

TDK dolgozat

Pózna Anna Ibolya

Technológiai rendszerek színezett Petri háló alapú modellezése és diagnosztikája

Pannon Egyetem
Műszaki Informatikai Kar
Villamosmérnöki és Információs Rendszerek Tanszék

Pózna Anna Ibolya
Mérnök Informatikus MSc I. évf.

Témavezetők:
dr. Gerzson Miklós egyetemi docens
dr. Leitold Adrien egyetemi docens

Abstract

Identifying faults and analyzing their consequences are important tasks during the investigation of technological systems. A number of diagnostic methods are known in the literature and the model-based methods are very popular among them.

Petri nets and their different extensions (such as colored, timed or hierarchical nets) are powerful tools for modeling discrete event systems. For example colored Petri nets are often used for modeling production lines. It is important that the resulted models describe not only the normal (faultless) operation of the system but they also take account of different, randomly occurring errors occurring in the system.

In my thesis I develop the colored Petri net model of a manufacturing system using CPNTools. The investigated manufacturing system consist of two production lines and a robot serving them. I also build randomly occurring faults into the model using a probabilistic function. I develop both the timed and the untimed models of the manufacturing system then I analyze the operation of the system via simulation and with the occurrence graph of the Petri net.

In many cases the normal or faulty operations of technological processes can be characterized by a series of events having discrete or qualitative valued variables. In this case the occurring deviations can be generated by the comparison of the normal and the actual events. Based on these deviations the P-HAZID diagnostic method can find the cause of the faults using backwards reasoning.

In my thesis I develop the qualitative model based colored Petri net model of a simple tank filling process as an example of a technological process. The resulted model describes both the normal (faultless) and the different faulty operations of the process. The developed Petri net model is also able to generate the deviations belonging to the different operational modes. Taking advantage of the hierarchical modeling I develop the Petri net models of complex technological systems such as different connections of tanks. I investigate how the occurrence graph can be used for identifying the possible faults knowing the real sequence of events from the technological system.

This research is partially supported by the Hungarian State and the European Union under the project TAMOP-4.2.2.A- 11/1/KONV-2012-0072.

Absztrakt

Technológiai rendszerek működésének vizsgálata során fontos feladat a rendszerben esetlegesen fellépő hibák azonosítása és az egyes hibák következményeinek elemzése. A szakirodalomban számos diagnosztikai eljárás ismert, melyek közül a modell alapú módszerek igen elterjedtek.

A Petri hálók és azok különböző kiterjesztései (színezett, időzített, hierarchikus hálók) jól használhatók diszkrét eseményű rendszerek modellezésére. Színezett Petri háló alapú modelleket gyakran alkalmaznak például gyártósorok modellezésére. Ezen modellek elkészítésénél fontos, hogy azok ne csak a normál, hibamentes működését írják le, hanem különböző, véletlenszerűen előforduló hibalehetőséget is figyelembe vegyenek.

Dolgozatomban két gyártósorból és egy azokat kiszolgáló robotból álló rendszer színezett Petri háló modelljét készítem el CPNTools szoftver csomag segítségével, amelybe az ún. hibageneráló függvény segítségével véletlenszerűen fellépő hibalehetőségeket is beépítetek. A modell időzített és időzítetlen változatát is elkészítem, és szimulációval, illetve az elérhetőségi gráf elemzésével vizsgálom a gyártórendszer működését.

Sok esetben a technológiai folyamatok normál vagy hibás működése olyan eseménysorozattal jellemezhető, amelyeknek változói diszkrét vagy kvalitatív értéket vesznek fel. Ilyenkor a normális és tényleges működés eseménysorának összehasonlításával az előforduló eltérések generálhatóak. A P-HAZID diagnosztikai módszer az észlelt eltérések alapján visszafelé irányuló következtetéssel keresi a hiba okát.

Dolgozatomban egy tartály feltöltési folyamat példáján készítem el a technológiai folyamat kvalitatív modell alapú színezett Petri hálóját, amely egyaránt megadja a normál, hibamentes, valamint a különböző hibalehetőségeket rejtő modelljét. Az elkészített Petri háló alkalmas a különböző üzemmódok esetén előálló eltérések generálására is. A Petri hálóval való hierarchikus modellezési lehetőséget felhasználva elkészítem több tartályból álló, különböző kapcsolású rendszerek Petri háló alapú modelljét. Vizsgálom, hogy az elkészített Petri hálók elérhetőségi gráfja miként használható fel egy valós eseménysort ismerve az esetlegesen előforduló hibák azonosítására.

A dolgozat a TÁMOP-4.2.2.A-11/1/KONV-2012-0072 projekt keretében Magyarország és az Európai Unió közös támogatásával készült.

Tartalomjegyzék

I. Bevezetés.....	6
II. Elméleti áttekintés.....	8
1. Színezett Petri hálók.....	8
1.1. A multihalmaz fogalma.....	8
1.2. A színezett Petri háló formális definíciója.....	9
1.3. Színezett Petri hálók dinamikai viselkedése.....	10
1.4. Színezett Petri hálók dinamikai tulajdonságai.....	12
1.5. Színezett Petri háló analízise az elérhetőségi gráf alapján.....	13
2. Időzített színezett Petri hálók.....	15
2.1. Időzített multihalmazok.....	15
2.2. Időzített színezett Petri hálók formális definíciója.....	15
3. Technológiai rendszerek kvalitatív modell alapú diagnosztikája.....	17
3.1. Események, eseménysorok kvalitatív leírása.....	17
3.2. Kvalitatív modell alapú diagnosztikai módszerek.....	18
III. Saját eredmények.....	21
4. Gyártórendszer modellezése színezett Petri hálóval.....	21
4.1. A gyártórendszer bemutatása.....	21
4.2. A gyártórendszer színezett Petri háló modellje hibamentes működés esetén.....	22
4.3. Technológiai rendszerben előforduló hibalehetőségek beépítése a modellbe.....	23
4.4. Időzítés beépítése a színezett Petri háló modellbe.....	24
4.5. Színezés megváltoztatása.....	24
4.6. Szimulációs vizsgálatok.....	26
4.7. Vizsgálat az elérhetőségi gráf alapján a CPNTools-ban.....	26
4.8. Rendszerállapotok bekövetkezési valószínűségeinek számítása az elérhetőségi gráf élsúlyozásával.....	33
5. Technológiai folyamatok kvalitatív modell alapú diagnosztikája színezett Petri háló használatával.....	38
5.1. Eseménysorok reprezentációja színezett Petri hálóval.....	38
5.2. Az elérhetőségi gráfon alapuló diagnosztikai módszer.....	39
5.3. Egyszerű technológiai rendszer diagnosztikai vizsgálata.....	40
5.4. Összetett technológiai rendszerek diagnosztikai vizsgálata.....	45
5.5. Összetett technológiai rendszerek diagnosztikája strukturális dekompozícióval.....	47
IV. Összefoglalás.....	51
V. Irodalomjegyzék.....	52
VI. Mellékletek.....	53

I. Bevezetés

Diszkrét eseményű rendszerek modellezésének és analízisének egyik gyakori alkalmazási területe a különféle gyártórendszerek működésének vizsgálata [1]. A működtetéssel kapcsolatos különféle problémák, mint például hibalehetőségek modellezése, hibák azonosítása, diagnosztikája, dinamikus vizsgálatok, egyes ütemezési problémák, stb. hatékonyan kezelhetők diszkrét eseményű dinamikus modellek használatával, például Petri háló alapú modellek alkalmazásával [2].

