

Tautológia következménye csak tautológia lehet: tautológia $\models_0$ tautológia.

A tautológia bármely α formula következménye: α $\models_0$ tautológia,

Kontradikciónak bármi lehet a következménye (speciális esetben A is és az A tagadása is) : kontradikció $\models_0$ α

Kontradikció csak kontradikciónak lehet következménye (hiszen más formula esetén igaznak kellene lennie ott, ahol a formula igaz): kontradikció $\models_0$ kontradikció.

Azokat a következtetési sémákat tekintjük helyesnek, amelyekben a következmény valóban a feltételek (szemantikai) következménye, így felírható például, hogy a Modus Ponens esetében (leválasztási szabály): $\{\alpha, \alpha\rightarrow\beta\} \models_0 \beta$

Hogy a szemantikus következményfogalom meglétét megmagyarázzuk, ebben az esetben azt kell vizsgálnunk, hogy ahol α és α→β igaz, ott a β igaz-e. Ha igen, akkor helyes a következményfogalom, ha nem, akkor helytelen a következtetési séma. Csak az első interpretációban teljesül, hogy α és α→β igaz. Ebben a interpretációban β is igaz, tehát valóban

$\{\alpha, \alpha\rightarrow\beta\} \models_0 \beta$.

Ítéletváltozók tábázata:

α	β	α→β
i	*i*	*i*
i	*h*	*h*
h	*i*	*i*
h	*h*	*i*

Tétel: $\alpha^1, \alpha^2, \ldots, \alpha^n \models_0 \beta$ akkor és csak akkor, $\alpha^1 \wedge \alpha^2 \wedge \ldots \wedge \alpha^n \models_0 \beta$

Bizonyítás: $\alpha^1, \alpha^2, \ldots, \alpha^n$ együttesen akkor és csak akkor igaz, ha $\alpha^1 \wedge \alpha^2 \wedge \ldots \wedge \alpha^n$ igaz.◊

E tétel alapján a szemantikus következményfogalom feltételét a továbbiakban egyszerűen α-val jelöljük, ahol α-n mindig $\alpha=\alpha^1\wedge\alpha^2\wedge\ldots\wedge\alpha^n$ formulát értjük.

Mutassuk meg most a Modus ponens helyességét úgy, hogy a $\{\alpha \wedge (\alpha\rightarrow\beta)\} \models_0 \beta$ formulát vesszük alapul!

Bizonyítsuk továbbá, hogy az alábbi következtetési sémák is helyesek:

Modus tollens (elvető mód, kontrapozíció): $\{\alpha\rightarrow\beta, \neg\beta\} \models_0 \neg\alpha$

Hipotetikus szillogizmus (feltételes szillogizmus, láncszabály): $\{\alpha\rightarrow\beta, \beta\rightarrow\gamma\} \models_0 \alpha\rightarrow\gamma$

Modus tollendo ponens / diszjunktív szillogizmus (elvéve helyező mód):

$\{\alpha\vee\beta, \neg\beta\} \models_0 \alpha$

Indirekt: $\{\neg\alpha\rightarrow\neg\beta, \beta\} \models_0 \alpha$

Fontos a következő tétel, amely gyakorlati segítséget ad ahhoz, hogy a szemantikus következmény helyességét bizonyíthassuk.

Tétel: $\alpha \models_0 \beta$ akkor és csak akkor, ha $\alpha\rightarrow\beta$ tautológia.

Bizonyítás: a.) ha $\alpha \models_0 \beta$ akkor $\alpha\rightarrow\beta$ tautológia:

a jelölt sor ez esetben nem lehet az igazságtáblában, ugyanis akkor $\alpha\models_0 \beta$ nem teljesülne, hiszen ekkor β-nak legalább akkor kell igaznak lennie, amikor α igaz. A maradék sorokra pedig valóban az *i* az igazságérték.

b.) ha $\alpha\rightarrow\beta$ tautológia, akkor $\alpha \models_0 \beta$:

Ha $\alpha\rightarrow\beta$ tautológia, akkor a fenti igazságtáblában jelölt sor nem szerepelhet, hanem csak a jelöletlen, *i* sorok. Ezekben a sorokban viszont valóban a β legalább ott igaz, ahol az α.◊

1.7. példa: A Modus ponensnek $\{\alpha, \alpha\to\beta\} \models_0 \beta$ megfeleltethető implikáció tautológia, tehát a következtetési séma helyes:

α	β	$\alpha\to\beta$	$(\alpha\wedge(\alpha\to\beta))$	$(\alpha\wedge(\alpha\to\beta)) \to \beta$
i	i	i	i	i
i	h	h	h	i
h	i	i	h	i
h	h	i	h	i

A fenti módszerrel bizonyítható, hogy az alábbi következtetési szabályok helyesek!

Modus tollens: $\{\alpha\to\beta, \neg\beta\} \models_0 \neg\alpha$

Hipotetikus szillogizmus: $\{\alpha\to\beta, \beta\to\gamma\} \models_0 \alpha\to\gamma$

Modus tollendo ponens / diszjunktív szillogizmus (elvéve helyező mód)

$\{\alpha\vee\beta, \neg\beta\} \models_0 \alpha$

Indirekt: $\{\neg\alpha\to\neg\beta, \beta\} \models_0 \alpha$

Tétel: $\alpha \models_0 \beta$ akkor és csak akkor, ha $\alpha\wedge\neg\beta$ azonosan hamis.

Bizonyítás: $\alpha \models_0 \beta$ akkor és csak akkor, ha $\alpha\to\beta$ tautológia, vagyis ha $\neg(\alpha\to\beta)$ kontradikció (azonosan hamis). Ebből $\neg(\alpha\to\beta)=\neg(\neg\alpha\vee\beta)\equiv\neg\neg\alpha\wedge\neg\beta\equiv\alpha\wedge\neg\beta$ ◊

1.8. példa: Modus ponens $\{\alpha, \alpha\to\beta\} \models_0 \beta$ helyes:

α	β	$\alpha\to\beta$	$(\alpha\wedge(\alpha\to\beta))$	$(\alpha\wedge(\alpha\to\beta))\wedge\neg\beta$
i	i	i	i	h
i	h	h	h	h
h	i	i	h	h
h	h	i	h	h

Gyakorlatképpen a fenti módszerrel bizonyítható, hogy az alábbi következtetési szabályok helyesek, azaz a negáltjuk kontradikció!

Modus tollens: $\{\alpha\to\beta, \neg\beta\} \models_0 \neg\alpha$

Hipotetikus szillogizmus: $\{\alpha\to\beta, \beta\to\gamma\} \models_0 \alpha\to\gamma$

Modus tollendo ponens / diszjunktív szillogizmus (elvéve helyező mód)

$\{\alpha \vee \beta, \neg\beta\} \models_0 \alpha$

Indirekt: $\{\neg\alpha \rightarrow \neg\beta, \beta\} \models_0 \alpha$

Tétel: $\{\alpha \vee \beta, \gamma \vee \neg\beta\} \models_0 \alpha \vee \gamma$ (diszjunktív szillogizmus általánosabban).

Bizonyítás.: igazságtáblával, a következők alapján többféleképpen lehet:

a.) a definíció alapján

b.) $\alpha \models_0 \beta$ akkor és csak akkor, ha $\alpha \rightarrow \beta$ tautológia

c.) $\alpha \models_0 \beta$ akkor és csak akkor, ha $\alpha \wedge \neg\beta$ azonosan hamis

A bizonyítást ezen módszerek bármelyikével az olvasó is elvégezheti.◊

Megjegyzés: Az $\alpha \vee \beta$ és $\gamma \vee \neg\beta$ formulák ún. klózok. E két klóz rezolvense $\alpha \vee \gamma$.

A helyes következtetési szabályok minden lehetséges kiértékelése szerinti vizsgálata hosszadalmas, nagy számítási igényű, szemantikus eljárás. Ezért olyan eljárásokat, szintaktikai levezetési formákat kerestek, amelynek segítségével eldönthető, hogy egy formulával leírt következtetési séma helyes-e. A logika ezen elvei még inkább alkalmazást nyertek az automatikus tételbizonyításnál és a PROLOG nyelv megalkotásánál. A következményfogalom eldöntésére a bizonyított tételek alapján az összes interpretációt meg kell vizsgálni, ezért forradalmi jelentőségű a rezolúció alkalmazása, amelyben Robinsonnak[15] 1965-től fontos szerepe volt. A rezolúció alkalmazásához szükség van további formális megközelítésekre, a formulák normalizált alakjára és azokra az alapvető formulákra (axiómákra) illetve levezetési szabályokra, amelyek az automatizmust lehetővé teszik.

1.5 Normálformák

A konjunktív normál forma (KNF) a K^i klózok konjunkciója (i=1,2,...,n).

A klóz literálok diszjunkciója, a literál pedig egy atomi kijelentés, vagy annak tagadása.

Azaz a KNF diszjunkciók konjunkciója, $1\, K^1 \wedge K^2 \wedge \ldots \wedge K^n$, ahol

[15] John Alan Robinson (1928-), amerikai matematikus

$K^i = A^1 \vee A^2 \vee \ldots \vee A^n$, az A^i literál pedig pozitív, ha A, negatív, ha ¬A.

Tétel : Minden formulához létezik vele ekvivalens konjunktív normálforma.

Bizonyítás: megtalálható a (Pásztorné, 2003) forrásban.

1.9. példa

$\alpha \to \beta \equiv \neg\alpha \vee \beta$

De Morgan $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$

$\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$

$\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$

$\alpha \wedge (\beta \vee \gamma) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ *

A diszjunktív normálforma (DNF) konjunkciók diszjunkciója.

A DNF a KNF duálisa, hiszen a $\wedge$ helyett $\vee$, $\vee$ helyett $\wedge$ műveletek szerepelnek.

Feladat. Hozzuk konjunktív normálformára az alábbi formulát!

$\neg[(A \to B) \to ((C \vee A) \to (C \vee B))] \equiv$

$\neg[(\neg A \vee B) \to (\neg(C \vee A) \vee (C \vee B))] \equiv$

$\neg[\neg(\neg A \vee B) \vee (\neg(C \vee A) \vee (C \vee B))] \equiv$

$\neg[\neg(\neg A \vee B) \vee ((\neg C \wedge \neg A) \vee (C \vee B))] \equiv$

$\neg[(A \wedge \neg B) \vee (\neg C \wedge \neg A) \vee (C \vee B)] \equiv$

$\neg(A \wedge \neg B) \wedge \neg(\neg C \wedge \neg A) \wedge \neg(C \vee B)] \equiv$

$(\neg A \vee B) \wedge (C \vee A) \wedge (\neg C) \wedge \neg B$

A klózok:

$K1 = (\neg A \vee B)$

$K2 = (C \vee A)$

$K3 = (\neg C)$

$K4 = \neg B$

1.5.1 Teljes műveleti rendszer

Minden formula kifejezhető a $\vee$, $\neg$ műveletekkel. Ezért azt mondjuk, hogy e műveletek teljes rendszert alkotnak.

A De Morgan azonosságokból azonnal adódik, hogy $\neg(\alpha\vee\beta)\equiv\neg\alpha\wedge\neg\beta$, így $\wedge$, $\neg$, és hasonlóképpen bizonyíthatóan a $\vee$, $\neg$ is funkcionálisan teljes műveleti halmaz. A $\wedge$, $\vee$ műveletekkel viszont nem lehet a $\neg$-t kifejezni, így a konjunkció és diszjunkció együttesen nem alkot teljes rendszert.

Feladat. Bizonyítsa be, hogy a $\rightarrow$ és $\neg$ teljes rendszert alkot! (Hogyan alakítható az implikáció ekvivalens diszjunktív formulává?). Önállóan elvégzendő feladat!

1.5.2 A rezolúció alapelve

Tétel: $\alpha \models_0 W$ akkor és csak akkor, ha $\alpha\wedge\neg W$ kontradikció,

vagyis $\alpha=\alpha^1\wedge\alpha^2\wedge\ldots\wedge\alpha^n$ miatt $\alpha^1\wedge\alpha^2\wedge\ldots\wedge\alpha^n\wedge\neg W$ kontradikció.

(A bizonyítás igazságtáblával elvégezhető.)

Tétel: $\{\alpha\vee\beta, \gamma\vee\neg\beta\} \models_0 \alpha\vee\gamma$.

(A bizonyítás igazságtáblával elvégezhető.)

A fenti két tétel adja meg a az a rezolúció alapelvét, amelyet a továbbiakban így alkalmazunk:

Adott a $\{\alpha^1,\alpha^2,\ldots,\alpha^n\}$ formulahalmaz. E tétel alapján el szeretnénk dönteni, hogy igaz-e: $\{\alpha^1,\alpha^2,\ldots,\alpha^n\} \models_0 W$?

W-t is, és a feltételhalmaz formuláit is konjunktív normálformára hozzuk. Mikor igaz egy KNF? Ha minden benne szereplő klóz igaz.

A klózokban viszont lehetnek negált és negálatlan, azonos atomok. Ezek együttesen nem lehetnek igazak az egész formulában, ezért ezeket a klózpárokból, amelyekben szerepelnek, elhagyjuk, és a „maradékból” egy klózt képezünk, ez a *rezolvens.*

Az így kapott klózzal bővítjük a formulahalmazt. Ezen új formula modelljét (amely interpretációban igaz a formula) keressük.

A gyakorlatban tehát az adott $\{\alpha^1,\alpha^2, \ldots,\alpha^n\}$ formulahalmazhoz el szeretnénk dönteni, hogy annak szemantikus következményfogalma-e a W, azaz $\{\alpha^1,\alpha^2, \ldots,\alpha^n\} \models_0 W$?

Miután W-t is, és a feltételhalmaz formuláit is konjunktív normálformára hozzuk, az $\alpha^1 \wedge \alpha^2 \wedge \ldots \wedge \alpha^n \wedge \neg W$ formula is KNF-ben lesz.

Azt kell tehát megnézni, hogy van-e modellje. A fenti megjegyzés értelmében olyan klózokat keresünk, amelyekben azonos atom pozitív és negatív literálja szerepel. Ezekből a fent leírt módon konstruáljuk az új klózt, a rezolvenst. Az eljárás akkor ér véget, ha egy negált és egy negálatlan literál önmaga alkot egy-egy klózt. Ezek rezolvense az üres klóz, az azonosan hamis klóz (NIL-nek is nevezik). Jele EGY KIS NÉGYZET, □

1.10. példa. Adott $\{P^1, P^1 \rightarrow Q^1, Q^1 \rightarrow Q^2\}$=AB (alap formulahalmaz, egy adatbázis tényhalmaza például)

Kérdés: Q^2 következmény-e ezen formuláknak?

Megoldás: $\{P^1, P^1 \rightarrow Q^1, Q^1 \rightarrow Q^2\}$ a feltételek halmaza, mindegyiket KNF-re kell hozni:

$AB=\{P^1, \neg P^1 \vee Q^1, \neg Q^1 \vee Q^2\}$

$W= Q^2$, tagadása: $\neg Q^2$ (a tagadás itt olyan mint az indirekt feltevés)

Fentiek értelmében azt kell belátni, hogy az $AB \cup \{\neg Q^2\}$ formulahalmaz elemeinek nincsen modellje, nincsen olyan interpretáció, amelyben igaz lehetne.

Ha például P^1 igaz, akkor $\neg P^1$ nem lehet igaz a 2. klózban, ezért mivel minden klóznak igaznak kell lennie, a Q^1 literálnak igaznak kell lennie ezért $\neg Q^1$ ekkor hamis a 3. klózban, ami szerint tehát Q^2 igaz, de ekkor már ellentmondásra jutottunk, hiszen Q^2 és $\neg Q^2$ egyszerre nem lehet igaz. (Azért kellene nekik egyszerre igaznak lenni, mert különböző klózokban szerepelnek, és az egész formula igazságát az összes klóz igaz értéke garantálná). Így azonban nagyon nehéz bizonyítani, hiszen minden lehetséges értékadást végig kellene nézni. Ezt oldja meg a rezolúció gyakorlati alkalmazása.

Igaz-e, hogy $\{P_1, P_1 \rightarrow Q_1, Q_1 \rightarrow Q_2\} \models_0 Q_2$

Ha igaz, akkor az ezzel ekvivalens $(P_1 \wedge P_1 \rightarrow Q_1 \wedge Q_1 \rightarrow Q_2) \rightarrow Q_2$ formula is igaz.

Igaz továbbá az ezzel ugyancsak ekvivalens formula:

$\neg(P_1 \wedge P_1 \rightarrow Q_1 \wedge Q_1 \rightarrow Q_2) \vee Q_2$

A formula negáltja:

$\neg(\neg(P_1 \wedge P_1 \rightarrow Q_1 \wedge Q_1 \rightarrow Q_2)) \wedge \neg Q_2$

azaz a

$(P_1 \wedge P_1 \rightarrow Q_1 \wedge Q_1 \rightarrow Q_2) \wedge \neg Q_2$

formula viszont hamis.

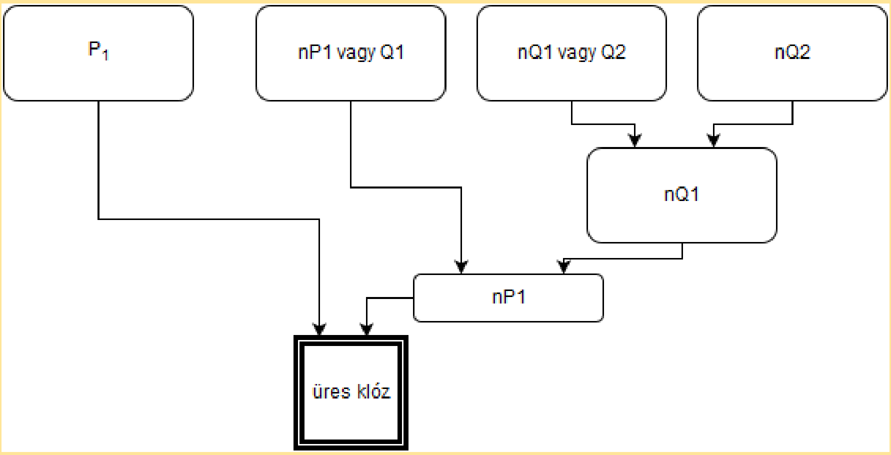
Alakítsuk tovább:

$(P_1 \wedge P_1 \rightarrow Q_1 \wedge Q_1 \rightarrow Q_2) \wedge \neg Q_2$

$P_1 \wedge (\neg P_1 \vee Q_1) \wedge (\neg Q_1 \vee Q_2) \wedge \neg Q_2$

A formula KNF-ban van.

Hamisnak kell lennie, tehát keressünk ellentmondást!



6. ábra
Az 1.10. példa rezolúciós fája[16]

Eredeti klózhalmaznak logikai következménye ($\models_0$) a rezolvenssel bővített klózhalmaz.

Az üres klóz az ellentett literálpárból jön létre.

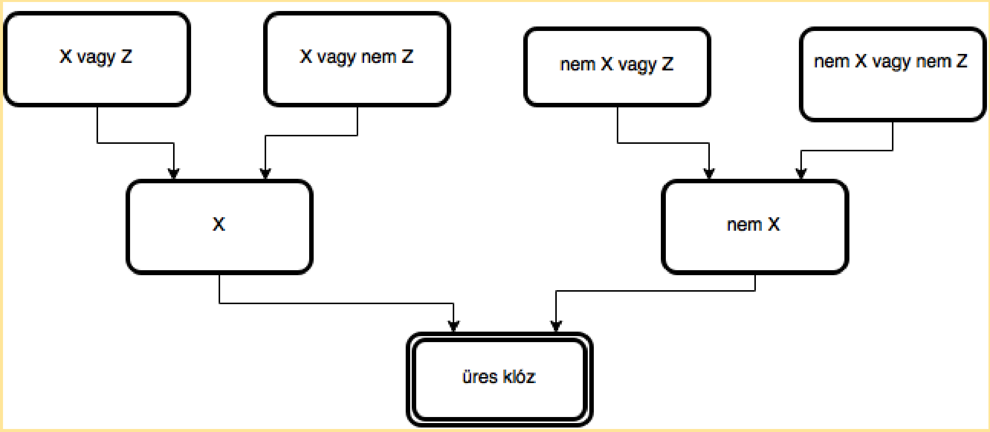
[16] Az ábra elkészítéséhez a

Ha a negált eredeti implikáció normálformájából a rezolvensek kielégítetlen klózhalmazt alkotnak, akkor az eredeti formula igaz.

A rezolúció további lehetőségeket is kínál. Egy formulahalmazról megállapíthatjuk, hogy ellentmondásmente-e. A formulahalmaz formuláit *és* művelettel összekapcsolva, és a formulákat egyenként DNF alakra hozva rezolúcióval megtalálhatjuk az egymásnak ellentmondó formulákat, ha léteznek.

1.11. példa. Ellenőrizzük, hogy ellentmondásos-e a $\{X\vee Z, \neg X\vee Z, X\vee\neg Z, \neg X\vee\neg Z\}$ formulahalmaz!

Vezessük le rezolúciós módszerrel az üres klózt!



11. ábra
Az 1.11. példa rezolúciós levezetése

Ha figyelembe vesszük, hogy a szemantikus következményfogalom átírható implikációvá, és a kapott formulának tautológiának kell lennie a következményfogalom helyességéhez, akkor a rezolúció ezen a feladattípus esetében is alkalmazható.

1.12. példa. Igazoljuk, hogy az alábbi φ formula tautológia!

$\varphi=[(A\to B)\to[(C\vee A)\to(C\vee B)]]$

Megoldás:

φ akkor és csak akkor tautológia, ha $\neg\varphi$ kielégíthetetlen, ezért $\neg\varphi$-t KNF-re írjuk át.

$\neg\varphi=\neg[(A\to B)\to[(C\vee A)\to(C\vee B)]]\equiv\ldots\equiv(\neg A\vee B)\wedge(C\vee A)\wedge(\neg C)\wedge(\neg B)$

Végezze el az olvasó önállóan a rezolúciós levezetést!*

A mindennapjainkban sokszor kell egy tényhalmaz alapján egy új következtetés levonnunk, de ugyanígy működnek a szakértői rendszerek és a logikai programozási paradigma is. Modellezünk és alkalmazzuk a rezolúciót!

1.13. példa. Ha a lóversenyek eredményeit az összeesküvők előre eldöntik, vagy a játéktermeket kezükbe veszik a hamis játékosok, akkor a turizmus kevesebb bevételt hoz, és a város kárt szenved. Ha a turizmus kevesebb bevételt hoz, a rendőrség meg lesz elégedve. A rendőrség sohasem elégedett. Következésképp a lóversenyek eredményeit nem az összeesküvők döntik el[17].

Megoldás: Modellezzük az alap-mondatokat, tényeket, kijelentéseket:

A: a lóversenyek eredményeit az összeesküvők előre eldöntik

B: a játéktermeket kezükbe veszik a hamis játékosok

C: a turizmus kevesebb bevételt hoz

D: a város kárt szenved

E: rendőrség meg lesz elégedve

Az összetett mondatok modelljei tehát a következők:

Ha a lóversenyek eredményeit az összeesküvők előre eldöntik, vagy a játéktermeket kezükbe veszik a hamis játékosok, akkor a turizmus kevesebb bevételt hoz, és a város kárt szenved.

$(A \vee B) \rightarrow (C \wedge D)$

Ha a turizmus kevesebb bevételt hoz, a rendőrség meg lesz elégedve.

$(C \rightarrow E)$

A rendőrség sohasem elégedett.

$\neg E$

Következésképp a lóversenyek eredményeit nem az összeesküvők döntik el.

$\neg A$

A kiinduló feltételhalmaz (formulahalmaz) a $\{(A \vee B) \rightarrow (C \wedge D), (C \rightarrow E), \neg E\}$, a feltételezett következmény pedig $\neg A$. Bizonyítandó tehát, hogy

[17] A feladat forrása http://www.tankonyvtar.hu/hu/tartalom/tamop425/0046_a_matematikai_logika_alkalmazasszemleletu_targyalasa/ch04s04.html

$\{(A\vee B) \to (C\wedge D), (C\to E), \neg E\} \models_0 \neg A$

Készítsük elő a rezolúciót! Átírva igaz implikációra

$((A\vee B) \to (C\wedge D)) \wedge (C\to E) \wedge \neg E) \to \neg A$

KNF alakra hozva a feltételeket

$(\neg (A\vee B) \vee (C\wedge D)) \wedge (\neg C\vee E) \wedge \neg E) \to \neg A$

$((\neg A\wedge\neg B) \vee (C\wedge D)) \wedge (\neg C\vee E) \wedge \neg E) \to \neg A$

A formula első részében a disztributív szabályokat alkalmazva haladjunk tovább

$((((\neg A\wedge\neg B) \vee C)) \wedge ((\neg A\wedge\neg B) \vee D) \wedge (\neg C\vee E) \wedge \neg E) \to \neg A$

$(((\neg A \vee C) \wedge (\neg B \vee C)) \wedge ((\neg A \vee D) \wedge (\neg B \vee D)) \wedge (\neg C\vee E) \wedge \neg E) \to \neg A$

A feltételek KNF alakban vannak (a klózokat kiemeltük). Ha megnegáljuk az eredeti, igaznak vélt formulát, a következmény negáltja is *és*-sel kapcsolódik a feltételekhez, és akkor várhatóan ellentmondásos formulahalmazt, kontradikciót kapunk, azaz ellentmondásos párokat kell keresünk.

$((\neg A \vee C) \wedge (\neg B \vee C)) \wedge ((\neg A \vee D) \wedge (\neg B \vee D)) \wedge (\neg C\vee E) \wedge \neg E \wedge A$

Jelöljük a klózokat és végezzük el a levezetést:

K1: $(\neg A \vee C)$

K2: $(\neg B \vee C)$

K3: $(\neg A \vee D)$

K4: $(\neg B \vee D)$

K5 $(\neg C\vee E)$

K6 $\neg E$

K7: A

K1 és K5 rezolvense R1: $(\neg A \vee E)$

R1 és K6 rezolvense R2 $\neg A$

R2 és K7 rezolvense az üres klóz.

Tehát a negált formula ellentmondásos, az eredeti igaz. *

Rezolúciós stratégiák összefoglaló:

A formulából klózhalmazt (diszjunkciókat) kell létrehoznunk.

Például, mint egy szemantikus következmény levezetésekor:

{formulahalmaz}→F (igaz)

¬ {formulahalmaz} ∨F (igaz)

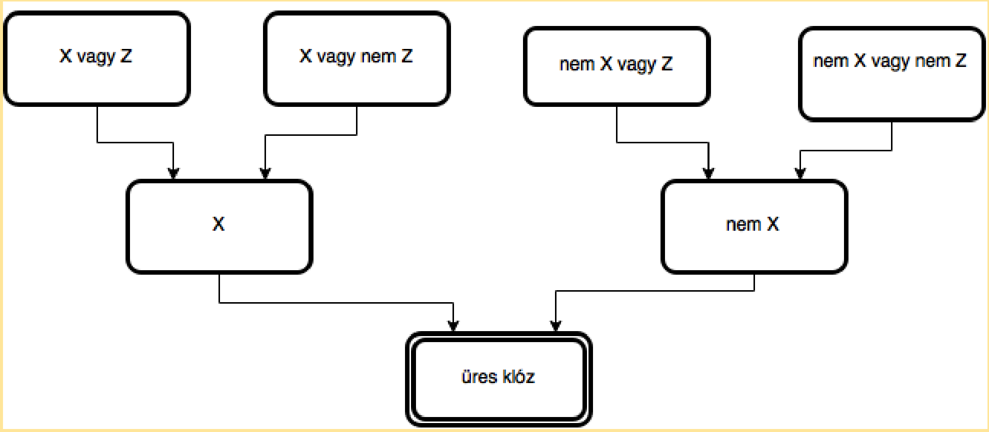
{formulahalmaz} ¬∧F (hamis)

Fontos megjegyzések:

klózpár	rezolvense
X∨Y,¬Y∨Z	X∨Z
X∨ ¬ Y,¬Y∨Z	Nincs, mindkét azonos alapú literál negált
X∨ ¬ Y, Z ∨ ¬V	Nincs, nics azonos alapú literál
¬ X∨ ¬ Y, X∨ Y∨ Z	Nincs, két komplemens literálpár van
X, ¬ X	Üres klóz

Különböző rezolúciós stratégiák léteznek.

1.14. példa. Vezessük le különböző rezolúciós stratégiákkal az üres klózt az *{X∨Z,¬X∨Z,X∨¬Z, ¬X∨¬Z}* formulahalmazból!



Vezessük most le levezetési (lineáris) fával!

{X∨Z,¬X∨Z,X∨¬Z, ¬X∨¬Z}

Használjuk fel sorban a klózokat és a kapott (centrális) →rezolvenseket

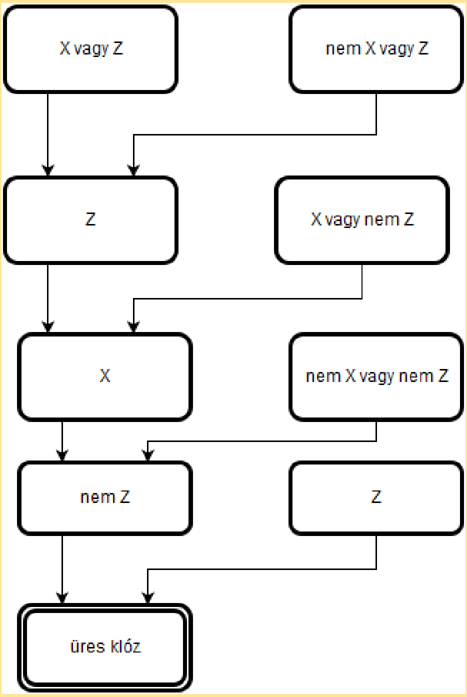
$X \vee Z, \neg X \vee Z, X \vee \neg Z, \neg X \vee \neg Z$

$X \vee Z, \neg X \vee Z, X \vee \neg Z, \neg X \vee \neg Z, Z$

$X \vee Z, \neg X \vee Z, X \vee \neg Z, \neg X \vee \neg Z, Z$

$X \vee Z, \neg X \vee Z, X \vee \neg Z, \neg X \vee \neg Z, Z, X$

$X \vee Z, \neg X \vee Z, X \vee \neg Z, \neg X \vee \neg Z, Z, X$



A lineáris rezolúcióról, ahol a központi klóz újra felhasználható

S klózhalmazból való **rezolúciós levezetés** egy olyan $c_1, c_2, \ldots c_j, \ldots$ klózsorozat, ahol

$c_j \in S$ vagy c_j rezolvense (c_s, c_t) klózoknak, és s,t<j (azaz c_s, c_t "megelőzik" a c_j rezolvenst).

A levezetés **célja az üres klóz**.

Lineáris rezolúciós levezetéskor egy S klózhalmazból egy olyan $p_1, q_1, p_2, q_2 \ldots p_n,$ q_n klózsorozatot számítunk, ahol $p_1, q_1 \in S$.

A további levezetéskor (azaz i=2, 3,...,n esetben) p_i a (p_{i-1}, q_{i-1}) rezolvense ahol:

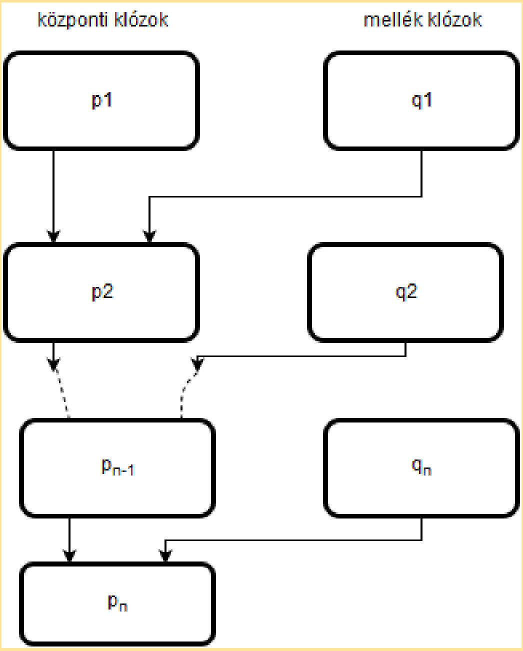
$q_{i-1} \in S$ vagy

($s<i-1$)-re valamely (p_s, q_s) párnak a rezolvense (azaz egy korábban megkapott ***centrális klóz***).

A lineáris rezolúciós levezetést ábrázoló levezetési fa jellegzetes.

Az eredeti S klóz-halmazból a lineáris rezolúcióval levezetett központi klózok csatlakoztathatók a felhasználandó klózokhoz.

Központi klózok, Mellék-klózok



13. ábra
Lineáris input rezolúció

Lineáris input rezolúciós levezetésnél csak egyszer használjuk fel az eredeti klózokat.

Egy S klózhalmazból való **lineáris input** rezolúciós levezetés egy olyan p_1, q_1, p_2, q_2 ... p_n, q_n klózsorozat, ahol $q_n \in S$ (mellék-, vagy oldalklóz) minden i-re. Fontos, hogy $p_1 \in S$, de p_i a p_{i-1},q_{i-1} rezolvense ha $i>2$.

A **lineáris rezolúció** teljes, a **lineáris input**-rezolúció (egység rezolúció) nem teljes.

A helyesség és teljesség fogalma a teljes bizonyítási eljárás létezésének kérdésével kapcsolatos. Egy teljes bizonyítási eljárás egyrészt azt jelenti az összes tétel előfeltételeit mechanikusan elő tudjuk állítani, másrészt azt mondja, hogy a teljes matematika felépíthető alapaxiómák halmazának logikai következményeként. A teljesség kérdésének vizsgálata így a 20. századi matematika néhány legfontosabb eredményének a megszületéséhez vezetett. 1930-ban Gödel bebizonyította az első **teljességi tétel**t (**completeness theorem**) az elsőrendű logikára, megmutatva, hogy minden szemantikailag kikövetkeztetett mondatnak létezik véges szintaktikai bizonyítása. Viszont Robinson publikálta a rezolúciós algoritmust 1965-ben, amellyel ez megvalósítható lett. 1931-ben Gödel bebizonyította a **nemteljességi tétel**t (**incompleteness theorem**). A tétel kimondja, hogy egy logikai rendszer, amely tartalmazza az indukció elvét szükségszerűen nem teljes, azaz léteznek olyan kikövetkeztetett mondatok, amelyeknek a rendszeren belül nincs véges bizonyítása.

1.15. példa. Vezessük le lépésenként lineáris rezolúcióval az üres klózt a formula-halmazból !

$\{X\vee Z, \neg X\vee Z, X\vee\neg Z, \neg X\vee\neg Z\}=F$

1. $X\vee Z$ mert $X\vee Z \in F$
2. $\neg X\vee Z$ mert $\neg X\vee Z \in F$
3. Z mert 1. és 2. rezolvense
4. $X\vee\neg Z$ mert $X\vee\neg Z\in F$
5. X mert 3. és 4. rezolvense
6. $\neg X\vee\neg Z$ mert $\neg X\vee\neg Z\in F$
7. $\neg Z$ mert 5. és 6. rezolvense
8. ◊ mert 3. és 7. rezolvense

A következő kérdés az, hogy mely formulahalmazból lehet, érdemes lineáris rezolúciót szerkeszteni.

1.5.3 Horn formulák és Hilbert axiómák

A Horn[18] formula egy olyan $F = C_1 \wedge ,\ldots, \wedge C_n$ konjunktív normálforma, melyben minden C_i klóz legfeljebb egy pozitív literált tartalmaz.

Megjegyzés

$\neg q_1 \vee \ldots \neg\vee q_k \vee q \equiv (q_1 \wedge \ldots \wedge q_k) \rightarrow q$

$\neg q_1 \vee \ldots \neg\vee q_k \equiv (q_1 \wedge \ldots \wedge q_k) \rightarrow$ igaz

Ha k = 0, akkor $q \equiv$ igaz $\rightarrow q$ és hamis $\equiv$ igaz $\rightarrow$hamis, ahol $\equiv$ a formulák ekvivalenciáját jelenti.

Olyan formulákat tekintünk, amelyekben nem fordul elő $\wedge, \vee, \leftrightarrow, \neg$ és az *igaz* (jelölheti $\uparrow$ is).

Hilbert **axiómák:**

- 1. $(F \rightarrow (G \rightarrow H)) \rightarrow ((F \rightarrow G) \rightarrow (F \rightarrow H))$
- 2. $F \rightarrow (G \rightarrow F)$
- 3. $((F \rightarrow h) \rightarrow h) \rightarrow F$, ahol h a *hamis*-t jelöli

ahol *F, G, H* tetszőleges formulák.

Következtetési szabály: leválasztás, (Modus ponens)

$$\frac{F,\quad F \rightarrow G}{G}$$

ahol *F, G* tetszőleges formulák.

Az érvényesség Hilbert-féle bizonyításának vagy levezetésnek nevezzünk egy olyan $F_1, F_2, \ldots. F_n$ sorozatot, ahol az F_i formulák mindegyike

1. axióma, vagy
2. előáll az őt megelőző formulákból leválasztással.

Azt mondjuk, hogy *F* bizonyítható, vagy levezethető, ha létezik olyan $F_1, F_2, \ldots. F_n$ bizonyítás, melyre $F = F_n$. Hasonlóképpen egy Σ formulahalmazra:

[18] Alfred Horn (1918 –2001), amerikai matematikus

A Σ formulahalmaz feletti Hilbert-féle bizonyításának vagy levezetésnek nevezzünk egy olyan $F_1, F_2, \dots. F_n$ sorozatot, ahol az F_i formulák mindegyike

1. axióma, vagy
2. Σ -beli formula vagy
3. előáll az őt megelőző formulákból leválasztással.

Azt mondjuk, hogy F bizonyítható, vagy levezethető a Σ felett, ha létezik olyan F_1, $F_2, \dots. F_n$ bizonyítás, melyre $F = F_n$. Jelölése: $\Sigma \vdash_H F$.

1.16. példa. Bizonyítsuk be, hogy $\vdash_H F\to F$, azaz hogy levezethető az $F\to F$ formula, bármely F formulára

Bizonyítás. Legyen $G = F \to F$.

1. $F \to (G \to F)$, (Ax 2)
2. $F \to G$, (Ax 2)
3. $(F \to (G \to F)) \to ((F \to G) \to (F \to F))$, (Ax 1)
4. $(F \to G) \to (F \to F)$, (1. és 3., MP-szel)
5. $F \to F$, (2. és 4., MP-szel)

Tehát tetszőleges F-re: $\vdash_H F\to F$

Megjegyzés: további példák találhatók az (Ésik, 2011.) anyagban.

1.5.4 Helyesség és teljesség (soundness & completeness)

Összefoglaló a szemantikai következményfogalomról:

Legyen P zárt formulahalmaz. F lezárt formula a P logikai következménye, akkor és csak akkor, ha F igaz P minden modelljében. Jelölése: $P \models F$

A szintaktikus levezethetőségről:

Ha az F következményt formális úton, a következtetési szabályok alkalmazásával állítjuk elő, következtetési lépések sorozatán keresztül növelve a premissza-formulák P halmazát, akkor F levezethető P-ből. Jelölése: $P \vdash F$.

Ennek tudatában tárgyalhatunk a helyességről és a teljességről.

Következtetési szabályok valamely halmaza **helyes** (sound), ha minden P zárt formulahalmazra és F zárt formulára, ha $P \vdash F$, akkor $P \models F$.

Következtetési szabályok valamely halmaza **teljes** (complete), ha minden P zárt formulahalmazra és F zárt formulára, ha $P \models F$, akkor $P \vdash F$.

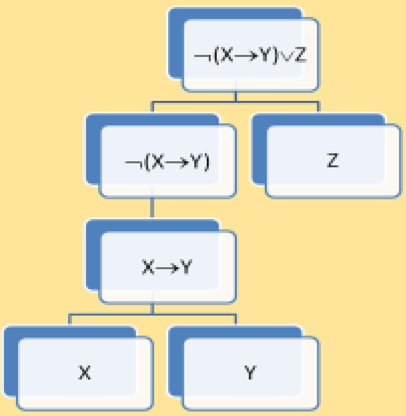
Hilbert rendszerének helyességét a és teljességét a következőképpen mondhatjuk ki: tetszőleges formula halmazra és F formulára $\models F$, akkor es csak akkor, ha $\vdash_H F$.

1.6 Fák a logikai problémamegoldásban

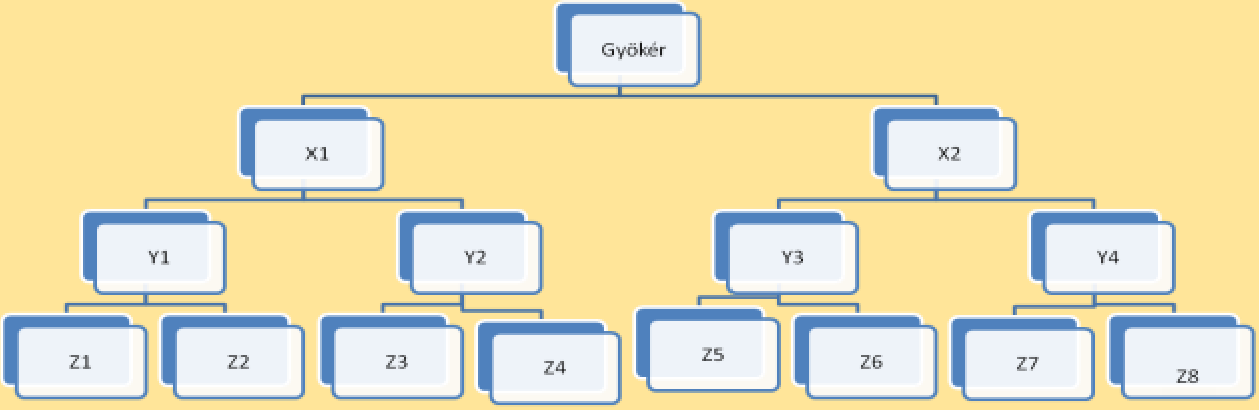
Egy F formula szerkezeti fája egy olyan véges rendezett fa, melynek csúcsai formulák:

- a gyökere F,
- $\neg A$ csúcsának pontosan egy gyermeke van, az A formula
- az $A*B$ csúcsának pontosan két gyermeke van, rendre az A és B formulák (ahol * logikai művelet)
- levelei elemi (vagy prím, vagy atomi) formulák.

1.17. példa. Rajzoljuk meg a $\neg(X \to Y) \vee Z$ formula szerkezeti fáját!



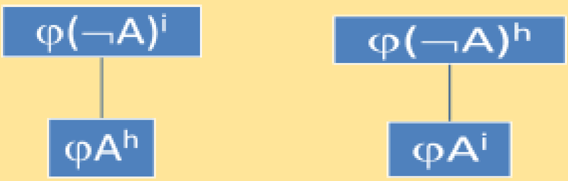
1.18. példa. Az X,Y,Z ítéletváltozókat tartalmazó formula általános szemantikus fájának levelei maguk az ítéletváltozók.



