



BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM  
KÖZLEKEDÉSMÉRNÖKI ÉS JÁRMŰMÉRNÖKI KAR  
KÖZLEKEDÉS- ÉS JÁRMŰIRÁNYÍTÁSI TANSZÉK

# DIPLOMATERV

**Formális modellezés alkalmazásának lehetőségei a  
vasúti biztosítóberendezések területén**

2016

# ADATLAP

## DIPLOMATERV FELADATHOZ

Név:	Farkas Balázs	Neptun kód:	ANNPWV
Témakör (közúti, vasút, légi stb.):	vasút		
Szakdolgozat/diplomaterv címe:	Formális modellezés alkalmazásának lehetőségei a vasúti biztosítóberendezések területén		
A feladat részletezése, vázlata:			
<ul style="list-style-type: none"><li>• Ismertesse a fontosabb biztosítóberendezés alapelveket (táblázatos és nyomvonal-elvű)! Hozzon példákat az elvek megvalósítására!</li><li>• Válasszon a biztosítóberendezések működési elveinek modellezéséhez két eltérő formális módszert (modellezési formalizmust és analízis eszközt)! Mutassa be ezek háttérét, felépítését!</li><li>• Modellezze a vágányutas és a nyomvonal-elvű berendezések működését a két választott modellezési formalizmussal egy megfelelően választott esettanulmányon! Külön térjen ki az elhanyagolásokra és az egyszerűsítésekre!</li><li>• Igazolja a modellek helyességét modellellenőrzéssel!</li><li>• Hasonlítsa össze és értékelje a fenti modellezési folyamat tanulságait a mérnöki munkában való felhasználás szempontjából (előnyök/hátrányok)!</li></ul>			
Egyetemi (belső) konzulens neve:	Dr. Bartha Tamás		
Ipari (külső) témavezető neve, munkahelye:	Lukács Gábor, MKTM Kft.		

# Tartalomjegyzék

1.	Bevezetés .....	3
2.	Elméleti háttér.....	4
2.1.	Biztosítóberendezések .....	4
2.1.1.	Bevezetés.....	4
2.1.2.	Vasúti járműmozgások biztosítása .....	5
2.1.3.	Biztosítóberendezési szerkesztési elvek.....	7
2.1.4.	Vágányúti logika leképezése a gyakorlatban.....	9
2.2.	Formális módszerek .....	20
2.2.1.	Verifikáció.....	20
2.2.2.	Modellellenőrzés .....	21
2.2.3.	Átmeneti rendszerek, időzített automaták .....	23
2.2.4.	Petri hálók .....	24
2.2.5.	Temporális logikák (CTL).....	27
2.2.6.	Formális módszerek alkalmazása a vasúttechnikában.....	28
2.2.7.	Modell/analízis eszközök.....	29
3.	Modellezés .....	36
3.1.	Esettanulmány leírása .....	36
3.2.	Modellezési döntések: egyszerűsítések, elhanyagolások .....	37
3.3.	Modellek bemutatása .....	40
3.3.1.	Nyomvonalas elv modellezése Petri hálóval a PetriDotNet eszközben .....	41
3.3.2.	Vágányutas elv modellezése Petri hálóval a PetriDotNet eszközben .....	49
3.3.3.	Nyomvonalas elv modellezése automatákkal az UPPAAL eszközben .....	54
3.3.4.	Vágányutas elv modellezése automatákkal az UPPAAL eszközben .....	69
4.	Modellellenőrzés.....	77
4.1.	A modellellenőrzés megvalósítása .....	78
5.	Tanulságok, tapasztalatok .....	86
5.1.	A szerkesztési elvek összehasonlítása .....	87
5.2.	A formalizmusok, eszközök összehasonlítása, értékelése .....	88
5.3.	Fejlesztési lehetőségek.....	91
6.	Összefoglalás .....	94
	Felhasznált irodalom .....	95
	Ábrajegyzék .....	100
	Táblázatjegyzék .....	103

# 1. Bevezetés

Napjainkban a vasúti (biztonságkritikus) rendszerek (szoftver és hardver) fejlesztése számítógéppel támogatottan történik. A fejlesztés során mára a modellezés (nem csak a formális) általánosan bevettnek tekinthető. Egy adott mérnöki területhez közel álló modellek készítése a fejlesztés során ajánlott, ugyanakkor korszerűnek is tekinthető. A modellezés segítségével bizonyos hibák a fejlesztés korai fázisában kizárhatók, ezáltal a javítási költségek megfelelően alacsonyan tarthatók. A formális vagy félfórmális modellek szintakszisa és szemantikája jól meghatározott, egyértelmű és teljes, azaz egyik sem rendelkezik félreérthető vagy nem definiált elemekkel. Ennek megfelelően a modellek megalkotása és a modellek illetve formális specifikációk révén történő együttműködés és információcsere a fejlesztési folyamat résztvevői között nem jelent újabb bizonytalanságot vagy félreértést, ellentétben a szöveges és ad hoc jelölésrendszereken alapuló elavult, de egyelőre elterjedt mérnöki gyakorlattal. A végrehajtható (szimulálható és futtatható működésű) modellek már készítésük korai fázisában tesztelhetőek. Megfelelő modellező eszköz esetén lehetőség van modellellenőrzésre is, mely formális modellek esetén teljes körű, azaz kimerítő ellenőrzést tesz lehetővé. Ha lehetőség van automatikus kódgenerálásra és minősített kódgenerátor állítja elő a kódot, akkor a modellre vonatkozó helyességbizonyítás (az ellenőrzött tulajdonságok terjedelmében) a kódra vonatkozó helyességbizonyításként is megállja a helyét.

Jelen dolgozat célja megvizsgálni a formális modellezés és modellellenőrzés vasúti biztosítóberendezésekben való alkalmazhatóságát. Ennek érdekében két eltérő formalizmus segítségével került egy adott vasútbiztosítási feladat két eltérő szemléletmódú megközelítése vizsgálatra. Az állomási biztosítóberendezések két elterjedten használt konstrukciós alapelvét (vágányutas és nyomvonalas elvű) azonos helyszínrajzra képeztem le. A modellezést Petri hálókkal a PetriDotNet, és automatákkal az UPPAAL eszközökben hajtottam végre. A biztosítóberendezésekre vonatkozó informális követelmények formalizálásával elvégeztem a modellellenőrzést a modelleken. Az elvégzett munka alapján összehasonlítottam és értékeltem a két eszközt a vasúti biztosítóberendezésekkel kapcsolatos felhasználási lehetőségük alapján.

## 2. Elméleti háttér

Az elméleti háttér ismertetése két részből áll: a szerkesztési elvek bemutatása példákon keresztül, valamint az alkalmazott modellezési, modellellenőrzési ismeretek összefoglalása. Először ismertetésre kerül modellezett rendszer a megfelelő tulajdonságaival, működésével. Ezután a rajta elvégzett folyamatok, és azok elméleti alapjainak bemutatása történik meg.

### 2.1. Biztosítóberendezések

A vasúti járművek közlekedésének biztonságát alapvetően a vasúti jelző- és biztosítóberendezések szavatolják. A biztonság elérését különböző időszakokból származó berendezések más-más elven oldották meg. Jelen alfejezet célja bemutatni a berendezések kialakulását, működését, fajtáit, és a modellezés alapját képező szerkesztési elveket, illetve a köztük lévő különbségeket. A kétféle elv megvalósulására jelen alfejezet példákat is hoz.

#### 2.1.1. Bevezetés

A vasúti közlekedés megjelenése idején a járművek biztonságos közlekedését csupán szervezési úton oldották meg. A forgalom növekedésével viszont viszonylag hamar felmerült az igény olyan berendezésekre, melyek működésükkel automatikusan kizárják az emberi tévedések lehetőségét. [1] Ezt a funkciót eleinte a vágányúti elemek között különböző mechanikus elemek kényszerkapcsolatával oldották meg. Például a vonat továbbközlekedésére engedélyt a mozdonyvezetőnek egy jelző csak akkor adhatott, ha a vonat által igénybe venni kívánt váltók megfelelően álltak. A meghibásodások ellen a mechanikus elemek megfelelő méretezésével és folyamatos karbantartással védekeztek. A biztonság mellett elvárás volt az is, hogy a berendezések ne akadályozzák a forgalmat, sőt lehetőleg gyorsítsák azt. Megjelent az igény az elemek (elsősorban váltók, jelzők) távállítására, melyet rudak, vezetékek segítségével oldottak meg az állítóközpontokból. A különböző vonatérzékelő elemek feltalálásával és alkalmazásával a vonatok közlekedését is be lehetett vonni a függőségek kialakításába.

A következő jelentős lépést a jelfogók vasúti biztosítóberendezésekben való alkalmazása jelentette. Alkalmazásukkal egy jóval nagyobb fokú automatizáltságot lehetett elérni. Ettől fogva a jelfogós berendezések kezelése csupán a parancsok kiadását jelentette, végrehajtásuk további beavatkozást nem igényelt. A parancsokhoz tartozó függőségek elvégzését, illetve a feltételek teljesülése esetén az objektumok állítását a berendezés önműködően elvégezte. A különböző utasítások kiadásához viszont nem szükséges a kezelőnek a működtetett objektumok közvetlen közelében lennie. Egy irányító egy központba összevontan több állomás vezérlését is el tudja látni. Így az állomásokon a személyzet elhagyása válik lehetővé. Ez a folyamat kettős előnnyel bír, csökkenthető az emberi tévedésből eredő hibázás lehetősége (automatikus vágányút-beállítás a váltókezelők ténykedése helyett), illetve kevesebb munkavállaló szükséges. Egy következő lépés innen a kezelőszemélyzet teljes elhagyása, és a berendezés automatikus működtetése a vonatok helyzetétől függően. A meghibásodások ellen a berendezések vagy a jelfogók biztonsági (első osztályú) kialakításával, vagy másod osztályú kivitel esetén az áramkörök megfelelő, valamivel bonyolultabb szerkesztésével védekeztek.

Az elektronikus elemek megjelenése a biztosítóberendezésekben kibővítette az elérhető funkciók körét (pl. a megengedettnél alacsonyabb sebességi fogalom kivezérlési lehetősége egy főjelzőre). Az így adódó előnyök mellett az elektronikus biztosítóberendezések legnagyobb hátránya az ebből fakadó összetettségük. Míg korábban a mechanikus függőségek, áramköri kapcsolások átláthatók, megfelelő hozzáértéssel könnyen olvashatók voltak, addig egy-egy számítógép működése kívülről egyálta-

lán nem „látható”. A különböző beépített szisztematikus (jellemzően „szoftver”) hibák ellen egyik típus sem képes önmagában védekezni, azonban a mechanikus és jelfogós berendezésekben átláthatóságuk, egyszerűbb felépítésük okán ezek a hibák egyszerűbben feltárhatók. Míg a véletlenszerű (hardver) meghibásodások ellen a korábbi berendezések kialakításukban védekeztek, addig az elektronikus berendezésekben ezeket külön kell felfedni, és kezelni. Látható tehát, hogy az elektronikus berendezések megjelenésükkel új kihívások elé állították a fejlesztőket. A különböző hibák ellen elsősorban diverz felépítéssel, több csatornán eltérő hardver és/vagy szoftver felépítéssel, illetve szabványokban meghatározott fejlesztési folyamatokkal lehet védekezni.

Az előzőek értelmében a vasúti biztosítóberendezések kialakításuk szerint három fő csoportba sorolhatók: mechanikus, jelfogós és elektronikus. Az egyes csoportok között nincs éles határ, előfordulnak kevert felépítésű biztosítóberendezések is. Emellett a vasúti automatikák alkalmazására többféle feladat ellátására lehet szükség, ezek a következők szerint csoportosíthatók:

- Állomási berendezések
- Vonali berendezések
- Vonatbefolyásoló berendezések
- Központi forgalomirányító berendezések
- Rendező-pályaudvari berendezések

A biztosítóberendezések klasszikus megvalósulásai az állomási berendezések, melyek a váltók és jelzők közötti függőségeket biztosítják. Az állomások ugyanakkor a legfontosabb forgalomszervező helyek. Itt van lehetőség az ellenkező irányba közlekedő vonatok találkozását lebonyolítani, a vonatok menetirányát megfordítani, de ugyanakkor a szerelvények átrendezései, különböző tolatómozgások is az állomásokon történnek. A vonali berendezések közé tartoznak elsősorban az utolérés és ellenmenet kizáró (köztük a térközi), illetve az útátjáró fedező berendezések. Ebben a csoportba sorolhatók továbbá a nyíltvonali fedezőberendezések és a közlekedő szerelvények állapotát figyelő (pl.: hőnfutásjelző, rakományfigyelő, stb.) szerkezetek. A telepített biztosítóberendezések és a közlekedő vonatok közötti gépi függőséget a vonatbefolyásoló berendezések valósítják meg. A helyi biztosítóberendezések forgalomirányító központokba való szervezése, az állomások távolról való kezelésének lehetősége a központi forgalomirányító berendezések révén valósul meg. A vasúti automatikák egy külön csoportját alkotják a rendező-pályaudvari, gurítódombi berendezések. E berendezések feladata a gurítódombokon a váltók (minél gyorsabb) állítása, és a guruló járművek sebességszabályozása. A továbbiakban csak a dolgozat tárgyát is képező állomási berendezések kerülnek tárgyalásra.

### **2.1.2. Vasúti járműmozgások biztosítása**

A vasúti járművek közlekedése során többféle veszélyeztetés adódhat (pl. váltó átállítása a jármű alatt, más vasúti vagy közúti járművek kerülhetnek az útjába). A veszélyeztetések elkerülése érdekében a vasúti szerelvények csak a megfelelő függőségeket létrehozó vágányutakon közlekedhetnek. [2] [3] [4] A vágányutak egy kezdő (fő- vagy tolatás-) jelzőtől a (következő) céljelzőig, vagy az utána következő egy meghatározott vágányszakaszig terjedhetnek. A főjelzők a vonatmenetek, a tolatásjelzők a tolatómenetek szabályozására szolgálnak. A vágányútban érdekelt elemek lehetnek közvetlenül igénybe vett (érintett), vagy közvetetten igénybe vett (védő-) váltók, vágányszakaszok, jelzők, továbbá egyéb oldalvédelmi elemek, sorompók, és a céljelző mögött lefoglalt megcsúszási szakaszok.

A vágányutak elrendelése, beállítása történhet manuálisan, vagy automatikusan. A manuális beállítás lehet elemenkénti kezelés (váltók egyesével történő állítása), vagy vágányutas elrendelés. Az előbbi jellemzően a kézi vagy vonóvezetékes váltóállítás, de néhány kisállomási jelfogós berendezésben is egyénileg kell a váltókat állítani. Jellemzően azonban a jelfogós és elektronikus berendezések a vágányút manuális kijelölése után a váltókat a kiválasztott vágányútnak megfelelő állásba beállítják. Az automatikus beállítás emberi beavatkozás nélkül a közlekedő vonatok hatására történik. A modellekben a vágányút beállítási parancs nem kerül szimulálásra, csupán a kezdő- és célpontok meghatározásának kimenete kerül felhasználásra.

A vágányút és a startjelző között jelzőfüggés van, mely szerint a jelző csak akkor állítható továbbhaladást engedélyező állásba, ha a váltók helyes állásban állnak és le vannak zárva. A váltó lezárása jelenti egyrészt a csúcscsín megfelelő rögzítését a tősinhez, másrészt az állíthatóság megakadályozását. A jelzők csak az érintett váltók állásának és konstrukciójának megfelelő jelzési képet mutathatnak. Az oldaldélmi elemek (védőváltók, kisiklasztó saruk, vágányzáró sorompók és jelzők) biztosítják, hogy egy lezárt vágányútba oldalirányból (a menetben érintett váltók másik szárán) más járművek ne kerülhessenek. A céljelző mögött lefoglalt megcsúszási szakaszok arra az esetre jelentenek megoldást, ha a jelzőnél a vonatok valamilyen okból kifolyólag nem tudnának megállni, és veszélyeztetnék az ott lévő járműveket. A sorompók csukott állapota és ezáltal a keresztező közúti forgalom tiltása szükséges a jelző szabad állásához. A vágányutak beállítása során jellemző, hogy minden egyes működés a megelőző(ke)t ellenőrzi, lezárja, és engedélyt ad egy következő ténykedésre. Ez a jellegű működés a modellekben is megfigyelhető, pl. az UPPAAL modell egy-egy szinkronizáció során előbb ellenőrzi a változókat, majd a megfelelő értékre beállít egyet. Az újonnan beállított változó értéke pedig a következők során kerül ellenőrzésre (ld. 3.3.3. fejezet). A vágányutat az elemek lezárása után rögzíteni kell egészen a felhasználásig. A beállított vágányút, az oldaldélmi szakaszok és a jelző mögötti megcsúszási szakaszok foglaltságát folyamatosan ellenőrizni kell vonatvágányutak esetében. Tolatévágányutak esetén (mint a modellekben) ez az oldaldélmet kivéve nem feltétel, mivel lehetséges, hogy a tolatás éppen a szakaszon álló járművek felvételére, rendezésére irányul. A startjelző csak addig lehet szabad állapotban, amíg minden feltétel teljesül.

Az egymást veszélyeztető menetek tiltását menetkizárásokkal biztosítják. Egyszerű menetkizárás valósítható meg a jelzőfüggés révén. Erre példa, ha egy váltó két vágányút számára két különböző állásban van igénybe véve. Ekkor a később beállítandó menet jelzője nem állítható szabadra, mert nem lehet a vágányútban érintett összes váltót a kívánt állásba állítani. Különleges kizárásokat alkalmaznak a berendezésekben a jelzőfüggéssel ki nem zárható menetekre. Például azonos vágányra történő egyidejű bejárás az állomás két végén, vagy azonos vágányúti elemek ellentétes irányban (be-/kijárat) való érintése. A modellezés során csak egyszerű menetkizárások történnek. A vágányút beállításának folyamata az előbbiek alapján:

1. Vágányút kijelölése
2. Vágányút beállítása
3. Vágányút lezárása
4. Vágányút rögzítése
5. A vágányút szabad voltának ellenőrzése
6. A startjelző feloldása
7. A startjelző szabadra állítása

A jelzők „Megállj!” állásba való visszaállítása a járművek elhaladása után történhet. Önműködő főjelzőt rendszerint az elhaladó járművek első, míg a tolatásjelzőt jellemzően az utolsó tengelye állítja továbbhaladást tiltó állásba (a modellekben az utóbbi valósul meg). A vágányúti elemek rögzítését a jelző „Megállj!” állásba állása után a vonat vagy tolatómenet teljes leközeledéséig fenn kell tartani. Az érintett elemek feloldhatók, ha a menetet fedező jelző szabad állás után ismét továbbhaladást tiltó állásba került, és a vágányútban érintett foglaltságérzékelési szakaszok (a célvágányt kivéve) foglalttá váltak, majd felszabadultak. A vágányút teljes feloldása történhet egyben, mikor a menet az utolsó érintett váltót elhagyta, vagy elemenként. Utóbbi esetben az elemek feloldásához teljesülnie kell:

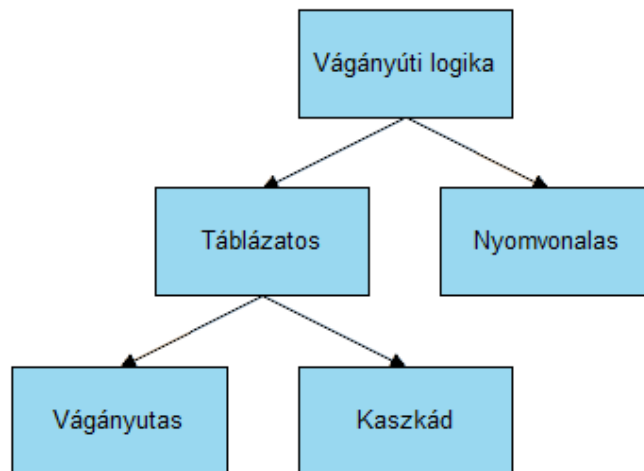
- Minden, az elem előtt érintett szakasz felszabadult/feloldódott.
- Az érintett szakasz foglalt lett, majd szabaddá vált.
- A következő szakasz foglalttá vált.

A céljelző mögötti szakaszok kezelés hatására vagy időzítés leteltére oldódnak. Ha a vonat a beállított vágányutat valamilyen okból nem használja fel, neheztített kényszerkezeléssel a berendezés kezelője is feloldhatja. A vágányút feloldásának folyamata az előbbiek alapján:

1. A startjelző továbbhaladást tiltó állásba állítása
2. A startjelző lezárása
3. Vágányút feloldása
4. (Váltók alaphelyzetbe állítása)

### 2.1.3. Biztosítóberendezési szerkesztési elvek

A biztosítóberendezések konstrukciós felépítésük, a vágányúti logika leképezése szerint két fő csoportba sorolhatók. [3] [5] [6] Ezek a táblázatos (vagy koordináta) és a nyomvonalas (vagy geografikus) elvek, ezen belül előbbi tovább bontható vágányutas illetve kaszkád elvekre. (1. ábra)



1. ábra A vágányúti logika leképezése (forrás: [5])

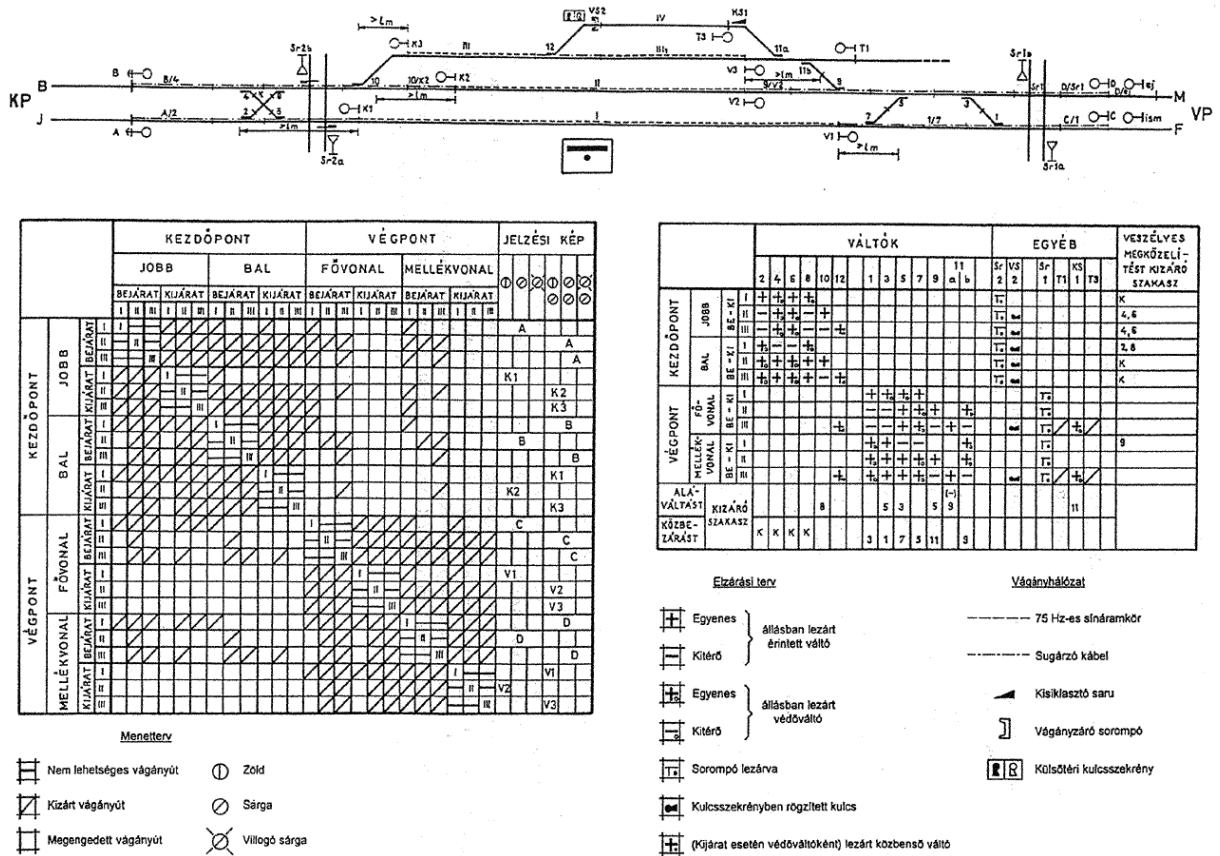
#### **Kaszkád elv [7]**

A kaszkád logika elsősorban a brit nyelvterületen, mechanikus berendezésekben terjedt el. A kizárások lényege, hogy egy vágányúthoz tartozó váltók csak meghatározott sorrendben, egymás után állíthatók. Ez a megoldás leegyszerűsíti a vágányút-ellenőrzést, ugyanakkor bonyolultabb mechanikai kizárásokat követel meg. A kaszkád elv a továbbiakban nem képezi a vizsgálatok tárgyát.



### Vágányutas elv [3] [5]

A koordináta elv legjellemzőbb leképezései a vágányutas elvű berendezések. A ki- és lezárások alapját a vágányutak adják. A függőségek megvalósítása két táblázat, a meneterv és az elzárási terv alapján történik. (2. ábra) A meneterv oszlopai és sorai is egy adott helyszínrajzú állomás beállítható vágányútjait jelenítik meg. Az egyes vágányutak metszetében lévő cellák kitöltése a kizárásokat mutatja. A táblázat a főátlójára szimmetrikus. Az egyszerű kizárásokat vízszintes vonal, a különleges kizárásokat rézsútós vonal jeleníti meg. Ha két vágányút nem zárja ki egymást, a hozzájuk tartozó cella üres.



2. ábra Egy négyvágányos állomás menetterve és elzárási terve (forrás: [8])

Az elzárási terv esetén a képzeletbeli koordináta-rendszer egyik tengelyét a vágányutak, másik tengelyét a különböző objektumok képezik. Ez a táblázat adja meg a soraihoz tartozó vágányutakban az oszlopokhoz tartozó elemek elvárt állását. A váltó egyenes állását „+” jel, kitérő állását „-” jel szimbolizálja. A cellák jobb alsó sarkában lévő kör a váltó védőállásra utal. A váltók beállítása után a vágányút lezárható, rögzíthető és a jelző szabadra állítható. A vágányút feloldódása jellemzően egy részben történik, miután a vonat a váltóközvetet elhagyta.

### Nyomvonalas elv [3] [5]

A geografikus elv alkalmazásában a vágányúti elemek önálló objektumokként jelennek meg, melyek a topológiának megfelelően kapcsolódnak össze egymással. (ld. később 7. ábra és 25. ábra) A váltók elvárt állása nincs előre tárolva, azt a berendezés minden vágányút számára beállításkor határozza meg. A vágányút start- és célpontjának kijelölése után az elemek között útvonalkeresés indul mindkét irányból. A keresés csúccsal szemben talált váltóknál szétágazik, gyök felől találtknál lenyesi a másik ágat, ezzel kizárva az onnan érkező esetleges keresést. A folyamat jellemzően az egyik irányból

(startból vagy célból) indul, és a másikba elérve visszafordul. Így minden vágányútban releváns elem egyértelműen, kétszer lesz a keresés során érintve. A vágányútban érintett váltók állása is egyértelműen meghatározott, a gyök felől érkező keresés alapján. Az oldalvédelmet nyújtó elemek szintén a keresés során jelölődnek ki. A kiválasztott váltók a kijelölt állásukkal ellentétes száron keresnek oldalvédelmet. A vágányút felépítéséhez szükséges, hogy minden váltóhoz tartozzon oldalvédelmet nyújtó elem. A vágányúti kizárások megvalósítása is elemenként történik. Két vágányút kizárja egymást, ha egy vágányúti elemet közösen, vagy egy oldalvédelmi elemet eltérő állásban használnának fel.

A kiválasztott útvonalon fekvő és az oldalvédelmi elemek lefoglalódnak a vágányút számára. A kiválasztott elemek aktuális állása összehasonlításra kerül a kijelölttel, és ha szükséges, a mozgó elemek állítási parancsot kapnak. A (szükséges állítást követően a) megfelelően álló elemek lezáródnak. Minden elem lezáródását követően a vágányút rögzítésre kerül és startjelző szabad állásba állítható. A vágányút feloldása a közlekedő vonat hatására elemenként történik, az oldalvédelmi elemek a védett elemmel együtt oldódnak fel.

#### **2.1.4. Vágányúti logika leképezése a gyakorlatban**

Jelen alfejezet célja példákat hozni a hazai jelentősebb biztosítóberendezés típusokban a kétféle konstrukciós elv megjelenésére. Nem célja ugyanakkor a berendezések működésének teljes ismertetése, csupán a diplomaterv szempontjából releváns részek kiemelése. A berendezéstípusok leírása során történnek utalások az egyes funkciók megjelenésére a modellekben.

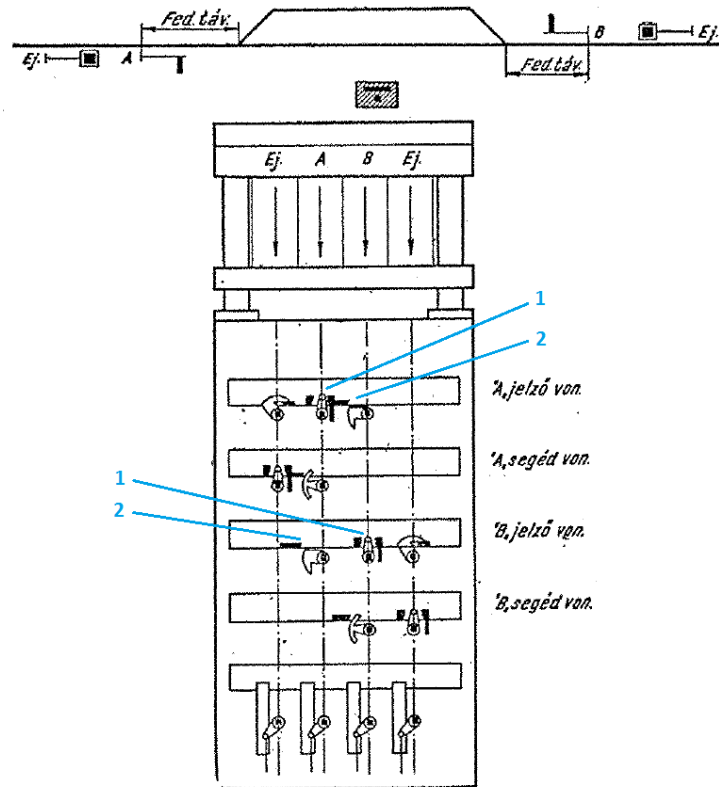
A mechanikus berendezések mind koordináta elv szerint szerkesztettek. Ezek a szerkezetek jellemzően többközpontosak. A meneteket a forgalmi irodában dolgozó forgalmi szolgálattevő rendeli el rendelkezőkészüléke segítségével. Az elrendelt vágányutak beállítását a váltókörzetek közelében elhelyezett állítóközpontokból a váltókezelő végzi el. Külön állítóközpont alkalmazása azért szükséges, hogy beállításakor a vágányút szabad voltáról a helyszínen, rátekintéssel meg lehessen győződni, a tolatási mozgásokat innen lehessen irányítani, illetve hozzá a váltókat állítani (amit szintén csak azok szabad állapotában lehet elvégezni). A harmadik ok a vonóvezetékes váltók korlátozott állítási távolsága, illetve helyszíni állítású váltók esetén a rövid gyaloglási távolság.

Mechanikus berendezéseknél a menettervet a rendelkezőkészülékek valósítják meg. Mivel ezek rendelkeznek a teljes állomás felett, így az összes beállítható vágányút közötti kizárás bennük oldható meg. Az elzárási tervet ezzel szemben az állítóközpontok berendezései valósítják meg. Ezekkel történik a vágányutak beállítása, az egyes vágányutakhoz tartozó függőségeket e berendezésekben kell leképezni. A következőkben, időrendi sorrendben a mechanikus, jelfogós elektronikus berendezések ismertetése történik meg. A különböző berendezéstípusokat jelölő alcímek mögött zárójelben a berendezések hazai megvalósulásának nevei szerepelnek, az alfejezet végén található 1. táblázatban való azonosíthatóságuk végett.

#### ***Jelzőberendezések [1] [9]***

A biztosítóberendezések által nyújtott biztonsági szintet el nem érő mechanikus jelzőberendezéseknél is megjelenhetnek a vágányutas elv szerinti függőségek. A klasszikus értelemben vett jelzőberendezéseknél nincs semmilyen függés a váltók és jelzők között. Egyszerű elrendezésű középállomások (egy-egy csatlakozó nyílt vonal az állomás két végén) berendezései a menettervi függőségeket az alábbiak szerint valósítják meg. Ilyen esetben a menettervi függés azt jelenti, hogy a két bejáratú jelző közül egy időben mindig csak egy állítható szabad állásba. Forgalmi irodából kezelt jelzők esetén a kölcsönös kizárást a jelzőemelyűkhöz tartozó tengelyek által mozgatott vonalzókat valósítják meg.

(3. ábra) Mindkét bejárati jelző emeltyűjéhez tartozik egy tengely, ami az emeltyű átállításával jobbra elfordul, és a rá szerelt 1 jelű kilincsel egy acél vonalzót eltol. A vonalzó megfelelő helyén felszegecselt 2 jelű tuskó pedig a másik jelző átállítását akadályozza annak tengelyén és a rá szerelt kilincsen keresztül. Állítóközpontból kezelt bejárati jelzők állítására a forgalmi szolgálattelvő a jelzőblokk feloldásával ad engedélyt. A forgalmi irodában a blokk szekrényben megvalósuló kölcsönös kizárás biztosítja, hogy mindig csak egy bejárati jelzőhöz tartozó jelzőblokk lehessen feloldva.



3. ábra Menettervi kizárások megvalósulása jelzőberendezésben (forrás: [9])

### **Kulcsazonosító és kulcsrögítő berendezések (kulcsos berendezések) [1] [9]**

A jelzőberendezéseket gyakran kiegészítik váltózárkulcs-azonosító berendezésekkel. Ezek feladata csupán az elzárási terv megvalósítása, a hozzájuk tartozó irodai elrendelő készülék nem tartalmaz menettervi függőségeket (azt továbbra is a jelzőberendezés biztosítja). A nem biztosított állomásokon a váltók jellemzően helyszíni állításúak, lezárásuk a váltóra szerelt, és a csúcssínek állását ellenőrző váltózárakkal történik. Megfelelően álló váltó esetén a váltózárak kallantyúja elfordítható, és az állásnak megfelelő kulcs kivehető, majd az állítóközponti berendezésbe helyezhető. A berendezésen a beállítandó vágányúthoz tartozó kallantú vagy kapcsológomb csak akkor állítható a vágányútnak megfelelő irányba, ha a kívánt menethez tartozó összes (érintett és védő) váltó megfelelő kulcsa a berendezésbe van helyezve. A kallantú vagy kapcsológomb tengelye által mozgatott vonalzó egyszerre ellenőrzi a megfelelő kulcsok meglétét, és zárja el azokat. A vágányút lezárása során mind a forgalmi irodai, mind az állítóközponti berendezésen ugyanannak a vágányútnak kell kijelölve lennie. A vágányút lezárását a forgalmi szolgálattelvő saját irodai berendezésével végzi. A jelzőberendezés csak az egyidejű bejárásokat tiltja meg, az állomás két végén történő egyidejű, nem megengedhető be- és kijáratok letiltása a forgalmi szolgálattelvő felelőssége.

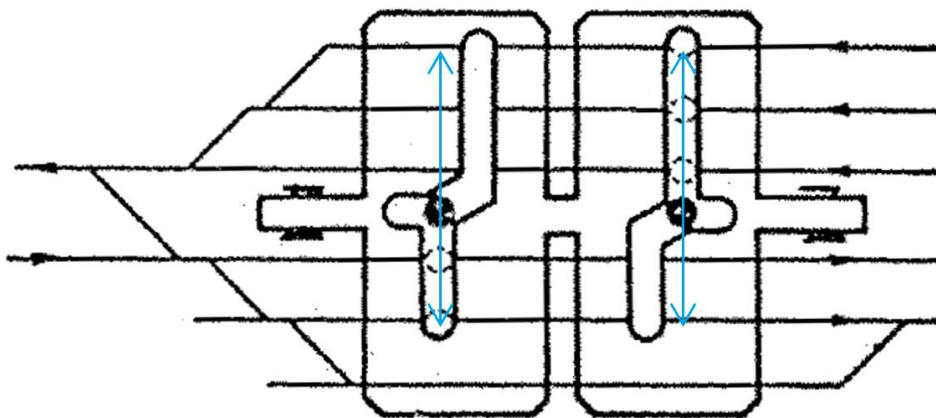
A váltók és jelzők közötti függőségeket már tartalmazó, tehát biztosítóberendezésnek minősülő kulcsrögítő berendezések felépítése a váltózárkulcs-azonosítóval kiegészített jelzőberendezésekével

rokon. A konstrukciós elvek megvalósítása az állítóközponti készülékekben is jellemzően megegyezik, azonban ezek a berendezések már rendelkeznek (többnyire közös) kijárat jelzőkkel és/vagy feloldóberendezéssel. Ebből adódó különbség, hogy e berendezések rendelkezőkészüléke már megkülönböztet be- és kijárat vágányutakat, így biztosítani tudják a menettervi függőségeket. A kisebb állomásokon alkalmazott kallantyús készülék vonalzó eltolásával biztosítja a kizárásokat. Összetettebb (pl. elágazó) állomások esetén a rendelkezőkészülék felépítése a következő részben ismertető vonóvezetékes berendezésekével rokon.

Egyközpontos, helyszíni állítású váltókkal rendelkező állomás berendezése a váltók állásáról a berendezésbe helyezett kulcsokon vagy a váltók állását ellenőrző vonóvezetékes reteszek segítségével szerez információt. Ezekben a berendezésekben a váltók lezárását követően a beállítandó vágányúthoz tartozó kallantyú elfordítása után lehet a jelzőket kezelni. Egyközpontos berendezésekben tehát a vágányúti kallantyúk egyszerre végeznek menettervi (ellenséges menetekhez tartozó vonalzó eltolásának megakadályozása) és elzárási tervi (váltóállások ellenőrzése és rögzítése) függőségeket.

#### **Vonóvezetékes mechanikus berendezések (Siemens-Halske) [1] [9]**

Vonóvezetékes állítású berendezéseknél, de nagyobb állomások kulcsazonosítós illetve kulcsrögzítős berendezéseinél is tologombos rendelkezőkészüléket alkalmaznak. A Rank készüléknek is nevezett berendezésben az állomás torzított helyszínrajzán tologommbal lehet kiválasztani a vágányúthoz tartozó állomási vágányt, és a vele egy vonalban lévő kallantyúval a közlekedés irányát (be/kijárat). Minden csatlakozó vonali vágányhoz (és gyakran irányhoz) tartozik egy tologombkészlet (tologomb + kallantyú). Mechanikus függőségek okán először a tologombot kell a kijelölt vágány fölé állítani, és utána lehet a kallantyút elfordítani. A kallantyú elfordításával a kiválasztott vágányúthoz tartozó vonalzó is elmozdul, és egy függőségi tengely elfordításával elzárja az ellenséges vágányutakhoz tartozó vonalzókat, tehát azok kallantyúját nem lehet (egy bizonyos irányba vagy egyáltalán) elfordítani. Ha két vágányút menetiránytól függetlenül veszélyes egymásra, akkor lemezidom függést lehet alkalmazni, mely már a kizárandó menetek tologombját megfogja. (4. ábra)



4. ábra Lemezidom függés a rendelkezőkészülékben (forrás: [9])

Többközpontos biztosítóberendezéseknél a központok közötti függőségeket blokkelemek segítségével hozzák létre. Alapvető működésük az, hogy a kezelő berendezésében valamilyen zárást, míg az ellenoldali berendezésben oldást hozzanak létre. A blokkelemek kezeléskor átváltanak, és a függőségeket rúdjaik segítségével hozzák létre. Vonóvezetékes berendezéseknél is az őrhelyi készülék valósítja meg az elzárási tervet. Az állítóközpontban a vágányúti kallantyúval a vágányúti vonalzó csak akkor mozdítható el, ha a váltó- és reteszemeltűk a vágányútnak megfelelő állásban állnak. A váltókezelő a

vágányút blokk lezárásával zárja el a vágányúti vonalzó, majd a forgalmi szolgálattevő feloldja a jelzőblokkot, és a jelzők kezelhetők. A vágányút feloldására a jelző „Megállj!” állásban visszaállítása és a váltókörzet után elhelyezett oldószakasz foglalttá válása majd felszabadulása után van lehetőség.

### ***Elektromechanikus berendezések (VES) [10]***

Elektromechanikus berendezések esetén a váltók villamos állításúak, az alkalmazott jelzők fényjelzők vagy motoros állítású alakjelzők, a függőségek kialakítása viszont továbbra is többnyire mechanikus. A berendezés általában többközpontos. Az irodai rendelkezőkészülék, az állítóközponti készülékek és a külsőtéri objektumok (váltók, jelzők, szigeteltsínek) közötti kapcsolat kábelhálózat útján valósul meg. Minden kezelés kapcsológombok segítségével történik.

Az irodai rendelkezőkészüléken a vágányutat egy háromállású kapcsológomb elfordításával lehet kijelölni (pl. egy adott irányból bejárat a II. vagy a III. vágányra). A kapcsológomb elfektetésekor tengelye egy vonalzó mozdit el. Erre csak akkor van lehetőség, ha nincs tiltott egyidejű menet beállítva. Emellett az eltolt vonalzó megakadályozza, hogy ezután tiltott vágányúthoz tartozó másik kapcsológombot fektessenek el. Ez a két függőség valósítja meg a szükséges menettervi kizárásokat. A kapcsológomb által működtetett villamos érintkezők révén valósul meg a vágányút (irány, és vágányszám) kijelölése az állítóközpont felé. A kapcsológombokhoz egy-egy reteszmágnes is tartozik vágányutanként. Ezek feladata, hogy a jelzőállítás kezdeményezése után a kapcsológombot elfektetett helyzetében (és ezáltal a vágányút beállítás tényét az irodai rendelkezőkészülékben) egészen a vágányút feloldásáig rögzítsék.

Az alapvető vágányúti kezeléseket az állítóközponti készülékekben is kapcsológombokkal végzik. Háromféle kapcsológomb kerül alkalmazásra, melyek a következők: váltó-, vágányúti és jelzőkapcsoló gombok. A váltók állítására a fekete színű váltókapcsoló gombok szolgálnak. Függőleges helyzetének a váltó egyenes, 90°-kal elfektetett helyzetének a váltó kitérő állása felel meg. A váltók állítása egyenként, a megfelelő kapcsológombok elfordításával történik. A vágányutak beállítására vonali vágányok szerinti csoportosításban két-két vágányút számára egy-egy háromállású (zöld) vágányúti kapcsológombot, valamint vonali vágányonként egy-egy háromállású (vörös) jelzőkapcsológombot építenek be. A vágányút kijelölése során meghúzza a vágányszámjelölő és irányjelző mágnes. A váltók állítását követően a megfelelő irányba (45°-kal) elfektethető a vágányúti kapcsológomb. A hozzá tartozó tengely egy vágányúti vonalzó mozgat, ami ellenőrzi a váltó(kapcsoló)k helyes állását, és egyúttal rögzíti azokat (elzárási terv megvalósítása). Ezután fordítható el a jelzőkapcsoló gomb a megfelelő be/ki irányba, melynek hatására a jelző szabadra áll. Az elfordítás során kerül rögzítésre a rendelkező és a vágányúti kapcsológomb, illetve ellenőrzésre kerül további feltételek megléte (pl. váltóvégállások). Ha a vonat a jelzőt meghaladva a mögötte, de a fedezett váltókörzet előtt elhelyezett szigetelt sínre lép, a jelző „Megállj!” állásba kapcsolódik. Továbbhaladva a vonat a váltókörzet végénél lévő oldó szigeteltsíneket működteti. Amikor a szerelvény utolsó tengelyével is lehalad azokról, létrejön az oldás, a kapcsológombok alapállásba állíthatók.

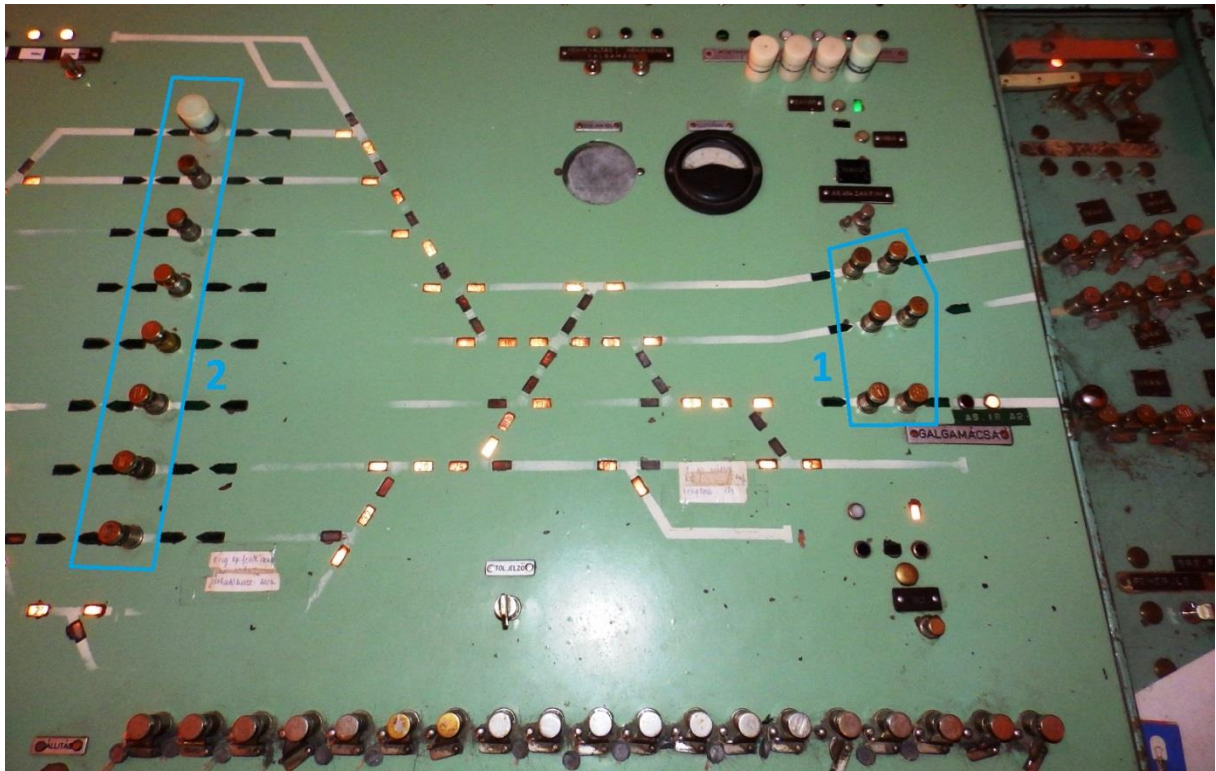
### ***Integra szabadkapcsolású berendezés [10] [11]***

Magyarországon ez a berendezés volt az első tisztán jelfogófüggéses típus. A váltók villamos állításúak, a jelzők fényjelzők, a teljes vágányhálózat kiszigetelésre kerül. Mechanikai függések helyett teljes egészében jelfogós, áramköri függőségek kerülnek alkalmazásra. Az egyedi jelfogókra épülő berendezés áramköreit egyedi tervezés során, szabadkapcsolásúan alakították ki. A forgalmi irodában elhelyezett pultszerű nyomógombos rendelkezőkészülék és a falra szerelt vágánytábla ad visszajelzést a

külsőtéri elemek állapotáról a kezelő felé, valamint nyomógombokkal teszi lehetővé a vezérlő parancsok adását.

A rendelkező pulton az állomás torzított helyszínrajza van felfestve. (5. ábra) A vonali vágánycsatlakozásokat reprezentáló vágánycsíkokban az 1 jelű Be és Ki iránykijelölő, az állomási vágányoknak megfelelő csíkok közepén pedig a 2 jelű vágánykijelölő nyomógombok helyezkednek el. A vágányút kijelölése az irány- és vágánynyomógomb egyidejű megnyomásával történik. A kezelés hatására a vágányút önműködően beáll. A gombok megnyomására átváltanak a vágányúthoz tartozó iránykivezérlő és vágánykivezérlő támaszjelfogók. A művelet csak akkor lehet sikeres, ha tiltott egyidejű vágányút nincs beállítva (menettermi függőség). A támaszjelfogók átváltott állapota rögzíti a vágányút kijelölés tényét, egészen a vágányút oldásáig.

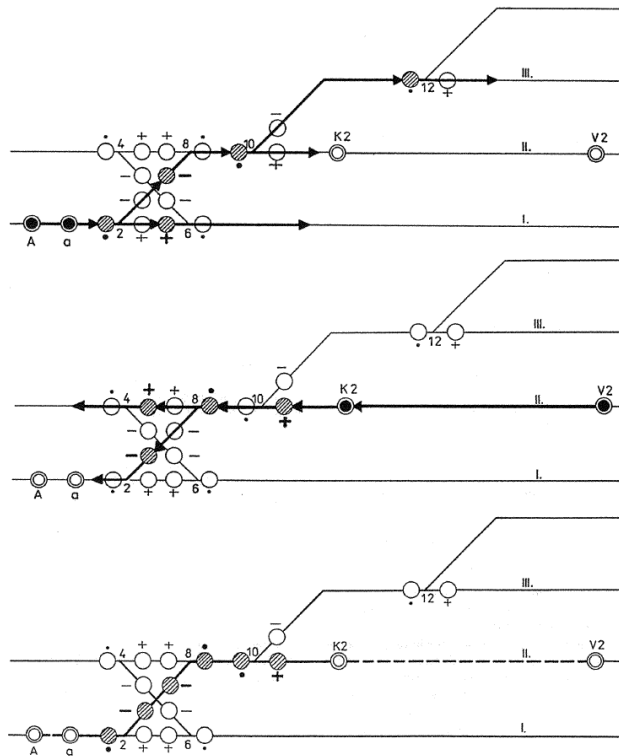
Az irány- és a vágánykivezérlő jelfogók érintkezői az elzárási tervnek megfelelő kapcsolásban az önműködő váltóállítást indító jelfogókat működtetik, melyek révén jön létre a váltók vágányútnak megfelelő automatikus állítása. A váltók helyes állásba való beállása után az adott vonali vágányhoz rendelt vágányútzáró támaszjelfogópár vált át annak ellenőrzésével, hogy az érintett és védőváltók valóban a helyes végállásukban vannak, és a vágányútban érintett, valamint a profilvédelmet szolgáló szigetelt szakaszok szabadok. Az ily módon történő vágányútlezárást követően megtörténik a startjelző szabad állásba állítása. A vágányút feloldása a közlekedő vonat hatására, önműködően megy végbe. Amikor a vonat a jelzőt meghaladja, az önműködően „Megállj!” állásba kapcsolódik. A vágányút feloldódása akkor következik be, amikor a vonat utolsó tengelye is áthaladt a vágányúthoz tartozó utoljára érintett szigeteltsínen.