Gyártórendszerek esetén a Petri háló alapú modellek hatékony alkalmazhatósága azzal magyarázható, hogy ezek a technológiák gyakran felbonthatók különböző műveleti lépésekre, amelyek egymás után vagy egymással párhuzamosan játszódhatnak le. A gyártórendszerek tervezésekor leggyakrabban az ún. hibamentes működést leíró modellt szokás elkészíteni, de a rendszer működésének vizsgálatára fontos és hasznos az is, ha különböző előforduló hibaeseményeket is beépítünk a modellbe. Korábbi kutatásokból ismert a különböző technológiai hibáknak alacsony szintű Petri háló alapú modellbe történő hatékony beépítése [3]. Ennek során színezett Petri hálóvá alakítva a modellt, a hibalehetőségeket is tartalmazó modell mérete nem növekszik meg lényegesen.

Gyártórendszerek működésének modellezésére jól használhatóak a Petri háló alapú modellek közül a színezett Petri hálók is [4], így jelen dolgozatomban első részében a fenti hibabeépítési eljárást egy olyan gyártórendszerre alkalmazom, ahol a rendszer hibamentes működését is színezett Petri hálóval modellezem. A vizsgálatban elkészítem egy két gyártósorból és az azokat kiszolgáló robotból felépülő gyártórendszer színezett Petri háló alapú modelljét. A modellbe a gyártósorokon fellépő hibalehetőségeket építem be. A Petri hálókban alap esetben az egyes eseményeket modellező átmenetek pillanatszerűen játszódnak le. Ezzel szemben, a valós rendszerek működése során az egyes események általában nem pillanatszerűek, lejátszódási idejük pedig jelentősen befolyásolhatja a technológiai lépések végrehajtását. Ezért elkészítem a vizsgált gyártórendszer időzített színezett Petri hálóját is. A hibamentes, illetve a hibalehetőségeket is magába foglaló színezett Petri háló modelleket időzítés nélküli és időzített esetben is a CPNTools szoftver segítségével állítom elő.

A gyártórendszer működésének vizsgálatára az elkészített Petri háló alapú modellek szimulációs vizsgálata, valamint az elérhetőségi gráfok elemzése ad lehetőséget a CPNTools-ban. Az elérhetőségi gráf analízisének lehetőségei a CPNTools-ban meglehetősen korlátozottak, másrészt a hibaesemények bekövetkezési valószínűségeire vonatkozó információk az elérhetőségi gráfon a CPNTools-ban elvesznek. Így a kutatócsoportban korábban kifejlesztett, OGAnalyzer elnevezésű szoftvert is felhasználok, amellyel az elérhetőségi gráfon élsúlyozást bevezetve az egyes rendszerállapotok bekövetkezési valószínűségei meghatározhatók. A vizsgált gyártórendszer működését a CPNTools-t és az OGAnalyzer szoftvert használva szimulációkkal és elérhetőségi gráf alapján vizsgálom mind hibamentes, mind hibalehetőségeket is magába foglaló változatban, időzítés nélküli és időzített esetekben is.

Diszkrét eseményű rendszereknél sok esetben a technológiai folyamat normál vagy hibás működése olyan esemenysorozattal jellemezhető, amelynek változói diszkrét vagy kvalitatív értékeket vesznek fel. Ilyenkor a normál és a tényleges működés

eseménysorának összehasonlításával az előforduló eltérések generálhatóak. A kvalitatív modell alapú diagnosztikai módszerek egy része, pl. a P-HAZID eljárás [5] az észlelt eltérések alapján visszafelé irányuló következtetéssel keresi a hiba okát.

A diszkrét vagy kvalitatív változókat tartalmazó normális vagy hibás működéseket leíró eseménysorok színezett Petri hálóval is modellezhetőek, és a Petri háló elérhetőségi gráfja egy valós eseménysor ismeretében felhasználható az előforduló hibák azonosítására.

Ezen új diagnosztikai módszer illusztrálására egy tartály feltöltési folyamatának kvalitatív modell alapú színezett Petri hálóját készítettem el, amely egyaránt modellezi a normál (hibamentes) és a különböző hibalehetőségeket leíró eseménysorokat. A Petri háló alkalmas a különböző üzemmódok esetén előálló eltérések generálására is.

A Petri hálóval való hierarchikus modellezési lehetőségeket kihasználva elkészíttem több tartályból álló, különböző kapcsolású tartályrendszerek feltöltési folyamatának Petri háló modelljét is. Bemutatom, hogy az elkészített Petri hálók elérhetőségi gráfja miként használható fel egy valós eseménysort ismerve az előforduló hibák azonosítására.

Az összetett technológiai rendszerekhez tartozó elérhetőségi gráfok túlságosan nagy méretűek, így az azokon történő keresési feladatok időigényesek. Ezért megvizsgálom, hogy összetett rendszerek esetén a teljes rendszer diagnosztikája strukturális dekompozícióval hogyan vezethető vissza az egyes alrendszerek vizsgálatára.

A dolgozat felépítése: A bevezetés után a II. fejezet elméleti áttekintést ad az általam modellezési eszközként felhasznált színezett Petri hálókról, ismertetve a legfontosabb alapfogalmakat és hálótulajdonságokat. Ezután áttekintem a kvalitatív modell alapú diagnosztikai módszereket. A III. fejezet saját eredményeimet tartalmazza: a 4. fejezet ismerteti a gyártórendszer különböző színezett Petri háló alapú modelljeit, majd a gyártórendszer működésének vizsgálatát a CPNTools és az OGAnalyzer szoftverek segítségével. Az 5. fejezetben az új, kvalitatív modell alapú diagnosztikai eljárást ismertetem, különböző példákkal illusztrálva azt. A IV. fejezetben összefoglalom az eredményeimet. A nagyobb méretű ábrák egy része a Mellékletben található.

II. Elméleti áttekintés

1. Színezett Petri hálók

A Petri hálók olyan modellek, amelyek segítségével a vizsgált rendszerek egyszerre grafikus és matematikai módon is leírhatóak. Nevét C.A. Petri német matematikusról kapta, aki 1962-ben publikálta módszerét. A Petri háló előnye, hogy egyszerre ad információt a rendszer struktúrájáról és működéséről is. Alkalmazásának másik előnye, hogy a Petri háló formális analízisével a modellezett rendszer tulajdonságairól további információkat kaphatunk. A Petri hálókat elsősorban diszkrét eseményű és diszkrét idejű rendszerek modellezésére használják, de ezen kívül sok más területen is alkalmazhatók, és számos továbbfejlesztett változatuk van [6]. A következő fejezetekben a színezett Petri hálókhoz kapcsolódó fogalmakat mutatom be. [6][7]

1.1. A multihalmaz fogalma

A színezett Petri háló leírásához szükséges egyik alapvető fogalom a multihalmaz fogalma. A multihalmaz abban különbözik a közönséges halmazoktól, hogy ugyanaz az elem többször is előfordulhat benne. Egy multihalmaz megadásánál tehát az elemek felsorolása mellett meg kell adni azt is, hogy az egyes elemek hányszor fordulnak elő benne. A multihalmazokat mindig egy S nemüres, véges halmaz felett definiáljuk, ami azt jelenti, hogy a multihalmaz elemei az S halmazból származnak. Megadásához S minden eleméről tudnunk kell, hogy hányszor fordul elő a multihalmazban. Ezt általában olyan formális összegzéssel tehetjük meg, ahol S minden elemének van egy együtthatója, amely megadja az adott elem előfordulásainak számát.