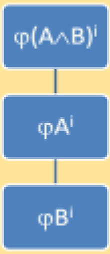
Legyen A tetszőleges nulladrendű formula. Az A interpretációira vonatkozó φA^i és φA^h feltételeket a szerkezeti rekurzió alapján így határozzuk meg:

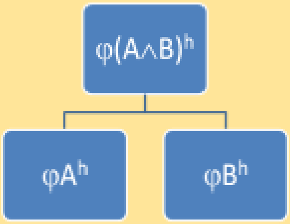
a) Ha A atomi formula, akkor φA^i feltételt azok az I interpretációk teljesítik, amelyekben $I(A)=i$, a φA^h feltételt pedig azok, amelyekre $I(A)=h$.

b) a $\varphi(\neg A)^i$ feltételt azok az I interpretációk teljesítik, amelyekben a φA^h feltétel teljesül.

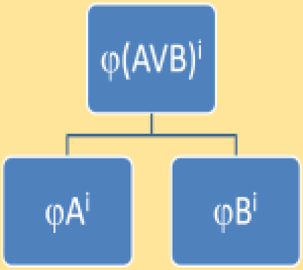


c) a $\varphi(A \wedge B)^i$ feltételt azok az I interpretációk teljesítik, amelyekben a φA^i és a φB^i feltételek együttesen teljesülnek.

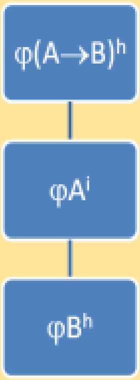


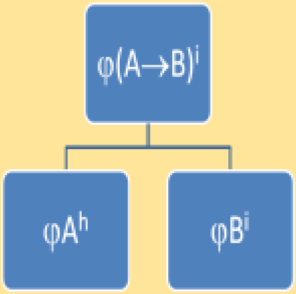


d) a $\varphi(A \vee B)^i$ feltételt azok az I interpretációk teljesítik, amelyekben a φA^i vagy a φB^i feltételek teljesülnek.



e) A $\varphi(A \rightarrow B)^i$ feltételt azok az I interpretációk teljesítik, amelyekben a φA^h vagy a φB^i feltételek teljesülnek.





1.19. példa. Adjuk meg az (Y∨Z) ∧(Z¬→X) formula igazságtábláját, majd a kiértékelési fáját, és vegyük észre az összefüggéseket!

X	Y	Z	¬X	Z¬→X	(Y∨Z)	(Y∨Z)∧(Z¬→X)
i	*i*	*i*	*h*	*h*	*i*	*h*
i	*i*	*h*	*h*	*i*	*i*	*i*

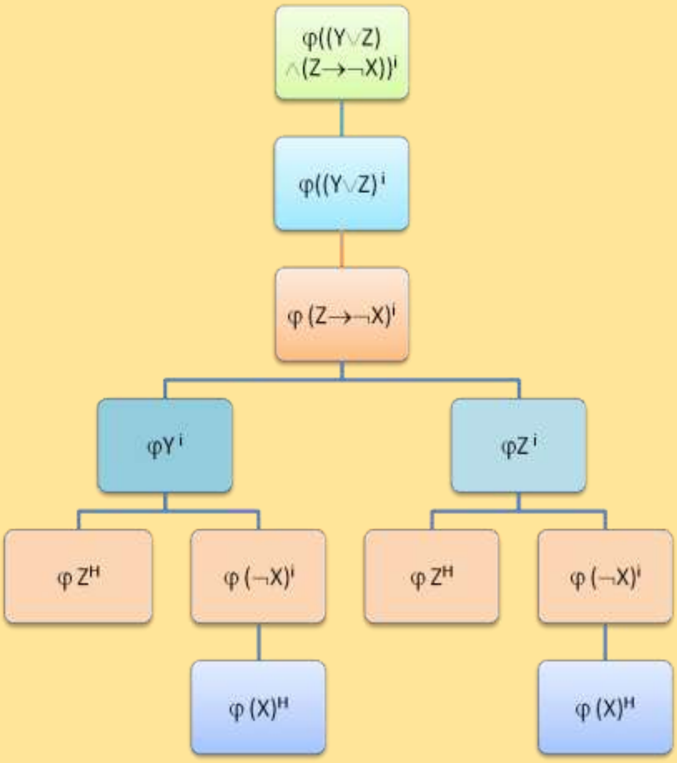
i	*h*	*i*	*h*	*h*	*i*	*h*
i	*h*	*h*	*h*	*i*	*h*	*h*
h	*i*	*i*	*i*	*i*	*i*	*i*
h	*i*	*h*	*i*	*i*	*i*	*i*
h	*h*	*i*	*i*	*i*	*i*	*i*
h	*h*	*h*	*i*	*i*	*h*	*h*

A kiértékelési táblát egy úgynevezett lusta táblával is leírhatjuk:

X	Y	Z
-	*i*	*h*
h	-	*i*
h	*i*	-

Ezek szerint a formula modelljei: *{(i,i,h),(h,i,i),(h,i,h),(h,h,i)}*

Az (Y∨Z) ∧(Z¬→X) formula igazságértékelés-fája:[19]



A különböző részformulák kiértékelési ága más-más színnel van jelölve.

[19] A feladat forrása: Pásztorné Varga Katalin , Várterész Magda, *A matematikai logika alkalmazásszemléletű tárgyalása* című könyve, 41. oldal

A bal oldali ágon visszafelé haladva azt látjuk, hogy Z hamis és Y igaz igazságértéke mellet X igazságértékétől függetlenül modellje adódik a formulának, azaz mondhatjuk, hogy teljesíthető, és ezt a lusta tábla is igazolta.

Látjuk azt is, hogy a jobboldali ág bal ágán egyszerre lenne Z kiértékelése igaz és hamis, ez az ág mutatja tehát, hogy ezen igazságértékek mentén a formula teljesíthetetlen.

Mit láthatunk még a táblából? A formula akkor igaz, ha a $\{(i,i,h),(h,i,i),(h,i,h),(h,h,i)\}$ rendezett hármasokból álló halmazban, ahol a rendezett hármasok rendre a (φX, φY, φZ) igazságértékeket tartalmazzák, legalább egy igaz, azaz ha $(i,i,h)\vee(h,i,i) \vee (h,i,h) \vee (h,h,i)$. Ha (i,i,h) azt jelenti, hogy $\varphi X^i \wedge\varphi Y^i\wedge\varphi Z^h$, akkor ez a konjunkció akkor igaz, ha $\varphi X^i \wedge\varphi Y^i\wedge\varphi\neg Z^i$. Ennek alapján a formula akkor igaz, ha

$(X\wedge Y\wedge\neg Z) \vee(\neg X\wedge Y\wedge Z)\ (X\wedge Y\wedge\neg Z) \vee(\neg X\wedge Y\wedge Z) \vee(\neg X\wedge Y\wedge\neg Z) \vee (\neg X\wedge\neg Y\wedge Z)$.

Így egy diszjunktív normálformát (DNF) állítottunk elő az igazságtáblából.

A konjunktív normálforma (KNF) előállítása az igazságtáblából a kontradikció lehetőségeiből adódik, tehát:

$((Y\vee Z) \wedge(Z\neg\rightarrow X))\leftrightarrow((\neg X\vee\neg Y\vee\neg Z) \wedge (\neg X\vee Y\vee\neg Z) \wedge (\neg X\vee Y\vee Z) \wedge (X\vee Y\vee Z))$

X	Y	Z	¬X	Z¬→X	(Y∨Z)	(Y∨Z) ∧(Z¬→X)		
i	*i*	*i*	*h*	*h*	*i*	*h*	*	¬X ∨ ¬Y ∨ ¬Z
i	*i*	*h*	*h*	*i*	*i*	*i*		
i	*h*	*i*	*h*	*h*	*i*	*h*	*	¬X ∨ Y ∨ ¬Z
i	*h*	*h*	*h*	*i*	*h*	*h*	*	¬X ∨ Y ∨ Z
h	*i*	*i*	*i*	*i*	*i*	*i*		
h	*i*	*h*	*i*	*i*	*i*	*i*		
h	*h*	*i*	*i*	*i*	*i*	*i*		
h	*h*	*h*	*i*	*i*	*h*	*h*	*	X ∨ Y ∨ Z

A konjunktív normálformát egyszerűsíthetjük a rezolúciós elv alapján. Ha

$((Y\vee Z) \wedge(Z\neg\rightarrow X)) \leftrightarrow$

$((\neg X \vee\neg Y \vee\neg Z) \wedge (\neg X \vee Y \vee\neg Z) \wedge (\neg X \vee Y \vee Z) \wedge (X \vee Y \vee Z))$

Mutassuk meg igazságtáblával is az ekvivalenciát!

X	Y	Z	$(\neg X \vee \neg Y \vee \neg Z)$	$\neg X \vee Y \vee \neg Z$	$(\neg X \vee Y \vee Z)$	$(X \vee Y \vee Z)$	*KnF*	$(Y \vee Z) \wedge (Z \neg \rightarrow X)$	*ekvivalencia*
i	i	i	h	i	i	i	*h*	*h*	*i*
i	i	h	i	i	i	i	*i*	*i*	*i*
i	h	i	i	h	i	i	*h*	*h*	*i*
i	h	h	i	i	h	i	*h*	*h*	*i*
h	i	i	i	i	i	i	*i*	*i*	*i*
h	i	h	i	i	i	i	*i*	*i*	*i*
h	h	i	i	i	i	i	*i*	*i*	*i*
h	h	h	i	i	i	h	*h*	*h*	*i*

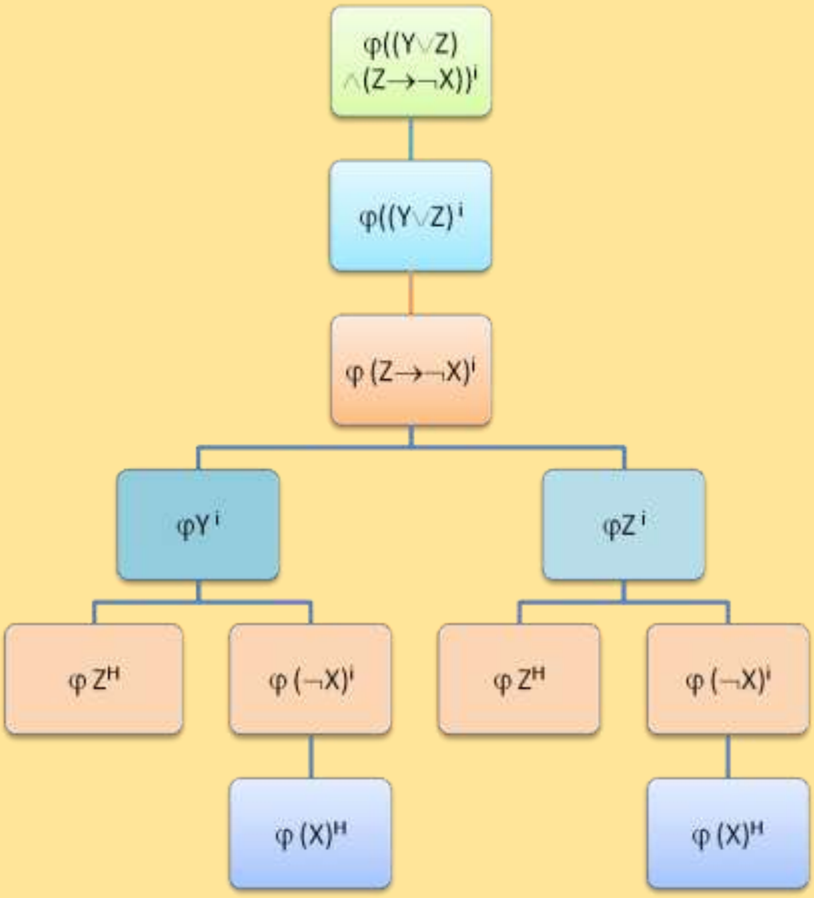
A rezolúciós elv alapján is levezethető mindkét formulából ugyanaz a rezolvens.

$(\neg X \vee \neg Y \vee \neg Z) \wedge (\neg X \vee Y \vee \neg Z) \wedge (\neg X \vee Y \vee Z) \wedge (X \vee Y \vee Z)$

azaz K1 és K2 és K3 és K4. K1 és K2 rezolvense $(\neg X \vee \neg Z)$, K3 és K4 rezolvense $(Y \vee Z)$. A két rezolvens rezolvense $(\neg X \vee Y)$, ami ekvivalens az eredeti $((Y \vee Z) \wedge (Z \neg \rightarrow X)$ formulából levezethető rezolvenssel.

X	Y	Z	$\neg X$	$Z \neg \rightarrow X$	$(Y \vee Z)$	$(Y \vee Z) \wedge (Z \neg \rightarrow X)$	$\neg X \vee Y$	$(Y \vee Z) \wedge (Z \neg \rightarrow X) \rightarrow \neg X \vee Y$
i	*i*	*i*	*h*	*h*	*i*	*h*	*i*	*i*
i	*i*	*h*	*h*	*i*	*i*	*i*	*i*	*i*
i	*h*	*i*	*h*	*h*	*i*	*h*	*h*	*i*
i	*h*	*h*	*h*	*i*	*h*	*h*	*h*	*i*
h	*i*	*i*	*i*	*i*	*i*	*i*	*i*	*i*
h	*i*	*h*	*i*	*i*	*i*	*i*	*i*	*i*
h	*h*	*i*	*i*	*i*	*i*	*i*	*i*	*i*
h	*h*	*h*	*i*	*i*	*h*	*h*	*i*	*i*

Hol látszik a $((Y \vee Z) \wedge (Z \neg \rightarrow X)) \rightarrow (\neg X \vee Y)$ a fában?



Bizonyítsuk be, hogy $\{(Y\vee Z), (Z\rightarrow\neg X)\} \models (\neg X \vee Y)$, használjunk lineáris rezolúciót!

- $\{(Y\vee Z), (Z\rightarrow\neg X)\} \models (\neg X \vee Y)$ igaz, akkor
- $(Y\vee Z)\wedge(Z\rightarrow\neg X) \rightarrow (\neg X \vee Y)$ igaz
- $(Y\vee Z)\wedge(\neg Z\vee\neg X) \rightarrow (\neg X \vee Y)$ igaz
- $\neg((Y\vee Z)\wedge(\neg Z\neg\vee X)) \vee (\neg X \vee Y)$ igaz
- $\neg (\neg((Y\vee Z)\wedge(\neg Z\neg\vee X)) \vee (\neg X \vee Y))$ hamis
- $(Y\vee Z)\wedge(\neg Z\neg\vee X) \wedge \neg (\neg X \vee Y)$ hamis
- $(Y\vee Z)\wedge(\neg Z\neg\vee X) \wedge X \wedge \neg Y)$ hamis

A Klózok:

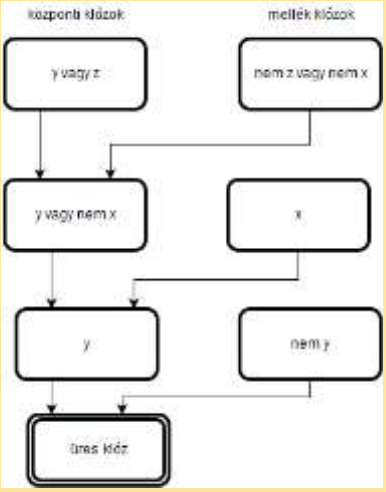
$Y\vee Z$

$\neg Z\neg\vee X$

X

$\neg Y$

Két párosítható klózzal indítunk, és a kapott rezolvenshez egy újabb, még nem felhasznált klózt párosítunk.



1.7 Halmazelmélet és a logika

A matematika egyik leggyakorlatiasabb, a tapasztalati forrásokhoz legközelebb eső ága a halmazelmélet. A matematikai logika mellé a XIX. század második felében Georg Cantor[20] alapozta meg a matematikai elméletek axiomatikus felépítésének e fontos alapját. Az általános halmazelmélet arról szól, hogy a halmazokat függetlenítjük elemeinek tulajdonságaitól, azoktól elvonatkoztatjuk, mert akkor általános összefüggéseket mondhatunk ki, amelyek bármely halmazra vonatkozhatnak.

A halmazt ugyan alapfogalomnak tekintjük, de meghatározhatjuk például a következőképpen:

A halmaz olyan egyedek sokasága, amelyeket egy vizsgált tulajdonság szerint egy csoportba sorolhatunk. A halmazt alkotó egyedeket a halmaz elemeinek nevezzük.

Halmazt képezhetnek például a számok, az egy évfolyamra járó hallgatók, egy könyv fejezetei, és bármi más csoportosulás. Ugyanakkor láthatjuk, hogy bármi körülöttünk egyed, mi magunk is azok vagyunk, elemei különböző halmazoknak. Ha matematikai szempontból vizsgálódunk, akkor fontos, hogy behatároljuk, mely

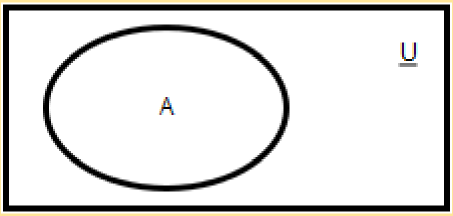
[20] Georg Cantor (1845-1918), Oroszországban született, Németországban tanult és dolgozott

univerzum elemeivel foglalkozunk. Tehát ha a számok világát vizsgáljuk, akkor nincs szükség az emberek halmazának vizsgálatra, figyelembe vételére.

Egy vizsgált halmazt a saját univerzumában figyelünk meg.

A halmazokat a leggyakrabban Venn[21] diagrammok segítségével vizualizáljuk. A halmaz elemeit görbe zárt vonallal határoljuk, és ha ezt egyértelművé kell tenni, akkor egy téglalapba rajzoljuk, amely az univerzumot hivatott jelölni (2.1. ábra).

A halmazokat és az univerzumot (U) általában az ábécé nagybetűivel, a halmaz elemeit kisbetűkkel jelöljük.

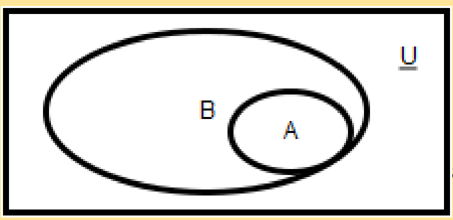


A halmaz elemeit berajzolhatjuk a Venn diagramba, de megadhatjuk leírással vagy felsorolással is.
Ha a halmaznak véges sok eleme van, akkor a felsorolást kapcsos zárójelbe tesszük.
Ha egy halmazban nincs elem, akkor *üres halmaznak* nevezzük és $\emptyset$ vagy $\{\}$ módon jelöljük.
Ha egy x elem az A halmaz eleme, akkor ezt így jelöljük: $x \in A$. Ha egy y elem az A halmaznak nem eleme, akkor ezt így jelöljük: $y \notin A$, vagy $\neg y \in A$.
Két halmaz egyenlő, ha elemeik egyenlők, azaz ha egy elem benne van az egyik halmazban, akkor benne van a másikban is, és fordítva. Formalizálva, a matematikai logika formalizmusát használva:

$A = B \leftrightarrow \forall x\big((x \in A \rightarrow x \in B) \wedge (x \in B \rightarrow x \in A)\big)$, vagy másképpen

$A = B \leftrightarrow \forall x(x \in A \leftrightarrow x \in B)$

A halmaz részhalmaza a halmaz egyes elemeinek a halmaza. A halmaz ennek alapján saját részhalmaza is egyben. A részhalmaz minden eleme ugyanakkor eleme az eredeti halmaznak is, továbbá az eredeti halmaznak lehetnek olyan elemei, amelyek nem elemei a részhalmaznak. A részhalmaz fogalmának formalizálása pontosan ezen alapul.



atikusés filozófus

$A \subseteq B \leftrightarrow \forall x(x \in A \rightarrow x \in B)$

Az ***A*** halmazt a ***H*** halmaz valódi részhalmazának nevezzük, ha az ***A*** halmaz részhalmaza a ***H*** halmaznak, de nem egyenlő vele. Jelölése: $A \subset H$.

A részhalmaz fogalmából következik az is, hogy ha $A \subseteq B$ és $B \subseteq A$, akkor $A = B$.

Egy halmaz összes részhalmazainak halmazát partikuláris halmaznak nevezzük. Bizonyítás nélkül fogadjuk el, hogy 4 elemű halmaznak $2^4 = 16$, 5 elemű halmaznak $2^5 = 32$, ..., n elemű halmaznak 2^n darab részhalmaza van. (Milyen magyarázat adható erre az összefüggésre?)

A halmazelméleti egyenlőségek bizonyítása a logikai ekvivalens formulák segítségével egyszerű. A legalapvetőbb egyenlőségek a következőképpen feletethetők meg a logikai ekvivalenciáknak:

Tétel. $(p \vee q) \leftrightarrow (q \vee p)$ azaz $A \cup B = B \cup A$

Bizonyítás: Ha $(x \in (A \cup B)) \leftrightarrow (x \in A \vee x \in B)$. Legyen a p kijelentés az $x \in A$, a q kijelentés az $x \in B$. Azaz a $(x \in A \vee x \in B)$ kijelentés formalizálása $(p \vee q)$. Tekintettel arra, hogy a $(p \vee q) \leftrightarrow (q \vee p)$ ekvivalencia igaz, a következőképpen írhatjuk le a bizonyítást:

$(x \in (A \cup B)) \leftrightarrow (x \in A \vee x \in B) \leftrightarrow ((p \vee q) \leftrightarrow (q \vee p)) \leftrightarrow (x \in B \vee x \in A) \leftrightarrow (x \in (B \cup A))$ ◊

Hasonlóképpen belátható a következők kapcsolata:

$(p \wedge q) \leftrightarrow (q \wedge p)$ és $A \cap B = B \cap A$

$((p \vee q) \vee r) \leftrightarrow (p \vee (q \vee r))$ és $(A \cup B) \cup C = A \cup (B \cup C)$

$((p \wedge q) \wedge r) \leftrightarrow (p \wedge (q \wedge r))$ és $(A \cap B) \cap C = A \cap (B \cap C)$

Igaz állításnak (*i*) tekinthetjük a következőt: $x \in U$, hamis állításnak (*h*) pedig ezt: $x \in \varnothing$. A negáció ebben az egyszerű halmazelméleti megközelítésben a komplementer halmazt jelöli, azaz ha $\neg(x \in A)$, de mindenképpen $x \in U$, akkor

$(\neg(\neg p)) \leftrightarrow p$ azt jelenti $(A^C)^C = A$

$(p \wedge i) \leftrightarrow p$ azt jelenti $A \cap U = A$

$(p \wedge h) \leftrightarrow h$ azt jelenti $A \cap \varnothing = \varnothing$

$(p \vee i) \leftrightarrow i$ azt jelenti $A \cup U = U$

$(p \vee h) \leftrightarrow p$ azt jelenti $A \cup \varnothing = \varnothing$

$(p \wedge \neg p) \leftrightarrow h$ azt jelenti $A \cap A^C = \varnothing$

$(p \vee \neg p) \leftrightarrow i$ azt jelenti $A \cup A^C = U$

De Morgan[22] azonosságok

$\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$ azaz $(A \cup B)^C = A^C \cap B^C$

$\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q)$ azaz $(A \cap B)^C = A^C \cup B^C$

A disztributív szabályok

$(p \wedge (q \vee r)) \leftrightarrow ((p \wedge q) \vee (p \wedge r))$ azaz $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

$(p \vee (q \wedge r)) \leftrightarrow ((p \vee q) \wedge (p \vee r))$ azaz $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

1.7.1 A halmazelmélet, a logika és a Boole algebra

Az olyan struktúrákat, amelyekhez egy nem üres (legalább kételemű) halmaz, és a halmazon definiált két binér művelet tartozik **BOOLE[23] ALGEBRÁnak nevezzük.** Így a $A = \{0,1\}$ két igazságérték halmaza az $\wedge$ és a $\vee$ művelettel, (A, $\wedge,\vee$) Boole algebrát alkot. A műveletek és kitüntetett elemek kapcsolatrendszerét a műveletek kommutatív, asszociatív és disztributív tulajdonságai jellemzik, és érvényesül a komplementer-képzés szabálya, azaz például $i \wedge h = h, i \vee h = i$...

Boole algebrát alkot továbbá az eseményalgebra, ahol a két kitüntetett elem (itt halmaz) a biztos esemény (lehet a jelölése **1**) és a lehetetlen esemény (jelölheti **0**), a műveletek az események metszete és uniója.

A halmazok halmaza, kitüntetett elemként az univerzummal és az üres halmazzal, továbbá ugyancsak a metszet és unió műveletével ugyancsak Boole algebrai tulajdonságokkal bír.

Boole elméleti alapozása jelentősen hozzájárult a számítástechnikában alkalmazott logikai kapuk megépítéséhez.

---

[22] **Augustus de Morgan** (1806-1871), angol matematikus.

[23] Goerg Boole (1815-1864), angol matematikus

1.8 Elsőrendű logika[24]

Figyeljük meg a következő állítást: $x<0$.

Megállapítható-e az igazságértéke? Igen, de két komoly megválaszolandó kérdés merül fel:

- Melyik számhalmazba tartozik az x számváltozó?
- Vajon ha meghatározzuk a halmazt, akkor annak minden x számváltozójára igaz az állítás, vagy csak létezik olyan eleme a halmaznak, amelyre igaz az állítás?

Legyen például x egy természetes szám. Az $x<0$ állítás ilyenkor hamis. (Nincs olyan természetes szám, amely kisebb lenne, mint nulla.)[25]

Legyen x egész szám, az állítás ilyenkor igaz, de *nem minden* egész számra, azaz *létezik olyan* egész szám, amelyre igaz.

Legyen x negatív egész szám, az állítás ilyenkor igaz, de *minden* egész számra.

Két fontos következménye van az előbbieknek:

Ha egy formulában olyan kifejezés (függvény, reláció) szerepel, amelyben egy halmaz elemére (elemeire) vonatkozó változó szerepel, akkor a kifejezéshez tartozó kijelentést csak úgy tudjuk kiértékelni, ha:

- tudjuk, mely halmazból való az elemet helyettesítő változó;
- tudjuk, hogy a halmaz **bármely elemére**, vagy **csak egyes elemekre**, (vagy egyikre sem) igaz a kijelentés. Azaz: helyettesítve az összes elemet a halmazból, vagy csak helyettesítve néhány elemet lesz igaz az állítás.

Azt a halmazt, amelyből az elemeket helyettesítő változó értékeit veheti *univerzumnak* (U) vagy háttérhalmaznak nevezzük.

Ha azt szeretnénk kifejezni, hogy az univerzum minden elemére igaz (vagy hamis) egy állítás, akkor ezt így jelöljük:

$(\forall x \in U)$ állítás (amely igazságértéke x helyettesítő értékétől függ)

---

[24] ezen fejezet példái, meghatározásai elsősorban két forrásra hagyatkoznak, ezek az Ésik(2011), illetve Bércesné (2012)

[25] A számhalmazokat és annak elemit a következő, a halmazokról szóló fejezetben definiáljuk. Kérjük ott tekintse meg a leírásokat.

Ha azt szeretnénk kifejezni, hogy az univerzumban létezik olyan elem, amelyre igaz (vagy hamis) egy állítás, akkor ezt így jelöljük:
$(\exists x \in U)$ állítás (amely igazságértéke x helyettesítő értékétől függ).

A fenti példákban:
$(\exists x \in Z)(x < 0)$ (így olvassuk: létezik olyan x egész szám, amely kisebb, mint nulla). Az állítás így **igaz**.
$(\forall x \in Z)(x < 0)$ (így olvassuk: minden x egész szám kisebb, mint nulla). Az állítás hamis, mert ha x helyére pozitív számot vagy nullát helyettesítünk, akkor ez nem igaz.
$(\forall x \in Z^-)(x < 0)$ (így olvassuk: minden számra a negatív egész számok halmazából (Z^-), a szám kisebb, mint nulla). Az állítás ezen az univerzumon igaz.

Az elsőrendű logika tehát a predikátumlogika kiterjesztése mind szintaktikai, mind szemantikai szempontból.
A predikátumlogikában bevezetett formulák továbbra is szabályosak, de megjelenik két új *kvantor*, a $\forall$ és a $\exists$. A kvantorokat úgy írjuk be a formulába, hogy egyértelmű legyen a hatásuk a formula többi részében szereplő változókra. Ha kell, hangsúlyozzuk ki a kvantorok hatáskörét zárójelezéssel. A különböző kvantorok sorrendje egy formulában meghatározó, mert a formula jelentését képes megváltoztatni egy kvantor-sorrend csere.
A kvantorokkal bővített elsőrendű logika szemantikája nem értelmezhető a háttérhalmaz, az univerzum nélkül, mert csak ennek ismeretében vagyunk képesek kiértékelni a formulát.
Ha a $(\forall x \in Z^-)(x < 0)$ formulát még általánosabb alakban írjuk, azaz a kisebb relációt $R(x,0)$ módon jelöljük, akkor a $(\forall x \in Z^-)R(x,0)$ kijelentés az x halmazától, az R reláció jelentésétől (interpretációjától) függ, és az összes lehetséges behelyettesítéssel nyert reláció igazságértékétől, azaz meg kell vizsgálnunk az egyenlőtlenséget minden olyan x-re, amely a $Z^- = \{-1, -2, \ldots\}$ halmazból való.

Az elsőrendű logikában is működnek a helyes következetési szabályok, de csak úgy vizsgálódhatunk, ha kiértékeléskor "végigfutunk" a változóértékekkel a teljes Univerzumon.

1.19. példa. Figyeljük meg a következő mondatokat:

Minden édesanya szereti a gyermekét. Mária édesanya. János Mária gyermeke. Ebből következik, hogy Mária szereti Jánost.

Az első dolog, hogy elkülönítsük a feltételeket és a következményt. Majd a függvényeket, relációkat, így:

Minden édesanya szereti a gyermekét.	**Feltételek**
János édesanyja Mária.	
Mária szereti Jánost	**Következmény**

Formalizálva: $(\forall x \in E)(\forall y \in E)\left(Anyja(x,y) \rightarrow Szereti(x,y)\right)$, ahol E az emberek halmaza.

$Anyja(Maria, Janos)$ tehát $Szereti(Maria, Janos)$.

A példában most figyelmen kívül hagytunk további tényeket és kapcsolatokat, a szereplők nemére, korára és egyébre vonatkozóan, de ezekkel is bővíthető a feltételrendszer természetesen.∗

Mondjuk el a következő mondatokat:

Minden veréb madár. Minden madár gerinces. Tehát minden veréb gerinces.

Kimondtunk feltételeket: Minden veréb madár. Minden madár gerinces. És kimondtuk a következményt: minden veréb gerinces. Mégsem használhatunk nulladrendű, azaz kijelentés-logikai formalizálást, hiszen azt mondtunk például, hogy **minden** madár, tehát a sasok és más madárfajták is gerincesek. Szükség van az általános osztály (madár. gerinces) és annak elemei közötti különbség (veréb. madár) jelölésére. A prédikátumokra vonatkozó s kvantorok bevezetésével ezt megoldhatjuk. A formalizálás lehet például:

Ha $\forall$x (V(x)→M(x)) és

$\forall$x (M(x)→G(x))

akkor

$\forall$x (V(x)→G(x))

A V(x), M(x), G(x) predikátumok, a $\forall$ az univerzális kvantor. Az x változószimbólum, a háttérhalmaz elemeit jelöli.

A formula értelmezéséhez az azokban szereplő jelöléseket interpretálni kell. Az interpretálás azt jelenti, hogy megmondjuk, mi a jelentése a formulában használt prédikátumszimbólumoknak, a változók milyen értékeket vehetnek fel, stb., hiszen csak így tudjuk megmondani, hogy a predikátum, azaz kijelentés igaz-e. A dolgot tovább bővíthetjük halmaz elemein definiált függvények bevezetésével, amelyek csak részét képezik egy predikátumnak.

Figyeljük meg a következő formulát:

$(\exists x \in H)\left(R\left(f(x,2),0\right)\right)$

Igaz-e?

Legyen a halmaz a racionális számok Q halmaza.

Megjelent egy függvény: $f(x,2)$. Őt is interpretálni kell. Legyen ennek a jelentése: az f függvény a két argumentuma, x és 2 összege, azaz $(x,2)\xrightarrow{f}x+2$. Láthatjuk, hogy a hozzárendelt érték egy elem a háttérhalmazból.

Interpretáljuk most az R relációt. Ez már predikátum, tehát igazságértéket kell hozzárendelnünk. Jelentse R most az egyenlőséget.

Azaz a formulát ezek ismeretében átírhatjuk így is:

$(\exists x \in Q)(x+2=0)$

Így már látjuk, hogy a kijelentés igaz, hiszen ha x helyére –2-t helyettesítünk, ami benne van a Q halmazban, akkor a kijelentés igaz.

Adjunk most egy másik interpretációt a formulának.

Legyen a halmaz a természetes számok N halmaza, $N=\{1,2,3,...\}$.

Legyen az f függvény jelentése: az f függvény a két argumentuma, x és 2 szorzata, azaz $(x,2)\xrightarrow{f}x\cdot 2$. Láthatjuk, hogy a hozzárendelt érték egy elem a háttérhalmazból.

Interpretáljuk most az R relációt. Jelentse R most is az egyenlőséget.

Azaz a formulát ezek ismeretében átírhatjuk így is:

$(\exists x \in N)(x \cdot 2 = 0)$

Így már látjuk, hogy a kijelentés hamis, hiszen x helyére nem találunk olyan természetes számot, amely 2-vel nullát adna szorzatul.

Mindebből az a tanulság, hogy az ilyen predikátum kifejezések kiértékelése a halmazbeli változók háttérhalmazának (univerzumának), a függvények és relációk interpretációja és a helyettesítési értékekkel elvégzett számítások nélkül elképzelhetetlen. Konkrét igazságértéke csak akkor lesz tehát a formuláknak, ha vagy minden szereplő változója *kvantált*, azaz a kvantorokkal kötött vagy minden változó helyén konstans szerepel. Mindez értelemszerűen a formulák szemantikai elemzéséhez kapcsolódik.

Ami a szintaxist illeti, már láttuk hogyan használjuk a $\forall$ és $\exists$ kvantorokat a mondatok modellezésekor. Ha a kvantorokkal kötött kijelentéseket figyeljük, akkor azok nyilván lehetnek ugyanúgy összetettek, mint a nulladrendű logikában, azaz az *és* illetve a *vagy*, továbbá a *nem* és a *ha...akkor* tartalmú mondatok, amelyeket rendre az $\wedge, \vee, \neg, \rightarrow$ jelekkel modellezünk.

1.8.1 Az elsőrendű logika szintakszisa, nyelve

1.8.1.1 A jelkészlet elemei

1.Elsőrendű változók, vagy individuum változók (az univerzum elemeit helyettesíthetik): $x, y, \ldots, x_1, y_1, \ldots$

2. Függvény jelek, vagy függvény szimbólumok: $f, g, \ldots, f_1, g_1, \ldots$

3. Predikátum jelek, predikátum szimbólumok, vagy reláció szimbólumok:
$p, q, r, \ldots, p_1, q_1, r_1, \ldots$ (ha összetettek akkor általában $P, Q, \ldots$)

4. Logikai műveleti jelek: $\wedge, \vee, \rightarrow, \leftrightarrow, \neg$, *i* (igaz), *h* (hamis),

5. kvantorjelek: $\exists, \forall$

6. Elválasztó jelek:) (és a ,

A változók halmaza megszámlálhatóan végtelen, a függvény és predikátum szimbólumok halmaza véges vagy megszámlálhatóan végtelen.

Minden függvényszimbólumra és predikátumszimbólumra adott a szimbólum rangja, vagy aritása, amely nem-negatív egész szám.

A 0 aritású függvényjel: konstans, vagy konstans szimbólum.

Mielőtt tovább építenénk az elsőrendű konstrukciókat, ki kell emelnünk, hogy az egyenlőségjeles elsőrendű logikában az = reláció szimbólum kitüntetett, hozzárendelést is jelent.

Az alapjelekből (a logika nyelvének alapszimbólumaiból) termeket képezünk a következő szabályok szerint:

A termek halmaza a legszűkebb olyan halmaz, melyre teljesül:

1. Minden változó term.

2. Ha $t_1, \ldots, t_n$ termek, *f* egy *n*-ed rangú függvényjel, akkor $f(t_1, \ldots, t_n)$ is term. ($n = 0$ esetén az *f* konstans jel.)

Például: $f(g(x, h(y)), c)$, ahol f és g 2-rangú függvényjelek, h 1-rangú függvényjel, c konstans szimbólum, x, y változók.

Egy term alapterm, ha nem fordul elő benne változó.

Minden term egyértelműen olvasható, azaz vagy változó, vagy egyértelműen írható le: $f(t_1,\ldots, t_n)$ alakban, ahol f függvényjel, $t_1,\ldots, t_n$ termek.

Formulaképzés

Atomi formula egy $p(t_1,\ldots, t_n)$ alakú kifejezés, ahol p n-rangú predikátum szimbólum, $t_1,\ldots, t_n$ pedig termek. (ha $n = 0$, akkor ez maga a p jel)

A formulák halmaza a legszűkebb olyan halmaz, amelyre teljesül:

1. Minden atomi formula egyben formula is.
2. i és h formulák. Ha F és G formulák, akkor $(F \wedge G)$, $(F \vee G)$, $(F \rightarrow G)$, $(F \leftrightarrow G)$ és $(\neg F)$ is formulák.
3. Ha F formula, x változó, akkor $(\exists x F)$ es $(\forall x F)$ formulák.

Megjegyzés: A szokásos precedencia szabályokkal élve gyakran elhagyjuk a külső, és a feleslegessé váló zárójeleket. (Az $F \rightarrow (G \rightarrow H)$ formula ugyanaz, mint a $F \rightarrow G \rightarrow H$ formula)

1.20. példa. $p(x, f(y)) \rightarrow (\exists z(\neg\, q(x,z))$, ahol x,y,z változók, f 1-rangú függvény jel, p,q 2-rangú (aritású) predikátum jelek.$*$

Minden formula egyértelműen olvasható, de fontos itt is elkülöníteni a részformulákat és a kvantorok hatáskörét.

Legyenek F es G formulák.

F a G közvetlen részformulája, ha a G formula $(\neg F)$, $(F*H)$, $(H*F)$ vagy (QxF) alakú, ahol

$* \in\{\wedge,\vee,\rightarrow,\leftrightarrow\}$ és $Q \in\{\exists,\forall\}$, H formula, x változó.

F a G részformulája, ha létezik a formulák olyan $H_0, \dots, H_n$ (n > 0) sorozata, hogy $H_0 = G$, $H_n = F$, és H_i a H_{i-1} közvetlen részformulája, $i = 1,\dots,n$ esetén.

Egy F formula pontosan akkor a G formula részformulája, ha G felírható

G = uFv

alakban alkalmas u és v szavakra, azaz ha F rész-szóként előfordul G-ben.

Legyen F formula, G az F egy Q*x*H alakú (Q kvantor) részformulája. Ekkor az *x* H-ban való előfordulásai kötöttek. Az *x* egy F-ben való előfordulása szabad, ha nem kötött.

1.21. példa. Az alábbi formulában a piros változó előfordulások kötöttek, a zöldek szabadok.

$\exists x(p(x, y) \wedge \forall z(q(x,y) \vee r(x,z))) \vee q(x,x)$ *

Zárt formulának vagy mondatnak nevezünk egy olyan formulát, melyben egyetlen változó sem fordul elő szabadon.

1.22. példa. A $\exists x \forall y\, p(f(x,y))$ formulában minden változó kötött, tehát mondat.

1.8.2 Az elsőrendű logika szemantikája

Legyen L elsőrendű nyelv (mely a függvény és predikátum szimbólumokkal adott).

L-típusú struktúra egy olyan B = (A, I, φ) hármas, ahol:

1. A egy nemüres halmaz (univerzum),
2. I (interpretáció) minden f n-rangú függvényszimbólumhoz egy $I(f) : A^n \to A$ függvényt, és minden n-rangú **p** predikátum szimbólumhoz egy $I(\mathbf{p}) : A^n \to \{0,1\}$ *igaz* vagy *hamis* igazságértéket felvevő predikátumot (vagy reláciot) rendel,
3. φ minden változóhoz az A egy $\varphi(x)$ elemét rendeli („helyettesítő érték”).

Megjegyzés: Az n=0 esetben I(f)-et az A, I(p)-t a {0,1} halmaz egy elemével azonosíthatjuk.

Megjegyzés: Néha a struktúra harmadik komponensét elhagyjuk, ekkor struktúra egy (A, I) pár.

Megjegyzés: amennyiben a nyelv egyenlőséges, kikötjük, hogy I(=) az A halmazon értelmezett egyenlőségi predikátum.

Legyen t term, **A** = (A, I, φ) struktúra. Ekkor a t által az **A** struktúrában jelölt $A(t) \in A$ elemet az alábbi módon definiáljuk:

1. Ha $t = x$, akkor $A(t) = \varphi(x)$.
2. Ha $t = \mathrm{f}(t_1, \ldots , t_n)$, akkor $A(t) = I(\mathrm{f})(A(t_1), \ldots , A(t_1))$.

Megjegyzés: I(*f*) helyett gyakran *f*-et, *I*(*p*) helyett *p*-t írunk.

1.23. példa.[26] (Egy adott interpretációhoz) Legyenek + és × 2-rangú függvényjelek; ´ 1-rangú függvényjel; $\underline{0}$, $\underline{1}$ konstansjelek, < 2-rangú predikátum szimbólum.

N = (N_0, I, φ) ahol

1. N_0 = {0, 1, 2, ... },
2. I(´) a rákövetkezési függvény, I(+) es I(×) az összeadás és szorzás függvények, $I(\underline{0})$ = 0, $I(\underline{1})$ = 1 az N_0 halmazból I(<) a szokásos rendezés.
3. φ(x) = 2, φ(y) = 3, ... (mintha helyettesítési érték lenne)

Ekkor:

1. $t = (x + \underline{1}) \times y$, N (t) = 9,
2. t = (x × y) + $\underline{0}$, N (t) = 6,
3. t = ($\underline{0}$ + $\underline{1}$) × $\underline{1}$, N (t) = 1. ∗

Legyen **A** = (A, I, φ) struktúra, x változó, $a \in A$. Ekkor az $\mathbf{A}[x \mapsto a]$ az az (A, I, φ´) struktúra, ahol

$$\varphi'(y) = \begin{cases} \varphi'(y) & ha \quad x \neq y \\ a & ha \quad x = y \end{cases}$$

[26] A példák forrásai Fülöp Zoltán, Ésik Zoltán és Németh L. Zoltán elérhető kapcsolódó tananyagai

Megjegyzés: a helyettesítést gyakran egyszerűbben x/a módon jelöljük.