5. ábra Integra típusú berendezés pultrjának részlete

### Dominó 55 [12]

Magyarországon széles körben alkalmazott, jelfogóegységekből geografikusan felépülő, szabadkapcsolású berendezés. Bár a berendezés működéses alapvetően nyomvonalas elvű, néhány funkció (például oldalvédelem) inkább táblázatos jellegű. Emellett a berendezés előtervében táblázatos menetterv illetve elzárási terv szerepel, valamint a váltók jelölése is a vágányutas elvű berendezésekben megszokottat követi. A kezeléseket az állomás vágányhálózatát leképező kezelőpulton kell elvégezni. A vágányút kijelölése start- és célgombok egyidejű megnyomásával történik. A kezelés hatására a váltóállító jelfogólánc működik, melynek feladata a kijelölt vágányút villamos leképezése, és igény esetén a váltóállítás vezérlése. (6. ábra)



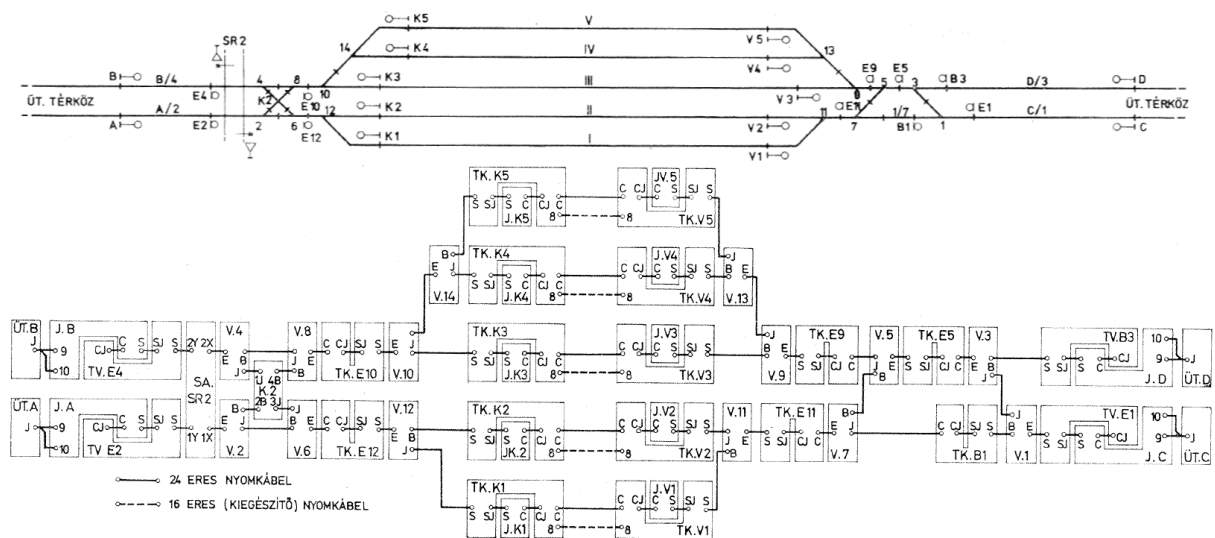
6. ábra A vágányút-kijelölés folyamata a D55 berendezésben (forrás: [8])

Minden biztosított váltóhoz három váltó állító jelfogó tartozik (közös, plusz és mínusz). A váltóállító jelfogók jelfogóláncot alkotnak, amiben egy következő jelfogó csak az előző meghúzása után húzhat fel. A vágányút kijelöléséhez két lánc, az adó és a nyugtázó indul el. Az adólánc minden esetben a nyílt vonal felől indul el (a 8. ábra szerint bejárat esetén a „A”, kijárat esetén pedig a „a” nyomógomb megnyomására). A nyugtázóláncot minden esetben kijárat jelzőhöz tartozó nyomógombjelfogó indítja (a 8. ábra szerint bejárat esetén a „V2”, kijárat esetén pedig a „K2” gomb). A láncok úgy épülnek fel, hogy az adott pontból elérhető minden lehetséges vágányút bejárásra kerül. A bejárás során csúcs felől érintett váltók közös, gyök felől érintett váltóknál pedig a bejárás irányától függően a plusz vagy a mínusz váltóállító jelfogó húz meg. A plusz és mínusz jelfogók meghúzásukkal a másik jelfogó áramkörét szakítják, így biztosítva, hogy mindig csak egyikőjük lehessen meghúzva. A vágányút az adó- és nyugtázóláncban egymást fedő útvonalak mentén alakul ki úgy, hogy kezdőpontját a kezdő nyomógombjelfogó, célpontját a cél nyomógombjelfogó, váltóit pedig a két (közös és plusz vagy mínusz) váltóállító jelfogó képezi le. A 8. ábra szemlélteti a folyamatot, a húzott állapotú jelfogókat szürkével jelölve. A modellekben a váltók kijelölése hasonlóan történik, Petri hálós modell esetén három hellyel, UPPAAL modell esetén három változóval.

Az egyszerű menetkizárásokat váltólezárással, a különleges kizárásokat célpontkijelöléssel – mely során az ellenséges menetek beállításának megakadályozását a célponthoz tartozó jelfogók végzik el – valósítja meg a berendezés. A kijelölés után megtörténik a váltók kívánt állásba állítása, majd a vágányút lezárása, mely több lépésből áll. Ezek a működések az állomás geografikus felépítésű áramköri rácsain futnak végig. A vágányút lezáródása után külön kezeléssel lehet a startjelzőt szabad állásba állítani. A vágányútban érintett elemek a vonat közlekedésének hatására egyenként oldódnak fel.

### Dominó 70 [13]

Jellemzően nagyobb állomásokon alkalmazott, jelfogóegységekből felépített tolatóvágányutas biztosítóberendezés. A külsőtéri pályaelemekhez tartozó jelfogóegységeket egy 24 érű nyomkábel köti össze, mellyel így különféle működésekhez kapcsolódóan 24 nyomáramkör jön létre. Az egységek helyszínrajznak megfelelő összekapcsolását mutatja a 7. ábra.



7. ábra Jelfogóegységek nyomkábelezése a D70 berendezésben (forrás: [14])

A vágányút kijelölését (első lépésben tárolását) az 1/2 nyomáramkörök végzik. A kijelölés a célpont-hoz tartozó egységéből indul a váltóknál szétágazva a start felé. A startpontba elérve, és annak kijelölését elvégezve visszafordul. A D55-tel ellentétben itt csak a keresés közben gyök felől érintett váltók jelölődnek ki (a nyomvonalas elvre jellemző megnevezések szerint bal vagy jobb szárukon). Az egyik száron való kijelölés a váltó másik szárát levágja a keresésből. A kijelölés a startba visszaérése után csak a vágányútban érintett elemekhez tartozó jelfogók maradnak húzva, és az áramkör saját magát tartja. Ekkor lehet újabb vágányutat kijelölni. A 3. nyomáramkör végzi a váltók állítását és lezárását. A célból indulva a nyomkábelezésnek megfelelően egységről egységre épül fel az áramkör. A váltóknál elvégzi az állírást, és helyes állás esetén a lezárást. Az oldalvédelem keresés (4/5 nyomáramkör) és nyugtázás (6/7 nyomáramkör) is nyomvonalaszerűen történik, csúccsal szemben érintett váltóknál kettéágazva, gyök felől találtaknál pedig a kereséssel ellentétes állásba állítva és lezárva a váltót.

A vágányút több nyomáramkörön történő ellenőrzése (végállások, lezárások, oldalvédelem, vágányszakaszok szabad voltának ellenőrzése, illetve ismételt kizárások) után a startjelző automatikusan szabadra áll. A vonatvágányutak további ellenőrzésekkel a tolatóvágányutakra épülnek fel. A vágányút feloldása a járművek közlekedése során elemenként, a 10. és 11. nyomáramkör működésével történik.



### **Elektronikus berendezések [15]**

A vágányúti logika az elektronikus biztosítóberendezésekben többnyire vágányúti elven valósul meg. Minden függőség megadása táblázatokkal történik, az elemek vágányutakhoz rendelésével. A táblázatok vágányutanként tartalmazzák a kizáró meneteket (meneterv), a beállítandó elemek állását, az oldalvédelmi elemeket és állásukat, a jelző szabadra állításának további feltételeit és a kivezérlendő szabad jelzési fogalmat, valamint az üzemszerű, elemenkénti oldás feltételeit.

Egyes elektronikus biztosítóberendezések azonban nyomvonalas elven működnek. Ezekben a vágányúti elemekhez (pl. váltók, jelzők) külön, az objektumoknak megfelelő szoftvermodulok tartoznak. A vágányúti logika a modulok vágányhálózatnak megfelelő modul- és elemkapcsolati tervei alapján épül fel. Az elemek feloldása mindkét elv szerint megvalósított berendezésben egyesével történik.

### **Összefoglalás**

A magyarországi biztosítóberendezések összehasonlítása szerkesztési elvük szerint az 1. táblázatban látható. Az előzőekben nem ismertetett berendezések jellemzése csak kialakításukra korlátozódik.

1. táblázat A Magyarországon alkalmazott berendezéstípusok osztályozása kialakítás és szerkesztési elv szerint (forrás: [1] [10] [15])

<b>Típus</b>	<b>Kialakítás</b>	<b>Szerkesztési elv</b>
Kulcsos berendezések	mechanikus	vágányutas
Siemens-Halske	mechanikus	vágányutas
VES	mechanikus/jelfogós	vágányutas
Integra	jelfogós	vágányutas
Dominó 55	jelfogós	nyomvonalas
Dominó 67	jelfogós	nyomvonalas
KÁ 69	jelfogós	vágányutas
Dominó 70	jelfogós	nyomvonalas
MRC	jelfogós	nyomvonalas
ECM	jelfogós	vágányutas
WSSB	jelfogós	vágányutas
KSW 90	jelfogós	nyomvonalas
Simis C	elektronikus	nyomvonalas
Elektra I	elektronikus	vágányutas
Elektra II	elektronikus	vágányutas
Simis IS	elektronikus	vágányutas

A táblázatos logika a mechanikus biztosítóberendezésekben egyszerűen leképezhető függőségi tengelyekkel és vonalzókkal (ld. 2.1.4. fejezet). Ugyanakkor a táblázatos forma kedvező a számítógépes feldolgozásnak is, így egyes elektronikus berendezésekben is alkalmazott elv. A kisebb állomási berendezések többnyire vágányutas elvűek (pl.: KÁ 69, ECM, WSSB), míg a nagyállomási berendezések jellemzően nyomvonalas kialakításúak (pl.: Dominó 70, MRC).

A vágányúti logika mindkét leképezése járhat előnyökkel és hátrányokkal. Ezek, és a bemutatott berendezések alapján összegyűjtött különböző szempontok szerinti összehasonlítás látható a

2. táblázatban. A két elv példák alapján történő vizsgálata során meghatározott tulajdonságok modellekben való megjelenése is említésre kerül. A modellezési döntések részletes leírása későbbiek során, a 3.1. és 3.2. fejezetekben látható.

2. táblázat A vágányutas és nyomvonalas elvek összehasonlítása

Szempont	Vágányutas elv	Nyomvonalas elv
Vágányút kijelölése	Fogadóvágány + csatlakozó vonal + be/ki irány.	Start és cél megadása után kereséssel.
Vágányút kijelölésének jellemző fenntartása	A vágányút feloldásáig.	A vágányút kijelöléséig vagy lezáródásáig.
Vágányúti kizárások megvalósítása	Vágányutakhoz kapcsolódóan, a menetterv alapján. Ha van, külön rendelkezőkészülékben.	Elemenként, vágányútban való érintettség alapján.
Vágányúti elemek beállítása	Vágányutakhoz kapcsolódóan, az elzárási terv alapján. Ha van, külön állítókészülékben.	A kijelölő láncok alapján, elemenként az elvárt állással a ténylegest összevetve.
Oldalvédelem	Vágányutanként előre definiált.	Az érintett váltók másik szarán keresve.
Jellemző vágányúttoldás	Egyben, a váltókörzet utáni oldószakasszal.	Elemenként, szekvencia szerint.
Váltóállás jellemző jelölése	+: egyenes -: kitérő	+: jobbra terel -: balra terel
Jellemző megvalósítás	Mechanikus, (jelfogós), elektronikus.	Jelfogós, elektronikus.
Jellemző alkalmazás	Kisállomási berendezések.	Nagyállomási berendezések.
Előny	Könnyű elvi tervezés.	Könnyű telepíthetőség, átépíthetőség, ellenőrizhetőség.
Hátrány	Nehéz átépítés, bonyolultabb ellenőrzés.	Bonyolultabb elvi tervezés. Kisebb állomásokon feleslegesen bonyolult felépítés.

A vágányutas elvű berendezésekben egy vágányút beállításához jellemzően három információ megadása szükséges, ám jellemzően ezekből több egyszerre kerül meghatározásra. Például elektromechanikus berendezésekben a kapcsológomb elfordításával egyszerre mindhárom, mechanikus berendezéseknél a tologomb eltolásával egyszerre kettő (fogadóvágány + csatlakozó vonal). A modellekben csak egy irányban lehet vágányutakat beállítani, a kiválasztás csatlakozó vonal, fogadóvágány sorrendben történik. Nyomvonalas elvű berendezésekben a vágányút kijelölés a start és célgombok megnyomásával történik. A modellekben először a start, majd a cél kiválasztása történik.

Jellemző, hogy a menettervi függőségeket koordinátás elvű berendezésekben a kijelölő eszközök végzik, így ezekkel új menet beállítására nincs lehetőség, tehát nem rendelkeznek vágányúttárolási funkcióval. Kivételek ez alól az elektronikus berendezések, melyekben a kijelölés más szinten, a központi logikától függetlenül történik. A geografikus felépítésű berendezésekben új vágányút kijelölése jóval hamarabb, egy korábbi sikeres kijelölést illetve lezárást követően lehetséges. Ebben az esetben egy új parancs eltárolódik, és az előző vágányút feloldása után lesz hatásos (vágányúttárolás).

Vágányutas elvű berendezésekben a kizárások az objektumoktól függetlenül, az állítási parancs kiadása előtt kerülnek megvizsgálásra (akár úgy is, hogy a kezelőeszköz nem mozdítható el). Amennyiben egy menet nem állítható be, egy elem sem kap vezérlést. Ezzel szemben a nyomvonalas berendezésekben egy vágányút beállítására vonatkozó kezelés akkor is hatásos lesz, ha van már kizáró menet beállítva. Ekkor a teljes vágányút nem áll be, csupán a fel nem használt elemek vezérlődnek a kívánt állásba. A modellekben egyszerre csak vágányút beállítására adható ki utasítás.

Az oldalvédelmi elemek beállítása vágányutas elven felépülő berendezések esetében az elzárási terv függőségeinek megvalósításával egy időben történik. A nyomvonalas elvű berendezésekben a váltók oldalvédelmet a lezárás során keresnek maguknak. A modellek az oldalvédelmet nem kezelik.

A koordináta elvű berendezésekben a feloldás jellemzően egyszerre történik. A váltókörzetek végén elhelyezett oldószakaszok foglalttá válása majd felszabadulása egyértelműen jelzi, hogy a vonat a vágányutat felhasználta, így az feloldható. A nyomvonalas berendezésekben az elemek egyesével, saját felhasználásuk szerint oldhatók, minek előnye, hogy új vágányutak hamarabb állíthatók be. A modellekben a vágányutas elv esetén a vágányút feloldása egyben, a nyomvonalas elv esetén elemenként történik.

A váltóállások jelölése is a két elv közötti szemléletmódot tükrözi. Vágányutas elv esetén a váltó két iránya közötti alá-fölérendeltség (egyenes az alapállás, stb.) határozza meg a jelölést. A geografikus elven felépülő berendezések esetén fontosabb az a szempont, hogy az elemek kapcsolati rendszerében a váltó melyik ága mivel szomszédos. Így ezekben a berendezésekben a váltók állását az jeleníti meg, hogy csúcs felől közlekedve melyik irányba terelnek. A jelölések a modellekben is követik az elnevezéseket, de +/- jelek helyett az irányok kezdőbetűjével történik azonosításuk.

Mindkét elv rendelkezik előnyös és hátrányos ismérvekkel. Napjainkban azonban az elektronikus berendezésekben nem jelennek meg ilyen tisztán a táblázatban szereplő tulajdonságok. Inkább jellemző, hogy a két elv forgalomszervezés szempontjából előnyösebb tulajdonágával rendelkeznek (pl. vágányútoldás elemenként mindkét esetben). Ennek ellenére általánosan elmondható, hogy a moduláris felépítésű nyomvonalas berendezések alkalmazása nagyobb rugalmasságot tesz lehetővé. Amennyiben átépítések, vágányzárak miatt az állomás topológiáján módosítani kell, a berendezéssel könnyen lehet lekövetni a változtatásokat. Ez nem mondható el a koordináta elvű berendezésekről, melyeket az új vágányutak szerint kell átalakítani. A moduláris felépítésből adódik, hogy nagyobb állomások esetén is könnyebben tervezhető meg, mint egy vágányutas berendezés. Tehát elsősorban az objektumokhoz tartozó egységeket kell jól megalkotni, ezek összerendezése már egyszerűbben történik. Ezzel szemben a vágányutas berendezések generikus létrehozása kisebb ráfordítással elvégezhető, viszont egy adott helyszínre történő alkalmazása összetettebb. Ilyen esetben alkalmazhatók olyan szoftverek, melyek segítségével a topológia alapján akár a modulok automatikus összerendezése, akár a függőségi táblázatok automatikus legenerálása elvégezhető.

Az újonnan telepítendő elektronikus biztosítóberendezéseket vizsgálni, tesztelni kell. [16] Akár nagyobb állomások esetében is viszonylag kis időráfordítással elvégezhető a nyomvonalas elvű berendezések tesztelése. Mivel ezek generikus és országspecifikus alkalmazása korábban elfogadott, csak a telepített berendezés ezekkel való egyezését és az objektumok megfelelő összerendezését kell vizsgálni. Ezzel szemben a vágányúti logika leképezését vágányutas elven megvalósító berendezések állomásspecifikus szoftverének tesztelése időigényesebb lehet. Mivel ezekben minden függőség, funkció vágányutakhoz kapcsolódik, az összes lehetséges vágányutat be kell állítani, és a feltételek teljesülését elemezni kell. A két különböző elven megvalósított berendezések vizsgálatában így a

ráfordított munkabeli különbség nagyobb állomások esetén jelentős. A dolgozatban a modellellenőrzés mindkét elv modelljére azonos kifejezésekkel, vágányutakhoz kapcsolódóan történt.

## 2.2. Formális módszerek

A formális módszerek olyan elsősorban az informatika területén használt matematikai (jellemzően diszkrét matematika és matematikai logika) alapú technikák, melyek a rendszer- illetve a szoftver- és hardverfejlesztés során alkalmazhatóak. [17] A formális specifikációs módszerek használata egyértelműséget, ezáltal mások általi könnyű értelmezhetőséget, ellenőrizhetőséget, és ezek mellett automatikus feldolgozhatóságot tesz lehetővé. Jelen alfejezetben elsősorban a dolgozat tárgyát képező biztosítóberendezési elvek modellezéséhez és modellellenőrzéséhez használt mélységben kerülnek bemutatásra a formális módszerek elméleti alapjai, illetve az alkalmazott eszközök.

### 2.2.1. Verifikáció

Napjainkban a számítógépes rendszerek egyre inkább körülveszik az embereket, egyre nagyobb szerepet játszanak a hétköznapokban és egyre több helyen, egyre kiterjedtebben jelennek meg. [19] Ezáltal e rendszerek összetettsége, bonyolultsága is folyamatosan növekszik. A komplexitás ilyen irányú változás azt is magával hozza, hogy az újabb rendszerek kevésbé átláthatók, nehezebben ellenőrizhetők. A kijelentés nem csak a hétköznapi életben tehető meg, hanem a magasabb elvárások miatt fokozott jelentőségű biztonságkritikus rendszerek tekintetében is. Ez esetben olyan rendszerekről van szó, melyek meghibásodásukkal emberek életét vagy a környezet épségét veszélyeztethetik. Közéjük tartoznak a közlekedést irányító berendezések, így a vasúti biztosítóberendezések is. Mivel az ilyen kimenetelű baleseteket meg kell előzni, az egyre összetettebb rendszerek megjelenése mellett egyre nagyobb hangsúlyt kap az őket felépítő hardverek és szoftverek ellenőrzése.

Kétféle ellenőrzési folyamatot kell megkülönböztetni: a verifikációt és a validációt. Egy rendszer validációja azt ellenőrzi, hogy sikerül-e megfelelni a külső körülményeknek, a felhasználó elvárásainak. A validáció során a „Jó rendszert építünk?” kérdésre kell választ keresni. A verifikáció során egy fejlesztési fázis végén, a fázis elején megfogalmazott feltételek teljesülését kell ellenőrizni. Tehát a verifikáció a „Jól építjük a rendszert?” kérdésre keres választ.

A verifikáció során ellenőrizendő különböző elvárt tulajdonságokat a specifikáció tartalmazza. Tehát a verifikálás során csak a specifikációnak való megfelelés bizonyítható, a rendszer abszolút jósága nem. A rendszer specifikációjából származik egyrészt a tervezés során létrejött formális modell, prototípus, illetve termék, ezeken kell a verifikációt elvégezni. Másrészt szintén a specifikáció alapján fogalmazódnak meg a követelmények, elvárt tulajdonságok, melyek teljesülését ellenőrizni kell. A verifikáció végrehajtható a formális modellen, illetve a prototípuson vagy az elkészült terméken (ld. a későbbiekben: 8. ábra).

A fejlesztés minél korábbi fázisaiban felfedezett hibák kijavítása jóval olcsóbban kivitelezhető. Ebből következik, hogy a verifikációnak is a korábbi fázisoktól kezdődően lehetőleg a teljes fejlesztési ciklust végig kell követnie. Szoftververifikáció során leggyakrabban alkalmazott technikák a szakértői felülvizsgálat a tesztelés. Előbbi esetében a fejlesztéstől független, de kompetens szoftverfejlesztők az elkészült programkódot futtatás nélkül statikusan elemzik, és értékelik. Utóbbi viszont kifejezetten dinamikus módszer, melynek lényege az elkészült szoftver különböző bemenetekre adott válaszainak vizsgálata. Az előre, akár automatikusan meghatározott, és a szoftverre adott tesztesetek eredményeit a specifikációval kell összehasonlítani. Mivel a két módszer eltérő típusú hibák ellen használható, gyakran együtt alkalmazzák őket. Ennek ellenére sem külön-külön, sem együttesen nem alkalmasak az összes elkövetett hiba feltárására.

Hardver rendszerek esetén a leggyakrabban alkalmazott hagyományos verifikációs technikák az emuláció és a szimuláció. Mindkét módszer a teszteléssel állítható párhuzamba. Emuláció során egy általánosan használható hardver, az emulátor kerül a tervezett hardver áramköreinek megfelelő konfigurálásra. Az így elkészített emulátorra ráadott bemenetekre válaszul kapott viselkedés a specifikációval összevethető. A szimuláció során a fejlesztett hardver modellje készül el, amit a szimulátor programban lehet vizsgálni, az előzőekkel analóg módon. Bár a szimuláció az egyik legelterjedtebb hardververifikációs technika, az itt felsorolt módszerekkel sem lehet a hibák feltárásában teljeskörűséget elérni (a teszteléshez hasonlóan).

Az említett technikák mellett mind a szoftver-, mind a hardverfejlesztés során egyre hangsúlyosabb szerepet kap a formális módszerek alkalmazása. A formális módszerek rendszerek tervezésében és verifikációjában alkalmazott matematikai alapú módszerek. Az elnevezés széles körben fed le különböző technikákat, mint például különböző specifikációs nyelvek vagy formalizmusok, számítási modellek, automatikus tételbizonyítás, programhelyesség bizonyítás, modellellenőrzés, modelltranszformáció, de különböző absztrakciós és finomítási technikák is ide sorolhatók. Nem minden módszer alkalmazható a fejlesztés összes fázisában, egyesek inkább a tervezés, specifikáció, míg mások a verifikáció folyamatát segítik.

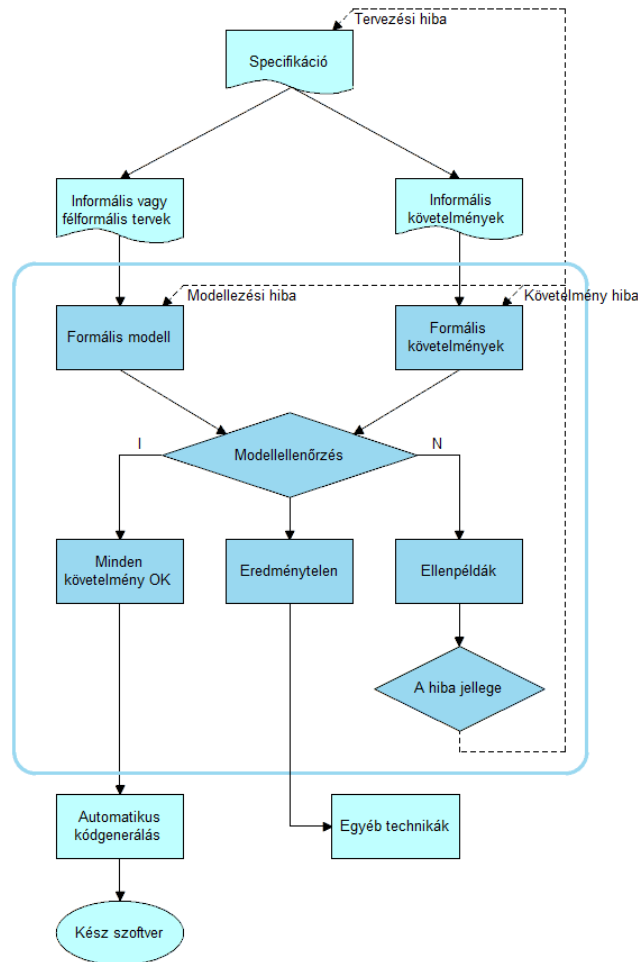
### **2.2.2. Modellellenőrzés**

A modellellenőrzés egy olyan formális módszer, amely a vizsgálandó rendszer egy modelljéről és annak az elvárt működését tartalmazó specifikációjáról a rendszermodell állapotainak (vagy más szóval állapotterének) szisztematikus bejárásával dönti el, hogy a rendszermodell a specifikációt teljesíti-e vagy sem. Ha a teljesítés nem sikerül, a modellellenőrzés (lehetőség szerint) ellenpéldát hoz a specifikációtól eltérő működésre. [17] [19] [18]

A modellellenőrzésnek két pontosan, egyértelműen leírt bemenetet kell kapnia. (8. ábra) Az ellenőrzést a rendszer modelljén kell elvégezni a specifikáció alapján meghatározott tulajdonságokra. A modell azt fogalmazza meg, hogyan viselkedik a rendszer, míg a követelmények azt írják elő, mit kell és mit nem szabad tennie a rendszernek. A rendszer modellezése általában valamilyen véges átmeneti rendszerrel történik. Az átmeneti rendszerek állapotokból és az állapotok közti átmenetekből állnak. Az egyes állapotok a rendszer pillanatnyi jellemzőit írják le, az átmenetek pedig azt határozzák meg, hogyan, milyen úton és milyen feltételekkel kerülhet a rendszer egy másik állapotba. A modellezés során meg kell határozni a célokat, el kell végezni a lehatárolást és meg kell találni a megfelelő absztrakciót, azaz a modell ne legyen se feleslegesen részletes, se túlságosan egyszerűsített. Verifikáció előtt lehetőség van a modell szimulációjára, és bizonyos hibák felfedésére, kijavítására. Helyes verifikációhoz szükséges az ellenőrizendő követelmények szabatos megfogalmazása. Erre leginkább egy tulajdonságokat leíró nyelv alkalmas, melyek közül a temporális logikák a legelterjedtebbek. A verifikáció során az vizsgálandó, hogy az elkészített modell valóban modellelje az elvárt tulajdonságokat megadó temporális logikai kifejezéseknek. Nem verifikációs, hanem validációs eset annak bizonyítása, hogy a formális modell és a formális követelmények együttese megfelelő leírása a nem formális tervezési elgondolásoknak.

A modellellenőrzés menetét (bekeretezve), és a hozzá kapcsolódó egy lehetséges fejlesztési folyamatot mutat az 8. ábra. A rendszerspecifikáció meghatározása után abból kétféle dokumentum keletkezhet. Egyrészt elkészülnek a leendő rendszer tervei, másrészt összegyűjtésre kerülnek a fejlesztett rendszerrel szemben támasztott követelmények, elvárt tulajdonságok. A tervek alapján megfelelő

absztrakcióval megalkotható a rendszer formális modellje, míg egy külön, akár párhuzamos folyamat során a szöveges követelmények formális követelményekké fogalmazhatók át.



8. ábra A formális módszereken alapú fejlesztés egy lehetséges folyamatábrája (forrás: [17] [18])

A modell és a formális követelmények megfogalmazása után kezdődhet a(z akár automatikus) modellellenőrzés. A követelményspecifikáció általában több kifejezésből áll, melyek ellenőrzése egyenként történik. A modellellenőrző program a modell teljes állapotterének bejárásával, minden állapotban ellenőrzi a specifikációban megadott tulajdonságokat. Ha a feltétel nem teljesül, a program előírt tulajdonságot sértő ellenpéldát generál. A nem kívánt állapotba való eljutás a szimulátorban visszakövethető és elemezhető. A jelenleg általánosan használatos modellellenőrző programok  $10^8$ -tól különböző technikákkal akár  $10^{20}$ -ig terjedő méretű állapotterek bejárására, elemzésére is képesek.

Az elvégzett modellellenőrzés háromféle kimenettel járhat. Elsőként, teljesült tulajdonság esetén az ellenőrzés további feltételekkel folytatható. Valamennyi követelmény teljesítése esetén a modellellenőrzés sikeres volt. Sikeres modellellenőrzés esetén automatikus kódgenerálással a formális modellből (kiegészítésekkel) akár futtatható programkód készíthető.

Másodsorban amennyiben az elvárt eredmény nem teljesül, annak több oka is lehet:

- modellezési hiba
- követelmény hiba
- tervezési hiba.

Modellezési hiba esetén a modell nem felel meg a valós rendszer működésének, ez a hibafajta a modell korrigálásával javítható. Módosított formális modell esetén a korábban ellenőrzött követelmények újraellenőrzése szükséges. Ha nincs indokolatlan hiba a modell és a tervezett rendszer között, akkor követelmény vagy tervezési hiba áll fenn. Előbbi esetén a formálisan megfogalmazott követelmény nem felel meg az informális specifikáció által leírtaknak. A hiba a formális specifikáció javításával kezelhető. A tervezési hibák felderítése a modellellenőrzés alapvető feladata. Ebben az esetben a tervezett rendszert javítani kell, és vele együtt a formális modell átdolgozása is szükséges.

Valós rendszerek modellezése során az állapotok száma jóval nagyobb lehet annál, amit a rendelkezésre álló memóriában tárolni lehetséges. A modellellenőrző szoftverek elvégzik az állapottér belső reprezentálását (pl. logikai függvények alakjában) annak érdekében, hogy nagyobb méretűeket is kezelni tudjanak. Ennek ellenére a modellellenőrzés harmadik kimenete lehet, hogy nem jár eredménnyel. A helyzet kezelésére többféle további technika nyújt megoldást. Erre példa a valóság további absztrakciója, a modell felépítésében lévő szabályosságok kihasználása vagy az állapottér csak egy meghatározott részének ellenőrzése.

A modellellenőrzés használata során számos előnye mellett figyelemmel kell lenni néhány megkötésre is. Hiányosságok lehetnek az ellenőrzés mindkét bemenetében. A modellellenőrzés végrehajtása nem a valós rendszeren, hanem annak modelljén történik. Tehát az ellenőrzés pontossága nagyban függ a modell pontosságától. Ha a modell ellenőrzése sikeres volt, még akkor is előfordulhatnak a hardverben gyártási, a szoftverben kódolási hibák. A modellellenőrzés során csak a felírt tulajdonságok ellenőrzése lehetséges. Fontos tehát, hogy a verifikáció másik bemenete, a specifikáció minden ellenőrizendő követelményt tartalmazzon. A formális modellek és specifikáció elkészítéséhez megfelelő szakértelemmel és absztrakciós készséggel rendelkező egyén szükséges. Akár helyes absztrakció esetén is figyelembe kell venni, hogy a modellellenőrző szoftverben is lehetnek hibák, így a végeredmény helyessége nem feltétlenül garantált.

### **2.2.3. Átmeneti rendszerek, időzített automaták**

A modellellenőrzéshez használt alkalmas modell megalkotása olykor nehéz feladat lehet. Olyan egyszerűséggel illetve részletességgel kell létrehozni, hogy lényeges elem ne vesszen el, de kezelhetetlenül nagy se legyen. Az állapotokból és a köztük definiált átmenetekből álló átmeneti rendszerek széles körben alkalmazott technikák a konkurens rendszerek modellezésére. [20] Az átmenetek lehetnek a rendszer által végrehajtott akciók vagy külső események. Például a dolgot tárgyát képező modellekben esemény a vágányút kezdő- és célpontjának kiválasztása, akciók az erre végbemenő biztosítóberendezési funkciók.

Átmeneti rendszerek esetén az állapotok és az átmenetek lehetnek paraméterezettettek. Meg lehet különböztetni kezdő- és végállapotokat, vagy bemeneti és kimeneti átmeneteket, de a két elem kritikusága is megadható. Továbbá a jobb érthetőség, illetve a verifikáció egyszerűsítése végett az átmenetek lehetnek címkézettek. Ez utóbbi tulajdonsággal a dolgotban alkalmazott modellező eszközök (2.2.7. fejezet) nem rendelkeznek.

Az átmeneti rendszerek modellezése rendszerint az állapotoknak megfelelő körökkel és az átmeneteknek megfelelő nyilakkal történik. Az effajta rajztechnika a rendszer modelljét (bizonyos méretig) könnyen áttekinthetővé és értelmezhetővé, illetve működését követhetővé teszik. Az egymást követő állapotátmenetekből álló halmazokat utaknak nevezzük. Végtelen utak esetén csak kezdőállapotról beszélhetünk.



Egy-egy átmeneti rendszerrel csak egy-egy rendszerkomponenst vagy folyamatot lehet modellezni. Az egész rendszer és működésének leírásához szükséges a komponensek kapcsolatainak, kölcsönhatásainak leírása is. Erre a célra szolgál a rendszerek szinkronizált szorzata, melynek segítségével a rendszer komponenseinek összekapcsolásával az egész rendszert leíró globális átmeneti rendszer származtatható. A szinkronizált átmeneteket globális átmeneteknek nevezzük.

Megkülönböztethetők szinkron és aszinkron átmeneti rendszerek. Szinkron működés esetén minden egymást követő időegység alatt minden komponens végrehajt egy átmenetet. Ha azonban az egyes rendszerkomponensek tetszőleges ideig maradhatnak egy állapotban, vagy az átmenetek egymáshoz viszonyított sebessége nem kerül definiálásra, aszinkron működés áll fenn. Modellezés során elképzelhetők szinkron-aszinkron rendszerek is, illetve szinkron rendszerekkel is lehet aszinkron működést modellezni, ugyanabba az állapotba visszatérő átmenetekkel.

Számos biztonságkritikus rendszerben játszik fontos szerepet az idő. Ennek megfelelően az időt az ilyen rendszerek modellezésében is kezelni kell tudni. Az időzített rendszerek olyan rendszerek, melyek helyessége nem csak az átmenetek logikai sorrendjétől, hanem azok idejétől is függ. Valós idejű rendszerek modellezésére időzített automatákat használnak. Ezek olyan nem determinisztikus automaták, melyek véges sok időzítővel (órával) rendelkeznek. Az órák segítségével az átmenetekhez feltételek (őrfeltételek) szabhatók, miszerint egy átmenetre csak az óra megfelelő állása esetén van lehetőség. A követelményeket egész számokkal, matematikai összehasonlításokkal (kisebbs/nagyobb/egyenlő) kell megadni. Az átmenetek nulla időegység alatt hajtódnak végre. Feltételek az órákkal kapcsolatban hozzárendelhetők helyekhez is, ezek neve invariáns. Az órák alaphelyzetbe állítása az átmenetek során lehetséges. Minden óra azonos ütemben jár, és az utolsó lenullázása óta eltelt időt tárolja. Az órák értékének megváltoztatására szünettel vagy újraindítással van lehetőség. Időzített automatákban az aktuális állapot megadása a helyel, amiben az automata éppen tartózkodik, és a hozzá tartozó órák idejével történik.

Összetett időzített rendszerek modellezése időzített automata hálózatokkal történik. Az automaták közötti kommunikáció többféleképp történhet, a legalapvetőbb a kézfogás elvén alapuló szinkronizáció. Ennek során a szinkronizációt végrehajtani kívánó komponens igényt jelent be egy előre definiált csatornán, melyet egy másik komponens fogadni képes. A szinkronizálás csak akkor lehetséges, ha mindkét automata egyszerre végre tudja hajtani saját átmenetét. Az időzített automaták valamennyi előbb említett működése egyszerre, egymással közösen történik (pl. a szinkronizációs átmenethez is rendelhető őrfeltétel, stb.).

#### **2.2.4. Petri hálók**

A Petri hálókat az 1930-as években Carl Adam Petri határozta meg először, eredetileg kémiai folyamatok leírása céljából. A matematikai alapok kidolgozását az 1960-as években végezte el. A Petri hálók alkalmazhatók konkurens, aszinkron, elosztott, párhuzamos, nemdeterminisztikus rendszerek modellezésére. [21] [22] [23] [24] Előnye más formalizmusokhoz képest, hogy kompaktabban fejezi ki az állapotokat, és szemléletesebben jeleníti meg a szinkronizációt. Bizonyos szempontból a Petri hálók is tekinthetők átmeneti rendszernek.

A Petri háló helyekből, átmenetekből és irányított élekből, mint elemekből áll. Az élek kötik össze a helyeket az átmenetekkel és megfordítva, ugyanakkor a helyek és az átmenetek saját csoportja között nincsen közvetlen éllel megvalósított kapcsolat, azaz a Petri hálók irányított páros gráfok. A helyek állapotát a hozzájuk tartozó tokenek határozzák meg. A helyek jelölése körrel, a bennük lévő

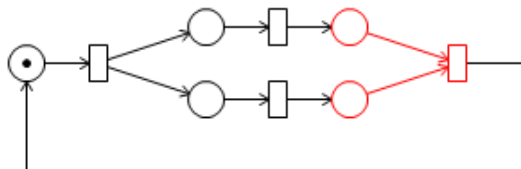
tokeneké ponttal, a tranzícióké négyszöggel vagy vastag vonallal, az éleké nyilakkal történik. Egy él élsúllyal látható el, ami az azonos tranzíciók és helyek között kihúzott képzeletbeli élek számával egyezik meg.

Az egyes helyeken tetszés szerinti számú token fordulhat elő, amely tokenek úgy kerülnek át más helyekre, hogy őket az átmenetek, a tranzíciók eltüzelik. Egy tranzíció csak akkor tüzelhet, ha engedélyezett. Egy tranzíció csak akkor lehet engedélyezett, ha a bemeneti helyein legalább az élsúlyoknak megfelelő mennyiségű token található. Több, egyszerre engedélyezett tranzíció között a tüzelési sorrend meghatározása nondeterminisztikusan, véletlenszerűen történik. Egy tüzelés során a bemeneti és kimeneti tokenek számának nem kell megegyezniük, token elnyelés és generálás megengedett. Egy hálózat állapotát a helyein lévő tokenek száma, a tokeneloszlás jeleníti meg. Egy tranzíció tüzelésével a tokeneloszlás megváltozik, a rendszer új állapotba kerül. A Petri hálók kiterjeszthetők helyekhez kapacitáskorlát, tranzíciókhoz prioritás rendelésével, tiltó élek, stb. bevezetésével.

A Petri hálók dinamikus tulajdonságai az állapotváltozások során vizsgálhatók. Az elérhetőségi analízis segítségével bizonyos dinamikus tulajdonságok megállapíthatók. Elérhetőségi probléma során azt kell felderíteni, hogy egy adott kezdőállapotból a vizsgált állapot elérhető. Egy Petri háló korlátos, ha bármely állapotban minden helyén, helyenként legfeljebb a korlátnak megfelelő token található. A korlátosság speciális esete a biztosság, amikor minden helyen legfeljebb egy token lehet. A korlátos Petri hálók véges állapotterrel rendelkeznek és fordítva. Egy tranzíció élősége arra utal, hányszor tüzelhet egy adott állapotátmenet-sorozatban. Ennek egy speciális esete a holtpontmentesség, azaz minden állapotban létezik legalább egy tranzíció, ami engedélyezett. A megfordíthatóság azt jelenti, hogy a kezdőállapot bármely követő állapotból elérhető. A visszatérő állapot pedig olyan a kezdőállapottól különböző állapot, mely bármely őt követő állapotból elérhető.

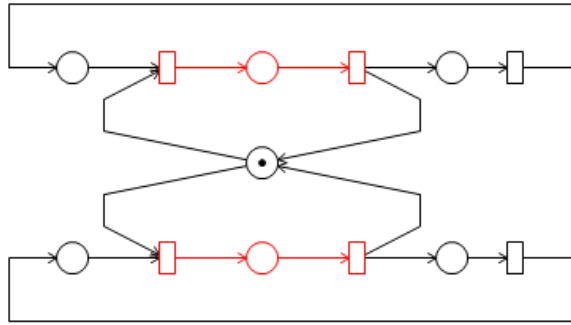
A Petri hálók modellezésénél lehetőség van a modelleket több hierarchia szinten kezelni. Ekkor az egyes hálórészek alhálókbá szervezhetők, aminek célja és ezzel együtt iránya kettős lehet. Célszerű lehet a modellt finomítani, egyes folyamatok, részrendszerek így külön részletezhetők (top-down modellezési irány). Ugyanakkor a nagyobb rendszer is összeállítható az előre elkészített alhálókból (bottom-up modellezési irány). Az alhálók közötti kapcsolat megteremtésére két standard eszköz létezik: az ún. interfész helyek és a helyettesítő tranzíciók. A Petri hálók segítségével lehetőség nyílik speciális funkciók modellezésére is, például:

- Több folyamat szinkronizációját a megfelelő számú helyről egy tranzícióba vezető élekkel lehet megjeleníteni. (9. ábra) Egy állapotátmenet tehát csak akkor jöhet létre, ha valamennyi feltétele egyszerre teljesül.



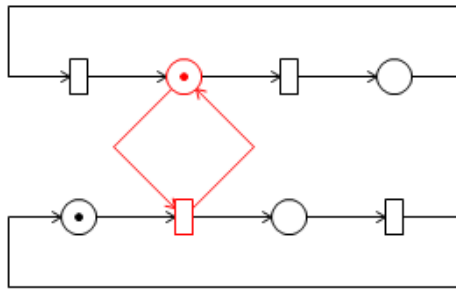
9. ábra Szinkronizáció megvalósítása Petri hálóval

- Konkurencia, egymást kizáró folyamatok megjelenítésére például egy közös helyel és a benne lévő tokennel van lehetőség. (10. ábra) Ekkor a helyekhez csatlakozó tranzíciók közül csak az elsőnek tüzelővel megjelenített folyamat tud végbemenni, a többi nem lehet engedélyezett.



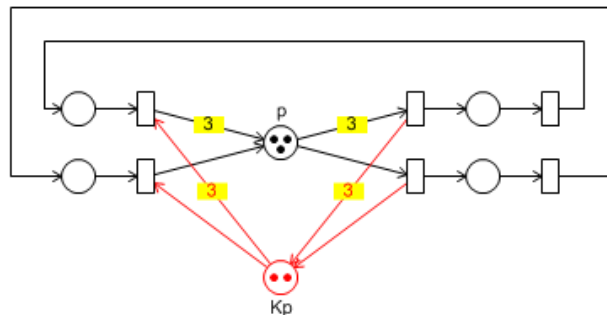
10. ábra Kölcsönös kizárás megvalósítása Petri hálóval

- Petri hálókkal megvalósítható ellenőrzés modellezése is. (11. ábra) Ebben az esetben egy adott helyről a token elvétele nem szükséges, csupán annak ottlétéről kell bizonyosságot nyerni. Az ellenőrzést az ellenőrzött helyről a tranzícióba és onnan visszavezető éllel lehet megoldani. Ez lényegében a helyen lévő token(ek) eltüzelését és ugyanabban a lépésben a visszahelyezését jelenti.



11. ábra Ellenőrzés megvalósítása Petri hálóval

- Szükséges lehet, hogy egy folyamat (hálórész) lefutása bizonyos feltételek között egymás után csak korlátozott számban (például csak egyszer) legyen megengedhető. Ekkor egy kapacitáskorlátként szolgáló hely felvétele szükséges. (12. ábra) Ezen a helyen annyi tokenet kell elhelyezni, ahányszor az adott folyamat futása megengedett. Az indítás kezdetét jelző tranzíció a kapacitásként szolgáló helyről mindig egy tokenet eltüzel. Tehát a kezdőtranzíciók csak addig lesznek engedélyezettek, amíg ezen a helyen token van. A folyamat lefutásának végén lévő tranzíció a tokenet a kapacitáskorlátként szolgáló helyre visszahelyezi.



12. ábra Korlátos kapacitás megvalósítása Petri hálóval

### 2.2.5. Temporális logikák (CTL)

A vizsgált rendszerek tulajdonságainak megfogalmazására alkalmas eszköz a logika. [25] Különböző logikák alkalmazhatóak a vizsgált rendszerektől és a vizsgálandó tulajdonságoktól függően. Kijelentés logikával ellenőrizhetők az átmeneti rendszerek állapotainak és átmeneteinek lokális tulajdonságai. A temporális logika alkalmas az átmeneti rendszerek időbeli változásainak vizsgálatára. Az időbeliség jelen esetben az események sorrendjére vonatkozik egy diszkrét vagy folytonos időtartományban. Az időt időpillanatok egymást követő sorozataként kezelő temporális logikák alkalmasak az átmeneti rendszerekben utakra vonatkozó tulajdonságok kezelésére. Lineáris temporális logikák alkalmazása során a modell egy-egy lefutása kerül figyelembevételre, mely során minden állapotnak egy rákövetkezője lehet. Léteznek azonban olyan elágazó temporális logikák, amelyek kezelni tudják azt, hogy egy eseményt több is követhet. Így ezek alkalmasak az átmeneti rendszer elágazó struktúráját figyelembe vevő állapot tulajdonságok specifikálására.

Elterjedten használtak a CTL kifejezések, a név az angol Computation Tree Logic, azaz számítási fa logika kifejezés rövidítéséből ered. [26] [27] [28] [29] A logika neve utal arra, hogy formuláival leírhatókká válnak az állapotokhoz tartozó számítási fák elágazó tulajdonságai. Az egyes összetett  $p$ ,  $q$ ... jelölésű formulák a következő elemekből állnak össze:

- Atomi  $P$ ,  $Q$ ... kijelentések
- Logikai operátorok
- Útvonalakon értelmezett temporális operátorok
- Útvonalkvantorok

A  $p$  és  $q$  kijelentésekre alkalmazhatók a különböző logikai operátorok:

- Negálás (jele:  $\neg$  vagy !)
- És (jele:  $\wedge$  vagy &)
- Vagy (jele:  $\vee$  vagy |)
- Implikáció (jele:  $\rightarrow$ )

Az implikáció a „ha  $p$ , akkor  $q$ ” jellegű követelmények leírására alkalmas. A kifejezés átírható a  $p \rightarrow q = (\neg p) \vee q$  alakba. Későbbiekben a modellellenőrzés során (4. fejezet) ez a formula gyakran kerül alkalmazásra.

A temporális operátorok a következők lehetnek:

- A következő állapotban  $p$  igaz lesz (neXt  $p$ , jele:  $X p$ )
- Egy elérhető állapotban igaz lesz  $p$  (Future  $p$ , jele:  $F p$ )
- Minden elérhető állapotban igaz lesz  $p$  (Globally  $p$ , jele:  $G p$ )
- Egy elérhető állapotban igaz lesz  $p$ , és addig minden állapotban igaz  $q$  ( $q$  Until  $p$ , jele:  $q U p$ )

Az alkalmazható két útvonalkvantor az alábbi:

- Legalább egy útvonal létezik, ahol  $p$  igaz lesz (Exists  $p$ , jele:  $E p$ )
- Minden útvonalon igaz lesz  $p$  (forAll  $p$ , jele:  $A p$ )

A CTL formulákban a temporális operátorok csak útvonalkvantorokkal együtt alkalmazhatók úgy, hogy utóbbiak mindig közvetlenül megelőzik az előbbieket. A fentiekből következően a CTL formális szintaxisa a következőképp adható meg:

- Minden P atomi kijelentés egy állapotkifejezés
- Ha p és q állapotkifejezés, akkor  $\neg p$  és  $p \wedge q$  is állapotkifejezések (tehát minden velük logikai műveletekkel felírt kifejezés is)
- Ha p útvonal-kifejezés, akkor E p és A p állapotkifejezések
- Ha p és q állapotkifejezések, akkor X p és p U q útvonal-kifejezések

A temporális logikák számos fontos tulajdonságot tudnak kifejezni. Kideríthető egy rendszerről, hogy eljuthat-e olyan állapotba, ahonnan nem képes továbblépni, azaz a rendszer holtponmentessége ellenőrizhető. Ellenőrizhető funkcionális helyesség, elevenség is, például elérhető-e egy adott elvárt állapot. A biztonsági feltételek teljesüléséről is meg lehet győződni annak vizsgálatával, hogy bekövetkezhet-e egy nem kívánt állapot. Jelen dolgozatban a modellellenőrzés során leggyakrabban (illetve a biztonsági feltételek tekintetében kizárólagosan) alkalmazott kifejezések „AG p” jellegűek (ld. 4. fejezet), tehát olyan követelmények, melyeknek bármely állapotból indulva minden állapotban igaznak kell lenniük.

### 2.2.6. Formális módszerek alkalmazása a vasúttechnikában

A vasúti biztosítóberendezések fejlesztése jelenleg három fő szabványon alapul. Az EN 50126 szabvány a vasúti alkalmazásban a megbízhatóság, az üzemkészség, a karbantarthatóság és a biztonság (angol nevük kezdőbetűiből röviden: RAMS) követelmények előírására és bizonyítására vonatkozik. Míg az EN 50128 szabvány a vasúti biztosítóberendezések szoftvereire, addig az EN 50129 szabvány azok hardvereire, illetve a teljes rendszerre vonatkozó előírásokat tartalmazza. Mindhárom előírás keretet ad a vasúti fejlesztésnek, meghatározza az adott rendszerek életciklusát, és az ezek során elkészítendő dokumentumok körét. A megfelelő dokumentáció bizonyítja a szabványnak való megfelelést, ami alapján a rendszer engedélyezése megtörténhet. A dokumentumok egyben azt is bizonyítják, hogy a fejlesztési folyamat az előírtak szerint történt, tehát a kialakított rendszer a vele szemben támasztott (elsősorban biztonsági) követelményeket teljesíti. A szabványok a fejlesztési folyamatot segítik, ezért előírásaik alkalmazása ajánlott.

Az EN 50129 különböző biztonságintegritási szinteket (Safety Integrity Level, SIL) definiál. [30] Az egyes szintek meghatározzák a rendszer elvárt biztonságintegritásához szükséges, a fejlesztés során figyelembe veendő szempontok körét. Öt szint küldöztethető meg, nullától a negyedik szintig terjedően. A nagyvasúti biztosítóberendezés funkciók fejlesztése a legmagasabb szinten, SIL 4-en történik.

A két szabványban (EN 50128, EN 50129) táblázatokban vannak összegyűjtve az életciklus különböző fázisaiban alkalmazandó technikák, melyek az adott szintek elérésének érdekében osztályozottak is. A szabványok a SIL 1-es és a SIL 2-es, valamint a SIL 3-as és a SIL 4-es szinteket közösen kezelik. Egy adott szint eléréséhez az egyes módszerek alkalmazása lehet:

- Kötelező (Mandatory, M)
- Nyomatékosan ajánlott (Highly Recommended, HR)
- Ajánlott (Recommended, R)
- Ajánlás nélkül (-)
- Ellenjavallt (Not Recommended, NR)

Nyomatékosan ajánlott előírások nem alkalmazása esetén megfelelő alternatív megoldást kell biztosítani. Az ajánlott technikák tetszés szerint alkalmazhatók, az ellenjavallt előírások használatát indokolni kell. Formális módszerek használatára mindkét szabvány tesz ajánlásokat. Az EN 50129 a rendszer illetve részrendszerek fejlesztéséhez ajánlja formális módszerek alkalmazását. [31]

Az EN50129 szabványnál konkrétabb ajánlásokat tesz a vasúti biztonságkritikus szoftverek fejlesztéséhez használható EN 50128 szabvány a saját hatókörében. [32] [33] Formális módszerek és a modellezés alkalmazása HR besorolású a szoftverekre vonatkozóan a követelményspecifikáció (13. ábra), az architektúra kialakítás valamint a tervezés és implementálás során. Modellezéshez a formális módszerek alkalmazása, tehát a formális modellezés szintén nyomatékosan ajánlott.

**Table A.2 – Software Requirements Specification (7.2)**

TECHNIQUE/MEASURE	Ref	...	SIL 3	SIL 4
1. Formal Methods (based on a mathematical approach)	D.28	...	HR	HR
2. Modelling	Table A.17	...	HR	HR

13. ábra Az EN 50128 szabvány egy részlete (forrás: [33])

A szabvány a D mellékletében (D.27, D.28, D.28.6, D.28.10, D.55 fejezetek) rövid leírással bemutatja a különböző alkalmazható technikákat, és rávilágít az életciklus mely fázisaiban alkalmazhatók. Külön megemlíti a véges állapotgépeket és a Petri hálókat, valamint a formális módszerekről szóló rész alatt temporális logikákat és a modellellenőrzést.

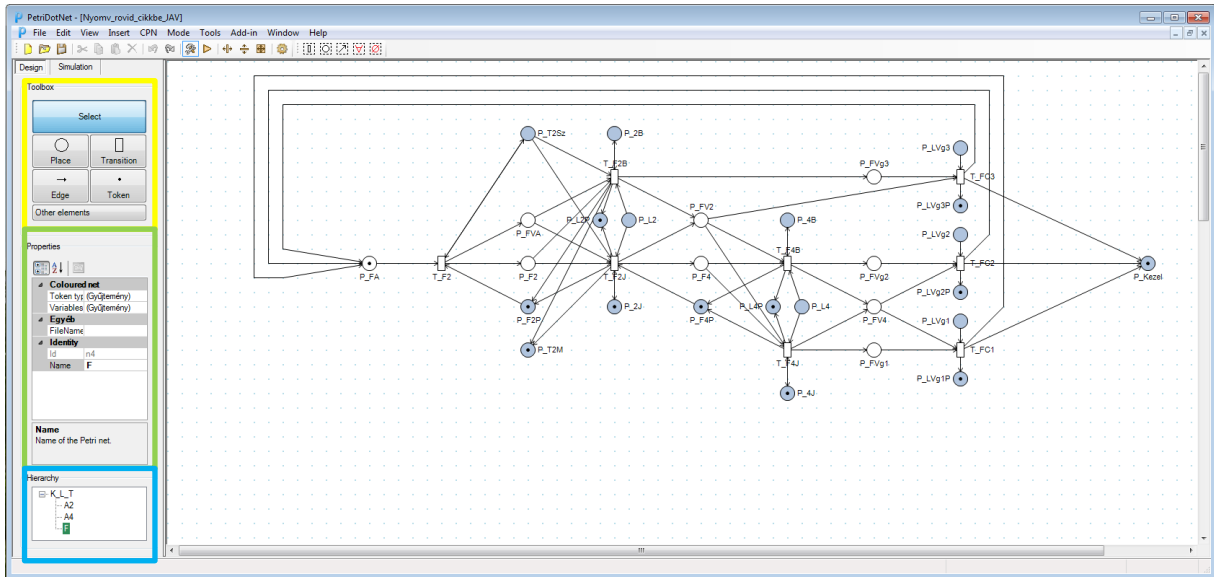
### 2.2.7. Modell/analízis eszközök

Jelen alfejezet célja röviden ismertetni a modellezés illetve a modellellenőrzés céljából megvizsgált, illetve igénybe vett eszközök háttérét, használatát. A Petri hálós modellezésre a PetriDotNet és a Snoopy/Charlie eszközöket vizsgáltam meg, míg automaták modellezésére az UPPAAL-t vettem igénybe.