Definíció Egy véges, nemüres S halmaz feletti m multihalmazon az $m \in [S \rightarrow \mathbb{N}]$ függvényt értjük, ahol $\mathbb{N}$ a természetes számok halmaza, az $[A \rightarrow B]$ pedig a valamennyi A -ból B -be képező függvény halmaza. Az $m(s) \in \mathbb{N}$ nemnegatív egész szám az $s \in S$ elem előfordulásainak számát adja meg az m multihalmazban. Az m multihalmazt általában a következő formális összeggel adjuk meg: $\sum_{s \in S} m(s) \cdot s$. Az S fölötti összes multihalmaz halmazát S_{MS} -sel jelöljük. Az $\{m(s) | s \in S\}$ nemnegatív egészeket az m multihalmaz együtthatóinak hívjuk, $m(s)$ az s együtthatója. Egy $s \in S$ elem akkor tartozik az m multihalmazhoz, ha $m(s) \neq 0$. Jelölése: $s \in m$.

A multihalmazok között különböző műveleteket definiálhatunk:

Definíció Legyen $m, m_1, m_2 \in S_{MS}$ és $n \in \mathbb{N}$. Ekkor:

1. $m_1 + m_2 = \sum_{s \in S} (m_1(s) + m_2(s)) \cdot s$ (összeadás)
2. $n \cdot m = \sum_{s \in S} (n \cdot m(s)) \cdot s$ (skalárral való szorzás)
3. $m_1 \neq m_2 \Leftrightarrow \exists s \in S : m_1(s) \neq m_2(s)$ (összehasonlítás)
 $m_1 \leq m_2 \Leftrightarrow \forall s \in S : m_1(s) \leq m_2(s)$
4. $|m| = \sum_{s \in S} m(s)$ (számosság)

Ha $|m| = \infty$, akkor az m multihalmaz végtelen, egyébként m véges. Ha $m_1 \leq m_2$, akkor

definiáljuk a kivonást is:

$$5. \quad m_2 - m_1 = \sum_{s \in S} (m_2(s) - m_1(s)) \cdot s \quad (\text{kivonás})$$

1.2. A színezett Petri háló formális definíciója

A színezett Petri háló egy irányított, páros gráf. A gráf csúcspontjait két csoportra oszthatjuk: az egyikbe a körökkel jelölt *helyek* (*place*), a másikba a téglalapokkal jelölt *átmenetek* (*transition*) tartoznak. Az átmenetek a modellezett rendszerben lejátszódó eseményeknek, a helyek az események feltételeinek, illetve következményeinek feleltethetők meg. A helyeket az átmenetekkel irányított élek kötik össze úgy, hogy él mindig csak helyből átmenetbe, vagy átmenetből helybe vezethet. Az élekhez és az átmenetekhez különböző kifejezéseket rendelhetünk, amelyek megadják, hogy a folyamat milyen feltételek mellett, milyen módon játszódik le. Ha a rendszer működése során bekövetkező változásokat is szeretnénk kezelni, akkor a helyekhez jelzőpontokat (*token*) rendelhetünk. A színezett elnevezés arra utal, hogy a helyekhez és a jelzőpontokhoz ún. színeket rendelhetünk, így megkülönböztetve őket egymástól. A formális definíció során meg kell adnunk a színhalmazokat, a gráf csúcsait és éleit, valamint a hozzájuk kapcsolódó különböző logikai kifejezéseket, függvényeket.

A színezett Petri háló formális definíciójához vezessük be a következő jelöléseket:

- $Type(v)$ jelölje egy tetszőleges v változó típusát.
- $Type(expr)$ jelölje egy tetszőleges $expr$ kifejezés típusát.
- $Var(expr)$ jelölje egy tetszőleges $expr$ kifejezésben szereplő változók halmazát.
- $b(VAR)$ jelölje a változók egy VAR halmazának lekötését. A lekötés során a VAR halmaz minden v_i eleméhez hozzárendelünk egy értéket, a következő módon: $\langle v_1 = c_1, v_2 = c_2, \dots, v_n = c_n \rangle, c_i \in Type(v_i), i = 1, \dots, n$.
- $expr \langle b \rangle$ jelölje egy tetszőleges $expr$ kifejezés értékét az adott b lekötésben. A kifejezés kiértékelésekor minden $v \in VAR$ változó esetén behelyettesítjük a $b(v) \in Type(v)$ értékeket, a b lekötésnek megfelelően.
- Zárt kifejezésnek nevezzük az olyan kifejezéseket, amelyek nem tartalmaznak változókat. A zárt kifejezések tetszőleges lekötés esetén kiértékelhetők, és eredményül mindig ugyanazt kapjuk.
- B jelentse a $\{true, false\}$ logikai értékeket tartalmazó halmazt.

Definíció A színezett Petri háló az alábbi kilencessel írható le: $CPN = (\Sigma, P, T, A, N, C, G, E, I)$, ahol

- Σ : színhalmazok véges halmaza; a háló megadása során alkalmazható nemüres, véges színhalmazokat (típusokat) tartalmazza
- P : a helyek véges halmaza;
- T : az átmenetek véges halmaza;
- A : az ívek (élek) véges halmaza, ahol $P \cap T = P \cap A = T \cap A = \emptyset$;
- N : a csomópont-függvény; az ívekhez rendeli hozzá a helyekből és átmenetekből álló rendezett párokat, $N \in [A \rightarrow P \times T \cup T \times P]$;
- C : a színfüggvény; minden helyhez hozzárendeli az ott lehetséges

- jelzőpontszínek halmazát, $C \in [P \rightarrow \Sigma]$; a $p \in P$ helyen lévő minden jelzőpont c színére teljesülnie kell, hogy $Type(c) \in C(p)$.
- G : a korlátfüggvény; az átmenetekhez egy logikai kifejezést rendel hozzá, aminek igaz értéke esetén játszódhat le az átmenet, $G \in [T \rightarrow \{\text{logikai kifejezés}\}]$, ahol $\forall t \in T: [Type(G(t)) = B \wedge Type(Var(G(t))) \subseteq \Sigma]$.
 - E : az ívkifejezés-függvény; az ívekhez rendel hozzá kifejezéseket, melyek meghatározzák, hogy az átmenet hogyan játszódik le. $E \in [A \rightarrow \{\text{kifejezés}\}]$, ahol $\forall a \in A: [Type(E(a)) \in C(p(a))_{MS} \wedge Type(Var(E(a))) \subseteq \Sigma]$, ahol $p(a)$ az a ívhez a csomópont-függvény által rendelt csúcspárból a hely. A kifejezés csak a $p(a)$ helyhez rendelt színek halmazának elemeit tartalmazhatja. Az a ívhez rendelt képelemként egy, az adott helyhez tartozó színhalmaz feletti multihalmazt kapunk.
 - I : az indulóállapot-függvény; a helyekhez rendeli hozzá az induló állapotnak megfelelő jelzőpont értékeket $I \in [P \rightarrow \{\text{zárt kifejezés}\}]$, ahol $\forall p \in P: Type(I(p)) \in C(p)_{MS}$. Az indulóállapot-függvény által a helyekhez rendelt kifejezések nem tartalmazhatnak változókat. Az I függvény által a helyekhez rendelt kifejezés is az adott helyhez tartozó színhalmaz feletti multihalmazba tartozik.

1.3. Színezett Petri hálók dinamikai viselkedése

Az előbbi definíció a Petri háló struktúráját írja le. A tokenek bevezetésével azonban a Petri hálóban bekövetkező változásokat is követni tudjuk. A Petri háló dinamikai viselkedésének leírásakor azt adjuk meg, hogy az átmenetek milyen szabályok szerint játszódhatnak le, valamint hogyan befolyásolják a jelzőpontok eloszlását.