Legyen F formula, **A** = (A, I, φ) struktúra. Az F értéke az **A** struktúrában az alábbi A(F)∈ {0,1} érték:

Ha F a $p(t_1, \ldots, t_n)$ atomi formula, akkor A(F)=$I(p(A(t_1), \ldots, A(t_n))$.

Ha F =i vagy F =h↓, akkor sorrendben A(F) = 0 illetve A(F) = 1.

Ha F = G*H, ahol *∈{∧,∨,→,↔}, akkor A(F) = A(G)*A(H).

Ha F =¬G, akkor A(F)= ¬ A(G).

(vagyis a helyettesítések alapján számítjuk a logikai értéket)

Ha F = ∃xG, akkor

$$A(F)=\begin{cases}1 & \text{ha van olyan } a\in A \text{ hogy} \quad A_{[x\mapsto a]}(G)=1\\ 0 & \qquad\qquad\qquad\qquad \textit{különben}\end{cases}$$

Azaz ha van olyan $x \mapsto a$ (x/a) helyettesítés, amelyre igaz az állítás. *(a ∈A)*

Ha F = ∀xG, akkor

$$A(F)=\begin{cases}1 & \text{ha bármely } a\in A \text{ elemre} \quad A_{[x\mapsto a]}(G)=1\\ 0 & \qquad\qquad\qquad\qquad \textit{különben}\end{cases}$$

Azaz ha bármely $x \mapsto a$ helyettesítésre igaz az állítás. *(a ∈A)*

Megjegyzés: Amennyiben A(F) = 1, akkor az

A|= F vagy **A**∈Mod(F)

jelölést is használjuk, és azt mondjuk, A kielégíti az F formulát, vagy modellje az F formulának. **A** a formulában szereplő függvények és predikátumok A halmazbeli interpretációjára, illetve a halmaz elemeivel történő számolásra vonatkozó struktúra.

Ellenkező esetben az A |≠ F jelölést is használhatjuk.

1.24. példa: Legyen **N**=(N, I, φ) a korábban megadott példából.

- Ha φ(x) = 0, akkor **N**|≠ ∃y(y < x).
- Ha φ(x) = 3, akkor N |= ∃y(y < x).

Tetszőleges φ esetén **N** modellje az alábbi formulák mindegyikének:

1. $x < x + 1$, $\forall x(x < x + 1)$.

2. $\forall x(x \times x = x \rightarrow (x = 0 \vee x = 1))$.

3. $\forall x \forall y(x = y \;\vee\; x < y \;\vee\; y < x)$.

Legyen t term, F formula, és legyenek **A**= (A, I, φ) és **A**'= (A, I, φ') struktúrák.

1. Ha φ(x) = φ'(x) minden olyan x változóra, amely előfordul t-ben, akkor A(t)= A'(t).

2. Ha φ(x) = φ'(x) minden olyan x változóra, mely szabadon előfordul F-ben, akkor A (t)= A'(t).

Következmény: A fenti jelölésekkel, A(t) független minden olyan változó értékétől, mely nem fordul elő t-ben. Továbbá A(t) független minden olyan változó értékétől, mely nem fordul elő szabadon F-ben.

Ezért ha F mondat, akkor értelmes arról beszélni, hogy A $\models$ F teljesül-e egy **A**(A, I) változó hozzárendelés nélküli struktúrára.

Az elsőrendű logikában is érvényes:

1. Egy F formulát kielégíthetőnek nevezünk, ha létezik modellje. Ellenkező esetben F kielégíthetetlen, vagy azonosan hamis.

2. Egy F formulát tautológiának (vagy érvényesnek, vagy azonosan igaznak) nevezünk, ha minden struktúra kielégíti. Jelölése: $\models$ F.

1.25. példa. Tautológiák: i, F$\vee$ ¬F, F → F, F →G → F, ahol F,G tetszőlegesek.

Kielégíthető formulák, melyek nem tautológiák: (p ∧q) → ¬p, $\exists x p(x)$.

Azonosan hamis: h, F∧ ¬F, ahol F tetszőleges.∗

F akkor és csak akkor kielégíthető, ha ¬F nem tautológia. F akkor es csak akkor tautológia, ha ¬F azonosan hamis.

Tetszőleges F formulára és x változóra:

$\models$F akkor és csak akkor, ha $\models$ $\forall xF$.

(jelentése: egy formula akkor és csakis akkor azonosan igaz, ha univerzális lezártja igaz)

F akkor és csak akkor kielégíthető, ha $\exists x F$ az.

(jelentése: egy formula akkor és csak akkor kielégíthető, ha egzisztenciális lezártja az.)

Azt mondjuk, hogy az F és G formulák ekvivalensek, ha Mod(F) = Mod(G). Jelölése: F≡G.

(azaz ha megfeleltető modelljeik ekvivalensek. Mi is a modell? Tekinthető egyfajta interpretációnak is.)

Állítás: F≡G akkor és csakis akkor, ha $\models$ (F↔G).

1.8.3 Modell, igazságérték, interpretáció az elsőrendű logikában példákon keresztül

Igaz-e, hogy ∀x ∃y P(x,y)?

Fontos, hogy

- milyen értékeket vehet fel az x és y változó?
- mi a jelentése a P prédikátumnak?

Azaz milyen modellben, struktúrában vizsgálódunk?

I. interpretáció :

Legyen az univerzum a kibővített természetes számok halmaza N_0, és $P(x,y)$ jelentse azt, hogy $x<y$. Ekkor a $\forall x \exists y\, P(x,y)$ formula jelentése, interpretációja: $(\forall x \in N_0)(\exists y \in N_0)x<y$ Vagyis, minden x természetes számhoz létezik egy nálánál nagyobb y természetes szám.

Ebben az interpretációban tehát a formula igaz.

II. interpretáció: az univerzum a kibővített természetes számok halmaza, azaz $U := N_0$, $P(x,y) := y<x$, vagyis, igaz-e, hogy $(\forall x \in N_0)(\exists y \in N_0)y<x$.

Ez nem igaz, mert az x=0-hoz nincs ilyen y. Hamis.

III. interpretáció: univerzum:= racionális számok $x, y \in Q$, a $P(x,y)$ prédikátum jelentése: $x \cdot y = 1$ (azaz $P(x,y)$ igaz, ha $x \cdot y = 1$).

A formula kiértékelése hamis, mert $0 \in Q$ elemre nincs ilyen y.

IV. interpretáció: univerzum:= emberek halmaza, $P(x,y)$ predikátum jelentése: x-nek y az édesanyja. Igaz-e, hogy $\forall x \exists y\, P(x,y)$, vagyis, hogy $\forall$ (minden) embernek $\exists$ (létezik) édesanyja?

Ez igaz.

A kvantorok sorrendjéről

Struktúrának nevezzük a $(H,*)$ rendezett párt, ahol H egy nem üres halmaz, $*$ a halmaz elemein definiált binér művelet (azaz két elem rendezett párjához a halmazból egy elemet rendel, lehetőség szerint a ugyancsak a halmazból). Most tárgyaljunk csak olyan eseteket, amelyeknél a hozzárendelt elem is a halmazból való, azaz a művelet zárt. (A természetes számok halmazán a kivonás nem zárt, azaz az $(N,-)$ struktúrát nem tárgyaljuk.)

Általánosan a binér művelet tulajdonságait így írhatjuk le helyes, elsőrendű logikában megfogalmazott mondatokkal:

1. $(\forall x \in H)(\forall y \in H)(x * y \in H)$, azaz a művelet zárt a halmazon.
2. $(\forall x \in H)(\forall y \in H)(\forall z \in H)((x*y)*z = x*(y*z))$, azaz a művelet asszociatív a halmazon.
3. $(\exists e \in H)(\forall x \in H)(x*e = e*x = x)$, azaz van egy semleges eleme a halmaznak
4. $(\forall x \in H)(\exists x^{-1} \in H)(x*x^{-1} = x^{-1}*x = e)$, az elemeknek van semleges párjuk
5. $(\forall x \in H)(\forall y \in H)(x*y = y*x)$, azaz a művelet kommutatív.

Ha a struktúra rendelkezik az 1. és 2. tulajdonsággal, akkor félcsoport, ha az 1., 2., 3., és 4. tulajdonsággal akkor csoport, ha pedig minden tulajdonsággal (1-5.), akkor kommutatív csoport, vagy Abel[27] csoport.

Ha egy műveletet vizsgálunk egy számhalmazon, akkor tudnunk kell, van-e olyan elem, amely nem változtat az eredetin, azaz "semleges". Legyen a példánkban a

[27] Niels Henrik Abel (1802–1829.) norvég matematikus.

halmaz az egész számok Z halmaza, a művelet pedig az összeadás, azaz a $(Z,+)$ struktúrát vizsgáljuk. [28]

Két egész szám összege egész szám, és az összeadás a számhalmazokon asszociatív. Ha bármely elemhez hozzáadjuk a nullát, akkor az eredeti összeadandó lesz az összeg:

$$(\exists 0 \in Z)(\forall x \in Z)(x+0=x)$$

Így olvassuk: **létezik** egy olyan elem (a nulla) az egész számok halmazában, amelyet **bármely** egész számhoz (x-hez) hozzáadva azon nem változtat (x+0=x).

Az előzőek ismeretében elvárható, **bármely** egész számhoz **létezik** egy olyan párja, amellyel összeadva az összeg a nulla lesz (azaz van mivel "semlegesíteni" az elemeket):

$$(\forall x \in Z)(\exists(-x) \in Z)(x+(-x)=0)$$

A semlegesítésre azért van szükség, mert az egyenletek megoldása így lehetséges, hiszen például az x–2=3 egyenlet akkor marad egyensúlyban (azaz ekvivalens a megoldáshalmazt illetően), ha mindkét oldalához hozzáadjuk ugyanazt a számot: x–2+2=3+2. Így a bal oldal már kezelhető, hiszen –2 és +2 ellentett párok.

Figyeljük meg: amíg az $(\exists 0 \in Z)(\forall x \in Z)(x+0=x)$ formulában egy semleges elem van mindegyik elemhez, addig az $(\forall x \in Z)(\exists(-x) \in Z)(x+(-x)=0)$ formulában mindegyik elemre van egy hozzá tartozó ellentett elem.

Ha ezekben a formulákban felcserélnénk a kvantorok sorrendjét, akkor a mondatok egészen mást jelentenének.

Megjegyzés:

Ha P prédikátum, láthattuk, hogy P(x,y) jelentheti például azt, hogy $x<y$.

Önmagában a $P(x,y)$ jelsorozat csak annyit jelent, hogy az univerzum elemei között egy kétargumentumos reláció van értelmezve. A kétargumentumos predikátumokat, illetve az interpretálásukkal a relációkat sokféleképpen lehet jelölni, ha általános formulát írunk akkor általában az úgynevezett prefix jelölést használjuk, de léteznek más típusú jelölések is:

[28] A (Z,+,) struktúráról az 5, fejezetben esik szó.

az x<y, xPy jelölés az infix (közben jelölt),

az < x, y a prefix (elöl jelölt) módszer

a P(x,y) vagy Pxy általános prefix

és az (x,y)P vagy xyP a postfix (hátul jelölt) módszer.

Példa a helyettesítésekre[29].

Legyen adott a

$\exists y(P1(f1(f1(f0)),y) \vee \forall x\, (P2(f(x)) \rightarrow P3(x,y)))$

elsőrendű formula, melynek nyelve az L1, (P1, P2, P3; f0, f1) predikátumokat és függvényeket használ, melyeknek aritása rendre (2,1,2;0,1).

Az interpretációhoz tarozik, hogy az univerzum, azaz *x* és *y* háttérhalmaza a {0,1} kételemű halmaz.

P1 értelmezése: P(0,1)=*h,* P(0,0)=*i,* P(1,1)= *i,* P(1,0)=*h* (ha egyenlők az argumentumok, akkor igaz a P)

P2 értelmezése: Q(0)=*h,* Q(1)=*i*

P3 értelmezése: S(0,1)=*i,* S(0,0)=*h,* S(1,1)= *h,* S(1,0)= *h,* (ha az első argumentum kisebb mint a második, akkor igaz)

f0:=1, f(0):=1, f(1):=0.

A formula kiértékeléséhez ellenőrizzük, hogy minden változó kötött-e. Ebben az esetben igen. (Ellenőrizhető, hogy ha a teljes formulát nem kötjük meg az *y*-ra vonatkozó univerzális kvantorral, akkor a formulát ne tudjuk egyértelműen kiértékelni.)

Az P1(f1(f1(f0)),y) L1-beli formula kiértékelés az adott interpretáción történik.

Adjunk először értéket a megadott f0:=1, f(0):=1, f(1):=0 tények alapján az első részformulának, ami így P1(f(f(1)),y).

Ha y=1, akkor P1(f(f(1)),1) amiből helyettesítéssel P(1,1)=*i.*

Ha y=0, akkor P1(f(f(1)),0), P(1,0)=*h.*

A formula második részében: $\forall x\, (P2(f(x)) \rightarrow P3(x,y))$, ami a következőképpen alakul az interpretáció és helyettesítés után:

---

[29] A példa forrása: Bércesné Novák Ágnes jegyzete: http://users.itk.ppke.hu/~szako1/elsorendu_logika_20112.pdf

Ha x=1, és tudjuk, hogy P2 értelmezése: Q(1)=*i*, Q(0)=*h*, P3 értelmezése: S(1,1)= *h*, S(1,0)= *h*, akkor a P2(f(x))→ P3(x,y) formula, ami ezek szerint Q(f(1))→ S(1,y), azaz Q(0)→ S(1,y), helyettesítve *h*→*h*, ami *y* értékétől függetlenül mindenképpen igaz.

Ha x=0, és tudjuk, hogy P2 értelmezése: Q(1)=*i*, Q(0)=*h*, P3 értelmezése: S(0,1)=*i*, S(0,0)=*h*, akkor a P2(f(x))→ P3(x,y) formula, ami ezek szerint Q(f(0))→ S(0,y), azaz Q(1)→ S(0,y), helyettesítve *i*→*h*, azaz *h* ha *y*=0, és *i*→*i* azaz *i* ha *y*=1.

Összesítve:

Ha y=1 akkor (P1(f1(f1(f0)),y)∨ ∀x (P2(f(x))→ P3(x,y))) formula igazságértéke *i*∨(*i*→*i*), ami igaz.

Ha y=0 akkor (P1(f1(f1(f0)),y)∨ ∀x (P2(f(x))→ P3(x,y))) formula igazságértéke *i*∨(*i*→*h*), ami igaz.

Van tehát olyan *y* a halmazban, amelyre a P1(f1(f1(f0)),y)∨ ∀x (P2(f(x))→ P3(x,y)) formula igaz. Láthatjuk, akár azt is mondhattuk volna, hogy minden *y*-ra igaz a formula. *

Ami azonban ezekből a példákból is kiderül, hogy az ilyen elemzés bonyolult, és korán sem mondható automatikusnak. Felmerül tehát a kérdés, hogy vannak-e az elsőrendű logikában olyan azonosan igaz formulák (amelyek univerzum-függetlenek), és egyszerűsíthetik a folyamatot? Vannak-e helyes következtetések? Vannak-e normálformák? Mi történik a helyességgel és teljességgel? Hogyan képzelhető el a rezolúció?

1.8.4 Fontosabb elsőrendű ekvivalens formulák

Olvassuk el a formulákat, és bizonyítás nélkül, de értelmüket vizsgálva fogadjuk el a következőket:

$\neg\forall x P(x) \equiv \exists x \neg P(x)$

$\neg\exists x P(x) \equiv \forall x \neg P(x)$

$\neg\exists x \neg P(x) \equiv \forall x P(x)$

$\neg\forall x \neg P(x) \equiv \exists x P(x)$

$\forall x_i \forall x_j A(x_i,x_j,\ldots) \equiv \forall x_j \forall x_i A(x_i,x_j,\ldots)$

$\exists x_i \exists x_j A(x_i,x_j,\ldots) \equiv \exists x_j \exists x_i A(x_i,x_j,\ldots)$

$\exists x \exists y(A(x) \vee B(y)) \equiv \exists x A(x) \vee \exists x B(y)$

$\exists x(A(x) \vee B(x)) \equiv \exists x A(x) \vee \exists x B(x)$

$\forall x(A(x) \wedge B(x)) \equiv \forall x A(x) \wedge \forall x B(x)$

$\forall x \forall y(A(x) \wedge B(y)) \equiv \forall x A(x) \wedge \forall y B(y)$

A fenti formulák a kvantorkiemeléshez elengedhetetlenek, hiszen már korábban láttuk, hogy a kvantorokkal kötött változójú formulákat, *mondatokat* tudjuk egyértelműen kiértékelni.

1.8.5 Prenex formulák

Azokat a mondatokat (kvantorokkal kötött változójú formulákat), amelyekben a kvantorok kizárólag a mondat elején szerepelnek prenex formuláknak nevezzük.

A prenex formába való átírás algoritmusa nagy vonalakban a következő:

1. A logikai összekötőjelek átírása ¬, ∧, ∨-ra.

2. A DeMorgan szabályok alkalmazása addig amíg a ¬ hatásköre atomi formula nem lesz.

3. A kvantorkiemelési szabályok alkalmazása addig amíg minden kvantor a formula elejére nem kerül (ami kvantormentes formulát eredményez).

1.28. példa[30].

$\forall x\,(\forall y P(x,y) \wedge \exists y \neg\,(Q(y) \rightarrow P(x,a))) \rightarrow \neg \forall x \exists y(\neg P(x,y) \rightarrow R(x,y))$

1. lépés. Írjuk át a bejelölt implikációt ¬ és ∨ műveletre

$\forall x\,(\forall y P(x,y) \wedge \exists y \neg\,(Q(y) \rightarrow P(x,a))) \rightarrow \neg \forall x \exists y(\neg P(x,y) \rightarrow R(x,y))$

$\neg(\forall x\,(\forall y P(x,y) \wedge \exists y \neg\,(Q(y) \rightarrow P(x,a)))) \vee (\neg \forall x \exists y(\neg P(x,y) \rightarrow R(x,y)))$

Írjuk át a belső bejelölt implikációkat ¬ és ∨ műveletre

$\neg(\forall x\,(\forall y P(x,y) \wedge \exists y \neg\,(Q(y) \rightarrow P(x,a)))) \vee (\neg \forall x \exists y(\neg P(x,y) \rightarrow R(x,y)))$

$\neg(\forall x\,(\forall y P(x,y) \wedge \exists y \neg\,(\neg Q(y) \vee P(x,a)))) \vee (\neg \forall x \exists y(\neg\neg P(x,y) \vee R(x,y)))$

[30] A példa forrása: Bércesné Novák Ágnes jegyzete: http://users.itk.ppke.hu/~szako1/elsorendu_logika_20112.pdf

2. lépés. Vezessük végig a negációkat az atomi formulákig!

$\neg(\forall x\,(\forall y P(x,y) \wedge \exists y \neg\,(\neg Q(y) \vee P(x,a)))) \vee (\neg \forall x \exists y(\neg\neg P(x,y) \vee R(x,y)))$

$(\exists x \neg\,(\forall y P(x,y) \wedge \exists y\,(Q(y) \wedge \neg P(x,a)))) \vee (\exists x \neg \exists y(P(x,y) \vee R(x,y)))$

$(\exists x\,(\neg \forall y P(x,y) \vee \neg \exists y\,(Q(y) \wedge \neg P(x,a)))) \vee (\exists x \forall y \neg\,(P(x,y) \vee R(x,y)))$

$(\exists x\,(\exists y \neg P(x,y) \vee \forall y \neg\,(Q(y) \wedge \neg P(x,a)))) \vee (\exists x \forall y\,(\neg P(x,y) \wedge \neg R(x,y)))$

$(\exists x\,(\exists y \neg P(x,y) \vee \forall y\,(\neg Q(y) \vee \neg\neg P(x,a)))) \vee (\exists x \forall y\,(\neg P(x,y) \wedge \neg R(x,y)))$

$(\exists x\,(\exists y \neg P(x,y) \vee \forall y\,(\neg Q(y) \vee P(x,a)))) \vee (\exists x \forall y\,(\neg P(x,y) \wedge \neg R(x,y)))$

3. lépés. Alkalmazzuk a kvantorkiemelési szabályokat, de figyeljük meg, van-e ugyanolyan nevű változónk különböző, eltérő hatáskörű kvantorokkal kötött részformulákban! Ezek ugyanis más változóértékekre utalnak.

$(\exists x(\exists y \neg P(x,y) \vee \forall y\,(\neg Q(y) \vee P(x,a)))) \vee (\exists x \forall y\,(\neg P(x,y) \wedge \neg R(x,y)))$

Három olyan független részformulát is láthatunk, amelyekben *y* a kötött változó.

y-ra először végrehajtjuk az y/y1 helyettesítést a $\forall$y-al kezdődő második részformulában és az y/y2 helyettesítést a $\forall$y al kezdődő harmadik részformulában

$(\exists x(\exists y \neg P(x,y) \vee \forall y1\,(\neg Q(y1) \vee P(x,a)))) \vee (\exists x \forall y2\,(\neg P(x,y2) \wedge \neg R(x,y2)))$

Ha figyelembe vesszük a $\exists x(A(x) \vee B(x)) \equiv \exists x\, A(x) \vee \exists x\, B(x)$ szabályt, akkor

$(\exists x(\exists y \neg P(x,y) \vee \forall y1\,(\neg Q(y1) \vee P(x,a)))) \vee (\exists x \forall y2\,(\neg P(x,y2) \wedge \neg R(x,y2)))$

$\exists x((\exists y \neg P(x,y) \vee \forall y1\,(\neg Q(y1) \vee P(x,a)))) \vee (\forall y2\,(\neg P(x,y2) \wedge \neg R(x,y2))))$

A többi kvantor hatásköre akkor is egyértelmű, ha a formula elejére írjuk, tehát

$\exists x \exists y\, \forall y1\, \forall y2\, (\neg P(x,y) \vee \neg Q(y1) \vee P(x,a)) \vee (\neg P(x,y2) \wedge \neg R(x,y2)))$

Ez a formula már prenex alakú.

További kérdésekre kell választ kapnunk. Az, hogy $\exists x \exists y$ azt jelenti, hogy ha megtaláltuk a megfelelő x illetve y változó, helyettesíthetjük-e, van-e a többi kötött változóknak erre valamilyen hatása? Hiszen ha csak univerzális kvantorunk marad, akkor világos, hogy *minden* háttérhalmazbeli elemre helyettesítenünk kell a változókat és ellenőrizni a formula igazságértékét.

A másik kérdés, hogy ha a formula törzsét tekintjük, akkor a következtetések levonásához korábban is normálformát vártunk el. Vajon ennek mi a módja az elsőrendű formulák esetében?

A elsőrendű mondat kielégíthető, ha van olyan interpretáció, amelyikben igaz. Ezt az interpretációt a formula modelljének nevezzük. Az elsőrendű mondat érvényes, ha minden interpretációban igaz. Az elsőrendű mondat kontradikció (kielégíthetetlen), ha minden interpretációban hamis.

1.8.6 Skolem normálforma

A $\forall x_1 \forall x_2 ... \forall x_n F(x_1, x_2, ... x_n)$ alakú formula Skolem normálforma alakú, ha *F*, azaz a formula törzse kvantormentes, konjunktív normálforma alakú formula.

Skolem bizonyította, hogy tetszőleges elsőrendű formulához megszerkeszthető egy Skolem normálforma. Bizonyítható továbbá, hogy annak ellenére, hogy a skolemizáció folyamán bevezetünk új változókat, átnevezünk, kvantort emelünk ki, a kiértékelés szempontjából ez a szemantikai értelmezést nem változtatja meg.

A korábbi példában láttuk, hogyan készítjük el a formula törzsének normál alakját, és hogyan terjesztjük ki a kvantorok hatáskörét a teljes formulára, mondattá zárva azt. Az egzisztenciális kvantorok szerepe viszont külön vizsgálandó.

Az egzisztenciális kvantoros előtagokat úgy hagyhatjuk el, ha a kötött változót helyettesítjük.

Ha nincs univerzális kvantor hatáskörében az egzisztenciális kvantorral kötött változó, akkor egy új konstans-szimbólummal helyettesítünk.

Ha univerzális kvantor hatáskörében van az egzisztenciális kvantorral kötött változó (azaz azon $\forall$ általi kötések esetében, amelyek megelőzik az illető $\exists$ kötést), akkor egy új függvény-szimbólummal helyettesítünk, melynek argumentumai az univerzális kvantor(ok) által kötött változók, hiszen előfordulhat, hogy a helyettesítendő változó értékeire hatással van az univerzális kvantorral kötött változó.

1.29. példa. A $\exists x \forall y \exists z \forall s \exists t\, f(x,y,z,s,t)$

formulában a t változót két univerzális kvantor előzi meg, y és s változók tekintetében, tehát t függhet tőlük. Ezért a $\exists$ kötés elhagyása csak akkor lehetséges, hogyha biztosítjuk ennek az esetleges függőségi viszonynak a leírását, például egy $t/g(y,s)$ helyettesítéssel.

Ha az adott formula nyelvében nem foglalt az a név, akkor konstansként helyettesíthetjük vele az x változót, mert a formula teljesüléséhez az kell, hogy létezzen egy x változóérték, például a.

Minden figyelembe véve a következő kapjuk:

$$\begin{array}{cccccc} \exists x & \forall y & \exists z & \forall s & \exists t & f(x,y,z,s,t) \\ \uparrow & & \uparrow & & \uparrow & \\ x/a & & z/f(y) & & t/g(y,s) & \end{array}$$

A kapott formula tehát skolemizált:

$\forall y \forall s\, f(a,y,f(y),s,g(y,s))$.

Ha teljes kiértékelést szeretnénk elvégezni, akkor alkalmaznunk kell a skolemizáción túl az egyesítő szabályokat is, hiszen felismerhetőnek kell lenniük a formulatörzsben a megfelelő helyettesítendő változók.

Egy formulát kiigazítottnak nevezünk, ha nincs olyan változó, mely kötötten ´és szabadon is előfordul, és különböző kvantor előfordulások rendre különböző változókat kötnek le.

Jelöljünk a következőképpen: a predikátumszimbólumok: p,q,r,P,Q,R,... a változók: x,y,z,s,t,u,v,w,... a konstansok (nulla-változós függvényszimbólumok): c,d,e,..., a függvényszimbólumok pedig: f,g,h,i,j,k,l,m,n.

A predikátumszimbólumnak van aritása, ami nem más mint a változóinak a száma.

A termeket így értelmeztük:

Minden változó term.

Ha *f* függvényszimbólum, mely n változós és t1,t2,...,tn termek, akkor f(t1,t2,...,tn) is term. A második pontban n=0 esetén kapjuk, hogy minden konstans term. Mindig csak változók helyére szabad helyettesíteni, konstansok helyére nem!

Az alaptermek a változómentes termek, azaz: minden konstans alapterm.

Ha f függvényszimbólum, mely n változós és t1,t2,...,tn alaptermek, akkor f(t1,t2,...,tn) is alapterm.

Például, ha Fgv={c,f(.),g(.,.)}, akkor g(x,f(c,g(y,c))) term, f(g(c,f(c))) pedig alapterm. Az Fgv feletti alaptermek halmaza

T0={c,f(c),g(c,c),f(f(c)),g(f(c),c),g(c,f(c),g(f(c),f(c)),f(g(c,c)),...}.

Az Fgv-ról csak annyit fogunk feltenni, hogy a vizsgált formulák minden függvényszimbólumát tartalmazza, hogy T0 ne legyen üres. Ezért, ha a formulákban nem lenne konstans, akkor vegyünk be egy c konstansot a Fgv-be.

1.8.7 Herbrand kiterjesztés

Ha F=∀x1∀x2...∀xnF* egy zárt Skolem normálforma, akkor F Herbrand kiterjesztése a következő: E(F):={F*[x1/t1][x2/t2]...[xn/tn], ahol ti∈T0, i=1...n} azaz F* változóit alaptermekkel kell helyettesíteni minden lehetséges módon. Egy Σ zárt Skolem normálformákat tartalmazó formulahalmaz Herbrand kiterjesztése: E(Σ)=[F∈ΣE(F)].

Σ akkor és csak akkor elégíthető ki, ha E(Σ) kielégíthető. Végül legyen E'(Σ) az E(Σ) konjunktív normálformáinak klózhalmaza. Ennek kielégíthetősége nulladrendű rezolúcióval vizsgálható, hiszen már csak alapklózokból áll, amikben nincsenek változók.

1.30. példa.[31] Írjuk fel a következő formulák, illetve formulahalmazok Herbrand kiterjesztését, majd vizsgáljuk annak kielégíthetőségét alaprezolúcióval.

Természetesen, ha szükséges, akkor előbb végezzük el a Skolemizációt.

a)$\Sigma=\{\exists z \forall x p(x,y,z), \forall x \exists z \neg p(z,y,x)\}$

b)$F=\forall x(p(x)\rightarrow \exists y q(y))$

c)$F=\forall x \forall y(\exists z p(z) \wedge \exists u(q(x,u)\rightarrow \exists v q(y,v)))$

d)$F=\exists x(\neg \exists y p(y)\rightarrow \exists z(q(z)\rightarrow r(x)))$

e)$F=\forall x((p(f(x)) \vee q) \wedge \neg p(f(f(x))) \wedge \neg q)$

f)$F=\forall x \forall y((\neg p(x) \vee \neg p(f(a)) \vee q(y)) \wedge p(y) \wedge (\neg p(g(b,x)) \vee \neg q(b)))$

Megoldás

$\Sigma=\{\exists z \forall x p(x,y,z), \forall x \exists z \neg p(z,y,x)\}$

Lezárás (az egész halmazra együtt). Az y szabad változó új, mondjuk a konstanssal helyettesítésével: $\Sigma'=\{\exists z \forall x p(x,a,z), \forall x \exists z \neg p(z,a,x)\}$

Skolemizáció (formulánként) Az első formulában $\exists$ z miatt z helyére b, a másodikban $\exists$z helyére f(x) új Skolem függvények bevezetésével:

$\Sigma''=\{\forall x\, p(x,a,b), \forall x\, \neg p(f(x),a,x)\}$

Ez már zárt Skolem normálformák halmaza.

$\Sigma''=\{\forall x p(x,a,b), \forall x \neg p(f(x),a,x)\}$ halmazban szereplő függvényszimbólumok: Fgv={a,b,f(.)}

Szerencsére most van benne konstans, nem kell hozzávenni.

Így T0={a,b,f(a),f(b),f(f(a)),f(f(b),...}

Ezért Σ'' Herbrand kiterjesztése:

$E(\Sigma'')=\{p(a,a,b),p(b,a,b),p(f(a),a,b),p(f(b),a,b),\ldots\neg p(f(a),a,a),\neg p(f(b),a,b),\neg p(f(f(a)),a,f(a)),\ldots\}$

Most $E'(\Sigma'')$ az $E(\Sigma'')$ elemeit tartalmazó egyelemű klózok: $E'(\Sigma'')=\{\{p(a,a,b)\},\{p(b,a,b)\},\ldots$

Így $E(\Sigma'')$ és ezzel Σ kielégíthetetlensége alaprezolúcióval így bizonyítható:

1. $\{p(f(b),a,b)\}\in E'(\Sigma'')$, mert a $\{p(x,a,b)\}$ klózból kapható [x/f(b)] alaphelyettesítéssel.

[31] A példa forrása: Németh (2008)

2. $\{\neg p(f(b),a,b)\} \in E'(\Sigma'')$, mert a $\{\neg p(f(x),a,x)\}$ klózból kapható [x/b] alaphelyettesítéssel

3. Res1,2∗

1.8.8 Egyesítő helyettesítések

Az elsőrendű rezolúcióhoz mindenképpen szükség van az egységesítő helyettesítésekre. A Skólem normálformát feltételezve, prenex elhagyható, csak megjegyezzük, hogy valóban, minden változó univerzálisan kvantált. Tehát a formulatörzsre alkalmazzuk a rezolúciót.

Az egyik gond, hogy mikor rezolválható két literál, hiszen benne helyettesítendő változók is szerepelhetnek? Nulladrendben egyértelműen valamely atom és tagadása, vagyis a komplemens literálpárt tartalmazó klózok rezolválhatók.

Elsőrendben mit kell tenni, ha például Q(x,y) és Q(f(a),z) alakú literálokat kellene rezolválni? Az argumentumok nem azonosak, vajon helyes-e , ha pl. a $P(x)\vee Q(x,y)$ és az $R(w)\vee Q(f(a),z)$ klózok rezolváltjának a $P(x)\vee R(w)$ klózt tekintjük?

Végezzük el az x/f(a) helyettesítést a

K1: $P(x)\vee Q(x,y)$ és K2: $R(w)\vee Q(f(a),z)$ klózok esetében, hogy kiszámítsuk a rezolvenst, amely $P(x)\vee R(w)$ lenne.

A változók univerzálisan kvantáltak, ezért a formula minden helyettesítésre igaz kell, hogy legyen, tehát ezzel a helyettesítéssel is. Így ezt kapjuk:

K1: $P(x)\vee Q(f(a),z)$

K2: $R(w)\vee Q(f(a),z)$

Res: $P(x)\vee R(w)$

Az egységesítő helyettesítést mutató további példák előtt a formális megfogalmazások helyett néhány általános alapelvet fogalmazunk meg.

1. Változóba szabad konstanst vagy másik változót lehet helyettesíteni.
2. Változóba szabad olyan függvényt is lehet helyettesíteni, amelynek argumentumában más változó, vagy konstansok szerepelnek. (függvénybe is,

a termek képzésének szabályai szerint helyettesíthetők változók, illetve konstansok, illetve újabb függvények.)

1.31. példa. Tegyük fel, hogy a rezolválandó pár: P(a, x, h(g(z))) és P(z,h(y), h(y)). Hasonlítsuk össze a P argumentumában szereplő termeket balról jobbra, és álljunk meg azon a pozíción, ahol eltérést tapasztalunk : (a, x, h(g(z))) (z,h(y), h(y)) Az első ilyen: **a** és **z**. Így a **z** változóba az **a** konstans behelyettesíthető. A **z** változó mindegyik előfordulásába ezután be kell helyettesíteni **a**-t, ezért a kapott literálok:
P(a, x, h(g(a))) és P(a,h(y), h(y)).
Most a következő szimbólumokat, x-et és h(y)-t összehasonlítva azt találjuk, hogy x-be h(y) beírható. Ennek oka, hogy a h(y) az y univerzumbeli elemhez hozzárendelt másik univerzumbeli elem. Mivel x végigfut az összes elemen (hiszen minden változó univerzálisan kvantált), ezt a h(y) elemet is felveheti. Ezért ez x helyébe beírható. A kapott literálok: P(a, h(y), h(g(a))) és P(a,h(y), h(y)).
Az utolsó helyeken álló szimbólumok: h(g(a)) és h(y). Az előbbi gondolatmenetet most g(a)-ra és y-ra alkalmazva, érthető, hogy az univerzálisan kvantált y változóba a g(a) behelyettesíthető. Itt is minden előfordulásába az y változónak be kell írni a g(a) konstanst, így a következőt kapjuk: P(a, h(g(a)), h(g(a))) és P(a,h(g(a)), h(g(a))). Így ellentett literálpárt kaptunk, amiken a rezolúció elvégezhető.

2. Vizsgáljuk meg a következő párt: P(f(a), g(x)) és P(y,y).
Hasonlítsuk össze a P argumentumában szereplő első termeket: f(a) és y. Az elsőbe semmit sem lehet helyettesíteni, hiszen a konstans, f függvényszimbólum. De y változó, ezért az y/f(a) helyettesítés elvégezhető. Az így kapott literálok: P(f(a), g(x)), P(f(a), f(a)). Továbbhaladva a második argumentumokat hasonlítjuk: g(x) és f(a). Mivel g és f különböző szimbólumok, ugyan az x-be behelyettesíthető az f(a), de g sosem lesz ugyanolyan alakú, mint f, hiszen más a nevük. Ez a két literál tehát nem egységesíthető!

3. Tegyük fel, hogy a rezolvens pár: P(f(g(x, A)), x) és P(z, B) Balról az első argumentummal kezdve: a 2.-ban egy változó szerepel, ebbe behelyettesíthetjük az

1.-ben szereplő függvényt: z/f(g(x, A). Az első formula második argumentumában szerepel változó, ennek a helyére behelyettesíthetjük a második formulában szereplő konstanst: x/B. Ezt viszont a változó összes előfordulási helyén meg kell tenni, a többi argumentumban is: f(g(B, A), így a helyettesítés végleges alakja : x/B , z/f(g(B, A)}, a atomi formula egységesítése pedig: P(f(g(B, A), B).

1.8.9 Elsőrendű modellezés és a rezolúció alkalmazása

Ahhoz, hogy egy szabály- és tényhalmaz alapján következtetéseket vonhassunk le, a rendszert leíró mondatokat formalizáljuk, a formulákat Skolemizáljuk, és a rezolúció elvégzése érdekében elvégezzük a szükséges helyettesítéseket, egyesítéseket.

Legyen a modellezendő feladat:

1.32. példa[32]

Van olyan páciens, aki minden doktorban megbízik. A kuruzslókban egyetlen páciens sem bízik meg. Bizonyítsuk be rezolúcióval, hogy ennek az a következménye, hogy egyetlen doktor sem kuruzsló.

Megoldás:

Legyen az alaphalmaz az emberek halmaza. Jelölje P(x) azt hogy x páciens, D(x) azt hogy x doktor, K(x) azt, hogy x kuruzsló, M(x,y) azt, hogy x megbízik y-ban.

A mondatok modelljei:

Első mondat: F1: $\exists x \{P(x) \wedge \forall y\, [D(y) \rightarrow M(x,y)]\}$

(*Megjegyzés:* gyakori hiba a modellezésnél, hogy a $D(y) \rightarrow M(x,y)$ mondat helyett $D(y) \wedge M(x,y)$ mondatot írnak. Ha a mondatok értelmét újragondoljuk, beláthatjuk, hogy az implikációs változat a valóságnak megfelelő.)

F2: $\forall x \{P(x) \rightarrow \forall y\, [K(y) \rightarrow \neg M(x, y)]\}$

A harmadik mondat a feltételezett következtetés

F3: $\forall x\, [D(x) \rightarrow \neg K(x)]$

A következőkből indulunk tehát ki, azaz bizonyítandó, hogy:

$\{F1, F2\} \models_1 F3$

[32] A példák forrása: http://users.itk.ppke.hu/~szako1/elsorendu_logika_20112.pdf

A rezolúcióhoz a következő formulahalmazból kell rezolvenst képeznünk.

$F1 \wedge F2 \wedge \neg F3$

azaz

$(\exists x\{P(x) \wedge \forall y[D(y) \rightarrow M(x,y)]\}) \wedge (\forall x\{P(x) \rightarrow \forall y[K(y) \rightarrow \neg M(x, y)]\}) \wedge \neg(\forall x[D(x) \rightarrow \neg K(x)])$

A negációk elemi formulákig való visszafejtése, valamint a formulatörzs normalizálása után:

$(\exists x\{P(x) \wedge \forall y[\neg D(y) \vee M(x,y)]\}) \wedge (\forall x\{\neg P(x) \vee \forall y[\neg K(y) \vee \neg M(x,y)]\}) \wedge (\exists x \neg [\neg D(x) \vee \neg K(x)])$

$(\exists x\{P(x) \wedge \forall y[\neg D(y) \vee M(x,y)]\}) \wedge (\forall x\{\neg P(x) \vee \forall y[\neg K(y) \vee \neg M(x,y)]\}) \wedge (\exists x[D(x) \wedge K(x)])$

Figyelembe véve az azonos változónevekre vonatkozó szabályokat:

$(\exists x\{P(x) \wedge \forall y[\neg D(y) \vee M(x,y)]\}) \wedge (\forall x1\{\neg P(x1) \vee \forall y1[\neg K(y1) \vee \neg M(x1,y1)]\}) \wedge \exists x3[D(x3) \wedge K(x3)])$

A prenex formula így

$\exists x \forall y \ \forall x1 \forall y1 \exists x3(\{P(x) \wedge [\neg D(y) \vee M(x,y)]\}) \wedge (\{\neg P(x1) \vee [\neg K(y1) \vee \neg M(x1,y1)]\}) \wedge [D(x3) \wedge K(x3)])$

Egy KNF-ról van szó, így az utolsó klóz akár előbbre is helyezhető, azaz x3 nem függ szükségszerűen a megelőző univerzális kvantorokkal kötött változóktól, tehát a formula írható:

$\exists x \exists x3 \forall y \ \forall x1 \forall y1 \ (\{P(x) \wedge [\neg D(y) \vee M(x,y)]\}) \wedge (\{\neg P(x1) \vee [\neg K(y1) \vee \neg M(x1,y1)]\}) \wedge [D(x3) \wedge K(x3)])$

alakban is. Így egyszerűbb lesz a helyettesítésünk.

Az első kötött x helyettesítésével (egzisztenciálissal kötött) x/a, továbbá az x3/b helyettesítés után:

$\forall y \ \forall x1 \forall y1$

$(\{P(a) \wedge [\neg D(y) \vee M(a,y)]\}) \wedge (\{\neg P(x1) \vee [\neg K(y1) \vee \neg M(x1,y1)]\}) \wedge [D(b)) \wedge K(b)])$

Maga a formulatörzs KNF-ban van:

$P(a) \wedge [\neg D(y) \vee M(a,y)] \wedge (\neg P(x1) \vee \neg K(y1) \vee \neg M(x1,y1)) \wedge D(b) \wedge K(b)$

A klózok tehát:

K1: $P(a)$

K2: $[\neg D(y) \vee M(a,y)]$

K3: $P(x1) \vee \neg K(y1) \vee \neg M(x1,y1)$

K4: $D(b)$

K5: K(b)

K1 és K3 egyesítése alapján x1/a helyettesítést kell elvégeznünk.

K1: P(a)

K2: [¬D(y)∨M(a,y)]

K3: ¬P(a) ∨ ¬K(y1)∨¬M(a,y1)

K4: D(b)

K5: K(b)

K1 éd K3 rezolválása után belátható, hogy a

R1: ¬K(y1)∨¬M(a,y1)

A következő lépés K2 és R1 rezolválása lehet. Ehhez az egyesítést az y1/y helyettesítést végezhetjük el.

K2: [¬D(y)∨M(a,y)]

R1: ¬K(y)∨¬M(a,y)

K4: D(b)

K5: K(b))

K2 éd R1 rezolválása után belátható, hogy a kapott klózhalmaz

R2: ¬K(y)∨ ¬D(y)

K4: D(b)

K5: K(b)

K4 éd R2 rezolválásához az y/b egyestő helyettesítéssel, a rezolválás után (hiszen b konstans)

R3: ¬K(b)

K5: K(b)

K5 éd R3 rezolválása üres klóz ad.

Vagyis ha úgy tetszik, akkor egy vizsgált *a* paciens és *b* orvos/nem kuruzsló kapcsán kimondhattuk: a doktor nem kuruzsló.