#### **PetriDotNet**

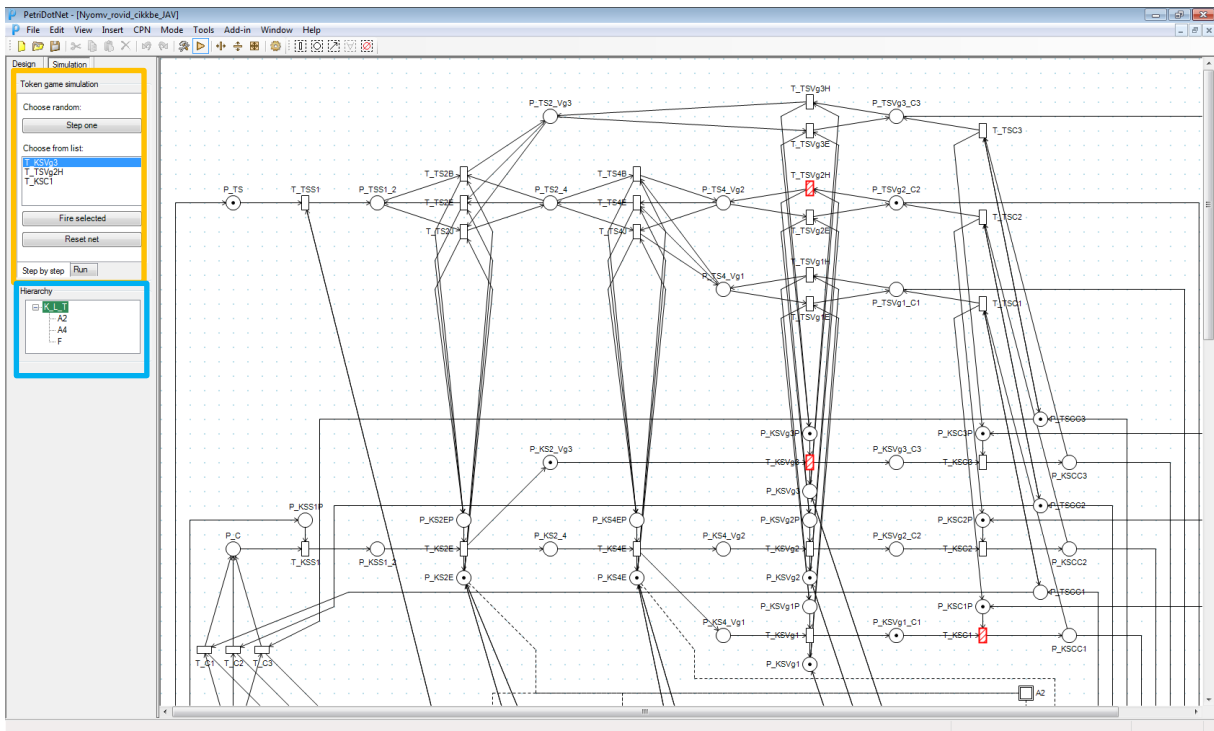
A Petri hálós modellezés elsősorban a PetriDotNet programmal támogatottan történt, ami a Petri hálók szerkesztésére, szimulációjára és analízisére szolgáló keretrendszer. A programot a Budapesti Műszaki és Gazdaságtudományi Egyetem Méréstechnika és Információs Rendszerek tanszékén fejlesztik elsősorban oktatási célból. [22] [23] [34] [35] A program letölthető a tanszék honlapjáról (<https://inf.mit.bme.hu/research/tools/petridotnet>).

A PetriDotNet eszköz modellező részét a 14. ábra mutatja. A helyek, tranzíciók, élek és tokenek egyszerűen helyezhetők el a bal oldalon található, sárga kerettel jelzett menürész segítségével. Innen választható ki az alhálók kezelésére szolgáló helyettesítő tranzíció (coarse transition). A főhálón a helyettesítő tranzícióba (szaggatott vonallal) kötött helyek az alhálóban is megjelennek, sötétkék kitöltő színnel ábrázolva (14. ábra). Az elemek tulajdonságai a bal oldalon, a zöld színnel jelzett toolbox alatt szerkeszthetők. E menüpont alatt van lehetőség a főháló és az egyes alhálók között választani (kék keretben).



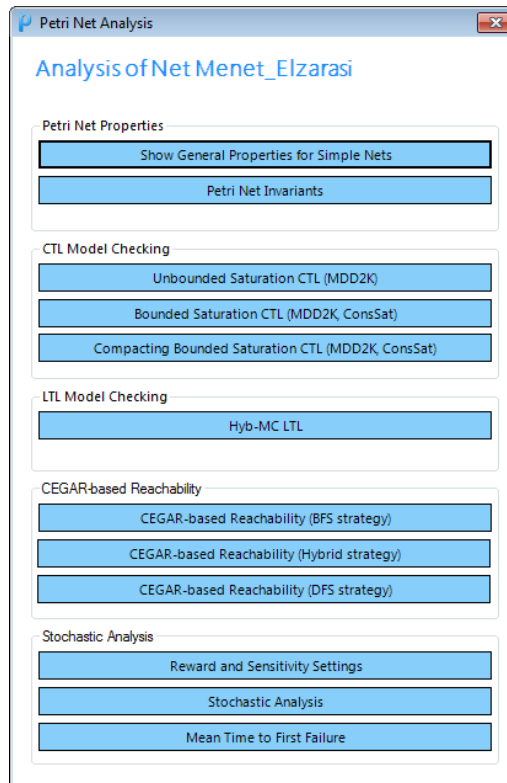
14. ábra A PetriDotNet modellezői felülete

Az elkészült modelleken a szimuláció a PetriDotNet eszköz szimuláció módba váltásával futtatható. (15. ábra) Az engedélyezett tranzíciók piros színnel jelöltek. A szimuláció történhet a tranzíciók egyenkénti eltüzelésével, a bal oldali narancsszínű kerettel jelzett szimulátor ablakban a megfelelő kiválasztásával és a „Step one” gombra való kattintással. A tranzíció úgy is képes tüzelni, ha a listában a nevére, vagy a modellen a piros szimbólumára történik a kettős kattintás. Lehetőség van a szimuláció beállítható időközű automatikus futtatására is. A hálókat bármikor visszaállíthatók a kezdőállapotba.



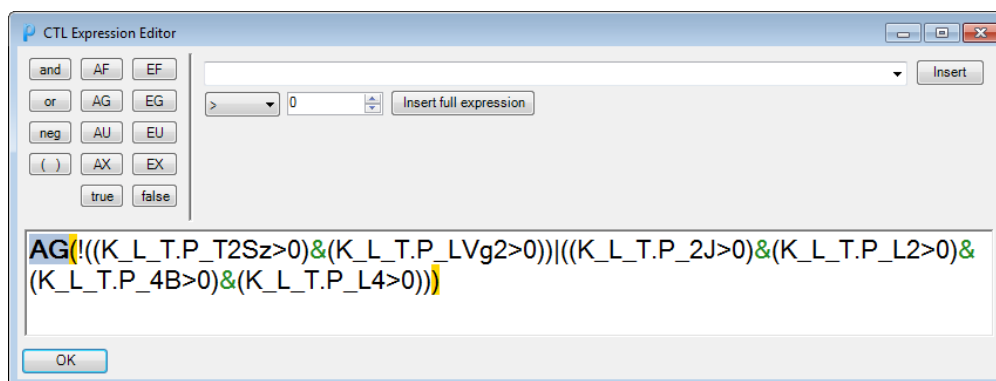
15. ábra A PetriDotNet szimulációs felülete

Modellellenőrzésre szerkesztés és szimuláció módban is lehetőség van. Az „Add-in” menü alatt a „NET ANALYSIS” menüpontot választva felugrik egy ablak, ahol a különböző vizsgálati módszerek közül lehet választani. (16. ábra)



16. ábra A PetriDotNet hálóanalízis felugró ablaka

A „Show General Properties for Simple Nets” gombra kattintva a Petri háló statikus és dinamikus tulajdonságai jelennek meg, többek között a korlátosság, holtpontmentesség, megfordíthatóság, stb. A CTL modellellenőrzéshez jelen dolgozatban az „Unbounded Saturation CTL (MDD2K)”, majd a „Manually”, ezután az „Every N” gombra kattintva került az állapotter legenerálásra. Ezután a „Go to CTL model checking” gombra kattintva érhető el a CTL kifejezés szerkesztő. (17. ábra)

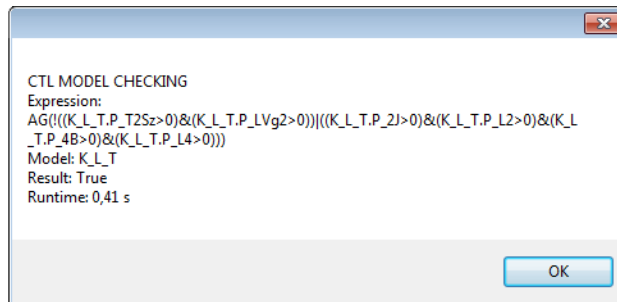


17. ábra A PetriDotNet CTL modellellenőrző kifejezésszerkesztője

A szerkesztőablakban bal oldalon megtalálhatók a szükséges logikai és CTL operátorok. A jobb oldali részen lehetőség van akár egész kifejezések kialakítására és egyben történő beillesztésére. A felső legördülő menüből a modell helyeit lehet kiválasztani, majd a kiválasztott hely tokenszámára vonatkozó összehasonlítások alakíthatók ki az alatta lévő ablak segítségével. Az ellenőrzés lefuttatásakor



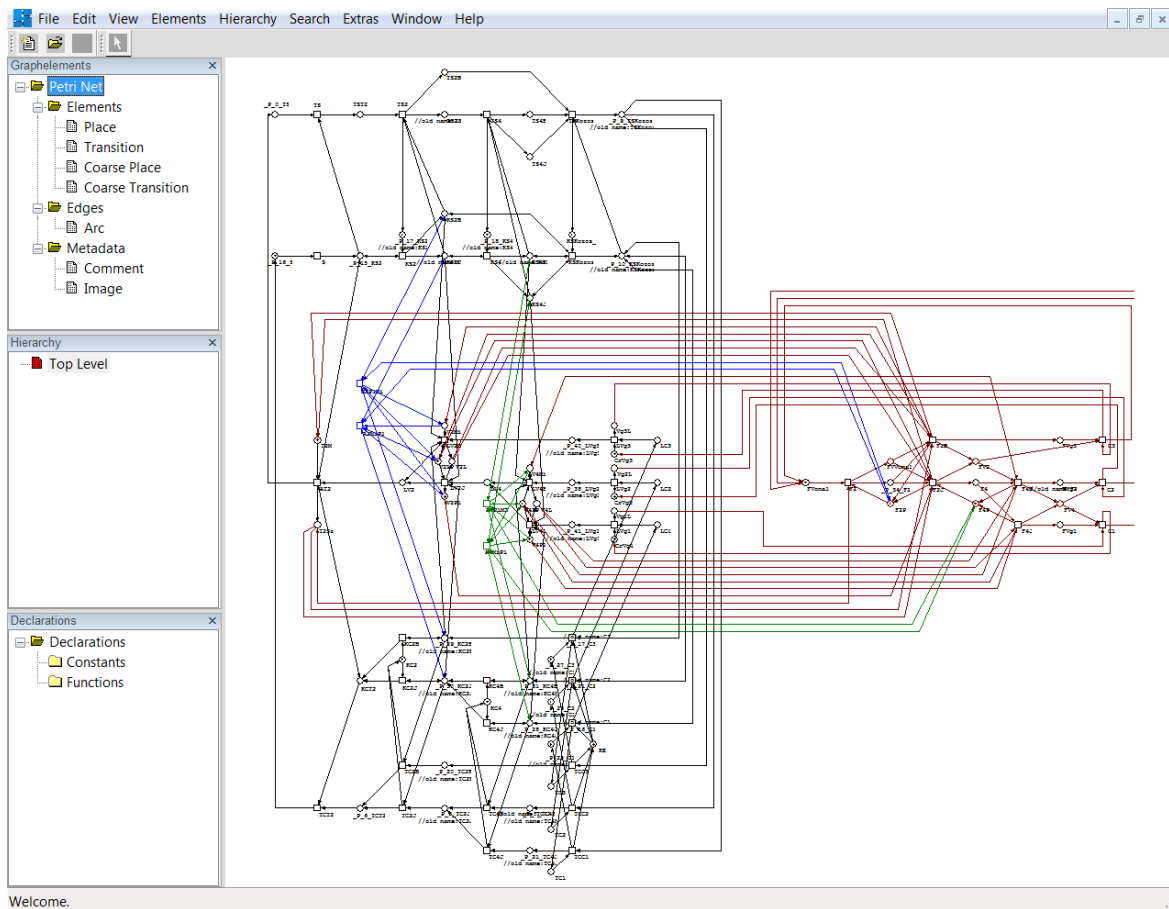
üzenetben jelenik meg a tulajdonság (nem) teljesülése. (18. ábra) Ezután újabb kifejezések bevitelére van lehetőség. Az alkalmazás hátránya, hogy nem tárolja a bevitt képleteket és nem generál ellenpéldát.



18. ábra PetriDotNet üzenet sikeres modellellenőrzési feltételről

### Snoopy/Charlie

A modellezési folyamat során a Snoopy/Charlie program párost is kipróbáltam. A Snoopy modellező és Charlie modellellenőrző eszközöket Cottbusban, a Brandenburgi Műszaki Egyetemen fejlesztették ki. [23] [36] [37] A Snoopy program különböző hierarchikus hálókat, köztük Petri hálókat tervezésére és szimulációjára használható. A verifikációra kifejlesztett Charlie a Petri hálókat statikus és dinamikus vizsgálatára, valamint CTL modellellenőrzésre is alkalmas. A modellezési feladat egy korai szakaszában felhasználásra kerültek a Snoopy és Charlie eszközök is (19. ábra), azonban a programok korlátai miatt a továbbiakban nem lettek alkalmazva.



19. ábra A nyomvonalas elvű biztosítóberendezés egy korai modellje a Snoopy eszközben

## UPPAAL

Az automaták alkalmazása során alternatív lehetőségek nem lett megvizsgálva, a modellezés az UPPAAL szoftver támogatásával történt. Az UPPAAL egy valós idejű rendszerek modellezésére, szimulációjára és verifikációra használható eszköz. [38] [39] [40] [41] [42] Kifejlesztése a svédországi Uppsalai Egyetem és a dániai Aalborgi Egyetem együttműködésében történt. A két város első három betűjéből áll össze az eszköz neve. Akadémiai alkalmazásokra ingyenes letölthető a honlapjáról ([www.uppaal.com](http://www.uppaal.com)).

Az eszköz olyan rendszerek leírására alkalmas, melyek modellezhetők nemdeterminisztikus, véges struktúrával, valós értékű órákkal, és csatornákon keresztül vagy közös változókkal történő kommunikációval. A tipikus alkalmazási területek közé tartoznak a valós idejű vezérlők és kommunikációs protokollok, különösen azok, ahol az időzítési szempontok kritikusak. Ezek alapján megállapítható, hogy vasúti biztosítóberendezések szimulációjára is felhasználható. Jelen modellezési feladatban az időzítések nem kerülnek használatra.

Az UPPAAL modell három fő részből épül fel, melyek a globális és lokális deklarációk („Declarations”), az automata sablonok („Automata templates”) és a rendszerdefiníció („System declaration”). Az UPPAAL deklarációi a C nyelvhez hasonlóak. A deklarációk között kezelhetők a következők: típusok, konstansok, változók, csatornák, órák, függvények, stb. Az automata sablonok az egyes folyamatok modelljeit és a hozzájuk tartozó lokális deklarációkat tartalmazzák. A rendszerdefiníció az automaták felsorolásából áll. Az azonos felépítésű automaták példányosíthatók a „Parameters” mezőben megadott, globálisan deklarált paraméterek segítségével.

Az állapotgépek két alapelemből, az állapotokból (korongok) illetve állapotátmenetektől (nyilak) épülnek fel. (20. ábra) Az állapotok névvel rendelkezhetnek, és egy automatában mindig csak egy lehet aktív (csak egy állapotban tartózkodhat egyszerre). Minden állapotgép rendelkezik egy kezdőállapottal, melynek jele dupla kör. Az állapotok közötti átmenetekhez feltételek, akciók kapcsolódnak. Egy adott állapotból több engedélyezett átmenet vezethet ki, ekkor a következő állapot kiválasztása véletlenszerűen történik. Az állapotátmenetekhez négy különféle folyamat kapcsolódhat, melyek (kiértékelésük sorrendjében) a következők:

- Véletlenszerű érték sorsolása egy megadott intervallumban („Select”, sárga)
- Őrfeltételek vizsgálata („Guard”, zöld)
- Szinkronizáció automaták között („Sync”, kék)
- Értékdadás („Update”, lila)



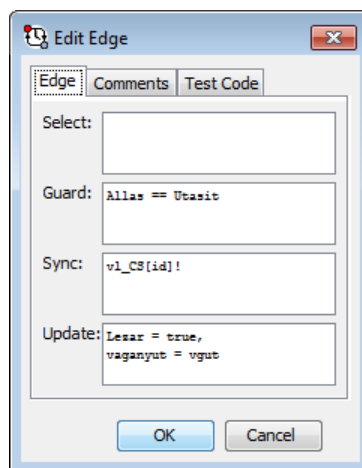
20. ábra Az UPPAAL modellezői felülete

A négy folyamat az átmenetet jelölő élekre kattintva elérhető élszerkesztőben szerkeszthető. (21. ábra) A „Select” parancs megadása „változó : intervallum” formátumban történik. Az átmenet során a csak itt elérhető változóba kerül az intervallum egy véletlenszerűen választott eleme.

Az őrfeltételek logikai kifejezések, melyek kiértékelése mellékhatásmentesen történik órákra, egész változókra és konstansokra, illetve ezek tömbjeinek elemeire. Az állapotátmenet csak a kifejezés true értékre történő kiértékelése esetén mehet végbe.

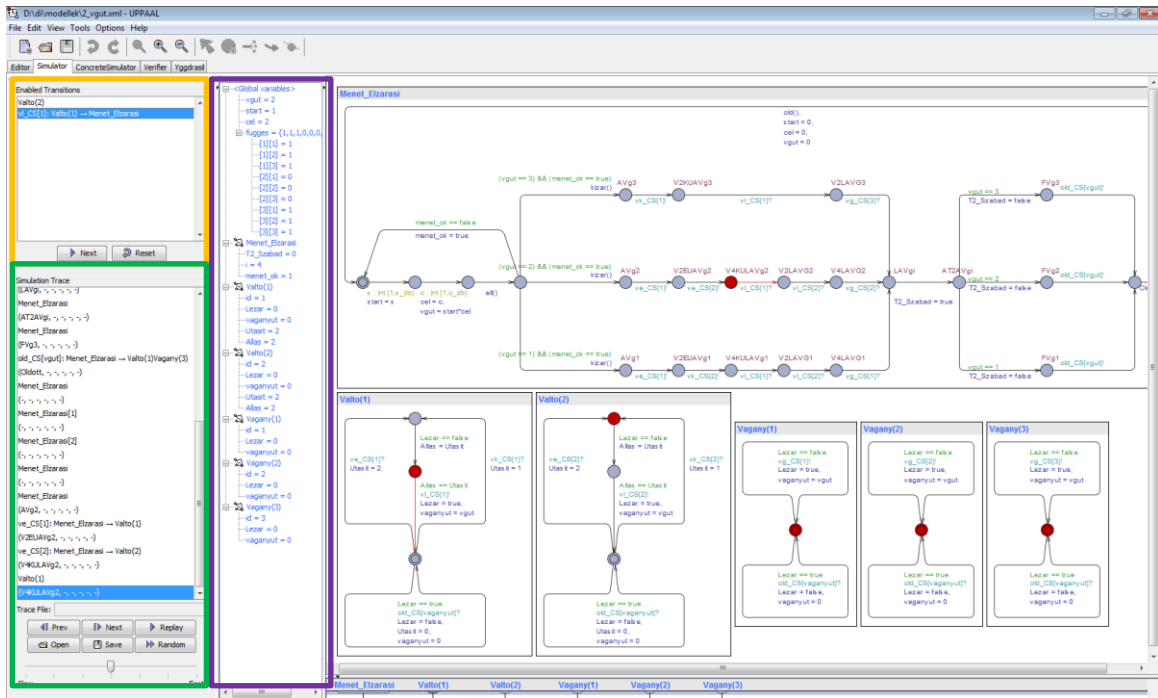
A szinkronizáció a globálisan definiált kétféle csatorna segítségével történhet. Egyszerű szinkronizáció esetén egy előre definiált csatornán („chan”) egy küldő („Kifejezés!” formában) kezdeményezésére egy fogadó („Kifejezés?” formában) egyszerre történő állapotátmenete megy végbe. Mindig csak két fél között mehet végbe, ha mindkét fél összes vonatkozó feltétele teljesül, fogadó nélkül az átmenet blokkolt. Üzenetszórásos szinkronizáció broadcast csatornán („broadcast chan”) történik. Egy küldője, de több fogadója is lehet. A küldő állapotátmenete akkor is végrehajtható, ha nincs fogadó fél.

Az értékadások mellékhatásosak, és szintén csak órákra, egész változókra és konstansokra, illetve ezek tömbjeinek elemeire vonatkozhatnak, illetve meghívhatnak függvényeket is. Beállítható az egyes állapotokhoz sürgősség vagy elkötelezettség, illetve további feltételek is, ezek a modellekben nem kerülnek alkalmazásra.



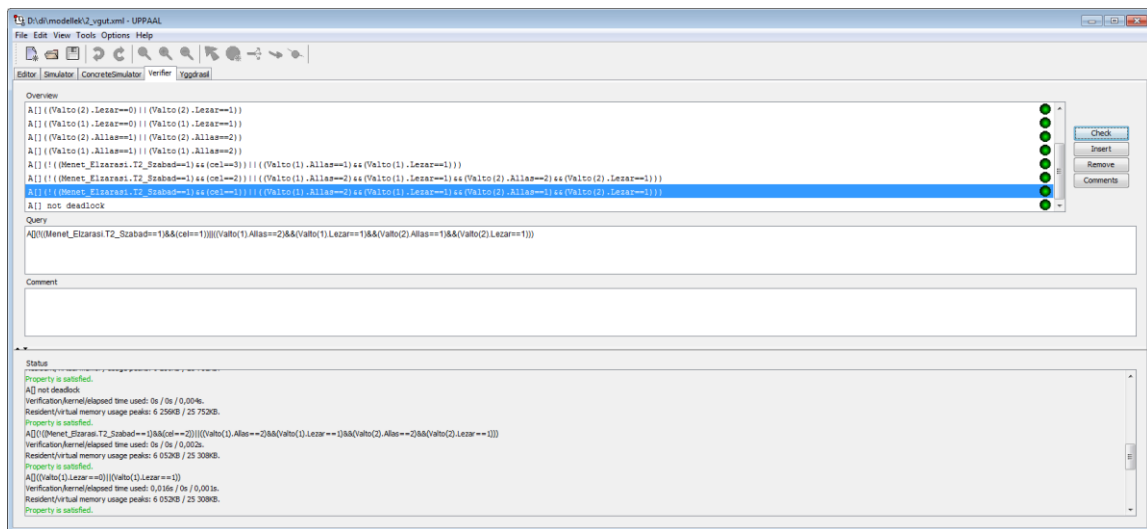
21. ábra Az UPPAAL élszerkesztője

Az elkészült modellek működése szimulálható a szimulátorban (22. ábra). Itt látható az automaták (már példányosított) felsorolása, melyek a pirossal jelölt állapotukban találhatók. Bal oldalon, a narancsszínű kerettel jelzett helyen vannak felsorolva az engedélyezett átmenetek, melyek közül az aktuálisan kiválasztott a modelleken piros színnel látható. A lépésenként történő szimuláció során középtűt, a lila kerettel jelzett helyen megfigyelhető az összes változó értéke. A végrehajtott átmeneteket a szimulátor a zöld kerettel jelzett részen tárolja.



22. ábra Az UPPAAL beépített szimulátora

Az UPPAAL ellenőrző része („Verifier”) lehetőséget ad az elkészült modellek a CTL nyelv egy megszorított változatával történő ellenőrzésre. (23. ábra) Az ellenőrizendő formula megadása csak operátor + logikai kifejezés formában történhet, tehát több operátor együttes használatára nincs lehetőség. A logikai műveleti jelek kettőzve írandók (pl.: &&, ||, ==). Jelölésbeli különbség, hogy a G temporális operátor megadása [], míg az F temporális operátoré <> jelöléssel történik. Bizonyos esetekben alkalmazható a „leads to” ( $p \rightarrow q$ ) formula, melynek jelentése, hogy minden úton, ha p teljesül, valamikor q is teljesülni fog.



23. ábra Az UPPAAL modellellenőrző felülete

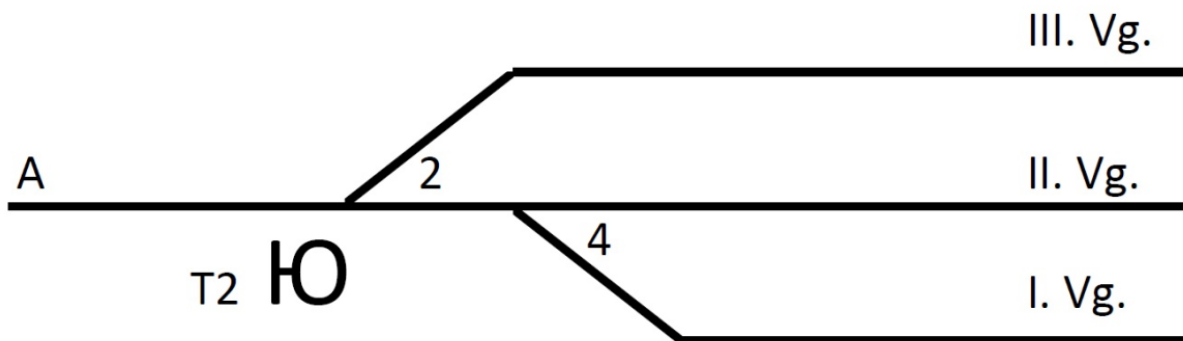
A verifikáció céljából felvett kifejezéseket a program elmenti. Az ellenőrzés lefuttatásakor a kifejezés mögötti kör színe jelzi az eredményt (zöld – teljesült, piros – nem teljesült, szürke – még nem ellenőrzött/az ellenőrzés sikertelen). Nem teljesült követelmény esetén a program ellenpéldát is generál, mely a szimuláció fül alatt végigkövethető.

### 3. Modellezés

A modellezés elsődleges célja eltérő formalizmusok (Petri hálók és automaták) vasúti alkalmazásának vizsgálata, illetve összehasonlítása. A két fő eltérő biztosítóberendezési (vágányutas és nyomvonalas) elv e két különböző megközelítésű formális módszer segítségével került leképezésre. A Petri hálós modellek a hazai fejlesztésű PetriDotNet program segítségével készültek el, míg az időzített automatás megközelítés az UPPAAL szoftverben lett kifejlesztve. A fejezet fő tartalmi részét az elkészült négy modell ismertetése teszi ki. Ezeket megelőzően bemutatásra kerül az a topológia, mely alapján az elvek modellezése történt. Mivel a modellezés kivitelezése szükségszerűen absztrakcióval jár, felsorolásra kerülnek a megtett egyszerűsítések is.

#### 3.1. Esettanulmány leírása

A jelen dolgozatban elkészített modellek alapját a BSc. képzésben Vasúti információs és kommunikációs rendszerek I. tárgy házi feladatában feladott modellezési példa képezi. Ennek során egy háromvágányos fejállomásra (24. ábra) kellett nyomvonal elves vágányút-kijelölést, -beállítást és -lezárást, majd jelző szabadra állást modellezni. A feladat elsődleges célja a modellezés és szimuláció volt. Az elkészített modell a szimuláció során a funkciók végrehajtását követően deadlock állapotba került. Így modellellenőrzésre sem volt alkalmas. Később kis mértékben átdolgozásra került, és a Vezetékek Világa című vasúttechnikai lapban egy Petri hálókról szóló cikk példamodelleként jelent meg. [43] [44]



24. ábra A modellezés alapjául szolgáló helyszínrajz

A helyszínrajz egy háromvágányos állomásrész kezdőponti (páros) váltóközetét és a hozzá tartozó T2 jelű (másképp: E2) jelű biztosított tolatásjelzőjét jeleníti meg. Az állomásrészben szereplő objektumok felsorolását a 3. táblázat tartalmazza.

3. táblázat Az állomásrész objektumai

Objektum neve	Objektum leírása
A	a tolatásjelző előtti szakasz ((nyílt vonal))
T2	T2 jelű tolatásjelző
2	2-es számú váltó
4	4-es számú váltó
I. Vg.	az állomás első vágánya
II. Vg.	az állomás második vágánya
III. Vg.	az állomás harmadik vágánya

A feladatban tehát az „A” szakasról induló, a T2 tolatásjelző által biztosított, a három vágány egyikére beállítható tolatóvágányutak modellezése történt meg. Az „A” jelű szakasz nem része a modellnek, csupán a csatlakozó vonal azonosítására szolgál (a vágányutas elvű modellekben). Az alfejezet elején leírt funkciókör bővítésre került a vonat szimulálásával, és hozzá kapcsolódóan a vágányút feloldódásával. Ezzel lehetőséget teremtve új vágányút beállítására, és így a funkciók ismétlődésére, azaz a modellek az elvárás szerint holtpontmentesek. Ezek alapján a modellezni kívánt funkciók:

- Vágányút kijelölése
- Váltók átállítása és lezárása, vágányszakaszok lezárása
- Vágányút lezárása
- A lezárt vágányút kijelölésének törlése
- Jelző szabadra állítás
- Tolató szerelvény közlekedése
- A vágányút oldódása, és lehetőség új vágányút beállítására

Jelen dolgozatban a megadott topológiára készült el a két legjelentősebb biztosítóberendezés konstrukciós (vágányutas és nyomvonalas) elv modellje, az egyes elvek tulajdonságainak figyelembevételével.

### **3.2. Modellezési döntések: egyszerűsítések, elhanyagolások**

Modellezés során szükséges meghatározni a rendszerhatárokat, illetve különböző egyszerűsítéseket megtenni annak érdekében, hogy a vizsgálat tárgyát képező rendszer önmagában minél pontosabban elemezhető legyen. A dolgozatban kijelölt feladat, illetve helyszínrajz is magával von különböző egyszerűsítéseket, illetve elhanyagolásokat. Ezek a következők:

- Csak egy irányba állíthatók be vágányutak, „A” felől a három vágány egyikére. Az ellenkező irányba való közlekedésre lehetőség nincs.
- Az állomásnak nincsen „másik vége”, ezért valódi céljelzők helyett virtuális célok szerepelnek a példában. Emiatt megcsúszásokat sem kell figyelembe venni.
- Oldalvédelmet adó objektumok (védőváltók, kisiklasztó saruk, vágányzáró sorompók illetve háttal álló jelzők) nem szerepelnek, így az oldalvédelemmel a példában nem kell foglalkozni.
- Tolatóvágányút beállítása esetén a szakaszok foglaltsági információit nem kell felhasználni a vágányút lezárásához (a váltók állíthatóságához viszont igen).
- A váltók állítása csak vágányutasan történhet (egyéni állításra nincs lehetőség).
- Közúti sorompót nem kell modellezni.

Ezek mellett történtek nem a helyszínrajzhoz kötődő elhanyagolások, megkötések is, melyek a következők:

- Egyszerre csak egy vágányút állítható be, annak jobb vizsgálhatósága érdekében (nincs vágányúttárolási lehetőség).
- Az idő nincs kezelve, csak az egyes állapotok kerülnek figyelembevételre (pl. a váltó csak a két végállásban lehet, ezek között a váltás pillanatszerű, stb.).
- A vágányút lezárása és ellenőrzése egy lépésben történik
- A tolatómenet leközlekedésének szimulálása rövid szerelvényel történik.
- Különböző meghibásodásokat, pl. hamis foglaltságokat nem kell modellezni.

A modellezés és a modellezendő funkciók lehatárolásának elsődleges célja az volt, hogy egy vágányúthoz tartozó valamennyi működés a lehető legegyszerűbben bemutatásra kerülhessen. Lehetőség szerint a négy modell azonos funkciókörrel rendelkezzen a két elv tulajdonságainak minél jobb megtartása mellett. Ennek érdekében a 3.1. fejezetben bemutatott helyszínrajzot megtartottam, és ez alapján tettem megkötéseket.

A csak egy irányba beállítható vágányutak nem jelentenek jelentősebb megszorítást a modellezendő rendszerrel kapcsolatban, mivel az ellenkező irány elhagyása nem jelent funkcióbeli elhagyást a biztosítóberendezésre nézve. A vágányút célpontjai virtuális célokként kerültek leképezésre. Mivel az alapul vett helyszínrajzon nincsenek oldalvédelmet adni tudó elemek, így az oldalvédelemmel nem foglalkoztam. Az a döntés, hogy vonatmenet helyett tolatómenet beállítása kerül modellezésre hordozza az előnyt, hogy a vágányút lezárásához az elemek foglaltságát nem kell figyelembe venni. Az oldásban viszont szerepet játszanak a foglaltsági szakaszok.

A modellekben csak egyféle kezelés létezik, ez a start és cél kijelölése. Minden működés a vágányút beállításához kötődik. Egyéni kezelések nincsenek modellezve. Egyszerre csak egy vágányút beállítására van lehetőség. Ennek oka, hogy modellellenőrzés során az egy vágányúthoz tartozó feltételek teljesülését ellenőrizni lehessen. (Pl.: Ha a vonat elhaladt, akkor biztosan feloldódott minden elem?)

A modellek nem időzítettek, csak az elemek lehetséges állapotai lettek figyelembe véve. Ennek oka, hogy a működés során az idő nem játszik kritikus szerepet, elegendő csupán az események bekövetkezését vagy be nem következését vizsgálni. A 2.1.2. és a 2.1.4. fejezetekben említettek szerint a berendezések egyes feltételek ellenőrzését a vágányút beállítás során többször, illetve másokét a jelző szabadra állítása után folyamatosan ellenőriznek. Mivel a modellezés célja elsősorban a működés leírása volt, így a modellekben ezek az ellenőrzések elmaradnak, a vágányút lezárása és rögzítése egy lépésben történik. Részben ebből kifolyólag meghibásodással, hibás működéssel a modellek nem számolnak.

A vágányutak feloldását a közlekedő szerelvények végzik, melyek a valós vasúti rendszerben bizonyos határok között bármilyen hosszúak lehetnek, és egyszerre több foglaltságérzékelési szakaszt is elfoglalhatnak. Vágányutas elv esetén csak a feloldó szakaszokat modelleztem, ezért ezekben az esetekben a szerelvények hossza nem releváns. A szakasz foglalttá válásával áll a jelző továbbhaladást tiltó állásba, és felszabadulásával oldódik a vágányút. Nyomvonalas elv esetén a vasúti járműveket minimális hosszúságúnak tekintettem. A Petri hálós modellben ez egységnyi hosszúságot (egy tokent) jelent, azaz mindig csak egy szakasz foglalt, és így halad végig a tolatómenet a starttól a célig. Az UPPAAL modell esetén a szerelvény a legrövidebb szakasznál rövidebb, azaz egyszerre legfeljebb két szakaszt foglal el. A különbség az eszközök sajátosságából adódott, működésbeli különbséget a modellekre nézve nem jelent.

Egy vágányút beállítása, lezárása és feloldódása után a modellekben újabb vágányutak beállítására van lehetőség. Ez nem minden esetben jelenti a kezdőállapotba való visszajutást. Pontosabban megfogalmazva, ez csak a váltóállás esetében nem jelenti feltétlenül az alapállás újbóli felvételét. A váltók állásánál a kezdeti állapot csak a modellek elindulásához szükséges, a két végállás között nincs biztonsági különbség. A modellekben a váltóállások kezdeti állapota a + jelű, azaz az egyenes/jobbra terelő állás. Valamennyi itt megemlített egyszerűsítés és elhanyagolás a későbbiekben feloldható, illetve a modellek tovább bővíthetők (ld. 0. fejezet).

A modellek képei az 1-4. Mellékletekben láthatók, a futtatható modellek pedig letölthetők a <http://dportal.kozlek.bme.hu/> oldalról.

### Vágányutas elv

A vágányutas elv modellezéséhez szükséges, a topológiának megfelelő menetterv a 4. táblázatban, az elzárási terv az 5. táblázatban látható. A modellekben a modellezett funkciók lefutása során a kezdő-állapotból indulva először megtörténik a beállítandó vágányút véletlenszerű kiválasztása. Ekkor történik meg a menettervi függőségek ellenőrzése és beállítása is. A modellekben az állapotváltozások sorozata inentől az elzárási terv leképezésének megfelelően a beállítható három vágányút szerint ágazik szét. Ezeken az ágakon végrehajtódik a menetbeállítási parancsok kiadása, a váltók állítása és az egyes objektumok egyenkénti, majd a vágányút teljes lezáródása. A lezárást követően a T2 jelű tolatásjelző szabadra áll, és a tolatómenet leközeledhet. A vágányút feloldását a váltókörzet után, a fogadóvágányokhoz tartozó oldószakaszok végzik. A modellekben mindig csak a beállított vágányúthoz tartozó szakasz érzékelheti a járműveket. Ha az oldószakasz foglalt lett, majd felszabadult, a vágányút lezárása és az elemek feloldódnak, és új menet beállítása válik lehetségessé.

4. táblázat Az állomásrész menetterve

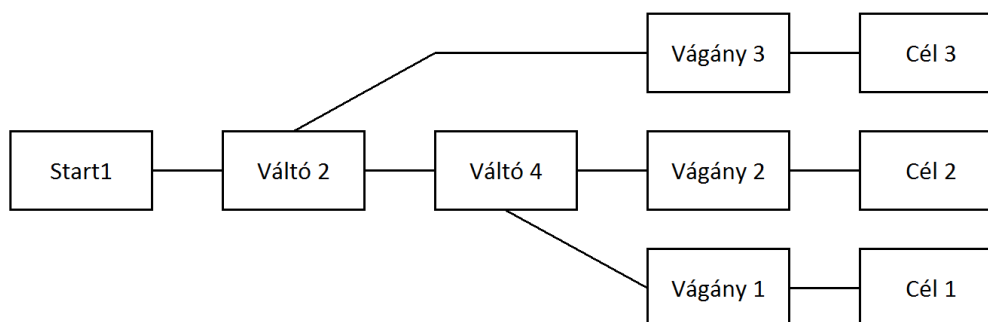
	A-I. Vg.	A-II. Vg.	A-III. Vg.
A-I. Vg.	I.		
A-II. Vg.		II.	
A-III. Vg.			III.

5. táblázat Az állomásrész elzárási terve

Menetek	Beállítandó elemek		Állítható jelző	Jelzésekép
	2	4		
A-Vg. I.	-/K		T2	F ⊙
A-Vg. II.	+/E	+/E	T2	F ⊙
A-Vg. III.	+/E	-/K	T2	F ⊙

### Nyomvonalas elv

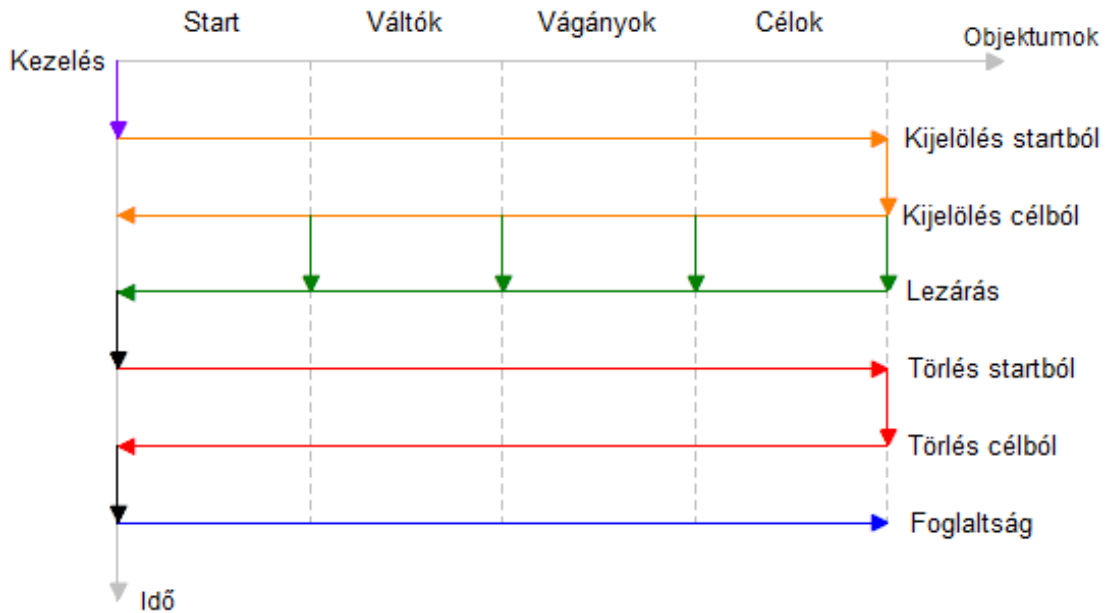
A nyomvonalas elv megvalósításához felépíthető az egyes vágányúti elemek kapcsolatrendszere, melyek között a kommunikáció zajlik, és melyek a hozzájuk rendelt vágányúti funkciókat önmagukban képesek ellátni. Az egyes objektumok neve az őket szimbolizáló téglalapokban látható a helyszínrajznak megfelelő elrendezésben (25. ábra).



25. ábra A vágányúti elemek geografikus kapcsolatrendszere



A modellek működése a vágányutas modellekhez hasonlóan a start- és célpontok véletlenszerű kiválasztásával történik. (ld. 26. ábra, Az egyes funkciók színezése megegyezik a később alkalmazottakéval.) Ezután egy kijelölő lánc fut végig a startból a topológiának megfelelően összekapcsolt elemeken. A kijelölés a célba elérve visszafordul, és a folyamat megtörténik a start irányába is. Ha egy váltó mindkét irányból megkapta a kijelölést, akkor a kívánt állásba állhat. Ha egy elem jól áll, és mindkét irányból ki van jelölve, lezáródhat. Ezután a célból a start felé végigfut egy lezáró lánc.



26. ábra A vágányúti funkciók időbeli lefutása

Ha a vágányút le van zárva, akkor a kijelölésre már nincs szükség, így az eltörölhető. A kijelölés törlése szintén start → cél, majd cél → start irányban fut le. Ha vágányút kijelölés megszűnt, a T2 jelző szabadra állítható. Ekkor a tolatómenet a beállított vágányúton leközeledhet. A menet a start felől indul és a célig haladva minden elemet érint, és egyenként felold. A vágányút teljes feloldása után újabb menetek beállítására van lehetőség. Az elemek jobb kezelhetősége érdekében a hivatkozás rájuk a 6. táblázatnak megfelelően a rövidítéseikkel, és „i” indexeléssel történik.

6. táblázat Az egyes vágányúti elemek szokásos jelölése a modellekben

Objektum neve	Szokásos jelölése
Start	Si
Váltó	i
Vágány	Vgi
Cél	Ci

### 3.3. Modellek bemutatása

Jelen alfejezetben kerül bemutatásra a négy modell. Az egyes modellek ismertetése nem követi az alapul vett biztosítóberendezés szerkesztési elvek megjelenésének időbeliségét (és az eddigi ismertetésük sorrendjét), hanem a modellek elkészülte alapján történik. Először a PetriDotNetben, majd az UPPAAL-ban készült modellek leírásai kerülnek ismertetésre. A modellek bemutatása a struktúra ismertetése után a vágányút-beállítás folyamatát követi.

### 3.3.1. Nyomvonalas elv modellezése Petri hálóval a PetriDotNet eszközben

A modell a funkciók öt nagyobb csoportra osztása szerint öt fő részből áll, melyek a működésük sorrendje szerint a következők:

1. Vágányút kijelölés
2. Váltóállítás
3. Vágányútlezárás (és jelzőállítás)
4. A korábban kijelölt és már beállított vágányút kijelölésének törlése
5. Tolatómenet közlekedése (foglaltság szimulálása) és a lezárt vágányút feloldása

A jobb áttekinthetőség érdekében a modell fő- és alhálókból áll. A folyamatok lefutásának jobb követhetősége érdekében azonban nem minden objektum lett külön alhálóba szervezve, hanem csak egyes funkciók. A felosztás a következőképp történt:

- Főháló: kijelölés, lezárás, kijelölés törlése
- Alháló 1: 2-es váltó állítás
- Alháló 2: 4-es váltó állítás
- Alháló 3: foglaltság és oldás

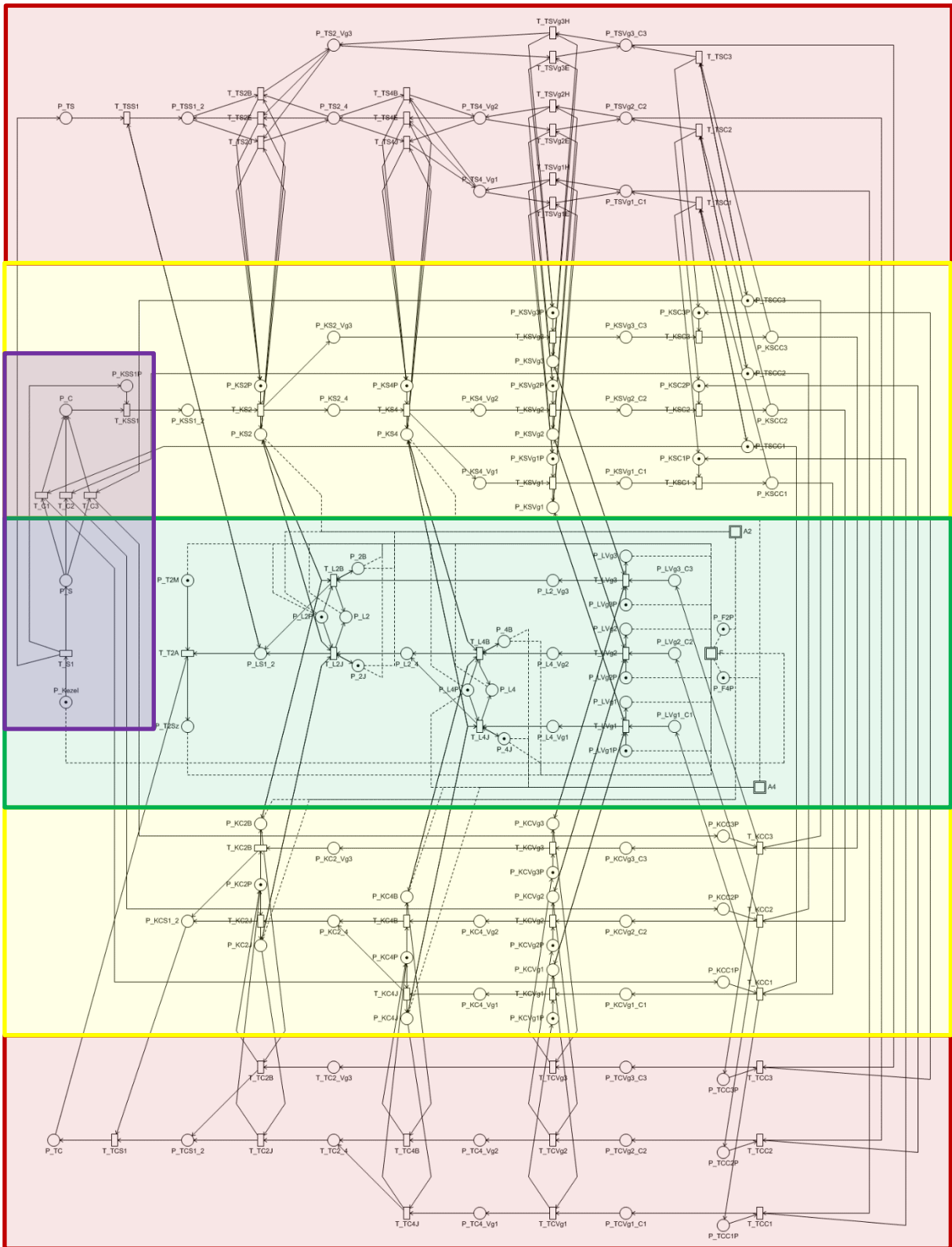
A főháló egyszerre több funkciót modellez, ezért több elérő hálórészből épül fel:

- Start és cél kiválasztása
- Vágányútkijelölés a startból a cél felé
- Vágányútkijelölés a célból a start felé
- Vágányútlezárás (célból a start felé) és jelzőállítás
- Kijelölés törlése a startból a cél felé
- Kijelölés törlése a célból a start felé

Az egyes funkciók, folyamatok az objektumokból és azok a helyszínrajznak megfelelő kapcsolatából összeállított hálórészekben futnak le. Így összesen hat hálórész lett összeállítva a topológiának megfelelően (az előbbi felsorolás utolsó öt tétele, illetve a foglaltság/oldás). A start- és célpont kiválasztása külön hálórészen történik, a topológiától függetlenül. A 27. ábra jeleníti meg a főháló funkciók szerint színezett képét. A sárga részek jelölik a vágányút kijelölését végző hálórészeket, felül a start, alatta a cél felől végfutó kijelöléssel. A középső zöld területen történik a vágányút lezárása, míg a piros területek végzik a kijelölés törlését a kijelölésnek megfelelő irányból. Oldalt, a kis lilával jelölt hálórészen történik egy beállítandó vágányút start- és célpontjának kiválasztása. A 28. ábra a főháló egy más felosztás szerinti színezését szemlélteti. Az egyes színek a különböző vágányúti elemeknek felelnek meg a 7. táblázat szerint. A teljes nyomvonalas elvű Petri hálós modell megtalálható az 1. Mellékletben.

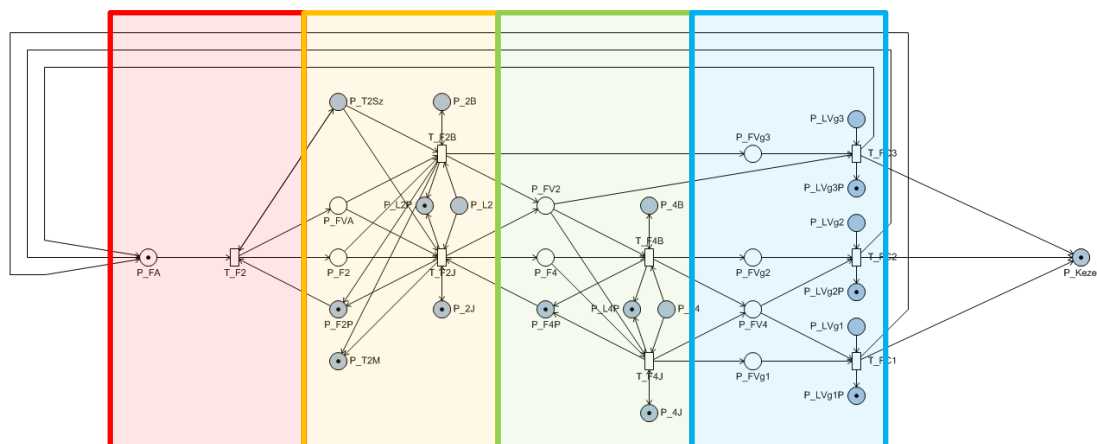
7. táblázat A nyomvonalas elvű Petri hálós modell egyes objektumainak színezése

Szín	Tartalom
Piros	Start objektum, startjelző
Sárga	2-es váltó
Zöld	4-es váltó
Kék	I – III. vágány
Lila	Cél objektum



27. ábra A nyomvonal elvű Petri hálós modell főhálójának funkciók szerint színezett képe





29. ábra A nyomvonal elvű Petri hálós modell foglaltság hálójának objektumok szerint színezett képe

Az egyes elemek (tranzíciók és helyek) elnevezése a következő logikát követi. A helyek (Place) neve P\_ kezdettel, a tranzíciók (Transition) T\_ előtaggal írónak. Az ez után következő betűsor az elem elhelyezkedésére, funkciójára utal (8. táblázat).