A Petri háló dinamikai viselkedésének leírásához vezessük be a következő jelöléseket:

- $X = P \cup T$ jelölje az összes csúcs halmazát;
- Legyen A_s az a függvény, amely minden x csúcshoz hozzárendeli azokat az éleket, amelynek x kiindulópontja vagy végpontja:
 $A_s \in [X \rightarrow A_{MS}]$, $A_s(x) = \{a \in A \mid \exists x' \in X: [N(a) = (x', x) \vee N(a) = (x, x')]\}$
- A t átmenethez tartozó változók $Var(t)$ halmaza:
 $\forall t \in T: Var(t) = \{v \mid v \in Var(G(t)) \vee \exists a \in A_s(t): v \in Var(E(a))\}$
- Az (x_1, x_2) rendezett párhoz tartozó $E(x_1, x_2)$ kifejezés:
 $\forall (x_1, x_2) \in (P \times T \cup T \times P): E(x_1, x_2) = \sum_{(x_1, x_2) \in N(a)} E(a)$, azaz $E(x_1, x_2)$ az x_1 és x_2 közötti ívekhez tartozó ívkifejezések összege. Az összegzés jól definiált, mivel minden kifejezés típusa azonos.

Definíció Egy t átmenet lekötése egy olyan, az átmenet változóinak $Var(t)$ halmazán értelmezett b függvény, amelyre:

- $\forall v \in Var(t): b(v) \in Type(v)$;
- $G(t) < b > = true$, ahol $G(t) < b >$ a $G(t)$ korlátfüggvény kiértékelése a b

lekötésre.

A t átmenet összes lekötésének halmazát $B(t)$ -vel jelöljük.

Definíció

- Egy jelzőpontelem egy (p, c) pár, ahol $p \in P$ és $c \in C(p)$
- Egy lekötéselem egy (t, b) pár, ahol $t \in T$ és $b \in B(t)$
- Az összes jelzőpontelem halmaza TE
- Az összes lekötéselem halmaza BE
- Egy jelzőponteloszlás legyen egy TE feletti multihalmaz, jele: M
- Egy lépés legyen egy BE feletti nemüres, véges multihalmaz, jele: Y
- Az M_0 induló jelzőponteloszlás az inicializáló kifejezések kiértékelésével kapott jelzőponteloszlás: $\forall (p, c) \in TE: M_0(p, c) = (I(p))(c)$
- Az összes jelzőponteloszlás halmazát jelölje M_Σ , az összes lépés halmazát jelölje Y_Σ .
- Egy átmenet akkor játszódhat le, ha az engedélyezett. Az engedélyezettség azt jelenti, hogy az átmenethez kapcsolódó összes feltétel teljesül, vagyis az összes bemeneti helyen legalább annyi token van, amennyi elegendő ahhoz, hogy az összes ívkifejezés-függvényben szereplő kifejezést kielégítse.

Definíció Egy Y lépés *engedélyezett* egy adott M jelzőponteloszlás esetén, ha a következő feltétel teljesül: $\forall p \in P: \sum_{(t,b) \in Y} E(p, t) < b > \leq M(p)$. (Az $E(p, t) < b >$

kifejezés kiértékelésekor megkapjuk a jelzőpontszínek azon multihalmazát, amelyeket eltávolítunk a p helyről a t átmenet b lekötése során.) Legyen az Y lépés engedélyezett az M jelzőponteloszlás esetén. Ha $(t, b) \in Y$, akkor azt mondjuk, hogy t engedélyezett az M jelzőponteloszlásban (a b lekötésre vonatkozóan). Ha $(t_1, b_1), (t_2, b_2) \in Y$ és $(t_1, b_1) \neq (t_2, b_2)$ akkor azt mondjuk, hogy (t_1, b_1) és (t_2, b_2) , ill. t_1 és t_2 egyidejűleg engedélyezettek. Ha $|Y(t)| \geq 2$, akkor t önmagára nézve párhuzamosan engedélyezett. Ha $Y(t, b) \geq 2$ akkor (t, b) önmagára nézve párhuzamosan engedélyezett.

Egy lépés végrehajtása során a bemeneti helyekről tokeneket távolítunk el, a kimeneti helyekhez tokeneket adunk hozzá. A tokenek színét és számát az ívkifejezés-függvények adott lekötésre való kiértékelése adja meg.

Definíció Y lépés *lejátszódása*: Ha egy Y lépés engedélyezett egy M jelzőponteloszlásban, akkor az le is játszódhat, ennek során az eredeti M_1 jelzőponteloszlás M_2 -re változik, a következő szabály szerint:

$$\forall p \in P: M_2(p) = M_1(p) - \sum_{(t,b) \in Y} E(p, t) < b > + \sum_{(t,b) \in Y} E(t, p) < b > .$$

Az első összeg az egyes helyekről eltávolított jelzőpontoknak, a második a hozzáadott jelzőpontoknak felel meg. Továbbá ez esetben azt mondjuk, hogy M_2 közvetlenül elérhető M_1 -ből az Y lépés lejátszódásával, amit így jelölünk: $M_1[Y > M_2]$.

Ha egy lépés végrehajtása megváltoztatja a tokeneloszlást, akkor az új tokeneloszlásban egy másik lépés válhat engedélyezetté, ami szintén lejátszódhat. Így egy adott tokeneloszlásból lépések sorozatát végrehajtva új tokeneloszlásokba juthatunk el.

Definíció Egy *véges lejátszódási sorozat* jelzőponteloszlások és lépések egy sorozata:

$M_1[Y_1 > M_2[Y_2 > M_3 \dots M_n[Y_n > M_{n+1}]]$, ahol $n \in \mathbb{N}$ és minden $i \in 1, \dots, n$ esetén $M_i[Y_i > M_{i+1}]$. M_1 -t a lejátszódási sorozat induló jelzőponteloszlásának, M_{n+1} -t a végső jelzőponteloszlásnak nevezzük. Az n nemnegatív egész szám a lejátszódási sorozat lépéseinek száma, azaz a *lejátszódási sorozat hossza*. Hasonlóan, a *végtelen lejátszódási sorozat* a jelzőponteloszlások és lépések egy sorozata: $M_1[Y_1 > M_2[Y_2 > M_3 \dots]]$, ahol minden $i \in \mathbb{N}^+$ esetén $M_i[Y_i > M_{i+1}]$. M_1 a lejátszódási sorozat induló jelzőponteloszlása, és a sorozat végtelen hosszú.

Definíció Az M'' jelzőponteloszlás *elérhető* az M' jelzőponteloszlásból, ha létezik olyan véges lejátszódási sorozat, amely M' -ből indul és M'' -ben végződik, azaz ha létezik olyan $Y_1 Y_2 \dots Y_n$ $n \in \mathbb{N}$ lépéssorozat, amelyre $M'[Y_1 Y_2 \dots Y_n > M'']$. Ekkor azt mondjuk, hogy M'' n lépésben elérhető M' -ből. Röviden egy M jelzőponteloszlást elérhetőnek nevezünk, ha M_0 -ből elérhető. Az M' -ből elérhető összes jelzőponteloszlás halmaza $R(M')$.

1.4. Színezett Petri hálók dinamikai tulajdonságai

Korlátosság

A korlátosság vizsgálatakor arra vagyunk kíváncsiak, hogy az egyes helyeken a jelzőpontok száma hogyan változik, mekkora az adott színű tokenek számának minimuma, ill. maximuma egy adott helyen.

Definíció Legyen $X \subseteq TE$ a jelzőpontelemek egy halmaza, és $n \in \mathbb{N}$.

- n felső korlátja X -nek, ha $\forall M \in R(M_0): |M|_X \leq n$.
- n alsó korlátja X -nek, ha $\forall M \in R(M_0): |M|_X \geq n$. Az $n=0$ triviális alsó korlát minden X -re.

Az $|M|_X$ kifejezés az X jelzőpontelem-halmazbeli elemek előfordulásának számát adja meg az M jelzőponteloszlásban.