1.8.10 Helyesség és teljesség még egyszer

A korábbi fejezetekben kimondtuk, hogy a nulladrendű logika helyes és teljes, azaz amit szemantikailag bizonyítunk, az levezethető és fordítva. A levezetések alapját a

Hilbert axiómák és a Modus ponens képezte. Az elsőrendű logikában a szemantikai kiértékelés nem vonatkoztatható el az interpretációtól és a változók háttérhalmazától illetve a helyettesítésektől. Ennek ellenére a megfelelő bővített axiómarendszer lehetővé teszi a számunkra, hogy az elsőrendű logika rendezett hármasának (Univerzum,Interpretáció,Helyettesítés) ismeretében itt is egyenértékű legyen a szemantikus és szintaktikus következményfogalom. A helyességet korábban bizonyították, a teljességi tételt Henkin[33] 1949-ben bizonyította be. Kimondhatjuk tehát, hogy a megadott szintaktikus és szemantikus következményfogalom az elsőrendű logikában is azonos egymással, azaz a (szemantikusan) helyes következtetések formálisan bizonyíthatók, valamint hogy minden szintaktikus következmény egyben szemantikus következmény is. A predikátumkalkulus helyessége és teljessége azt jelenti, hogy az elsőrendű logikát fel lehet építeni szintaktikai alapokon és ez a felépítés ekvivalens a szemantikai alapú felépítéssel.

A teljességet biztosító axiómarendszer megtalálható a (Pásztorné 2012) forrásban.

1.9 Logikai programozási paradigma

Az elsőrendű logika alapja a logikai programozási paradigmának. Legjellemzőbb képviselője a Prolog nyelv, neve a *programmation en logique* kifejezés rövidítése. A Prolog egy megadott logikai formuláról (célformula) képes eldönteni, hogy logikai következménye-e formulák egy adott halmazának. Ezek a formulák vagy korábban megadott illetve a program által kikövetkeztetett formulák, vagy olyan állítások (tények), amelyek a formula rezolúciós levezetését, illetve a konstansokkal történő helyettesítéseket segítik. Utóbbi formulák és a célformula a program bemenete, a kimenet pedig a válasz, hogy következik-e a célformula a többi formulából.

[33] Leon Albert Henkin (921- 2006), amerikai matematikus[

1.9.1 A Prolog mint logikai gép

A Prolog eredetileg egyszerű felépítésű és könnyen használható nyelv volt. A szintaxis és a szemantika az elsőrendű logikának szimbólum- és gondolatvilágára épül: a Prolog nyelv kifejezései általában elsőrendű logikai kifejezéseknek feleltethetőek meg. A Prolog azonban nem az elsőrendű logika számítógépes megvalósítása, annak csak bizonyos részeit képes alkalmazni.

Egyrészt nem képes az egész elsőrendű logika rendszerének és eredményeinek kezelésére, csak annak egy részét, a Horn formulákon alapuló képes visszaadni, másrészt a logikai negációt és az univerzális kvantor szemantikáját sem képes teljes mértékben hűen kezelni; nincsenek erre szolgáló beépített eszközei. Viszont bizonyos eszközei túlszárnyalják az elsőrendű logikát, mer például a rekurziót vagy további dinamikus elemeket képes kezelni.

Ezek a jellemzők szükségképp leszűkítik a Prolog alkalmazhatóságát a nem-algoritmizált, elméleti matematikai levezetőrendszerekkel szemben. A korlátok egy része szintaktikai vagy *deklaratív* jellegű; tehát a lineáris inputrezolúcióra és a Horn-klózokra vonatkozó szintaktikai megkötés okozza őket (emiatt például nincs beépítve a negáció, a logikai ekvivalencia és az univerzális kvantifikáció kezelése, bár kreatív módon alkalmazva egyes lehetőségeket, e korlátok bizonyos mértékben csökkenthetőek, tágíthatóak). A korlátok egy másik része már szemantikai jellegű, ide tartoznak a programot működtető eljárás algoritmizálásából adódó, a program beállításában megnyilvánuló ún. procedurális korlátok (például adott literál-kiválasztó stratégia alkalmazása egy másikhoz képest adott program esetén kedvezőtlen hatással lehet a végeredményre, például végtelen ciklusba eshet a program).

A Prolog deklaratív nyelv, azaz a programozónak azt kell leírnia, hogy a megadott adatokból mit kellene kiszámítani (és a Program azt jelzi vissza, lehetséges-e ez), és nem azt az algoritmust, hogy egy adathalmazból hogyan kellene kiszámítani valamit. A Prolog nyelvének jelkészletével termeket alkothatunk, mint az elsőrendű logikában. A program parancsai a *parancstesből* és a *fejből* állnak, ahol a parancstest egy KNF. A célkijelentés, vagy várható következmény pedig a fej, amely a rezolúciós szabályok miatt a parancstesthez implikációval kapcsolódik. Matematikailag, elméleti szempontból minden ilyen kifejezést (programklózt) lehetséges és célszerű Horn klózként felírni.

A programok alapját tehát a programklózok alkotják; maga a logikai (forrás)program tehát felfogható úgy, mint egy rendezett elsőrendű klózhalmaz.

A programnak megadjuk a célformulát (ez is klóz, célklóz), ezután a program ellenőrzi, hogy a célklóz a logikai program (azaz már meglevő klózhalmaz és tényhalmaz) logikai következményei közt van-e. A döntési eljárás az elsőrendű logika egyik levezető módszere, a lineáris input rezolúció. Ily módon tehát a Prolog az automatikus tételbizonyítás egy megvalósítása. Az elsőrendű logikán alapuló következtetés további alapvető feltétele, hogy a tényhalmazok alapján el lehessen végezni az illesztő helyettesítést. A nagyfokú kifejezőerő mellett azonban vannak olyan szemantikai elemek, amelyeket nehézkesen kezeli ez a programozási paradigma, mint például a negációt.

1.32. példa.

1. tények (hipotézisek), pl. : Szülő(Albert,Béla), Férfi (Albert), Férfi (Béla), Szülő(Béla, Csaba).

2. már meglevő szabályok (feltételek), pl.:

Apja(x,y):-szülő(x,y), Férfi(x)

Nagyapja (X,Z):-Apja(X,Y),szülő(Y,Z)

(így olvassuk: ha x szülője y-nak, és y szülője z-nek, akkor x nagyszülője z-nek, azaz a feltételek a szabályszerű formulában a jobb oldalról bal irányba folytatódnak a következménnyel)

1. és 2. alkotják az alap-tudásbázist.

3. A kérdés, vagy célformula, pl., ?-Nagyapja(X,Csaba).

A célformula teljesítéséhez illesztést végezhetünk el: a tényből és a feltételből.

A tényhalmazt és a meglevő alapszabályokat egyesítve:

Apja(Albert, Béla):-szülő(Albert, Béla), Férfi(Albert),

Apja(Béla, Csaba):-szülő(Béla, Csaba), Férfi(Béla),

továbbá: Nagyapja (Albert, Csaba):-Apja(Albert,Béla),szülő(Béla,Csaba)

Ez utóbbiból következtethetjük, hogy Nagyapja(X,Csaba), hiszen Nagyapja(Albert,Csaba) illesztéssel bizonyított lett.

A válasz tehát, hogy a feltételben x változó értéke Albert lesz. ∗

1.10 További logikák

1.10.1 Modális logika[34]

A nulladrendű logika vagy predikátumlogika elemei, ahogyan már kifejtettük a kiértékelhető állítások: p,q,r,..., amelyekre minden esetben azonnali egyértelmű igazságérték adható meg.

Például:

- p = „Esik az eső."
- q = „Fúj a szél."
- r = „Rossz idő van."

Ezek között a műveletek:

p és *q*: $p \land q$

p vagy *q*: $p \lor q$

nem p: $\neg p$

ha p akkor q: $p \rightarrow q$

ha *p* akkor és csak akkor *q*: $p \leftrightarrow q$.

A példánkon:

$p \land \neg q$ = „Esik az eső és nem fúj a szél."

$(p \land q) \rightarrow r$ = „Ha esik az eső és fúj a szél, akkor rossz idő van."

[34] A fejezet megírásához felhasználtam Szabó Balázs hallgató összefoglaló munkáját

A többértékű illetve folytonos logikáknál (Łukasiewicz / Fuzzy) az igaz és hamis közötti érték(ek) bevezetése volt a mérvadó.

Az elsőrendű logikában a változók és azok halmazai, függvények és változó bemenetű predikátumok, **minden** ($\forall$) és **van olyan** ($\exists$) kvantorok bővítik a kifejező erőt. Például:

A=állatok halmaza, K=kutyák halmaza, p(x) = „x állat.”, q(x) = „x kutya.”

$\forall x \in K(p(x))$ = „Minden kutya állat.”

$\exists x \in A(q(x))$ = „Van olyan állat ami kutya.”

Vegyük észre a. dualitás elvét

$\forall x \in K(p(x)) \equiv \neg \exists x \in K(\neg p(x))$

„Minden kutya állat.” ≡ „Nincs olyan kutya ami nem állat.”

$\exists x \in A(q(x)) \equiv \neg \forall x \in A(\neg q(x))$

„Minden kutya állat.” ≡ „Nem minden állat nem kutya.”

Alkalmazási területe többek között a logikai programozás.

A temporális logika a predikátumokra az időbeliség bevezetését segítette.

Például: (p = „Esik az eső.” q = „Kitisztul az ég.”)

Fp: valamikor a jövőben p – „Valamikor majd esni fog.”

Gp: ezentúl mindig p – „Ezentúl mindig esni fog.”

pUq – Addig p ameddig q be nem következik – „Addig esik az eső, amíg ki nem tisztul az ég.”

Komoly alkalmazási területe a rendszerelemzés, programtervezés.

Többféle temporális logika létezik, számunkra most fontos, hogy olyan is, ami a modális logika segítségével vezethető be

A predikátumlogikán túl tehát ezen kiterjesztések kombinációi is értelmezhetőek, főleg az elsőrendű logika mint kiterjesztés a többi számára, és ezt a modális logikában is látni fogjuk.

Modalitás fogalma

„a beszélőnek a mondat valóságtartalmához való viszonya, ill. azok a nyelvi eszközök, amelyekkel ezt kifejezi” [4]

A modális logika alapja[4]:

„A modális logikában egy kijelentés nem pusztán igaz vagy hamis lehet, hanem egy bizonyos módon igaz is – például szükségszerű, tudott vagy hitt, kötelező, bizonyítható.” [5]

A modális logika lényege a modális operátorok bevezetése a predikátumok elé

– □: „erős”modalitás,

- ◊: „gyenge” modalitás –

mint a nulladrendű logika kiterjesztései, amelyek szintaktikailag mindig ugyanúgy használandók, de jelenthetnek mást-mást is.

Első és legfontosabb, a szükségszerűséget, és lehetőséget leíró, *Alethikus modalitás:*

□p = „szükségszerű, hogy p”

◊p = „lehetséges, hogy p”

A megengedhetőség vagy deontikus modalitás megfogalmazásai:

□p = „kötelező, hogy p”

◊p = „megengedett

Az alapvető modális operátorok tehát[4]:

Erős: □p = „Szükségszerű, hogy p.”

Gyenge: ◊p = „Lehetséges, hogy p.”

Például: (p = „Esik az eső.” q = „Felhős az ég.”)

◊p = „Lehetséges, hogy esik az eső.

¬□p = „Nem szükségszerű, hogy felhős az ég.”

¬◊(p ∧ ¬q) = „Nem lehetséges, hogy esik az eső, és nem felhős az ég.”

q → ◊p = „Ha felhős az ég, lehetséges, hogy esik az eső.”

□(p → q) = „Szükségszerű, hogy ha esik az eső, akkor felhős az ég.”

Hogyan lehet még kimondani ezen operátorokat és negáltjaikat?

□p: szükségszerű, mindenképpen úgy van, biztos hogy így kell lennie.

¬□p: nem szükségszerű, nem feltétlenül úgy van, nem biztos hogy így van.

◊p: lehetséges; meg van az esélye, hogy így van; könnyen lehet, hogy így van.

¬◊p: nem lehetséges; lehetetlen; biztos, hogy nem így van.

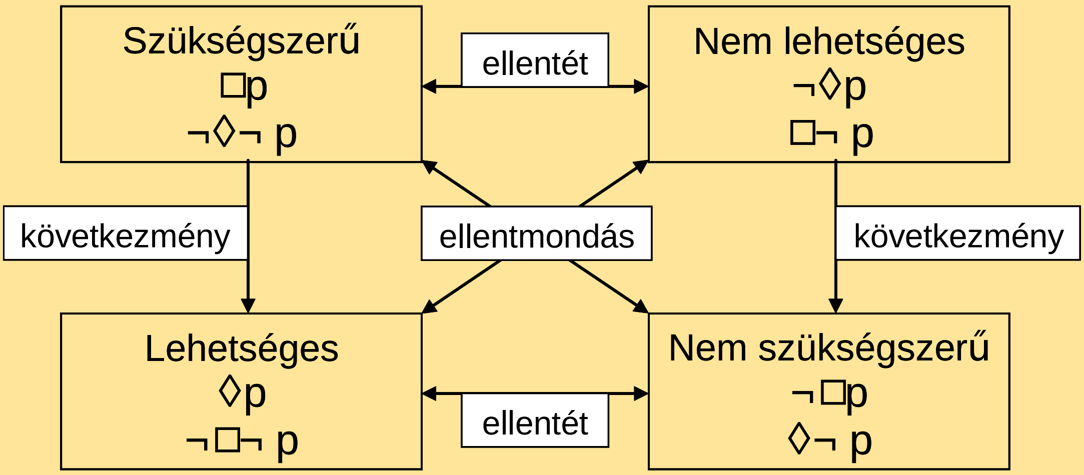
Operátorok közötti kapcsolat, azaz a dualitás így fejezhető ki:

szükségszerű, hogy p = nem lehetséges, hogy nem p, azaz:

$\Box p \equiv \neg\Diamond\neg\, p$,

lehetséges, hogy p = nem szükségszerű, hogy nem p, azaz

$\Diamond p \equiv \neg\Box\neg\, p$.



A http://upload.wikimedia.org/wikipedia/hu/d/d3/Mod%C3%A1lis_hatsz%C3%B6g.JPG alapján szerkesztve

Miben is más a nulladrendű logika „Esik az eső.” mondata, mint a modális logika „Szükségszerű, hogy esik az eső.” mondata?

Képzeljünk el egy ablakok nélküli házat. Benn ülünk, azt látjuk, hogy az ajtó alatt folyik be a víz. Tudjuk, hogy ilyet csak két dolog produkálhat, semmi más: eső folyik be vagy csőtörés van a házunk előtt. Ugyanolyan formában ugyanolyan víz folyik be. A házban ülve az a kijelentés, hogy „Esik az eső.” nem kiértékelhető, nem tudjuk eldönteni az igazságértékét, így nem használható nulladrendű predikátumként!

Amint kimegyünk a házból, már látjuk, hogy esik-e vagy sem, innentől kezdve viszont egyértelmű az igazságértéke, ellentétben azzal a helyzettel, ha a házban maradva, az „Esik az eső.” nem kiértékelhető. Ugyanakkor a

„Lehetséges, hogy esik az eső.” – kiértékelhető a modális logika megfogalmazása alapján, méghozzá IGAZ, mert tényleg lehetséges, hogy esik.

Képzeljük el ismét az előző példát! Képzeljünk egy „párhuzamos világot" ahol minden ugyanaz, ugyanazok a fizikai törvények (és a matematikai logika is), (szinte) ugyanúgy történt minden a múltban, ugyanolyanok vagyunk mi, ugyanúgy benn ülünk, és folyik be a víz, lényegében tehát teljesen olyan, mint a mi világunk, egyetlen apró dolog kivételével: a mi világunkban esik az eső, nem történt csőtörés és azért folyik be a víz. Az „övékben" nem esik az eső, hanem csőtörés történt, és azért folyik be a víz.

Mit mondhatunk el a kijelentéseinkről?

Az „Esik az eső." kijelentésünk a házban maradva mindkét világban kiértékelhetetlen.

Ha kimegyünk, az egyik világban igaz, másikban hamis lesz.

Ellenben a „Szükségszerű, hogy esik az eső." és a „Lehetséges, hogy esik az eső" kijelentések mindkét világban értelmezhetőek maradnak, ráadásul ugyanazt az igazságértéket veszik fel, a szükségszerűség hamis, a lehetőség igaz.

Folytassuk a gondolatmenetet. Nem csak két univerzumunk van, hanem sokkal több. Több olyan van, ahol csőtörés történt, több, ahol esik az eső, sőt olyanok is előfordulnak, ahol mindkettő megtörtént egyszerre.

Az esős, csőtöréses kijelentések hol igazak, hol nem, de a lehetségességek, és a szükségességek minden univerzumban ugyanolyan igazságértéket vesznek fel.

Mitől függ tehát ezeknek az igazságértéke?

A vizsgált témában vegyük számba az összes lehetséges univerzumot, azaz forduljon elő az események kimeneteleinek minden kombinációja.

Szükségszerű egy kijelentés, ha **minden** univerzumban igaz.

Lehetséges egy kijelentés, ha **van olyan** univerzum, amiben igaz.

A megfogalmazások ismerősek az elsőrendű kvantorok világából $\forall$, $\exists$.

„A $\Box$ és $\Diamond$ modális operátorokat fölfoghatjuk tehát egyfajta 'álcázott' kvantoroknak, melyek a lehetséges világok fölött kvantifikálnak." [35]

[35] kislexikon.hu – http://www.kislexikon.hu/modalitas_a.html

Modális kalkulusok[36]

Kalkulusok alatt lényegében egy olyan modális rendszert értünk, ahol kiválasztjuk a modalitás fajtáját, és megadjuk az axiómákat, valamint a levezetési szabályokat.

Sokféle kalkulus létezik:

- a Kripke után elnevezett K-kalkulus, vagy normális modális logika
- Alethikus rendszerek
- Deontikus rendszerek
- Temporális rendszerek és mások.

A modális axiómarendszer a következő:

Ismertek a Hilbert-féle klasszikus axiómák, illetve a következő axiómacsoport:

– $A \rightarrow (B \rightarrow A)$

– $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$

– $(\neg A \rightarrow \neg B) \rightarrow (B \rightarrow A)$

A kapcsolódó levezetési szabály:

$$\frac{A,\ A \rightarrow B}{B}$$

A kibővített modális axiómák így hangzanak:

$\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$

A modális levezetési szabály így hangzik:

$$\frac{A}{\Box A}$$

A *„de re”* és *„de dicto”* olvasatokat például így értelmezzük[37]:

„Minden informatikus lehet rendszergazda.”

„de re” azaz „a dologról szóló” olvasat:

„Minden informatikusra igaz, hogy megvan rá a lehetősége, hogy rendszergazda legyen.”

$\forall\ x \in I\ (\Diamond\ \text{rendszergazda}(x))$

„de dicto” azaz „az állításról szóló” olvasat:

„Lehetséges hogy az a helyzet, hogy minden informatikus rendszergazda.”

[36] Wikipédia szócikk alapján, "Modális logika"
[37] Wikipédia szócikk alapján, "Modális logika"

$\Diamond\ \forall\ x \in I$ (rendszergazda(x))

Barcan-típusú elsőrendű modális logikák

Minden lehetséges világban ugyanazok az individuumok léteznek, azaz ugyanazok felett kvantifikálunk. Alapja aBarcan formula:

$\Diamond\ \exists\ x \in A\ (\ p(x)\ ) \rightarrow \exists\ x \in A\ (\ \Diamond\ p(x)\ )$

„Ha lehetséges, hogy van olyan x, amire p(x), akkor van olyan x, amire lehetséges, hogy p(x)."

1.11 Többértékű logikák

1.11.1 Lukasiewitz féle logika

A matematikai logika célja a helyes következtetési sémák, helyes definíciók vizsgálata, beleértve a matematikai logika által alkalmazott következtetési sémákat, szabályokat, definíciókat is. Ezek a szabályok közel állnak az emberi gondolkodásmódhoz, ugyanakkor az eddig tárgyalt logikákban csakis az igaz (jelölje most 1) és hamis (jelölje most 0) igazságértékekkel működtek.

Ha azt tekintjük, hogy a matematikai logika korábban a szimbolikus logika részét képezte, abból fejlődött ki azáltal, hogy a szimbolikus logika formális módszereit kezdte alkalmazni a matematikai következtetések és bizonyítások vizsgálatára, akkor a **kizárt harmadik elvét is tárgyalnia kell.** A kizárt harmadik a logika történetében többféleképpen megfogalmazott alapelv. Általánosabb megfogalmazásban így hangzik: „Vagy *P*, vagy nem-*P*", ahol *P változót* jelöl, amelyet kijelentő mondattal lehet helyettesíteni például a nulladrendű logikában, jelölésmódja formálisan: $A \vee \neg A$.

Hogyan közelíthetjük ezt a nagyon kizáró elképzelést az emberi gondolkodás, az igazságértéket mérlegelő módszer felé? Először **Lukasiewicz**[38] vetette fel a háromértékű logika lehetőségét, ahol egy állításnak három logikai értéke lehet:

- Igaz
- Hamis
- Harmadik lehetőség (például eldönthetetlen vagy lehetséges)

Ez az elmélet a gyakorlatban is felhasználható. Kvantummechanikai alkalmazhatóságának egyik felfedezője Neumann János volt. Később sokban támaszkodott rá a fuzzy logika is.

Lukasiewicz 1920 és 1922 között dolgozta ki logikai rendszerét. Tőle függetlenül Post is ebben az időben jutott hasonló eredményekhez. Ők készítették az első publikált leírásokat a többértékű logikai rendszerről. Majdnem húsz évvel később Bochvar es Kleene, külön - külön, kidolgozták saját többértékű logikai rendszerüket egyedi műveletekkel.

Lukasiewicz többértékű rendszere az arisztotelészi logikát tagadja, elutasítja a kizárt harmadik törvényét. A harmadik érték: 'eldönthetetlen' vagy 'lehetséges', jelölése $\frac{1}{2}$ (szövegkörnyezetben időnként az ½ jelülést is hazsnáljuk).

A háromértékű logikai rendszer két olyan alapműveleten alapul, melyek bázist alkotnak (azaz a többi műveletet definiálhatjuk a segítségükkel): $\supset$ és $\neg$, amelyek az implikációt és a negációt jelölik. Ezek igazságtáblái a következők:

$\supset$	0	½	1
0	1	1	1
½	½	1	1
1	0	½	1

[38] Jan Lukasiewicz (1878 – 1956), lengyel származású matematikus

p	¬p
0	1
½	½
1	0

A 0 a *hamis*, 1 az *igaz* , az ½ tehát a *lehetséges* értéket jelenti. A negációt tükrözésként értelmezi. Mivel {⊃ , ¬} bázis, ezért kifejezhető velük a többi művelet (konnektívumoknak is nevezik őket):

- p ∨ q ⇔ (p ⊃ q) ⊃ q
- p ∧ q ⇔ ¬(¬p ∨ ¬q)
- p ≡ q ⇔ (p ⊃ q) ∧ (q ⊃ p)

Hogyan jutott Lukasiewicz a fenti igazságtáblákhoz? Nem határozott a kritikus helyeken, csak azt mondja, hogy a kívánt eredmény eléréséhez erre van szükség. Ebben az esetben a *kívánt eredmény* nem más, mint az az alapvető elvárás, hogy a (p ⊃ p) tautológia legyen. (A tautológia itt is azt a formulát jelenti, amely minden kiértékelése 1-t ad.) Az 1 értéket kitüntetett értéknek nevezzük.

Tekintsük az igazságértékeket a klasszikus igazságértékek halmazainak:

0 = {F}, 1 = {T}, ½ = {T, F}, ahol F false, hamis, T true, azaz igaz. ,

A cél az, hogy a klasszikus értékek minden egyes halmaza azoknak az értékeknek a halmazát reprezentálja, amelyeket a kijelentés a jövőben felvehet.

⊃	{F}	{T,F}	{T}
{F}	{T}	{T}	{T}
{T,F}	{T,F}	XXXX	{T}
{T}	{F}	{T,F}	{T}

p	¬p
{F}	{T}

{T,F}	{T,F}
{T}	{F}

Hogyan tölthető ki a hiányzó XXXX elem, hogy az alapvető gondolkodási sémák ne sérüljenek? A reprezentáció szerint {T, F} kellene oda. Ebben az esetben azonban nem lenne a $(p \supset p)$ formula tautológia.

Lukasiewicz ezért kritika nélkül átvesz két elvet a klasszikus logikából:

1. Principia Mathematica elve: a logikát az axiómák, a helyettesítés és a modus ponens segítségével kell felépíteni.
2. Az összetett állítások értékét részei értékének függvényeként kell megadni.

Bizonyítható, hogy Lukasiewicz háromértékű logikájában az alábbi következtetési sémák tautológiák:

1. $A \supset (B \supset A)$
2. $(A \supset B) \supset ((B \supset C) \supset (A \supset C)$
3. $((A \supset \neg A) \supset A) \supset A$
4. $(\neg A \supset \neg B) \supset (B \supset A)$

Azonban fontos megjegyezni, hogy egy nagyon fontos ekvivalencia nem áll fenn a Lukasiewitz logikában, pedig a rezolúció egyik alapvető feltétele. Ez a $(A \supset B) \equiv (A \vee \neg B)$ ekvivalencia. Ezért újabb következtetési módok kialakulása volt várható a többértékű logikák esetében, és ezek rendre meg is születtek (Fullér (1998)).

Lukasiewicz n-értékű, végtelen értékű logikája (1922)

Lukasiewicz továbblépett: az n-értékű (n véges) logikát is formalizálta. Az igazságértékek halmaza:

$$L_n = \{0, 1/(n-1), 2/(n-1), \ldots, (n-2)/(n-1), 1\}$$

és bevezethető a végtelen értékű logika is:

$$L_N = \{s/w: 0 \leq s \leq w, s, w \in N, w \neq 0\}$$

Ez a definíció valójában a n-re a [0, 1] intervallumon értelmezett racionális számok halmazát adja.

A kitüntetett érték mindkét esetben az 1 marad.

A műveletek értelmezése (amely egyébként a háromértékű logikájánál is működik):

- $|\neg p| = 1 - |p|$
- $|p \supset q| = \min(1, 1 - |p| + |q|)$
- $|p \wedge q| = \min(|p|, |q|)$
- $|p \vee q| = \max(|p|, |q|)$
- $|p \equiv q| = 1 - \text{abs}(|p| - |q|)$

Továbbá fontos elmondani, Lukasiewicz igazságszabályai esetén a {0, 1} halmaz zárt bármely konnektívumra nézve, és a bevezetett konnektívumok ezen a halmazon ugyanúgy működnek, mint a klasszikus igazságértékelések, ezért ami tautológia az *n* értékű logikában, az lesz a kétértékűben is (fordítva, mint láttuk nem mindig igaz).

1.11.2 A fuzzy logika és a lágy számítási módszerek

A lágy számítási módszerek (soft computing) gyűjtőnév, a folytonos igazságértékű fuzzy logika mellett a neurális hálózatok és a genetikus algoritmusok modelljeit is ide sorolják. Olyan matematikai eszköztárat biztosítanak számunkra, amely a hagyományos számítási módszerekkel szemben jól kezelik a modellezendő rendszerekben megjelenő bizonytalanságot, pontatlanságot, információhiányt és a közelítő következtetést. Mindent összevetve azt mondhatjuk, hogy a lágy számítási módszerek az emberi gondolkodás modelljét követve adnak eredményt, gyakran nagyon emberközeli megjelenítésben.

A lágy számítási módszerek gyakorlati alkalmazásához jelentősen hozzájárult Lofti A. Zadeh 1965-ben a fuzzy halmazokról megjelentetett cikke, majd az ezt követő publikációk, amelyek a komplex rendszerekben történő döntéshozatalról illetve a posszibilisztikus modellekről szóltak. Ezt követték a genetikus algoritmusokról és a neurális hálózatokról szóló kutatási és gyakorlati eredmények. Ezek a módszerek egyenként is, és hibrid alkalmazásokban is teret hódítottak, és együttesen alkotják a lágy számítási módszerek csoportját.

A lágy számítási módszer (angolul Soft Computing, rövidítve SC) a mesterséges intelligencia (MI) egyik olyan kutatási és megvalósítási módszere, amely közelebb áll a természetes emberi következtetésekhez, az emberi feladatmegoldásokhoz, mégis mesterséges, automatizálható számítási módszer. Jellemezően a bizonytalanság, és a döntéshez, következtetéshez szükséges információk hiánya miatt a hagyományos matematikai modellekkel nehezen kezelhető, komplex rendszerekben, a nem-lineáris problémák megoldásában hozott forradalmi változást, bonyolult rendszerek esetében is hatékonyabb, mint más hagyományos modellek, módszerek.

A lágy számítási módszerek alapja a fuzzy halmazok elmélete, a fuzzy következtetési rendszer. A fuzzy szó jelentése eredetileg bolyhos, elmosódott, nem éles határvonalú. Ha arra gondolunk, hogy valamely állítás egy adott pillanatban, helyzetben mennyire igaz, akkor beláthatjuk, mi is mérlegelünk: vagy teljesen és egyértelműen igaz vagy hamis az állítás (akkor nem fuzzy), vagy az adott körülmények között megbecsüljük mennyire igaz amit vizsgálunk, és még ha nem is számszerűsítjük az igazságértékét, a mérlegelés eredménye alapján döntünk adott helyzetben - vagyis fuzzy megközelítést használunk. Például annak az igazságértéke, hogy valaki rendelkezik tanári oklevéllel nem fuzzy, mert az állítás vagy igaz vagy hamis, de annak az állításnak az igazságértéke, hogy valaki jó tanár-e, már nem egyértelmű. Az igazságérték meghatározása szubjektív, esetleg csak más szavakkal írható le, mint jó vagy nem jó, vagy az igazságérték legfeljebb valahol "félúton", vagy "útközben" van az igaz és hamis között.

Meg kell tanulni "fuzzy-ul gondolkodni"? Vagy ez a természetes módja a gondolkodásunknak? Inkább ez utóbbi, hiszen a megismerés és elfogadás után a fuzzy és a soft computing elméletek széles körben elterjedtek. Jól mutatja ezt az is, hogy az első fuzzy cikk megjelenése óta (Zadeh, 1965.) az elméletből mindennapos gyakorlat lett. A gyakorlati alkalmazások általában jóval megelőzik azok matematikai hátterének részletes kidolgozását és beillesztését az egyetemes matematikai axiómarendszerbe.

A fuzzy atyja, Lofti A. Zadeh[39] nevezte először soft computing gyűjtőnéven az 1991-ben megjelent cikkében a fuzzy-, neurális hálózat- és genetikus algoritmus-elméleteket, továbbá a valószínűségi következtetési rendszereket (Zadeh, 1991.). Azóta a SC módszerek továbbiakkal bővültek, amelyek részben a már említettekre épültek, részben újabb és újabb hibrid rendszerekből alakulnak ki, mint például a káoszelmélet vagy a tanulási algoritmusok jó része. Mostanság a már mindennapos műszaki alkalmazások mellé felsorakoznak azok az algoritmikus megoldások is, amelyeket az intelligens, részlegesen ellenőrzött (azaz bizonyos mértékig önállóan működő) rendszerekben, kockázatkezelő rendszerekben vagy a keresési (web keresési) és természetes nyelven leírt rendszerekben használunk.

Vitathatatlan azonban, hogy mindezen elméletek alapja a fuzzy elmélet, amely nélkül ezekről a szabadon szárnyaló MI modellekről nem is álmodhattunk volna. Sőt ezek nem álmok, megvalósult, szoftver- és hardver-háttérrel rendelkező valódi problémamegoldó mechanizmusok. Ezekről nagyon sok forrásban olvashatunk, de a leginkább kompetens platform erre a The Berkeley Initiative in Soft Computing (BISC) platform, amelyet Zadeh felügyel/felügyelt, és a legbonyolultabb matematikai felvetésektől a legnaivabb soft computing-gal kapcsolatos kérdésekre is választ kaphatunk, hiszen minden valamit is magára adó érintett szakember olvassa, és mint közösségi portált, tapasztalatcserére használja azt[40]. Beszéljünk tehát elsősorban és először a fuzzyról, hiszen az elmúlt évtizedek lágy számítási módszereinek alapját képezi, és ez a módszer az, amely ténylegesen idestova 50 éves. A többi lágy számítási módszer magában is megér egy-egy kiemelt fejezetet egy következő időszakban.

1.11.2.1 A kezdetek

A fuzzy a matematikában jól ismert, korábbi halmazelméleti, logikai és a műveleti elméletek általánosítása. A halmazelméleti karakterisztikus függvény fogalmát kiterjesztette a tagsági függvény fogalmára, a kétértékű logikában alkalmazott helyes következtetési szabályokat (mint például a Modus Ponens) általánosította, és a kétértékű logikai igazságérték-univerzumot (0-hamis, 1-igaz) kiterjesztette a [0,1] zárt

[39] Lofti Askar Zadeh Azerbajdzsánban született 1921-ben, matematikus, villamos-mérnök, a fuzzy gondolat megteremtője, a kaliforniai Berkeley Egyetem emeritus professzora

[40] http://www-bisc.cs.berkeley.edu/BISCProgram/

valós intervallumra, túllépve a diszkrét értékekkel manipuláló többértékű logikákat (Łukasiewicz, 1920.). Eszközként a megalkotott logikai univerzumon a logikai kijelentések igazságértékének modellezésére a Schweizer és Sclar (Schweizer, Sclar, 1960.) által korábban megírt, az alkalmazható műveleti tulajdonságokat összefoglaló, cikkét vette alapul, amelyben a szerzőpáros a t-norma és konorma családokat tárgyalta.

Az 1965-ben megjelentette cikk néhány matematikus pozitív kritikája és több szakember, az elmélet filozófiai újszerűségét bíráló, véleményén túl mindaddig nem váltott ki nagyobb érdeklődést a tudományos világban, amíg elsősorban japán szakemberek nem valósítottak meg olyan hatékony műszaki alkalmazásokat, amelyekben fuzzy alapú következtetési rendszereket vezettek be. Ezek a hagyományos szakértői és más probléma-modellező rendszereket helyettesítve nagyságrendekkel jobban, hatékonyabban működtek, hiszen kezelték a bizonytalanságot, pontatlanságot. A Távol-keleten néhány éven belül felismerték annak lehetőségét, hogy ezek a formai, elméleti megoldások jól hasznosíthatóak a nemlineáris irányítási problémák megoldásában, ahol a hagyományos differenciálegyenletekkel leírt modellek helyett fuzzy alapú közelítő következtetési rendszereket alkalmaztak. Ehhez nagy lendületet adott Mamdani (Mamdani-Assilian, 1975., illetve Mamdani, 1974.) máig gyakran alkalmazott következtetési rendszere, amely egyszerű, érthető, és nem csak a szimulációs rendszerekben, hanem a valós rendszerekben is hatékonyabb más numerikus módszereknél. A gond vele, és más jól működő fuzzy alkalmazásokkal is, az, hogy a megalapozottsága a hagyományos matematikai eszköztárral nehézkes, meg kell találni az axiomatikus felépítmény minden elemét, hogy beépülhessen a matematikai elméletek közé teljes értékűként, továbbá hogy a hagyományos logika néhány általánosan elfogadott szabályához nem igazodik teljes mértékben (az ellentmondástalanság és a harmadik kizárásának elvéhez például).

Érdekes módon alakult a fuzzy fejlődése az elmúlt évtizedekben. A Távol-Keleten egy évtized sem telt el a megjelenésétől, és alkalmazást nyert az iparban és a gazdasági alkalmazásokban, és a felhasználók az egyszerű matematikai háttérrel is beérték. Fuzzy irányítással működnek többek között az ott gyártott autók automata váltói, a háztartási gépek, a gyorsvasutak forgalomirányítása (Tsuna, Takamasa, 1988). Sokak véleménye szerint nem véletlen hogy éppen Japán járt az élen az alkalmazásban. Ez nem csak a nyolcvanas évek rohamos ipari fejlődésének köszönhető ebben a térségben, hanem a keleti filozófia nyitottságának a nem "éles" igazságértékre, azaz könnyebben elfogadják a nem egészen igaz, nem egészen hamis megfogalmazásokat is, míg az európaiak, a nyugati kultúrák sokkal inkább a határozott igaz vagy hamis mellett állnak ki. (Kóczy, Tikk 2012.)

A fuzzy megteremtőjének hazájában, az Amerikai Egyesült Államokban évtizedekig csak az űrkutatás és a hadiipar volt érdekelt az alkalmazásokban. A matematikusok világa vagy hallgatásba burkolózott, vagy bírálta az elméletet, a valószínűség-számítás elméletének egy ágaként kezelte. Pedig az nem az eseményalgebráról szól, legfeljebb fellelhetők algebrai szempontból hasonló eljárások a két terült között (Dubois, Prade, 1984). Az európai fuzzy iskolák (Linz, Gent, Toulouse, Budapest) az elméleti kutatásokkal foglalkoztak elsősorban. Zadeh, aki a mai napig a Berkeley Egyetemről figyeli alkotásának fejlődését, elfogadottságát, és számon tartja az összes sikeres alkalmazási területet, sőt kutatók százait, akik a témával foglalkoznak, már 1984-ben adott nyilatkozatában kiemelte[41], hogy a fuzzy egy új élet- és tudományfilozófia is egyben, mely nem a formalizmusokat követi. Alkalmazásorientált, és célja az, hogy az adott problémához a leghatékonyabb eszköztárat állítsa fel, nem pedig az, hogy az mindenben az elfogadott matematikai elmélet-rendszerhez igazodjon. Nyilatkozatában kiemelte, hogy a valós élet sokkal inkább igazodik a fuzzy elméleten alapuló logikához. Az emberi gondolkodáshoz sokkal közelebb áll, mint a korábbi számítógépes alkalmazásokban, szakértői rendszerekben, és általában a mesterséges intelligenciában alkalmazott kétértékű, éles határokkal rendelkező logika. Már akkor kiemelte, hogy szerinte a számítógépek szoftver és hardver architektúrájában ez előbb utóbb tükröződni fog. És igaza lett, lásd például a szemantikus web-kereső algoritmusokat vagy a felhő-hardwer architektúrákat (Sanchez, 2006.). Az akkori nyilatkozat szerint a fuzzy olyan a tudományban (a matematikában és a logikában), mint egy farmernadrágos, pólóba

[41] Communications of the ACM, April 1984, Volume 27, Number 4, REPORTS AND ARTICLES COPING WITH THE IMPRECISION OF THE REAL WORLD, An Interview with Lotfi A. Zadeh, 304-311. old

öltözött vendég egy előkelő fogadáson, ahol előírás a szmoking, a nyakkendő. Jön, megbotránkoztat, de már nem lehet kirekeszteni. Ki tudja mi történt a világgal az elmúlt közel öt évtized alatt? Talán a fogadásokon lettek engedékenyebbek, vagy a fuzzy öltözött elvárt ruhába, ha már megállíthatatlan alkalmazási lendülete tekintélyt adott neki is és az őt kutató és fejlesztő megszállottaknak.

1.11.2.2 Fuzzy halmazok, műveletek és következtetési szabályok

A fuzzy alapfogalmak bemutatásához egy példával élek, és nem a szigorú matematikai formai felvezetési követelményrendszert veszem alapul (axiómák, definíciók, tételek). Teszem ezt nem titkoltan a fuzzy népszerűsítése érdekében, és nem utolsó sorban Zadeh elképzelése alapján, miszerint "pólóban és farmerben", alkalmazásorientáltan, fuzzy szellemben dolgozzunk, ha a modellezendő feladat azt kívánja. A matematikai megalapozáshoz magyar nyelven Kóczy T. László és Tikk Domonkos könyvét (Kóczy, Tikk, 2012.), angol nyelven többek között a (Klement, mesiar, Pap, 2000) forrást ajánlom figyelmükbe.

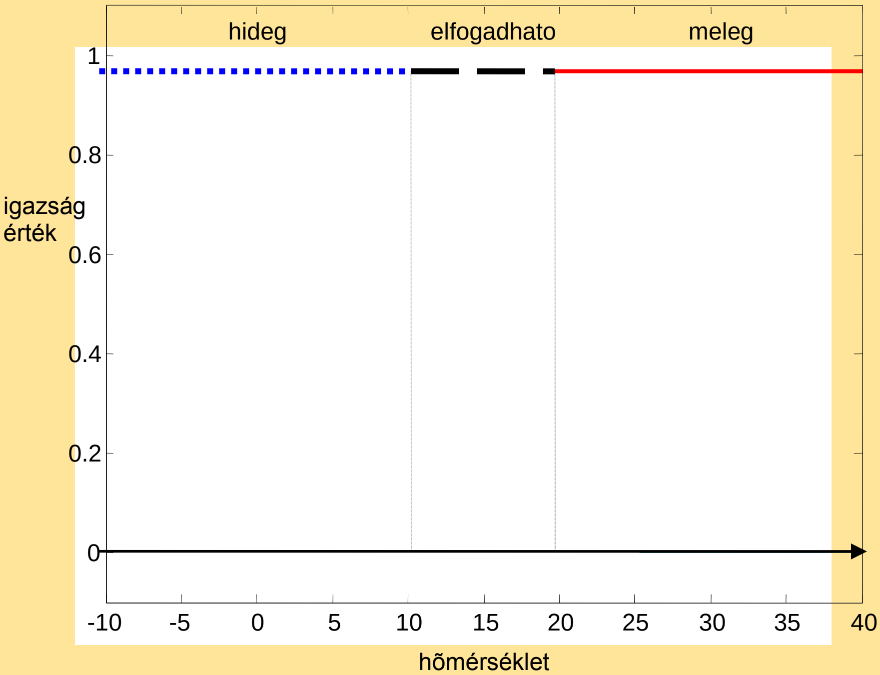
Mikor mondjuk, hogy hideg van a szobában?

A kétértékű logikában csak szigorúan éles határokkal tudunk dolgozni, ha egy állítás igazságértékét kell megállapítanunk. Számszerűsítve, ha valami igaz, akkor igazság értéke 1, ha pedig hamis, akkor igazság értéke 0.

Ha a szobahőmérsékletről nyilatkozunk, akkor a hőmérsékleti skálán (univerzumon) a következőket mondhatjuk el: -10 és +10 Celsius fok között **hideg** van, itt a **hideg** állítás igazságértéke 1, az összes többi hőmérsékletértékre a **hideg** állítás igazságértéke 0.