8. táblázat Az elemnevek első betűjének magyarázata a nyomvonalas modellben

Funkció	Rövidítés
Vágányút kijelölés	K
Lezárás	L
Vágányút kijelölés törlés	T
Foglaltság	F

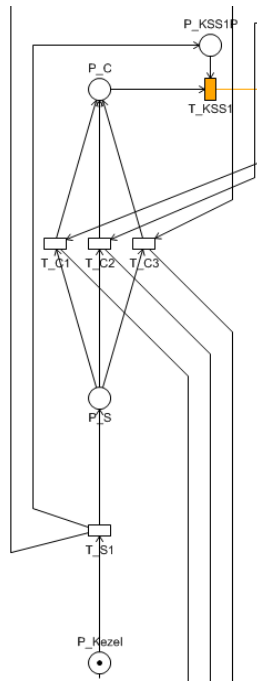
Kijelölésnél és törlésnél a második betűk jelzik, hogy az adott funkció melyik irányból működik (S - Startból, C - Célból). Ezeket követik az adott objektum neve (a 9. táblázatnak megfelelően), illetve állapota, elhelyezkedése vagy funkciója. Az egyes (fő- és al)hálók nevei is a megvalósított funkciót jelölik. Bizonyos helyek a különböző objektumok vágányúti funkciókenti összekapcsolását jelenítik meg (a jelfogóegységek nyomkábellel való összekötéséhez hasonlóan). Ezek neve a két szomszédos objektum nevében végződik. A startból célba elért funkciók visszafordítását a start felé „SC” taggal közbeiktatott nevű elemek (helyek és tranzíciók) végzik. A „P” betűre végződő „pót” helyek jellemzően azok, amelyek egy-egy vágányúti funkció legfeljebb egyszeres lefutását biztosítják, tehát kapacitáskorlátként szolgálnak. Ugyanakkor az ilyen jelölésű helyek biztosítják bizonyos információk (objektum kijelölve/lezárva/foglalt) negáltját. A leírás során előfordulnak az egyes elemekre (helyek és tranzíciók) vonatkozó paraméteres utalások. Az „i” és „j” betűk esetében a megfelelő szám, „x” esetében a megfelelő objektum vagy funkció (pl. Vg2) értendő.

9. táblázat Az elemnevekben szereplő további betűk magyarázata

Jelölt	Jelölés
Váltó jobb szára	J
Váltó bal szára	B
Váltó eleje	E
Állítás (jelző vagy váltó)	A
Tilos a tolatás („Megállj!”)	M
Szabad a tolatás	Sz
Az „objektum foglalt volt” információ	FV
„Pót” hely	P

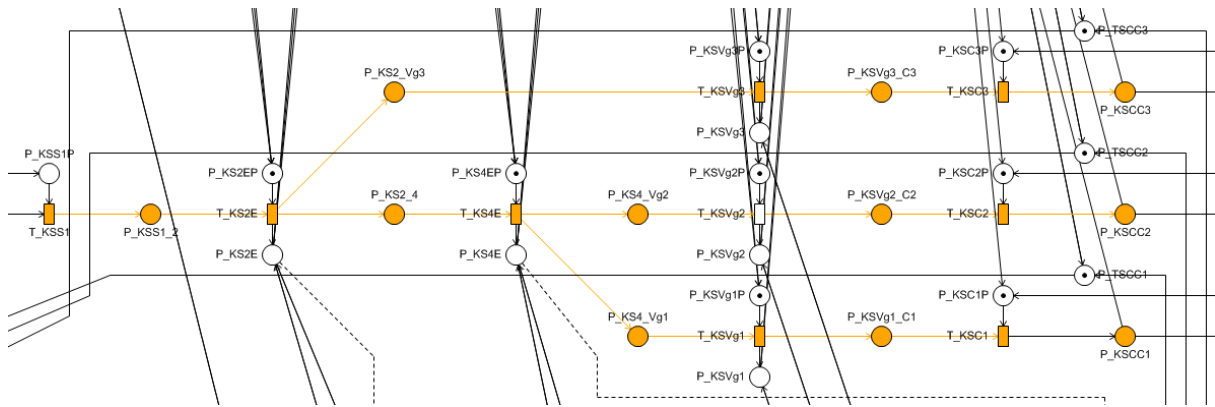
### A vágányút-kijelölés folyamata

Az egyes ábrákon az eddig alkalmazott funkció szerinti színezéssel ki vannak emelve a topológia alakját lekövető hálórészek, melyek az elemek nyomvonalas kapcsolatának felelnek meg. A modell működése a főhálón kezdődik, a „P\_Kezel” hely kapacitásának felhasználásával. (30. ábra) A „P\_Kezel” és „P\_S” helyek közötti tranzíciókkal lehet a startok közül választani. Jelen példában egy start van, így a „T\_S1” tranzíció tüzel. A tüzelés előkészíti a vágányút start felőli kijelölését azáltal, hogy tokenet juttat a „P\_KSS1P” helyre, valamint a kijelölés start felőli törlését, azáltal, hogy tokenet juttat a „P\_TS” helyre. A vágányút kijelölés a cél kiválasztásával folytatódik, melyet a „T\_C1”, „T\_C2” vagy „T\_C3” egyike végez el. A cél kijelölése a megfelelő „P\_KCCiP” helyen tárolódik. (32. ábra) Emellett előkészül a vágányút kijelölésének cél felőli törlése is, a megfelelő „P\_TSCCi” helyről való token elvételével. (31. ábra)



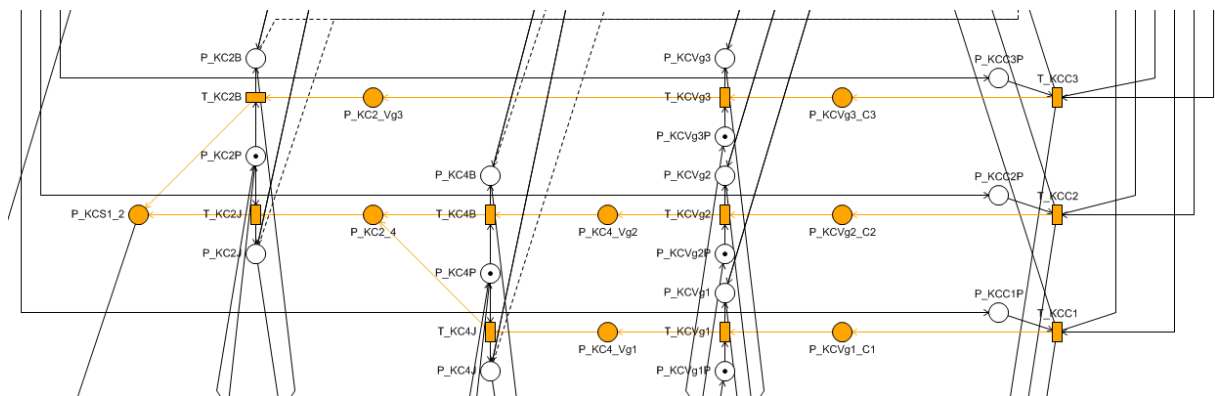
30. ábra A start- és célpontok kiválasztása a nyomvonalas elv Petri hálós modelljében

A „T\_KSS1” tüzelésével megkezdődik a vágányút kijelölése, mely csúccsal szemben érintett váltónál ketté ágazik, míg gyök felől érintetteknel levágja a másik ágon érkező kijelölés (keresés) lehetőségét. (31. ábra és 32. ábra) Ez úgy valósul meg, hogy csúcs felől érkező kijelölés tranzícióban („T\_KSiE”) ágazik el, illetve gyök felől jövő keresések hely elemekben találkoznak („T\_KC2\_4” és „T\_KCS1\_2”). Egy elem kijelölése úgy történik, hogy a kijelölést végző tranzíció a kijelölés továbbadása mellett az adott objektumhoz tartozó kijelölő pót helyről a tokenet áthelyezi a kijelölő helyre („P\_KSxP” helyről a „P\_KSx” helyre). A gyök felől érkező kijelölésekhez egy közös pót hely tartozik, azáltal biztosított csak az egyik ág kijelölése mellett a másik ki nem jelölése („P\_KCiP”). A start és cél objektumok külön nem záródnak le, csak a kijelölés indítását, illetve visszafele fordítását (célből a start felé), valamint a kijelölés törlésének előkészítését végzik el („P\_KSCCi”).



31. ábra A vágányút start felőli kijelölése a nyomvonalas elv Petri hálós modelljében

Már a kijelölés során elkezdődik azon „oldalágak” kijelölésének törlése, melyek a vágányútban nem vesznek részt. Jelen topológián csak a start felől érkező kijelölés fog kettéágazni. (30. ábra) Mikor a startból érkező kijelölés elér a célba a „P\_KSSCI” helyek valamelyikére, akkor innen a token két irányba haladhat tovább („T\_KCCI” vagy „T\_SCI”). Ha az adott cél a kijelölt cél, akkor a token továbbadódik a cél felőli kijelölés hálórészbe, a megfelelő „T\_KCCI” tranzíció tüzelésével és a célhoz tartozó „P\_KCCiP” hely (a cél kiválasztása során itt tárolódott el a token) kapacitásának felhasználásával. A kijelölés az előzőekben leírtak szerint halad tovább a start felé, és a „P\_KCS1\_2” helyen véget ér. (32. ábra)

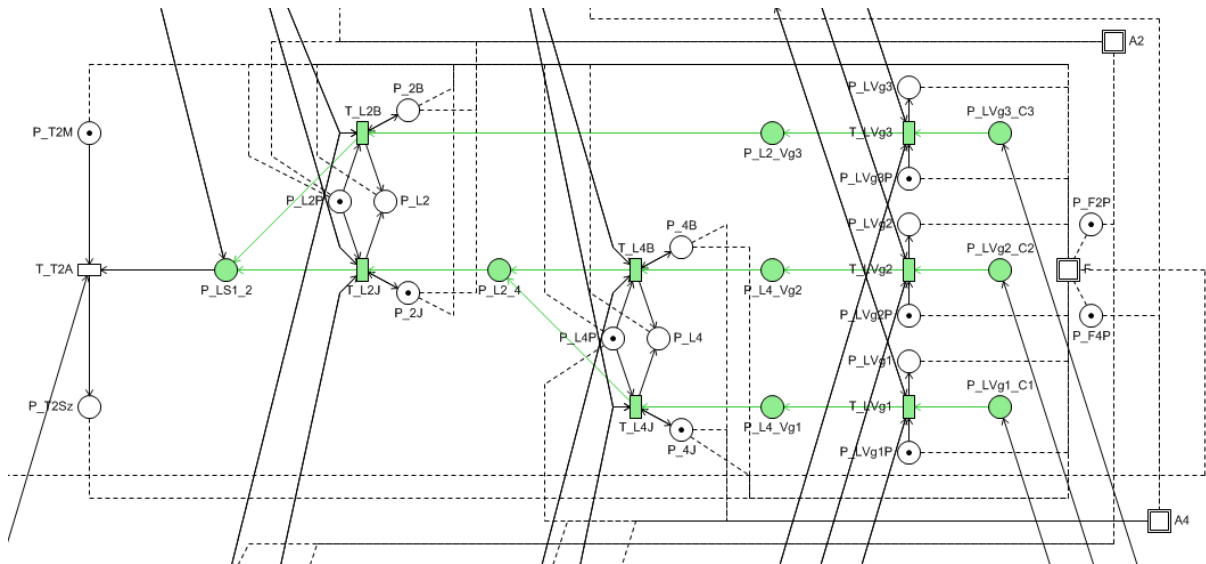


32. ábra A vágányút cél felőli kijelölése a nyomvonalas elv Petri hálós modelljében

Ha azonban a kijelölés a startból olyan célhoz ér, amely nincs a vágányút céljaként kiválasztva, a hozzá tartozó, a vágányútban már biztosan részt nem vevő oldalág kijelölése eltörölhető. A törlést a „T\_TSCI” tranzíciók fordítják meg, mely csak akkor lehet hatásos, ha a „P\_TSCCI” nevű helyen található token (tehát a cél kiválasztásakor innen nem lett elvéve), ami a tüzeléssel fel is használódik. (35. ábra) Az elemek kijelölésének törlése addig tud visszafele haladni, amíg olyan elemhez nem ér, ami a vágányútban részt vesz. A törlés folyamatának leírása, a későbbiekben történik.

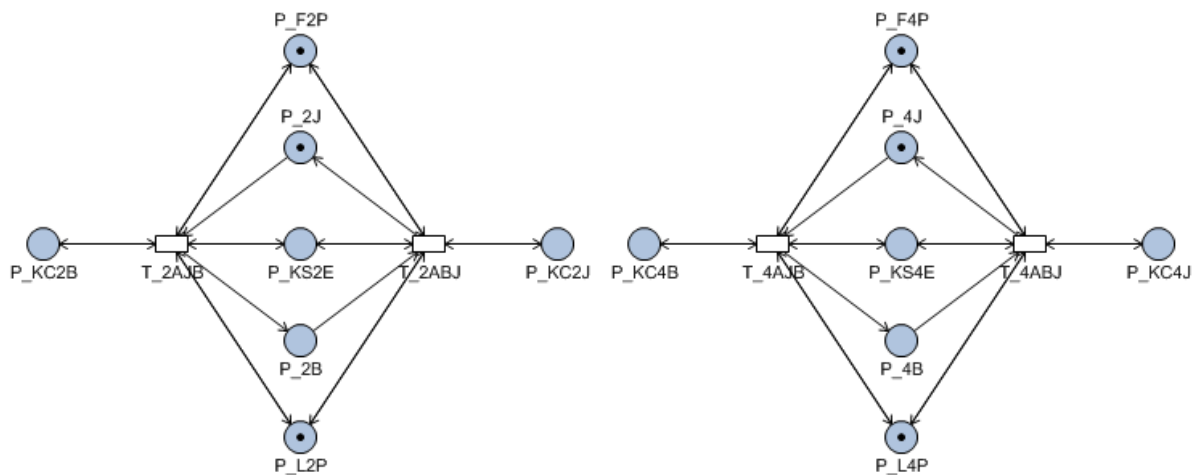
### A vágányút lezárásának folyamata

Amint egy elem mindkét részhálóban kijelölésre kerül (1-1 token tartalmaz a „P\_KSx” és „P\_KCx” kezdetű kijelölés helye), lehetőség van az adott elem lezárására. A lezárás a kijelöléshez hasonlóan történik, a célpontok felől indulva. (33. ábra) A célpontok cél felőli kijelölése (a megfelelő „T\_KCCI” tüzelése, 32. ábra) során token kerül a lezárás hálórész „P\_LVgi\_Ci” helyére, ezzel elindítva a folyamatot. Adott objektum lezáródása során a hozzá tartozó „P\_LxP” helyről tokenet vesz el a lezárást végző tranzíció („T\_Lx”) és tokenet helyez el a „P\_Lx” nevű helyen.



33. ábra A vágányút lezárása a nyomvonalas elv Petri hálós modelljében

A váltók lezárásához szükséges megfelelő végállásuk (token „P<sub>iJ</sub>” vagy „P<sub>iB</sub>” helyen), tehát adott esetben a váltók állítása lehet szükséges, melyet az „A<sub>i</sub>” alhálóok valósítanak meg. (34. ábra) A váltóállítás alhálóok működése a következő. Egy váltó csak akkor állítható („T<sub>iABJ</sub>” vagy „T<sub>iAJB</sub>” tüzelése), ha ki van jelölve az eleje (token a „P<sub>KSiE</sub>” helyen) és valamelyik szára felől (token a „P<sub>KCiJ</sub>” vagy „P<sub>KCiB</sub>” helyen), nincs lezárva (token a „P<sub>LiP</sub>” helyen) és nem foglalt (token a „P<sub>FiP</sub>” helyen). Egy vágányútban érintett váltó a start-cél és cél-start kijelölések során kijelölést kap az eleje felől, ekkor a kijelölés szétágazik, tehát nem egyértelmű, a váltó mely végállása kell a vágányút beállításához. Kijelölés érkezik a szára felől is (mindig csak az egyik felől), ez fogja meghatározni a váltó elvárt állását. Amennyiben a váltó állítása nem szükséges, egyből lezáródhat, ellenkező esetben a fent leírt feltételek együttes teljesülése esetén a váltó átáll, és azután záródik le. A tranzíciók nevében az álláshoz tartozó betűk sorrendje az állításra utal (pl. „T<sub>2AJB</sub>”: 2-es váltó állítás jobb végállásból bal végállásba). A váltólezárás feltétele a megfelelő állás. (33. ábra)



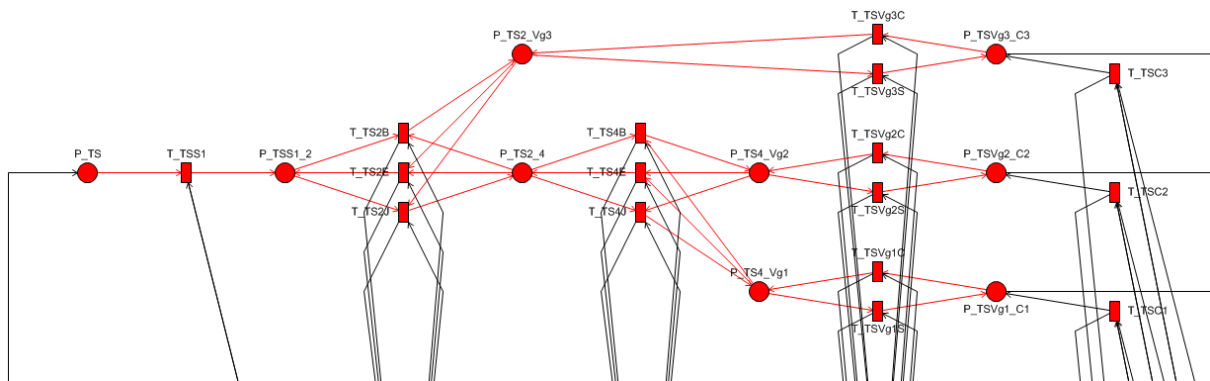
34. ábra A 2-es és 4-es váltók alhálói a nyomvonalas elv Petri hálós modelljében

### A kijelölés törlésének folyamata

A lezáródás a „P<sub>LS1\_2</sub>” helyen ér véget. Ekkor lehetőség lenne a vágányúthoz tartozó (tolatás)jelző állítására. Ezelőtt azonban el kell törölni a már beállított és lezáródott vágányút kijelölését. (35. ábra)

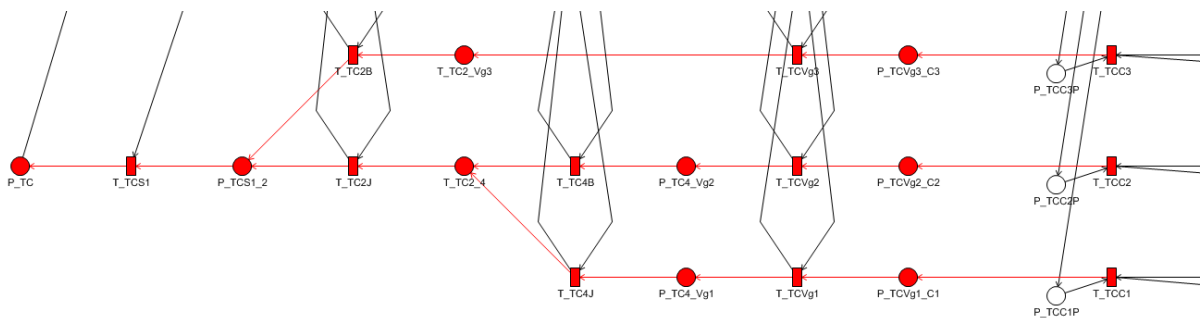


A kijelölés törlése is a start felől indul, a „T\_TSS1” tranzíció tüzelésével elfogyasztja a start kijelölése során a „P\_TS” helyre került token, valamint ellenőrzi, hogy a lezárás visszaért a startba (token a „P\_LS1\_2” helyen, 33. ábra). A kijelölés törlésének alapelve, hogy a kijelölés során a pót helyről elvett, és a kijelölést biztosító helyre tett token onnan elvéve, a kijelölés megszűnjön, és a pót helyre a token visszakерüljön. A start felől történő kijelölés törlésének folyamatát a vágányszakaszokhoz tartozó hálórészek bármely irányba továbbadják („T\_TSVgiS” vagy „T\_TSVgiC” tüzelése), a váltókhöz tartozó hálórészek pedig a két irányból érkezőt a harmadik irányba továbbítják („T\_TSiE”, „T\_TSiJ” vagy „T\_TSiB” tüzelése). A startból végigfutó kijelölés törlés nem ágazik szét, hanem csak a beállított vágányútnak megfelelő nyomvonalon fut le. Az egyes váltóktól viszont csak akkor halad tovább, ha az oldalágak kijelölésének törlése (a „T\_TSCi” tranzíció tüzelésével kezdődően) befejeződött.



35. ábra A vágányút kijelölésének törlése start felől a nyomvonalas elv Petri hálós modelljében

Mivel csak egyetlen start kiválasztása lehetséges, a célból történő kijelölés törlése egyszerűbb hálórészrel rendelkezik. (36. ábra) Ha a célból érkező kijelölés is „találkozna” csúccsal szemben fekvő váltóval, a hozzá tartozó oldalág is a fentihez hasonló bonyolultságú lenne. A célból csak a kiválasztott célnak megfelelő törlés indul el („T\_TCCi” token tüzelése), melyet a kijelöléskor tokennel ellátott „P\_TCCiP” kapacitásának felhasználása biztosít.



36. ábra A vágányút kijelölésének törlése cél felől a nyomvonalas elv Petri hálós modelljében

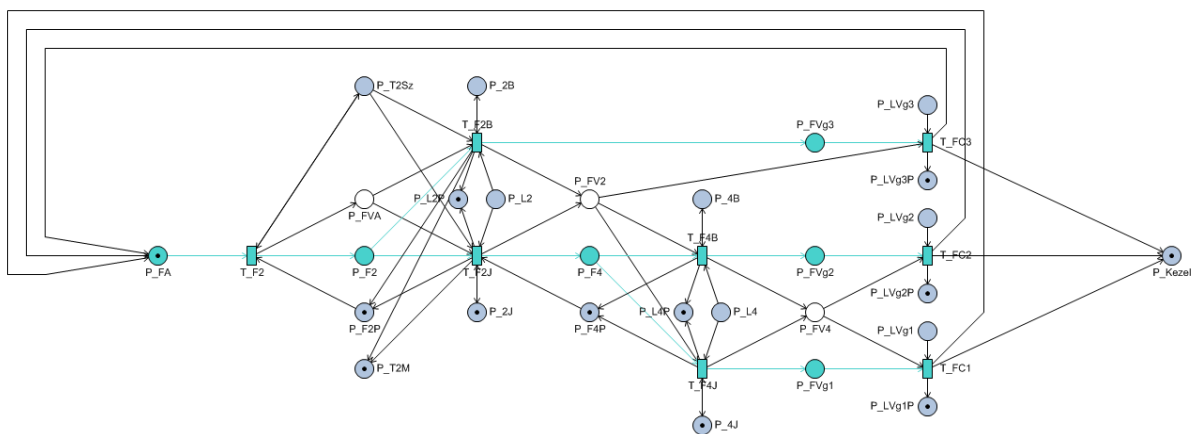
### **A tolatómenet leközeledése és az oldás folyamata**

Ha a kijelölés törlés visszaért a startba (token a „P\_TC” helyen, 36. ábra), a T2 jelző szabadra állítható („T\_T2A” tüzelhet, 33. ábra). A jelző szabadra állása után a menet leközeledhet a beállított vágányúton, mely során oldja a lezárást. (37. ábra) A tolatómentet egy, az oldás hálón a beállított vágányútnak (váltók állásának) megfelelően végigfutó token jeleníti meg. Egy lezárt elem oldásának feltételei (a vágányutat fedező jelző szabadra kapcsolásán és „Megállj!” állásba visszaállításán kívül, 2.1.2. fejezet alapján):

- Az előző foglaltsági szakasz foglalt volt
- Az éppen vizsgált szakasz foglalt lett
- A következő szakasz foglalt lett
- A vizsgált szakasz felszabadult

A fenti működés egy, a foglaltságon felüli „Foglalt volt” helyvel került megoldásra. Amint egy szakasz felszabadul (a hozzá tartozó helyet követő „T\_Fx” tranzíció tüzel), egy token helyez a hozzá tartozó „P\_FVx” helyre, és egyet elvesz az előző szakaszhoz tartozó „P\_FVx” helyről. Tehát egy tranzíció a következő feladatokat látja el, összhangban az előző felsorolással:

- Elvesz egy token a megelőző foglaltsági szakasz FV helyéről
- Elhelyez egy token a vizsgált foglaltsági szakasz FV helyére
- Elhelyez egy token a következő foglaltsági szakasz F helyére (a váltó állásának megfelelően)
- Elvesz egy token a vizsgált foglaltsági szakasz F helyéről



37. ábra Foglaltságok és vágányút feloldás a nyomvonalas elv Petri hálós modelljében

Emellett a (fent említett „T\_Fx”) tranzíció tüzelése elvégzi az adott elem feloldását is, a kijelölés törléshez hasonlóan a lezárás helyekről („P\_Lx”) a token visszahelyezi a lezárás pót helyekre („P\_LxP”). A tolatómenet közlekedésének feltétele, hogy a tolatásjelző továbbhaladást engedélyező jelzési képet mutasson. A tolatásjelző „Tilos a tolatás” állásba az első mögötte lévő szakasz (2-es váltó) foglalttá válását követő felszabadulásával egy időben kapcsolódik vissza. A vágányúttoldás elindulásának tehát feltétele a jelző ismételt felvett továbbhaladást tiltó állása.

Ha a foglaltsági alháló végfutott és a lezárt vágányút feloldódott (minden háló a váltóállást kivéve alapállásban van), új vágányút beállítására kell lehetőséget adni. A foglaltsági alháló „végén” szereplő („T\_FCi”) tranzíciók szerepe kettős. Egyrészt új tolatómenet szimulálására a token visszajuttatják a jelző előtti szakasznak megfeleltethető „P\_FA” helyre, másrészt token juttatnak a főháló „P\_Kezel” helyére, ezáltal új start- és célpontok kijelölésére nyílik lehetőség.

### 3.3.2. Vágányutas elv modellezése Petri hálóval a PetriDotNet eszközben

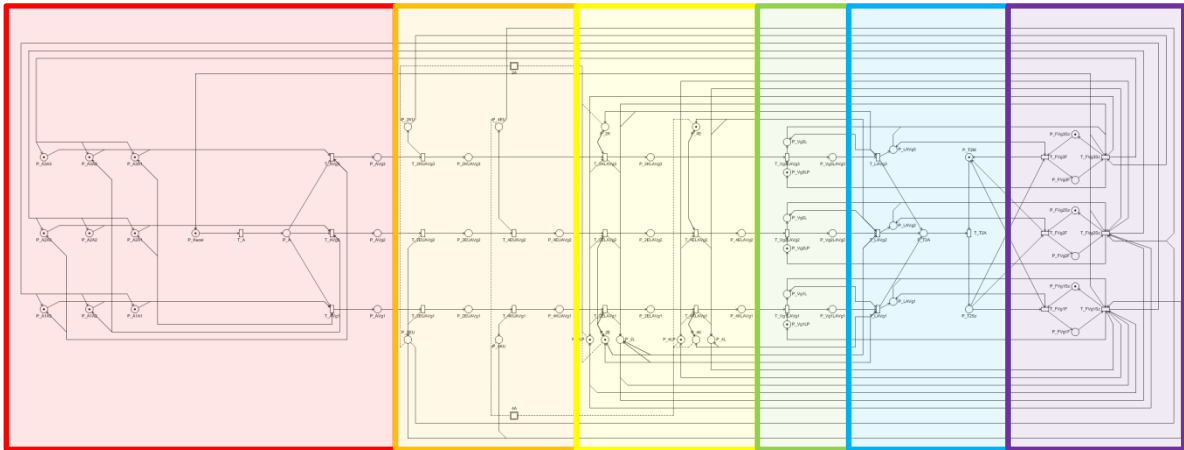
A vágányutas elv modelljének alapját a nyomvonalással ellentétben a két táblázat, a menetterv és az elzárási terv adja. Az alkalmazott jelölések alapvetően a nyomvonalas modellben alkalmazottak logikáját követik. Eltérés, hogy ebben a modellben a vágányutakat start-cél helyett a honnan-hová megnevezés azonosítja. Ezért az egyes vágányutak AVg1-AVg3 jelölése az A-ból a három vágány egyikére beállítandó tolatóvágányutaknak megfeleltethető. Az eltérő jelölések magyarázata a táblázatban

látható. (A váltóállítás parancshoz tartozó P betű más célra való felhasználása miatt inkább a váltóállítás utasítás, és az U betű kerül használatra.)

10. táblázat A vágányutas elv Petri hálós modelljének elemneveiben szereplő betűk magyarázata

Jelölt	Jelölés
A menettervet megvalósító helyek, elnevezésük a vágányút két végpontját jelentik (A: nyílt vonal, i, j: vágány száma)	P_AiAj
A váltó egyenes állása	E
A váltó kitérő állása	K
Utasítás a váltó állítására	U
A tolatásjelző továbbhaladást tiltó/engedélyező („Megállj!”/Szabad) állása	M/Sz
A vágányokhoz tartozó oldószakaszok foglalt/szabad állapota	F/Sz

A modell három fő sorra oszlik, ami a három beállítandó vágányútnak megfelelő függőségeket tartalmazza. A vágányutaknak megfelelő színezett ábra nem készült, mivel ez a felosztás jól megkülönböztethető. A főháló egyes funkciók szerint színezett képét a 38. ábra mutatja, a színek magyarázata a 11. táblázatban látható. A teljes vágányutas elvű Petri hálós modell megtalálható a 2. Mellékletben.



38. ábra A vágányutas elvű Petri hálós modell főhálójának funkciók szerint színezett képe

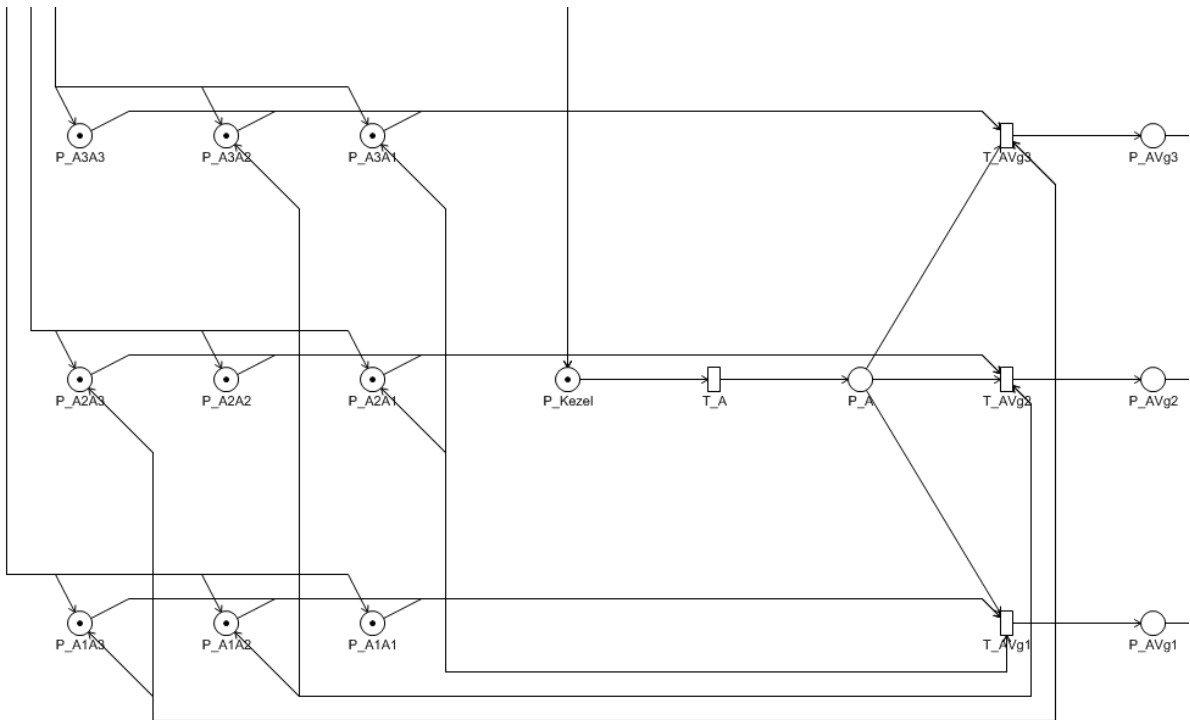
11. táblázat A vágányutas elvű Petri hálós modell egyes funkcióinak színezése

Szín	Tartalom
Piros	Vágányút kiválasztása, menettervi függőségek
Narancs	Váltóállítási parancsok kiadása
Sárga	Váltólezárás
Zöld	Vágányszakaszok lezárása
Kék	Vágányút lezárása és jelzőállítás
Lila	Foglaltság, feloldás

Az egyes helyek és tranzíciók neve az aláhúzás után az objektum nevéből, az elvégzett funkcióból, és abból áll, hogy ez melyik vágányút számára történik. Tehát a sorok mentén a helyek és tranzíciók végződése azonos, és az előbbieknél megfelelően az adott vágányútra utal. A három fő soron kívül található helyek és tranzíciók nevei nem tartalmazzák ez utóbbi információt, mivel működésük globális, több vágányútban is érdekeltek lehetnek (pl. egy váltó lezárása).

### **A menettervi függőségek megvalósítása**

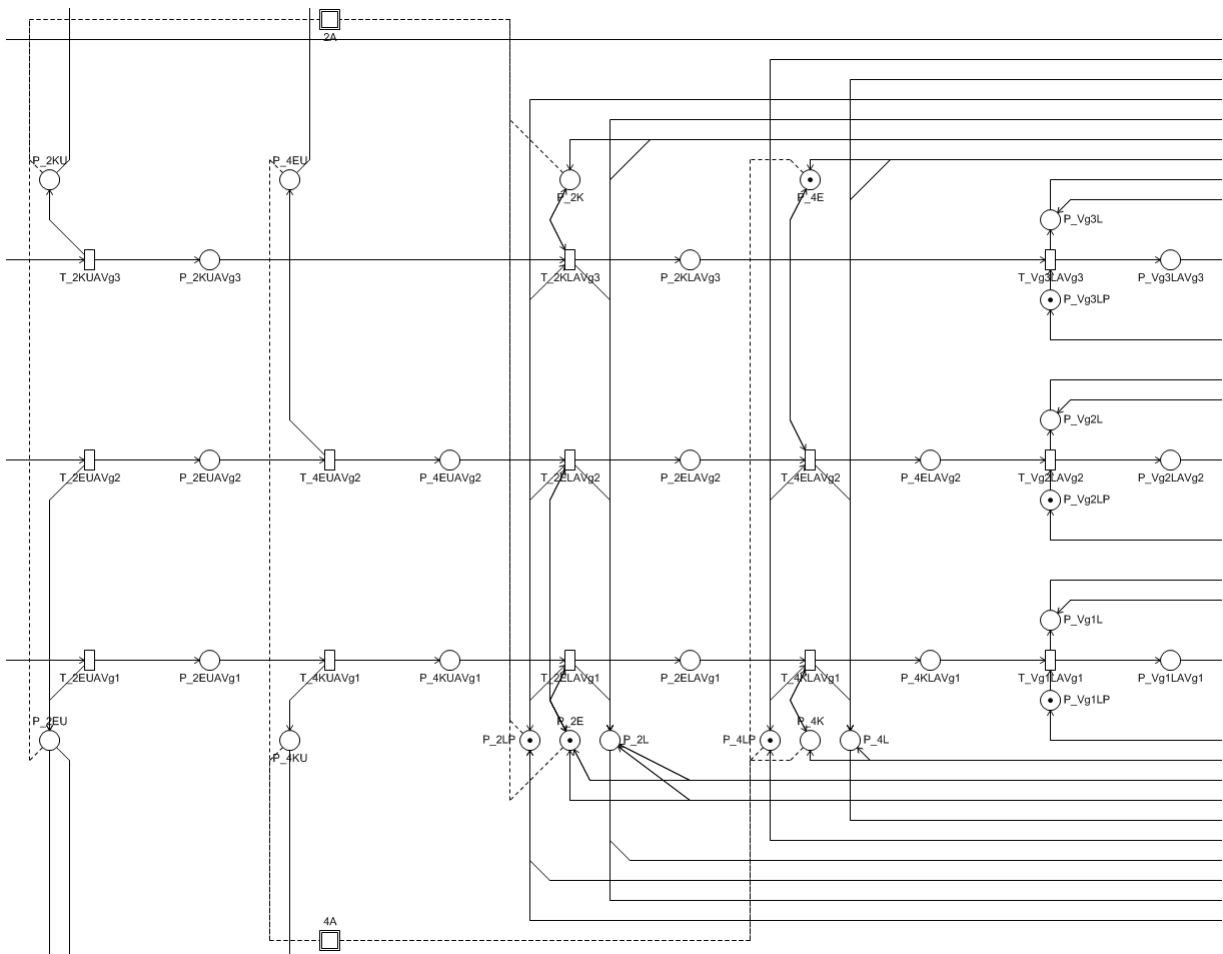
A vágányút kijelölés a „P\_Kezel” helyen tárolt token eltüzelésével, a „honnan” -hely meghatározásával kezdődik, mely jelen példában csak az A (irányú csatlakozó vonal) lehet („T\_A” tüzelése). (39. ábra) Ehhez lehet ezután egy olyan „hová” -helyet választani, amibe a kiválasztott irányból beállított vágányúton el lehet jutni (ez esetben Vg. 1-3). Ebben a lépésben („T\_AVgi” tranzíció tüzelése) kerülnek a menettervi függőségek megvalósításra. A menettervnek megfelelő, helyekből álló mátrix minden helyén található egy-egy token. A kiválasztott beállítandó vágányút tranzíció tüzelésének feltétele, hogy a táblázat vágányúthoz tartozó oszlopának minden helyén legyen token (kivéve a főátlót). Ezzel kerül vizsgálatra, hogy nincs beállított (elrendelt) ellenséges menet. Amennyiben az elrendelés lehetséges, a tranzíció a táblázat vágányúthoz tartozó sorának minden olyan helyéről, mely az adott oszlophoz tartozó vágányúttal ellenséges, elvesz egy token. Ezáltal egyben megvizsgálja azt is, hogy az éppen beállítandó menet nincs még elrendelve, mert ha a főátlóban nincs token, azt nem lehet eltüzelni, tehát a tranzíció le lesz tiltva. Tehát a vágányút kiválasztásakor tüzelő („T\_AVgi”) token a képeletbeli táblázat soraiban tilt le, és az oszlopaiban ellenőriz. Ha két menet nem zárja ki egymást, a metszetükhöz tartozó helyről nem lesz elvéve token. A kizárás a vágányút feloldásáig marad meg (ekkor kerülnek vissza a tokenek).



39. ábra Vágányút-kijelölés és menettervi függőségek a vágányutas elv Petri hálós modelljében

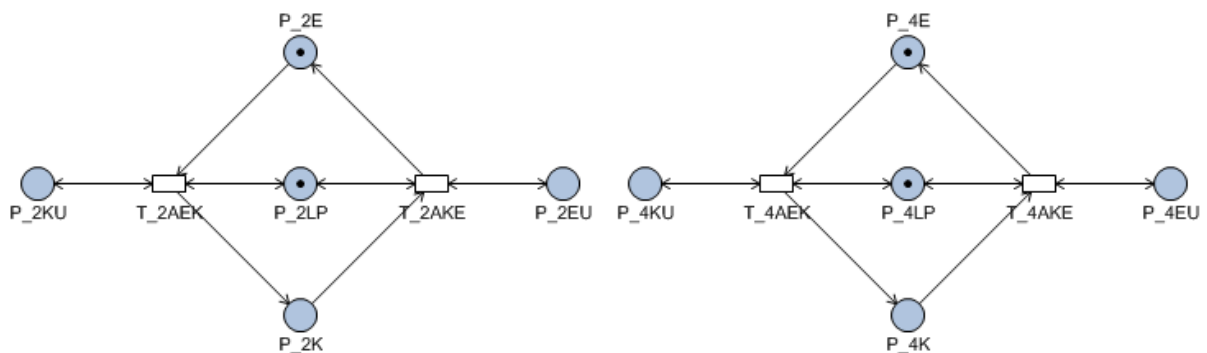
### **Az elzárási terv függőségeinek megvalósítása**

Ha token kerül a „P\_AVgi” helyre, akkor azt jelenti, hogy az adott vágányút ki lett választva, és a menet beállítható. Ezután három lépcsőben történik a vágányút lezáródása. (40. ábra) Először a vágányútban érintett váltók parancsot kapnak a megfelelő állásba való állításukra. A vágányúti függőségeket végző tranzíció tokent juttat a fő hálórész alatt, illetve felett elhelyezkedő „P\_iEU” vagy „P\_iKU” helyekre.



40. ábra Az elzárási terv függőségeinek végrehajtása a vágányutas elv Petri hálós modelljében

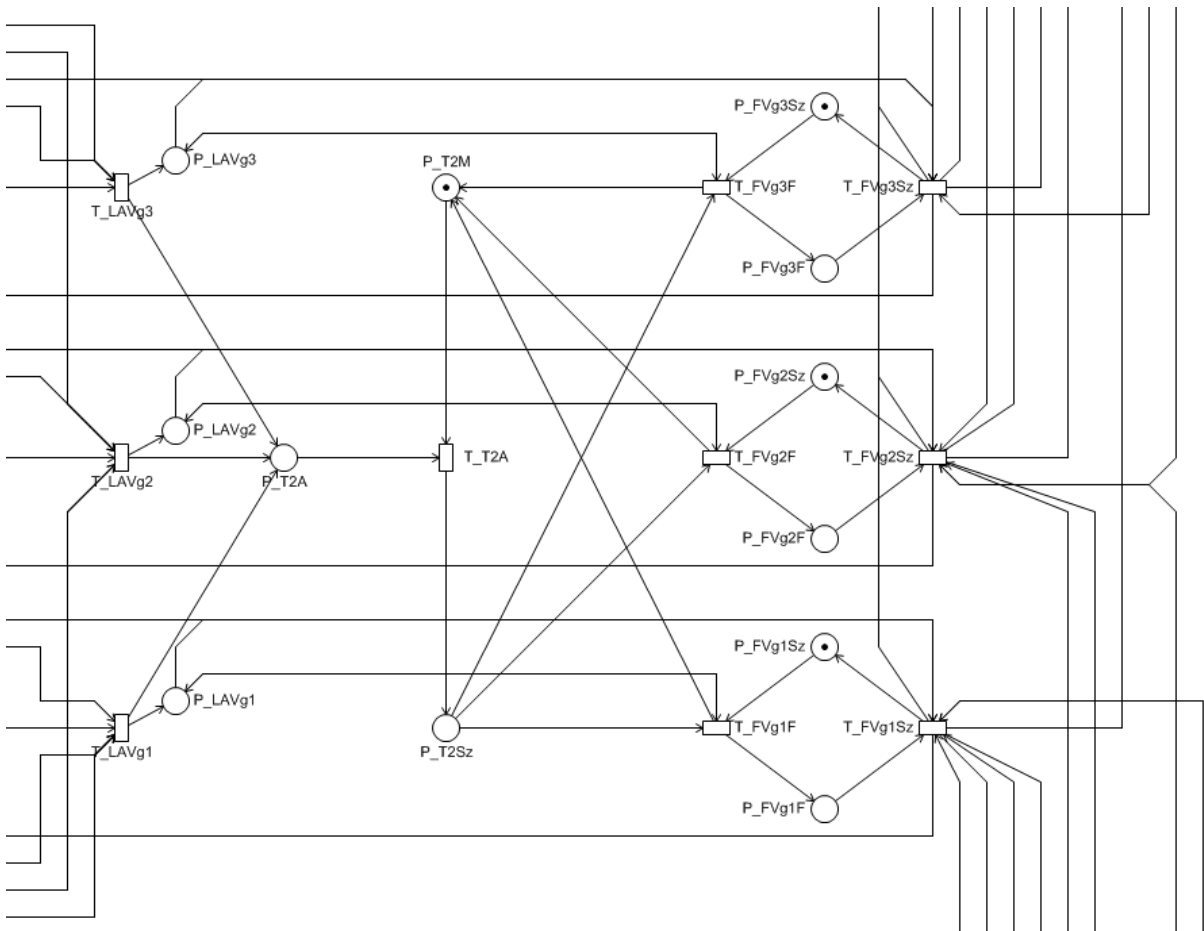
A nyomvonalas modellel ellentétben itt egyértelmű a váltó állásának kijelölése, így az utasítás után a váltó átállhat. (41. ábra) A váltó átállításának csak két feltétele van: kapja meg az álláshoz tartozó parancsot (token „P\_iEU” vagy „P\_iKU” helyen), és ne legyen lezárva (token „P\_iLP” helyen). Mivel a modellben kiszigetelt szakaszként csak az oldószakaszok jelennek meg, a foglaltság modellezett ellenőrzésére (gépi úton történő ellenőrzésére) nem kerül sor. Foglalt váltó állítása természetesen nem megengedett, a példában ez úgy is értelmezhető, hogy kezelő felelőssége a váltó szabad voltáról meggyőződni.



41. ábra A 2-es és 4-es váltók alhálói a vágányutas elv Petri hálós modelljében

Az utasítások kiadása után a vágányúti funkciókat végző hálórész a váltó elvart állásának ellenőrzése mellett lezárja azokat. (40. ábra) A „P\_xLP” helyről a token a „P\_xL” helyre tüzel át a vágányúthoz tartozó („T\_xLAVGi”) tranzíció. A váltóállások ellenőrzése (van-e token „P\_iE” vagy „P\_iK” helyen) a fő hálórész alatt, illetve felett, a lezárás a fő hálórész alatt elhelyezett helyek segítségével történik. A váltó lezárása után az érintett vágányszakaszok lezárása is egyesével megtörténik.

Mikor minden vágányútban érintett elem lezáródott, létrejöhet a vágányút külön lezáródása. (42. ábra) Ezt a „T\_LAVgi” tranzíciók végzik el. Tüzelésük során ellenőrzik valamennyi vágányútban részt vevő váltó helyes (elvart) végállását, valamint valamennyi vágányútban érintett elem lezárt voltát. A tranzíciók működésükkel tokenet juttatnak a vágányút lezáródását jelentő „P\_LAVgi” helyre, valamint a jelző állítását lehetővé tevő „P\_T2A” helyre. A „P\_AVgi” helyen lévő token egyben előkészíti a vágányút oldását (élesíti a fogadóvágányhoz tartozó oldószakaszt). Mint a 3.2. fejezetben említésre került, az elemek egyenkénti lezáródása után külön ellenőrzések és lezárások modellezése nem képezi részét a feladatnak. Jelen modellben ennek ellenére azért készült el, mert egy lépésben az összes ellenőrzés elvégezhető volt (részletesebb elemzést ld. 5.2. fejezet).



42. ábra Vágányútlezárás, jelzőállítás és vágányút feloldás a vágányutas elv Petri hálós modelljében

A „P\_T2A” helyen lévő token eltüzelve a „T\_T2A” tranzíció a T2 tolatásjelzőt szabad állásba állítja (a „P\_T2M” helyről a token a „P\_T2Sz” helyre helyezi át). Ezután a tolatómenet leközeledhet, a jelzőt meghaladva a vágányutat végigjárja (a modellen nem jelenik meg), és rálép a megfelelő vágányon elhelyezett oldószakaszra (token „P\_FVgiF” helyen). Az a szakasz fogja érzékelni a menetet, amelyeket

a megfelelő „P\_LAVgi” hely kijelöli („T\_FVgiF” tranzíció ellenőrzi, hogy a „P\_LAVgi” helyen van token). A foglalttá válással egy időben („T\_FVgiF” tranzíció tüzelése) a jelző is átáll „Tilos a tolatás” állásba (a token visszakerül a „P\_T2Sz” helyről a „P\_T2M” helyre). A szakasz szabadulásával („T\_FVgiSz” tranzíció tüzel) egyszerre megtörténik a főháló (váltóállást kivéve) alapállapotba állítása:

- Oldódik a vágányút (token elvétele a „P\_LAVgi” helyről).
- A vágányúti elemek (váltók és vágányszakaszok) oldódnak, a lezáráshoz tartozó helyekről a token visszakerül a lezáráshoz tartozó pót helyekre („P\_Lx” helyekről a „P\_LxP” helyekre).
- Megszűnik a váltók állására vonatkozó utasítás (token elvétele a „P\_xU” helyekről).
- A menettervi függőségek megszűnnek, újra token kerül a háló elején található helyekből álló mátrix vágányúthoz tartozó, és kizárásokat megvalósító helyeire.
- Új vágányút válik kiválaszthatóvá, token kerül a „P\_Kezel” helyre.

Ezzel véget ér egy vágányút beállításának, felhasználásának és feloldásának folyamata, újak beállítására van lehetőség.

### 3.3.3. Nyomvonalas elv modellezése automatákkal az UPPAAL eszközben

A modell a kezelést szimuláló állapotgépből és négy különböző funkciójú vágányúti elemet reprezentáló, példányosított állapotgépből áll, ezek a következők:

- Vágányút start- és célpontjának kijelölése (vágányúti kezelés)
- Start objektum
- Váltó objektum
- Vágány objektum
- Cél objektum

Az UPPAAL modell alapját ez az öt állapotgép adja, míg a biztosítóberendezési funkciókat ebből az utóbbi négy valósítja meg. A vágányúti elemek a hozzájuk rendelt működést az egymással való kommunikáció (szinkronizáció) alapján, illetve következtében végzik el. A négy különböző vágányúti feladatokhoz tartozó szinkronizáció négy csatornatömb (paraméterezett csatorna) segítségével valósul meg, a következőképp (44. ábra):

- Vágányút kijelölés
- Vágányút lezárás
- A korábban kijelölt és már beállított vágányút kijelölésének törlése
- Tolatómenet közlekedése (foglaltság szimulálása) és a lezárt vágányút feloldása

A csatornatömbök „\_CS” végződést kaptak, első betűjük a Petri hálós modellhez hasonlóan a 12. táblázat szerint értelmezendő.

12. táblázat A nyomvonalas elv UPPAAL modelljének csatornatömbjei

Funkció	Jelölés
Vágányút kijelölés	k_CS
Lezárás	l_CS
Vágányút kijelölés törlés	t_CS
Foglaltság	f_CS

Mivel a vágányútban (és a jelen topológiában) előforduló elemek tipizálhatóak, az objektumok példányosítása aszerint történik, hány darab fordul elő belőlük jelen helyszínrajz esetén. Az egyes elemek példányszámát a 13. táblázat és a 44. ábra mutatja. A példányosítás során az egyes objektumok automatikus ID-t, számozást kapnak nevük mögé. A példában a start, cél és vágány elemek számozása a helyszínrajznak megfelelő, míg a váltó elemek jelölése alatt a szám kétszeresét kell érteni. Az eltérés oka az állomások kezdőponti és végponti (páros/páratlan) oldalának megkülönböztetése. Nagyobb, átmenő állomás modellezése esetén a számozás (a helyszínrajztól függően közel) folytonos.

13. táblázat Példányosításhoz használt változók és értékeik a nyomvonalas elv UPPAAL modelljében

Objektum	Jelölés	Darabszám
Start	s_db	1
Váltó	v_db	2
Vágány	vg_db	3
Cél	c_db	3

A vágányúti elemek kapcsolatait (szomszédjaikat) a „szomszed” nevű tömb írja le. A mátrix és a helyszínrajz kapcsolatát mutatja a 14. táblázat, a 43. ábra és a 44. ábra. Minden elemet egy él köt össze, egy él csak két elemhez csatlakozhat. A táblázat négy (kétszer két) oszlopból áll, melyek jelölése (KP -kezdőpont felől, VP -végpont felől) a kapcsolatok irányultságát jelölik az egyes elemek szemszögéből. A fentiek értelmében tehát egy szám a táblázatban biztosan kétszer szerepel, egyszer a kezdőponthoz, egyszer a végponthoz tartozó oszlopok valamelyikében. Az elemek kapcsolatainak számozása a példában a helyszínrajzon fentről lefelé és balról jobbra történt.