Az X halmazt *korlátosnak* nevezzük, ha van felső korlátja. A cél általában a legnagyobb alsó korlát, illetve a legkisebb felső korlát megkeresése. A korlátosságot értelmezhetjük a helyekre és magára a színezett Petri hálóra is:

Definíció Legyen $p \in P$ egy hely, $m \in C(p)_{MS}$ egy multihalmaz és $n \in \mathbb{N}$ egy nemnegatív egész szám.

- Az n szám felső egész korlát p -re, ha $\forall M \in R(M_0): |M(p)| \leq n$.
- Az m multihalmaz felső multihalmaz korlát p -re, ha $\forall M \in R(M_0): M(p) \leq m$.

Az alsó korlátokat hasonlóan definiálhatjuk. A p helyet *korlátosnak* nevezzük, ha van felső korlátja. Ha a színezett Petri hálóban minden hely korlátos, akkor a hálót korlátosnak nevezzük.

Alapállapot

Egy Petri háló alapállapota azt mutatja meg, hogy van-e olyan jelzőponteloszlás, azaz állapot, amelyhez mindig vissza tudunk térni.

Definíció Legyen $M \in M_\Sigma$ egy jelzőponteloszlás, és $X \subseteq M_\Sigma$ jelzőponteloszlások egy halmaza. Ekkor:

- M egy alapállapot jelzőponteloszlás, ha $\forall M' \in R(M_0): M \in R(M')$

- X egy *alapállapot*, ha $\forall M' \in R(M_0): X \cap R(M') \neq \emptyset$.

Ha a Petri hálóban van alapállapot jelzőponteloszlás, akkor megvan a lehetőség arra, hogy elérjük ezt, de egyáltalán nem biztos, hogy el is érjük.

Élőség

Egy lekötéselem élősége azt mutatja meg, hogy adott állapotból indulva van-e olyan lépéssorozat, amelyben a lekötéselem engedélyezetté válik.

Definíció Legyen $M \in M_\Sigma$ egy jelzőponteloszlás, és $X \subseteq BE$ a lekötéselemek egy halmaza. Ekkor:

- M halott, ha egy lekötéselem sem engedélyezett, azaz $\forall x \in BE: \neg M'[x]$.
- X halott az M jelzőponteloszlásban, ha X egyik eleme sem válhat engedélyezetté, azaz $\forall M' \in R(M) \forall x \in X: \neg M'[x]$.
- X élő, ha nincs olyan elérhető jelzőponteloszlás, amelyben X halott, azaz $\forall M' \in R(M_0) \exists M'' \in R(M') \exists x \in X: M''[x]$.

Röviden azt mondjuk, hogy X halott, ha halott M_0 -ban. Egy M tokeneloszlás halott, ha BE halott M -ben. Fontos megjegyezni, hogy az élő tulajdonság nem a halott ellentéte. Például a lekötéselemek minden élő halmazára igaz, hogy nem halott, de az állítás fordítva már nem igaz.

1.5. Színezett Petri háló analízise az elérhetőségi gráf alapján

Ha elkészítettük a vizsgált rendszerünk színezett Petri háló modelljét, akkor a legegyszerűbben szimulációval győződhetünk meg a modell működésének helyességéről. Egy nagyobb rendszer modellezésénél igen fontos szerepe van a szimulációnak a tervezés, illetve a modell validálása során. Azonban csak szimulációval nem lehet meghatározni a modell összes tulajdonságát. Ezért szükség van olyan módszerekre, amelyek a Petri háló formális analízisét biztosítják. Az egyik ilyen módszer a Petri háló elérhetőségi grájának vizsgálata.

Elérhetőségi gráf fogalma

Egy színezett Petri háló elérhetőségi grájja egy olyan irányított gráf, amely az elérhető tokeneloszlásokat és az előforduló lekötéselemeket tartalmazza. Az elérhetőségi gráf segítségével a Petri háló összes dinamikai tulajdonsága meghatározható. Az elérhetőségi gráf generálása jól automatizálható folyamat. Hátránya, hogy a gráf mérete már kis színezett Petri hálók esetén is nagyon nagy méretű lehet.

Definíció A színezett Petri háló teljes *elérhetőségi grájja* egy $OG = (V, A, N)$ irányított gráf, ahol

- V a csúcsok halmaza, ahol a csúcsok az M_0 induló állapotból elérhető állapotok: $V = R(M_0)$;
- A az élek halmaza, ahol egy él az M_1 állapotból az M_2 állapotba vezető lépésnek felel meg: $A = \{ (M_1, b, M_2) \in V \times BE \times V \mid M_1[b]M_2 \}$;
- N a csomópont függvény, amely minden élhez hozzárendeli annak induló és

végpontját: $\forall a=(M_1, b, M_2) \in A: N(a)=(M_1, M_2)$.

Az elérhetőségi gráf egy csúcsa egy tokeneloszlásnak (állapotnak), egy éle pedig egy lekötéselemnek felel meg. Két csúcs között akkor van él, ha a kiinduló csúcs tokeneloszlását egy lépés végrehajtása a másik csúcsba viszi át, az élnak megfelelő lekötésben.

Az elérhetőségi gráffal vizsgálhatjuk például, hogy egy adott állapotból milyen állapotok érhetők el, egy állapotból egy másik állapot elérhető-e, vannak-e olyan állapotok, amelyek esetén nincs további engedélyezett átmenet stb.

2. Időzített színezett Petri hálók

Az átmenetek lejátszódását az eddigi színezett Petri hálókból pillanatszerűnek tekintettük. Az ilyen hálókból csak az átmenetek lejátszódásának egymáshoz képesti sorrendjéről van információnk, arról nincs, hogy az egyes események ténylegesen mennyi ideig tartanak. A valós rendszerek modellezése során azonban fontos szerepe lehet az események időbeli lejátszódásának.

2.1. Időzített multihalmazok

Definíció Legyen $R \subset \mathbb{R}_0^+$. Ekkor egy nemüres S halmaz felett értelmezett tm időzített multihalmazon egy olyan $tm \in [S \times R \rightarrow \mathbb{N}]$ függvényt értünk, amelyre a $tm(s) = \sum_{r \in R} tm(s, r)$ összeg véges, minden $s \in S$ esetén. A $tm(s)$ nemnegatív egész

szám az s elem előfordulásainak számát adja meg a tm időzített multihalmazban. A $tm[s] = [r_1, r_2, \dots, r_{tm(s)}]$ lista megadja azokat az $r \in R$ idő értékeket, amelyekre $tm(s, r) \neq 0$. Minden listában $tm(s, r)$ darab időérték fordul elő. A listát így rendezzük:

$$r_i \leq r_{i+1} \quad \forall i \in 1 \dots tm(s) - 1.$$

A tm időzített multihalmazt általában a következő formális összeggel adjuk meg:

$\sum_{s \in S} tm(s) \cdot s @ tm[s]$. Az S felett értelmezett összes időzített multihalmaz halmazát S_{TMS} -sel jelöljük. A $\{tm(s) | s \in S\}$ nemnegatív egészeket az tm időzített multihalmaz együtthatóinak hívjuk, $tm(s)$ az s együtthatója. Egy $s \in S$ elem akkor tartozik a tm időzített multihalmazhoz, ha $tm(s) \neq 0$. Jelölés: $s \in tm$. Valamint minden $tm \in S_{TMS}$ időzített multihalmaz meghatároz egy közösleges $tm_U \in S_{MS}$ multihalmazt, a következő módon: $tm_U = \sum_{s \in S} tm(s) \cdot s$.