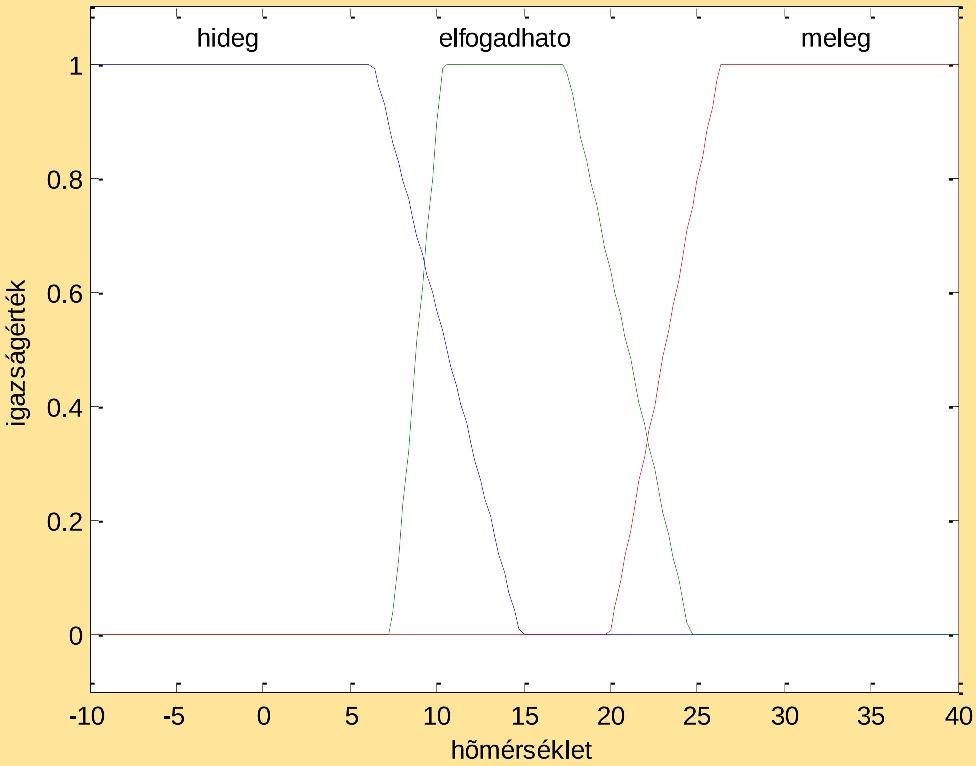
+10 és +21 között **elfogadható** a hőmérséklet, +21 és +40 fok között **meleg** van (a -10 és +40 fokos határokon kívül eső értékeket most ne vegyük figyelembe, ha kell a határok bővíthetők, nincs jelentőségük), és ezt az igazságérték *karakterisztikus függvényével* így ábrázolhatjuk (a következő ábra)[42]:

[42] Az ábrákat, következtetési szabályokat és egyebeket a szerző MATLAB környezetben készítette el.



Éles (crisp) határokkal ábrázolt igazságérték

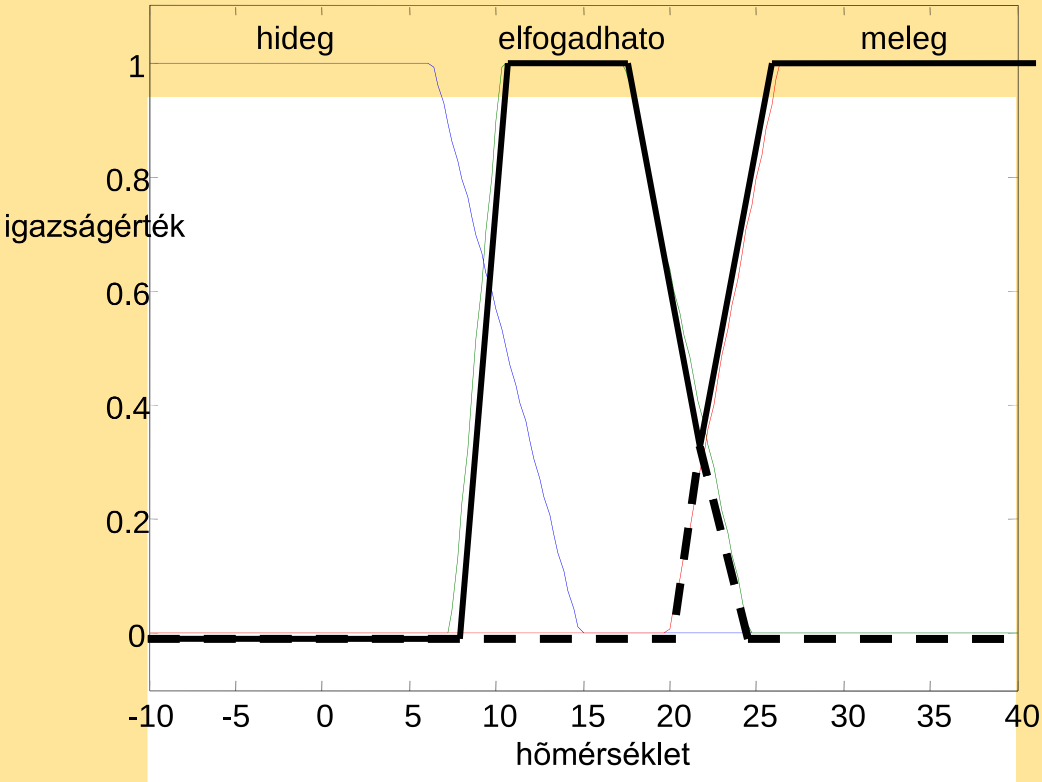
Természetesen a valóságban ezt senki nem így gondolja, mindenkinek más a hőérzete, a megítélése. Azért megegyezhetünk abban, hogy ha egy kicsit másképp alakítjuk az igazságértékek függvényét, akkor elfogadható lesz mindannyiunk számára. Ezeket a függvényeket *tagsági függvényeknek* hívjuk, hiszen azt mutatják, mennyiben tartozik az adott hőmérsékletérték a **hideg, elfogadható** vagy **meleg** tartományba.



Fuzzy halmazokkal (kijelentésekkel) ábrázolt igazságérték.

Látható, hogy egyes hőmérsékleteknél van akinek melege van, van aki elfogadhatónak ítéli meg a szobahőmérsékletet, és az ítéletének "mértéke" valahol 0 és 1 között változik.

Figyeljük meg az ábrán, hogy azt a tartományt, ahol például a **meleg** ***és*** **elfogadható** (logikai *és*) kijelentéseknek is 0-nál nagyobb igazságértéke, pontosan a két tagsági függvény minimuma határolja, egy erősebb szaggatott vonal jelöli. Ahol a **meleg** ***vagy*** **elfogadható** (logikai *vagy*) kijelentéseknek is 0-nál nagyobb igazságértéke, pontosan a két tagsági függvény maximuma határolja, egy erősebb folytonos vonal jelöli.



Fuzzy *és* illetve *vagy*

Hogyan szabályozzuk a hőmérsékletet fuzzy eszköztárral?

Ha hideg van**, erősen** fűtsünk, ha elfogadható a hőmérséklet, akkor **gyengén** fűtsünk, ha meleg van, **nem kell** fűtenünk. Ezek a fuzzy szabályaink.

A következtetési rendszerünk hasonlít a Modus ponensre, de minden szabálykimenetet (következményt: erősen, gyengén, nem kell) olyan mértékben veszünk figyelembe, amilyen szinten azt az adott pillanatban a bementet, a mért hőmérséklet azt indokolja, azaz amilyen szinten találkozik a szabálypremisszával (hideg, elfogadható, meleg).

Ezt a módosított Modus ponenst így értelmezzük:

Ha **A**-ból **B** következik

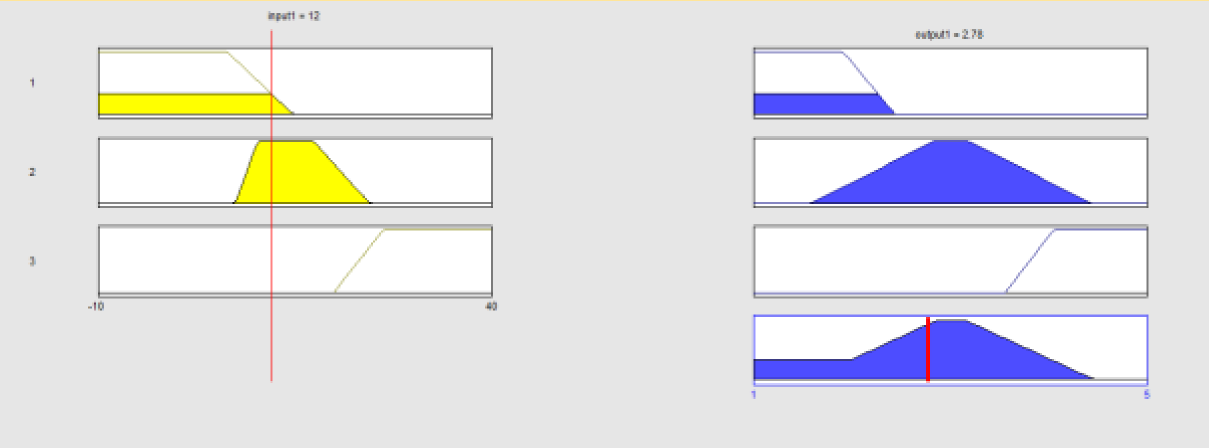
Adott **A'** (azaz körülbelül **A**)

Akkor a következmény **B'**

(azaz körülbelül B, tehát B és B' is olyan szinten egyeznek, amilyen szinten A és A' találkoznak)

A következő ábrán ezt vizualizáljuk: egy-egy sor egy-egy szabályt, illetve a kapcsolódó fuzzy premisszákat (hideg, elfogadható, meleg) és szabálykövetkezményeket (erősen, gyengén, nem kell) ábrázoljuk, látható a mért hőmérséklet (12 fok), és látható az érintettség mértéke a szabályban. Minden egyes szabálykimenetet összegzünk, és egy jellemző fűtési intenzitást adunk meg ennek alapján (a fűtésintenzitást 1 és 5 közé skáláztuk az egyszerűség kedvéért, ahogyan az a hőszabályzókon általában látható). A kimenet 2,78-as fűtési intenzitás.

Így gondoltuk mi is. És nem írtunk fel egyetlen egyenletet sem!



A szabály és következtetési rendszer (Mamdani módszerrel, MATLAB fuzzy Toolbox környezetben)

Természetesen a bonyolult irányítástechnikai problémák, a több-bemenetes fuzzy alapú kockázatkezelők ennél összetettebbek, de általában felhasználónak és szakembernek is egyszerűbben kezelhető eszköztárat biztosít a fuzzy a megoldáshoz, mint a hagyományos módszerek[43] (Takács, 2010).

1.11.2.3 Alapműveletek

Ahogyan azt a példán is láttuk, a Lukasiewitz féle megközelítéshez hasonlóan a minimum és a maximum jól alkalmazhatóak az *és* és *vagy* logikai műveletek modellezésére, ha az állításokat fuzzy tagsági függvényekkel írjuk le. Ugyanakkor gyakran alkalmazzuk azt a két általánosított műveleti csoportot, amelyekbe a minimum és a maximum is tartozik, ezek a t-normák és a konormák.

[43] Az Óbudai Egyetem Neumann János Informatikai Karán több fuzzy-val, lágy számítási módszerekkel foglalkozó kurzus is indul, így a téma részletes tárgyalása ezek keretében történik.

A crisp, azaz "éles" értékű (kétértékű) elméletben definiált halmazokhoz tartozás igazságértékére vonatkozó alapműveletek megfelelőinek meghatározásához alapul a ezezk a Schweizer és Sclar által bevezetett operátor-családok bizonyultak megfelelőnek, hiszen az elvárt tulajdonságokkal rendelkeznek és értéktartományaik is megfelelőek [Kelement, Mesiar, Pap 2000]. Az operátorcsaláddal kapcsolatos további kutatások és gyakorlati alkalmazások megmutatták, hogy különböző problémák esetén különböző származtatott és új bevezetett operátorok lehetnek a legalkalmasabbak, vagyis az operátor kiválasztása alkalmazásfüggő is lehet.

1.11.2.4 Fuzzy **és** *(halmazelméleti szempotntból metszet) - a t-norma*

Legyen adott a $t:[0,1]\times[0,1]\rightarrow[0,1]$ függvény a következő tulajdonságokkal:

1. $t(a,1)=a\ \forall a\in[0,1]$ (peremfeltétel)
2. ha $b\leq c$ akkor $t(a,b)\leq t(a,c)\ \forall a,b,c\in[0,1]$ (monotonitás)
3. $t(a,b)=t(b,a)\ \forall a,b\in[0,1]$ (kommutativitás)
4. $t(a,t(b,c))=t(t(a,b),c)\ \forall a,b,c\in[0,1]$ (asszociativitás)

A fenti tulajdonságokat további megszorításokkal egészíthetjük ki a jobb gyakorlati alkalmazhatóság érdekében.

1. t folytonos függvény
2. $t(a,a)<a$ (szubidempotencia), vagy $t(a,a)=a$ a Zadeh-féle metszetre (idempotencia)
3. ha $a_1<a_2$ és $b_1<b_2$ akkor $t(a_1,b_1)<t(a_2,b_2)$ (szigorú monotonitás) [16].

A t operátor, vagy a gyakran használt elnevezés szerint t norma a halmazelméleti metszet, illetve a logikai ***és*** kapcsolat tulajdonságait hordozza, így két, tagsági függvénnyel leírt fuzzy halmaz, $A(x)$ és $A'(x)$ metszete a következőképpen definiálható (mindkettő ugyanazon X univerzumon definiáltak):

$A(x)\cap A'(x)=t\left(\mu_A(x),\mu_{A'}(x)\right)$.

Logikai jelentése a kijelentésnek a következő: az A ***és*** A' tulajdonsággal rendelkezés igazságértéke a $t\left(\mu_A(x),\mu_{A'}(x)\right)$ függvénnyel számítandó.

A t-normát a fuzzy következtetési rendszer szabályainak feltétel részében használjuk az egyes feltételek összekapcsolására, illetve a Mamdani típusú következtetési szabály alkalmazásakor. A leggyakrabban használt t-norma operátorok a minimum (min) és a szorzat (prod) operátorok

Min operátor (Zadeh-féle t-norma):

$$t(a,b) = min(a,b)$$

Algebrai szorzat:

$$t(a,b) = ab$$

Fuzzy *vagy* (halmazelméleti szempontból unió), azaz a konorma

Legyen adott a $s:[0,1]\times[0,1]\rightarrow[0,1]$ függvény a következő tulajdonságokkal:

1. $s(a,0)=a \ \forall a \in [0,1]$ (peremfeltétel)
2. ha $b \le c$ akkor $s(a,b) \le s(a,c) \ \forall a,b,c \in [0,1]$ (monotonitás)
3. $s(a,b)=s(b,a) \ \forall a,b \in [0,1]$ (kommutativitás)
4. $s(a,s(b,c))=s(s(a,b),c) \ \forall a,b,c \in [0,1]$ (asszociativitás)

A fenti tulajdonságok kiegészíthetők továbbiakkal a jobb gyakorlati alkalmazhatóság érdekében:

1. s folytonos függvény
2. $s(a,a)>a$ (szuperidempotencia), vagy $s(a,a)=a$ a Zadeh-féle unió esetén (idempotencia)
3. ha $a_1<a_2$ és $b_1<b_2$ akkor $s(a_1,b_1)<s(a_2,b_2)$ (szigorú monotonitás).

A t-konorma (a továbbiakban konorma) a t-normához hasonlóan a fuzzy következtetési rendszer szabályainak feltétel részében, abban egyes feltételek összekapcsolására használatos, illetve aggregációs operátorként a szabálykimenet számításakor.

Az s konorma operátor a halmazelméleti unió tulajdonságaival rendelkezik, a logikai ***vagy*** operátor tulajdonságait hordozza. A leggyakrabban használt operátorai a maximum (max) és az algebrai összeg (probor).

Max operátor (Zadeh-féle t-konorma):

$$s(a,b) = max(a,b)$$

Algebrai összeg (Probabilistic OR)

$$s(a,b) = a + b - ab$$

Mind a t-norma, mind a t-konorma családokon belül számos, a fentiektől különböző operátor közül a fuzzy következtetési rendszerekben a feladatnak legmegfelelőbbet választhatjuk.

1.11.2.5 Mamdani féle következtetési rendszer

A fuzzy következtetés (azaz szemantikai számítás) során a bemenetekhez azok jellegétől függően fuzzy halmazok rendelhetők. A két legelterjedtebb változata fuzzy következtetésnek a Mamdani, illetve a Takagi-Sugeno típusú következtetési rendszer, melyek közül a feladathoz jobban illeszkedő választható. Alapvető különbség a két módszer között, hogy míg a Mamdani-típusú következtetés esetén a kimenet általában nem konvex és normális tagsági függvény, amit szükség esetén defuzzifikálni kell, addig a Takagi-Sugeno rendszer esetén a konzekvensek eleve defuzzifikált formában adottak. Ebből következően a Sugeno módszer számításigénye jóval kedvezőbb, ami alkalmassá teszi optimalizációs és adaptív technikákat igénylő rendszerekben való használatra. A hagyományos Mamdani-típusú kiértékelés során a szabályok kimeneteire vonatkozó aggregáció eredménye egy bonyolult alakú tagsági függvény, aminek a defuzzikációja rendkívül számításigényes, de nagy előnye, hogy az emberi gondolkodáshoz jóval közelebb álló modell építhető a segítségével, az intuíció is beépíthető a modellbe. Mindkét rendszerben *HA feltétel AKKOR következmény* típusú természetes nyelvi szabályokat alkalmaznak, a különbség a szabálykimenetben jelenik meg annak megfelelően, hogy a rendszer kimenete crisp érték, vagy fuzzy halmaz. Amennyiben az input paraméterek $x_1, x_2, \dots, x_n$ és a kimeneti paraméter y, a Mamdani-típusú következtetési rendszer a következő felépítésű szabályokkal reprezentálható:

$$IF\ x_1\ is\ A_{1,i_1}\ and\ \dots\ and\ x_n\ is\ A_{n,i_n}\ THEN\ y\ is\ B_{i_1,\dots i_n}$$

ahol A_{k,i_k} a k-adik bemenethez tartozó i_k-adik feltétel (antecedens), $B_{i_1,\dots i_n}$ a szabályok következmény (konzekvens) részéhez tartozó fuzzy halmaz, $i_j=1..n_j$, n_j a j-edik inputhoz tartozó antecedens halmazok száma. A szabálypremisszák a bemenetek fuzzifikált értékeinek összes lehetséges kombinációjából állnak elő.

Takagi-Sugeno típusú következtetési rendszerben a konzekvensek crisp értékek, vagy a bementek függvényeként állíthatók elő. Ha az input paraméterek $x_1, x_2, \ldots, x_n$ és a kimenetek a $g_{i_1,\ldots,i_n}(x_1,\ldots,x_n)$ függvénnyel állíthatók elő, akkor a Sugeno-típusú következtetési rendszer, az alábbi szerkezetű szabályokkal reprezentálható:

$$x_1 \text{ is } A_{1,i_1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{n,i_n} \text{ THEN } y \text{ is } g_{i_1,\ldots,i_n}(x_1,\ldots,x_n)$$

ahol A_{k,i_k} a k-adik bemenethez tartozó i_k-adik antecedens, $g_{i_1,\ldots,i_n}(x_1,\ldots,x_n)$ a szabályok konzekvens része, $i_j=1..n_j$, n_j a j-edik inputhoz tartozó antecedens halmazok száma.

A hagyományos Mamdani-típusú lépései ismertetjük, melyek sorrendben a következők:

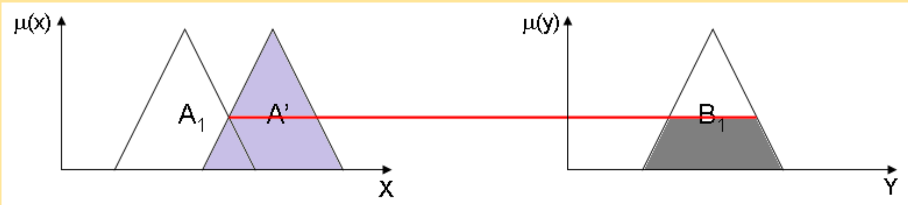
- a megfigyelés (rendszerbemenet) és a szabálypremissza (antecedensek) illesztése, azaz az
if $A(x)$ *then* $B(y)$ szabály és egy $A'(x)$ bemenet illesztése az általánosított Modus ponenes szabályai alapján;
- tüzelési szint kiszámítása a szabályban, azaz $A(x)$ szabálypremissza és az $A'(x)$ bemenet találkozásának szintjét számítjuk;
- fuzzy (Mamdani típusú) implikáció, azaz a tüzelési szint hatását a szabálykimenetre, ami egy módosítása lesz a $B(y)$ szabálykövetkezménynek, jelölje a kapott szabálykimenetet $B'(y)$;
- kimenetek aggregációja, azaz az összes lehetséges szabálykimenetek összesítése, és végül szükség esetén a
- defuzzifikáció, ahol a fuzzy kimenetből kivonatosan egy crisp értéket közvetítünk a környezetei rendszer felé.

Az illeszkedés mértékének meghatározása

A következtetési rendszer bemenetei lehetnek fuzzy számok és crisp értékek egyaránt a bemenet jellegétől függően. A művelet során meg kell határozni a megfigyelések és az antecedens halmazok illeszkedésének mértékét, vagyis azt, hogy az aktuális bemenetek milyen mértékben tartoznak az őket jellemző fuzzy halmazokhoz. Háromszög alakú tagsági függvények esetén az illeszkedés mértéke az *i*-dik szabályban a

$$\sup_{x \in X} (\min(A'(x), A_i(x))$$

képlettel határozható meg. Ez az érték a [0,1] intervallumba esik. Abban az esetben, ha a szabálypremissza egy bemenetet tartalmaz, az illeszkedés mértéke egyben a szabály tüzelési szintje is.



Az illeszkedés mértékének meghatározása fuzzy bemenet esetén

A tüzelési szint meghatározása és a következtetés

A szabályok antecedens része általában több feltétel összekapcsolásával jön létre, ennek kezelésére valamilyen fuzzy operátor alkalmazása szükséges. Az összekapcsolás jellegétől függően ÉS kapcsolat esetén t-norma, VAGY kapcsolat esetén t-konorma operátor
használható.

A következtetés általános matematikai modellje az *i*-dik szabályra vonatkozóan (ahol X a szabálybemenetek univerzuma, Y a szabálykimenetek univerzuma):

$$B_i'(y) = \sup_{x \in X} (T(A'(x), T(A_i(x), B_i(y)))$$

ahol $T(A_i(x), B_i(y))$ a Mamdani által modellezett *if* $A_i(x)$ then $B_i(y)$ szabály, T pedig egy t norma. Ezért nevezzük a módszeren belül ezt Mamdani féle implikációnak.

Ha tovább számolunk és alkalmazzuk a t norma asszociatív tulajdonságát, és ha egy balról folytonos t normáról van szó :

$$B_i'(y) = T\,(\sup_{x\in X}\,(T(A'(x),A_i(x)),B_i(y))$$

Általában a műszaki alkalmazásokban az alkalmazott t norma a minimum:

$$B_i'(y) = \min\,(\sup_{x\in X}\,(\min(A'(x),A_i(x)),B_i(y))$$

ahol most már mondhatjuk, hogy a *DOF -degree of firing* tözelési érték

$$\sup_{x\in X}\,(\min(A'(x),A_i(x), \text{azaz}$$

$$B_i'(y) = \min(DOF,B_i(y)).$$

Az következtetési szabály célja az adott szabály kimenetének meghatározása úgy, hogy a szabályhoz tartozó konzekvens halmazt és a premisszához tartozó tüzelési szintet illeszti valamilyen operátor, tipikusan egy t-norma operátor segítségével. Az illesztés eredményeként létrejött fuzzy halmaz lesz a szabály kimenete. A leggyakrabban alkalmazott operátorok a minimum és a szorzat operátor.

Aggregáció

A fuzzy kiértékelés fontos része a kiértékelő szabályokra alkalmazott következtetés eredményeként kapott konzekvens halmazok aggregációja, melynek során ezekből a fuzzy halmazokból különböző műveletek segítségével egyetlen fuzzy halmaz jön létre.

A $h: [0,1]^n \rightarrow [0,1]$ függvény n fuzzy halmazon ($n \geq 2$) értelmezett aggregációs operátor. Ha a függvény argumentumai az X alaphalmazon értelmezett $A_1(x), \ldots, A_n(x)$ fuzzy halmazok, akkor h minden $x \in X$ esetén fuzzy halmazt állít elő az argumentumok tagsági értékeinek segítségével, vagyis $A(x)=h(A_1(x), \ldots, A_n(x))$. Egy jól definiált aggregációs műveletnek ki kell elégítenie a következő axiomatikus feltételeket is:

- h1 axióma: $h(0,\ldots,0)=0$ *és* $h(1,\ldots,1)=1$ (peremfeltételek)
- h2 axióma: h monoton növekvő minden argumentumában, vagyis ha adott két tetszőleges n-es $\langle a_1,\ldots,a_n\rangle$ *és* $\langle b_1,\ldots,b_n\rangle$ ahol $a_i,b_i \in [0,1]$ és $a_i \le b_i$ minden $i \in [1,n]$-re, akkor $h(a_1,\ldots,a_n) \le h(b_1,\ldots,b_n)$
- h3 axióma: h folytonos függvény.

A fenti feltételek mellett további megszorításokat tehetünk:

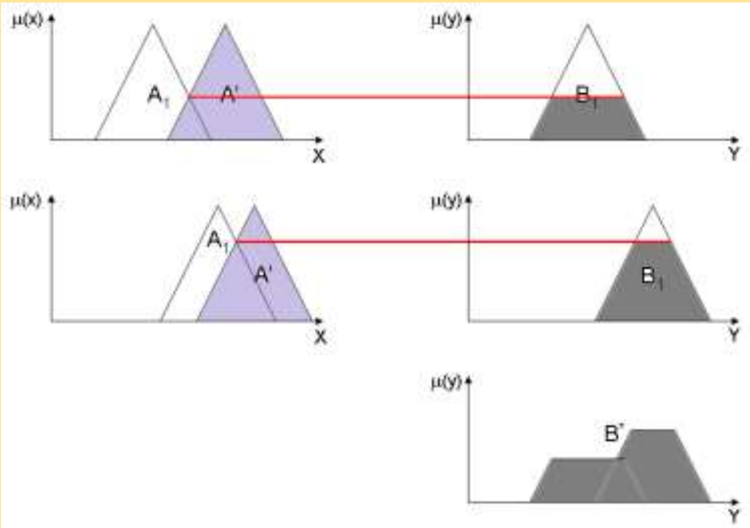
- h4 axióma: h szimmetrikus minden argumentumában, vagyis $h(a_1,\ldots,a_n)=h(a_{p(1)},\ldots,a_{p(n)})$ ahol p az $1,\ldots,n$ számok tetszőleges permutációja.
- h5 axióma: h idempotens, azaz $h(a,\ldots,a)=a$ minden $a \in [0,1]$ esetén.

A fenti öt axiómának eleget tevő aggregációs műveletekre minden $\langle a_1,\ldots,a_n\rangle \in [0,1]^n$ esetén teljesül a $min(a_1,\ldots,a_n) \le h(a_1,\ldots,a_n) \le max(a_1,\ldots,a_n)$ egyenlőtlenség.

A leggyakrabban használt aggregációs módszerek:

- Max: a fuzzy halmazok uniója; $s(a,b)=max(a,b)$
- Sum: a fuzzy halmazok korlátos összege; $s(a,b)=min(a+b,1)$
- Probor: a fuzzy halmazok algebrai összege; $s(a,b)=a+b-ab$

A megfelelő módszer kiválasztása általában az adott feladattól függ, nem lehet általánosságban meghatározni, hogy melyik a legjobb módszer, szükség esetén a fentiektől eltérő aggregációs operátorok is használhatók.



Kimeneti halmazok aggregációja minimum alapú következtetés és maximum alapúú aggregáció esetén

Defuzzifikáció

A defuzzifikáció az aggregáció eredményeként kapott fuzzy halmazból állít elő egy crisp értéket abban az esetben, ha kimenetként nem fuzzy halmazra van szükség. Olyan döntéstámogató rendszerekben, ahol a kimenetet emberi kezelőnek kell értelmeznie, nem feltétlenül szükséges a defuzzifikáció, hiszen számára a kapott halmaz több információt hordozhat, jobb értelmezhetőséget eredményezhet. Amennyiben szükséges, a crisp értéket úgy kell meghatározni a különböző defuzzifikációs módszerek segítségével, hogy a rendszert a lehető legjobban jellemezze. Fontos megjegyezni, hogy a defuzzifikáció nem inverz művelete a fuzzifikációnak, a két művelet semmilyen módon nem származtatható egymásból. Az egyik leggyakrabban használt defuzzifikációs módszer a Centroid (COG).

Ahol a kimenet az aggregáció eredményeként kapott tagsági függvény görbéje alatti terület közepe. A módszer alkalmazásának előfeltétele, hogy a B^* teljes aggregált következtetés tartója intervallum legyen, valamint hogy a kimeneti aggregált halmaz nem üres. Kiszámítása a

$$y_{COG} = \int_{y \in B^*} B^*(y)y\,dy \Big/ \int_{y \in B^*} B^*(y)\,dy$$

összefüggés segítségével történik.

2 Formális módszerek

A lehető legegyszerűbben közelítsünk meg mindent, de annál semmivel sem egyszerűbben.

Albert Einstein

2.1 Rendszerelméleti alapfogalmak

2.1.1 A rendszer fogalma

A rendszer meghatározott cél érdekében működő egységek halmaza, azaz egymással kölcsönhatásban lévő elemek meghatározott egységben megjelenő sokasága.

A rendszer elemei szervezetten kapcsolódnak egymáshoz: fizikai vagy fogalmi entitások, melyek kölcsönhatásai révén részt vesznek a rendszerhez tartozó új kimenetek (fizikai és/vagy a fogalmi minőségében) létrehozásában.

Bármely rendszer leírása során a következőket kell figyelembe venni:

- meg kell határozni a rendszer elemeit és tulajdonságaikat,
- fel kell tárni az elemek közötti kapcsolatokat,
- le kell írni, hogy az elemek és a közöttük fennálló kapcsolatok halmazából hogyan válik a rendszer működő, dinamikusan változó rendszerré.

A rendszer valós megfigyelése helyett azonban gyakran **modelleket** építünk, és azokon végezzük el kísérleteinket. A modellek a rendszer működőképes másai, és ezek segítségével vizsgáljuk a rendszer lehetséges állapotait, kimeneteit, leállásait és más, a rendszerműködéssel kapcsolatos eseményeket. A modell építésekor le kell zárnunk a modellezendő rendszert, azaz kijelöljük a rendszer egy olyan részét, amelynek a feltételezett viselkedését vizsgálni szeretnénk, hiszen a rendszerek kölcsönhatása, beágyazottsága vitathatatlan, és a kapcsolatrendszerük bonyolultsága miatt csak a vizsgálandó rendszerelemeket és azok viselkedését modellezzük. Így is túlnyúlunk a lezárt alrendszer határain modellezéskor, hiszen a modell bemeneteit és kimeneteit a környezeti rendszerből, szomszédos alrendszerek rendszerből nyerjük, illetve oda kell továbbítanunk. Ha pontosak akarunk lenni, azt kell mondanunk, hogy a modell maga is egy rendszer.

Amikor már csak a modellen, a modell környezetében ellenőrizzük a rendszer lehetséges állapotait, állapotátmeneteit, működését, akkor *szimuláljuk* a rendszerműködést.

A modell általában vizuálisan is megjeleníti a rendszerszereplőket és kapcsolatrendszerüket, és jobb esetben a viselkedésüket is. Ugyanakkor arra kell törekednünk, hogy formális, szintaktikailag és szemantikailag helyes leírást rendeljünk a modellhez, hogy a statikus kapcsolatrendszert (ágensek és azok kapcsolatrendszere) és a dinamikus viselkedést (működését) is matematikai és informatikai eszközökkel vizsgálhassuk. Ehhez a gráfelmélettől kezdődően a logikán keresztül nagyon sokféle elméleti megközelítést használhatunk, használunk.

A modellje építésekor feltárjuk:

- a rendszer szereplőit (ágenseket)
- kapcsolatrendszerüket
- működésüket.

2.1.2 A rendszermodell készítésének életciklusai

A tervezés (specifikáció) időszakában:

- egyértelmű, érthető, teljes, ellentmondás-mentes modellhez ezt támogató terv kell, hogy készüljön;
- szükségszerű az együttműködés a különböző területeken tevékenykedő szakértők között, és elengedhetetlen az interdiszciplináris megközelítés;
- megbízható szolgáltatókat kell választani a kivitelezéshez és jól működő kommunikációt kell szervezni megrendelő és kivitelező között (később is).

A fejlesztés (implementáció) időszakában:

- bizonyítottan helyes rendszermodellre van szükség, amelyet a rendszer

  -verifikációjával és

  - validációjával ellenőrzünk;

- fenntartjuk az egyensúlyt a minőség, a költség és a ráfordított idő között, hiszen ezek kölcsönhatása határozza meg, hogyan állja meg a helyét az elkészült modell;
- az modell működésével kapcsolatban automatizálásra kell törekedni;
- a moduláris szerkezetépítés és a komponensek integrációját szorgalmazzuk, hogy az esetleges módosítások egyszerűbben elvégezhetők legyenek.

A fenntartás - fenntarthatóság időszakában

- a korábban rendszeresen, szabványok szerint elkészült dokumentációt meg kell osztani és
- tisztázni kell a működtetéssel kapcsolatos jogosultságokat.

Természetesen a rendszermodell építésének minden részletére nem tértünk ki, hiszen az több más kurzusnak is témája, és gyakran rendszer-specifikus. Elmondható azonban, hogy egy IT rendszer is tulajdonképpen rendszermodell, hiszen egy probléma megoldására, egy cél érdekében összeállított erőforráshalmaz, melynek elemei a valós feladathoz kapcsolódó adatok, entitások rendszerbeli helyét szerepét állapotait modellezi.

A rendszermodellel kapcsolatban további általános elvárásokat is megfogalmazhatunk. Ilyenek például:

- az informatikai háttér és a szolgáltatás minősége;
- ISO és egyéb szabványoknak való megfeleltetés (a termékminőséggel kapcsolatosan, még akkor is egy előre gyártott, fejlesztőkörnyezetben csak reprodukálás történik konstrukció helyett);
- hibátlan, érthető, átlátható specifikáció (dokumentálás) ;
- zárt, ellentmondás-mentes megfelelés;
- hibátlan implementáció

• minősített (és lehetőleg automatizált) fejlesztés,

• ellenőrizhető automatikus tesztgenerálás (validáció) ;

- megállapodás alapján folyamatos fenntartás, visszacsatolás.

A rendszermodell készítésével kapcsolatban a következő további általános feladatok jelenhetnek meg:

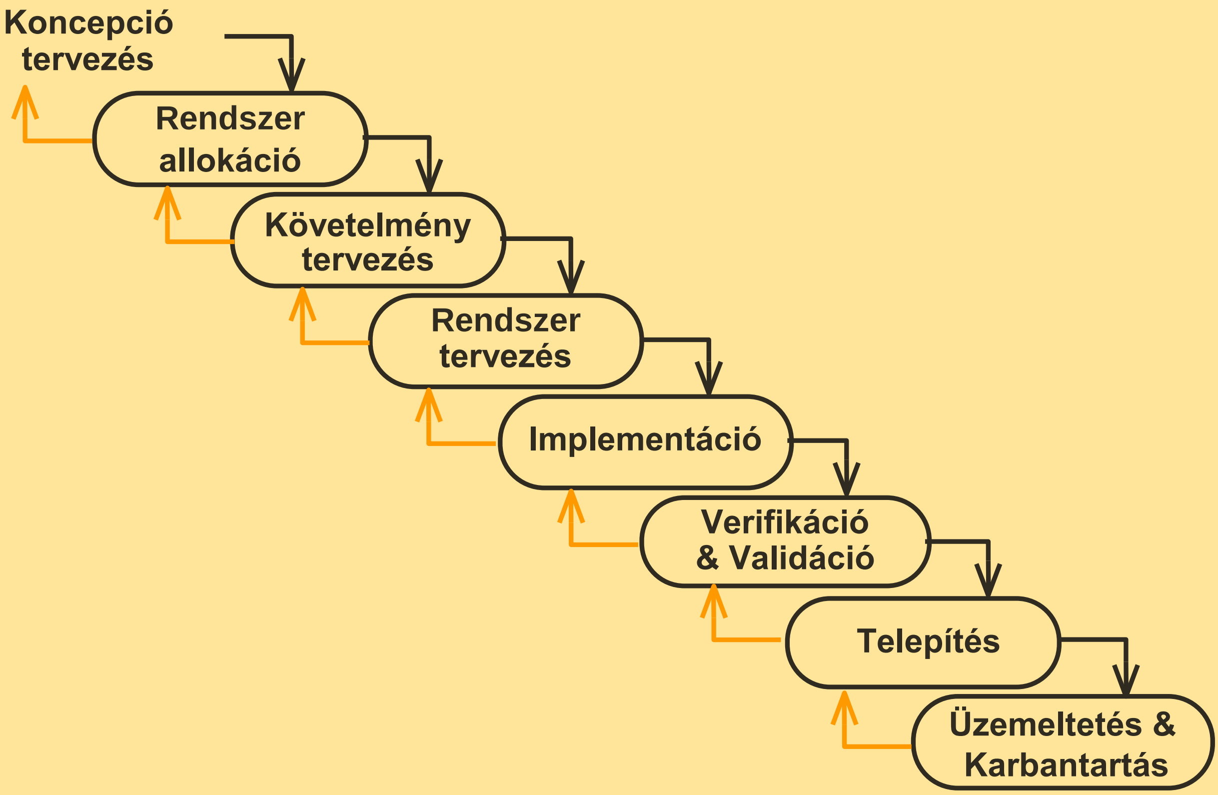
- igényfelmérés, eszköz- és környezetfelmérés;
- részletesebb, vagy valamely kitüntetett alrendszerhez további modell készítése és működésének tesztelése;
- üzembe helyezés és fenntartás megszervezése és mások.

Érdemes azon elgondolkodni, hogy egy IT rendszermodell felépítésekor ezek konkrétan milyen lépéseket jelentenek!

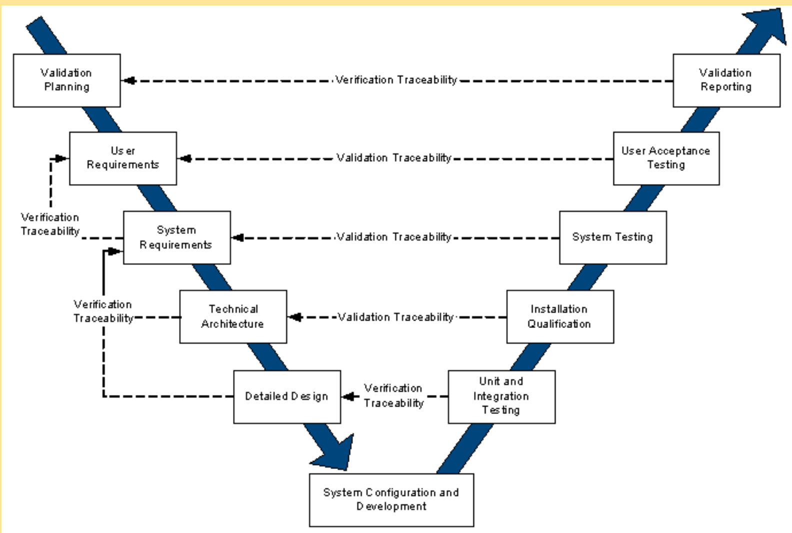
A rendszermodell megalkotásának és a kapcsolódó feladatoknak sematius ábrázolása:

lépések	**miről szól?**	**modellépítési globális lépés**
Követelményanalízis	Mi a megoldandó probléma?	Problémafelvetés
Koncepciótervezés	Milyen megoldási módszerek/eszközök léteznek?	
Rendszertervezés	Hogyan oldható meg a feladat?	Implementáció
IT implementáció	Hogyan valósítható meg a feladat megoldása?	
Tesztelés	Megoldottuk a problémát	
Üzembehelyezés	A megrendelő megfelelőnek tartja a kész rendszert?	Fenntarthatóság
Üzemeltetés, karbantartás	Továbbfejlesztés szükséges?	

A rendszermodell építésének vízesésmodellje arra utal, hogy a fejlesztési lépések mindegyikében lehetőséget adva a verifikációra: visszalépéssel módosíthatunk, illetve javíthatunk a modellen.

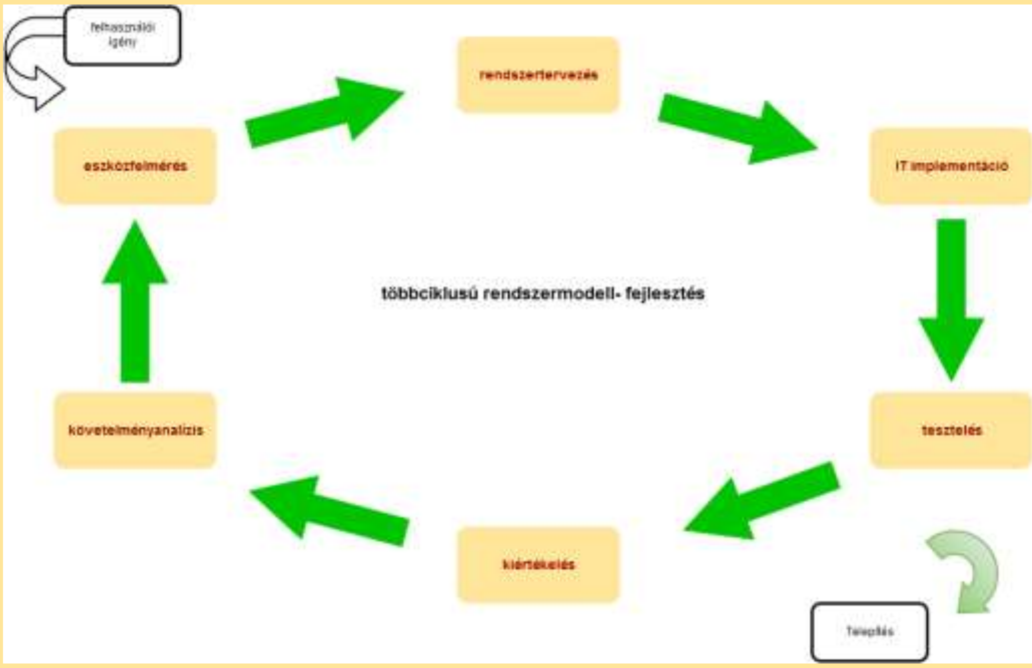


A következő, a V modell kiemeli a modellépítés azon módszerét, hogy a feladatot moduljaira bontva lépésről lépésre készítjük a modellstruktúrát, ugyancsak verifikációs lépésekkel ellenőrizzük, hogy jól építjük-e a modellt, majd amikor a modulokból a teljes modellt felépítettük, akkor validációs módszerekkel ellenőrizzük, hogy jó modellt építettünk-e, azaz a tervezett rendszerfeladatokat szimulálja-e a modell.