14. táblázat Objektumok kapcsolata táblázatos formában a nyomvonalas elv UPPAAL modelljében

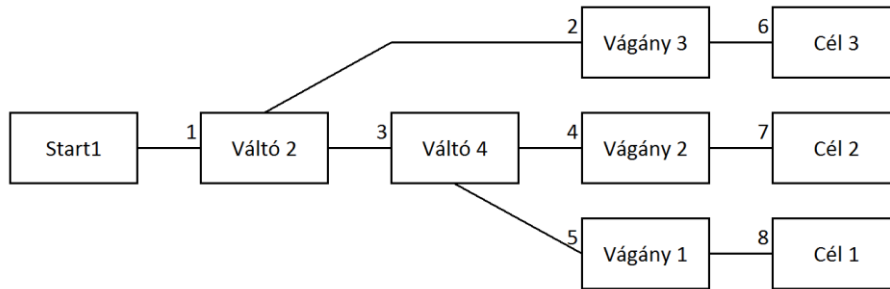
Objektum	KP1	KP2	VP1	VP2
S1			1	
C1	8			
C2	7			
C3	6			
Vg1	5		8	
Vg2	4		7	
Vg3	2		6	
V2 (1)	1		2	3
V4 (2)	3		4	5

A start- és célpontok sorához tartozó oszlopok közül mindig csak egyben lesz szám (jelen példában a startpont a végpont felé, a célpontok a kezdőpont felé „néznek”). A vágányszakaszok sorában egy-egy szám lesz mind a kezdő- mind a végponthoz tartozó oszlopok valamelyikében. A váltók sorában három szám lesz, és abban az oszloppárban lesz kettő, amely irányból haladó jármű a váltót gyök felől érinti (amerre a váltó elágazást létesít). A dolgozat tárgyát képező topológiának megfelelően minden váltó végponti oszlopaiban lesz szám, mivel a váltók a kezdőpont felől csúccsal szemben jártak.

A táblázat sorainak száma automatikusan kerül legenerálásra, az összes vágányúti elem számának megfelelő mennyiségben. A táblázat kitöltése viszont kézzel történik, a helyszínrajznak megfelelően.



Ez a lépés a gyakorlatban az egységek nyomkábelvel való összekötésének feleltethető meg (pl. Domínó 70 típusú berendezésnél).



43. ábra Objektumok kapcsolata grafikus formában a nyomvonalas elv UPPAAL modelljében

A táblázat a vágányúti működések lefutásakor kerül használatra. Az egyes elemek ugyanis a táblázat celláinak segítségével tudják kiválasztani szomszédjaitak szinkronizálásra. A különböző objektumok csoportosítva követik egymást a sorokban. Ezáltal lehetővé válik, hogy minden elem a saját sorára a fölötte lévő elemek maximális számának illetve saját azonosítójának összegével hivatkozzon. Például a II. vágány sora az  $x = s\_db + c\_db + id = 1 + 3 + 2 = 6$ . Mivel a példamodellben a start a vonal kezdőpontjának feleltethető meg, az egyes elemek a szinkronizációt az első oszlopban szereplő értéken várják a start felől és adják tovább a start felé. A harmadik vagy a negyedik oszlopban szereplő értéken szinkronizálnak a cél felé, vagy várják a szinkronizálást a cél felől.

```
//az egyes funkciókhoz tartozó csatornatömbök
chan k_CS[10]; //kijelölés
chan t_CS[10]; //kijelölés törlése
chan l_CS[10]; //lezárás
chan f_CS[10]; //foglaltság

//az egyes alkotórészek darabszáma
const int s_db = 1; //start elemek
const int c_db = 3; //cél elemek
const int vg_db = 3; //vágány elemek
const int v_db = 2; //váltó elemek

//az egyes alkotórészek azonosító típusai
typedef int[1,s_db] id_start;
typedef int[1,c_db] id_cel;
typedef int[1,vg_db] id_vagany;
typedef int[1,v_db] id_valto;

//az egyes elemek szomszédait reprezentáló mátrix
int szomszed[int[1,s_db+c_db+vg_db+v_db]][int[1,4]] = {
{0, 0, 1, 0},
{8, 0, 0, 0},
{7, 0, 0, 0},
{6, 0, 0, 0},
{5, 0, 8, 0},
{4, 0, 7, 0},
{2, 0, 6, 0},
{1, 0, 2, 3},
{3, 0, 4, 5} };

//kiválasztott start és cél
int [0,s_db] start;
int [0,c_db] cel;

//egy kezelés már történt
bool Kezelt = false;

//egy kezelés csak egy kijelölést indíthat
bool Ism = false;
```

44. ábra Globális deklarációk a nyomvonalas elv UPPAAL modelljében

A modellben több integer (egész) illetve boolean (logikai) változó kerül deklarálásra. (44. ábra) A logikai változók értéke megfeleltethető a jelfogók állapotainak is (ez alól a nyomvonalas elvű berendezés modelljénél a foglaltság kivétel, mert így nem a dolgozó áramú elvet követi):

- true → húzott
- false → ejtett (biztonságos).

A vágányúti elemek automatái több közös tulajdonsággal is rendelkeznek, működésükből adódó hasonlóságuk alapján. A 15. táblázatban láthatók összegyűjtve az egyes automatákhoz tartozó változók. A teljes nyomvonalas elvű UPPAAL modell megtalálható a 3. Mellékletben.

15. táblázat Az egyes automaták és változóik a nyomvonalas elv UPPAAL modelljében

Név	Típus	A változó		Szerepel a következő elemnél			
		Kezdeti érték	Magyarázat	Start	Váltó	Vágány	Cél
KijS	bool	false	Az elem kijelölve a start felől	X	X	X	X
KijC	bool	false	Az elem kijelölve a cél felől	X	X	X	X
Lezar	bool	false	Az elem lezárva	X	X	X	X
Foglalt	bool	false	Az elem foglalt		X	X	
Szabad	bool	false	A jelző szabad	X			
KijE	bool	false	Az váltó kijelölve az eleje felől		X		
KijSz	int [0,2]	0	Az váltó kijelölve az egyik szára felől		X		
Allas	int [1,2]	2	A váltó állása		X		

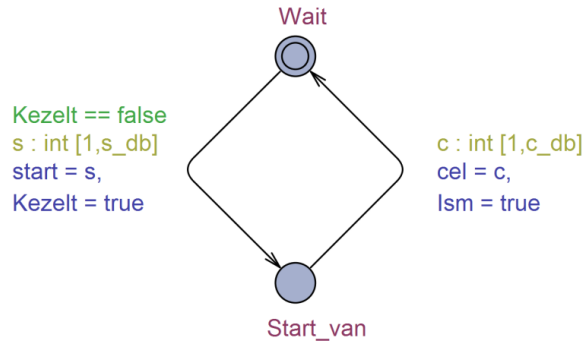
Minden vágányúti objektum ki lehet jelölve start és cél felől is. A start elem start, a cél elem cél felől akkor jelölődik ki, amennyiben a vágányút start- és célpontjának kerülnek kiválasztásra (a „Kezel” automata által). Minden más esetben egy elem akkor lesz start, illetve cél felől kijelölve (azaz a hozzátartozó változó true értékre beállítva), ha a megfelelő irányból a szomszédos objektummal a „k\_CS” csatornán szinkronizál. Mind a négy vágányúti elem lezárható. Foglaltsággal csak azok az automaták rendelkeznek, melyekhez a valós állomáson foglaltságérzékelési szakaszok is tartoznak (váltó és vágányszakasz). Állítható jelző csak a vágányút startpontjában lehet, így a tolatásjelző jelzési képét tartalmazó változó csak a „Start” elemnél található.

Váltók esetében a modell megkülönböztet további két kijelölési lehetőséget: a váltó eleje, illetve szára felől. Erre a váltó állítása szempontjából van szükség. Az, hogy a kettőből melyik változó vesz fel true értéket a „KijS” vagy „KijC” változókkal együtt, attól függ, hogy a váltó csúcsa milyen irányban áll az adott helyszínrajzban. A váltót két szára felől is ki lehet jelölni, ezért a „KijSz” változó három értéket vehet fel (0: nincs kijelölve, >0: az egyik szárán kijelölve). Egy változó tartozik a váltó állásához is, ennek azonban csak két értéke lehet (a kijelöléssel azonos értelemben, 2: jobb, 1: bal). Ez tehát azt jelenti, hogy a váltónak mindig valamely végállásában kell állnia.

#### **A vágányút-kijelölés folyamata**

A működés a vágányút start- és célpontjának kijelölésével kezdődik, melyet a „Kezel” automata végez. (45. ábra) Ezzel az automatával kerül a berendezésbe érkező kezelés szimulálásra. Az automata

két állapottal rendelkezik, mindegyikből egy átmenettel a másikba. A kezdőállapotból indulva egy Select paranccsal választ véletlenszerűen startot egy és a startelemek maximuma között. A visszavezető átmenet során hasonlóképp választ célpontot is. A kiválasztott értékeket egyből a „start” és „cel” változókban eltárolja. Újabb start- és célpontok sorsolására ezután csak a tolatómenet leközeledése után van lehetőség.

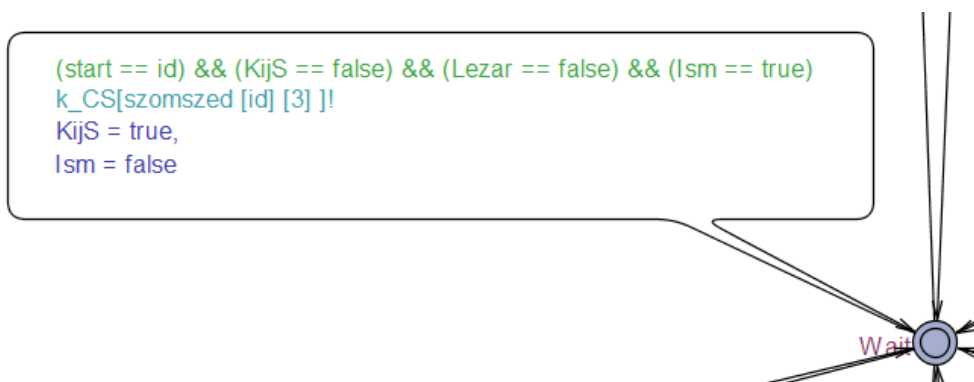


45. ábra A start- és célpontok kiválasztása a nyomvonalas elv UPPAAL modelljében

A vágányút kijelölése során kiválasztott start és cél a „start” és „cel” változókban kerül tárolásra. Annak megakadályozására, hogy a vonat leközeledése előtt újabb start és cél legyen kiválasztva, szolgál a „Kezelt” változó. Az „Ism” nevű változó pedig megakadályozza, hogy kiválasztott start és cél esetén a kijelölés egynél többször is elindulhasson. A vágányút beállása csak a start és cél kijelölése után kezdődik meg. Az automaták a „Wait” elnevezésű kezdőállapotaikban találhatóak. Ehhez az állapothoz kapcsolódnak szövegbuborékszerűen a szinkronizálás során végbemenő vágányúti funkciók. (Az ábrákon a nem releváns állapotátmenetekhez tartozó őrfeltételek, szinkronizációk és értékadások törlésre kerültek.)

A kijelölés először start → cél irányban megy végbe, a „Start” automatán kezdve. A kijelölés a következő feltételek teljesülése esetén indulhat el (46. ábra):

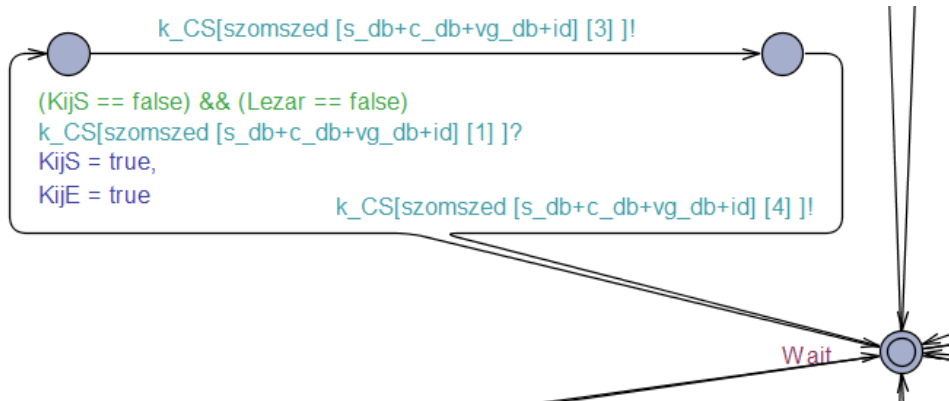
- az adott start elem lett kiválasztva (start == id)
- még nincs kijeölve start felől (KijS == false)
- még nincs lezárva (Lezar == false)
- a startpont kiválasztása óta ez az első kijelölés (Ism == true)



46. ábra A „Start” automata kijelölése startból a nyomvonalas elv UPPAAL modelljében

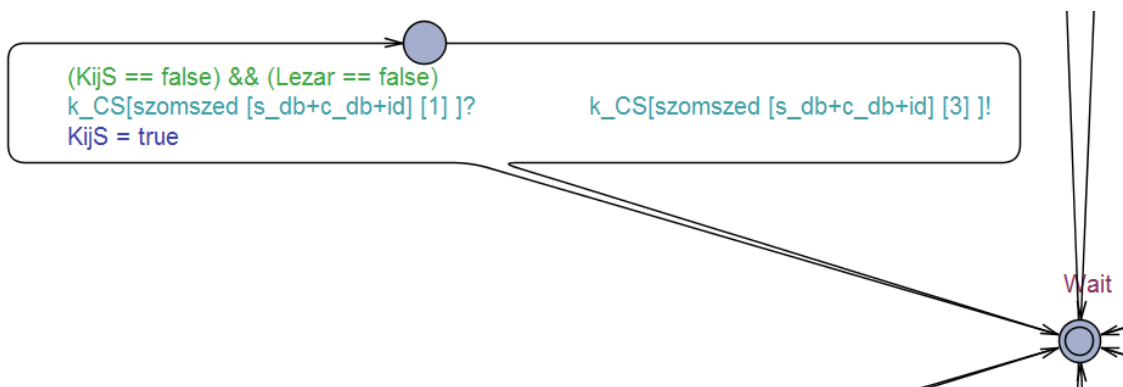
Ha a feltételek teljesülnek, az automata szinkronizál a (cél felé) következő szomszédjával, és beállítja a saját starthoz tartozó kijelölését, illetve az „Ism” változót, hogy újabb kijelölés ne indulhasson el. A „Start” automata után a 2. váltó (az UPPAAL modellben: „Valto(1)”) következik. (47. ábra) A

szinkronizáció akkor lehetséges, ha az objektum nincs még kijelölve a start felől, illetve nincs lezárva. A szinkronizálás során beállítja a starthoz tartozó, illetve a helysínrajznak megfelelően az elejéhez tartozó változókat.



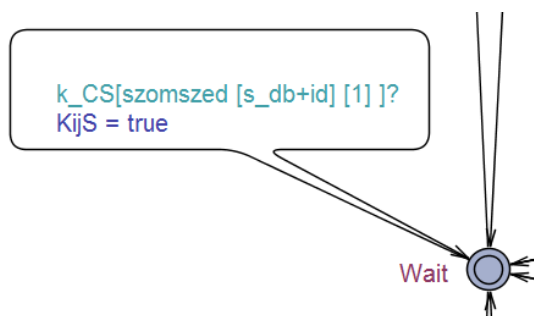
47. ábra A „Valto” automata kijelölése startból a nyomvonalas elv UPPAAL modelljében

Ezután két közbenső állapot segítségével továbbszinkronizál a két szárán elhelyezkedő szomszédai (III. vágány és 4. váltó), azaz továbbadja a kijelölést. A 4. váltó működése a 2. váltóéval teljesen megegyezik, így a következőben a vágány objektum működésének bemutatása következik. (48. ábra)



48. ábra A „Vagany” automata kijelölése startból a nyomvonalas elv UPPAAL modelljében

A szinkronizálások során, a váltóknál szétágazva a kijelölés a startból mindhárom vágányhoz eljut. Itt a kijelölés, illetve a kijelölés továbbadásának folyamata a váltóéhoz hasonló, azzal a különbséggel, hogy csak a „KijS” változó kerül beállításra, illetve csak egy közbenső állapoton keresztül egy irányba, a cél automaták felé kerül a kijelölés továbbadásra. (49. ábra) A cél automata a szinkronizálást fogadja, és beállítja „KijS” változóját.

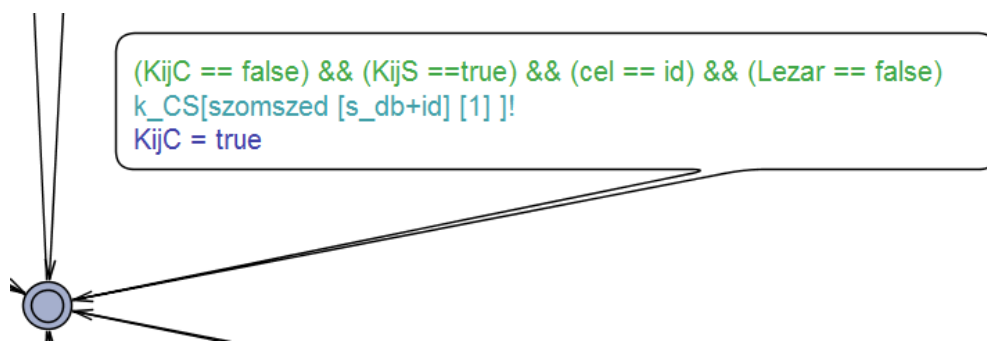


49. ábra A „Cel” automata kijelölése startból a nyomvonalas elv UPPAAL modelljében

Ha ez megtörtént, és nem az adott cél lett kiválasztva, akkor az automata a „Wait” állapotban marad. Amennyiben viszont a cél elem a kiválasztott cél, a starthoz hasonló feltételellenőrzést követően az automata visszafordítja cél → start irányba a kijelölést. (50. ábra) A szinkronizálás a következő feltételek teljesülése esetén indulhat el:

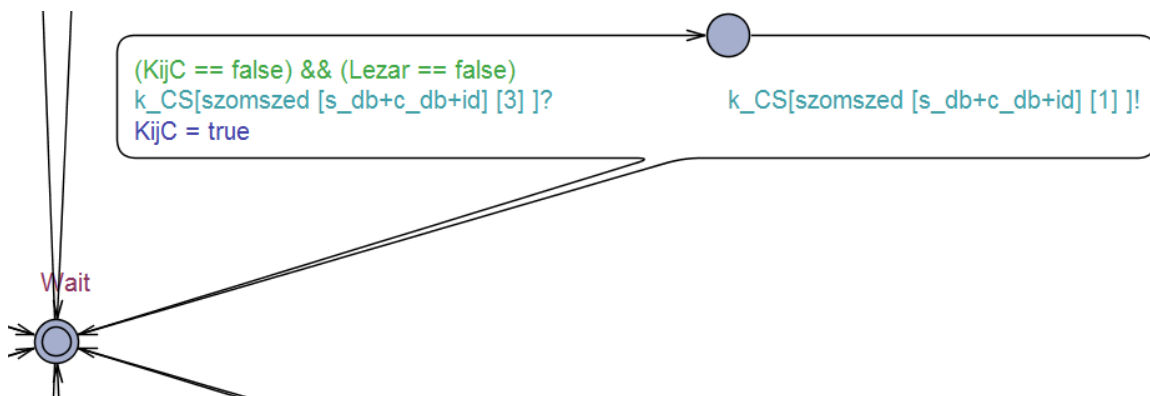
- még nincs kijeölve cél felől (KijC == false)
- már ki van jeölve start felől (KijS == true)
- az adott cél elem lett kiválasztva (cel == id)
- még nincs lezárva (Lezar == false)

A feltételek teljesülése esetén szinkronizálást kezdeményez a start felé lévő szomszédja felé, és true értékre beállítja saját „KijC” változóját. Ezáltal a vágányút-kijelölés megindul a célból a start felé.



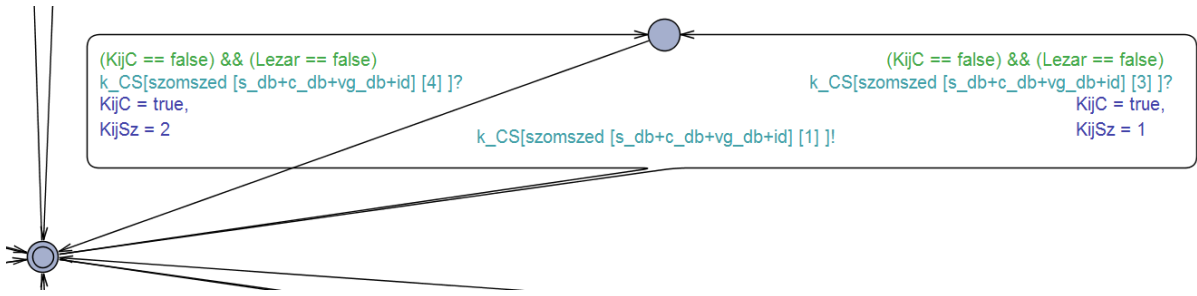
50. ábra A „Cel” automata kijelölése célból a nyomvonalas elv UPPAAL modelljében

A szinkronizációt a vágány automata a cél felől a start felé az ellenkező iránynak megfelelően hajtja végre. (51. ábra)



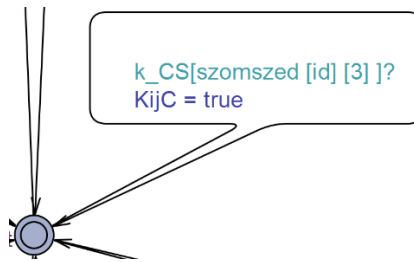
51. ábra A „Vagany” automata kijelölése célból a nyomvonalas elv UPPAAL modelljében

A váltó automata a szinkronizációt mindkét szárán várja, azzal a feltétellel, hogy még nincs cél felől kijelölve illetve lezárva. (52. ábra) Bármelyik irányból érkezen a szinkronizálás, az a másik ágat a „KijC” változó true értékével letiltja. Emellett a kijelölés irányát eltárolja a „KijSz” változóban. Az automata egy köztes állapotban szinkronizál tovább a start felé.



52. ábra A „Valto” automata kijelölése célból a nyomvonalas elv UPPAAL modelljében

A „Start” automata fogadja a szinkronizálást, és beállítja saját „KijC” változóját. (53. ábra)



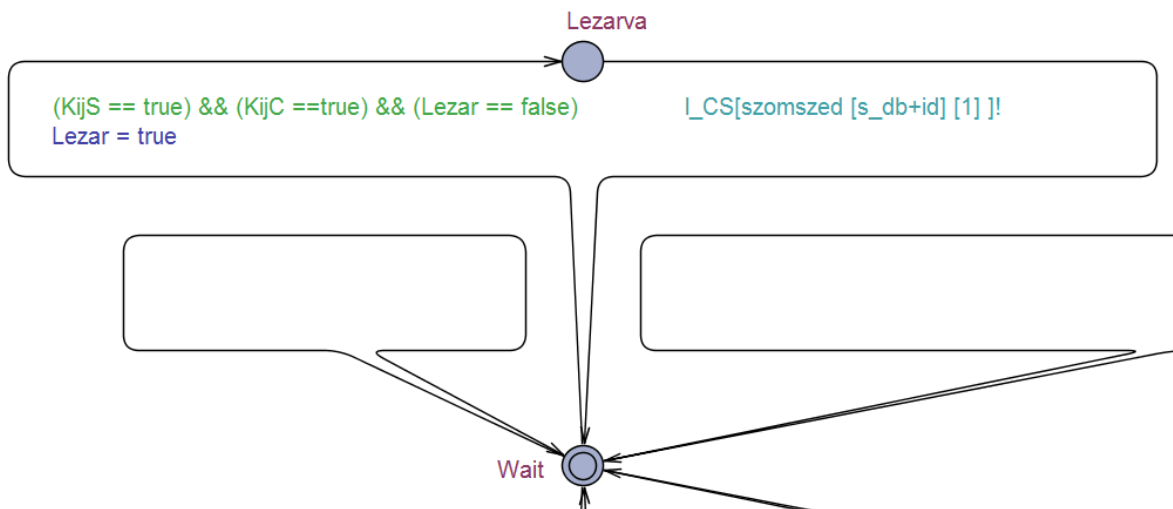
53. ábra A „Start” automata kijelölése célból a nyomvonalas elv UPPAAL modelljében

### A vágányút lezárásának folyamata

Ha egy elem mindkét irányból (start, illetve cél felől is) ki lett jelölve és megfelelően áll, akkor lezáródhat. A célból a start felé haladó kijelöléssel párhuzamosan így elindulhat a váltóállítás, lezárás folyamata is. A lezáródást ellenőrző lánc tehát a „Cel” objektumok egyikétől indul. (54. ábra) Az automata három feltételt ellenőriz:

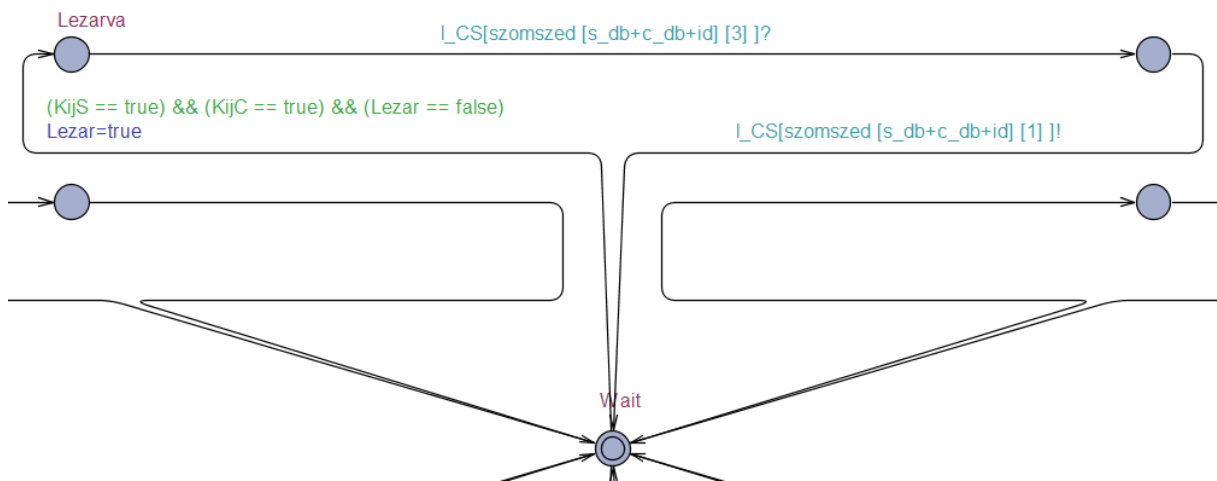
- már ki van jelölve start felől (KijS == true)
- már ki van jelölve cél felől (KijC == true)
- még nincs lezárva (Lezar == false)

Ezek teljesülése estén a célt reprezentáló elem lezáródik, azaz true értékre beállítja „Lezar” nevű változóját. Ezután egy köztes „Lezarva” állapoton keresztül elindítja a start felé a lezárás-ellenőrzés szinkronizációt.



54. ábra A „Cel” automata lezárása a nyomvonalas elv UPPAAL modelljében

A vágányszakaszhoz tartozó automata azonos feltételekkel végzi el a lezárását, majd várja a szinkronizálást a cél felől, és adja tovább egy köztes állapoton keresztül. (55. ábra)



55. ábra A „Vagany” automata lezárása a nyomvonalas elv UPPAAL modelljében

A váltót lezárás előtt először a kívánt állásba kell állítani. Ezt a feladatot a „Valto” automata „allit()” függvénye végzi el. (56. ábra) Meghívásakor a következő feltételek együttes teljesülését vizsgálja:

- nem foglalt (Foglalt == false)
- nincs lezárva (Lezar == false)
- ki van jelölve az eleje felől (KijE == true)
- ki van jelölve valamelyik szára felől (KijSz > 0)

Ezek után a váltót a kijelölt szár irányába állítja, azaz a „KijSz” változóban tárolt értékre állítja be az „Allas” változót.

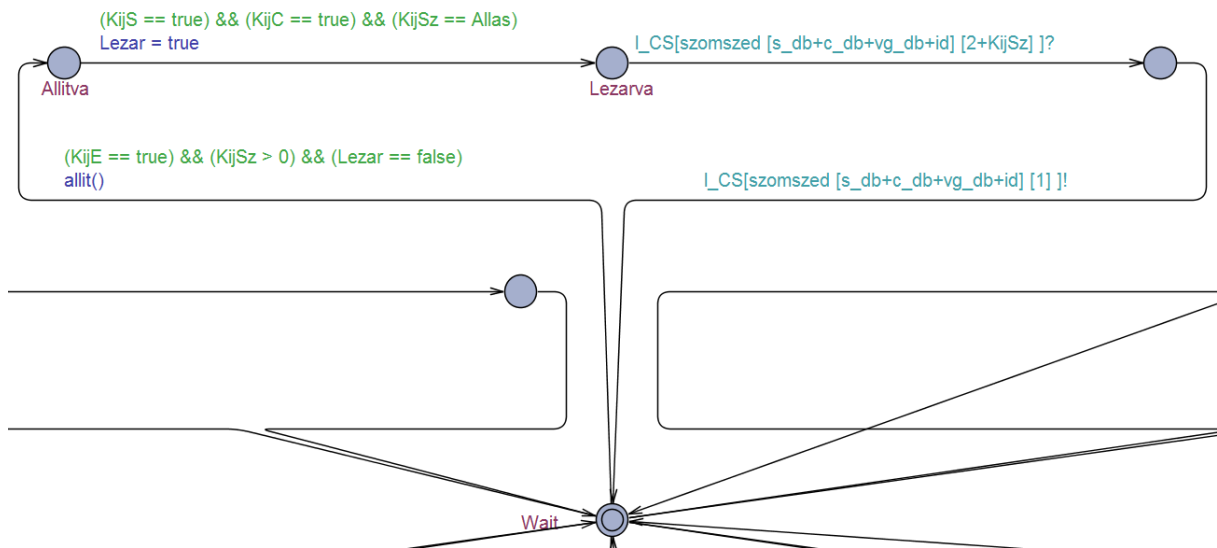
```
void allit()           //a váltót állító függvény, a kijelölt szár irányába állítja
{
    if ((Foglalt == false) && (Lezar == false) && (KijE == true) && (KijSz > 0)) Allas=KijSz;
}
```

56. ábra A „Valto” automata váltóállítást végző függvénye a nyomvonalas elv UPPAAL modelljében

A „valto” automata lezáródását az 57. ábra mutatja. Ha megtörtént a váltó mindkét irányból való kijelölése (az eleje és valamelyik szára felől is), de még nincs lezárva, akkor meghívódik az „allit()” függvény, és a váltó beáll a kívánt állásba, így az „Allitva” állapotba kerül. A lezáródás feltételeként a következők kerülnek ellenőrzésre:

- már ki van jelölve start felől (KijS == true)
- már van jelölve cél felől (KijC == true)
- a kijelölt szárnak megfelelő állásban áll (KijSz == Allas)

Ezután az automata beállítja „Lezar” változóját, a váltó lezáródik. Egy köztes „Lezarva” állapoton keresztül várja a kijelölt szára irányából a cél felőli lezárást ellenőrző láncot, majd adja tovább a start felé.

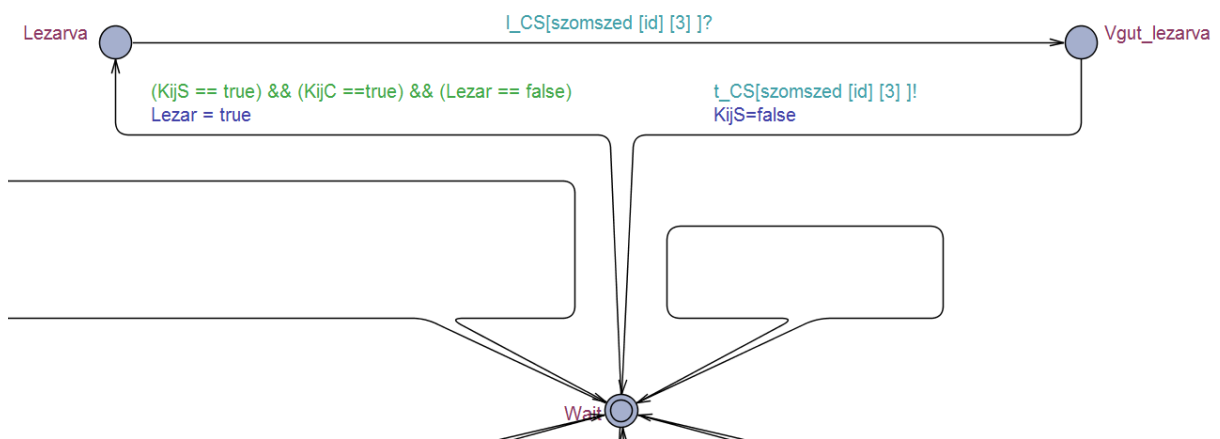


57. ábra A váltó állítása és a „Valto” automata lezárása a nyomvonalas elv UPPAAL modelljében

A „Start” automata a „Cel” és „Vagany” automatákhoz hasonlóan záródik le, és várja a lezárást ellenőrző lánc megérkezését (a szinkronizálást szomszédjával). (58. ábra) Ha a lezáródást ellenőrző lánc elérkezik a startba, az azt jelenti, hogy minden a vágányútban érintett (a cél és a start közötti) elem lezáródott, mivel minden automata csak a saját lezáródása után szinkronizálhat tovább. Az automata így a „Vgut\_lezarva” állapotba kerül. A vágányút lezáródását követően, de még a jelző szabadra állítása előtt a vágányút kijelölését el kell törölni, az érintett változókat alap- (false) állapotba kell állítani.

### A kijelölés törlésének folyamata

A már kijelölt és lezáródott vágányúti elemek kijelölésének törlését a „Start” automata indítja el az erre szolgáló csatornán való szinkronizálással. (58. ábra) A szinkronizáció során saját „KijS” változóját állítja alapba.



58. ábra A „Start” automata lezárása és kijelölésének törlése startból a nyomvonalas elv UPPAAL modelljében

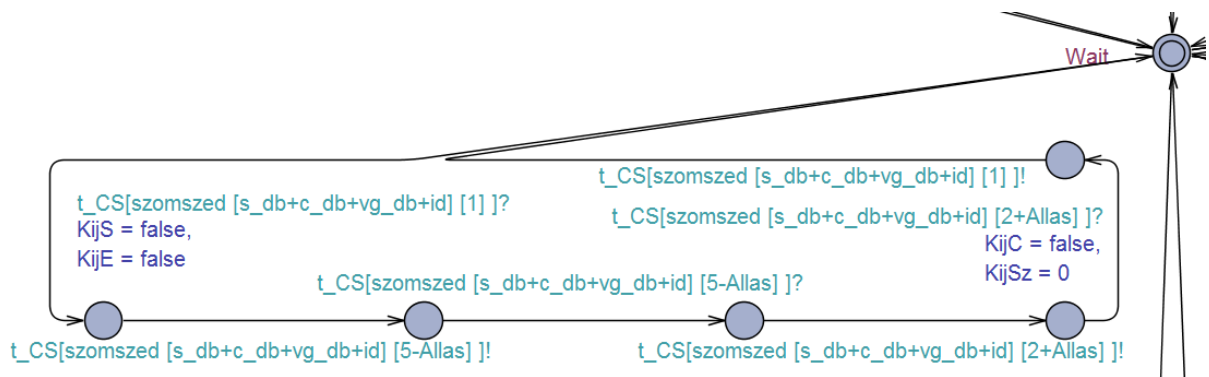
A kijelölés törlése a kijelöléshez képest nem külön-külön fut végig start → cél illetve cél → start irányokban, hanem az összes, legalább egy irányból kijelölt elem, amőbaszerűen. A lefutás részleteiben viszont, minden oldalán előbb start → cél majd cél → start irányban történik. Elágazást a váltóknál létesít, a vágányszakaszok egyszerűen továbbadják, a cél elemek pedig



megfordítják az irányát. A kijelölés törlése először mindig a le nem zárt részekben („oldalágakon”) hajtódik végre.

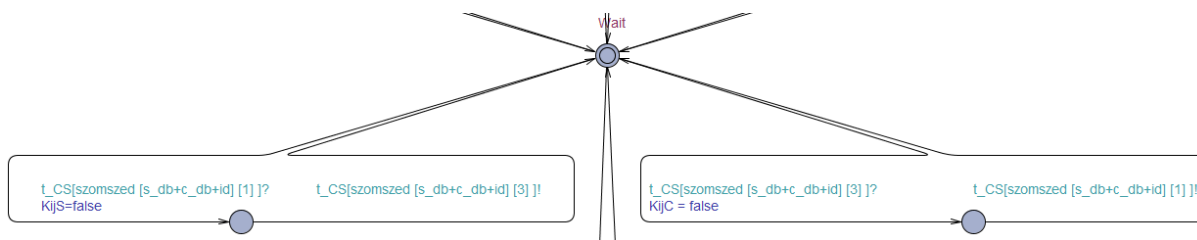
A modellezett helyszínrajzon csak start felől csúccsal szemben járt váltók vannak. Ezeknél a kijelölés törlése a következőkben leírtak szerint történik. (59. ábra) Az automata szinkronizál a megelőző (csúcs felőli) szomszédjával, és alapba tesz saját „KijS” és „KijE” változóit. Ezután továbbadja a kijelölés törlést a váltó állásával ellentétes irányba. Ha tehát a váltó balra terel (Allas = 1), akkor az automata az  $5 - Allas = 5 - 1 = 4$  számítás alapján a „szomszed” mátrix negyedik oszlopában lévő elemmel, tehát a váltó jobb szára felé lévővel szinkronizál. Ezt követően egy köztes állapotba kerül és ebben várja, hogy az elem visszaszinkronizáljon, ami azt jelenti, hogy az oldalág valamennyi elemének kijelölés törlése megtörtént. Ha a szinkronizálás létrejött, a kijelölés törlést továbbadja abba az irányba, amerre a váltó áll. Az előbb bemutatott példa esetén a  $2 + Allas = 2 + 1 = 3$ , azaz a „szomszed” mátrix harmadik oszlopában lévő csatornán szinkronizál (azaz a bal ágán). Egy köztes állapotban található addig, amíg a cél felé lévő szomszédos objektumtól a szinkronizáció vissza nem érkezik. Ennek hatására saját „KijC” és „KijSz” változóit alapba teszi. Ennek megtörténte után a szinkronizálást továbbadja a start felé.

A kijelölés törlés nem tesz különbséget lezárt és le nem zárt váltó között. Jelen modellben példa a le nem zárt váltóra, beállított harmadik vágányra vezető vágányút esetén a 4. számú váltó. Le nem zárt váltó esetén nincs jelentősége, merre adja tovább az automata először a kijelölés törlést, az abba az irányba történik meg, „amerre a váltó nem áll”. A kijelölés törléséhez azért az „Allas” változó kerül felhasználásra a „KijSz” helyet, mert (jelen modell esetén) a váltónak végállása mindig van, viszont elképzelhető, hogy csak az eleje felől kerül kijelölésre. Ebben az esetben  $KijSz = 0$  marad, emiatt a „szomszed” mátrix oszlopának kijelöléséhez tehát nem használható, mert rossz értékre mutatna.



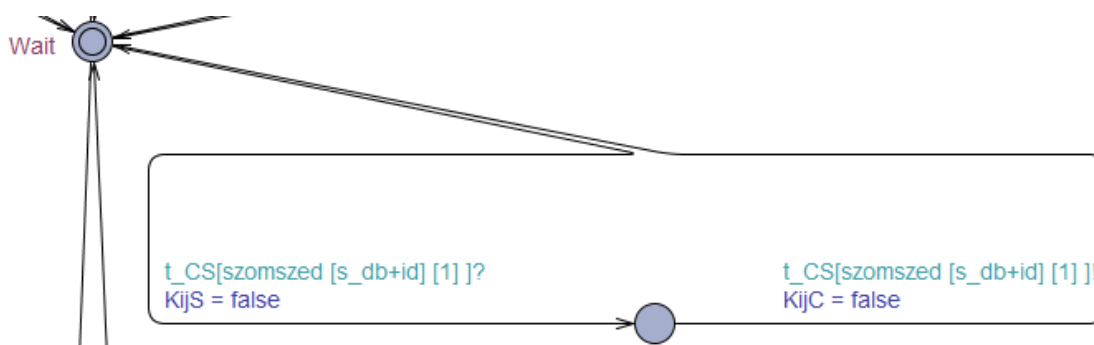
59. ábra A „Valto” automata kijelölésének törlése startból és célból a nyomvonalas elv UPPAAL modelljében

A „Vagany” objektum szimmetrikusan kezeli és adja tovább a bármely irányból érkező kijelölés törlése szinkronizációt. (60. ábra) A kijelölés információt hordozó változókat a váltóhoz hasonlóan a vétel során állítja alapba.



60. ábra A „Vagany” automata kijelölésének törlése startból és célból a nyomvonalas elv UPPAAL modelljében

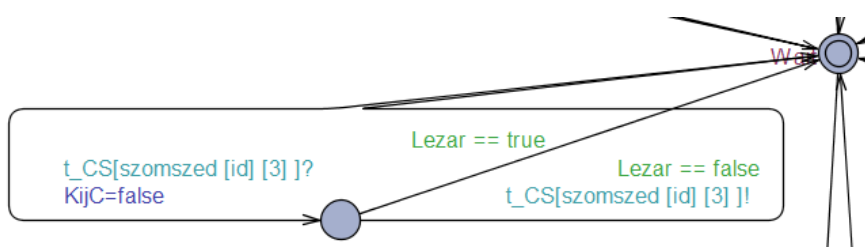
A „Cel” automata fogadott szinkronizáláskor a start felőli, visszaadott szinkronizáláskor pedig a cél felőli kijelöléshez tartozó változó értéket állítja be false értékre. (61. ábra)



61. ábra A „Cel” automata kijelölésének törlése startból és célból a nyomvonalas elv UPPAAL modelljében

A startba visszaérkező kijelölés törlést a „Start” automata fogadja, és a „KijC” változóját inicializálja. (62. ábra) Mivel a modellben csak egy startpont szerepel, a vágányút beállítás esetén mindig le lesz zárva, tehát az automata a köztes állapotból a kezdőállapotba tér vissza. (A másik ág esetét ld. az alfejezet végén).

Ha a kijelölés törlése ismét elérkezik a startba, az azt jelenti, hogy a vágányút le van zárva, és a lezárt vágányúthoz tartozó kijelölés el lett törölve.

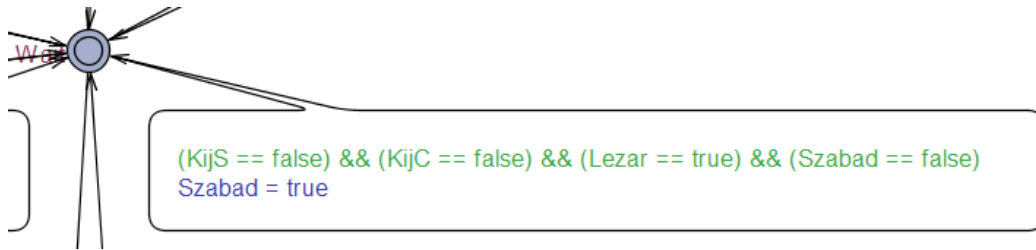


62. ábra A „Start” automata kijelölésének törlése célból a nyomvonalas elv UPPAAL modelljében

Ekkor lehetőség van a jelző továbbhaladást engedélyező állásba állítására. (63. ábra) Ez a kezdőállapothoz tartozó állapotátmenettel, a következő feltételek együttes teljesülése esetén történhet meg:

- már nincs kijelölve start felől (KijS == false)
- már nincs kijelölve cél felől (KijC == false)
- már le van zárva (Lezar == true)
- a jelző még nincs szabadra állítva (Szabad == false)

Ha minden feltétel teljesül (minden változó értéke megfelelő), a jelző szabadra (a „Szabad” változó true értékre) állítható.



63. ábra A „Start” automatához tartozó startjelző szabadra állítása a nyomvonalas elv UPPAAL modelljében

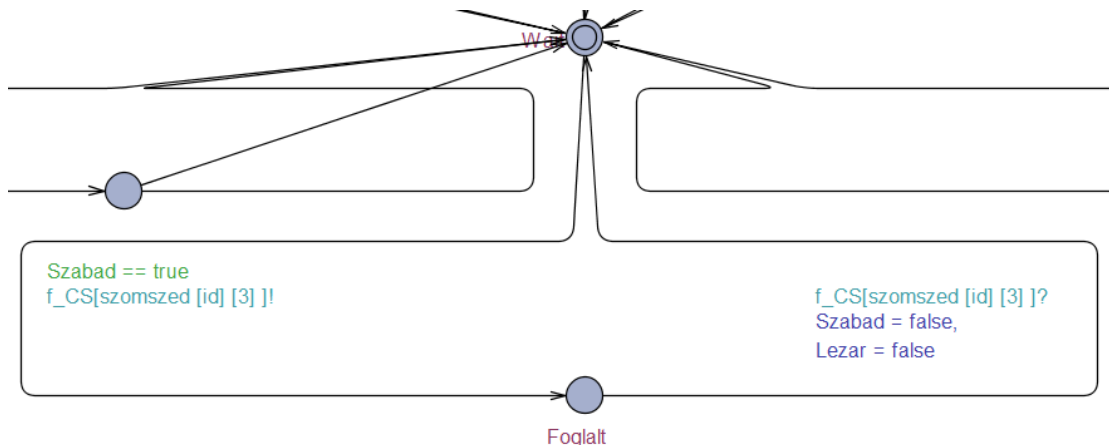
### A tolatómenet leközeledése és az oldás folyamata

Ha a jelző szabadra állt, a tolatómenet leközeledhet. A közlekedő vasúti járművek szimulálása szükséges a vágányút feloldódásához. Ehhez az UPPAAL modellben is egy megfelelően rövid (jelen példában a legrövidebb hosszúságú vágányúti elem hosszánál rövidebb) jármű lett alapul véve. Ennek megfelelően a tolatómenet mindig legfeljebb két szakaszt foglalhat el. Egy foglaltságérzékelési szakasz szempontjából négy lehetséges állapot képzelhető el és különböztethető meg:

- a menet a vizsgált szakaszra rálépett (szinkronizálás fogadása a megelőző szakasztól)
- a menet az előző szakaszból lelépett (szinkronizálása kérése a megelőző szakasztól)
- a menet a következő szakaszra rálépett (szinkronizálása kérése a követő szakasztól)
- a menet a vizsgált szakaszból lelépett (szinkronizálása fogadása a követő szakasztól)

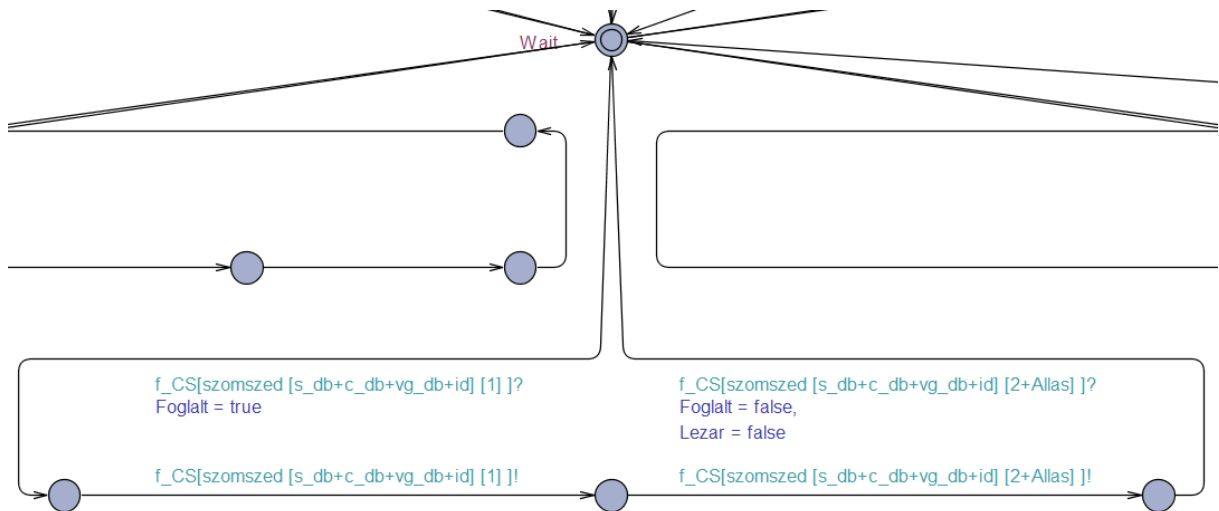
A start és cél objektumok nem rendelkeznek saját foglaltságérzékelési szakaszokkal, azonban a foglaltság-szimulációban részt vesznek. Emellett feloldásuk is szükséges, ami a modellekben a megelőző (foglaltságérzékelő szakasszal rendelkező) elemek oldódásával együtt történik meg. A start és cél elemek a foglaltsági szimulációban oly módon vesznek részt, hogy a tolatómenetet a startnál beléptetik, a célnél pedig kiléptetik. Ez elképzelhető úgy is, mintha a szomszédos foglaltságérzékelési szakaszokat jelenítenék meg ily módon.

A foglaltság szimulálása a következőképpen történik, a „Start” automatán kezdve (64. ábra). Ha a jelző szabadra áll, akkor a vonat beléphet. Az automata szinkronizál az őt követő elemmel. Ha az őt követő elem visszaszinkronizál (a tolatómenet a jelzőt megelőző szakaszból lelépett), akkor a jelző továbbhaladást tiltó („Tilos a tolatás”) állásba visszaáll, és a start elem lezárása oldódik.



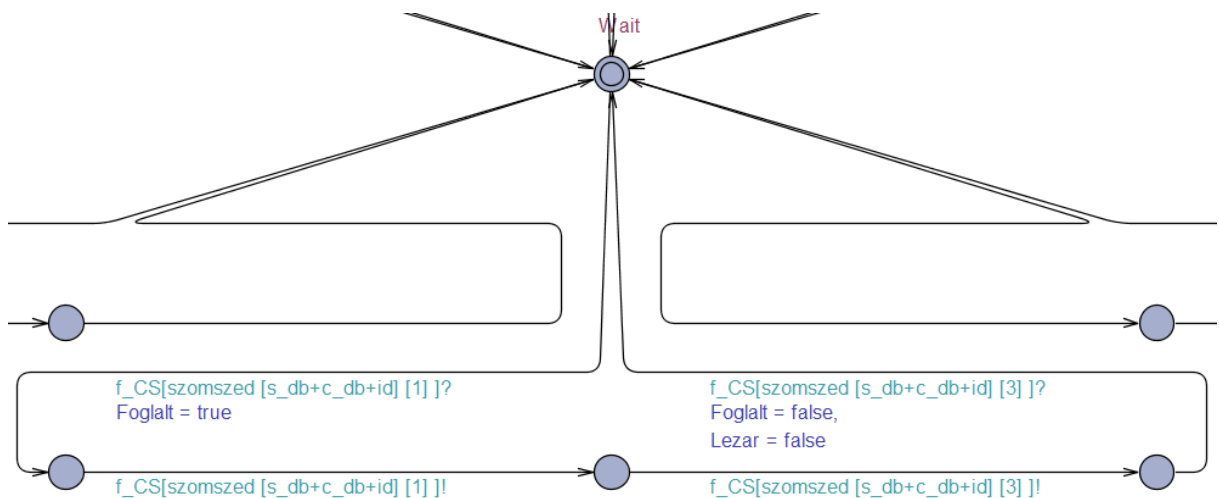
64. ábra A „Start” automata foglaltsága és feloldódása a nyomvonalas elv UPPAAL modelljében

A váltó objektum a fent leírt folyamattal kezeli a foglaltságot, melynek során a menet szakaszba lépésével a „Foglalt” változóját true értékkel beállítja. (65. ábra) A tolatómenet a váltó állásának megfelelően közlekedik tovább, tehát a szinkronizáció a váltó terelésének irányába adódik tovább. Mikor a menet a szakaszt elhagyja (a következő szakasz visszaszinkronizál), a „Foglalt” változó ismét false értéket kap, valamint az elem lezárása feloldódik (a „Lezar” változó szintén false értéket kap).



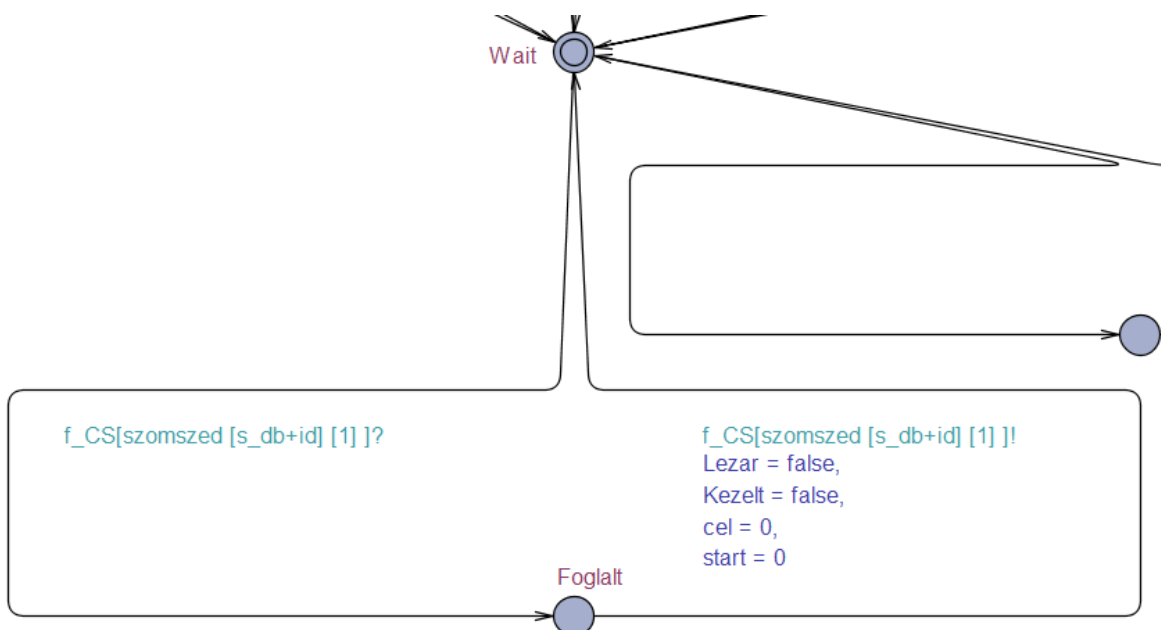
65. ábra A „Valto” automata foglaltsága és feloldódása a nyomvonalas elv UPPAAL modelljében

A „Vagany” automata foglaltsága, valamint feloldódása a váltóéval analóg módon történik, azzal a különbséggel, hogy a szinkronizáció továbbadása egyértelmű. (66. ábra)



66. ábra A „Vagany” automata foglaltsága és feloldódása a nyomvonalas elv UPPAAL modelljében

A cél elem a start elemmel analóg módon csak a kiléptetést tudja elvégezni, mely során ő maga is oldódik. (67. ábra) Mivel azonban a modellezni kívánt működés a végéhez ért, lehetővé kell tenni új vágányút beállítását. Ennek érdekében a „Kezelt” globális változó false, a „start” és „cel” globális változók 0 értékkel kerülnek inicializálásra. Ezek után a „Kezelt” automata működésével új vágányutak beállítása kezdődhet meg.