2.2. Időzített színezett Petri hálók formális definíciója

A színezett Petri hálókhoz hasonlóan definiálhatjuk az időzített színezett Petri hálókat is, néhány változtatást bevezetve. Az időzített színezett Petri hálók esetén a színhalmazok időzítettek lehetnek, és az ilyen színhalmazba tartozó jelzőpontokhoz időbélyegeket rendelhetünk. Az időbélyeg azt a legkorábbi időt adja meg, amikortól a token felhasználható. Továbbá az események lejátszódási idejét is megadhatjuk az átmenethez rendelt lejátszódási idővel, vagy az átmenet kimenő éléhez rendelt késleltetéssel. Az időt egy globális órához viszonyítjuk. A Petri hálót időzítettnek tekintjük, ha van legalább egy időzített színhalmaza.

Definíció Egy időzített színezett Petri hálón a $TCPN = (CPN, R, r_0)$ hármast értjük, ahol

- CPN teljesíti a színezett Petri háló definícióját, azzal a különbséggel, hogy $E(a)$ ill. $I(p)$ típusa egy $C(p(a))$ ill. $C(p)$ felett értelmezett időzített vagy nem időzített multihalmaz lehet.
- R az időbélyegek halmaza, $R \subset \mathbb{R}_0^+$.
- $r_0 \in R$ a start idő.

A nem időzített színhalmazokba tartozó tokenek bármikor rendelkezésre állnak, felhasználhatók.

Definíció Egy *jelzőponteloszlás* egy TE felett értelmezett időzített multihalmaz. Az induló jelzőponteloszlás M_0 az inicializáló kifejezések kiértékelésével kapott jelzőponteloszlás:

$$\forall p \in P: M_0(p) = I(p)_{r_0}.$$

Egy *állapot* egy (M, r) pár, ahol M egy jelzőponteloszlás és r egy idő érték. A kezdő állapot az (M_0, r_0) pár. Az összes jelzőponteloszlás halmazát jelölje M_Σ , míg az össze állapot halmazát jelölje S_Σ .

A lépések engedélyezettségénél figyelembe kell venni azt is, hogy az adott időpontban a tokenek felhasználhatók-e.

Definíció Egy Y lépés engedélyezett az (M_1, r_1) állapotban és az r_2 időpontban, ha teljesülnek a következő feltételek:

- $\forall p \in P: \sum_{(t,b) \in Y} E(p, t) \cdot b \leq M_1(p);$
- $r_1 \leq r_2;$
- r_2 az R halmaz legkisebb eleme, amely esetén létezik olyan lépés amely teljesíti az előző két feltételt.

Definíció Y lépés lejátszódása: Ha egy Y lépés engedélyezett egy (M_1, r_1) állapotban az r_2 időpontban, akkor lejátszódhat, az (M_1, r_1) állapotot (M_2, r_2) -re változtatva, ahol M_2 -t az alábbi módon kapjuk meg:

$$\forall p \in P: M_2(p) = M_1(p) - \sum_{(t,b) \in Y} E(p, t) \cdot b + \sum_{(t,b) \in Y} E(t, p) \cdot b.$$

Az első összeg a p helyről eltávolított tokeneknek, a második pedig a hozzáadott tokeneknek felel meg. Az Y lépés lejátszódása után a jelzőpontok időbélyegeit úgy kapjuk meg, hogy a modell időhöz hozzáadjuk az átmenet, ill. él késleltetését. Továbbá azt mondjuk, hogy az (M_2, r_2) közvetlenül elérhető az (M_1, r_1) állapotból az Y lépés lejátszódásával az r_2 időpontban, amit így jelölünk: $(M_1, r_1) [Y, r_2] (M_2, r_2)$.

A lejátszódási sorozatot és az elérhetőséget hasonlóan definiáljuk, mint a nem időzített Petri hálónál.

Definíció Egy *véges lejátszódási sorozat* állapotok, lépések és időértékek egy sorozata: $S_1[Y_1, r_2] S_2[Y_2, r_3] S_3 \dots S_n[Y_n, r_{n+1}] S_{n+1}$, ahol $n \in \mathbb{N}$ és minden $i \in 1, \dots, n$ esetén $S_i[Y_i, r_{i+1}] S_{i+1}$. S_1 -t a lejátszódási sorozat *induló állapotának*, S_{n+1} -t a *végző állapotnak* nevezzük.

Az n nemnegatív egész szám a lejátszódási sorozat lépéseinek száma, azaz a lejátszódási sorozat hossza. Hasonlóan, egy *végtelen lejátszódási sorozat* állapotok, lépések és időértékek egy sorozata: $S_1[Y_1, r_2] S_2[Y_2, r_3] S_3 \dots$, ahol minden $i \in \mathbb{N}^+$ esetén $S_i[Y_i, r_{i+1}] S_{i+1}$. S_1 a lejátszódási sorozat induló állapota, és a sorozat végtelen hosszú.

Definíció Az S'' állapot elérhető az S' állapotból, ha létezik olyan véges lejátszódási sorozat, amely S' -ből indul és S'' -ben végződik. Röviden egy S állapotot elérhetőnek nevezünk, ha S_0 -ból elérhető. Az S' -ből elérhető összes állapot halmaza $R(S')$.

3. Technológiai rendszerek kvalitatív modell alapú diagnosztikája

3.1. Események, eseménysorok kvalitatív leírása

Kvalitatív értéktartományok

Technológiai rendszerek vizsgálatakor nem mindig szükséges a változók pontos értékének ismerete. Sok esetben elegendő, ha egy változó értékéről csak annyit tudunk, hogy egy bizonyos értéktartományba esik-e. Például egy tetszőleges érzékelő méréstartományát feloszthatjuk a következőképpen:

$$Q_e = \{e-, 0, L, N, H, e+\},$$

ahol $0, L, N, H$ jelenti a nulla, alacsony, normál, magas szintet, $e-$ és $e+$ pedig a mérési tartományon kívül eső nagyon alacsony, illetve nagyon magas értéket, amik a szenzor meghibásodásából adódnak. Amennyiben szükséges a felosztás tovább finomítható, ezáltal a változók értéke pontosabban adható meg. Egy beavatkozó szerv állapota hasonló módon írható le. Például egy kétállású szelep állapota a $Q_{sz} = \{open, close\}$ értéktartománnyal jellemezhető, ahol *open* a szelep nyílt, *close* a szelep zárt állapotának felel meg.

Események, trace-ek, eltérések

Egy technológiai rendszer működtetése során az operátorok különböző műveleteket hajtanak végre. Egy technológiai folyamat megadható a folyamat lejátszódása során bekövetkezett események sorozatával. Egy *eseményt* annak bekövetkezési idejével (t), valamint a bemenő és kimenő változók t időponthoz tartozó értékeivel azonosíthatunk:

$$event_t = (t; input_1, \dots, input_n; output_1, \dots, output_m)$$

A t ebben az esetben nem konkrét időpontot jelent, hanem egy sorszám, amely azt jelöli, hogy az adott esemény hányadik lépésben következett be. Két időpont között bekövetkezett, egymást követő események sorozatát *trace*-nek nevezzük:

$$T(t_1, t_n) = event_{t_1}, \dots, event_{t_n}$$

Egy *trace*-en belül az események ugyanazokat a bemeneti és kimeneti változókat tartalmazzák. A technológiai rendszer vizsgálata során többféle *trace*-t különböztetünk meg. A *nominális trace* megadja, hogy a folyamat normál (hibamentes) lejátszódásakor mely események következtek be. Minden folyamathoz egy nominális *trace* tartozik. A lehetséges hibás működésekhez tartozó *trace*-ek a hiba hatására bekövetkező eseménysort adják meg. A vizsgált folyamat egy konkrét lejátszódását leíró eseménysort *karakterisztikus trace*-nek nevezzük. A nominális és a technológiai rendszerből származó karakterisztikus *trace*-t összehasonlítva meghatározhatók a két eseménysorozat közötti *eltérések*. Az eltérések időbeliek és mennyiségi lehetnek. Időbeli eltérésről akkor beszélünk, amikor a vizsgált karakterisztikus *trace*-ben szereplő esemény a nominális *trace*-hez képest eltérő időpontban következett be. Ez alapján háromféle időbeli eltérést különböztetünk meg:

- *later*: a bemenő és a kimenő változók értékeinek egy adott kombinációja a

karakterisztikus trace-ben későbbi időpontban jelenik meg, mint a nominális trace-ben.