Fontos kiemelni tehát, hogy a rendszermodell készítésekor azt a folyamatot, módszert, amikor azt ellenőrizzük, hogy jól építjük-e a rendszert (statikus felépítményét tekintve nem ellentmondásmentes-e például, funkcionalitásában és dinamizmusében pedig hogy helyesen működik-e, például helyes logikai szabályokat alkalmazva-e) verifikációnak nevezzük. Amikor a kész rendszermodell viselkedését vizsgáljuk, tesztelünk, a teljességet vizsgáljuk például, akkor validálunk, ahogyan azt a gráfábrázolás is mutatja.

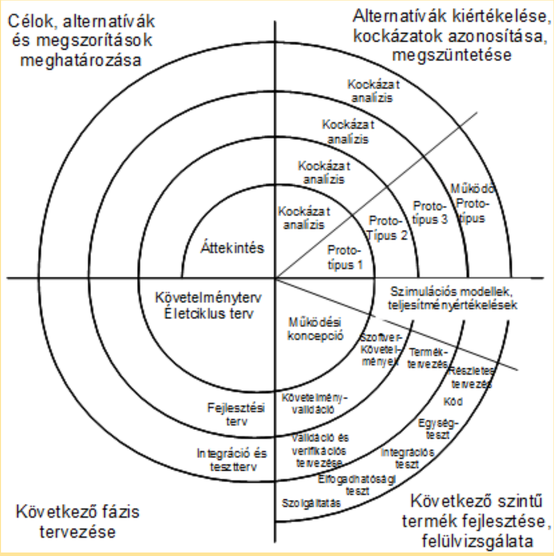
Az eddig bemutatott rendszermodell építési módszerek „egy-verziósak”, egy életciklust írnak le, de a rendszerfejlesztés iteratív: általában egy prototípusban a felfedett hibák alapján következik be a továbbfejlesztés, és megszületik az új prototípus.



A spirális modellfejlesztési séma[44] arra utal, hogy újabb prototípusok fejlesztésével mindig messzebb kerülünk az alapállapottól, azaz a nulla költségtől és a zero időponttól. Ábrázolásban ez azt jelenti, hogy az egyik koordinátarendszerbeli tengely az időé, a másik pedig a költségé. Tehát amikor újabb prototípuson dolgozunk, akkor

[44] http://moodle.autolab.uni-pannon.hu/Mecha_tananyag/szoftverfejlesztesi_folyamatok_magyar/images

már nem a korábbi költség- és idővonalon futunk, hanem egy a korábbitól távolabbi úton, magasabb fejlesztési költség mellett és több ráfordított idővel.



Milyen eszköztárral modellezzünk?

Fontos, hogy

- a modell lehetővé tegye, hogy **szintaktikailag** formalizálható legyen és interpretációja legyen ellenőrizhető **szemantikailag**;

- legyen matematikai eszközrendszere;

- legyen szoftvermegvalósítása.

Alapfeltétel, hogy

- hibátlan feladat- és rendszerspecifikációt írjunk, ennek módját a szabványoknak megfelelően válasszuk;

- modulárisan építkezve törekedjünk a komponensek együttműködésére, a statikus (strukturális) és dinamikus modellekben is;

- a matematikai leírás legyen funkcionális és strukturális (célszerű grafikus formában is megadni).

A végrehajthatóság és ellenőrizhetőség (validáció és verifikáció) szempontjai:

- konzisztencia, ellentmondás-mentesség;

- teljesség, zártság;
- a verifikáció modell-részek szintjén történjen;
- a validáció: a modellek és a rendszer között történjen.

A gyakorlatban gyakran találkozunk a következő problémákkal:

- a valósághű modellezés komplex (modellméret, állapottér méret miatt), „kisméretű" problémákra átlátható, aztán bonyolódik;
- megoldandó az időkezelés, mert hibrid módon is megjelenik, fizikai és logikai skála szerint is;
- a nemlineáris rendszerműködés modellezésénél az idővariánsok kérdése bonyolult eszköztárat igényel;
- a környezet modellezése különböző módon történhet, például lehet modellbázisú és/vagy nem modell bázisú;
- sokféle matematikai területről kell modellező eszközöket választani, és ez széleskörű tudást igényel, és nehéz egységesíteni a többféle matematikai jelölésrendszert és módszertant. A matematikai algoritmus hatékonysága is mindenképpen ellenőrizendő;
- speciális ismeretekre van szükség a felhasználótól: a modellező és a megrendelő összehangolt munkájára van szükség;
- az IT megvalósításhoz sokféle nyelv használható, hatékonyságuk, kifejezőerejük változó.

2.1.3 Modellelmélet, tételbizonyítás, bizonyításelmélet

A bizonyításelmélet részben szintaktikai megközelítésből vizsgálja az elsőrendű nyelvekkel modellezett rendszereket, de a modellelmélet szempontjából fontos hogy a szemantikai következtetések is helyesek legyenek.

Amint arról már beszéltünk, z elsőrendű logikában a predikátumkalkulus a logikai elméletet felépíti axiomatikusan. Megadtunk néhány, általában szemantikai érvek alapján elfogadott formulát, **axiómát,** és néhány szintaktikai **levezetési szabály**t. Ezek az alapvetően elfogadott formulák, melyek segítségével elfogadott formulából elfogadott formulák alakíthatók ki, vezethetők le, következtethetők ki úgy, hogy ami nem így alakul ki, az ne számítson elfogadottnak. Különböző alapokról indulva különböző ilyen *kalkulusok* alakultak ki. A célja és eredménye mindegyiknek ugyanaz: hogy egyrészt minden elfogadott formulát le tudjunk formálisan vezetni (a kalkulus **teljes** legyen), másrészt minden levezethető formula elfogadott legyen szemantikai szempontból is (azaz, hogy a a kalkulus **helyes** legyen). Röviden szólva, minden igazságot (modellben a bekövetkezhető állapotokat, eseményeket) bizonyítani lehessen formálisan, de a rendszerben a lehetetlenre, ellentmondásra a formális következtetési rendszer is mutasson rá. Az elsőrendű Hilbert kalkulus mellett (mi ezzel foglalkoztunk elsősorban), a Frege, Getzen és mások által felépített kalkulusok is ismertek (Vartrész, Kádek, 2012).

Emlékezzünk: egy formula vagy nyelv modelljének egyszerűen egy olyan interpretációt nevezünk, mely a formulát kielégíti. A modellelmélet, amelyben a modell mindenképpen a valóság formalizált mása, a modellinterpretációk egymáshoz és a formulához való viszonyaival foglalkozik. A bizonyításelmélet tehát a modellelmélet szemantikai és szintaktikai alapszabályainak és módszereinek olyan együttese, amelynek célja, hogy a modell és ezáltal a rendszer viselkedésével kapcsolatos folyamatok, állapotátmenetek helyességi, teljességi ellenőrzését végzi. Belőle nőtt ki önálló tudományággá a számításelméleti, számítógépes logika, melynek fő területe az automatikus tételbizonyítás kutatása, ami megint csak a rendszermodellben felépítettek és a logikai szabályok alapján automatikusan ellenőrizhető helyességről és teljességről szól. Legfontosabb elmélete, mint láttuk az előző fejezetekben a rezolúciós kalkulus. Felmerülnek a logika számításelméleti vonatkozásai is, a kielégíthetőségi problémák, a megoldhatóság kutatása és mások, de a jegyzet ezekkel a témákkal egyenlőre nem foglalkozik.

Az ítéletlogikában a szemantikus következtetéseket eldöntés-problémának is nevezzük, de *szintaktikus eldöntés-problémára* is adnunk kellett döntési eljárást, amely megoldása az eredeti szemantikus eldöntésprobléma megoldását vonja maga után. Egy ilyen (formális) döntési eljárás során „adatokból” – egy formulából vagy egy formulahalmazból – kiindulva az eljárás lépései során formulák egy sorozatát állítjuk elő. Ezt a formulasorozatot *levezetésnek* nevezzük. Az eljáráshoz *levezetési szabályok* tartoznak, amelyek formulákból új formulát alkotva lehetővé teszik a levezetés előállítását. Minden döntési eljárás esetén adott egy ún. *megállási feltétel*. Ez általában egy meghatározott formula megjelenése a levezetésben vagy a levezetés speciális szerkezetűvé válása. Egy konkrét döntési eljárás esetén az a kérdés, hogy „adott bemenet mellett elérhető-e a megállási feltétel” úgy is tekinthető, mint a döntési eljáráshoz tartozó (szintaktikus) eldöntésprobléma. Ezen döntési eljárásokat nevezzük gyakorlatilag *kalkulusoknak.* A döntési eljárások két fontos tulajdonságát fogalmazzák meg a helyesség és teljesség meghatározásai:

Azt mondjuk, hogy egy kalkulus *helyes*, ha a szintaktikus eldöntésproblémára adott igen válaszból következik, hogy a szemantikus eldöntésproblémára is igen a válasz. Egy kalkulus *teljes*, ha minden olyan esetben, amikor a szemantikus eldöntésproblémára igen a válasz, a döntési eljárás a szintaktikus eldöntésproblémára adott igen válasszal eléri a megállási feltételt.

A modellezés problematikáján belül tehát a logikai alapprobléma, a *tételbizonyítás* megoldása fontos, és alkalmazható bármely kalkulus, amely mögött következményfogalom és/vagy eldöntésprobléma áll. Ezért mondhatjuk azt, hogy a logika = nyelv + kalkulus. A logika ilyen felépítését (szintaktikus következményfogalom, szintaktikus eldöntésprobléma) szokták *a logika értékmentes tárgyalásának* is nevezni, mivel logikai igazságértékeket, igazságtáblát nem használ.

A rendszer alapján felépített modell tehát a bizonyításelmélet, automatikus tételbizonyítás, eldöntésprobléma eszköztárával tárja fel a rendszerműködés dolgait.

2.1.4 Diszkrét modell- és rendszerállapot tervezés

A diszkrét idejű állapottér modell az időtartományban kitüntetett időpontokban megfigyelt állapotváltozó-értékek alapján vizsgálható. A változók értékei csak ezekben a mintavételi időpontokban ismertek. A továbbiakban diszkrét állapotú, diszkrét idejű, diszkrét eseményterű rendszerek modellezésével foglalkozunk.

2.2 Petri hálók

A Petri-hálók egyidejűleg nyújtanak grafikus és matematikai reprezentációt egy rendszer modelljeként. Alkalmazhatók például:

- konkurens rendszereknél, ahol egyidejűleg működő, önálló egységek kommunikálnak egymással úgy, hogy ezen egységek egymáshoz képest tetszőleges működési fázisban vannak;
- aszinkron, azaz eseményvezérelt rendszereknél;
- elosztott rendszereknél, ahol egyes rendszerelemek között funkcionális tagolódás van, azaz valamilyen megegyezés arról, ki milyen feladatot lásson el a teljes és hatékony működés érdekében;
- párhuzamos rendszereknél, ahol konkurens párhuzamos rendszerek működnek és a rendszerelemek között szoros szinkronizáció áll fenn;
- nemdeterminisztikus és/vagy sztochasztikus rendszereknél, ahol egy-egy adott állapotából nem egyértelmű, melyik állapot lesz a következő.

Petri[45] a 60-as évek elején publikálta a róla elnevezett módszert. Először vegyi folyamatok leírására használta fel, később más alkalmazási területek mellett az operációs rendszerek modellezésére is gyakran alkalmazták.
A rendszer összetevőinek állapotait, illetve események által generált állapotváltozásait modellezi, benne aktív szereplőkkel. Hatékonyan képes modellezni a rendszerrészek élhetőségét, felfedi az esetleges ellentmondásokat, holtpontokat. **Struktúrával** fejezi ki a vezérlést és az adatszerkezetet egyaránt.
Nagy előnye továbbá, *hogy minden más ábrázolásmód kiteríthető Petri-hálóvá*, hátránya viszont, hogy már egyszerű feladatok leírása is hatalmas hálót eredményez.
Kiforrott matematikai háttere miatt ez a leírásmód rendkívül hatékony eszköz lehet rendszerek analízisére, ha a rendszer modelljét valamely kompaktabb modellezésből automatikusan származtatjuk.

---

[45] Carl Adam Petri (1926-2010), német matematikus, informatikus

Strukturálisan, gráfelméleti szempontból irányított, súlyozott, páros gráf.

Kétféle csomópontja van:

- helyek, azaz azok a pozíciók, amelyekben a rendszer szereplői a p-vel, p-ben leírt állapotban, pozícióban vannak : $p \in P$ és

- tranzíció, esemény, amely a szereplőket az egyik pozícióból átvezeti a másikba: $t \in T$.

Az irányított élek (hiszen páros gráf) a

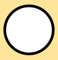
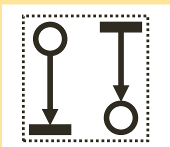
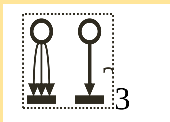
- hely $\rightarrow$ tranzíció illetve a
- tranzíció $\rightarrow$ hely

kapcsolatot írják le, azaz

- $e \in E : (P \times T) \cup (T \times P)$

A hely állapotát a benne levő tokenek (szereplők) száma jellemzi.

Így alakul ki a $PN = (P, T, E, W, M_0)$ struktúra, jelölési rendszerében:

P	T	E	W	$M0$
	vagy 			$M_0 = \begin{bmatrix} m_{1,0} \\ m_{2,0} \\ \vdots \\ m_{n,0} \end{bmatrix}$

.

Hálózat állapotát az egyes helyek állapotainak leírásával adjuk meg. Első matematikai leírásként az M_0 állapotvektorral, amely n pozícióra egy $n \times 1$ dimenziójú mátrix, és m_i komponensei a sorrend szerinti p_i pozícióban a tokenek számát mutatja (pozíciók halmazának számosságát).

A rendszer működése közben a tokenek vándorolnak a rendszerben, azaz átkerülnek más pozíciókba, így az elsődleges M_0 állapotvektorban a komponensek változnak, és a diszkrét időpillanatokban mért token-számok pozíciónként az ***M* token eloszlás vektorban** olvashatóak le (az m_i komponens a p_i helyen található tokenek száma az adott megfigyelt időpontban).

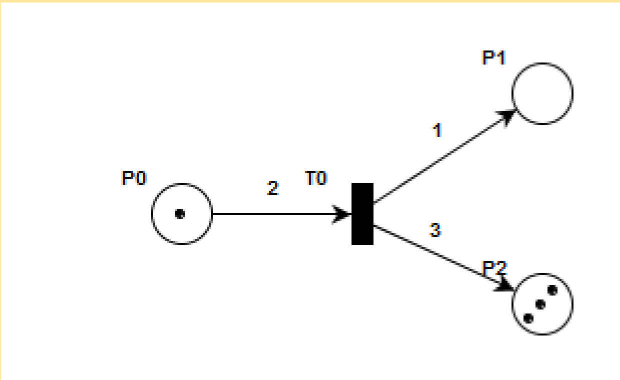
$$M = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix}$$

A kezdőállapotot tehát az M_0 kezdő token elosztás írja le, míg az állapot megváltozásához a tranzíciók „tüzelése”, azaz egy, az állapotváltozást eredményező esemény bekövetkezte vezet.

A tranzíció bekövetkezéséhez először vizsgálnunk kell az engedélyezettséget, azaz megvizsgáljuk, hogy a tranzicíóhoz kapcsolódó pozíciókban (állapotokban) rendelkezésünkre állnak a szereplők. Ha igen, akkor következik a tüzelés végrehajtása, amikor is megtörténik a tokenek elvétele a bemeneti helyekről és a tokenek kirakása a kimeneti helyekre. Mindez új, megváltozott token eloszlás vektort, új állapotot eredményez.

Megjegyzés: az úgynevezett *forrás tranzíciónak* nincs bemenete, és mindig képes tüzelni, a *nyelő tranzíciónak* nincs kimenete, így a tüzelés során a hozzá érkező tokeneket „elnyeli”.

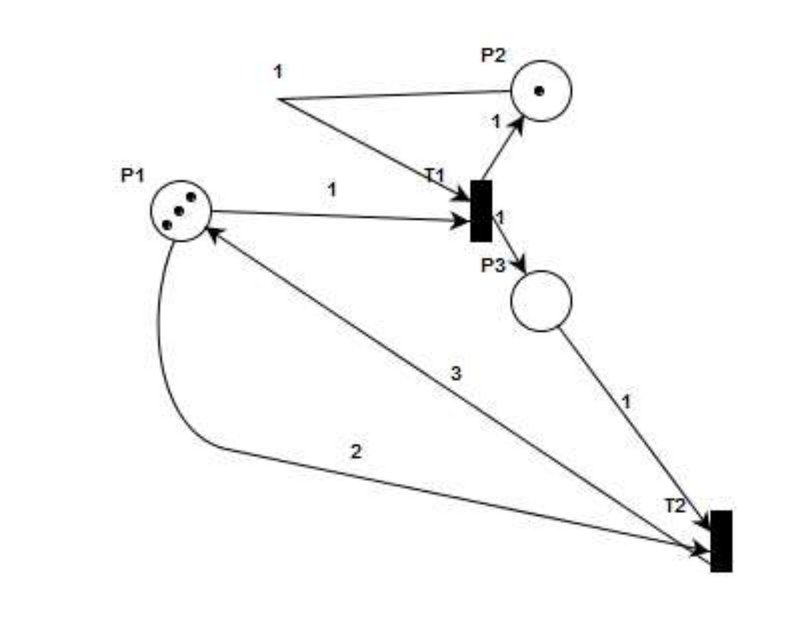
Példa: írjuk fel az ábrán látható PN részlet tokeneloszlás-vektorát.



$$M_0 = \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix}$$

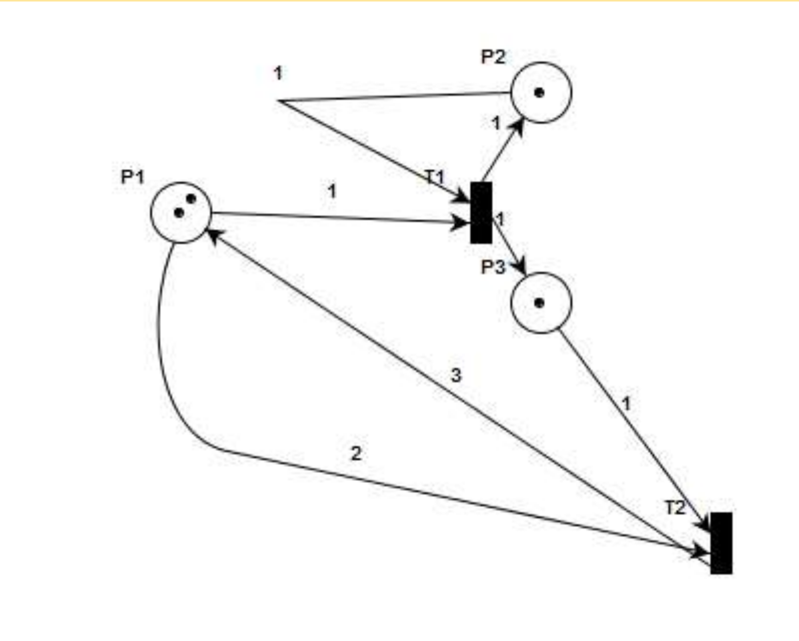
-
-
-

Példa. Figyeljük meg a következő PN-t, és vizsgáljuk meg, van-e tüzelőképes tranzíció, játsszuk le a token-játékot!



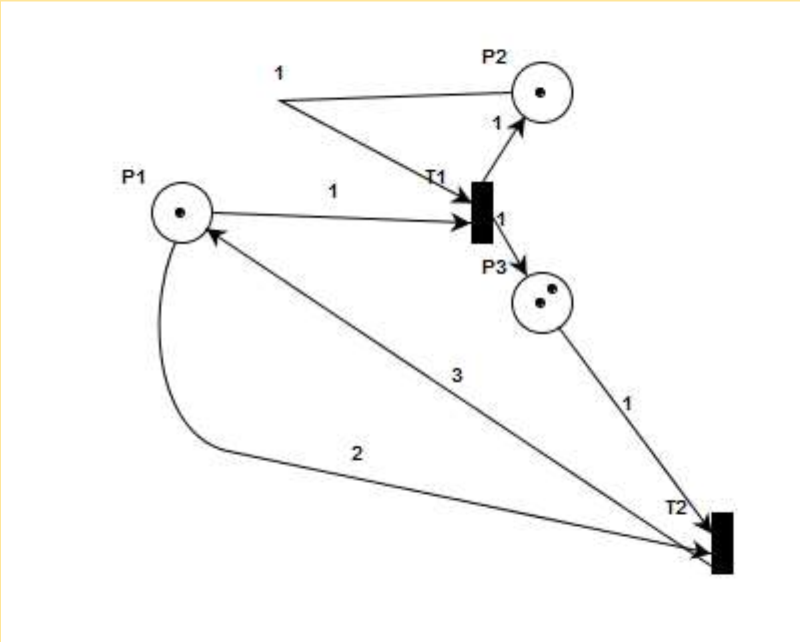
Két élnek van 1-től különböző súlya: a *P1→T2* él súlya 2, a *T2→P1* él 3 súlyú.

Állapotátmenet helyett *tüzelés*: az a folyamat, amely során a tokenek ide-oda vándorolnak a hálón belül. Egy tranzíció akkor tüzel*het*, ha az összes bemenő éléhez csatlakozó helyen van **legalább annyi token, amennyi az adott él súlya**. A következő tüzelőképes tranzíció a T1 lesz.

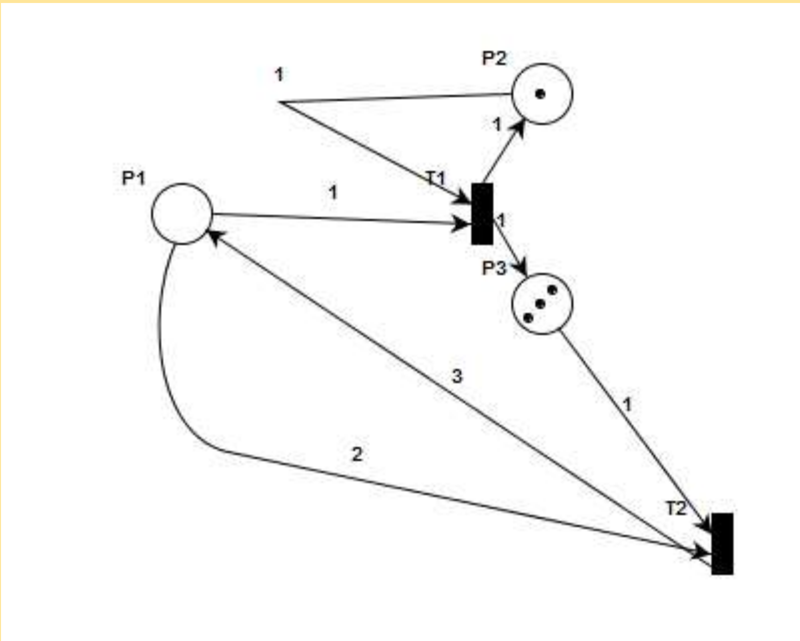


Most T1 és T2 tranzíció is tüzelhető. De melyik legyen az, amelyiket indítjuk? Íme az első olyan probléma, amely miatt azt mondjuk, hogy a PN viselkedése nem determinisztikus. Ha ugyanis egyszerre több tranzíció is tüzelhetővé válik, akkor is egyetlen tranzíció tüzel a következő alkalommal (a következő logikai időpillanatban), de hogy melyik, az előre teljesen kiszámíthatatlan.

Tüzeljen T1. A következő alakul ki:

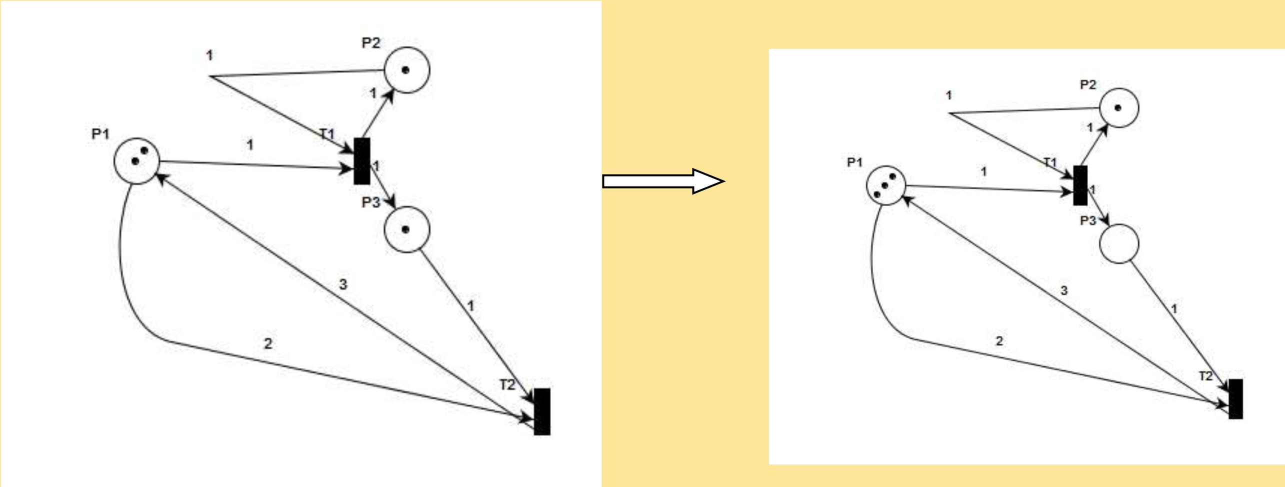


Ha még egyszer hagyjuk a T1 tranzíciót tüzelni, akkor



Ebben az állapotban T1 már nem tüzelőképes, hiszen P1 bemeneti pozíciójában nincs token. T2 sem tüzelhet (és már az előző lépésnél sem tehette), mert P1-ben nem maradt token (illetve nem volt elég az előző lépésben). Így a rendszer holtpontra került.

Lépjünk vissza az első felmerült kérdéses tüzelési ponthoz, és engedjük T2-t tüzelni.



Amint látjuk, visszakaptuk az eredeti állapotot.

Ha felírjuk a T1→T1→T1→T1 tranzíciósorozaban a mátrixok változását a következő sorozatot kapjuk: $\begin{bmatrix}3\\1\\0\end{bmatrix} \to \begin{bmatrix}2\\1\\1\end{bmatrix} \to \begin{bmatrix}1\\1\\2\end{bmatrix} \to \begin{bmatrix}0\\1\\3\end{bmatrix}$

Ha felírjuk a T1→T2→T1 tranzíciósorozaban a mátrixok változását a következő sorozatot kapjuk: $\begin{bmatrix}3\\1\\0\end{bmatrix} \to \begin{bmatrix}2\\1\\1\end{bmatrix} \to \begin{bmatrix}3\\1\\0\end{bmatrix}$.

2.2.1 A Petri háló mint általános állapotleíró módszer

A korábbi példákon is megtapasztalhattuk, hogy a PN nem-determinisztikus véges automata. Az állapotvektor maga a token-eloszlás vektor. Az állapot-átmeneti függvény szerepét a tranzíciók töltik be. Felépítése szerint egy-egy hely egy-egy logikai feltételnek feleltethető meg. A PN struktúrája követi a feladat logikai dekompozícióját. Az állapottét leírását a $PN = (P, T, E, W, M_0)$ PN struktúra komponenseivel végezzük el, ahol:

- a pozíciók halmaza $P = \{p_1, p_2, ..., p_\pi\}$

- tranzíciók halmaza $T = \{t_1, t_2, ..., t_\tau\}$

- ezekre igaz, hogy $P \cap T = \varnothing$

- az élek halmaza: $E \subseteq (P \times T) \cup (T \times P)$

- az élsúlyokat az élsúly-függvény adja meg: $w: E \rightarrow N^{+}$
- a kezdőállapot mátrix $M: P \rightarrow N$.

2.2.1.1 A Petri háló topológiája

Ha $n \in (P \cup T)$ csomópont, akkor $\bullet n$ ősei és $n\bullet$ utódai:

$t \in T$ ősei a bemeneti helyei: $\bullet t = \{p \mid (p,t) \in E\}$

$t \in T$ utódai a kimeneti helyei: $t\bullet = \{p \mid (t,p) \in E\}$

$p \in P$ ősei a bemeneti tranzíciói: $\bullet\, p = \{t \mid (t,p) \in E\}$

$p \in P$ utódai a kimeneti tranzíciói:$p\bullet = \{t \mid (p,t) \in E\}$.

A csomópontokat a $P' \subseteq P$ illetve a tranzíciók $T' \subseteq T$ részhalmazára nézve gyakran jelöljük :

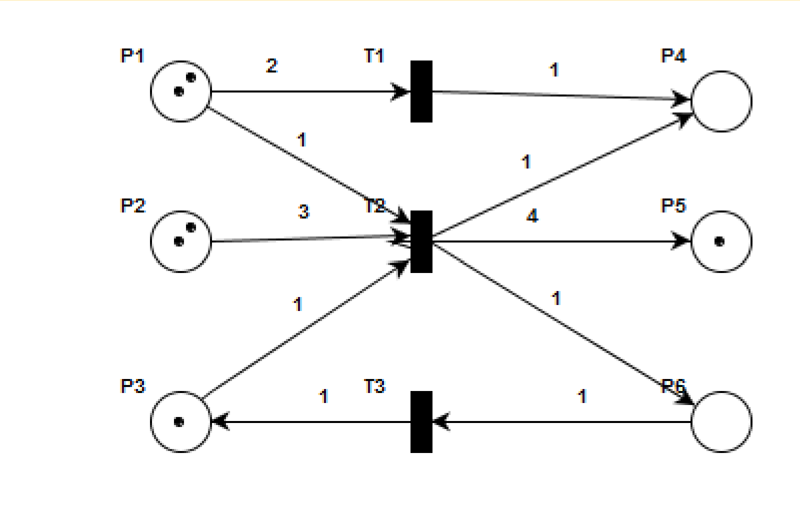
$$\bullet P' = \bigcup_{p \in P'} \bullet p$$
$$P'\bullet = \bigcup_{p \in P'} p\bullet$$
$$\bullet T' = \bigcup_{t \in T'} \bullet t$$
$$T'\bullet = \bigcup_{t \in T'} t\bullet$$

Ha $t \in T$ forrás tranzíció, akkor bemenő hely nélküli ($\bullet t = \emptyset$), illetve ha nyelő tranzíció akkor kimenő hely nélküli illetve ($t\bullet = \emptyset$).

A forrás tranzíció minden esetben tud tüzelni, és általában a külső környezeti hatás alapján teszi ezt.

A PN **tiszta**, ha nincsenek önhurkai, azaz ja $\forall t \in T: \bullet t \cap t\bullet = \emptyset$.

Példa.



$\bullet p_1 = \varnothing, \bullet p_2 = \varnothing, \bullet p_3 = \{t_3\}, \bullet p_4 = \{t_1, t_2\}, \bullet p_5 = \{t_2\}, \bullet p_6 = \{t_2\},$

$p_1 \bullet = \{t_1, t_2\}, p_2 \bullet = \{t_2\}, p_3 \bullet = \{t_2\}, p_4 \bullet = p_5 \bullet = \varnothing, p_6 \bullet = \{t_3\}$

$\bullet t_1 = \{p_1\}, \bullet t_2 = \{p_1, p_2, p_3\}, \bullet t_3 = \{p_6\}, t_1 \bullet = \{p_6\}, t_2 \bullet = \{p_4, p_5, p_6\}, t_3 \bullet = \{p_3\}$

2.2.1.2 A tranzíció tüzelési feltételei

Egy lépés, állapotváltozás, azaz a tranzíció tüzelése szavakkal gyakorlatilag a korábbi állapot megváltozásához vezet. A kezdeti token eloszlás vektor, illetve az előző tüzelés után létrejött token eloszlás vektor megváltozik. A tüzelés végrehajtásához ellenőriznünk kell az engedélyezettséget, majd a tokenek elvétele következik a bemeneti helyekről, utána pedig a tokenek kirakása a kimeneti helyekre. Az új állapotot a megváltozott token eloszlás vektor jellemzi. Fontos megjegyezni, hogy nem a "tokenmegmaradás törvénye" van érvényben, azaz a begyűjtött és kihelyezett tokenek száma nem kötelezően egyenlő. Lehet, hogy több bemeneti helyről is szükségünk van tokenekre, és mégis csak egy pozíción történik változás, egy token erejéig.

A tranzíció tüzelési feltétele matematikai formalizmussal, a topológiai jelüléseket felhasználva így írható le:

Legyen a $t \in T$ tranzíció bemeneti helyéiről a t felé induló élek súlya $w^-(p, t)$, ($w^-(p, t)$ a p -ből t -be vezető $e = (p, t)$ él $w^*(e)$ súlya), akkor a tranzíció tüzelése **engedélyezett**, ha

$$\boxed{\forall p \in \bullet t : m_p \geq w^-(p,t)}$$

A tranzíció tüzelésekor a rendszer:

- elvesz $w^-(p, t)$ darab tokent a $p \in \bullet t$ bemeneti helyekről, ahol $w^-(p, t)$ a $p \to t$ él súlya, és

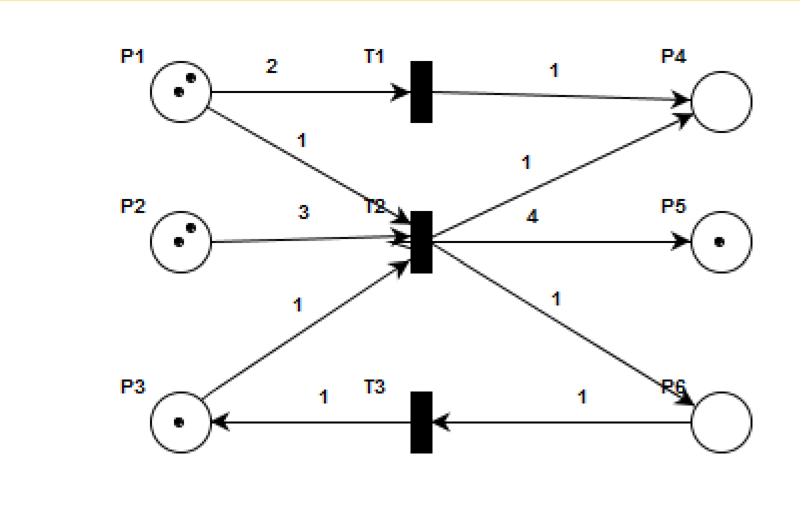
- elhelyez $w^+(t, p)$ darab tokent a $p \in t\bullet$ kimeneti helyekre, ahol $w^+(t, p)$ a $t \to p$ él súlya.

A $W=\left[w(t,p)\right]$ súlyozott szomszédossági mátrix az élsúlyok szerinti tüzelési előfeltételeket modellezi. Dimenziója: π × τ = $|T|$ × $|P|$, elemeit pedig így számítjuk:

$$w(p,t)=\begin{cases} 0 & ha \quad (t,p)\notin E \text{ és } (p,t)\notin E \\ w^{+}(t,p)-w^{-}(p,t) & ha \quad (t,p)\in E \text{ vagy } (p,t)\in E \end{cases}$$

Ha t tüzel, mennyit változik a p -beli tokenszám. A szomszédossági mátrix és az aktuális állapotmátrix alapján számíthatjuk az új állapotmátrixot. A számítás mátrixműveletekkel hajtható végre, részletesen a (Pataricza, 2004) forrásban olvashatunk róla.

Példa:



$$W^{-}=\left[w^{-}(p,t)\right]=\begin{bmatrix}2&0&0&0&0&0\\1&3&1&0&0&0\\0&0&0&0&0&1\end{bmatrix},\ W^{+}=\left[w^{+}(t,p)\right]=\begin{bmatrix}0&0&0&1&0&0\\0&0&0&1&4&1\\0&0&1&0&0&0\end{bmatrix},$$

$$W=\left[w^{+}(t,p)-w^{-}(p,t)\right]=\begin{bmatrix}-2&0&0&1&0&0\\-1&-3&-1&1&4&1\\0&0&1&0&0&-1\end{bmatrix},$$

$$M_0 = \begin{bmatrix} 2 \\ 2 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

A tüzelési feltétel ellenőrzéséhez illesszük egymás mellé a W^- és M_0 mátrixot, mintha szorzást végeznénk el.

	$\begin{bmatrix} 2 \\ 2 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = M_0 \rightarrow \begin{bmatrix} 0 \\ 2 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = M_0$
$W^- = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	W- első sor első eleme illesztve az M0 első elemével látszik, hogy van elég tokenük a t1 tüzelésre, a többi elemnél a mátrixokban nincs elég elem.

Az új tokeneloszlás vektor számításánál a W mátrixot és az M0 mátrixot illesztjük.

	$\begin{bmatrix} 2 \\ 2 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = M_0 \rightarrow \begin{bmatrix} 0 \\ 2 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = M_0$
$W = \begin{bmatrix} -2 & 0 & 0 & 1 & 0 & 0 \\ -1 & -3 & -1 & 1 & 4 & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}$	Adjuk össze a tüzelésre alkalmas sor és az eredeti token eloszlásvektor elemeit!

2.2.1.3 *Hogyan modellezünk Petri hálóval?*

A Petri-háló tevékenységeit, akcióit olyan elemi (atomi) eseményekre bontjuk, amelyek tovább már nem oszthatóak. Esetünkben atomi eseménynek mondjuk egy tranzíció tüzelését, így az összetett események a tüzelési szekvenciákat jelentik (az egymás után végrehajtható tüzelések sorozatát). Az eseményeket a rendszerben szereplő állapotváltozókkal szimuláljuk.

Az állapot-átmeneti trajektória egymást követő tüzelések hatására felvett állapotok sorozata. Az *i*-dik futási sorozat ***tüzelési szekvenciája,*** ami tartalmazza a tüzelő tranzakciókat és a tüzelés után kapott mátrixot jelölése*:*

$$\underline{\sigma} = \langle M_{i0}\, t_{i1}\, M_{i1}\, t_{in}\, M_{in} \rangle \text{ vagy röviden } \langle t_{i1} \ldots t_{in} \rangle.$$

Ha az összes tranzíció kielégíti a tüzelési szabályt, akkor a M_{in} állapot M_{i0}-ból elérhető a $\underline{\sigma}$ tüzelési szekvencia által. Jelölése: M_{i0} [$\underline{\sigma}$ > M_{in} .

Tulajdonképpen úgy is értelmezhetjük, hogy leírtuk az elérhetőség szemantikus követelményét.

A példákon keresztül láttuk, hogy a tüzelési feltételek teljesülése mellett sokszor nem-determinisztikus, véletlenszerűen folytatható az elérhető útvonal. A bizonytalanságokat és azok kiküszöbölésének néhány módját tárgyaljuk a folytatásban.

_Tüzelés végrehajtásakor az engedélyezett tranzíció például tetszése szerint tüzelhet vagy nem.

- Konfliktus helyzet alakulhat ki: több tranzíció engedélyezett, pedig egy lépésben csak egy engedélyezett tranzíció tüzelhet. Választhatjuk-e azt a megoldást, hogy a konfliktusfeloldás véletlen választással tesszük?

A véletlenszerű tüzelésnek további következményei is vannak, például nem a fizikai időzítés szerint hajtódnak végre. Bármelyik konfliktushelyzettel is találkozunk, tovább kell bővítenünk a PN $PN = (P, T, E, W, M_0)$ struktúráját.

2.2.2 Nem-determinisztikus viselkedésből adódó konfliktushelyeztek megoldása

2.2.2.1 *Időzítési probléma*

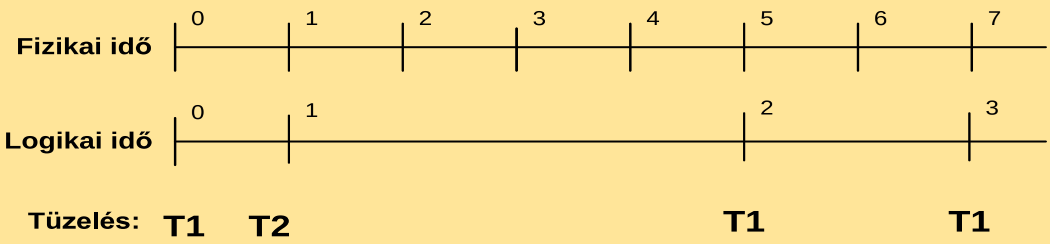
Az implicit időfogalom azt jelenti, hogy az időskálán a tüzelési időpontok legfeljebb eseményvezéreltek, de fizikai időskálán nem pontosíthatók. A tüzelés a [0, ∞) időintervallumban valahol megtörténhet.

Lehetséges megoldás a véletlenszerű indítás kiküszöbölésére, ha a tüzelésekhez tetszőleges konkrét időértéket rendelve, az azonos struktúrájú és kezdőállapotú nem-determinisztikus időzítetlen Petri háló annak minden lehetséges tüzelési szekvenciáját lefedi.

Általában a rendszeren belül kétféle időskálát figyelhetünk meg.

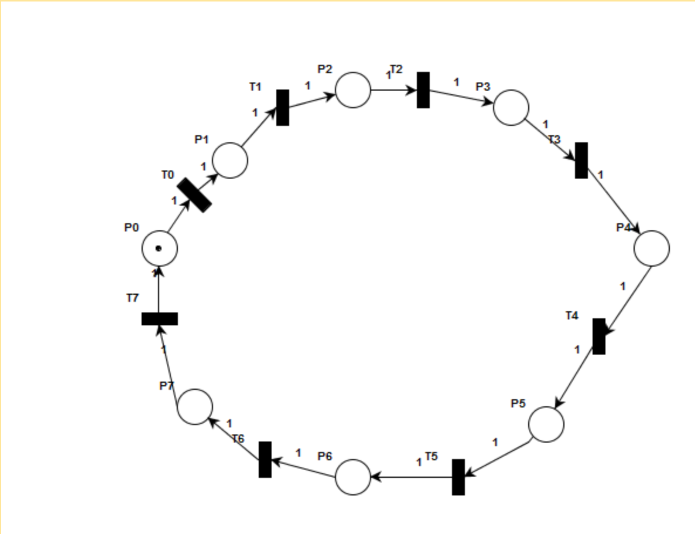
A fizikai idő (óránkkal, a napszakok változásával stb. mérhetjük, tehát ami a rendszertől független), objektív időskálán szemléltethető (többnyire szabályosan periodikus).

A logikai idő: a rendszer működésétől függ, viszonyítási pontjai a bekövetkezett események, Petri-hálók esetében *a tüzelések*.