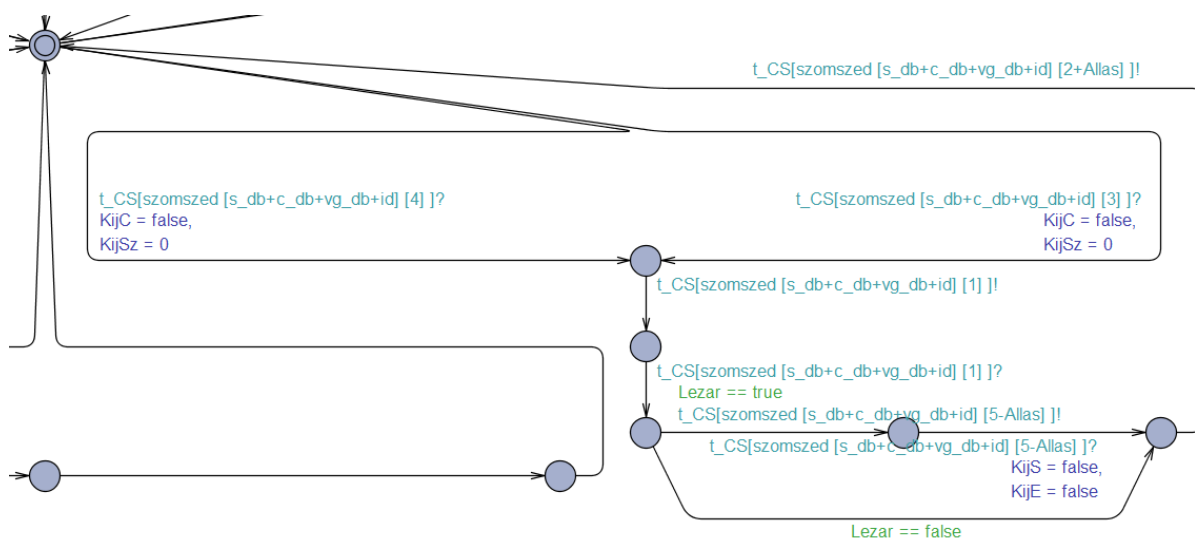


67. ábra A „Cel” automata foglaltsága és feloldódása a nyomvonalas elv UPPAAL modelljében

**A modellben nem használt funkciók**

Mivel az egyes automaták lehetőség szerint minél általánosabb használatának figyelembevételével készültek, néhány funkciójuk jelen modellben nem kerül használatra. A modell elő van készítve több startpont esetére is. (62. ábra) Ebben az esetben lesz olyan start, ami nem kiválasztott startpont, viszont hozzá a kijelölés a cél felől is elérkezik. A cél felől érkező kijelölés törlése a fent bemutatott köztes állapotból a másik állapotátmenet (amikor a start elem nincs lezárva) során történik meg.

A „Valto” automatában szerepel olyan váltó kijelölésének törlésére lehetőség, amely a start felől nem csúccsal szemben járt (68. ábra). Itt ez a funkció a következőképp került megoldásra. A váltó a szinkronizációt valamely szára felől várja, mely során a kijelöléséhez tartozó változókat alapba állítja, majd a kijelölés törlést továbbadja a csúcsa felé. A kijelölés eltörlésének folyamata végighalad a kijelölt célokig, majd visszafordul.



68. ábra Start felől gyök felől járt váltó kijelölésének törlése a nyomvonalas elv UPPAAL modelljében

A váltó eleje felől visszaérkező szinkronizálás után meg kell különböztetni két lehetőséget. Ha a váltó nincs lezárva, akkor a váltó objektum csak gyök felől lett kijelölve, tehát a másik száron nem kell törölni, a szinkronizáció egyszerűen továbbadható visszafele. Ezért a „Lezar” változó false értéke mellett a következő lépések kihagyhatók. Ha a váltó le van zárva (Lezar == true), akkor kijelölés történt a csúcs felől is, tehát azt el kell törölni. A kijelölés törlést tehát továbbadja az állással ellentétes száron, majd onnan várja vissza. A visszaérkező szinkronizálás során alapba állítja a start felőli kijelöléshez tartozó változóit. Ezután a szinkronizáció továbbadható az állásnak megfelelő száron.

Jelen UPPAAL modell tehát könnyen kiterjeszthető más topológiára is, az egyes elemek darabszámának, illetve az új „szomszed” tömb tartalmának, valamint a váltó automata kijelölést végző változóinak módosításával.

### 3.3.4. Vágányutas elv modellezése automatákkal az UPPAAL eszközben

A modell a vágányúti logikát (a menettervet és az elzárási tervet) megvalósító fő automatából („Menet\_Elzarasi”), valamint váltó („Valto”) és vágányszakasz („Vagany”) automatákból áll. A modellben szereplő egyetlen jelző (és annak egyetlen változója) a „Menet\_Elzarasi” automatában került leképezésre. Nagyobb modellezett állomás és több jelző esetén a jelzők külön automatába szervezése lenne ideális. A vágányutas elv UPPAAL modelljében nem szerepelnek start- és célpontok, mivel a vágányút beállítása nem segítségükkel, hanem központi logikában leképzett táblázatok alapján történik.

A vágányúti funkciók megvalósításához öt (4+1) csatornatömb került definiálásra. (16. táblázat és 69. ábra) Ebből az első kettő segítségével értelemsszerűen adja ki a „Menet\_Elzarasi” automata a váltók állítására vonatkozó utasítást a „Valto” automaták felé, és a lezárásukra vonatkozó információt a „vl\_CS” csatornán várja tőlük. A vágányszakasz objektumoktól csak a lezárási információt kapja meg a központi automata. Az összes egy vágányútban érintett elem feloldása egy közös broadcast csatornával történik. A teljes vágányutas elvű UPPAAL modell megtalálható a 4. Mellékletben.

16. táblázat A vágányutas elv UPPAAL modelljének csatornatömbjei

Funkció	Jelölés
Váltó kitérőbe állítás	vk_CS
Váltó egyenesbe állítás	ve_CS
Váltó lezárás	vl_CS
Vágányszakasz lezárás	vg_CS
Vágányúti elemek <b>old</b> ása (broadcast csatorna)	old_CS

*//az egyes vágányúti elemekhez tartozó csatornatömbök*

```
chan vk_CS[10]; //váltó kitérőbe állítás utasítás
chan ve_CS[10]; //váltó egyenesbe állítás utasítás
chan vl_CS[10]; //váltó lezárás információ
chan vg_CS[10]; //vágány lezárás információ

broadcast chan old_CS[10]; //oldó broadcast csatorna
```

69. ábra A vágányutas elv UPPAAL modelljének csatornatömbjei

Annak ellenére, hogy nincsenek külön start- és cél automaták, ugyanaz a négy globális állandó kerül definiálásra, mint a nyomvonalas elvű modellben. (70. ábra) Ebből az első kettő az elméletileg lehetséges vágányutak számának meghatározására, míg az utolsó kettő az egyes automaták (váltó és vágányszakas) példányosítására kerül felhasználásra. Legfeljebb annyi vágányút beállítása képzelhető el, mint amennyi a start és cél elemek számának szorzata.

```

//az egyes alkotórészek darabszáma
const int s_db = 1;      //start elemek
const int c_db = 3;      //cél elemek
const int vg_db = 3;     //vágány elemek
const int v_db = 2;     //váltó elemek

//az egyes alkotórészek azonosító típusai
typedef int[1,vg_db] id_vagany;
typedef int[1,v_db] id_valto;

//vágányutak száma
typedef int[0,s_db*c_db] vgut_db;
vgut_db vgut = 0;

```

70. ábra Globális deklarációk a vágányutas elv UPPAAL modelljében

A menetterv az egymást kizáró vágányutakkal tömb formájában, konstansként, „menetterv” néven kerül definiálásra. (71. ábra) A mátrix azon elemei nulla értékűek, amelyek az egymást kizáró menetekhez tartoznak. Jelen modellben mindegyik menet kizárja a többi lehetségest. Ahol egymást nem kizáró menetek vannak, ott egyes érték szerepel(ne). A függőségeket a beállított vágányutak és a menetterv függvényében módosított „fugges” tömb segítségével biztosítja a „Menet\_Elzarasi” automata. A „fugges” egy olyan táblázat, melynek kezdetben minden eleme egy értékű. Vágányút beállítása esetén a vágányúthoz tartozó sor elemei a menetterv azonos sorával megegyező értéket vesznek fel. Új menet beállítása esetén a beállítandó vágányúthoz tartozó oszlop elemei kerülnek ellenőrzésre. Ha az oszlopban csak egyesek szerepelnek, a vágányút beállítható.

```

//a menetterv
const int menetterv[int[1,s_db*c_db]][int[1,s_db*c_db]] = { { 0, 0, 0}, { 0, 0, 0}, { 0, 0, 0} };

//a kizárásokat megvalósító táblázat
int fugges[int[1,s_db*c_db]][int[1,s_db*c_db]] = { { 1, 1, 1}, { 1, 1, 1}, { 1, 1, 1} };

```

71. ábra A menettervi függőségeket megvalósító globális tömbök a vágányutas elv UPPAAL modelljében

A „Menet\_Elzarasi” automata a menettervi függőségeket három függvény segítségével végzi, míg az elzárási terv az állapotaiban került leképezésre. A három függvény felépítése hasonló, mivel mindhárom a menetterv egy-egy oszlopán vagy során végez műveletet. A központi automata feladatai ellátására három változó lett definiálva. (72. ábra)

```

bool T2_Szabad = false; //a T2 jelző állapotai
int i; //segédváltozó a menettervi függőségekhez
bool menet_ok = true; //vágányút beállítását engedélyezi

```

72. ábra A „Menet\_Elzarasi” automata változói a vágányutas elv UPPAAL modelljében

A korábban említettekkel összhangban a T2 jelű tolatásjelzőt (jelzési képeinek állapotait) egyetlen változó, a „T2\_Szabad” jeleníti meg. Az „i” segédváltozó és a „menet\_ok” változó a menettervi függőségek eldöntésében kapnak szerepet. Az egyes automatákban a nyomvonalas elvhez képest jelentősen kevesebb (maximum négy) változó jelenik meg. (17. táblázat)

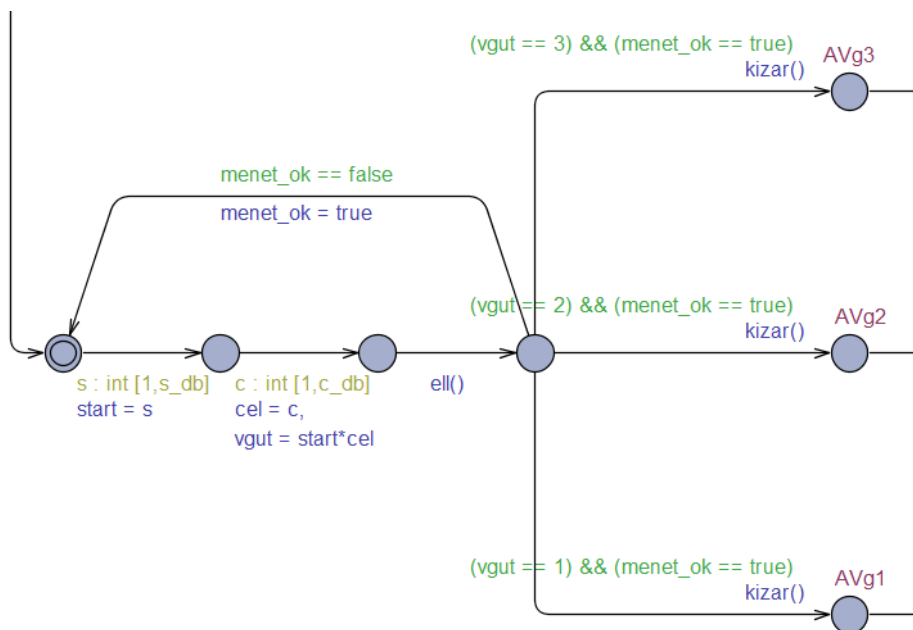
17. táblázat Az egyes automaták és változók a vágányutas elv UPPAAL modelljében

Név	Típus	A változó		Szerepel a következő elemnél	
		Kezdeti érték	Magyarázat	Váltó	Vágány
Lezar	bool	false	Az elem lezárva	X	X
vaganyut	int	0	az adott vágányút számára van lezárva	X	X
Utasit	int [0,2]	0	Utasítás a váltó állására	X	
Allas	int [1,2]	2	A váltó állása	X	

A „Lezar” változó tárolja az információt az elem lezáródásáról, illetve a későbbi oldáshoz kapcsolódóan a „vaganyut” nevű változóban tárolódik el, melyik vágányút számára lett az adott elem lezárva. Az „Utasit” változóban tárolódik el az információ a váltó kívánt állásáról, hasonlóan a nyomvonalas elv „KijSz” változójához. Az „Allas” változó a 3.3.3. fejezetben bemutatott modellel megegyezően a váltó állásának megfelelő értékkel rendelkezik, a vágányutas elvnek megfelelő értelemmel.

### A menettervi függőségek megvalósítása

A vágányút beállítása előbb egy start, majd egy cél a rendelkezésre álló maximumból való véletlenszerű kiválasztásával kezdődik. (73. ábra) A kiválasztott start és cél („start” és „cel”) értékek szorzata adja a beállítandó vágányutat meghatározó értéket („vgut”).



73. ábra Vágányút-kijelölés és menettervi függőségek a vágányutas elv UPPAAL modelljében



A kiválasztott vágányútra ellenőrizni kell, hogy nincs már beállítva a menetterv szerint más ellenséges menet. Ezt az „ell()” függvény végzi el a „menet\_ok” változó segítségével a „fugges” globális tömb alapján. (74. ábra) A „menet\_ok” változó kezdeti értéke true (1). Az „ell()” függvény e változó értékét változtatja úgy, hogy a függés mátrix vágányúthoz tartozó oszlopának minden elemével megszorozza, és a szorzat eredményét tárolja benne. Ez felfogható egyesek és nullák folyamatos szorzásaként, illetve true és false értékeken végzett logikai ÉS műveletekként is. Ha tehát az adott oszlopban minden elem egyes, a „menet\_ok” változó értéke is egy marad. Azonban ha akár egy elem is nulla értékű, a szorzat eredménye is nulla lesz.

```

//menettermi függőség ellenőrzése
void ell(){
for (i = 1; i <= s_db*c_db; i++) {
    menet_ok = ((fugges[i][vgut]) * menet_ok);
}
}

//menettermi függőségek beállítása
void kizar(){
for (i = 1; i <= s_db*c_db; i++) {
    fugges[vgut][i] = (fugges[vgut][i]) * (menetterm[vgut][i]);
}
}

```

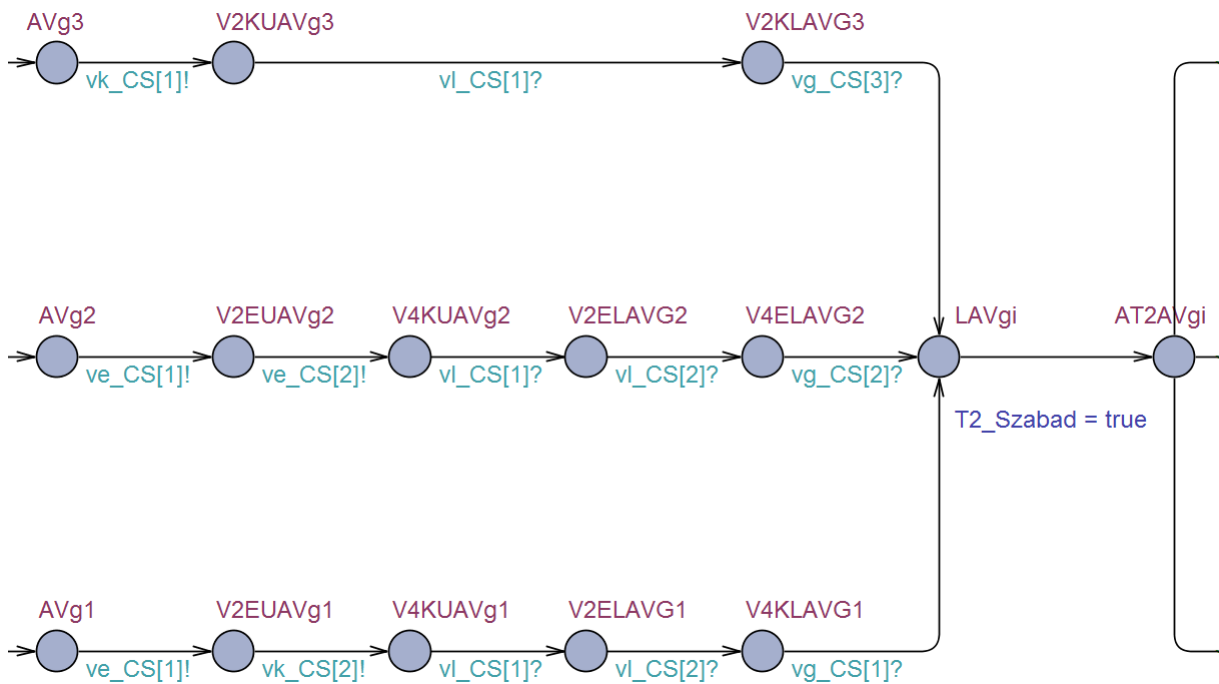
74. ábra A menettermi függőségeket megvalósító függvények a vágányutas elv UPPAAL modelljében

Az „ell()” függvény végrehajtása után a menet beállíthatóságáról döntés születik. Ha a „menet\_ok” változó false értékű, a menet nem állítható be. Ekkor egy állapotátmenet során a változó ismét igaz értéket vesz fel, és az automata a kezdőállapotba kerül vissza.

Ha viszont a vágányút beállítható, azaz a „menet\_ok” változó true értékét megtartotta, az elzárási tervet megvalósító automatarészben a korábban kiválasztott vágányútnak megfelelő állapotba kerül az automata. Ezen állapotátmenetek során végrehajtódik a „kizar()” függvény is, mely a „fugges” tömb vágányútnak megfelelő sorában lévő elemeket páronként összeszorozza a „menetterm” megfelelő sorában lévő értékekkel, és a „fugges” tömbben elmenti. (74. ábra) Ez azt jelenti, hogy minden helyre, ahol a „menetterm” megfelelő helyén nulla szerepel, a „fugges” mátrixba is nulla kerül. Tehát a „fugges” mátrix oszlopai kerülnek ellenőrzésre, és sorai kerülnek módosításra egy vágányút beállítása során.

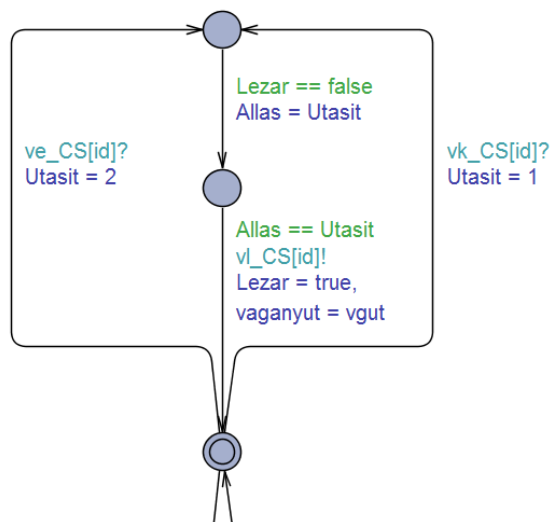
#### **Az elzárási terv függőségeinek megvalósítása**

A menettermi függőségek beállítása után kezdődik az elzárási terv végrehajtása, ami a Petri hálós modellhez hasonlóan történik meg. A vágányút beállítása és lezárásában résztvevő állapotok neve is igazodik a vágányutas elv Petri hálós modelljében alkalmazott helyek nevéhez (ld. 3.3.2. fejezet). Egyetlen eltérés, hogy az UPPAAL modellekben az állapotok neve nem kezdődhet számmal ezért (75. ábra)



75. ábra Az elzárási terv függőségeinek végrehajtása a vágányutas elv UPPAAL modelljében

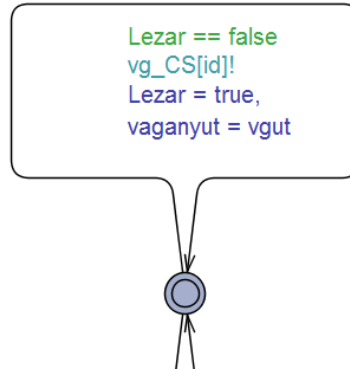
Először a váltó vágányútban elvárt állásának megfelelő csatornán való szinkronizációval ad a „Menet\_Elzarasi” automata váltóállítási-utasítási parancsot a váltó automaták felé. Ezt az információt a „Valto” automaták az „Utasit” változójukban eltárolják. (76. ábra) Ha a váltó nincs lezárva, akkor megtörténhet az állítása, azaz az „Allas” változóban eltárolódik az „Utasit” változó értéke. Ha a váltó állása az utasított állásnak megfelelő, akkor egy állapotátmenet során a váltó objektum automatája szinkronizál a központi logika automatájával és lezáródik. Emellett a későbbi feloldáshoz kapcsolódóan a lokális „vaganyut” változóban eltárolásra kerül a kiválasztott vágányutat jelölő globális „vgut” változó értéke.



76. ábra A „Valto” automata állítása és lezárása a vágányutas elv UPPAAL modelljében

A „Menet\_Elzarasi” automata először szinkronizálással megadja a „Valto” automatáknak az elvárt állásukat, majd az általuk indított szinkronizálással fogadja lezáródási információjukat. Mivel a vágányszakaszok számára állítási utasítás nem értelmezhető, az automaták csak a lezárást tudják vég-

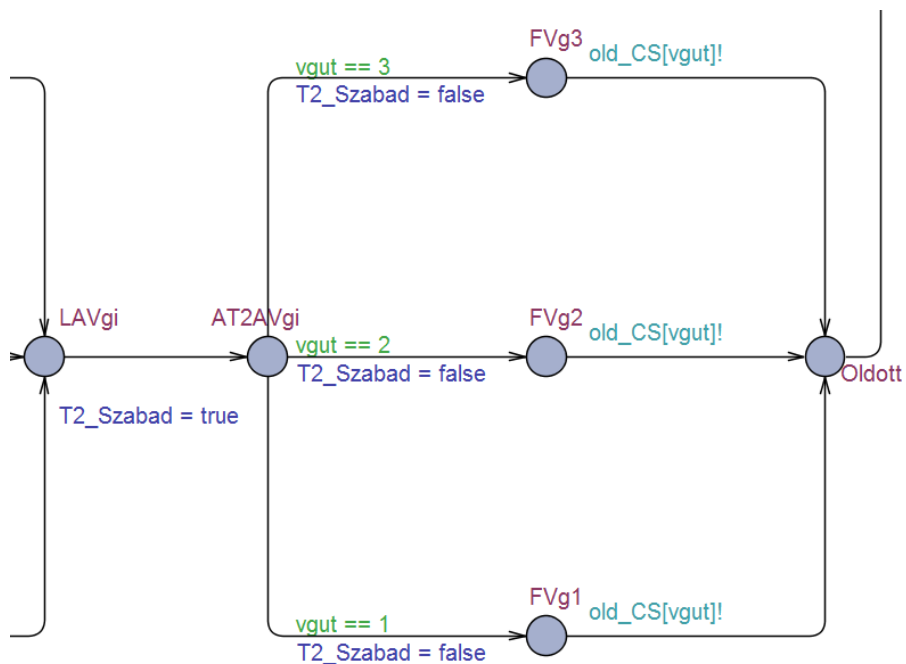
rehajtani. (77. ábra) Ennek feltétele, hogy a vágányszakasz még ne legyen lezárt állapotban. Az automata szinkronizálási kezdeményezését a „Menet\_Elzarasi” automata a váltók beállítása és lezáródása után tudja fogadni. Ebben a lépésben a „Vagany” automata működése a „Valto” automata megfelelő működésével megegyező, tehát lezáródik és eltárolja a kiválasztott vágányút értékét.



77. ábra A „Vagany” automata lezárása a vágányutas elv UPPAAL modelljében

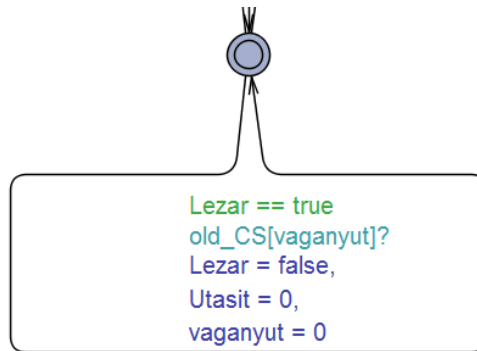
Ha minden vágányútban érintett elem lezáródott, és erről a fent említett módon, szinkronizálással információt adott, lehetőség van a jelző állítására. Az UPPAAL modellben (a 3.2. fejezetnek megfelelően) nem kerül újbóli ellenőrzésre, hogy az egyes elemek megfelelően állnak, és le vannak zárva. Ha az automata tehát elért az „LAVgi” állapotba, a vágányutakhoz tartozó egyetlen T2 jelző állítható, a „T2\_Szabad” változó true értéket vesz fel.

A tolatómenet leközeledésének szimulálása és a vágányút oldása az UPPAAL modellben is oldószakasszal történik. (78. ábra) A jelző „Szabad a tolatás” állásba állítása után ismét több állapotba kerülhet az automata, ami a három vágányhoz tartozó külön oldószakaszok foglalttá válásának feleltethető meg. Foglalttá csak az a szakasz válthat, amely a kiválasztott vágányúthoz tartozik ( $vgut == i$ ), ekkor a tolatásjelző is visszaáll „Tilos a tolatás” állásba (a „T2\_Szabad” false értéket vesz fel).

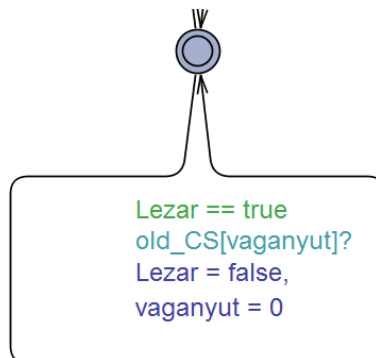


78. ábra Jelzőállítás és vágányút feloldás a vágányutas elv Petri hálós modelljében

Az oldószakasz felszabadulását az „Oldott” állapotba való állapotátmenet képi. Ezen átmenet során a „menet\_Elzarasi” automata egy broadcast csatornán, a globális „vgut” változóban beállított értékkel szinkronizálást kezdeményez. Minden olyan automata részt vesz a szinkronizálásban, amely „vaganyut” nevű változójában a beállított vágányútnak megfelelő értéket tárol, valamint a kezdőállapotban tartózkodik. A szinkronizálás során a „Valto” és „Vagany” automaták működése megegyezik. (79. ábra és 80. ábra) A szinkronizáció feltétele, hogy az adott objektum le legyen zárva. A szinkronizáció során a lezárás feloldódik, valamint a „vaganyut” változó is alapba kerül.



79. ábra A „Valto” automata feloldása a vágányutas elv UPPAAL modelljében



80. ábra A „Vagany” automata feloldása a vágányutas elv UPPAAL modelljében

Az „Oldott” állapot elérése egyben a modellezni kívánt funkciók végét is jelenti, tehát új vágányút beállítását kell lehetővé tenni. Erre egy állapotátmenet lett bevezetve a kezdőállapot irányába, melynek során kerül meghívásra a harmadik, „old()” nevű függvény. (81. ábra)

---

```
old(),
start = 0,
cel = 0,
vgut = 0
```

81. ábra A kezdőállapotba való átmenet a vágányutas elv UPPAAL modelljében

A függvény a „fugges” tömb kiválasztott (beállított, lezárt, majd feloldódott) vágányúthoz tartozó sorában egy for ciklus segítségével minden elemet visszaállít egy értékre. (82. ábra) Az állapotátmenet során ezután a „start”, „cel”, és „vgut” globális változók kinullázása is megtörténik. A „Menet\_Elzarasi” automata új start és cél sorsolásával új vágányút beállítására képes.

```
//menettermi függőség oldása  
void old(){  
for (i = 1; i <= s_db*c_db; i++) {  
    fugges[vgut][i] = 1;  
}  
}
```

82. ábra A menettermi függőségek oldását végző függvény a vágányutas elv UPPAAL modelljében

## 4. Modellellenőrzés

A modellellenőrzést a specifikáció alapján meghatározott követelményekre kell elvégezni (ld. 2.2.2. fejezet). A releváns követelményforrások közül kettőt választottam: az Országos Vasúti Szabályzat I. kötetét [45] és az elektronikus biztosítóberendezésekre vonatkozó feltétfüzetet [46]. (A diplomatervnek nem volt célja a teljes vonatkozó követelményazonosítás és -elemzés.) A kigyűjtött követelmények az állomási berendezések függőségeit általánosan, illetve a tolató-vágányutak függőségeit konkrétan írják le. A [45] dokumentumból származó releváns követelmények (állomási biztosítóberendezésekre):

- „A vágányút lezárásával a berendezéseknek valamennyi érintett váltót és védőváltót a vágányútnak megfelelő végállásban, valamennyi, a védelemben részt vevő vágányzáró sorompót és kisiklasztó sarut a vonatot védő helyzetben rögzíteniük kell, állításukat meg kell akadályozniuk.”
- „A berendezéseknek az egymásra veszélyes vágányutak beállítását vagy lezárását nem szabad megengedniük. A főjelzőn továbbhaladást engedélyező jelzés csak abban az esetben jelenhet meg, ha az ehhez megállapított feltételek teljesültek.”
- „Állomási tolató-vágányutas biztosítóberendezések: a váltók és a tolatási mozgást szabályozó jelzők között olyan szerkezeti függés van, amely a tolatási mozgást szabályozó jelzőn csak abban az esetben enged mozgást engedélyező jelzést megjeleníteni, ha a tolatás részére beállított vágányút lezárása megtörtént.”

A [46] dokumentumból származó releváns követelmények:

„Tolatóvágányutak függőségei

Tolatóvágányutak beállításakor a berendezésnek az alábbi függőségi feltételek teljesülését kell biztosítani, ill. ellenőriznie:

- a. az érintett váltók (vágányzáró szerkezetek) helyes állása és lezárása,
- b. a védőváltók (vágányzárók), ill. az oldalvédelmet szolgáló jelzők helyes állása és lezárása,
- c. a közúti sorompók lecsukott és lezárt állapota,
- d. az oldalvédelmet biztosító szigetelt szakaszok szabad állapota,
- e. a forgalmi szolgálat –helyi viszonyokat figyelembe vevő- esetenkénti döntése alapján az érintett szigetelt szakaszok szabad állapota,
- f. tiltott egyidejű menetek kizárása,
- g. a céljelző jelzési képének (legalább a kék, ill. vörös fény meglétének) ellenőrzése,
- h. a céljelző utáni szigetelt szakasz(ok) szabad állapota, ha azt az előterv előírja.

Az alpontokban előírt feltételek teljesülését az e. feltétel kivételével folyamatosan ellenőrizni kell. Ha valamelyik feltétel nem teljesül (pl. valamelyik oldalvédelmi szakasz foglalt lesz), akkor a jelzőnek továbbhaladást megtiltó – főjelző esetén „Megállj!”, tolatásjelző esetében „Tilos a tolatás” – jelzésre kell kapcsolódnia. A kizárásoknak az utóbbi esetben sem szabad feloldódniuk.”

A modellellenőrzéshez a formális követelményeket a fenti informális követelményekből kiindulva fogalmaztam meg. A 3.2. fejezetben ismertetett elhanyagolások, egyszerűsítések okán a követelményeknek is csupán egy részét kell alkalmazni. Nem vizsgáltam tehát a helyszínrajzban, lehatárolásban nem szereplő elemekre vonatkozó követelményeket:

- Oldalvédelmi elemek megfelelő, lezárt állapota, szakaszok szabad volta.

- Közúti sorompókkal kapcsolatos követelmények.
- Céljelzőre, céljelző mögötti szakaszra vonatkozó követelmények.

Ezekon felül a tiltott egyidejű menetek kizárására vonatkozó követelmények sem vizsgálandók, a következő két okból kifolyólag:

- A modellekben egyszerre csak egy vágányút állítható be, így nem vizsgálható nem megengedett menet beállításának lehetősége sem.
- A helyszínrajzból adódóan csak egyszerű kizárások (2.1.2. fejezet alapján) létesítendőek, amiket a váltók biztosítanak (csak egy vágányút számára állhat jól). Annak vizsgálata, hogy a váltó egy lezárt vágányút során ne legyen átállítható, a továbbiakban ismertetettek szerint megtörténik.

Az elsődlegesen vizsgálandó követelmények tehát a váltók helyes állására és lezárására, valamint a jelző továbbhaladást engedélyező állásba állítására vonatkoznak. Ezek a következő általános érvényű követelményekké fogalmazhatók át:

A váltók és a jelzők között szerkezeti függések vannak, oly módon, hogy:

- A váltót csak akkor lehet feloldani és átállítani, ha az őt fedező jelző „Megállj!” állásban van.
- A jelzőt csak akkor lehet „Megállj!” állásból szabad állásba állítani, ha az általa fedezett váltók helyesen állnak és ebben az állásban rögzítve vannak.

A modellellenőrzés elsődleges célja e két biztonsági követelmény ellenőrzése. Emellett azonban megvizsgáltam egyéb funkcionális, illetve a modell működése szempontjából releváns követelményeket (pl. holtpontmentesség, új vágányút beállításának lehetősége). A 4.1. fejezet bemutatja a modellellenőrzés folyamatát és a felhasznált kifejezéseket, eredményeket.

#### 4.1. A modellellenőrzés megvalósítása

A vasúti biztosítóberendezések biztonsági elvét leíró két fő állítást ellenőrizhető követelményekké kell átfogalmazni. Mivel mindkét állítás „ha ..., akkor ...” jellegű, logikai implikáció formájában írhatóak le formálisan. Eszerint a logikai műveleti jelekkel átírt állítások:

- (ha a jelző „Megállj!” állásban áll)  $\rightarrow ((a \text{ váltók feloldhatók}) \wedge (a \text{ váltók állíthatók}))$
- (ha a jelző szabad állásba állítható)  $\rightarrow ((a \text{ váltók a kiválasztott vágányútnak megfelelően állnak}) \wedge (a \text{ váltók le vannak zárva}))$

Az az állítás, hogy a váltó állítható, csak adott körülmények között ellenőrizhető. Ennek oka, hogy a modellekben nincs egyéni, csak vágányutas váltóállítási lehetőség. Tehát a váltó állításának különböző feltételei is vannak. A modell alapvető működése esetén a váltó valamilyen formában parancsot kap az állításra (nyomvonalas elv esetén két kijelölés, vágányutas elv esetén állítási utasítás). Állításra akkor van lehetőség, ha a váltó nem foglalt (csak nyomvonalas elv esetén), illetve nincs lezárva. Ennek megfelelően megkülönböztethetünk olyan feltételeket, amikor a váltó állítandó (kijelölés/utasítás) illetve állítható (foglaltság, lezárás). Az állíthatóság így csak az állítandóság teljesülésekor értelmezhető. Mivel váltófoglaltság jelen modellben csak a tolatómenet leközeledésekor (és csak a nyomvonalas elvű modellben) fordulhat elő, így amikor a váltó foglalt, akkor lezárva kell lennie. Ennek következtében, ha a váltó nincs lezárva, akkor nem lehet foglalt (meghibásodásokkal, így hamis foglaltsággal a modell nem számol, ld. 3.2. fejezet). Ebből adódik, hogy a váltó állíthatósági feltételeként elegendő vizsgálni azt a feltételt, hogy a váltó nincs lezárva. (18. táblázat, 1. állítás)

A jelző állás és a váltó feloldhatóság közötti két feltétel kezelése a következőképp, összevontan történik. Mind a jelzőnek („Megállj!” és szabad), mind a váltónak (lezárt, és feloldott) csak két különböző állapota lehet. Ezek között a következő feltételek állnak fenn:

- Ha egy váltó oldott állapotban van, akkor a hozzá tartozó jelző csak „Megállj!” állásban lehet.
- Ha a jelző szabad állásban áll, akkor a váltó csak lezárt állapotban lehet.

Ennek következtében ellenőrizendő követelménynek elegendőnek tekinthető az az állítás, hogy amennyiben a jelző szabad állásban áll, a jelzőhöz tartozó beállított vágányútban fekvő valamennyi váltó a vágányútnak megfelelő állásban áll, és le van zárva. Ezzel egyben megvizsgálásra kerül az a feltétel is, hogy a vágányútban érintett váltók nem állíthatók. (18. táblázat, 2. állítás)

Emellett azonban meg kell vizsgálni, hogy nem állhatnak fenn olyan érvénytelen állapotok, hogy a fent említett két állapottal rendelkező elemek mindkét, esetleg egy nem feltételezett állásban állnak. (18. táblázat, 3-5. állítások) Továbbá, mivel a váltók állását egy kiválasztott vágányúthoz kell ellenőrizni, bizonyítani kell, hogy ha jelző szabad, akkor valamelyik hozzá tartozó cél/célvágány ki van jelölve/le van zárva. Így ellenőrizhető a startjelző és a célpont közötti váltók megfelelő állása. (18. táblázat, 6. állítás) A 19. táblázatban, a 20. táblázatban, a 21. táblázatban és a 22. táblázatban a kifejezések pontosítása látható modellenként. A szürkített rész a nem ellenőrizhető követelményeket jelenti.

18. táblázat Az ellenőrizendő követelmények

Sorszám	Kifejezés
1. állítás	Ha a váltó foglalt, akkor le van zárva.
2. állítás	Ha a jelző szabad állásban áll, akkor a váltók a kiválasztott vágányútnak megfelelő állásban állnak és le van zárva.
3. állítás	A váltó csak bal vagy jobb illetve csak egyenes vagy kitérő állásban állhat.
4. állítás	A váltó csak lezárt, vagy le nem zárt (feloldott) állapotban lehet.
5. állítás	A jelző csak „Megállj!” vagy szabad állapotban lehet.
6. állítás	Ha a jelző szabad állásban áll, akkor van hozzá tartozó cél/vágányút kiválasztva.

19. táblázat Az ellenőrizendő követelmények pontosítása nyomvonalas elv, Petri hálós modell esetén

Áll.	Nyomvonalas elv, Petri hálós modell
1.	$P_{Fx} > 0 \rightarrow P_{Lx} > 0$
2.	$P_{T2Sz} > 0 \rightarrow ((P_{xJ}/B > 0) \wedge (P_{Lx} > 0))$
3.	$((P_{xJ} > 0) \wedge (P_{xB} = 0)) \vee ((P_{xJ} = 0) \wedge (P_{xB} > 0))$
4.	$((P_{Lx} > 0) \wedge (P_{LxP} = 0)) \vee ((P_{Lx} = 0) \wedge (P_{LxP} > 0))$
5.	$((P_{T2M} > 0) \wedge (P_{T2Sz} = 0)) \vee ((P_{T2M} = 0) \wedge (P_{T2Sz} > 0))$
6.	$P_{T2Sz} > 0 \rightarrow ((P_{LVg1} > 0) \vee (P_{LVg2} > 0) \vee (P_{LVg3} > 0))$

20. táblázat Az ellenőrizendő követelmények pontosítása vágányutas elv, Petri hálós modell esetén

Áll.	Vágányutas elv, Petri hálós modell
1.	
2.	$P_{T2Sz} > 0 \rightarrow ((P_{xE}/K > 0) \wedge (P_{Lx} > 0))$
3.	$((P_{xE} > 0) \wedge (P_{xK} = 0)) \vee ((P_{xE} = 0) \wedge (P_{xK} > 0))$
4.	$((P_{xL} > 0) \wedge (P_{xLP} = 0)) \vee ((P_{xL} = 0) \wedge (P_{xLP} > 0))$
5.	$((P_{T2M} > 0) \wedge (P_{T2Sz} = 0)) \vee ((P_{T2M} = 0) \wedge (P_{T2Sz} > 0))$
6.	$P_{T2Sz} > 0 \rightarrow ((P_{LAVg1} > 0) \vee (P_{LAVg2} > 0) \vee (P_{LAVg3} > 0))$



21. táblázat Az ellenőrizendő követelmények pontosítása nyomvonalas elv, UPPAAL modell esetén

Áll.	Nyomvonalas elv, UPPAAL modell
1.	$Valto(i).Foglalt = true \rightarrow Valto(i).Lezar = true$
2.	$Start(1).Szabad = true \rightarrow ((Valto(i).Allas = 1/2) \wedge (Valto(i).Lezar = true))$
3.	$(Valto(i).Allas = 1) \vee (Valto(i).Allas = 2)$
4.	$(Valto(i).Lezar = 0) \vee (Valto(i).Lezar = 1)$
5.	$(Start(1).Szabad = 0) \vee (Start(1).Szabad = 1)$
6.	$Start(1).Szabad = true \rightarrow 0 < cel \leq 3$

22. táblázat Az ellenőrizendő követelmények pontosítása vágányutas elv, UPPAAL modell esetén

Áll.	Vágányutas elv, UPPAAL modell
1.	
2.	$T2\_Szabad = true \rightarrow ((Valto(i).Allas = 1/2) \wedge (Valto(i).Lezar = true))$
3.	$(Valto(i).Allas = 1) \vee (Valto(i).Allas = 2)$
4.	$(Valto(i).Lezar = 0) \vee (Valto(i).Lezar = 1)$
5.	$(T2\_Szabad = 0) \vee (T2\_Szabad = 1)$
6.	$T2\_Szabad = true \rightarrow 0 < vgut \leq 3$

A 23. táblázat, a 24. táblázat, a 25. táblázat, és a 26. táblázat jeleníti meg a formális kifejezésekkel átfogalmazott, és a modellellenőrzésben ténylegesen használt követelményeket.

23. táblázat Az ellenőrzött formális követelmények nyomvonalas elv, Petri hálós modell esetén

Áll.	Objektum/ Vágányút	CTL kifejezés Nyomvonalas elv, Petri hálós modell
1.	2-es váltó	$AG(!((K\_L\_T.P\_F2>0)   (K\_L\_T.P\_L2>0)))$
	4-es váltó	$AG(!((K\_L\_T.P\_F4>0)   (K\_L\_T.P\_L4>0)))$
2.	A-Vg. I.	$AG(!(((K\_L\_T.P\_T2Sz>0) \& (K\_L\_T.P\_LVg1>0))   (((K\_L\_T.P\_2J>0) \& (K\_L\_T.P\_L2>0) \& (K\_L\_T.P\_4J>0) \& (K\_L\_T.P\_L4>0))))))$
	A-Vg. II.	$AG(!(((K\_L\_T.P\_T2Sz>0) \& (K\_L\_T.P\_LVg2>0))   (((K\_L\_T.P\_2J>0) \& (K\_L\_T.P\_L2>0) \& (K\_L\_T.P\_4B>0) \& (K\_L\_T.P\_L4>0))))))$
	A-Vg. III.	$AG(!(((K\_L\_T.P\_T2Sz>0) \& (K\_L\_T.P\_LVg3>0))   (((K\_L\_T.P\_2B>0) \& (K\_L\_T.P\_L2>0))))))$
3.	2-es váltó	$AG(((K\_L\_T.P\_2B>0) \& (K\_L\_T.P\_2J=0))   (((K\_L\_T.P\_2B=0) \& (K\_L\_T.P\_2J>0))))$
	4-es váltó	$AG(((K\_L\_T.P\_4B>0) \& (K\_L\_T.P\_4J=0))   (((K\_L\_T.P\_4B=0) \& (K\_L\_T.P\_4J>0))))$
4.	2-es váltó	$AG(((K\_L\_T.P\_L2=0) \& (K\_L\_T.P\_L2P>0))   (((K\_L\_T.P\_L2>0) \& (K\_L\_T.P\_L2P=0))))$
	4-es váltó	$AG(((K\_L\_T.P\_L4=0) \& (K\_L\_T.P\_L4P>0))   (((K\_L\_T.P\_L4>0) \& (K\_L\_T.P\_L4P=0))))$
5.	T2 jelző	$AG(((K\_L\_T.P\_T2Sz>0) \& (K\_L\_T.P\_T2M=0))   (((K\_L\_T.P\_T2M>0) \& (K\_L\_T.P\_T2Sz=0))))$
6.	T2 jelző	$AG(!((K\_L\_T.P\_T2Sz>0))   (((K\_L\_T.P\_LVg1>0) \& (K\_L\_T.P\_LVg2=0) \& (K\_L\_T.P\_LVg3=0))   (((K\_L\_T.P\_LVg1=0) \& (K\_L\_T.P\_LVg2>0) \& (K\_L\_T.P\_LVg3=0))   (((K\_L\_T.P\_LVg1=0) \& (K\_L\_T.P\_LVg2=0) \& (K\_L\_T.P\_LVg3>0))))))$

24. táblázat Az ellenőrzött formális követelmények vágányutas elv, Petri hálós modell esetén

Áll.	Objektum/ Vágányút	CTL kifejezés Vágányutas elv, Petri hálós modell
1.	2-es váltó	
	4-es váltó	
2.	A-Vg. I.	AG(!((Menet_Elzarasi.P_T2Sz>0)&(Menet_Elzarasi.P_LAVg1>0))   ((Menet_Elzarasi.P_2E>0)&(Menet_Elzarasi.P_2L>0)& (Menet_Elzarasi.P_4K>0)&(Menet_Elzarasi.P_4L>0)))
	A-Vg. II.	AG(!((Menet_Elzarasi.P_T2Sz>0)&(Menet_Elzarasi.P_LAVg2>0))   ((Menet_Elzarasi.P_2E>0)&(Menet_Elzarasi.P_2L>0)& (Menet_Elzarasi.P_4E>0)&(Menet_Elzarasi.P_4L>0)))
	A-Vg. III.	AG(!((Menet_Elzarasi.P_T2Sz>0)&(Menet_Elzarasi.P_LAVg3>0))   ((Menet_Elzarasi.P_2K>0)&(Menet_Elzarasi.P_2L>0)))
3.	2-es váltó	AG(((Menet_Elzarasi.P_2E>0)&(Menet_Elzarasi.P_2K=0))   ((Menet_Elzarasi.P_2E=0)&(Menet_Elzarasi.P_2K>0)))
	4-es váltó	AG(((Menet_Elzarasi.P_4E>0)&(Menet_Elzarasi.P_4K=0))   ((Menet_Elzarasi.P_4E=0)&(Menet_Elzarasi.P_4K>0)))
4.	2-es váltó	AG(((Menet_Elzarasi.P_2L=0)&(Menet_Elzarasi.P_2LP>0))   ((Menet_Elzarasi.P_2L>0)&(Menet_Elzarasi.P_2LP=0)))
	4-es váltó	AG(((Menet_Elzarasi.P_4L=0)&(Menet_Elzarasi.P_4LP>0))   ((Menet_Elzarasi.P_4L>0)&(Menet_Elzarasi.P_4LP=0)))
5.	T2 jelző	AG(((Menet_Elzarasi.P_T2Sz>0)&(Menet_Elzarasi.P_T2M=0))   ((Menet_Elzarasi.P_T2M>0)&(Menet_Elzarasi.P_T2Sz=0)))
6.	T2 jelző	AG(!((Menet_Elzarasi.P_T2Sz>0))   (((Menet_Elzarasi.P_LAVg1>0)& (Menet_Elzarasi.P_LAVg2=0)&(Menet_Elzarasi.P_LAVg3=0))   ((Menet_Elzarasi.P_LAVg1=0)&(Menet_Elzarasi.P_LAVg2>0)& (Menet_Elzarasi.P_LAVg3=0))   ((Menet_Elzarasi.P_LAVg1=0)& (Menet_Elzarasi.P_LAVg2=0)&(Menet_Elzarasi.P_LAVg3>0))))

25. táblázat Az ellenőrzött formális követelmények nyomvonalas elv, UPPAAL modell esetén

Áll.	Objektum/ Vágányút	CTL kifejezés Nyomvonalas elv, UPPAAL modell
1.	2-es váltó	A[]((Valto(1).Foglalt!=true)   (Valto(1).Lezar==true))
	4-es váltó	A[]((Valto(2).Foglalt!=true)   (Valto(2).Lezar==true))
2.	A-Vg. I.	A[](!((Start(1).Szabad==1)&&(cel==1))   ((Valto(1).Allas==2)&& (Valto(1).Lezar==1)&&(Valto(2).Allas==2)&&(Valto(2).Lezar==1)))
	A-Vg. II.	A[](!((Start(1).Szabad==1)&&(cel==2))   ((Valto(1).Allas==2)&& (Valto(1).Lezar==1)&&(Valto(2).Allas==1)&&(Valto(2).Lezar==1)))
	A-Vg. III.	A[](!((Start(1).Szabad==1)&&(cel==3))   ((Valto(1).Allas==1)&& (Valto(1).Lezar==1)))
3.	2-es váltó	A[]((Valto(1).Allas==1)   (Valto(1).Allas==2))
	4-es váltó	A[]((Valto(2).Allas==1)   (Valto(2).Allas==2))
4.	2-es váltó	A[]((Valto(1).Lezar==0)   (Valto(1).Lezar==1))
	4-es váltó	A[]((Valto(2).Lezar==0)   (Valto(2).Lezar==1))
5.	T2 jelző	A[]((Start(1).Szabad==0)   (Start(1).Szabad==1))
6.	T2 jelző	A[](!((Start(1).Szabad==1))   (cel>>0)))

26. táblázat Az ellenőrzött formális követelmények vágányutas elv, UPPAAL modell esetén

Áll.	Objektum/ Vágányút	CTL kifejezés Vágányutas elv, UPPAAL modell
1.	2-es váltó	
	4-es váltó	
2.	A-Vg. I.	$A[](((Menet\_Elzarasi.T2\_Szabad==1)\&\&(cel==1)) \vee ((Valto(1).Allas==2)\&\&(Valto(1).Lezar==1)\&\&(Valto(2).Allas==1)\&\&(Valto(2).Lezar==1)))$
	A-Vg. II.	$A[](((Menet\_Elzarasi.T2\_Szabad==1)\&\&(cel==2)) \vee ((Valto(1).Allas==2)\&\&(Valto(1).Lezar==1)\&\&(Valto(2).Allas==2)\&\&(Valto(2).Lezar==1)))$
	A-Vg. III.	$A[](((Menet\_Elzarasi.T2\_Szabad==1)\&\&(cel==3)) \vee ((Valto(1).Allas==1)\&\&(Valto(1).Lezar==1)))$
3.	2-es váltó	$A[]((Valto(1).Allas==1) \vee (Valto(1).Allas==2))$
	4-es váltó	$A[]((Valto(2).Allas==1) \vee (Valto(2).Allas==2))$
4.	2-es váltó	$A[]((Valto(1).Lezar==0) \vee (Valto(1).Lezar==1))$
	4-es váltó	$A[]((Valto(2).Lezar==0) \vee (Valto(2).Lezar==1))$
5.	T2 jelző	$A[]((Menet\_Elzarasi.T2\_Szabad==0) \vee (Menet\_Elzarasi.T2\_Szabad==1))$
6.	T2 jelző	$A[](((!(Menet\_Elzarasi.T2\_Szabad==1)) \vee (cel>>0)))$

A modellellenőrzés eredménye a 27. táblázatban látható. Minden követelmény teljesült, tehát az elkészült modellek valóban modelljei a vasúti biztosítóberendezés alapvető működését leíró követelményeknek.