- *earlier*: a bemenő és a kimenő változók értékeinek egy adott kombinációja a karakterisztikus trace-ben korábbi időpontban jelenik meg, mint a nominális trace-ben.
- *never happened*: a bemenő és a kimenő változók értékeinek egy adott kombinációja, amely a nominális trace-ben jelen van, a karakterisztikus trace-ből hiányzik.

Mennyiségi eltérések vizsgálatakor a trace-ek azonos időponthoz tartozó eseményeiben szereplő kimenő változók értékeit hasonlítjuk össze. A mennyiségi eltérések kétfélek lehetnek:

- *smaller var_out_i*: a karakterisztikus trace-ben egy adott időpontban az *i*-edik kimenő változó értéke kisebb, mint a nominális trace-ben
- *greater var_out_i*: a karakterisztikus trace-ben egy adott időpontban az *i*-edik kimenő változó értéke nagyobb, mint a nominális trace-ben

Az eltéréseket egy kulcsszó és egy nominális esemény párossal jelöljük: *kulcsszó(nominális esemény)*, ahol a kulcsszó az eltérés típusát jelenti, a nominális esemény pedig az az esemény, amihez viszonyítjuk az eltérést. Például az *earlier(2,open,close,L)* eltérés azt jelenti, hogy a *(2,open,close,L)* esemény az elvártnál korábban következett be a folyamat során. Egy karakterisztikus trace és a nominális trace eltérései alapján következtetni lehet arra, hogy a folyamat során történt-e hiba. Ha van eltérés a két trace között, akkor a folyamat során valamilyen hiba következett be. Ebben az esetben a kapott eltérések listájából visszafelé következtetve az is meghatározható, hogy mik azok a lehetséges hibaokok, amik az adott lejátszódáshoz vezettek.

3.2. Kvalitatív modell alapú diagnosztikai módszerek

A szakirodalomban és a gyakorlatban számos olyan diagnosztikai eljárás ismert, mely alkalmas a működés során fellépő hibák, illetve adott működési állapotból bekövetkező kívánt és nem kívánt állapotok meghatározására. Ezen módszerek közül az FMEA (*Failure Mode and Effect Analysis*), a HAZOP (*Hazard and Operability Analysis*), a P-HAZID (*Procedure Hazard Identification*) és hibafa-elemző módszereket mutatom be.

FMEA

Az FMEA egy rendszerben előforduló lehetséges hibák és azok hatásának elemzésével foglalkozik. Az FMEA célja a hibák korai felismerése és azonosítása, a hibákból eredő kockázatok becslése és számszerűsítése. Az FMEA során előre felé következtető módszert alkalmazunk a hibák hatásának vizsgálatára. Az FMEA eredményeként egyre jobb minőségű termékek egyre gazdaságosabb módon történő előállítására van lehetőség. Az FMEA-t széles körben alkalmazzák a feldolgozóiparban egy termék életciklusának különböző fázisaiban, de ezen kívül sok más területen is használható. Az FMEA-nak a felhasználási területek szerint több fajtája létezik, mint például konstrukciós FMEA, rendszer FMEA, folyamat FMEA (P-FMEA).[8]

HAZOP

A HAZOP egy kvalitatív kockázatelemző módszer, amelynek célja valamilyen folyamatban előforduló potenciális veszélyek és működési problémák azonosítása. A HAZOP módszer alapja az a feltevés, hogy a problémák a kívánt működéstől való eltérések miatt jönnek létre. Az eltérések jelölésére különböző kulcsszavakat használnak, amelyek az eltérés típusára jellemzőek. Az ilyen eltérések vizsgálatával egyrészt feltárhatók a kiváltó hiba okok, másrészt megadhatók a lehetséges következmények. A HAZOP módszert eredetileg vegyipari folyamatok vizsgálatára fejlesztették ki, de később kiterjesztették más típusú rendszerekre is (például atomerőművek). [9]

P-HAZID

A HAZOP és FMEA eljárások a technológiai rendszerekben előforduló hibák azonosítására, a hibák hatásának vizsgálatára széles körben alkalmazott módszerek. A két eljárás összevonásával és kiterjesztésével egy új, *procedure HAZID*-nak (röviden P-HAZID) nevezett eljárás hozható létre. A P-HAZID analízis alapja egy eltéréseket és kiváltó okaikat tartalmazó táblázat, aminek minden sora egy a technológiai rendszerben előforduló eltérést, ennek kiváltó okát és következményét tartalmazza. A P-HAZID táblázat alapján egy adott hibás működéshez tartozó eltérésekből meghatározhatók a hibás működést előidéző lehetséges okok. A vizsgálat során feltételezzük, hogy a kiváltó ok a folyamat kezdete előtt jött létre, és nem változott a technológiai rendszer működése során. A P-HAZID eljárás során visszafelé következtetünk az eltérésekből az őket kiváltó okokra a következőképpen: A táblázat két sora összekapcsolható egymással, ha az egyik sorban az (*eltérés, következmény*) pár és a másik sorban a (*ok, eltérés*) pár megegyezik egymással. A sorok összekapcsolását addig folytatjuk, amíg el nem jutunk az adott működést kiváltó okig, vagy amíg nem találunk olyan sort, ami a vizsgált sor *ok* mezőjében szereplő elemet tartalmazza [5].

1.táblázat: Részlet egy P-HAZID táblázatból.

Ok	Eltérés	Következmény
TANK-LEAK	NH(2,op,cl,L)	NH(3,op,cl,N)
NH(2,op,cl,L)	NH(3,op,cl,N)	NH(4,op,op,N)
TANK-LEAK	SML(2,op,cl,L)	SML(3,op,cl,N)
SML(2,op,cl,L)	SML(3,op,cl,N)	SML(4,op,op,N)
NEG-BIAS	SML(1,op,cl,0)	SML(2,op,cl,L)
SML(1,op,cl,0)	SML(2,op,cl,L)	SML(3,op,cl,N)
SML(2,op,cl,0)	SML(3,op,cl,L)	SML(4,op,op,N)
POS-BIAS	GRE(1,op,cl,0)	GRE(2,op,cl,L)
GRE(1,op,cl,0)	GRE(2,op,cl,L)	GRE(3,op,cl,N)
GRE(2,op,cl,0)	GRE(3,op,cl,L)	GRE(4,op,op,N)

Hibafa-elemzés

A *hibafa-elemzés* egy top-down, visszafelé következtető módszer, amelynek célja, hogy meghatározza azokat az elemi hibaeseményeket, amelyek a fennálló nem kívánt állapothoz vezettek. A hibafa egy rendszerben előforduló kritikus esemény kiváltó okainak és következményeinek logikai kapcsolatát megjelenítő diagram. A hibafa kiegészíthető az egyes hibák valószínűségeivel is, így felderíthetők azok az események, amelyek megelőzése különösen fontos. A hibafa-elemzést használhatják a nem kívánt állapotok megelőzésének és javításának céljából is. Alkalmazása általában nagy kockázati tényezővel rendelkező iparágakban jellemző, például: repülőgépipar, vegyipar, gyógyszeripar. [10]