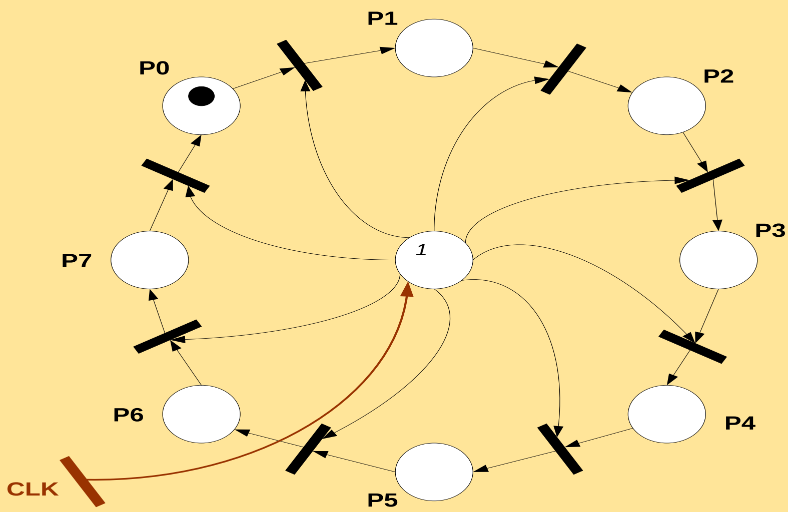


Példa a kiküszöbölésre, azaz arra, hogy ne a logikai idő szerint történjenek a tüzelések, hanem általunk vagy a környezet által szabályozott akár fizikai időskála szerint, akár általunk szabályozva.

Az operációs rendszeren belül a processzoridőt round rubin leosztással például 8 egyenlő hosszúságú időtartományba osztjuk be, hogy a például a 8 aktív processzus az általunk vezérelt időintervallumonként váltsa egymást a processzor felhasználását illetően. Ha nem szabályozzuk az időt, akkor ez a rendszermodell olyan, hogy egy token kering benne, és ha pozíciót vált, akkor átadja a processzort a következő processzusnak. De ez külső beavatkozás nélkül véletlenszerű tüzelésekkel történik. A token minden ugrása a számláló egy ugrását jelenti: kezdetben a token a *P0* helyen van[46].



A következő tüzeléskor a token átkerül az *P1*-es állapotba, majd a *P2*-be stb. A kör végén a *P7*-es helyről ismét a *P0*-ba ugrunk, és kezdődik az egész számlálás elölről. De a tüzelésekhez kötött logikai idő semmit nem mond arról, hogy fizikailag mikor következik be a következő ugrás[47]. Hogy ezt szabályozni tudjuk, első megközelítésben egyetlen forrás-tranzíciót vezetünk be, amely tokeneket juttat a rendszerbe az alábbi módon:



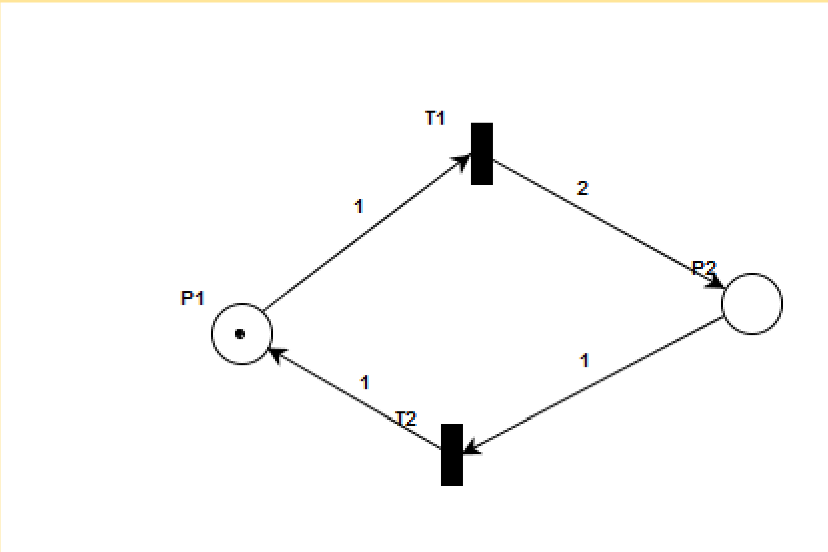
[46] A Petri hálók rajzai jórészt a PIPE nevű, szabadon letölthető programmal készültek. http://pipe2.sourceforge.net/

[47] forrás: Molnár Ágnes: Formális módszerek az informatikában (1) , NetAkadámia Tudástár

A CLK bizonyos időközönként tokent juttat a rendszerbe. Az újabb élek felvételével a tranzíciók már csak akkor tüzelhetnek, ha a középső állapotban van token, azaz „ütött az óra”.

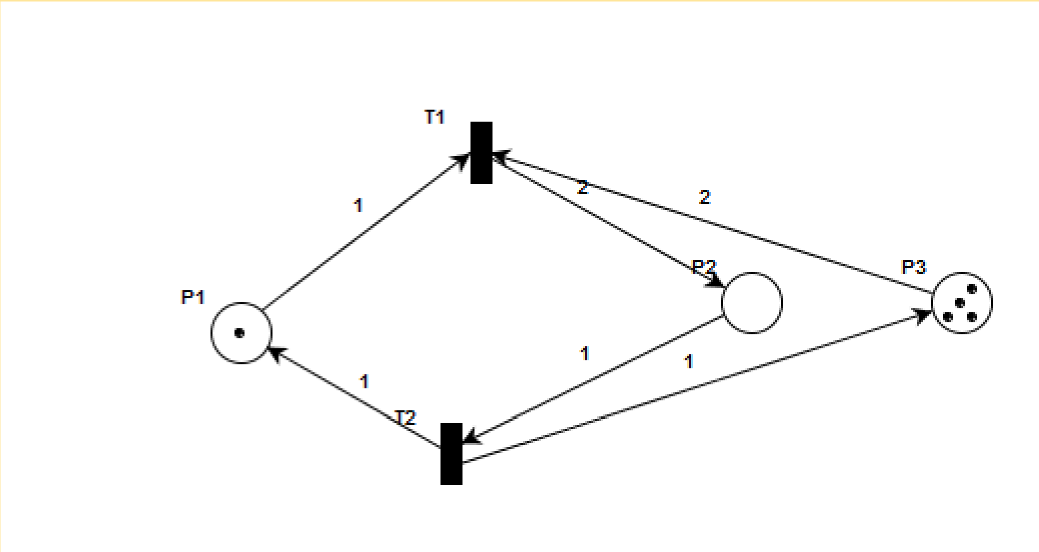
Ha van is órajel-tokenünk, akkor sem garantált, hogy a soron következő tranzíció tüzel a következő óraütés előtt. Sajnos a Petri-hálók sajátosságai miatt ezt nem tudjuk garantálni, de azt igen, hogy az óra ne üthessen addig, amíg az előző órajelre nem történt ugrás a számlálóban. Korlátozzuk tehát a középső állapot kapacitását egyetlen tokenre (ezt jelöli a beleírt 1-es). Ez azt jelenti, hogy azon a helyen maximum 1 token lehet egyszerre, tehát bemenő tranzíció nem tüzelhet, amíg a tüzelés túllépné a **kapacitáskorlátot** (ezt a folytatásban beszéljük meg). Így ha esetünkben egyszer már ütött az óra, tehát van órajel-token a rendszerben, akkor mindaddig nem üthet újra az óra, amíg ez el nem tűnik, azaz amíg a számláló nem lép egyet. Így biztosíthatjuk azt, hogy minden órajelre egyet és pontosan egyet lépjen a számlálónk.

2.2.2.2 *A kapacitáskorlát*



Példa: Ebben a PN-ban a kiindulási helyzetben egyetlen tokenünk van, a P1 helyen. A T1 tranzakció tüzelési feltétele, hogy P1-en legyen token, ezúttal ez teljesül. A T2 tranzakció jelenleg nem engedélyezett, hiszen P2-ben nincs token. *T1* minden egyes tüzelése eggyel megnöveli a tokenek számát a P2-ben, vagyis nincs felső korlát arra, hogy a rendszerben mennyi token lesz, ha tetszőlegesen hosszú ideig magára hagyjuk futás közben. Ezáltal arra sem tudunk korlátot adni, hogy egy-egy állapotban hány tokenünk lesz. Ezt hivatott megoldani a kapacitáskorlát,

Ha bevezetünk egy K(*P2*)=4 kapacitáskorlátot, az azt jelenti, hogy a *P2* helyen maximum 4 token lehet egyszerre, s ez a *T1* tranzíció tüzeléséhez is egy újabb, korlátozó feltételt jelent (ha *P2*-n már 4 token tartózkodik, akkor a *T1* tüzelése nem engedélyezett, hiába van token *P1*-en).



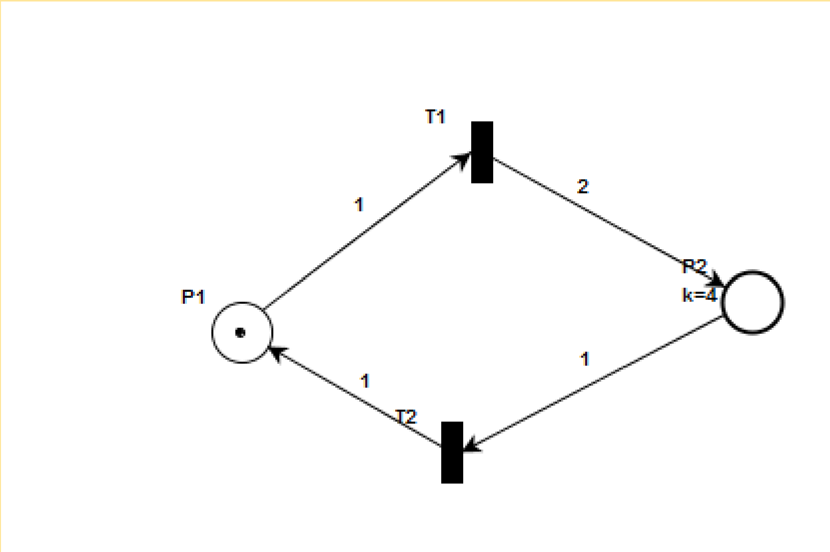
Ha azt akarjuk elérni, hogy a *P2* helyen maximum *k*=*4* darab token legyen, be kell vezetnünk egy **adminisztrációs helyet** (*P3*), ahol azt tartjuk számon, hogy a *P2* helyre hány token fér még el.

Így az adott helyen lévő tokenek számát *m*-mel jelölve mindig igaz lesz az alábbi összefüggés: m(*P3*)+m(*P2*)=*k*.

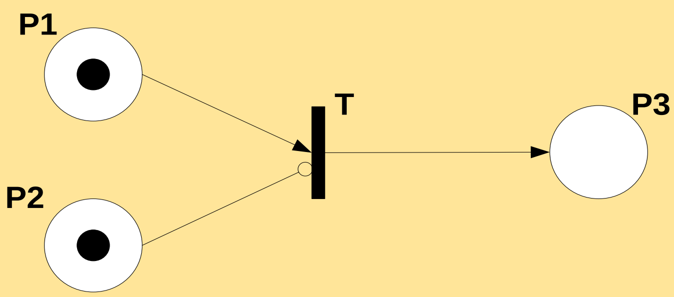
Hogyan felügyelhető, hogy ez minden esetben így legyen? Kezdetben a *P3'* helyen legyen *k* darab token, *P2*-n pedig 0. A *P2*-vel szomszédos tranzakciókhoz vegyünk fel új éleket az alábbiaknak megfelelően: ha a tranzíció *a* darab tokent vesz el a *P2* helyről, akkor az új élen adjon *a* darabot a *P3*-hoz. Ha pedig *b* tokent ad *P2*-höz, akkor ugyanennyit vegyen el *P3*-ból.

A módosított *T1* tranzíció tehát nemcsak hozzáad *P2*-höz 2 db. tokent, hanem ugyanennyit el is vesz a *P3* adminisztrációs helyről. Hasonlóan *T2* elvesz 1 tokent *P2*-ből, és hozzáad egyet *P3*-hoz.

Hogy a gráfot ne terheljük feleslegesen a segédpozíciókkal, a feladatot ellátó pozíció és élek helyett csak *k*=4 kapacitáskorlátot tüntetjük fel, de a háttérben beépítésre kerültek a feladatot elvégző PN elemek.



2.2.2.3 *Tiltó élek bevezetése*



A kapacitáskorlát helyett/mellett **tiltó éleket** vehetünk fel. A tiltó él azt jelöli, hogy a **tranzíció *ne* tüzeljen, amíg az adott feltétel teljesül**. (Az ábrán addig nem tüzel a tranzíció, amíg a P2 helyen van token)

Ha a tiltó élhez egy *k* súlyt is rendelünk, az azt jelenti, hogy ha az él bemeneti helyén az adott *k* számú, vagy annál több token van, akkor a tranzíció tiltott, ha *k*-nál kevesebb token szerepel a helyen, akkor a tranzíció engedélyezett.

Ezzel a módszerrel azt köthetjük például ki, hogy mindaddig nem jöhet óraütés, amíg van token a megfelelő helyen.

2.2.2.4 *A tranzíciókhoz rendelhető prioritás*

Az engedélyezett tranzíciók közül egy alacsonyabb prioritású mindaddig nem tüzelhet, amíg van engedélyezett és magasabb prioritású tranzíció. Ugyanazon prioritási szinten belül továbbra is nemdeterminisztikus választás, a struktúra mindenképpen bővítendő a prioritásokat tartalmazó matematikai leírással, mondjuk legyen egy π sorozat, amely a pozíciókhoz egy például számbelileg kifejezett prioritást rendel:

SPN = (P, T, E, W, M0, π)

2.2.2.5 *Sztochasztikus viselkedés*

A példákban láttuk, hogy a végrehajtási idő lehet:

- determinisztikus (általában teli téglalappal jelöljük), vagy
- sztochasztikus (általában csak a téglalap körvonalait rajzoljuk meg a gráfban).

Ennek alapján az engedélyezettséggel kapcsolatban felmerülő egyik kérdés a következő: az időzítetlen tranzíciók „prioritása" nagyobb-e, előbb tüzelnek?

Vezessünk be egy újabb lehetőséget a PN struktúrába. Ha a tüzelési intenzitás alapján a tranzíciókhoz egy exponenciális eloszlású valószínűségi változót rendelünk, azaz **SPN** = (P, T, E, W, M_0, λ), akkor ennek alapján ütemezhető a gyújtás.

A tüzelési intenzitás $\lambda : T \to R$, az időzítés: ξ valószínűségi változó, eloszlásfüggvénye: $P\{\tau < t\} = F(t) = 1 - e^{-\lambda t}$, a sűrűségfüggvénye: $f(t) = \lambda e^{-\lambda t}$.

A tüzelési úgy szabály változik, hogy az intenzitás közelíti a prioritást. PN és SPN elérhetőségi gráfja azonos. Tekintettel arra, hogy ez egy (a korábbi tapasztalatokat felhasználva) időzített elérhetőségi gráf, a folytonos Markov láncok módszerével kezeljük.

2.2.2.6 *A színezett Petri hálók (Coloured Petri Nets)*

A színezés, ahol a színek Petri alrendszereket emelnek ki, további lehetőségeket nyújt, hogy alrendszereket elkülönítve további címkékkel, jellemzőkkel látjuk el a PN komponenseit. A lényeges különbségeket, bővítéseket egy táblázatban mutatjuk meg.

Színezetlen Petri hálók	Színezett Petri hálók
zínezetlen tokenek	színes tokenek
tokenek halmaza	tokenek multihalmaza
token manipuláció	adat manipuláció
kezdeti jelölés	inicializáló kifejezések
tiltó élek	őrfeltételek
élsúlyok	élkifejezések (változokkal)
tranzició engedélyezése	lekötés engedélyezése
konfliktus különböző engedélyezett	konfliktus ugyanazon tranzició
ranzíciók között	ngedélyezett lekötései között
ssembly nyelv	magas szintű programnyelv

A színezett PN-re láthatunk példát a négy étkező filozófus problémáját felvázoló Petri hálóban[48].

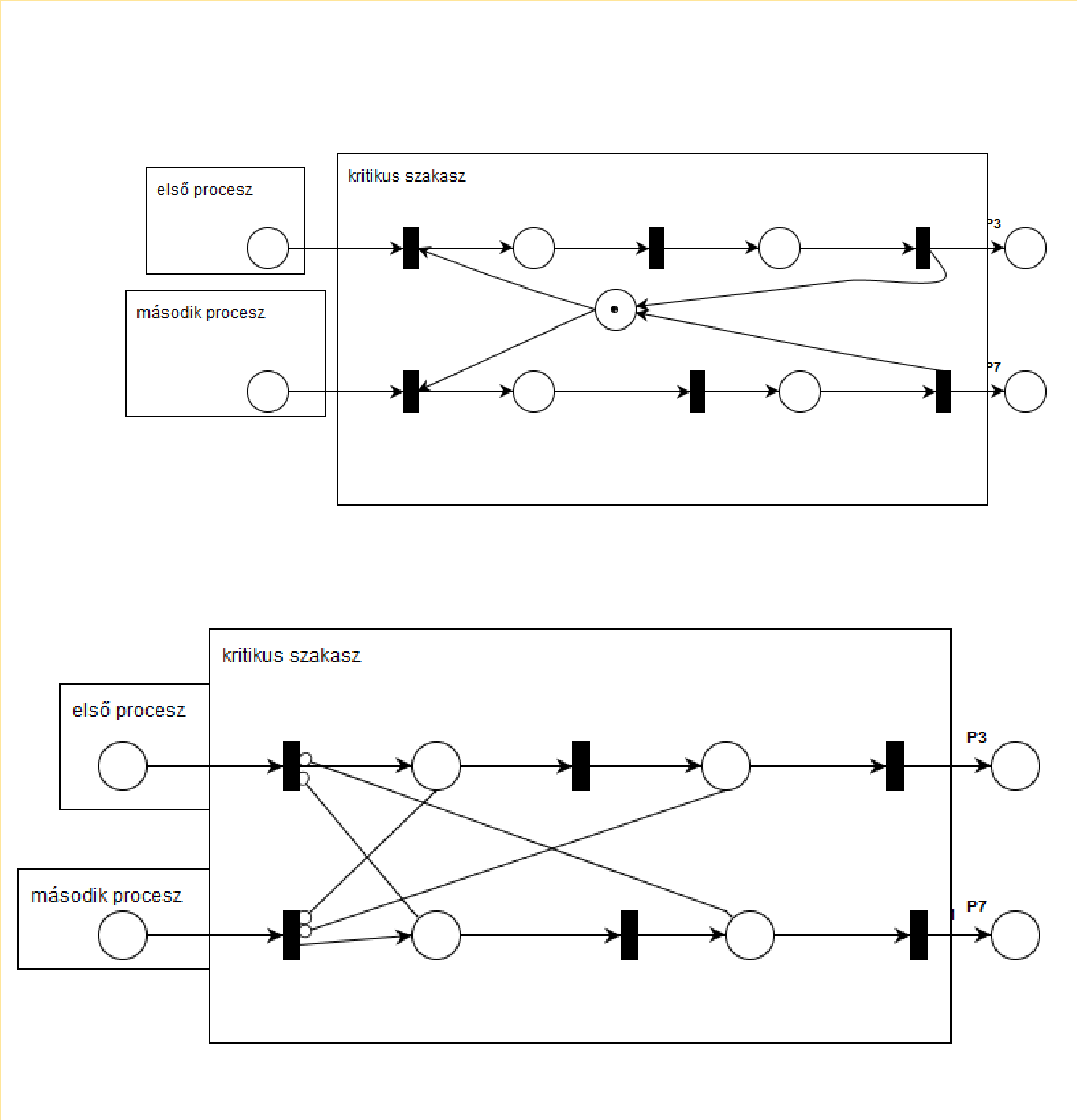
2.2.3 Petri hálós alkalmazások az operációs rendszereknél

Az informatikai rendszerek jól tagoltak, ezért rendszermodell építése a komponensek integrációjával jól kezelhető. Az elemi komponensek kapcsolatában felismerhető a sorrendiség, ok-okozati függőség és az implicit függőség: pl. osztott erőforrás használata esetében. Mindez a sugallja, hogy a Petri hálóval történő modellezés hatékony lehet. A célkitűzés: minőségi vagy/és mennyiségi analízis, mégpedig:

- a kvalitatív, logikai helyesség-bizonyítás, illetve a

- kvantitatív analízis, ahol teljesítményelemzés, megbízhatóság és rendelkezésre állás, biztonságosság dolgait ellenőrizzük.

[48] https://www.informatik.uni-hamburg.de/TGI/PetriNets/introductions/aalst/philosopher4_model.swf

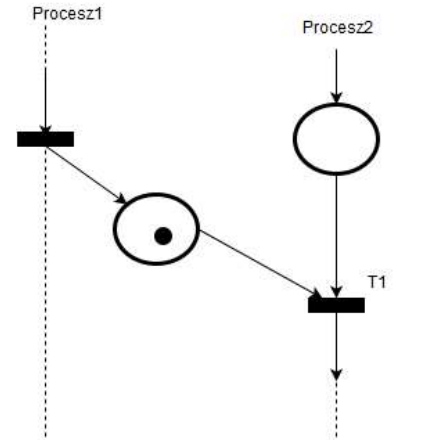


Az operációs rendszerek folyamatainak modellezésekor, Petri hálóval való ábrázolásakor követhetünk egy általános lépéssorozatot. Először különítsük el a fő modellelem-fajtákat (folyamatokat, erőforrásokat, dinamikusan mozgó szereplőket), majd a **folyamatokat, illetve tevékenységeket** kapcsoljuk össze az üzenetváltásaik és az erőforrások használata mentén. Diszjunkt további rendszerelem-részhalmazokat próbálunk meg hozzárendelni elemekhez, hogy elkerüljük modellrészletek túlzott keresztezését.

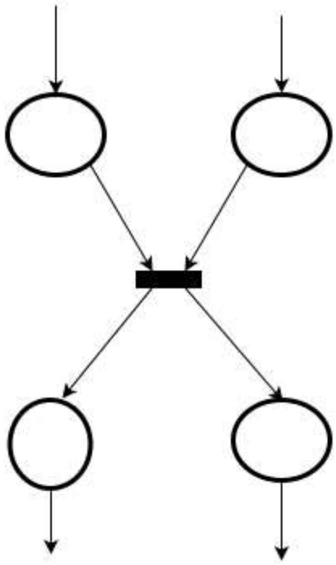
A részletezést egy későbbi finomító lépésben végezzük el. Az **erőforrásokhoz** beépítjük a foglalt/szabad állapotot jellemző kétállapotú véges automata modellrészt. A **logikailag csatolt folyamatok közti esetleges üzenetváltást** lemodellezzük. A folyamatok és erőforrások (illetve az erőforrások foglaltságáról szóló üzenetek puffere) modelljében a kapcsolódó állapotátmenetek összevonását végezzük el. Amint láthatjuk a moduláris építkezés a megfelelő módszer, hiszen gyakran szinte előre-gyártott elemeket kell összefűznünk. Ilyenek például a szemafor típusú üzenetváltások, az állapotváltozók, amelyek a szabad vagy foglalt erőforrásokat jelölik, a fork-join típusú szinkronizálás és mások.

Építsünk először nagy vonalakban, majd részletezzünk. Mindig egy tranzíciót finomítunk, helyettesíthetünk. A behelyettesítendő gráf tranzícióval kezdődjön és végződjön. Az eredeti tranzíció be/kimenő élei ezekbe/ezekből menjenek. Sajnos minden finomítás után a gráf és ezáltal a modell komplexitása növekszik. Gondoljunk például arra, hogy egy tranzíció egy beolvasást jelent. Ha ezt finomítjuk, akkor a bemenő jel hatására a finomításkor olyan részleteket kell modelleznünk, mint például a busz és a belső tár foglalása, a beolvasás helyén a az olvasó pozícionálása, a beolvasandó állomány helyének kikeresése, és így tovább.

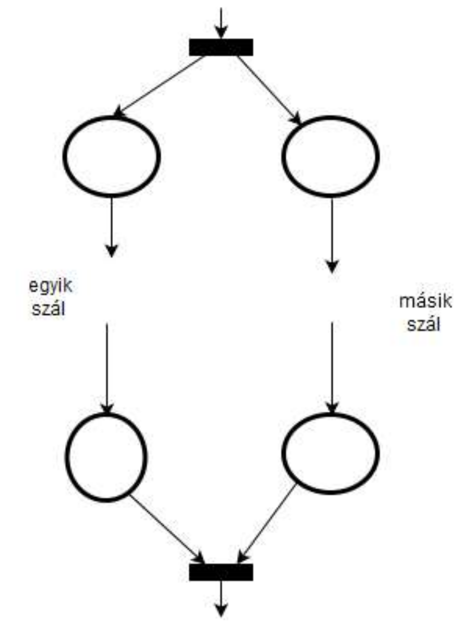
Tipikus helyzetmodellek Petri háló részlettel:



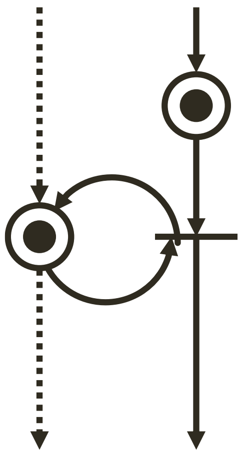
Szemafor típusú üzenetküldés, szinkronizáció



Ranevú típusú szinkronizálás



Szálankénti végrehajtás - fork-join típusú szinkronizált működés

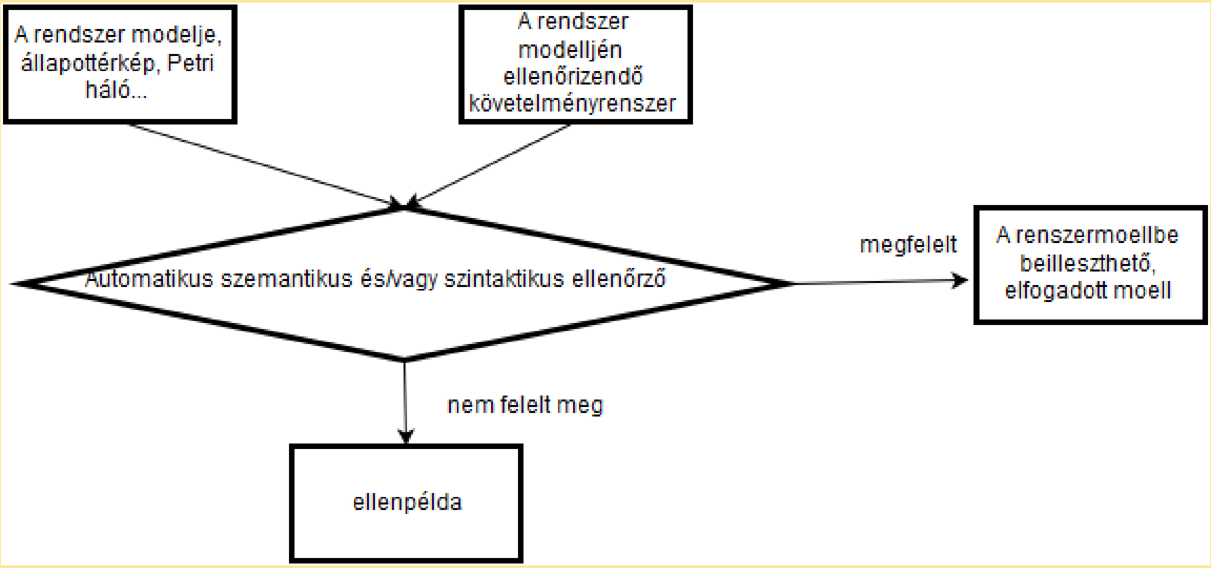


Állapot-változó leolvasása a baloldali proceszből. A jobb oldali csak akkor folytatódik, ha ott talál visszaigazoló tokent

2.3 Temporális logika[49]

A temporális logikát azért tárgyaljuk a rendszermodellek bemutatása után, mert gyakorlatilag egy általánosan elfogadott modellellenőrző eszközről van szó.

Az általános modellellenőrzők felhasználják a megszerkesztett rendszermodellt és az adott helyzetben adott bemeneteket, illetve az adott követelményeket, egy (lehetőleg) automatikus modellellenőrző mechanizmussal, és megállapítják, hogy az adott követelmények kielégítik-e a modellt, és akkor besoroljuk azt, mint működő modellt, vagy ha nem, akkor ellenpéldaként tüntetjük azt fel.



A rendszermodell lehet például Petri háló, vagy más állapottérkép. A dinamikus működés ellenőrzéséhez kellenek a szereplők, a változók értékei, interpretációk, helyettesítések, az ellenőrzés eredménye pedig valamilyen szemantikus vagy levezethető következményfogalom.

A részletes modell elkészítése és manuális ellenőrzése azonban sok esetben nem elégséges. A bizonyítottan jól működő rendszerek iránti igény növekedésével megjelentek az informatikában a különböző formális módszerek. Az rendelkezésre álló egyre nagyobb számítási kapacitás pedig lehetővé tette, hogy a rendszer viselkedését leíró modelleket akár *kimerítő* módon ellenőrizzék, azaz teljes

[49] A fejezet megírásához leginkább Dr. Majzik István: Temporális logikák című tananyagát használtam, elérhető változat: https://inf.mit.bme.hu/sites/default/files/materials/category/kategória/oktatás/doktorandusz-tárgyak/szoftver-verifikációés-validáció /11/SZVV_EA05_hml_ltl.pdf

állapotterüket bejárják, minden állapotot megvizsgáljanak, és bizonyos fontos tulajdonságok meglétét vagy hiányát így bizonyítsák. Az ilyen módon működő modell ellenőrző eszközök először a hardvertervezésben jelentek meg, majd fokozatosan alkalmazták őket a protokollok és szoftverek ellenőrzésében is. A modellellenőrzés során a modell specifikálására valamilyen matematikailag jól kezelhető formalizmus szolgál, például a temporális logika Kripke-struktúrája.

Fontos, hogy a rendszermodell élhetőségét, teljességét, ellentmondás-mentességét ellenőrizzük. Ez azt jelenti többek között, hogy

- valamilyen időben változó állapot-átmeneti sorozat után elérhető (élhető a rendszer) a kijelölt cél, hogy
- eközben minden lehetséges állapotot és átmenetet felvázoltunk (teljesség), és
- nem mond ellent egymásnak több lehetséges útvonal (ellentmondás-mentesség).

Az ellenőrizni kívánt tulajdonságok megadására az úgynevezett temporális logikák használatosak. Ezek az időbeliség, sorrendiség kezelésére temporális operátorokat vezetnek be (például „mindig igaz" vagy „valamikor igaz lesz"). A sokféle temporális logika közül a modell ellenőrzésben kettő használata terjedt el, ezek a PLTL(Propositional Linear Time Logic) és a CTL (Computational Tree Logic). Ezek ugyan kijelentés-logikai lapon működnek, de láttuk, hogy az elsőrendű logikai modellek esetében is a halmaz meghatározása, az interpretáció és helyettesítés és egyesítés után már egyértelmű kijelentés-logikával dolgozunk.

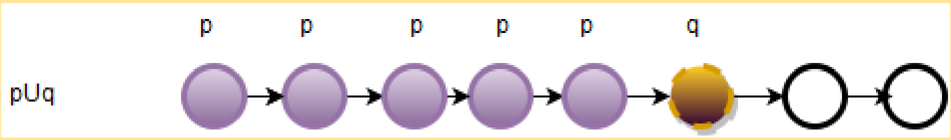
Az atomi kijelentéseket az ismert logikai operátorcsaláddal (és, vagy, negáció, ...) kapcsoljuk. A PLTL-ben az időben kialakult lépéssorozat lineáris. Egy olyan idővonalon mozgunk, amely olyan egyenes (a rendszer egymás utáni állapotainak sorozata), amin a temporális operátorok értelmezhetőek. Az állapot-sorozat újabb és újabb pontjaiban (elemeiben) vizsgáljuk az atomi kijelentések értékét.

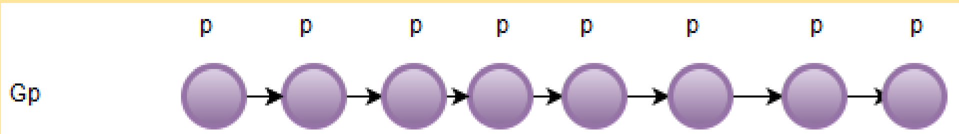
A fontosabb operátorok, amelyekkel az állapotsorozat időbeli kialakulását mutatják a következők:

- G p: p kijelentés minden időpillanatban igaz (General),
- F p: p kijelentés majd valamikor igaz lesz (Future),

• X p: p kijelentés a következő időpillanatban igaz lesz (a PLTL is diszkrét időt feltételez, így értelmezett a következő időpillanat fogalma) (neXt),

• p U q: p minden állapotban igaz, amíg q igaz ne m lesz (Until).





Nézzünk néhány példát LTL kifejezésekre:

• **G** (p ⇒ *F*q): **minden** állapotban igaz, hogy ha p igaz lesz, akkor *utána valamikor* q is igaz lesz (például, ha kiadunk egy kérést, arra mindig kapunk választ).

• F (p ∧ Xq): majd valamikor teljesül, hogy p igaz lesz és közvetlenül utána q is igaz lesz.

• FG p: valamikor olyan állapotba kerülünk, ami után p mindig igaz lesz (például a kezdeti átmenetek után beáll a rendszer egy stabil állapotba).

A CTL (Computational Tree Logic) a PLTL-lel szemben egy elágazó idejű logika, itt egyetlen időpillanathoz több rákövetkező tartozhat, az időbeni állapotváltozások sorozata (vagyis a rendszer által egymás után bejárható állapotok sora) a jelenből kiinduló fába rendezhető. A különböző ágak kezelésére két útvonal kvantor használatos:

- A: minden, az adott állapotból kiinduló úton,
- E: valamelyik, az adott állapotból kiinduló úton.

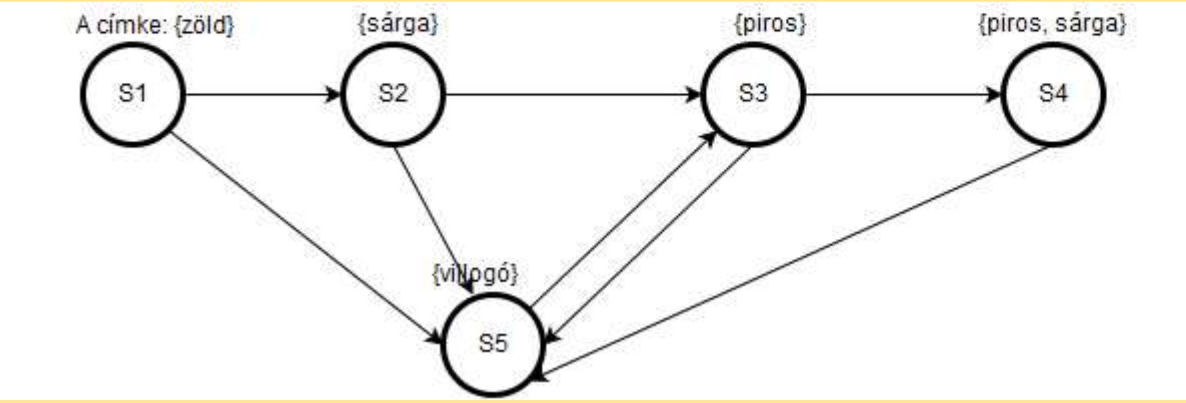
Az atomi kijelentések és az operátorok segítségével formulákat alkothatunk. A CTL operátorok használatát a következő példák szemléltetik:

- **E***G* p: **létezik** olyan lefutás, ahol a p kifejezés *mindig* teljesül (ezt például nem tudjuk kifejezni LTL segítségével).
- AGEF p: bármelyik állapotban vagyunk is a rendszer futása során (AG), létezik olyan további végrehajtás (E), hogy elérünk majd valamikor (F) egy olyan állapotot, ahol p teljesül.

Az alapvető fogalmak megismerése után tehát elmondhatjuk, hogy a modellellenőrzési probléma a következő: egy felépített állapot-átmeneteket modellező *M* struktúrában a p temporális logikai kifejezés igaz-e?

Ha az összes ilyen p-t keressük, akkor globális, ha csak egy adott konkrét állapotra vizsgáljuk a formula teljesülését, akkor lokális modell ellenőrzésről beszélünk. A modellellenőrzés legnagyobb kihívása, hogy egy viszonylag egyszerű modellnek is több százezer állapota lehet, és gyakorlati alkalmazáskor konkurens rendszerekben az egyes lokális állapotátmenetek nagyszámú lehetséges átlapolt végrehajtása miatt exponenciálisan növekszik az állapottér, így az állapottér robbanás jelensége következik be. Ezért nagy jelentőségű az állapottér hatékony tárolása, és a bejárás optimalizálása.

Mielőtt a temporális logika formális szintaktikai és szemantikai leírását megadnánk, nézzünk egy példát. Ábrázoljuk a közlekedési lámpa állapotait és állapotátmeneteit, és nézzünk majd később a felrajzolt modell mellé *követelményt,* azaz egy adott helyzetet, és a temporális logika szemantikája (modellellenőrzés, dinamikus modellellenőrzés) és az adott állapot-átmeneti struktúra (statikus rendszermodell) alapján döntsük el, igaz-e a formula az adott modellen.



Vezessük be a formális leírás elemeit.

2.3.1 A Kripke struktúra

A temporális logika állapottérképe a Kripke-struktúrát használja, ahol a struktúra az M = (S, R, L) hármas, ahol:

- S: állapotok véges halmaza S={s^1,s^2, ...,s^m},
- R: állapot-átmeneti relációk halmaza, R $\subseteq$ S x S,
- L: állapotok címkézése atomi kijelentésekkel, ahol L={l^1,l^2, ..., l^n}.

Az atomi kijelentések között olyan, a rendszerrel kapcsolatos állítások szerepelnek, amelyek már tovább nem bonthatók. Egy állapothoz több címkét is rendelhetünk, azaz minden állapot mellett egy rendezett *m* -mes megmutatja, hogy ott van-e a címke, avagy nincs, azaz egy ilyen állapothoz 2^n lehetőséggel írhatjuk le a címkék odatartozását vagy hiányát.

Ha párhuzamot akarunk vonni különböző rendszermodell ábrázolások között, azt látjuk, hogy a mindegyik modellben találunk megfeleltethető modellelemet, hogy a temporális logika módszerét alkalmazhassuk.

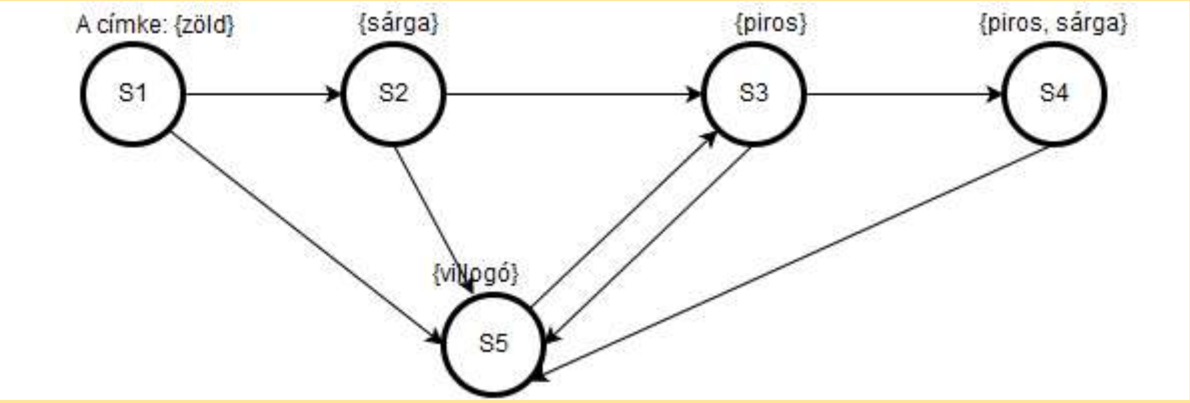
Modellező módszer	**állapotok és átmenetek**	**címkézés**
Petri háló	elérhetőségi Petri gráf	a pozíciók és tranzíciók nevei
Általános állapottérképek	Állapot-konfigurációk, tüzelések	Állapotok nevei
Adatfolyam-hálók	Csomópontok és csatornák állapotai, tüzelések	Állapotok nevei

Példa: közlekedési lámpa állapotai a Kripke struktúra modellben.

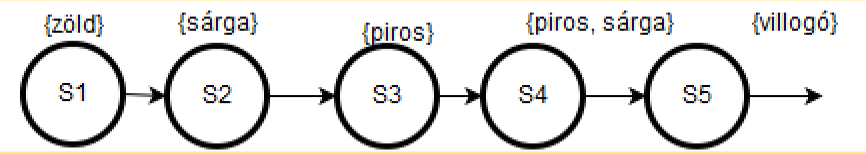
L={piros, sárga, zöld, villogó}.

S={s^1,s^2, s^3, s^4,s^5}.

Az állapotátmenetek gráfja:



A Kripke-struktúra egy-egy útvonalán értelmezhetők a "futások". (pl. egy bemenet hatására). Például:



2.3.2 A PLTL (Propositional Linear Time Temporal Logic) **szintaxisa és szemantikája**

Alapfogalmak:

Atomi kijelentések (S elemei)

Elemi logikai operátorok: $\wedge$ (ÉS), $\vee$ (VAGY), $\neg$ (NEGÁLÁS)

Temporális operátorok:

F p: (*F* uture) "Valamikor p", egy elérhető állapotban igaz lesz p

G p: (*G* lobally)"Mindig p", minden elérhető állapotban igaz lesz p

X p: (ne *X* time)"Következő p", a következő állapotban igaz lesz p

p U q: (*U* ntil)"p amíg q", egy elérhető állapotban igaz lesz q, és addig minden állapotban igaz p ("strong until")

Az érvényes PLTL kifejezések halmaza a következő szabályokkal képezhető: (formális szintaxis)

L1: Minden P atomi kijelentés egy kifejezés.

L2: Ha p és q egy-egy kifejezés, akkor p$\wedge$q illetve $\neg$p is

- L3: Ha p és q egy-egy kifejezés, akkor p U q illetve X p is az.

Ezekből származtathatók:

- p $\vee$q jelentése $\neg$(($\neg$ p) $\wedge$ ($\neg$q))
- p $\rightarrow$ q jelentése ($\neg$p) $\vee$ q
- p $\leftrightarrow$ q jelentése (p$\rightarrow$q) $\wedge$ (q$\rightarrow$p)
- true jelentése p$\vee$ ($\neg$p) (minden állapotra igaz),
- false jelentése p$\wedge$ ($\neg$p) (egy állapotra sem igaz)
- **Fp** jelentése true U p
- **Gp** jelentése $\neg$F($\neg$p)

A szintaxis szabályok alapján képzett kifejezésekhez megadható a formális szemantika:

Egy *p* kifejezést az *M*=(S,R,L) Kripke-struktúra egy $\pi=(s_0,s_1,s_2,...)$ útvonalán értelmezzük, ahol az s_0 a kezdőállapot és $(s_{i,}s_{i+1})\in R$.