27. táblázat A biztonsági követelményekre elvégzett ellenőrzések teljesülése

Áll.	Objektum/ Vágányút	Petri háló		UPPAL	
		nyomvonalas	vágányutas	nyomvonalas	vágányutas
1.	2-es váltó	true		true	
	4-es váltó	true		true	
2.	A-Vg. I.	true	true	true	true
	A-Vg. II.	true	true	true	true
	A-Vg. III.	true	true	true	true
3.	2-es váltó	true	true	true	true
	4-es váltó	true	true	true	true
4.	2-es váltó	true	true	true	true
	4-es váltó	true	true	true	true
5.	T2 jelző	true	true	true	true
6.	T2 jelző	true	true	true	true

Az alapvető biztonsági követelmények teljesítése mellett a modellektől elvárt további funkcionális követelményeknek való megfelelés is. A modellellenőrzés a következő egyéb tulajdonságok vizsgálatára terjedt ki (a formális követelmények a 28. táblázatban és a 29. táblázatban, az eredmények a 30. táblázatban láthatók):

- Holtpontmentesség
- Korlátosság
- Adott visszatérő állapot
  - Mindig lehet új vágányutat beállítani
  - Mindig lehet a jelzőt szabadra állítani
  - A vágányút meghatározásakor az objektumok alapállapotban vannak

A PetriDotNet eszköz a holtpontmentesség, korlátosság, megfordíthatóság tulajdonságok teljesülését egyszerűbb hálóknál esetén CTL kifejezések használata nélkül a 2.2.7. fejezetben leírtak szerint tudja megjeleníteni. Jelen összetettebb modellek ellenőrzése során a vizsgálat nem jár eredménnyel, a különböző tulajdonságokra „In progress” választ ad. Egyedül a vágányutas modellben mutatta ki a korlátosságot. A holtpontmentesség CTL kifejezés segítségével PetriDotNet eszközben nem vizsgálható. A korlátosság vizsgálatát helyenként el lehet végezni az  $AG(P_x \leq 1)$  kifejezés használatával. Minden helyre vizsgálva, hogy az ott lévő tokenek száma legfeljebb egy, a korlátosságnál megszorított tulajdonság, a biztosság vizsgálható. A modellellenőrzés eredménye szerint mindkét modell biztos.

Időzített automaták esetén a korlátosság a Petri hálókkal megegyezően nem értelmezhető. Ugyanakkor a végesség belátható, ha a rendszer véges sok állapotban lehet, és a változók is véges értékkel rendelkeznek. Az UPPAAL eszköz modellellenőrző részében a holtpontmentesség az  $A[] \text{not deadlock}$  egyszerű CTL kifejezéssel vizsgálható. Az ellenőrzés mindkét modellre true értéket adott.

28. táblázat A további ellenőrzött formális követelmények Petri hálós modellek esetén

Tulajdonság	Petri háló	
	nyomvonalas	vágányutas
Biztosság	AG(K_L_T.P_x<=1) AG(A2.P_x<=1) AG(A4.P_x<=1) AG(F.P_x<=1)	AG(Menet_Elzarasi.P_x<=1) AG(A2.P_x<=1) AG(A4.P_x<=1)
Mindig lehet újabb vágányutat beállítani	AG(EF(K_L_T.P_Kezel>0))	AG(EF(Menet_Elzarasi.P_Kezel>0))
Mindig lehet szabadra állítani a jelzőt	AG(EF(K_L_T.P_T2Sz>0))	AG(EF(Menet_Elzarasi.P_T2Sz>0))
A vágányutak meghatározásakor az objektumok alapállapotban vannak	AG(!((K_L_T.P_Kezel>0))   ((K_L_T.P_L2=0)&(K_L_T.P_L4=0)&(K_L_T.P_LVg1=0)&(K_L_T.P_LVg2=0)&(K_L_T.P_LVg3=0)&(K_L_T.P_KS2E=0)&(K_L_T.P_KS4E=0)&(K_L_T.P_KSVg1=0)&(K_L_T.P_KSVg2=0)&(K_L_T.P_KSVg3=0)&(K_L_T.P_KC2J=0)&(K_L_T.P_KC2B=0)&(K_L_T.P_KC4J=0)&(K_L_T.P_KC4B=0)&(K_L_T.P_KCVg1=0)&(K_L_T.P_KCVg2=0)&(K_L_T.P_KCVg3=0)&(K_L_T.P_T2Sz=0)))	AG(!((Menet_Elzarasi.P_Kezel>0))   ((Menet_Elzarasi.P_2L=0)&(Menet_Elzarasi.P_4L=0)&(Menet_Elzarasi.P_Vg1L=0)&(Menet_Elzarasi.P_Vg2L=0)&(Menet_Elzarasi.P_Vg3L=0)&(Menet_Elzarasi.P_2EU=0)&(Menet_Elzarasi.P_2KU=0)&(Menet_Elzarasi.P_4EU=0)&(Menet_Elzarasi.P_4KU=0)&(Menet_Elzarasi.P_LAVg1=0)&(Menet_Elzarasi.P_LAVg2=0)&(Menet_Elzarasi.P_LAVg3=0)&(Menet_Elzarasi.P_T2Sz=0)))

Egy adott állapot visszatérése helyett a Petri hálós modellek esetében egy lazább követelmény kerül ellenőrzésre. Mint a 3.2 fejezetben a modellezési döntések bemutatása során említésre került, egy vágányút feloldódása után új vágányút kiválasztására van lehetőség. Ennek az állapotnak az elérése

viszont nem feltétlenül jelenti azt, hogy a rendszer a kezdőállapotába került vissza, mivel a váltoálásoknak nincs kitüntetett kezdőállapota. A vizsgálat így csak arra terjed ki, hogy bármely állapotból elérhető olyan állapot, amelyikben új vágányút jelölhető ki. PetriDotNet eszköz használata esetén két operátor együttes  $AG(EF(P_x > 0))$  formátumú alkalmazásával ellenőrizhetők a követelmények.

29. táblázat A további ellenőrzött formális követelmények UPPAAL modellek esetén

Tulajdonság	UPPAAL	
	nyomvonalas	vágányutas
Holtpontmentesség	A[] not deadlock	A[] not deadlock
A vágányutak meghatározásakor az objektumok alapállapotban vannak	A[!(!(start==0)&&(cel==0)))]   ((Start(1).Lezar==0)&&(Valto(1).Lezar==0)&&(Valto(2).Lezar==0)&&(Vagany(1).Lezar==0)&&(Vagany(2).Lezar==0)&&(Vagany(3).Lezar==0)&&(Cel(1).Lezar==0)&&(Cel(2).Lezar==0)&&(Cel(3).Lezar==0)&&(Start(1).KijS==0)&&(Valto(1).KijS==0)&&(Valto(2).KijS==0)&&(Vagany(1).KijS==0)&&(Vagany(2).KijS==0)&&(Vagany(3).KijS==0)&&(Cel(1).KijS==0)&&(Cel(2).KijS==0)&&(Cel(3).KijS==0)&&(Start(1).KijC==0)&&(Valto(1).KijC==0)&&(Valto(2).KijC==0)&&(Vagany(1).KijC==0)&&(Vagany(2).KijC==0)&&(Vagany(3).KijC==0)&&(Cel(1).KijC==0)&&(Cel(2).KijC==0)&&(Cel(3).KijC==0)&&(Valto(1).KijE==0)&&(Valto(2).KijE==0)&&(Valto(1).KijSz==0)&&(Valto(2).KijSz==0)&&(Start(1).Szabad==0)))	A[!(!(start==0)&&(cel==0)))]   ((Valto(1).Lezar==0)&&(Valto(2).Lezar==0)&&(Vagany(1).Lezar==0)&&(Vagany(2).Lezar==0)&&(Vagany(3).Lezar==0)&&(Valto(1).Utasit==0)&&(Valto(2).Utasit==0)&&(Menet_Elzarasi.T2_Szabad==0)))

Az UPPAAL modellellenőrzője nem engedélyezi operátorok kettőzését, így a PetriDotNet eszközben vizsgált kifejezések egyszerű átírásai nem használhatók. Ehelyett létezik a 2.2.7. fejezetben leírtaknak megfelelően a „leads to” kifejezés, amely elérhetőség vizsgálatára használható lenne. Az ellenőrizendő követelmények azok, hogy a vágányút kijelölésétől mindig el lehet jutni a jelzőállításhoz, illetve a jelző szabadra állítástól ismét a vágányút kijelölésig. Az ellenőrzés végrehajtása azonban nem jár releváns eredménnyel. Mivel időzített automaták esetén egy megadott állapotban maradás, a várakozás

zás is állapotátmenetnek számít, az ellenőrzés eredménye nem teljesült lesz. A generált ellenpélda pedig a kiinduló állapotban maradás, és az ellenőrző nem keres tovább. Megoldás lenne minden helyet órával és invariánssal ellátni, így biztosítva, hogy egy adott állapotból előbb-utóbb továbblépjen az automata. Jelen modellezés során csak a biztosítóberendezések állapotainak leképezése volt a cél, az időbeliség kezelése nem.

Megvizsgálható feltétel, hogy a tolatómenet leközlekedése után, új menet beállítása előtt valóban minden vágányúti elem kijelölése és lezárása megszűnt. Ez egy egyszerű, a biztonsági követelmények ellenőrzéskor már használt és bemutatott „ha ..., akkor ...” jellegű kifejezés, mely mindkét eszközben vizsgálható. A formula azt vizsgálja, hogy a vágányút-kijelölés kezdetén egyik start, váltó, vágány és cél elem sincs lezárva, kijelölve start és cél felől, illetve a tolatásjelző „Tilos a tolatás állásban” áll. A vágányút kijelölésének kezdete a Petri hálós modellekben a „P\_Kezel” helyen lévő token, az UPPAAL modellekben pedig a „start” és „cel” változók nulla értéke.

30. táblázat A további követelményekre elvégzett ellenőrzések teljesülése

Tulajdonság	Petri háló		UPPAAL	
	nyomvonalas	vágányutas	nyomvonalas	vágányutas
Holtpontmentesség			true	true
Biztosság	true	true		
Mindig lehet újabb vágányutat beállítani	true	true		
Mindig lehet szabadra állítani a jelzőt	true	true		
A vágányutak meghatározásakor az objektumok alapállapotban vannak	true	true	true	true

Amely kifejezésekre a modellellenőrzés elvégezhető volt, mindre true értéket adott a vizsgálat. Tehát e követelmények modellezése is megfelelően történt.

## 5. Tanulságok, tapasztalatok

Jelen fejezet célja értékelni és összehasonlítani az elkészült modelleket, az alkalmazott formalizmusokat és a vágányúti logika leképezésének két fő elvét. Itt történik az elvégzett modellezési és modellellenőrzési folyamat során szerzett tapasztalatok megosztása, valamint az igénybe vett eszközök előnyeinek és hátrányainak leírása. A fejezet a modellek továbbfejlesztésének lehetőségeivel zárul.

Jelen fejezet célja értékelni és összehasonlítani az elkészült modelleket, az alkalmazott formalizmusokat, a vágányúti logika leképezésének két fő elvét, az elvégzett modellezési és modellellenőrzési folyamat során szerzett tapasztalatokat, az igénybe vett modellező és modellellenőrző eszközök előnyeit és hátrányait. A fejezet a modellek továbbfejlesztésének lehetőségeivel zárul.

Formális modellezés vasúti biztosítóberendezések területén történő alkalmazásának megvizsgálásához megfelelő formalizmust és vizsgálandó rendszert kellett találni. Fontos szempont volt lehetőleg több formális modellező eszköz több, de hasonló rendszeren történő alkalmazása. A biztosítóberendezések legfontosabb alkalmazásai az állomási berendezések, melyek esetében a két eltérő szerkesztési elv jellemzőinek modellezésével lehetőség nyílt két hasonló megközelítés vizsgálatára. Modellezett rendszernek egy olyan alkalmazást (helyszínrajz, vágányutak) választottam, mely az elhanyagolások ellenére a legfontosabb funkciókat lefedi.

A modellezésben Petri hálók és automaták használata mellett döntöttem annak figyelembevételével, hogy a modellezési munka mennyisége, minősége, hatékonysága összehasonlítható legyen az alapvetően két különböző elméleti háttérrel rendelkező modellező eszköz szemszögéből. A modellezési folyamat egyik első lépése volt a modellező eszközök megválasztása. A Petri hálók vizsgálatára korábbról már ismert PetriDotNet mellett megvizsgáltam a Snoopy/Charlie eszközöket is, majd a PetriDotNet további használata mellett döntöttem. Az automaták modellezésére az UPPAAL-t választottam, melyet korábban szintén használtam. A két eszköz előnye, hogy beépített modellellenőrzővel rendelkeznek, melyekben közel azonos szintakszis alkalmazásával végezhető el a modellellenőrzés is.

A fejlesztési folyamat során először a Petri hálós, és a nyomvonalas elvű modellek készültek el. A vágányutas modellek megvalósítása a két eszközben közel hasonló munka befektetése mellett volt lehetséges. A nyomvonalas elv leképezése az UPPAAL eszközben több ráfordítást, módosítgatást igényelt (ennek egyik lehetséges oka a Petri hálók jobb átláthatósága). Ugyanakkor a nyomvonalas elv nem külön objektumként történő modellezése miatt a modell egy korábbi szakaszban vágányutas tulajdonságokkal is rendelkezett, melyek felfedése és kijavítása utólagos többletmunkával járt.

Modellellenőrzésre mindkét eszköz CTL-t (is) használ. Azonban a két beépített modellellenőrző szemantikájában, alkalmazható operátoraiban is eltér. Ezzel megnehezítik azonos követelmények azonos megfogalmazású ellenőrzését. Több modell elkészítése és egyszerre történő végleges ellenőrzése azonban azzal az előnnyel járt, hogy a kifejezéseket az egyes modellek ellenőrzéséhez általában csak kis mértékben kellett átírni. Emellett olyan elgépelésből adódó hibák is könnyen felfedhetők voltak, melyeket egyszerre csak egy modellen alkalmazva nehezebben lehetett volna eldönteni, hogy a formális követelmény vagy a modell a hibás.

Az 5.1. és 5.2. fejezetekben a biztosítóberendezési konstrukciós elvek és felhasznált eszközök összehasonlítása történik a modellezési és modellellenőrzési folyamat során szerzett tapasztalatok alapján.

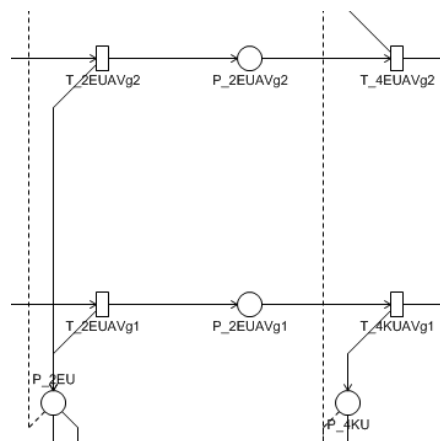
## 5.1. A szerkesztési elvek összehasonlítása

A modellezés során több, a 2.1.3. és 2.1.4. fejezetekben leírt tulajdonság adódott ki. Például míg a nyomvonalas elv esetében a vágányút lezáródása után a kijelölés eltörlésével lehetőség lenne új vágányút kiválasztására (tárolás funkció), addig a vágányutas elvnél csak a vágányút feloldódása után lehet új kezelést kezdeményezni. Annak érdekében, hogy a modellek funkcionalitása megegyezzen, és a modellellenőrzés önmagában egy-egy vágányúthoz minél pontosabban elvégezhető legyen, a nyomvonalas modellt átalakítottam. A jelenlegi modellben új vágányutat beállítani csak a megelőző teljes feloldódása után lehet.

A vágányutas elv egyértelmű előnye, hogy csak azok az elemekhez tartozik bármiféle működés egy vágányút beállítása során, melyek valóban érintettek benne. Emellett elegendő az elemeket csak „egyszer kijelölni”, azaz egy utasítás az állításra vagy lezárásra elegendő. Ezzel ellentétben a nyomvonalas elvű berendezésekben a kijelölés jóval több elemhez eljut, mint amennyi valóban érintett a vágányútban. Például jelen topológián a második vágányra beállított menetnél öt objektum érintett (start, 2 db váltó, vágány cél), de további négy (2-2 db vágány és cél) kap kijelölést. Ahhoz, hogy egy elem egy vágányúthoz egyértelműen kiválasztódjon, mindkét irányból kell kijelölést kapnia. Látható, hogy a nyomvonalas berendezésben a vágányút-beállítás körülbelül kétszer annyi működéssel jár. Emellett külön funkció szükséges a kijelölt, de a vágányútban nem érintett elemek kijelölésének megszüntetéséhez. Ennek helyes kidolgozása jelentős munkaráfordítást igényelt a modellezés során.

A nyomvonalas elv bár több működést tartalmaz, az egyes elemek kapcsolata mégis átláthatóbb. Ennek megjelenése Petri hálós modell esetén szemléletesebb. Geografikus elv esetén kapcsolatok csak „egységen belül”, illetve egységek között vannak. Ezzel szemben a vágányutas elv modelljében például a vágányút oldását végző tranzícióból számos él fut egyszerre végig a modell több helyére, ami az átláthatóságot csökkenti.

Nyomvonalas elv esetén a fentiekből adódóan egy objektumnak mindig ugyanannyi kapcsolata lehet, ezzel szemben vágányutas elv esetén egy váltó annyi vágányúttal van kapcsolatban, ahányban érintett. Ez nagyobb állomások esetén és az átmenő vágányokhoz közelebb fekvő váltók esetén igen nagy számú összeköttetést jelenthet. Ebből adódóan az egyes elemek kapcsolatainak számában is jelentős különbségek lehetnek. Például, a 83. ábra szerint a 2-es váltó egyenes állásában két vágányútban (I. illetve a II. vágányra vezető) érintett, míg a 4-es váltó kitérőben csak egy vágányútban (I. vágányra vezető) szerepel. Az állásoknak megfelelő „P\_xU” utasítás helyek így eltérő számú tranzíciót kaphatnak token (jelen esetben kettő és egy).



83. ábra A 2-es váltó és a 4-es váltó állítás utasítás kapcsolatai



Elmondható, hogy a modellek előállításának munkáigénye mindkét formalizmus alkalmazása során nyomvonalas esetben a vágányutashoz képest 2-3-szoros volt. Emellett figyelembe kell venni, hogy a geografikus elv leképezésekor több hibalehetőség adódik, ami a modellek elkészítésekor számos utólagos módosítást igényelt. Azonban míg a jól megalkotott objektum modellek más helyszínrajz esetén módosítás nélkül használhatók, addig minden topológiára külön vágányutas függőségek megvalósítása szükséges.

A modellellenőrzés során is jelentős különbség adódott az egyes kifejezések kiértékelésének idejében. A Petri hálós modellek ellenőrzésekor az  $AG(EF(p))$  jellegű kifejezések kiértékelését a modellellenőrző tájékoztatása szerint nyomvonalas esetben 15-20 másodperc alatt, míg vágányutas esetben 0,00-0,02 másodperc alatt végezte el. A különbség több nagyságrend. Ennek jelentősége összetett vágányhálózat esetén jelenik meg, automatikus ellenőrzés lefuttatásakor is igen nagy eltérés adódhat. Emellett az állapottér méretéről a PetriDotNet nem közöl információt, így azok megadására itt nincs lehetőség.

Általánosan elmondható, hogy a nyomvonalas elvű berendezéseknél minden információ lokálisan, az elemeknél kerül tárolásra. Innen kell a megfelelő információt úgy eljuttatni a vágányút valamelyik „végére”, hogy minden objektum csak a szomszédjaival kommunikálhat. Ezzel szemben a vágányutas elv egy központi logikára épül, mely az egyes elemek állapotát külön-külön „felülről” lekérdezheti, ellenőrizheti.

Ahol a nyomvonalas elv előnye megjelenik, azok a nagyállomási berendezések. Az ilyen állomások vágányutas elvű berendezéséhez nagy és összetett táblázatok (adatbázisok) készítése lenne szükséges a függőségek megvalósításához. Ezek elkészítése ugyanakkor hibás is lehet. (Erre nyújtana megoldás a függőségi táblázatok automatikus generálásása a helyszínrajz alapján.) Jól működő elemekből (egységekből) ezzel szemben bármilyen topológia leképezhető, csak az objektumok összerendezését kell elvégezni. Ebből kifolyólag jól alkalmazható átépítések, de későbbi állomásbővítések során is, mikor kevés munkával (az új topológiának megfelelően) átalakítható a berendezés.

## **5.2.A formalizmusok, eszközök összehasonlítása, értékelése**

Mindkét eszköz használata előnyökkel és hátrányokkal is jár. A Petri hálókkal történő modellezés könnyen érthető és tanulható, mivel kevés és egyszerű működésű elem használata szükséges (helyek, tranzíciók, élek, tokenek). Az alapvető ismeretek elsajátításával kisebb modellek könnyen létrehozhatók. A matematikai háttér megismerése és a különböző tulajdonságok értelmezése viszont nagyobb ráfordítást igényel. A Petri hálók kiterjesztései bonyolítják a modellezést és melyek helyes használatuk elsajátítása további befektetést igényel.

A Petri hálókkal szemléletesen lehet leírni különböző folyamatokat (pl. szinkronizáció). Kisebb modellek esetén egy állapot szemléletes a tokeneloszlás alapján. Nagyobb modellek, sok token esetén az átláthatóság csökken, több él használatával a modellek nehezen értelmezhetővé válnak. Például az egy helyre befutó élek közül ránézésre nem, vagy nehezen lehet megállapítani, melyek vezetnek ki a helyről, és melyek vezetnek be. Az értelmezést és bizonyos egyszerűbb hibák felfedését megkönnyíti a szimuláció.

A Petri hálók modellezésére igénybe vett PetriDotNet eszköz könnyen tanulható, jól kezelhető felhasználói felülettel rendelkezik. Az egyes elemek használata és beállításai logikusan kezelhetők. Az eszköz szerkesztő módjában lehetőség van akár több, eltérő típusú elem közös tulajdonságainak egyidejű kezelésére (pl. vonalszín megadására). Lehetőség van ugyanakkor csak egy adott, azonos elem-

típus elemeinek többes kijelölésére. Hasznos funkciónak bizonyult a modellezés során a helyek és tranzíciók névpozícióinak billentyűparanccsal történő egyszerű változtathatósága. Pozitívum, hogy a nevek nem függetleníthetők az adott elemtől (ellentétben a Snoopy eszközzel).

Az eszköz megfelelően kezeli az alhálókat, ezek használatakor azonban figyelemmel kell lenni arra, hogy a későbbi módosítás nehézséget okozhat. Egy hely áthelyezése a szintek között ugyanis az adott szintbeli kapcsolatainak elvesztésével jár, amiket ezután újra kell rajzolni.

A Petri hálós eszközök használatának további előnye, hogy az elkészült modellek különböző formátumokban menthetők el. Ezáltal megkönnyítve az átjárást más hasonló modellező programok között, például a PetriDotNet és a Snoopy között „\*.pn” kiterjesztés alkalmazásával. Az elkészült modellek képként, PNG formátumban kimenthetők, mely a további felhasználás szempontjából előnyös lehet.

A PetriDotNet alkalmas a biztosítóberendezések modellezésére, mert az egyes elemek elrendezésével a vágányhálózat topológiája, vagy a táblázatoknak megfelelő soros, oszlopos elrendezés jól lekövethető. Jól alkalmazható a vasút területén jellemző két állapottal rendelkező részrendszerek (pl.: váltóállítási utasítás vagy kijelölés +/-, jelző „Megállj!"/szabad, vágány foglalt/szabad, jelfogó húz/ejt) modellezésére (adott helyen van token, vagy nincs). További előnye, hogy a tranzíciók engedélyezett-ségével egyszerre több feltétel teljesülését is lehet ellenőrizni, tehát szinkronizáció egy időben egyszerre több elem között is létrejöhet.

Az UPPAAL eszköz által megvalósított automaták használatának elsajátítása a Petri hálók alkalmazásához képest több ráfordítást igényel. Az eszköz használatához a programozói ismeretek előnyt jelentenek. Az összetett szintaktika elsajátítása időbe telhet. Az eszköz súgója azonban megfelelő támogatást nyújt a kifejezések helyes használatához.

A modellezést jelentősen megkönnyíti az automaták példányosításának lehetősége. Ezáltal az azonos felépítésű és viselkedésű objektumokból egy sablon megalkotása után tetszőleges számú generálható le. A globális és lokális deklarációk kezelése egyértelműen, az automata sablonokhoz tartozóan történik. Az átmenetekhez tartozó kifejezések szerkesztése az élkre kattintva egyszerűen elvégezhető, színezésük az azonosíthatóságukat egyértelművé teszi. Ezekkel a kifejezésekkel az állapotátmenetekhez egyszerűen szabhatók feltételek és társíthatók funkciók.

A szimuláció lehetősége az UPPAAL használata esetén elősegíti egyszerűbb hibák felfedezését. Akár a modellek készítése során is jól használható a modellellenőrző, mely ellenpéldák generálásával a hibák helyének feltárását is segíti. További előnye, hogy ezáltal a holtpontokba vezető utak is megtalálhatók.

Az automaták használatának előnyös tulajdonsága, hogy egy automata mindig csak egy állapotban lehet, így azok önmagukban áttekinthetőek. Automatákból felépített hálózatok esetén viszont minden egyes automatának az állapotát ismerni kell. Tovább nehezíti az átláthatóságot, hogy az egyes állapotokban tartózkodva az automatákhoz tartozó lokális és a globális változók értéke nem jelenik meg közvetlenül. Az UPPAAL szimulációs részének használata során azonban a változók értékei nyomon követhetőek.

Az UPPAAL eszköz alkalmazható biztosítóberendezések modellezéséhez, mert az automatákkal jól leírhatók a több modulból felépülő nyomvonalas elvű berendezések, valamint a központi logikát megvalósító és a vezérelt elemek moduljaiból összeálló vágányutas elvű berendezések is. A Petri hálókhoz hasonlóan logikai értékű változóikkal alkalmasak a vasút területén jellemző két állapottal rendelkező részrendszerek leírására. Elsősorban vágányutas elvű berendezés modellezésénél jelent

azonban hátrányt, hogy (szinkronizációval) egyszerre csak egy objektumhoz tartozó változók értéke ellenőrizhető. Nem valósítható meg tehát egy lépésben, csak elemenként különböző feltételek teljesülésének vizsgálata. Az egyszerre több elemmel is kapcsolatot létesíteni tudó broadcast csatorna sem alkalmas ellenőrzésre, mert a kezdeményező fél állapotátmenetét akkor is végrehajtja, ha egyáltalán nincs, vagy az elvártnál kevesebb fogadó fél van. Ezzel ellentétben például oldásra használható (mint a modellekben, ld. 3.3.4. fejezet), mert ebben az esetben csak azok az elemek fognak feloldódni, melyek valóban készen állnak rá.

A két eszköz jelen dolgozat szempontjából való összehasonlítása a 31. táblázatban látható.

31. táblázat A PetriDotNet és UPPAAL eszközök általános összehasonlítása

Összehasonlítás alapja	PetriDotNet	UPPAAL
Grafikus szerkesztőfelület	✓	✓
Elemek egyszerű felhelyezése	✓	✓
Szimuláció manuális léptetése	✓	✓
Szimuláció automatikus léptetése	✓	✓
Súgó	X	✓
Beépített modellellenőrző	✓	✓
Átjárhatóság más eszközökkel	✓	X
Modellek kimentése képként, vektorábraként	✓	X

A két eszköz modellellenőrzőjének összehasonlítása a 32. táblázatban látható. A modellellenőrzést egyszerűsíti, hogy mindkét eszközben a CTL nyelv használható. Az azonos feltételek vizsgálatát azonban megnehezíti, hogy a két modellellenőrzőben alkalmazható kifejezések csak részben fedik egymást. Ennek oka a nagy állapotterek eltérő kezelése.

Az állapotterek hatékony belső reprezentációjára, illetve az állapotter felderítés, és az abban való keresés megvalósítására a két eszköz különböző megoldásokat, algoritmusokat használ. Ennek következtében azonban egyes temporális kifejezések kiértékelése az alkalmazott módszerek függvényében igen lassúvá, vagy akár lehetetlenné is válhat (miközben a többi kifejezés kiértékelési sebessége és a kezelhető állapotter mérete nagymértékben nő). Ez okozza a két eszköz CTL szintakszisában a különbségeket.

32. táblázat A PetriDotNet és UPPAAL eszközök modellellenőrzőjének összehasonlítása

Összehasonlítás alapja	PetriDotNet	UPPAAL	
Alkalmazható operátorok	AG, AF	✓	
	EG, EF	✓	
	AU, EU	✓	X
	AX, EX	✓	X
	--> („leads to”)	X	✓
Modell általános tulajdonságok vizsgálata	✓	X	
CTL kifejezés holtponmentességre	X	✓	
Kifejezésszerkesztő	✓	X	
Kifejezések elmentése a modellel	X	✓	
Ellenőrzés eredményének visszajelzése	✓	✓	
Ellenpélda generálása	X	✓	

A PetriDotNet Petri háló alaptulajdonságok vizsgálatára alkalmas programrésze elsősorban oktatási célokra készült. Ezért nagyobb állapotterek vizsgálatakor lelassulhat, illetve tartós „In progress” eredményt adhat.

A PetriDotNet modellellenőrzőjének hasznos funkciója, hogy szerkesztőjében kisebb logikai kifejezések (összehasonlítások) egyidejű bevitelére van lehetőség. Hátránya azonban, hogy szerkesztő módban a program többi ablaka nem lehet aktív, így az egyes vizsgálandó helyek elhelyezkedését nem lehet megkeresni a modellen. A beépített modellellenőrző további negatívuma, hogy a kifejezéseket nem tárolja, és ellenpéldát nem generál.

Az UPPAAL modellellenőrzőjének használatát nem segíti kifejezésszerkesztő, így az egyes változók begépelése kézzel történik. A program a követelmények nem teljesülése esetén (választhatóan: leg-rövidebb, leggyorsabb, stb.) ellenpéldát generál, és az ellenpéldát a szimulációs felületen megjeleníti. A modellellenőrzőbe bevitt kifejezések a modellel együtt elmenthetők.

Megállapítható tehát, hogy mind a két formális modellezési módszer (Petri hálós illetve automatákon alapuló megközelítés), valamint az alkalmazott PetriDotNet és UPPAAL eszközök alkalmasak vasúti biztosítóberendezések modellezésére és szimulációjára. Segítségükkel azonos problémák írhatók le és vizsgálhatók eltérő megközelítésekben. A modellellenőrzésben alkalmazott CTL nyelv segítségével a biztosítóberendezések különböző állapotaira vonatkozó feltételek hatásosan vizsgálhatók.

### 5.3. Fejlesztési lehetőségek

A továbbiakban lehetőség van a modellek bővítésére, a biztosítóberendezési funkciók körének szélesítésére. Ezen további fejlesztési lehetőségek elsősorban az elhanyagolásra került részek beépítésére vonatkoznak (ld. 3.2. fejezet). E részben felvázolom az egyes kiegészítések megvalósítására irányuló első lépéseket. Fontos megjegyezni, hogy jelenleg a modellek működésüket ellátják, illetve annak figyelembevételével készültek, hogy a még hiányzó részek később az aktuális modellekre lehetőleg azok változtatása nélkül ráilleszthetők legyenek. A bővítés során azonban számolni kell azzal, hogy a már elkészült modellrészek kismértékű módosítása lehet szükséges. Mint a fejlesztési tapasztalatokból is kiderült, a funkcióbővítés, további függőségek beépítése a nyomvonalas berendezésmo- dellnél jóval több (de ugyanakkor általános érvényű) munkával jár, ami később megtérül.

A legegyszerűbb módosítások egyike a vágányúttárolás megvalósítása lenne. Új vágányút beállítására szóló parancs kiadására legkorábban az előző beállítása és lezáródása után lenne lehetőség. A vágányúttárolási funkció egyszintű lenne, azaz minden olyan vágányútra/vágányútban lezárt elemre, csak egy további vágányúti parancs lenne betárolható. Ez a lehetőség a nyomvonalas elvű berendezésmo- dellekben az elosztott logika miatt egyszerűbben, kisebb módosításokkal megoldható. Ezeknél a modelleknél a le nem zárt elemek beállítása és lezáródása is megtörténhetne a kijelölést követően. Például a harmadik vágányra beállított vágányút mellett a 4-es váltó állítható és lezárható. A teljes vágányút beállítása az előző feloldódása után következne be. A vágányutas elvű modell esetén külön vágányúttároló modellrész elkészítése válna szükségessé. Itt a beállításról vagy a betárolásról a mo- dell a menetrend alapján döntene.

Következő lépésben történhetne a tolatóvágányutak vonatvágányutakká fejlesztése. Ennek egyik módosítása a nyomvonalas elvű berendezésnél a vágányút lezárásának feltételévé tenni az egyes szakaszok foglaltsági információit. Ebben az esetben már nem elegendő önmagában a jelző szabadra állítása, különböző jelzési fogalmak megjelenítése lenne szükséges. Táblázatos elvű modellenél ez vágányutanként egyszerűen programozható, míg a nyomvonalas modellen a váltók saját sebességi

információjukat a start felé továbbadva kellene meghatározni a legmegszorítóbb értéket. A döntés történhetne minden váltónál, ami a kapott értéket saját megengedett sebességével hasonlítani össze, és a kisebbet adná tovább.

Oldalvédelem megvalósítása. Ennek érdekében felkerülhetnének az egyes vágányok kijárat jelzői (K1-K3), illetve gyök felől álló váltók, egyéb oldalvédelmet adó elemek. Az oldalvédelmi elemek lezáródása a vágányút lezáródása során történhetne meg. Nyomvonalas elv esetén az egyes elemek (váltók) addig nem záródnának le, amíg a hozzájuk tartozó oldalvédelmi elemek nem adnak információ saját lezáródásukról. Első lépésben a szomszédos elemek által adott oldalvédelem megvalósítása történne, a továbbadott oldalvédelem, illetve kettős terelésű váltó megoldása egy következő fejlesztési lépésben történhetne. A lezáródás a célból a start felé tartva minden váltónál a nem kijelölt száron keresne oldalvédelmet, majd annak lezáródása, és a vágányútban érintett elem lezáródása után adódna tovább a start felé. Ehhez a nyomvonalas elvű UPPAAL modellben a jelenleg jelzőt tartalmazó „Start” objektum funkcióinak bővítése is szükséges lenne (az újonnan megjelent kijárat jelzők miatt). Vágányutas elvű modellekben az oldalvédelmi elemek állítás és lezárása a jelenlegi elzárási tervi függőségekkel együtt történne.

A kétirányú menetek lehetővé tétele. A jelenlegihez képest ellentétes irányú menetek megvalósítása időben az oldalvédelem elkészítése után történhetne meg. Az oldalvédelmet adó háttól álló jelzők ugyanis az ellentétes irányú menetek számára relevánsak lennének. Az ellenirányú menetek felvételéhez a modellek közel kettőzése lenne szükséges, hogy a start → cél és cél → start folyamatok a másik irányból is működhessenek.

Összetettebb modellezett vágányhálózat esetén a hibás kijelölést (a kiválasztott kezdő- és célpontok között nem állítható be vágányút) érdemes megakadályozni. Ezt célszerűen még a kijelölés megkezdése előtt le kell tiltani, mert a megkezdett kijelölés törlése nehezebb feladat. A vágányutas elv ezt jelenleg is megvalósítja, mivel az elsőnek kijelölt start után csak hozzá tartozó cél, azaz csak létező vágányút választható. Nyomvonalas elv esetén a kijelölések megkezdése előtt egy külön modellrészlet kell kialakítani az összetartozó start- és célgombok vizsgálatára.

A topológia bővítése, első lépésben az állomás másik végének elkészítése (a váltókörzet tükrözése vízszintesen). Ekkor már összetett vágányutak, bonyolultabb menetek beállítására is lehetőség lenne. A vágányúttárolási lehetőség ebben a lépésben kapna jelentősebb szerepet. Ebben a lépésben már szükséges lehet a nyomvonalas elvű UPPAAL modellben a „Start” és „Cél” objektumok átalakítása, akár összevonása is. Ennek oka, hogy a kijárat jelzők start- és céljelzők szerepét is betölthetik.

A topológia további bővítése (több) nyíltvonali csatlakozó vágánnyal. Ezen fejlesztés bevezetésével beállíthatóvá válnának párhuzamos menetek, illetve kezelni kellene tudni a kerülővágányutakat, az oldalvédelem továbbadását is.

Megcsúszási távolságok, vágányutak kezelése jelenthetne egy következő lépést. A funkció megvalósításához már időzítések figyelembe vétele is szükséges lenne (időzített oldódás). A megcsúszások beállítása a vágányutas modellben jelentősen egyszerűbb lenne az érintett elemek táblázatos felsorolásával. Nyomvonalas elvű modellben egy lehetőséget jelentene az egyes elemek hosszának objektumonkénti ismerete. Ebben az esetben a megcsúszási vágányút a szükséges hosszoknak (50 m vagy 300 m) megfelelően záródna le.

Későbbi fázisokban bekerülhetnének a modellekbe az egyéni és különleges kezelések (váltóállítás, jelző „Megállj!”, stb.). Külön nehézséget jelentene a szimuláció bővítése meghibásodásokkal, nem

üzemszerű működésekkel (pl.: hamis foglaltság) és a vonat hosszának figyelembe vételével. Külön megvalósítást igényelne a vágányúti feltételek folyamatos figyelése (váltó végállás, foglaltság), és ezek nem teljesülése esetén a startjelző „Megállj!” állásba állítása.

Az egyes lépésektől függetlenül a vágányutas elvű modell elemenkénti oldása is megvalósítható. Jelen dolgozatban a két fő biztosítóberendezés szerkesztési elv minél pontosabb elkülönítése érdekében ez csak egy oldószakasszal került megvalósításra. A jelenleg üzemelő korszerű biztosítóberendezésekben az oldószakaszos megoldás a forgalom nagymértékű akadályozása miatt nem elképzelhető. Ehhez elkészítendő egy táblázat, amelyben vágányutanként az egyes elemek elvárt foglaltsági szekvenciái kerülnének megadásra.

Külön bővítés esete lenne egy sorompó objektum vágányúti elemek közé való felvétele. Az állomási útátjárók kezelése az időzítések és az egyszerre több, párhuzamos vágány(út)ban való érintettségük okán jelentenének kihívást.

A modellezés egy későbbi fázisa lehetne több állomás és a nyílt vonalak (utolérés- és ellenmenet kizáró berendezések) kezelése, majd ezek központi forgalomirányító berendezésbe vonása.

## 6. Összefoglalás

A dolgozatban megvizsgáltam a formális modellezés és modellellenőrzés vasúti biztosítóberendezésekben való alkalmazhatóságát. E célból kiválasztottam és felépítésében, funkcióiban lehatároltam egy állomásrészt, mely a modellezés alapját adta. Erre a topológiára képeztem le formálisan a két legjelentősebb biztosítóberendezés szerkesztési elvet (vágányutas és nyomvonalas elvű). A felhasznált eszköz Petri hálók esetében PetriDotNet, az automatákon alapuló megközelítés esetében az UPPAAL volt. CTL nyelven alapuló beépített modellellenőrzőik segítségével elvégeztem a modellellenőrzést, mely minden követelményre eredményesen teljesült.

Az elvégzett munka alapján megállapítható, hogy mind a két formális modellezési módszer (Petri háló illetve automaták), valamint az alkalmazott PetriDotNet és UPPAAL eszközök alkalmasak vasúti biztosítóberendezések modellezésére és szimulációjára. Segítségükkel azonos problémák írhatók le és vizsgálhatók eltérő megközelítésekből. Petri hálókkal a vágányhálózat topológiája, vagy a táblázatoknak megfelelő soros, oszlopos elrendezés jól lekövethető. Az UPPAAL automatáival jól lehet modellezni az egyes vágányúti objektumokat, illetve a központi logika megvalósulását. Jól alkalmazhatók a vasút területén jellemző két állapottal rendelkező részrendszerek (pl.: jelző „Megállj!"/szabad, vágány foglalt/szabad) modellezésére adott helyen lévő tokennel, vagy logikai változókkal. A modellellenőrzésben alkalmazott CTL nyelv segítségével a biztosítóberendezések különböző állapotaira vonatkozó feltételek hatásosan vizsgálhatók.

A formális modellezés és modellellenőrzés, valamint a formális specifikáció használata egyértelműséget, ezáltal mások általi könnyű értelmezhetőséget, ellenőrizhetőséget, és ezek mellett automatikus feldolgozhatóságot tesz lehetővé.

## Felhasznált irodalom

- [1] Dr. BORBÉLY Tibor *Állomási biztosítóberendezések I.* Budapest, Tankönyvkiadó. 1983. pp. 3-69.
- [2] BME KÖZLEKEDÉS- ÉS JÁRMŰIRÁNYÍTÁSI TANSZÉK *A vasúti forgalom irányításának rendszere és eszközei* [számítógép-fájl].  
Budapest, BME KJIT, URL:  
[http://www.kjit.bme.hu/images/stories/targyak/kozlout/vasuti\\_ismerteto\\_2013.pdf](http://www.kjit.bme.hu/images/stories/targyak/kozlout/vasuti_ismerteto_2013.pdf)  
(letöltve: 2016. 07. 19.) pp. 25-50.
- [3] BME KÖZLEKEDÉS- ÉS JÁRMŰIRÁNYÍTÁSI TANSZÉK *Vágányutak* [számítógép-fájl].  
Budapest, BME KJIT, URL:  
<http://www.kjit.bme.hu/images/stories/targyak/vasirkom/fuggosegek2.zip>  
(letöltve: 2016. 07. 19.) pp. 1-8.
- [4] BME KÖZLEKEDÉS- ÉS JÁRMŰIRÁNYÍTÁSI TANSZÉK *Függőségek* [számítógép-fájl].  
Budapest, BME KJIT, URL:  
[http://www.kjit.bme.hu/images/stories/targyak/vasirkom/vasti\\_irnynt\\_s\\_kommunikcis\\_rendszerek.zip](http://www.kjit.bme.hu/images/stories/targyak/vasirkom/vasti_irnynt_s_kommunikcis_rendszerek.zip)  
(letöltve: 2016. 07. 19.) pp. 1-32.
- [5] BME KÖZLEKEDÉS- ÉS JÁRMŰIRÁNYÍTÁSI TANSZÉK *A vágányúti logika leképezése* [számítógép-fájl].  
Budapest, BME KJIT, URL:  
[http://www.kjit.bme.hu/images/stories/targyak/vasirkom/a\\_vgnyti\\_logika\\_lekpezese.pdf](http://www.kjit.bme.hu/images/stories/targyak/vasirkom/a_vgnyti_logika_lekpezese.pdf)  
(letöltve: 2016. 07. 19.) pp. 1-10.
- [6] BME KÖZLEKEDÉS- ÉS JÁRMŰIRÁNYÍTÁSI TANSZÉK *Függőségi tervek* [számítógép-fájl].  
Budapest, BME KJIT, URL:  
[http://www.kjit.bme.hu/images/stories/targyak/vasirkom/vasti\\_irnynt\\_s\\_kommunikcis\\_rendszerek.zip](http://www.kjit.bme.hu/images/stories/targyak/vasirkom/vasti_irnynt_s_kommunikcis_rendszerek.zip)  
(letöltve: 2016. 07. 19.) pp. 1-20.
- [7] WWW.STELLWERKE.DE *Mechanische Stellwerksbauformen* [számítógép-fájl].  
URL: [http://www.stellwerke.de/formen/seite2\\_2.html](http://www.stellwerke.de/formen/seite2_2.html)  
(letöltve: 2016. 12. 05.)
- [8] BOROS István *Dominó 55 rendszerű állomási biztosítóberendezések.* Budapest, MÁV Rt. Személyügyi Főosztálya, 1997. pp. 21-63.
- [9] GRÓF József *Vasúti vonóvezetékes biztosítóberendezések. Mechanikai rész*  
Budapest, Közlekedési Dokumentációs vállalat, 1958. pp. 154-219.
- [10] HEGEDÜS Géza. Biztosított állomások berendezései. In: Dr. CZÉRE Béla szerk. *A vasúti technika kézikönyve 2. kötet.* Budapest, Műszaki Könyvkiadó, 1977. pp. 357-365.  
ISBN 963 10 1237 9
- [11] SZENTKERESZTY Pál *Erősáramú biztosítóberendezések.* Budapest, Felsőoktatási Jegyzetellátó Vállalat, 1954. pp. 224-243.



- [12] Dr. KOSZNAI János. Dominó 55 típusú jelfogófüggéses állomási biztosítóberendezés. In: Dr. CZÉRE Béla szerk. *A vasúti technika kézikönyve 2. kötet*. Budapest, Műszaki Könyvkiadó, 1977. pp. 365-389. ISBN 963 10 1237 9
- [13] DIVINYI Sándor. Dominó 70 (D 70) állomási biztosítóberendezés. In: Dr. CZÉRE Béla szerk. *A vasúti technika kézikönyve 2. kötet*. Budapest, Műszaki Könyvkiadó, 1977. pp. 389-410. ISBN 963 10 1237 9
- [14] CZIFRA Zoltán és DIVINYI Sándor *Dominó 70 állomási biztosítóberendezés*. Budapest, Közlekedési Dokumentációs Vállalat, 1981. pp. 70-71. ISBN 963 552 077 8
- [15] Dr. TARNAI Géza *Elektronikus biztosítóberendezések* [számítógép-fájl]. Budapest, BME KJIT, URL: <http://www.kjit.bme.hu/images/stories/targyak/vasirkom/elbb.pdf> (letöltve: 2016. 11. 05.) pp. 1-15.
- [16] BERÉNYI László *Elektronikus biztosítóberendezések vizsgálati szempontjai és tapasztalatai, Vezetékek Világa*, 14. évf. 3. sz. (2009) pp. 10-13. ISSN 1416-1656
- [17] Dr. MAJZIK István [et al.] *A formális módszerek szerepe* [számítógép-fájl]. Budapest, BME MIT, URL: [http://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/msc-t%C3%A1rgyak/form%C3%A1lis-m%C3%B3dszerek/16/FM-2016\\_EA01b\\_formalis\\_modszerek\\_szerepe.pdf](http://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/msc-t%C3%A1rgyak/form%C3%A1lis-m%C3%B3dszerek/16/FM-2016_EA01b_formalis_modszerek_szerepe.pdf) (letöltve: 2016. 10. 15.) pp. 1-15.
- [18] ÉSIK Zoltán [et al.] *Hardver- és szoftverrendszerek verifikációja*. Szeged, Typotex, 2011., pp. 7-12. ISBN 978-963-279-497-6
- [19] ÉSIK Zoltán [et al.] *Hardver- és szoftverrendszerek verifikációja*. Szeged, Typotex, 2011., pp. 12-19. ISBN 978-963-279-497-6
- [20] ÉSIK Zoltán [et al.] *Hardver- és szoftverrendszerek verifikációja*. Szeged, Typotex, 2011., pp. 20-55. ISBN 978-963-279-497-6
- [21] Dr. BARTHA Tamás [et al.] *Petri hálók: Alapelemek és kiterjesztések* [számítógép-fájl]. Budapest, BME MIT, URL: [https://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/msc-t%C3%A1rgyak/form%C3%A1lis-m%C3%B3dszerek/16/FM-2016\\_EA07a\\_PN\\_alapelemek\\_kiterjesztesek.pdf](https://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/msc-t%C3%A1rgyak/form%C3%A1lis-m%C3%B3dszerek/16/FM-2016_EA07a_PN_alapelemek_kiterjesztesek.pdf) (letöltve: 2016. 10. 15.) pp. 1-75.
- [22] Dr. BARTHA Tamás [et al.] *Petri hálók dinamikus tulajdonságai* [számítógép-fájl]. Budapest, BME MIT, URL: [https://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/msc-t%C3%A1rgyak/form%C3%A1lis-m%C3%B3dszerek/16/FM-2016\\_EA07a\\_PN\\_alapelemek\\_kiterjesztesek.pdf](https://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/msc-t%C3%A1rgyak/form%C3%A1lis-m%C3%B3dszerek/16/FM-2016_EA07a_PN_alapelemek_kiterjesztesek.pdf) (letöltve: 2016. 10. 15.) pp. 1-53.

- [23] Dr. BARTHA Tamás [et al.] *Modellezés Petri hálókkal* [számítógép-fájl]. Budapest, BME MIT, URL: [https://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/msc-t%C3%A1rgyak/form%C3%A1lis-m%C3%B3dszerek/16/FM-2016\\_EA07a\\_PN\\_alapelemek\\_kiterjesztesek.pdf](https://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/msc-t%C3%A1rgyak/form%C3%A1lis-m%C3%B3dszerek/16/FM-2016_EA07a_PN_alapelemek_kiterjesztesek.pdf) (letöltve: 2016. 10. 15.) pp. 1-53.
- [24] MURATA, Tadao *Petri Nets: Properties, Analysis and Applications, Proceedings of the IEEE*. Vol. 77, No. 4. (1989) pp. 542-559.
- [25] ÉSIK Zoltán [et al.] *Hardver- és szoftverrendszerek verifikációja*. Szeged, Typotex, 2011., pp. 56-73. ISBN 978-963-279-497-6
- [26] Dr. MAJZIK István *Követelmények formalizálása: Temporális logikák* [számítógép-fájl]. Budapest, BME MIT, URL: [http://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/msc-t%C3%A1rgyak/form%C3%A1lis-m%C3%B3dszerek/16/FM-2016\\_EA02a\\_linearis\\_temporalis\\_logikak.pdf](http://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/msc-t%C3%A1rgyak/form%C3%A1lis-m%C3%B3dszerek/16/FM-2016_EA02a_linearis_temporalis_logikak.pdf) (letöltve: 2016. 10. 15.) pp. 1-54.
- [27] Dr. MAJZIK István *Elágazó idejű temporális logikák: Computational Tree Logic (CTL, CTL\*)* [számítógép-fájl]. Budapest, BME MIT, URL: [http://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/doktorandusz-t%C3%A1rgyak/szoftver-verifik%C3%A1ci%C3%B3-%C3%A9s-valid%C3%A1ci%C3%B3/14/SZVV-2014\\_EA06\\_ctl.pdf](http://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/doktorandusz-t%C3%A1rgyak/szoftver-verifik%C3%A1ci%C3%B3-%C3%A9s-valid%C3%A1ci%C3%B3/14/SZVV-2014_EA06_ctl.pdf) (letöltve: 2016. 10. 15.) pp. 1-46.
- [28] Dr. MAJZIK István *Követelmények formalizálása: Elágazó idejű temporális logikák* [számítógép-fájl]. Budapest, BME MIT, URL: [http://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/msc-t%C3%A1rgyak/form%C3%A1lis-m%C3%B3dszerek/16/FM-2016\\_EA02b\\_elagazo\\_temporalis\\_logikak.pdf](http://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/msc-t%C3%A1rgyak/form%C3%A1lis-m%C3%B3dszerek/16/FM-2016_EA02b_elagazo_temporalis_logikak.pdf) (letöltve: 2016. 10. 15.) pp. 1-60.
- [29] DARVAS Dániel *Temporális logika* [számítógép-fájl]. Budapest, BME MIT, URL: [https://mialmanach.mit.bme.hu/erdekessegek/temporalis\\_logika](https://mialmanach.mit.bme.hu/erdekessegek/temporalis_logika) (letöltve: 2016. 10. 15.)
- [30] MSZ EN 50129 Railway applications – Communication, signaling and processing systems – Safety related electronic systems for signaling Budapest, MSZT, 2003. pp. 30-41.
- [31] MSZ EN 50129 Railway applications – Communication, signaling and processing systems – Safety related electronic systems for signaling Budapest, MSZT, 2003. pp. 85-93.
- [32] MSZ EN 50128 *Vasúti alkalmazások. Távközlési, biztosítóberendezési és adatfeldolgozó rendszerek. Szoftverek vasúti vezérlő- és védelmi rendszerekhez* Budapest, MSZT, 2011. pp. 6-17.
- [33] MSZ EN 50128 *Vasúti alkalmazások. Távközlési, biztosítóberendezési és adatfeldolgozó rendszerek. Szoftverek vasúti vezérlő- és védelmi rendszerekhez* Budapest, MSZT, 2011. pp. 65-127.

- [34] Dr. BARTHA Tamás [et al.] *Hierarchikus Petri hálók* [számítógép-fájl]. Budapest, BME MIT, URL: [https://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/msc-t%C3%A1rgyak/form%C3%A1lis-m%C3%B3dszerek/16/FM-2016\\_EA09c\\_PN\\_hierarchikus.pdf](https://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/msc-t%C3%A1rgyak/form%C3%A1lis-m%C3%B3dszerek/16/FM-2016_EA09c_PN_hierarchikus.pdf) (letöltve: 2016. 10. 15.) pp. 1-13.
- [35] VÖRÖS András [et al.] *PetriDotNet 1.5 User Manual* [számítógép-fájl]. Budapest, BME MIT, URL: [http://petridotnet.inf.mit.bme.hu/releases/pdn1\\_manual.pdf](http://petridotnet.inf.mit.bme.hu/releases/pdn1_manual.pdf) (letöltve: 2016. 10. 15.) pp. 7-30.
- [36] BTU DATA STRUCTURES AND SOFTWARE DEPENDABILITY *Snoopy General Description* [számítógép-fájl]. Cottbus, BTU DSSZ, URL: <http://www-dssz.informatik.tu-cottbus.de/DSSZ/Software/Snoopy> (letöltve: 2016. 11. 12.)
- [37] BTU DATA STRUCTURES AND SOFTWARE DEPENDABILITY *Charlie General Description* [számítógép-fájl]. Cottbus, BTU DSSZ, URL: <http://www-dssz.informatik.tu-cottbus.de/DSSZ/Software/Charlie> (letöltve: 2016. 11. 12.)
- [38] UPPAAL *Home* [számítógép-fájl]. URL: <http://www.uppaal.org/> (letöltve: 2016. 11. 12.)
- [39] UPPAAL *UPPAAL v.4™ The verification platform* [számítógép-fájl]. URL: <http://www.uppaal.com/index.php?sida=186&rubrik=93> (letöltve: 2016. 11. 12.)
- [40] BEHRMANN, Gerd [et al.] *A Tutorial on Upaal 4.0* [számítógép-fájl]. Aalborg University, URL: <http://www.uppaal.com/admin/anvandarfiler/filer/uppaal-tutorial.pdf> (letöltve: 2016. 11. 12.) pp. 1-9.
- [41] DARVAS Dániel és HORÁNYI Gergő *Bevezető az UPPAAL használatába* [számítógép-fájl]. Budapest, BME MIT, URL: [https://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/rendszertervez%C3%A9s-szakk%C3%B6r/12/szakkor\\_uppaal\\_bevezeto\\_0.pdf](https://inf.mit.bme.hu/sites/default/files/materials/category/kateg%C3%B3ria/oktat%C3%A1s/rendszertervez%C3%A9s-szakk%C3%B6r/12/szakkor_uppaal_bevezeto_0.pdf) (letöltve: 2016. 11. 12.) pp. 2-9.
- [42] ÉSIK Zoltán [et al.] *Hardver- és szoftverrendszerek verifikációja*. Szeged, Typotex, 2011., pp. 123-130. ISBN 978-963-279-497-6
- [43] CSEH Attila [et al.] *Biztosítóberendezések modellezése Petri hálókkal* *Vezetékek Világa*, 19. évf. 1. sz. (2014) pp. 14-17. ISSN 1416-1656
- [44] FARKAS Balázs és PINGITZER Gábor *Vágányút-beállítás nyomvonal elven* [számítógép-fájl]. Budapest, BME KJIT, URL: <http://www.kjit.bme.hu/images/stories/targyak/vasirkom/petri-tf.zip> (letöltve: 2016. 12. 05.)
- [45] GAZDASÁGI ÉS KÖZLEKEDÉSI MINISZTERIUM *103/2003. (XII. 27.) GKM rendelet a hagyományos vasúti rendszerek kölcsönös átjárhatóságáról* [számítógép-fájl]. URL: [http://net.ioftar.hu/jr/gen/hjegy\\_doc.cgi?docid=A0300103.GKM](http://net.ioftar.hu/jr/gen/hjegy_doc.cgi?docid=A0300103.GKM) (letöltve: 2016. 11. 23.)

- [46] MÁV TEBK *Elektronikus állomási biztosítóberendezések feltétfüzete*. 1.02 verzió (TEBK 9240)  
1996. pp. 60-61.

## Ábrajegyzék

1. ábra A vágányúti logika leképezése (forrás: [5]) .....	7
2. ábra Egy négyvágányos állomás menetterve és elzárási terve (forrás: [8]) .....	8
3. ábra Menettermi kizárások megvalósulása jelzőberendezésben (forrás: [9]) .....	10
4. ábra Lemezidom függés a rendelkezőkészülékben (forrás: [9]) .....	11
5. ábra Integra típusú berendezés pultjának részlete .....	13
6. ábra A vágányút-kijelölés folyamata a D55 berendezésben (forrás: [8]) .....	14
7. ábra Jelfogóegységek nyomkábelezése a D70 berendezésben (forrás: [14]) .....	15
8. ábra A formális módszereken alapú fejlesztés egy lehetséges folyamatábrája (forrás: [17] [18]) ...	22
9. ábra Szinkronizáció megvalósítása Petri hálóval .....	25
10. ábra Kölcsönös kizárás megvalósítása Petri hálóval .....	26
11. ábra Ellenőrzés megvalósítása Petri hálóval .....	26
12. ábra Korlátos kapacitás megvalósítása Petri hálóval .....	26
13. ábra Az EN 50128 szabvány egy részlete (forrás: [33]) .....	29
14. ábra A PetriDotNet modellezői felülete .....	30
15. ábra A PetriDotNet szimulációs felülete .....	30
16. ábra A PetriDotNet hálóanalízis felugró ablaka .....	31
17. ábra A PetriDotNet CTL modellellenőrző kifejezésszerkesztője .....	31
18. ábra PetriDotNet üzenet sikeres modellellenőrzési feltételről .....	32
19. ábra A nyomvonalas elvű biztosítóberendezés egy korai modellje a Snoopy eszközben .....	32
20. ábra Az UPPAAL modellezői felülete .....	33
21. ábra Az UPPAAL élszerkesztője .....	34
22. ábra Az UPPAAL beépített szimulátora .....	35
23. ábra Az UPPAAL modellellenőrző felülete .....	35
24. ábra A modellezés alapjául szolgáló helyszínrajz .....	36
25. ábra A vágányúti elemek geografikus kapcsolatrendszere .....	39
26. ábra A vágányúti funkciók időbeli lefutása .....	40
27. ábra A nyomvonal elvű Petri hálós modell főhálójának funkciók szerint színezett képe .....	42
28. ábra A nyomvonal elvű Petri hálós modell főhálójának objektumok szerint színezett képe .....	43
29. ábra A nyomvonal elvű Petri hálós modell foglaltság hálójának objektumok szerint színezett képe .....	44
30. ábra A start- és célpontok kiválasztása a nyomvonalas elv Petri hálós modelljében .....	45
31. ábra A vágányút start felőli kijelölése a nyomvonalas elv Petri hálós modelljében .....	46
32. ábra A vágányút cél felőli kijelölése a nyomvonalas elv Petri hálós modelljében .....	46
33. ábra A vágányút lezárása a nyomvonalas elv Petri hálós modelljében .....	47
34. ábra A 2-es és 4-es váltók alhálói a nyomvonalas elv Petri hálós modelljében .....	47
35. ábra A vágányút kijelölésének törlése start felől a nyomvonalas elv Petri hálós modelljében .....	48
36. ábra A vágányút kijelölésének törlése cél felől a nyomvonalas elv Petri hálós modelljében .....	48
37. ábra Foglaltságok és vágányút feloldás a nyomvonalas elv Petri hálós modelljében .....	49
38. ábra A vágányutas elvű Petri hálós modell főhálójának funkciók szerint színezett képe .....	50
39. ábra Vágányút-kijelölés és menettermi függőségek a vágányutas elv Petri hálós modelljében .....	51
40. ábra Az elzárási terv függőségeinek végrehajtása a vágányutas elv Petri hálós modelljében .....	52
41. ábra A 2-es és 4-es váltók alhálói a vágányutas elv Petri hálós modelljében .....	52
42. ábra Vágányútlezárás, jelzőállítás és vágányút feloldás a vágányutas elv Petri hálós modelljében	53

43. ábra Objektumok kapcsolata grafikus formában a nyomvonalas elv UPPAAL modelljében .....	56
44. ábra Globális deklarációk a nyomvonalas elv UPPAAL modelljében .....	56
45. ábra A start- és célpontok kiválasztása a nyomvonalas elv UPPAAL modelljében .....	58
46. ábra A „Start” automata kijelölése startból a nyomvonalas elv UPPAAL modelljében .....	58
47. ábra A „Valto” automata kijelölése startból a nyomvonalas elv UPPAAL modelljében.....	59
48. ábra A „Vagany” automata kijelölése startból a nyomvonalas elv UPPAAL modelljében .....	59
49. ábra A „Cel” automata kijelölése startból a nyomvonalas elv UPPAAL modelljében .....	59
50. ábra A „Cel” automata kijelölése célból a nyomvonalas elv UPPAAL modelljében .....	60
51. ábra A „Vagany” automata kijelölése célból a nyomvonalas elv UPPAAL modelljében .....	60
52. ábra A „Valto” automata kijelölése célból a nyomvonalas elv UPPAAL modelljében .....	61
53. ábra A „Start” automata kijelölése célból a nyomvonalas elv UPPAAL modelljében .....	61
54. ábra A „Cel” automata lezárása a nyomvonalas elv UPPAAL modelljében .....	61
55. ábra A „Vagany” automata lezárása a nyomvonalas elv UPPAAL modelljében.....	62
56. ábra A „Valto” automata váltóállítást végző függvénye a nyomvonalas elv UPPAAL modelljében	62
57. ábra A váltó állítása és a „Valto” automata lezárása a nyomvonalas elv UPPAAL modelljében.....	63
58. ábra A „Start” automata lezárása és kijelölésének törlése startból a nyomvonalas elv UPPAAL modelljében .....	63
59. ábra A „Valto” automata kijelölésének törlése startból és célból a nyomvonalas elv UPPAAL modelljében .....	64
60. ábra A „Vagany” automata kijelölésének törlése startból és célból a nyomvonalas elv UPPAAL modelljében .....	65
61. ábra A „Cel” automata kijelölésének törlése startból és célból a nyomvonalas elv UPPAAL modelljében.....	65
62. ábra A „Start” automata kijelölésének törlése célból a nyomvonalas elv UPPAAL modelljében ...	65
63. ábra A „Start” automatához tartozó startjelző szabadra állítása a nyomvonalas elv UPPAAL modelljében .....	66
64. ábra A „Start” automata foglaltsága és feloldódása a nyomvonalas elv UPPAAL modelljében .....	66
65. ábra A „Valto” automata foglaltsága és feloldódása a nyomvonalas elv UPPAAL modelljében ....	67
66. ábra A „Vagany” automata foglaltsága és feloldódása a nyomvonalas elv UPPAAL modelljében .	67
67. ábra A „Cel” automata foglaltsága és feloldódása a nyomvonalas elv UPPAAL modelljében .....	68
68. ábra Start felől gyök felől járt váltó kijelölésének törlése a nyomvonalas elv UPPAAL modelljében .....	68
69. ábra A vágányutas elv UPPAAL modelljének csatornatömbjei .....	69
70. ábra Globális deklarációk a vágányutas elv UPPAAL modelljében.....	70
71. ábra A menettervi függőségeket megvalósító globális tömbök a vágányutas elv UPPAAL modelljében .....	70
72. ábra A „Menet_Elzarasi” automata változója a vágányutas elv UPPAAL modelljében .....	70
73. ábra Vágányút-kijelölés és menettervi függőségek a vágányutas elv UPPAAL modelljében.....	71
74. ábra A menettervi függőségeket megvalósító függvények a vágányutas elv UPPAAL modelljében .....	72
75. ábra Az elzárási terv függőségeinek végrehajtása a vágányutas elv UPPAAL modelljében.....	73
76. ábra A „Valto” automata állítása és lezárása a vágányutas elv UPPAAL modelljében .....	73
77. ábra A „Vagany” automata lezárása a vágányutas elv UPPAAL modelljében.....	74
78. ábra Jelzőállítás és vágányút feloldás a vágányutas elv Petri hálós modelljében.....	74
79. ábra A „Valto” automata feloldása a vágányutas elv UPPAAL modelljében.....	75

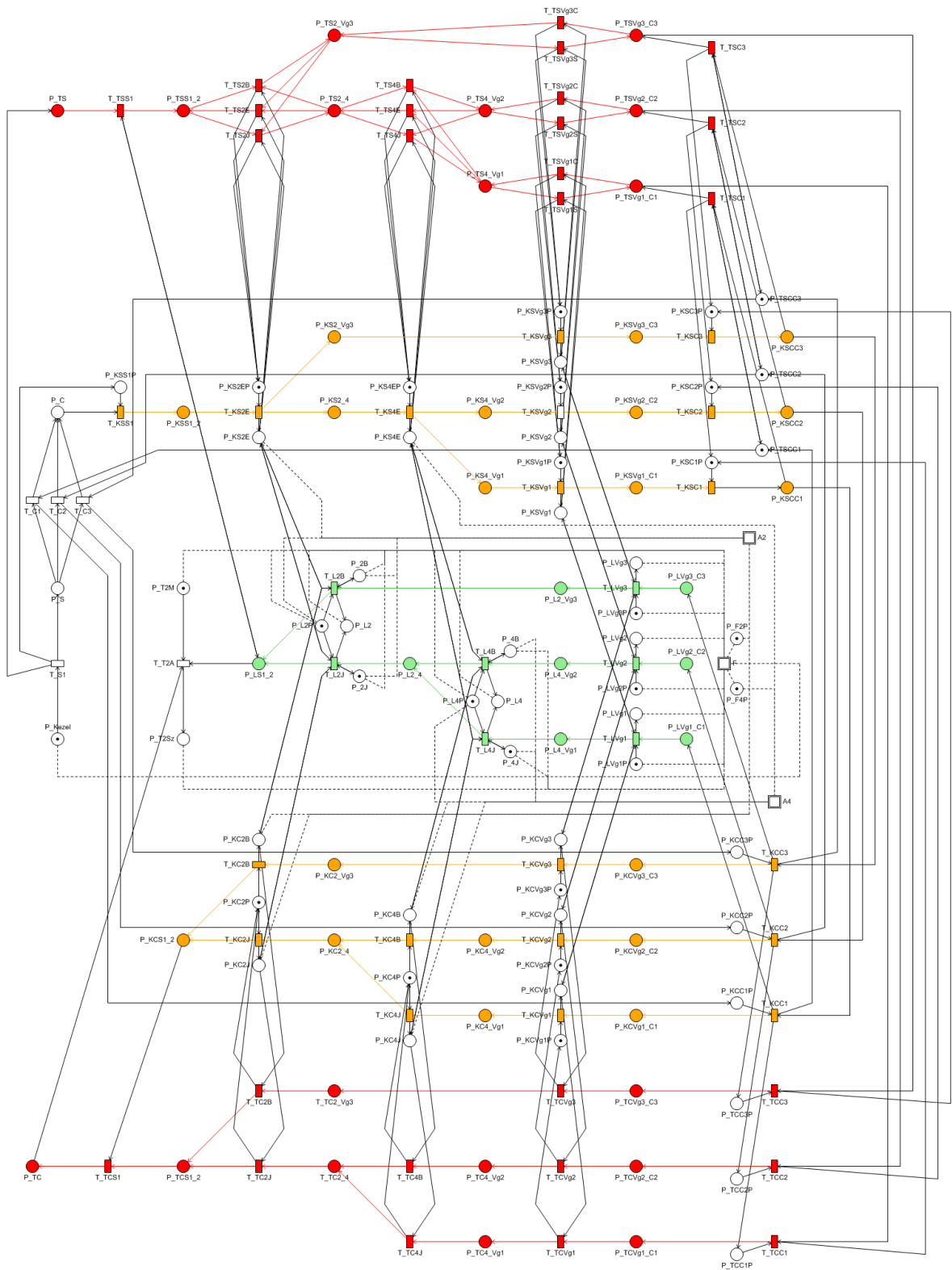
80. ábra A „Vagany” automata feloldása a vágányutas elv UPPAAL modelljében .....	75
81. ábra A kezdőállapotba való átmenet a vágányutas elv UPPAAL modelljében.....	75
82. ábra A menettervi függőségek oldását végző függvény a vágányutas elv UPPAAL modelljében ...	76

## Táblázatjegyzék

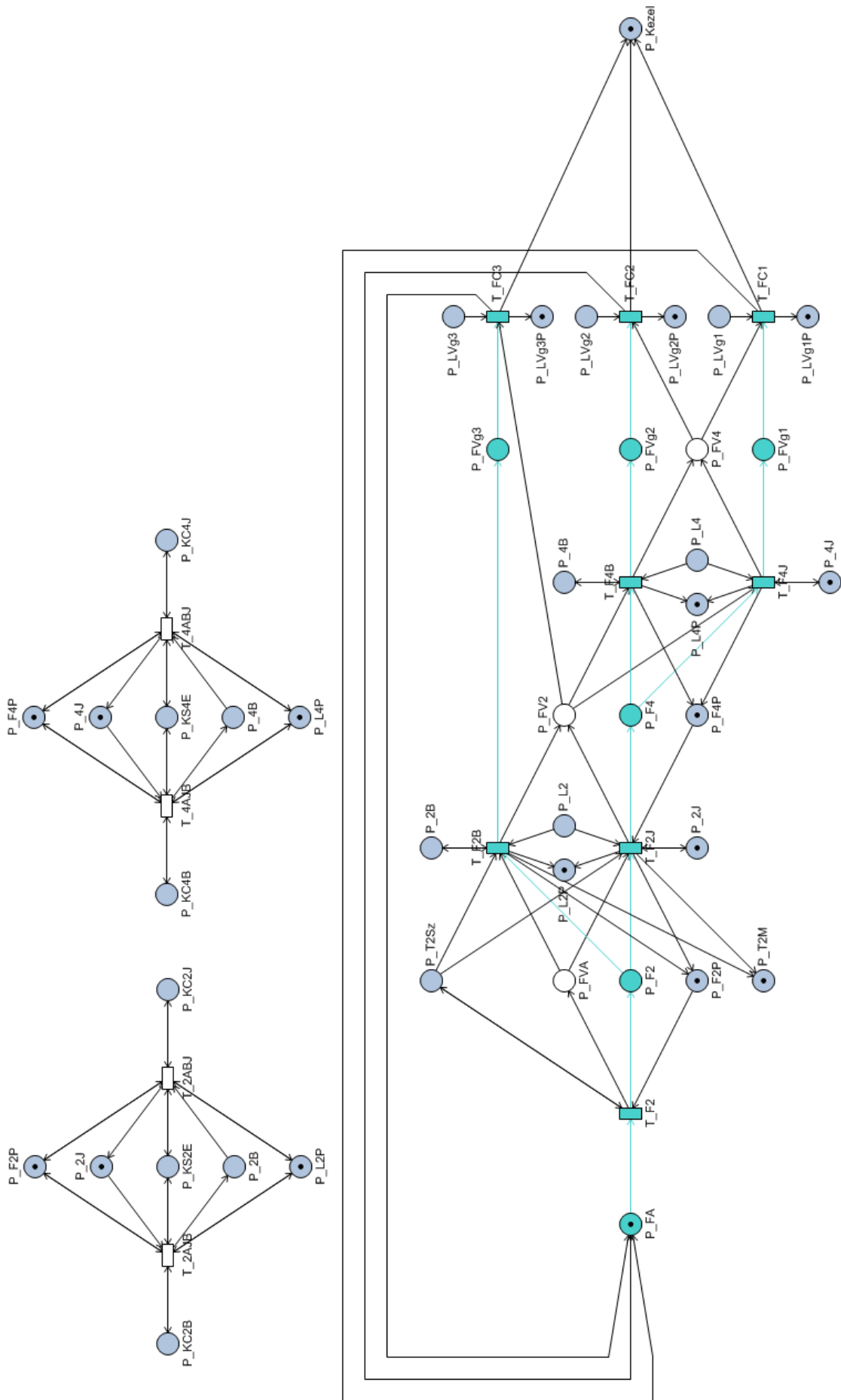
1. táblázat A Magyarországon alkalmazott berendezéstípusok osztályozása kialakítás és szerkesztési elv szerint (forrás: [1] [10] [15]) .....	16
2. táblázat A vágányutas és nyomvonalas elvek összehasonlítása .....	17
3. táblázat Az állomásrész objektumai .....	36
4. táblázat Az állomásrész menetterve .....	39
5. táblázat Az állomásrész elzárási terve.....	39
6. táblázat Az egyes vágányúti elemek szokásos jelölése a modellekben .....	40
7. táblázat A nyomvonal elvű Petri hálós modell egyes objektumainak színezése.....	41
8. táblázat Az elemnevek első betűjének magyarázata a nyomvonalas modellben .....	44
9. táblázat Az elemnevekben szereplő további betűk magyarázata .....	44
10. táblázat A vágányutas elv Petri hálós modelljének elemneveiben szereplő betűk magyarázata...	50
11. táblázat A vágányutas elvű Petri hálós modell egyes funkcióinak színezése .....	50
12. táblázat A nyomvonalas elv UPPAAL modelljének csatornatömbjei.....	54
13. táblázat Példányosításhoz használt változók és értékeik a nyomvonalas elv UPPAAL modelljében .....	55
14. táblázat Objektumok kapcsolata táblázatos formában a nyomvonalas elv UPPAAL modelljében.	55
15. táblázat Az egyes automaták és változók a nyomvonalas elv UPPAAL modelljében .....	57
16. táblázat A vágányutas elv UPPAAL modelljének csatornatömbjei.....	69
17. táblázat Az egyes automaták és változók a vágányutas elv UPPAAL modelljében .....	71
18. táblázat Az ellenőrizendő követelmények .....	79
19. táblázat Az ellenőrizendő követelmények pontosítása nyomvonalas elv, Petri hálós modell esetén .....	79
20. táblázat Az ellenőrizendő követelmények pontosítása vágányutas elv, Petri hálós modell esetén	79
21. táblázat Az ellenőrizendő követelmények pontosítása nyomvonalas elv, UPPAAL modell esetén	80
22. táblázat Az ellenőrizendő követelmények pontosítása vágányutas elv, UPPAAL modell esetén ...	80
23. táblázat Az ellenőrzött formális követelmények nyomvonalas elv, Petri hálós modell esetén.....	80
24. táblázat Az ellenőrzött formális követelmények vágányutas elv, Petri hálós modell esetén .....	81
25. táblázat Az ellenőrzött formális követelmények nyomvonalas elv, UPPAAL modell esetén .....	81
26. táblázat Az ellenőrzött formális követelmények vágányutas elv, UPPAAL modell esetén .....	82
27. táblázat A biztonsági követelményekre elvégzett ellenőrzések teljesülése .....	82
28. táblázat A további ellenőrzött formális követelmények Petri hálós modellek esetén .....	83
29. táblázat A további ellenőrzött formális követelmények UPPAAL modellek esetén .....	84
30. táblázat A további követelményekre elvégzett ellenőrzések teljesülése .....	85
31. táblázat A PetriDotNet és UPPAAL eszközök általános összehasonlítása	<b>Hiba! A könyvjelző nem létezik.</b>
32. táblázat A PetriDotNet és UPPAAL eszközök modelellenőrzőjének összehasonlítása .....	90



# 1. Melléklet: A Nyomvonalas elv Petri hálós modellje

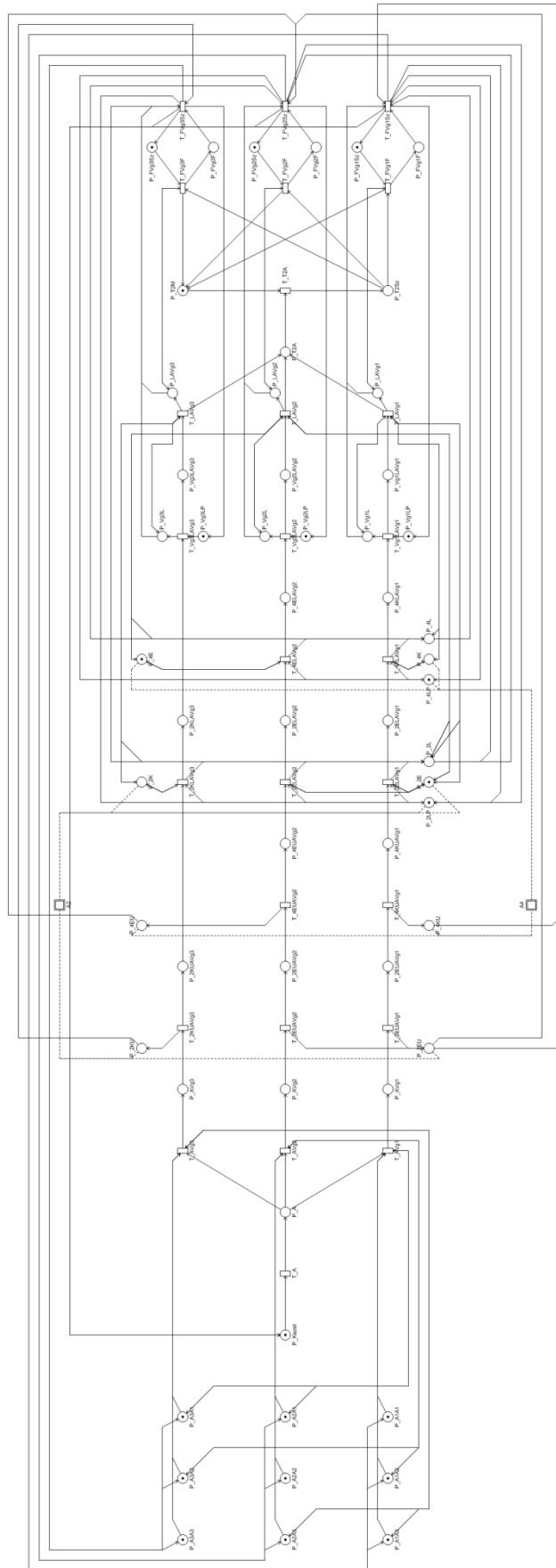


1. ábra A nyomvonalas elvű Petri hálós modell főhálója

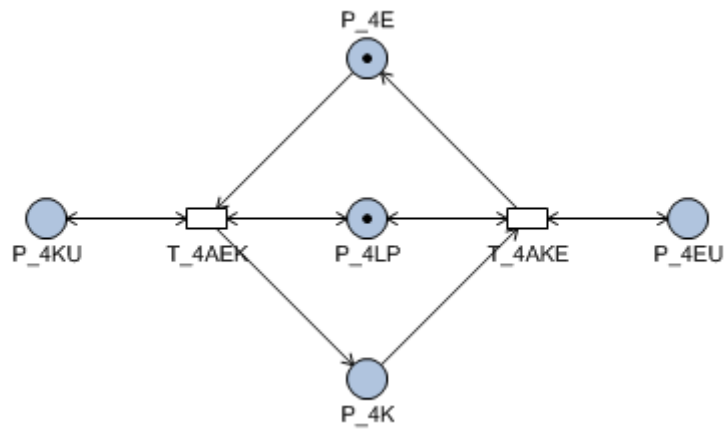
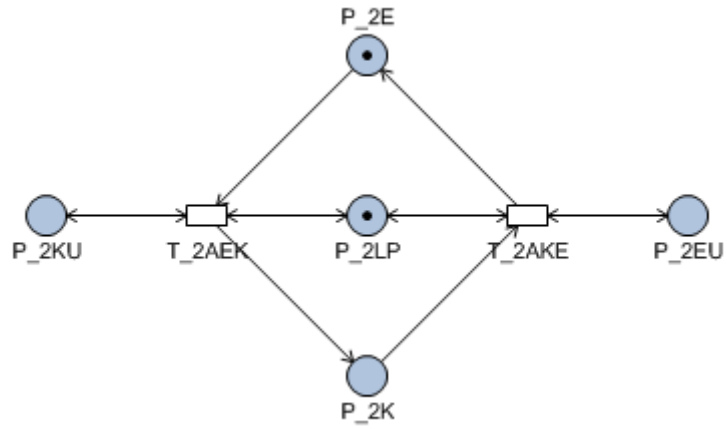


2. ábra A nyomvonalas elvű Petri hálós modell alhálói

## 2. Melléklet: A vágányutas elv Petri hálós modellje



1. ábra A vágányutas elvű Petri hálós modell főhálója



2. ábra A vágányutas elvű Petri hálós modell alhálói

### 3. Melléklet: A nyomvonalas elv UPPAAL modellje

```
//az egyes funkciókhoz tartozó csatornátömbök
chan k_CS[10]; //kijelölés
chan t_CS[10]; //kijelölés törlése
chan l_CS[10]; //lezárás
chan f_CS[10]; //foglaltság

//az egyes alkotórészek darabszáma
const int s_db = 1; //start elemek
const int c_db = 3; //cél elemek
const int vg_db = 3; //vágány elemek
const int v_db = 2; //váltó elemek

//az egyes alkotórészek azonosító típusai
typedef int[1,s_db] id_start;
typedef int[1,c_db] id_cel;
typedef int[1,vg_db] id_vagany;
typedef int[1,v_db] id_valto;

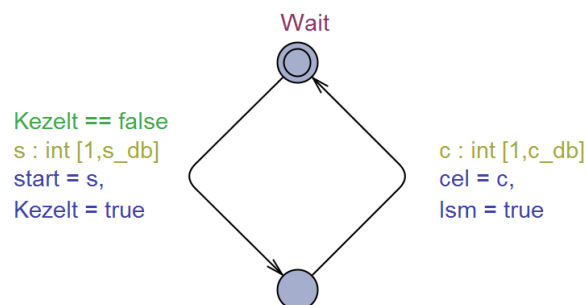
//az egyes elemek szomszédait reprezentáló mátrix
int szomszed[int[1,s_db+c_db+vg_db+v_db]][int[1,4]] = {
{0, 0, 1, 0},
{8, 0, 0, 0},
{7, 0, 0, 0},
{6, 0, 0, 0},
{5, 0, 8, 0},
{4, 0, 7, 0},
{2, 0, 6, 0},
{1, 0, 2, 3},
{3, 0, 4, 5} };

//kiválasztott start és cél
int [0,s_db] start;
int [0,c_db] cel;

//egy kezelés már történt
bool Kezelt = false;

//egy kezelés csak egy kijelölést indíthat
bool Ism = false;
```

1. ábra A nyomvonalas elvű UPPAAL modell globális deklarációi

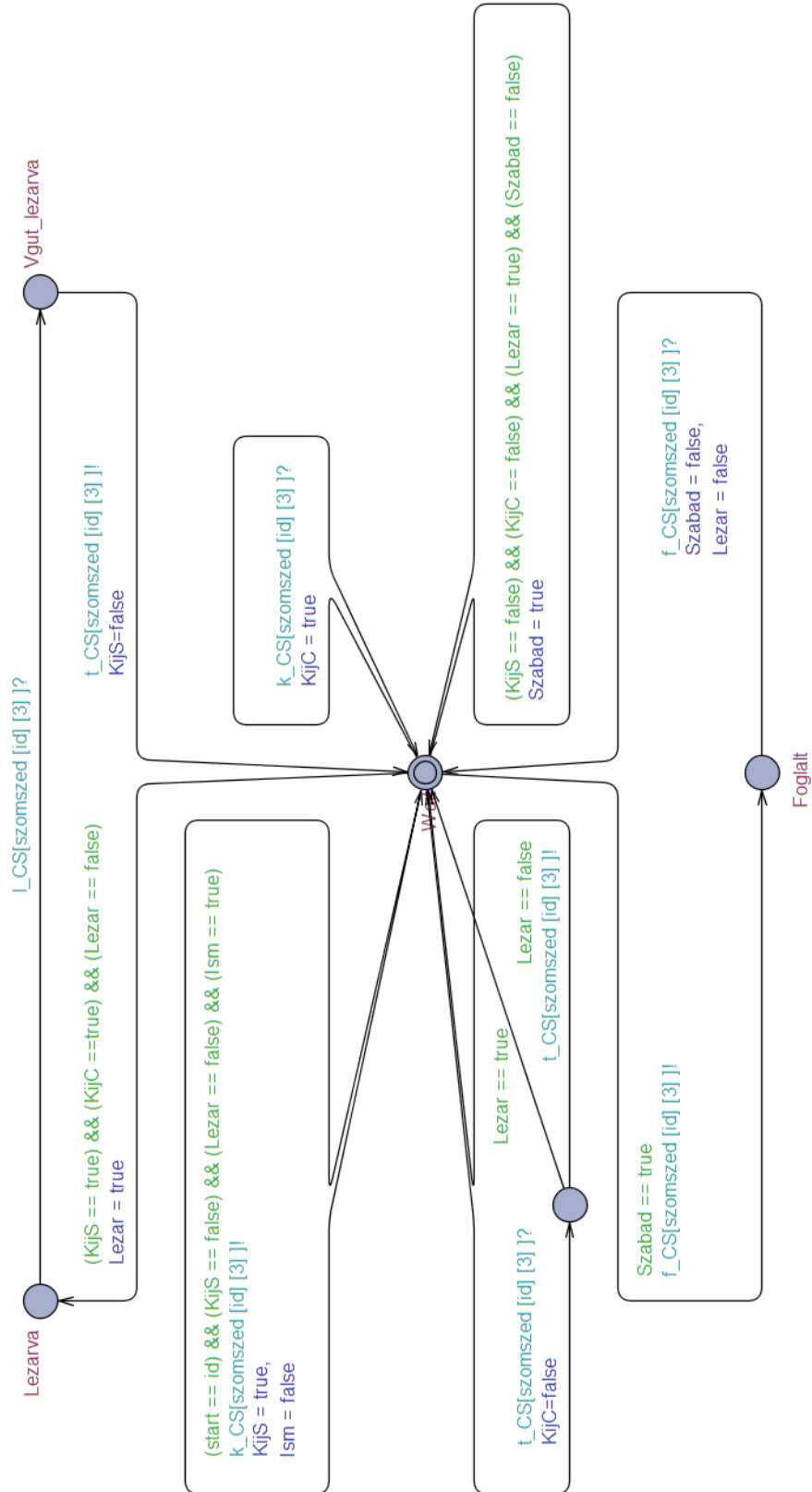


2. ábra A nyomvonalas elvű UPPAAL modell „Kezel” automatája (lokális deklarációkkal nem rendelkezik)

```

bool KijS = false; //elem kijelölve (start felől)
bool KijC = false; //elem kijelölve (cél felől)
bool Lezar = false; //elem lezárva (true=lezárva)
bool Szabad = false; //startjelző szabad (true=szabad)

```



3. ábra A nyomvonalas elvű UPPAAL modell „Start” automatája és lokális deklarációi

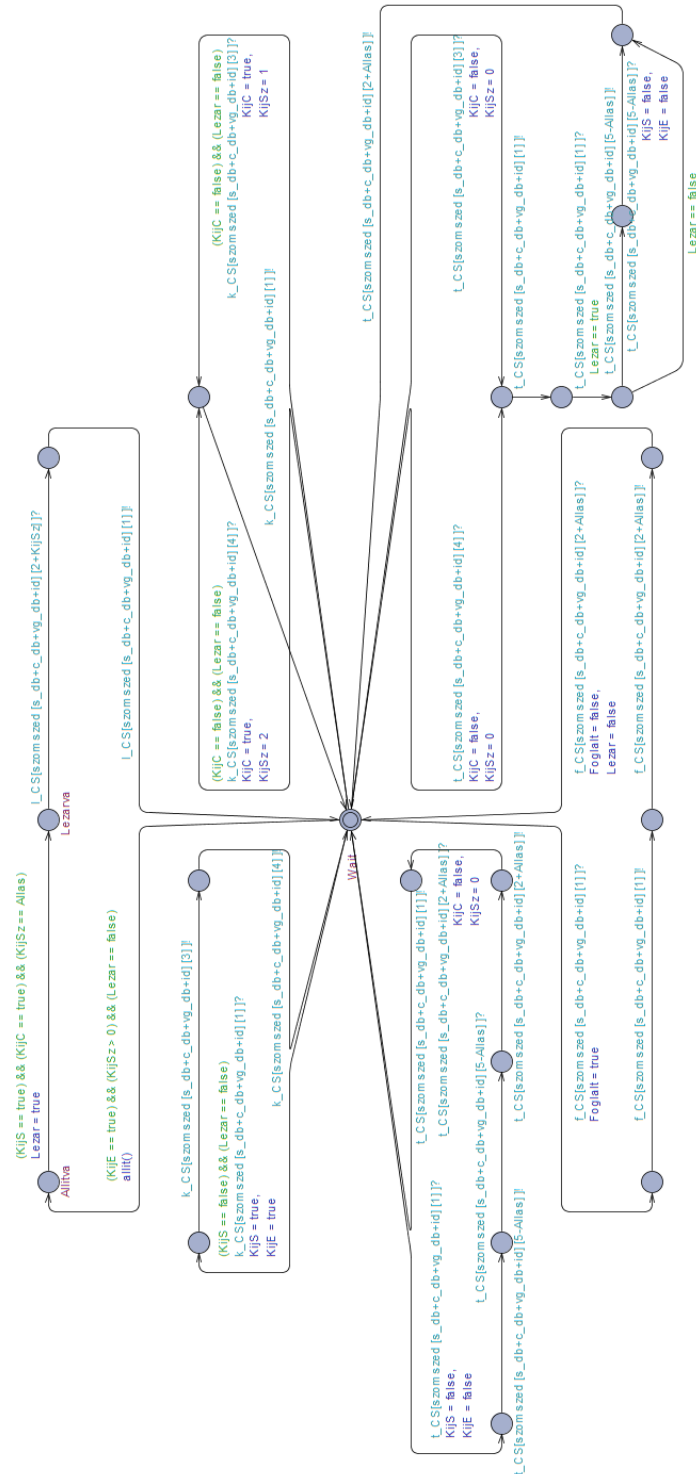
```

bool KijS = false; //elem kijelölve (start felől)
bool KijC = false; //elem kijelölve (cél felől)
bool Lezar = false; //elem lezárva (true=lezárva)
bool Foglalt = false; //elem foglalt (true=foglalt)

bool KijE = false; //a váltó az eleje felől kijelölve
int [0,2] KijSz = 0; //a váltó egyik szára felől kijelölve (0: nincs kijelölve, 1: bal, 2: jobb szára felől)
int [1,2] Allas = 2; //a váltó állása (1: bal, 2: jobb)

void allit() //a váltót állító függvény, a kijelölt szár irányába állítja
{
    if ((Foglalt == false) && (Lezar == false) && (KijE == true) && (KijSz > 0)) Allas=KijSz;
}

```

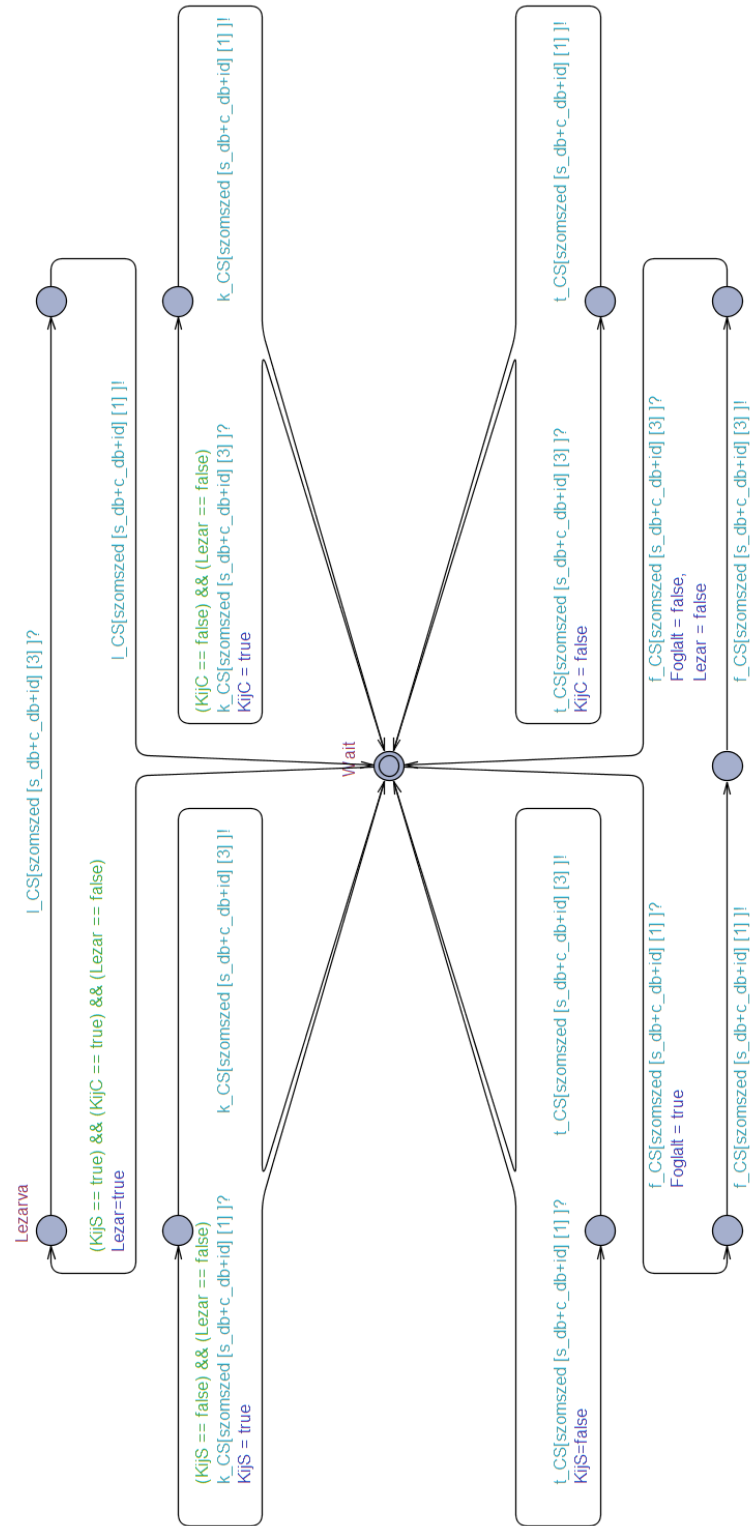


4. ábra A nyomvonalas elvű UPPAAL modell „Valto” automatája és lokális deklarációi

```

bool KijS = false; //elem kijelölve (start felől)
bool KijC = false; //elem kijelölve (cél felől)
bool Lezar = false; //elem lezárva (true=lezárva)
bool Foglalt = false; //elem foglalt (true=foglalt)

```



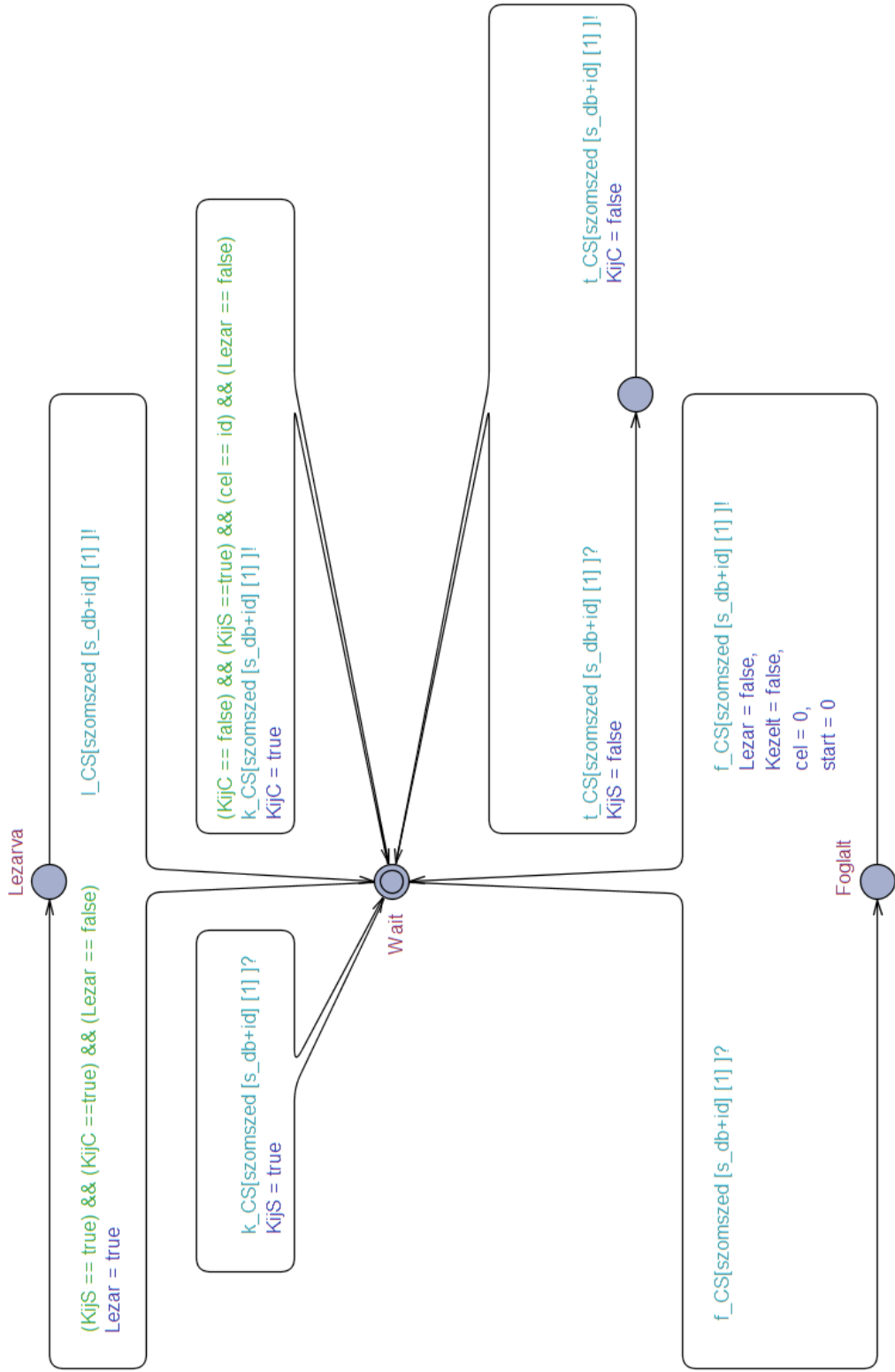
5. ábra A nyomvonalas elvű UPPAAL modell „Vagany” automatája és lokális deklarációi



```

bool KijS = false; //elem kijelölve (start felől)
bool KijC = false; //elem kijelölve (cél felől)
bool Lezar = false; //elem lezárva (true=lezárva)

```



6. ábra A nyomvonalas elvű UPPAAL modell „Cell” automatája és lokális deklarációi

## 4. Melléklet: A vágányutas elv UPPAAL modellje

```
//az egyes vágányúti elemekhez tartozó csatornatömbök

chan vk_CS[10]; //váltó kitérőbe állítás utasítás
chan ve_CS[10]; //váltó egyenesbe állítás utasítás
chan vl_CS[10]; //váltó lezárás információ
chan vg_CS[10]; //vágány lezárás információ

broadcast chan old_CS[10]; //oldó broadcast csatorna

//az egyes alkotórészek darabszáma
const int s_db = 1; //start elemek
const int c_db = 3; //cél elemek
const int vg_db = 3; //vágány elemek
const int v_db = 2; //váltó elemek

//az egyes alkotórészek azonosító típusai
typedef int[1,vg_db] id_vagany;
typedef int[1,v_db] id_valto;

//vágányutak száma
typedef int[0,s_db*c_db] vgut_db;
vgut_db vgut = 0;

//kiválasztott start és cél
int [0,s_db] start;
int [0,c_db] cel;

//a menetterv
const int menetterv[int[1,s_db*c_db]][int[1,s_db*c_db]] = { { 0, 0, 0}, { 0, 0, 0}, { 0, 0, 0} };

//a kizárásokat megvalósító táblázat
int fugges[int[1,s_db*c_db]][int[1,s_db*c_db]] = { { 1, 1, 1}, { 1, 1, 1}, { 1, 1, 1} };
```

### 1. ábra A vágányutas elvű UPPAAL modell globális deklarációi

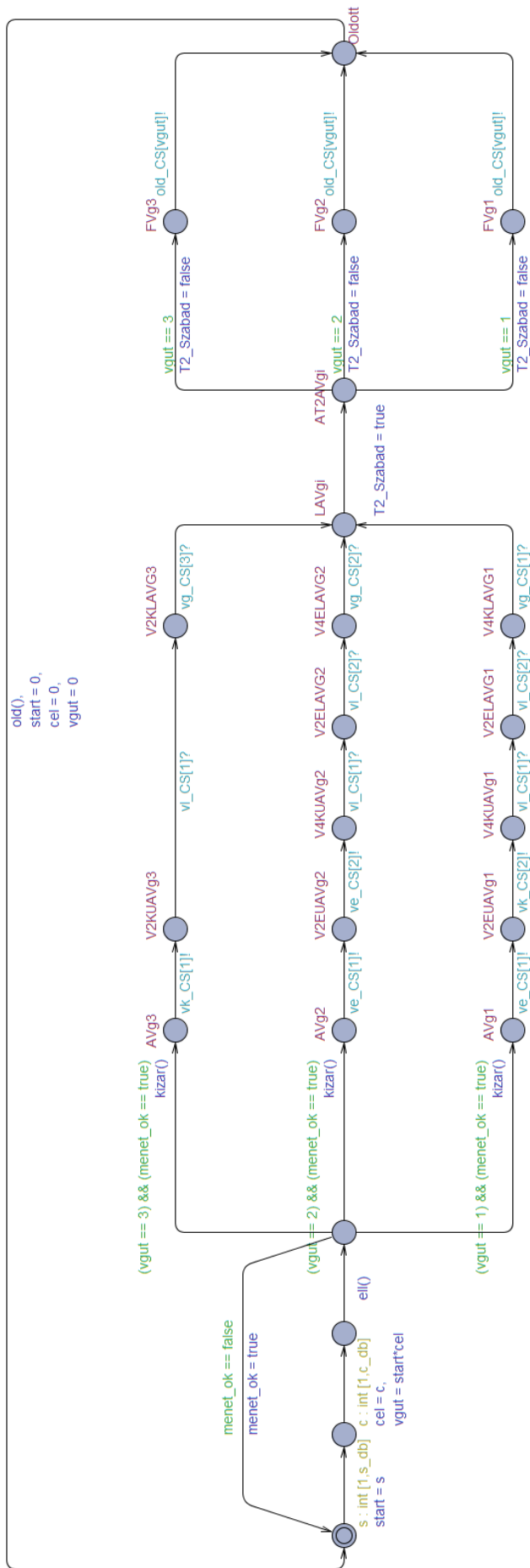
```
bool T2_Szabad = false; //a T2 jelző állapotai
int i; //segédváltozó a menettervi függőségekhez
bool menet_ok = true; //vágányút beállítását engedélyezi

//menettervi függőség ellenőrzése
void ell(){
for (i = 1; i <= s_db*c_db; i++) {
    menet_ok = ((fugges[i][vgut]) * menet_ok);
}
}

//menettervi függőségek beállítása
void kizar(){
for (i = 1; i <= s_db*c_db; i++) {
    fugges[vgut][i] = (fugges[vgut][i]) * (meneterv[vgut][i]);
}
}

//menettervi függőség oldása
void old(){
for (i = 1; i <= s_db*c_db; i++) {
    fugges[vgut][i] = 1;
}
}
```

### 2. ábra A nyomvonalas elvű UPPAAL modell „Menet\_Elzarasi” automatájának lokális deklarációi



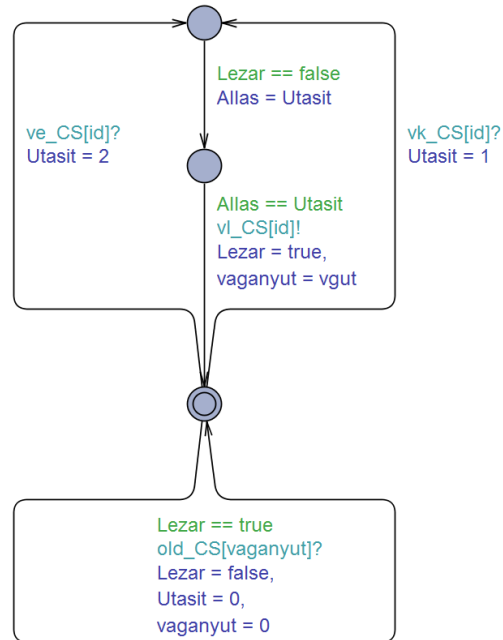
3. ábra A vágányutas elvű UPPAAL modell „Menet\_Elzarasi” automatája

```

bool Lezar = false;    //elem lezárva (true=lezárva)
int vaganyut = 0;     //melyik vágányútban van lezárva?

int [0,2] Utasit = 0; //a váltoállításra vonatkozó utasítás (1: kitérő, 2: egyenes)
int [1,2] Allas = 2;  //a váltó állása (1: kitérő, 2: egyenes)

```

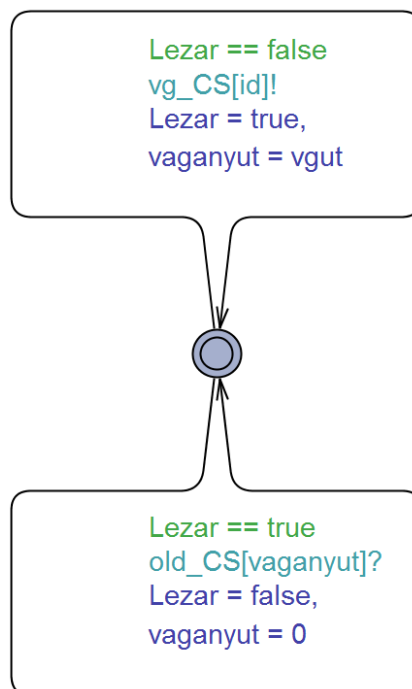


3. ábra A vágányutas elvű UPPAAL modell „Valto” automatája és lokális deklarációi

```

bool Lezar = false;    //elem lezárva (true=lezárva)
int vaganyut = 0;     //melyik vágányútban van lezárva?

```



5. ábra A vágányutas elvű UPPAAL modell „Vagany” automatája és lokális deklarációi