- *M,* π |= *p* jelentése: az *M* struktúrában a π útvonalra p igaz (azaz ezen az útvonalon elérhető).
- A $\pi^i=(s_i,s_{i+1},...)$ a jelölése a π útvonal *i*-dik elemétől kezdődő részútvonala.

A szintaktikai szabályokhoz illeszkedő szemantikai szabályok:

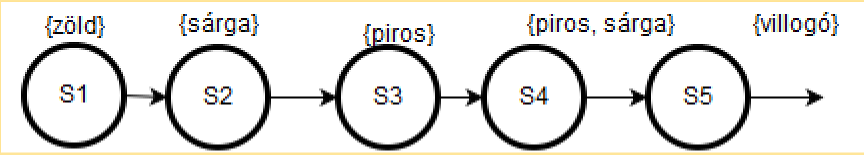
- L1: M,π |= P akkor és csak akkor, ha P$\in$L(s_0), azaz P címke a mostani állapotban látható
- L2: M,π |= p$\wedge$q akkor és csak akkor, ha M,π |= p és M,π |= q

- M,π |= ¬q akkor és csak akkor, ha M,π |= q nem igaz.
- L3: M,π |= (p U q) akkor és csak akkor, ha létezik *j* (π^j |=*q* és bármely *k*<*j* esetében π^k |= *p*), azaz minden olyan részútvonalon amely a π^j részútvonal valós részét képezik, *p* lesz elérhető, a π^j útvonal pedig *q*-hoz vezet.
- M,π |= X*p* akkor és csak akkor, ha π^1 |= *p*.

Származtatott operátorok szemantikája:

- M,π|= F*p* akkor és csak akkor, ha $\exists j$: M,π^j|=*p*
- M,π|= G*p* akkor és csak akkor, ha $\forall j$: M,π^j|=*p*, *j*=1,2,...

Példa: A villanyrendőr példáján, ahol a



útvonalat vizsgáljuk, (jelölje π) igazak például a következők:

M, π|=Zöld, (mert zöld a kezdőállapot címkéje)

M, π|=XSárga, (mert a következő állapotcímke)

M, π|=Fpiros,

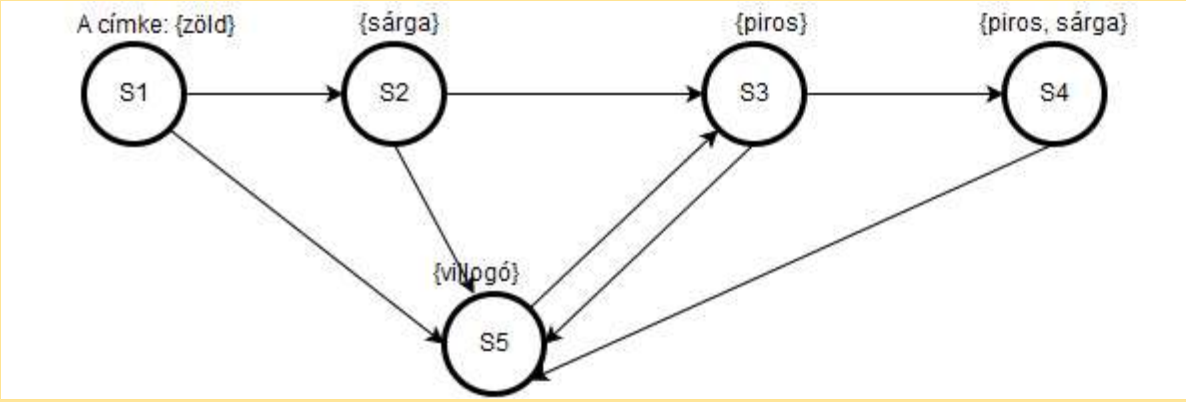
de nem igaz például, hogy:

M, π|=Zöld U Piros, , (mert nem Zöld a címke végig, amíg a Piros címke meg nem jelenik). Használjuk ilyenkor a M, π|≠Zöld U Piros jelölést.

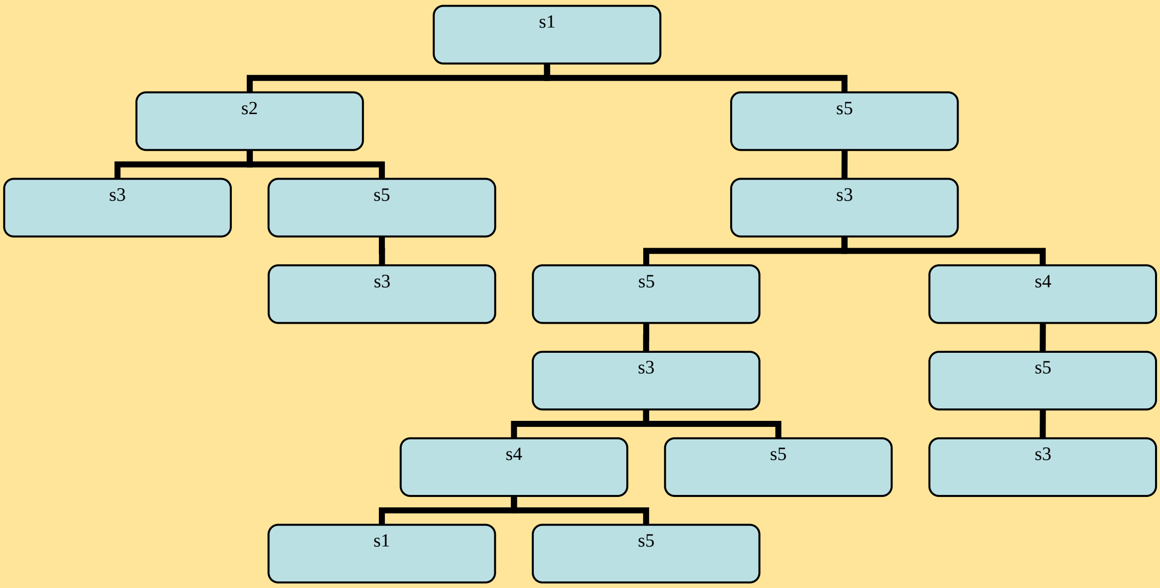
2.3.3 Elágazó idejű temporális logika szemantikája és szintaktikája

Már a villanyrendőr példáján is láttuk, hogy elágazó idő szerinti ábrázolással jobban érzékeltethetjük az állapotátmeneteket. Az egymás utáni állapotok egy fa struktúrát képeznek (minden állapotnak több rákövetkezője lehet). Logikai idő szerint haladva elágazó idővonalak mentén ábrázolhatóak az állapotváltások.

Példa: A villanyrendőr példájának egyszerű állapot-átmeneti gráfja, mint láttuk:



Ha ezt átírjuk fa struktúrába:



Minden levél még tovább bontható. Elindulva az ágak mentén különböző (lineáris) útvonalakat írhatunk le. Kérdés, hogy vajon valamit elérhetünk-e minden megkezdett útvonalon, vagy esetleg csak találunk olyan útvonalat, amelyen elérhetjük, vagy esetleg egyik eset sem fordul elő?

Ennek megfogalmazásában segítenek az útvonalkvantorok.

E*p* (Exists *p*) azt jelenti: létezik legalább egy útvonal, ahol a *p* követelmény teljesül. Egy lehetséges továbblépés mentén vizsgálódik, megfeleltethető a már ismert egzisztenciális kvantor operátornak.

A*p* (forAll *p*): Minden útvonalra fennáll, hogy a *p* követelmény teljesül, azaz minden lehetséges továbblépést magába foglalja a megfogalmazás, megfeleltethető az univerzális kvantor operátornak.

Attól függően, mennyire normalizáljuk a temporális logikai formulákat, külön tárgyaljuk a:

CTL*: (ComputationalTreeLogic*) logkát, ahol az E és A útvonal kvantorok és az útvonalakon az állapotokra vonatkozó temporális operátorok (F, G, X, U) tetszőleges sorrendiségben szerepelhenek.

CTL: (ComputationalTreeLogic) logikát, ahol az állapotokra vonatkozó operátorokat mindig közvetlenül meg kell előznie az útvonal kvantoroknak. Az állapotokra vonatkozó operátorok nem kombinálhatók

FairCTL: (Fair Computational Tree Logic) logikát, amely további megszorításokat vezet be.

2.3.3.1 CTL formális szintaxisa és szemantikája*

Alapfogalmak:

Az útvonalak kvantorai (állapotokon értelmezett):

A: „forallfutures", minden lehetséges útra az adott állapotból kiindulva

E: „forsomefuture", legalább egy útra az adott állapotból kiindulva

•Állapotok operátorai (útvonalakon értelmezett):

–F*p*, G*p*, X*p*, *p*U*q* a korábbi értelmezés szerint.

Néhány példa a CTL* kifejezésekre:

A($p \Rightarrow$Fq) jelentése: minden útvonalra igaz, hogy amennyiben *p* fennáll (az útvonal elejétől), akkor ezt a jövőben olyan állapot fogja követni, ahonnan indulva (az útvonal részútvonalán) *q* fennáll.

E($p \wedge$Gq), jelentése: létezik olyan útvonal, hogy ezen *p* fennáll (az útvonal elejétől), és az útvonalminden részútvonalán *q* is fennáll.

E(XXX$p \vee$Fq), jelentése: létezik olyan útvonal, hogy –vagy ennek negyedik állapotán fennáll *p*–vagy valamikor *q* fennáll az útvonalon.

CTL* szintaxis formálisan:

Állapot-kifejezések (állapotokon kiértékelhető):

S1: Minden P atomi kijelentés egy állapot-kifejezés

S2: Ha *p* és *q* állapot-kifejezések, akkor $\neg p$ és $p \wedge q$ is az.

S3: Ha *p* útvonal-kifejezés, akkor E*p* és A*p* is állapot-kifejezések.

Útvonal-kifejezések (útvonalakon kiértékelhetőek)

P1: Minden állapot-kifejezés útvonal-kifejezés is egyben

P2: Ha *p* és *q* útvonal-kifejezések, akkor $\neg p$ és $p \wedge q$ is az.

P3: Ha *p* és *q* útvonal-kifejezések, akkor X*p* és *p*U*q* is az.

Érvényes CTL* kifejezések azok, amelyeket a fenti szabályok alapján generált állapot-kifejezések adnak meg.

A származtatott operátorok a korábbiakhoz hasonló módon definiálhatjuk. Ezek:

p$\vee$q jelentése ¬((¬p)$\wedge$(¬q)), p→q jelentése (¬p) $\vee$q, p≡q jelentése (p→q)$\wedge$(q→p),

true jelentése p$\vee$(¬p), false jelentése p$\wedge$(¬p)

Fp jelentése trueU p

Gp jelentése ¬F(¬p).

A CTL* szemantika kezdeti megállapításai a lineáris esethez hasonlóan:

Egy *p* kifejezést az *M*=(S,R,L) Kripke-struktúra egy $\pi = (s_0, s_1, s_2, ...)$ útvonalán értelmezzük, ahol az s_0 a kezdőállapot és $(s_i, s_{i+1}) \in R$.

- *M, π |= p* jelentése: az *M* struktúrában a π útvonalra p igaz (azaz ezen az útvonalon elérhető).
- A $\pi^i = (s_i, s_{i+1}, ...)$ a jelölése a π útvonal *i*-dik elemétől kezdődő részútvonala.
- M, π|=*p* jelentése (*p* útvonal-kifejezés): az M modellben a π útvonalon igaz *p*
- M,s|=*p* jelentése (*p* állapot-kifejezés): az M modellben az *s* állapotban igaz *p*.

A szintaktikai szabályokhoz illeszkedő szemantikai szabályok (CTL* szemantika állapot-kifejezései):

M,s |= P akkor és csak akkor, ha P$\in$L(s), azaz P címke a mostani állapotban látható

M,s |= p∧q akkor és csak akkor, ha M,s |= p és M,s |= q

M,s |= ¬q akkor és csak akkor, ha M,s |= q nem igaz.

M,s |= (p U q) akkor és csak akkor, ha létezik *j* (π^j |=*q* és bármely *k*<*j* esetében π^k |= *p*), azaz minden olyan részútvonalon amely a π^j részútvonal valós részét képezik, *p* lesz elérhető, a π^j útvonal pedig *q*-hoz vezet.

Az új operátorok tekintetében:

:–M,*s*|=E*p* (ahol *p* útvonal-kifejezés) akkor és csak akkor, ha létezik π=(s_0, s_1, s_2,…) útvonal M-ben *s*= s_0 mellett, hogy M,π|=p.

–M,*s*|=A*p* (ahol *p* útvonal-kifejezés) akkor és csak akkor, ha minden π=(s_0, s_1, s_2,…) útvonalra M-ben, ahol *s*= s_0, fennáll, hogy M,π|=p.

CTL* szemantikához természetesen meg kell adnunk az útvonal-kifejezéseket is, úgy ahogyan korábban tettük:

M,π|=*p* (ahol *p* állapot-kifejezés) akkor és csak akkor, ha M,s_0|=*p*.

M,π|=¬*p* akkor és csak akkor, ha M,π|=*p* nem igaz

M,π|=$p \wedge q$ akkor és csak akkor, ha M,π|=p és M,π|=q

M, π|=Xp akkor és csak akkor, ha M,π^1|=p.

M,π |= (p U q) akkor és csak akkor, ha létezik j (π^j |=q és bármely $k<j$ esetében π^k |= p), azaz minden olyan részútvonalon amely a π^j részútvonal valós részét képezik, p lesz elérhető, a π^j útvonal pedig q-hoz vezet.

A modellellenőrzés komplexitása miatt, ami Worst-case esetben legalább $O(|S|^2 \times 2^{|p|})$ időkomplexitású ($|S|^2$ az átmenetek száma a Kripke-struktúrában, worst case esetben |p| az operátorok száma a temporális logikai kifejezésben)

Az exponenciális komplexitás a kiértékelések szempontjából igen nagy, bár a temporális logikai kifejezések általában rövidek. Ezért jó lenne a CTL* kifejezéseket egyszerűsíteni, normalizálni. A cél, hogy gyakorlati problémák esetén használható maradjon, de a modell ellenőrzés worst-case időkomplexitása csökkenjen.

E célból alakult ki a CTL (Computational Tree Logic) megközelítés.

2.3.3.2 CTL (Computational Tree Logic)

A CTL állapotokon értelmezhető összetett operátorokat ismer fel csak:

EX p: létezik útvonal, aminek következő állapotán p

EF p: létezik útvonal, aminek egy állapotán p

EG p: létezik útvonal, aminek minden állapotán p

E(pUq): létezik útvonal, amin p amíg q

AX p: minden útvonal következő állapotán p

AF p: minden útvonal egy-egy elérhető állapotán p

AG p: mindenútvonal mindenállapotán p

A(p U q): minden útvonalon p amíg q.

A CTL teljes szemantikai és szintaktikai bevezetése megtalálható a felsorolt forrásokban, csak példákon illusztráljuk a használatát.

Példák:

AG EF p jelentése: bárhonnan indulva olyan állapotba vihető a rendszer, ahol p igaz (operációs rendszerben például a AG EF Reset)

AG AF p jelentése: bárhonnan indulva mindenképpen elérhetünk olyan állapotba, ahol p igaz (operációs rendszerben például AG AF Terminated)

AG($p \rightarrow$AFq), jelentése: bárhonnan indulva teljesül, hogy ha p igaz, akkor valamikor mindenképpen elérhető olyan állapot, ahol q igaz (operációs rendszerben például AG(Request$\rightarrow$AF Reply)).

EF AG p jelentése: lehetséges, hogy a rendszer olyan állapotba kerül, hogy utána p minden állapotban igaz lesz

AF AG p jelentése: bármely úton eljuthatunk olyan állapotba, hogy utána p mindig igaz lesz

AG(p$\rightarrow$A(pUq)): Bármelyik elérhető állapotban ha p igaz, akkor minden úton p fennáll q eléréséig – „p fennáll q eléréséig" pontosabban: elérünk egy olyan állapotba, ahol q igaz, és addig minden állapotban p igaz.

Példa, arra hogy vannak olyan CTL* kifejezések, amelyek nem CTL formátumúak

E(XXX Piros), mert a többszörös X operátor használat miatt egymásba ágyazott útvonal-kifejezések vannak, azaz a két külső X operátort útvonal-kifejezésre alkalmaztuk.

E(XPiros$\vee$FSárga), mert Boole operátor van útvonal-kifejezések között

A(X G (Piros $\wedge$Sárga)), mert egymásba ágyazott útvonal-kifejezések vannak benne.

Példa az operációs rendszerek modelljeiből: Két processzből álló rendszert vizsgálunk, legyenek P1 és P2. Processz állapotok a követelmények szempontjából lehetnek:

P1, P2 rendre a kritikus szakaszában van: C1, C2

P1, P2 rendre a nem-kritikus szakaszában van: N1, N2

P1, P2 rendre a kritikus szakaszba való belépésre kész: W1, W2

Atomi kijelentések (címkék) halmaza AP = {C1, C2, N1, N2, W1, W2}

Modellezzük a következő mondatokat: Egyszerre csak egy processz lehet a kritikus szakaszban: AG ($\neg$(C1 $\wedge$ C2))

Ha egy processz be akar lépni a kritikus szakaszba, akkor előbb-utóbb beléphet:

AG (W1 $\rightarrow$ AF(C1)), AG (W2 $\rightarrow$ AF(C2))

A processzek felváltva kerülnek a kritikus szakaszba; egyikük kilép majd a másik lép be:

AG(C1 $\rightarrow$ A(C1 U ($\neg$C1 $\wedge$ A(($\neg$C1) U C2)))).

2.4 Ágens rendszerek[50]

A rendszer elemeivel kapcsolatban gyakran hangzik el az, hogy ágensek együttműködése alapján működik a rendszer. Röviden ezért kitérünk arra, hogy mit értünk ágens alatt, milyen típusai vannak, és milyen szerkezeti formákban jelenhet meg a rendszerben. Íme néhány definíció az ágensekkel foglalkozó ismert szakemberektől:

"Ágens akármi lehet, amit úgy lehet értelmezni, hogy szenzoraival a környezetét érzékeli és a beavatkozó szerveivel a környezetébe beavatkozik" (Russel, Norvig, 1995)

"Autonóm ágensek olyan számítási rendszerek, amelyek valamilyen komplex dinamikus környezetben tartózkodnak, érzékelnek és ebben a környezetben autonóm módon cselekednek, és ily módon olyan taszkokat vagy célokat valósítanak meg, amire megtervezték őket" (Maes, 1995)

Az ágensek jellegét a környezetükben betöltött szerepük határozza meg általában, hiszen az ágens a környezete nélkül nem ágens. Egy ágens intelligenciája is (racionalitása) a környezet függvénye (szenzorai és visszahatásai által).

Emberi ágensek például a szemek, fülek, egyéb szervek az érzékelésre, kéz, láb, száj, egyéb testrészek a beavatkozásra. A robot ágensek esetében kamerák, infravörös távolsági keresők, és különféle motorok jelentkeznek beavatkozó szervekként.

A szoftver ágensek kódolt bitsorozatok - érzékelnek a bemeneteik által és beavatkoznak a kimeneteik által (arról nem is beszélve, hogy működés közben több erőforrást is "megérintenek").

Az úgynevezett *intelligens ágens* szerepe, jellege és feladatai így foglalhatók össze:

- Érzékelés képessége (külső világ, belső állapot)
- Észlelés (felügyeli az érzékelést és csak a hasznos információt szelektálja)
- Tudás-szerzés (az észlelések és beavatkozások konzerválása)
- Tanulás (folyamat, amely során tudásra vagy gyakorlatra lehet szert tenni, ennek módszerei a soft computing és a szakértői rendszer)

---

[50] A fejezet jórészt Tadeusz P. Dobrowiecki *Kooperatív rendszerek* jegyzete és Pletl Szilveszter *Mesterséges intelligencia II MEMO_01* jegyzete alapján íródott

- Következtetés (megjósoljuk a beavatkozások hatását)
- Döntéshozatal (meghatározza, hogyan kell viselkedni a tudás és az érzékelés alapján. Például mobil robotok esetén itt történik a mozgástervezés és parancsok generálása)
- Beavatkozás

Az intelligens ágensek akkor sorolhatóak be magasabb intelligenciájú csoportba, ha rendelkeznek a következő tulajdonságokkal is:

–kezdeményezés: - cél-orientált viselkedés (felhasználói feladat hiányában az ágens maga fogalmazza meg a feladatait);

–mobilitás - képes helyet változtatni, megtartva saját belsőállapotát;

-következtetés - alapvetően logikai tudásreprezentációval dolgozik, tehát tudnia kell *logikai módon* következtetni;

-tervkészítési készség - a fentiekből értelemszerűen következik;

–tanulás, adaptáció;

- párbeszéd - ahhoz, hogy lássuk, hogy ágens megosztja-e a céljainkat és képes azokat megvalósítani, párbeszédre van szükség, amely tisztázza az intenciókat és képességeket; a párbeszéd eredménye a megegyezés;

- igazmondás -szándékosan nem hazudik a környezetének (segítőkész, ha többen vannak);

- jóindulat - megkíséreli teljesíteni mások kéréseit (ez utóbbiak már emberi tulajdonságokra és viselkedési formákra, érzelmi attitűdökre vonatkoznak).

Szervezeti szempontból kiemelkedő jelentőségűek a multiágensek.

A több, akár egymástól független ágensból (cselekvőből) állórendszereket multi-ágens rendszereknek nevezzük. Multi-ágens rendszerekkel igen gyakran találkozunk a bennünket körülvevő természetes illetve mesterséges környezetben:

– az emberi társadalom maga, és különböző szerveződései,

– a gazdaság szereplői,

– a rovartársadalmak és egyéb állatkolóniák, …

– bonyolult hardver- és szoftverrendszerek…

A rendszerek esetében az egyik legfontosabb, legérdekesebb folyamat a koordináció. Ez az a mechanizmus, melynek segítségével a rendszer szereplőinek tevékenysége összehangolódik a külső korlátok betartása, illetve valamilyen közös cél elérése érdekében. Például:

- a hangyák élelemkeresésének szervezése,
- a vastaps spontán kialakulása is (már emotív elemek függvényében).

2.4.1.1 IT multiágensek

A hálózatok és az elosztottság terjedésével a multi-ágensrendszerek egyre gyakoribbak a számítógépes világban is.

Hagyományosan ilyen a kliens-szerver architektúra, illetve az elosztott problémamegoldás (a szereplők egyazon probléma megoldásán fáradoznak).

Az internet, és vele együtt a "nyílt hálózat" gondolata előtérbe helyezi az egymástól független, csak részben közös célokkal rendelkező rendszereket (lényeges koordinálási problémákkal).

Multi-ágenses környezetben az entitás üzeneteket küld, fogad és feldolgozza azokat. Az üzenetek lehetnek: tájékoztató, felkérő, felajánló, ígéret, elfogadás, visszautasítás jellegűek. Ezek általában is jellemző "üzenetváltási" típusok az ágensek között.

Az ideális *racionális ágens* minden egyes észlelési sorozathoz a benne található tények és a beépített tudása alapján minden elvárható dolgot megtesz a teljesítmény mértékének maximalizálásáért. Megkövetelhetjük a racionális ágenstől, hogy információt gyűjtsön, és hogy lehetőség szerint **tanuljon (learn)** a megfigyeléseiből. Az ágens kezdeti konfigurációja valamilyen előzetes tudást tükrözhet a környezetről, de ahogy az ágens tapasztalatot szerez, ez a tudás módosulhat és átértékelődhet.

Vannak szélsőséges esetek, amikor a környezet a priori teljesen ismert. Ilyen esetekben az ágensnek nem kell érzékelnie vagy tanulnia, egyszerűen csak helyesen cselekszik (bogár esete). Cselekszik, ellenőriz, de nem tanul a „hibából”.

Amikor az ágenst tervezik, a számítás egy részét tervezőik végzik, de elvárható lenne, hogy amikor az ágens megfontolja a következő cselekvését, további számításokat végez; és amikor tanul a tapasztalataiból, még további számításokat végez annak eldöntésére, hogy hogyan módosítsa a viselkedését, azaz folyamatosan *tanul*. Így születik meg az **autonóm ágens**. Addig a szintig, míg az ágens a tervezői által beépített tudásra épít és nem saját megfigyeléseire, azt mondjuk, hogy az ágens nem autonóm.

Egy racionális ágensnek autonómnak kell lennie - mindent, amit csak megtanulhat, meg kell tanulnia ahhoz, hogy a hiányos vagy hibás előzetes tudását kompenzálja. Praktikus okból ritkán kell teljes autonómiát biztosítani már a kezdetektől: amikor egy ágensnek nincs tapasztalata, vagy csak kevés van, akkor véletlenszerűen kellene cselekednie, hacsak a tervezője nem ad valamilyen segítséget. Így míg saját maguk is megtanulják mindazt, ami a túléléshez szükséges, éppúgy, mint ahogyan az evolúció során az állatokat elegendő beépített reflexet szereznek a túléléshez. Az MI (mesterséges intelligencia) ágensek kezdő tudással és tanulási képességgel való ellátása ésszerű elvárás.

Miután a racionális ágens elegendő tapasztalatot szerzett a környezetéről, viselkedése gyakorlatilag függetlenné válhat a beépített előzetes tudásától. Ily módon a tanulás alkalmazásával olyan ágens tervezhető, amely sokféle környezetben is sikeres lesz.

2.4.1.2 Ágens és környezete

Az autonóm ágens és a környező rendszerével kapcsolatban elmondhatjuk, hogy egy rendszer olyan mértékig autonóm, amennyire a viselkedését saját tapasztalatai határozzák meg. Azaz az intelligens (és autonóm) ágensek struktúrája

Ágens = (környezeti)architektúra+program.

Az MI-ben felmerülő feladatkörnyezetek sokszínűsége vitathatatlan, de meghatározhatunk néhány olyan tényezőt, amelyek mentén a feladatkörnyezeteket kategorizálhatjuk. Lehet a környezet például

- teljesen megfigyelhető (fully observable) vagy részlegesen megfigyelhető (partially observable). Ha az ágens szenzorai minden pillanatban hozzáférést nyújtanak a környezet teljes állapotához, akkor azt mondjuk, hogy a környezet teljesen megfigyelhető.

- determinisztikus (deterministic) vagy sztochasztikus (stochastic). Amennyiben a környezet következő állapotát jelenlegi állapota és az. ágens által végrehajtott cselekvés teljesen meghatározza, akkor azt mondjuk, hogy a környezet determinisztikus, egyébként sztochasztikus.
- epizódszerű (episodic) vagy sorozatszerű (sequential). Epizódszerű környezetben az ágens tapasztalata elemi „epizódokra" bontható. Minden egyes epizód az ágens észleléseiből és egy cselekvéséből áll. Nagyon fontos, hogy a következő epizód nem függ az előzőben végrehajtott cselekvésektől. Epizódszerű környezetekben az egyes epizódokban az akció kiválasztása csak az aktuális epizódtól függ. Sok osztályozási feladat epizódszerű. Például az összeszerelő soron levő hibás alkatrészeket észlelő ágens minden egyes döntését az aktuális alkatrész alapján hozza, függetlenül a korábbi döntésektől, továbbá az aktuális döntés nem befolyásolja, hogy a következő alkatrész hibás lesz-e. A sorozatszerű környezetekben az aktuális döntés befolyásolhat minden továbbit. A sakk és a taxi vezetése sorozatszerű: a rövid távú akciók mindkét esetben hosszú távú következményekkel járhatnak. Az epizódszerű környezetek sokkal egyszerűbbek a sorozatszerűeknél, hiszen az ágensnek nem kell előre gondolkodnia.
- statikus (static) vagy dinamikus (dynamic). Ha a környezet megváltozhat, amíg az ágens gondolkodik, akkor azt mondjuk, hogy a környezet az ágens számára dinamikus; egyébként statikus. A statikus környezetekkel egyszerű bánni, mivel az ágensnek nem kell állandóan a világot figyelnie, miközben dönt a cselekvés felől, és nem kell az idő múlásával sem törődnie. A dinamikus környezetek állandóan azt kérdezik az ágenstől, hogy mit akar tenni: ha még nem döntötte el, az annak számít, hogy úgy döntött, hogy nem tesz semmit (azaz egyféle visszacsatolással van dolgunk).
- diszkrét (diserete) vagy folytonos (continuous). A diszkrét/folytonos felosztás alkalmazható a környezet állapotára, az időkezelés módjára, az ágens észleléseire, valamim cselekvéseire. Például egy diszkrét állapotú környezet, mint amilyen a sakkjáték, véges számú különálló állapottal rendelkezik. A sakkban szintén diszkrét az akciók és cselekvések halmaza. A taxivezetés

folytonos állapotú és idejű probléma: a sebesség, a taxi és más járművek helye folytonos értékek egy tartományát járja végig a folytonos időben.

- Egyágenses (single agent) vagy többágenses (multiagent). Az egyágenses és több-ágenses környezetek közötti különbségtétel egyszerűnek tűnhet. Például a keresztrejtvényt megfejtő ágens önmagában nyilvánvalóan egyágenses környezetben van, míg egy sakkozó ágens egy kétágensesben.
- versengő (competitive) vagy kooperatív (cooperative)

A felsorolt fogalmakkal már találkozunk a rendszerelmélettel kapcsolatos korábbi fejezetekben. Ebből is láthatjuk, hogy amikor rendszer vagy rendszermodell formalizálást végzünk, bátran használhatjuk a szereplő ágensekre vonatkozó meghatározásokat, mert azok a rendszer elemeinek (ágenseinek) szerepét, szereplését jól értelmezhetően leírják.

Az ágensek a következő, más területekről is ismert szervezeti architektúrákba csoportosulhatnak:

Hierarchia, Holarchia, Koalíció, Team, Kongregáció, Közösség, Federáció, Piac, Mátrix, Összetett (hibrid) struktúra.

2.5 Kognitív térképek

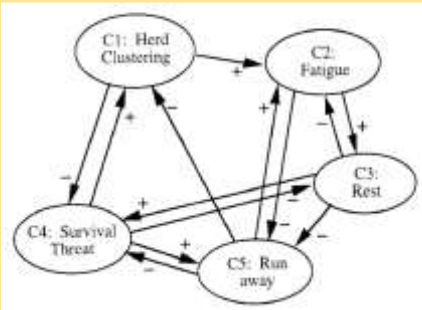
Az előrejelzés matematikai eszközeként fuzzy kognitív térképet használhatunk, melyet az ismert adatok alapján kódolt algoritmus szerkeszthet meg. Ez lényegében egy modellezett szimulációs folyamat.

Az elnevezés az amerikai **Kosko**[51] nevéhez fűződik, aki jelenleg villamosmérnök professzor a Dél-Kaliforniai Egyetemen (University of Southern California, USC, www.usc.edu). Fuzzy kognitív térképéről szóló cikkét 1986-ban publikálta elsőként az IJMMS (International Journal of Man-Machine Studies) 24. számában. Kosko szerteágazó érdeklődése tanulmányaiban is megmutatkozott. BSc fokozatot filozófia és közgazdaságtan területeken szerzett, MSc tanulmányait alkalmazott matematika, PhD-t villamosmérnöki területen folytatta, miközben jogi doktori címet is szerzett. Ennek megfelelően fuzzy logika, neurális hálózatok és zaj témakörben is számos cikket publikált.

[51] Bart Andrew **Kosko** (1960-), amerikai villamosmérnök

Fuzzy kognitív térképe lényegében Robert Axelrod politikai kutató 1976-ban publikált kognitív térképének továbbfejlesztése, kiegészítése élsúlyokkal és számítási mechanizmussal. A fuzzy kognitív térkép egy irányított páros gráf, csúcsokkal és súlyozott élekkel. A gráf csúcspontjai a rendszer egyes tényezőit (concepts, jelölésük *C*) szimbolizálják, az élek pedig az egyes tényezők egymásra gyakorolt hatását jelölik. Az élek és élsúlyok mindig konkrét tényező-párokra vonatkoznak. Az élsúlyok előjele is fontos. Pozitív élsúly ($w_{ij}>0$) arra enged következtetni, hogy a kiinduló tényező növekedésével a cél tényező értéke is növekszik, vagy éppen fordítva: a kiinduló tényező csökkenésével a cél tényező értéke is csökkenni fog. Előfordulhat persze negatív súlyú él is ($w_{ij}<0$), ami azt mutatja, hogy a forrás tényező növekedésekor a cél tényező értéke csökken, vagy a forrás csökkenésével a cél tényező értéke növekedni fog. Harmadik eset, amikor az él étéke nulla ($w_{ij}=0$). Ilyenkor a két tényező között nincs összefüggés.

Fuzzy kognitív térképről lévén szó az egyes tényezők kizárólag fuzzy, vagyis [0, 1] értékeket vehetnek fel, az egyes élsúlyok pedig hasonlóképpen csak [-1, 1] tartománybeli értékek lehetnek. Fontos megjegyezni, hogy a tényezők közötti összefüggés nem feltétlenül azonos mindkét irányban ($w_{ij}\neq w_{ji}$).



FCM általános modellje[52]

$$\mathbf{E} = \begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & C_3 & C_4 & C_5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & 1 & -1 \\ 1 & 0 & -1 & 0 & 1 \\ -1 & 1 & 0 & -1 & 0 \end{bmatrix} \end{array}$$

FCM modellhez tartozó súlymátrix[2]

Az élek értékeit korábban gyűjtött adathalmazzal betaníthatjuk, majd a csomópontok értékeit egy átviteli függvénnyel újraszámítjuk, éspedig a bemeneti élsúlyok, a bemeneti csomópontok értékei alapján.

[52] Julie A. Dickerson, Bart Kosko, "Virtual Worlds as Fuzzy Cognitive Map", *The Massachusetts Institute of Technology, Presence*: Vol. 3. No. 2. 173-189, Spring 1994

Irodalomjegyzék

Tóthné Laufer Edit (2014): Lágy számítási módszereken alapuló kockázatkiértékelés folyamatának és metódusainak optimalizálása, *PhD értekezés, Óbudai Egyetem 2014.*

Johanyák Zs. Csaba (2004): Fuzzy logika, Oktatási segédlet, 2004 [Online]. Available: *http://www.johanyak.hu/files/u1/publi/J_Fuzzy_logika_segedlet.pdf* [Febr 04, 2014]

L. T. Kóczy, D. Tikk (2001): Fuzzy rendszerek, *Kempelen Farkas Tankönyvtár*, [Online]. Available: *http://www.tankonyvtar.hu/hu/tartalom/tkt/fuzzy-rendszerek-fuzzy/adatok.html* [Febr 04, 2014]

Dubois, Didier; Prade, Henri (1983), Ranking fuzzy numbers in the setting of possibility theory, Information Science, Volume 30, Issue 3, 183-224. old.

Kóczy T. László, Tikk Domonkos (2012), Fuzzy rendszerek, Typotex Kiadó, ISBN 963 9132 55, 1http://www.typotex.hu/e_book.htm (megújított kiadás)

Łukasiewicz, Jan (1920), O logice trójwartościowej, *Ruch filozoficzny 5*, 170–171. old.

Klemen, Erich Peter; Mesiar, Ratko; Pap, Endre (2000), Triangular norms, Kluwer Academic Publishers, Dodrecht, ISBN 0792374163

Mamdani, E. H. (1974), Application of fuzzy algorithms for the control of a simple dynamic plant. In *Proc IEEE*, 121-159. old.

Mamdani, E. H., Assilian, S. (1975), An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller, *International Journal of Man-Machine Studies* 7(1): 1-13. old

Pásztorné Varga Katalin, Várterész Magdolna (2003), A matematikai logika alkalmazásszemléletű tárgyalása, Panem Kft., Budapest, ISBN 9635453647

Schweizer, Berthold; Sklar, Abe (1960), Statistical metric spaces, *Pacific Journal of Mathematics*, Volume 10, Number 1., 313-334. old.

Takács Márta (2010), Multilevel Fuzzy Approach to the Risk and Disaster Management, *Acta Polytechnica Hungarica*, Volume 7, Issue Number 4, 2010. , pp.91-102.

Tsuna, Sasaki; Takamasa Akiyama, (1988), Traffic control process of expressway by fuzzy logic, *Fuzzy Sets and Systems*, Volume 26, Issue 2., 165–178. old.

Sanchez, Elie (2006), Fuzzy Logic and the Semantic Web, Elsevier Science, ISBN 80444519481

Urbán János (2006), Matematikai logika, Műszaki Könyvkiadó, Budapest, ISBN 9789631630350

Zadeh, Lofti A. (1965) Fuzzy Sets, *Information Control 8*, 338-353. old.

Robert Fullér (1998): FS IV: The theory of approximate reasoning, Neural Fuzzy Systems Online Lecture Notes: Fuzzy Systems Lecture IV.

Vartrész Magda, Kádek Tamás (2012): Automatikus tételbizonyítás előadások, Debreceni Egyetem, 2012., http://www.inf.unideb.hu/~varteres/ATB_VM_KT.pdf

Pásztorné Varga Katalin, Várterész Magda, Sági Gábor (2012), A matematikai logika alkalmazásszemléletű tárgyalása, Panem Kft., Tankönyvtár, http://www.tankonyvtar.hu/hu/tartalom/tamop425/0046_a_matematikai_logika_alkalmazasszemleletu_targyalasa/ch06.html#id825119

STUART J. RUSSELL, PETER NORVIG: Mesterséges intelligencia modern megközelítésben, PANEM 2000., Mesterséges intelligencia almanach, https://mialmanach.mit.bme.hu/

Pásztorné Varga Katalin, Várterész Magda (2003), *A matematikai logika alkalmazásszemléletű tárgyalása,* Panem Kiadó, , WEB: http://www.tankonyvtar.hu/hu/tartalom/tamop425/0046_a_matematikai_logika_alkalmazasszemleletu_targyalasa/adatok.html, (2015. október)

Ésik Zoltán (2011), *Logika és informatikai alkalmazásai*, egyetemi jegyzet, SZTE, WEB: http://www.inf.u-szeged.hu/sites/default/files/szalap/logika.pdf

Bércesné Novák Ágnes(2012), *Elsőrendű logika,* egyetemi jegyzet, PPTE, , WEB: http://users.itk.ppke.hu/~szako1/elsorendu_logika_20112.pdf

Németh Zoltán, *Logika és informatikai alkalmazásai,* egyetemi jegyzet - gyakorlatok, SZTE, (2008), WEB: http://www.inf.u-szeged.hu/~zlnemeth/logika2008

Fullér Róbert, *Fuzzy Reasoning and Fuzzy Optimization*, Turku Centre for Computer Science, Abo, Finnland, (1998), WEB: http://uni-obuda.hu/users/fuller.robert/sda1.pdf

Pataricza András (2004), Formális módszerek az informatikában, Typotex

További internetes források:
http://aima.cs.berkeley.edu http://foner.www.media.mit.edu/people/foner/Julia/Julia.html http://www.auml.org/ http://www.agentbuilder.com/ http://www.fipa.org/ http://www.agentlink.org/index.php http://multiagent.com/ http://www.agentland.com/

Tartalomjegyzék

Tartalom

Előszó ... 4
1 A matematikai logika ... 6
1.1 A logika helye és szerepe a matematikában ... 6
1.2 Predikátum (nulladrendű) logika ... 7
1.2.1 A kijelentés-logika formális nyelve, szintaxisa ... 11
1.2.2. A predikátumlogika szemantikája ... 12
1.3 Helyes következtetési szabályok, alkalmazásaik ... 17
1.4 Modellelméleti vagy szemantikus következményfogalom ... 20
1.5 Normálformák ... 24
1.5.1 Teljes műveleti rendszer ... 26
1.5.2 A rezolúció alapelve ... 26
1.5.3 Horn formulák és Hilbert axiómák ... 35
1.5.4 Helyesség és teljesség (soundness & completeness) ... 37
1.6 Fák a logikai problémamegoldásban ... 38

... 38
1.7 Halmazelmélet és a logika ... 45
1.7.1 A halmazelmélet, a logika és a Boole algebra ... 48
1.8 Elsőrendű logika ... 49
1.8.1 Az elsőrendű logika szintakszisa, nyelve ... 54
1.8.2 Az elsőrendű logika szemantikája ... 56
1.8.3 Modell, igazságérték, interpretáció az elsőrendű logikában példákon keresztül ... 60
1.8.4 Fontosabb elsőrendű ekvivalens formulák ... 64
1.8.5 Prenex formulák ... 65
1.8.6 Skolem normálforma ... 67
1.8.7 Herbrand kiterjesztés ... 69
1.8.8 Egyesítő helyettesítések ... 71
1.8.9 Elsőrendű modellezés és a rezolúció alkalmazása ... 73
1.8.10 Helyesség és teljesség még egyszer ... 75
1.9 Logikai programozási paradigma ... 76
1.9.1 A Prolog mint logikai gép ... 77
1.10 További logikák ... 79
1.10.1 Modális logika ... 79
1.11 Többértékű logikák ... 85
1.11.1 Lukasiewitz féle logika ... 85

1.11.2 A fuzzy logika és a lágy számítási módszerek 89
2 Formális módszerek 107
2.1 Rendszerelméleti alapfogalmak 107
2.1.1 A rendszer fogalma 107
2.1.2 A rendszermodell készítésének életciklusai 108
Milyen eszköztárral modellezzünk? 113
2.1.3 Modellelmélet, tételbizonyítás, bizonyításelmélet 114
2.1.4 Diszkrét modell- és rendszerállapot tervezés 116
2.2 Petri hálók 117
2.2.1 A Petri háló mint általános állapotleíró módszer 122
2.2.2 Nem-determinisztikus viselkedésből adódó konfliktushelyeztek megoldása 128
2.2.3 Petri hálós alkalmazások az operációs rendszereknél 134
2.3 Temporális logika 138
2.3.1 A Kripke struktúra 142
2.3.2 A PLTL (Propositional Linear Time Temporal Logic) szintaxisa és szemantikája 143
2.3.3 Elágazó idejű temporális logika szemantikája és szintaktikája 145
2.4 Ágens rendszerek 151
2.5 Kognitív térképek 156
Irodalomjegyzék 158

CIP - Каталогизација у публикацији
Библиотека Матице српске, Нови Сад

510.6(075.8)

TAKÁCS, Márta, 1960-
Matematikai logika és formális módszerek [Elektronski izvor] / Takács Márta. - Szabadka : Magyar Tannyelvű Tanítóképző Kar, 2017. - 1 elektronski optički disk (CD-ROM) : tekst, slika ; 12 cm

Sistemski zahtevi: Nisu navedeni. - Nasl. sa naslovnog ekrana. - Tiraž 200. - Bibliografija.

ISBN 978-86-87095-74-8

a) Математичка логика
COBISS.SR-ID 319205383