III. Saját eredmények

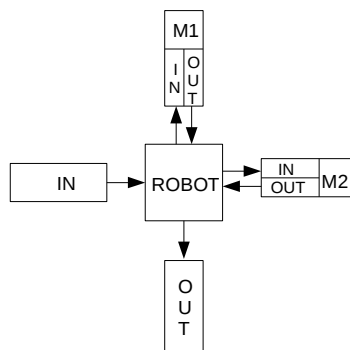
4. Gyártórendszer modellezése színezett Petri hálóval

4.1. A gyártórendszer bemutatása

Az általam vizsgált gyártórendszer egy IN belépő és egy OUT kilépő tárgyasztalt, egy M1 és egy M2 jelű gyártósort és egy robotot tartalmaz. Ezenkívül mindegyik gyártósornak van egy bemenő és egy kimenő tárgyasztala (lásd **1. ábra**). A robot feladata, hogy a belépő tárgyasztalról a feldolgozásra váró munkadarabokat a gyártósorok bemenetére rakja, a gyártósorok kimenetéről pedig vagy a másik gyártósor bemenetére, vagy a kilépő tárgyasztalra helyezze. Így összesen hat olyan manipulációs folyamat fordulhat elő, amelyhez szükség van a robotra:

1. IN belépő tárgyasztalról az M1 gyártósor bemenetére
2. IN belépő tárgyasztalról az M2 gyártósor bemenetére
3. M1 gyártósor kimenetéről az M2 gyártósor bemenetére
4. M2 gyártósor kimenetéről az M1 gyártósor bemenetére
5. M1 gyártósor kimenetéről az OUT kilépő tárgyasztalra
6. M2 gyártósor kimenetéről az OUT kilépő tárgyasztalra

Mivel mind a hat feladatot egyetlen robotnak kell ellátnia, és az egyes műveletek között lehetnek olyanok, amelyek egy időben igénylik a robot erőforrásait, ezért az egyes manipulációk között konfliktushelyzet alakulhat ki. Ilyenkor el kell dönteni, hogy a robot melyik műveletet hajtsa végre először.



1. ábra: A gyártórendszer vázlata.

Ezután azt kell meghatározni, hogy a két gyártósor között teszünk-e különbséget. A következő eseteket vizsgálhatjuk:

1. A gyártósorokat azonosnak tekintjük: Ekkor az őket összekötő manipulációra nincs szükség. A munkadaraboknak csak egy gyártósoron kell végigmenniük.
2. A gyártósorok különbözősége esetén az alábbi esetek lehetségesek:
 - a munkadaraboknak előre megadott információ alapján a meghatározott gyártósoron kell végigmenniük;
 - a munkadaraboknak mindkét gyártósoron végig kell menniük, melynek sorrendje lehet tetszőleges vagy előre meghatározott

A továbbiakban azokat az eseteket vizsgálom részletesebben, amikor a gyártósorokat

megkülönböztetjük egymástól.

4.2. A gyártórendszer színezett Petri háló modellje hibamentes működés esetén

A rendszer alacsony szintű Petri hálóját korábbi kutatásokból [11] rendelkezésemre állt (lásd **Melléklet 1. ábra**), a színezett Petri háló elkészítésekor ezt használtam kiindulópontként. A színezett Petri háló alkalmazásának fő oka az, hogy az alacsony szintű Petri hálóban a munkadarabokat nem tudtuk megkülönböztetni egymástól. Színezett Petri hálóra való áttéréssel a munkadarabok megkülönböztethetők egymástól, valamint a gyártórendszerrel is sokkal több információ nyerhető. A színezett Petri háló másik előnye, hogy a modell egyszerűbb, könnyebben kezelhető lesz.

A modellezést és a vizsgálatok egy részét a CPNTools szoftver segítségével végeztem el. A szoftvert a dániai Aarhus Egyetemen fejlesztették ki 2000-ben a színezett Petri hálók vizsgálatához, 2010-től fejlesztése Hollandiában folyik. A CPNTools-ban lehetőségünk van színezett és időzített Petri hálók létrehozására, szimulációk futtatására és a háló analízisére is. A színezett Petri hálók megadásához szükséges típusokat ívkifejezés-függvényeket, korlátfüggvényeket a Standard ML nyelv használatával adhatjuk meg. [12]

A gyártórendszer színezett Petri háló modellje a következő elemekből épül fel:

1. helyek:

- *start*;
- *IN*: a bemenő tárgyasztalt jelképezi;
- *OUT*: a kimenő tárgyasztalt jelképezi;
- *Robot*: a robot állapotát jelképezi (szabad, vagy foglalt);
- *Tin_empty*, *Tin_full*: a gyártósorok bemenetének állapotait jelképezik;
- *Tout_empty*, *Tout_full*: a gyártósorok kimenetének állapotait jelképezik;
- *atrak1*: az *IN* bemenő tárgyasztalról a gyártósorra való áthelyezés;
- *atrak2*: a gyártósorok kimenetéről az *OUT* kimenő tárgyasztalra történő áthelyezés;
- *atrak3*: az egyik gyártósor kimenetéről a másik gyártósor bemenetére történő áthelyezés;

2. átmenetek:

- *generator*: a munkadarabok „előállítását”, címkézését végzi;
- *gyartosor*: a gyártósorokat jelképezi;
- *atrak1_kezd*, *atrak1_vege*: az *IN* bemenő tárgyasztalról a gyártósorok bemenetére történő áthelyezés kezdete és vége;
- *atrak2_kezd*, *atrak2_vege*: a gyártósorok kimenetéről az *OUT* kimenő tárgyasztalra történő áthelyezés kezdete és vége;
- *atrak3_kezd*, *atrak3_vege*: az egyik gyártósor kimenetéről a másik gyártósor bemenetére történő áthelyezés kezdete és vége;

Az átrakásokat szimbolizáló átmeneteket akár össze is lehetne vonni, de azért választottam ezt a megoldást, mert így jobban látszik, hogy az egyes műveletek lefoglalják a robotot. A hálóban mindkét gyártósort ugyanazzal az átmenettel (*gyartosor*) és helyekkel (*Tin_empty*, *Tin_full*) modelleztem és a megkülönböztetéshez a

tokenek színét használtam fel. A tokenek színe *m1* vagy *m2* lehet, amik az 1-es illetve 2-es gyártósorra utalnak. Például, ha a *Tin_empty* helyen egy *m1* színű token található, akkor az azt jelenti, hogy az 1-es gyártósor bemenete üres.

Ha két átmenet egyszerre engedélyezett, és az egyik lejátszódása után a másik nem lesz engedélyezett, akkor azt mondjuk, hogy az átmenetek konfliktusban vannak egymással. Az általam létrehozott színezett Petri hálóban a következő átmenetek között konfliktushelyzetek jelentkeznek:

- *atrak1_kezd* és *atrak2_kezd* és *atrak3_kezd*;
- *atrak2_kezd* és *atrak3_kezd*;
- *atrak1_kezd* és *atrak2_kezd*.

A technológia szempontjából ez azt jelenti, hogy a robotnak egyszerre több műveletet kellene végrehajtania, ami fennakadáshoz vezet. A konfliktusok feloldásának egyik lehetséges módja, hogy az átmenetekhez prioritást rendelünk. Ekkor az az átmenet játszódik le előbb, amelynek nagyobb a prioritása. Jelen esetben ezt úgy választottam meg, hogy mindig az az átmenet játszódjon le előbb, amely a gyártási folyamatban előrébb tartó munkadarabot juttatja tovább. Így a *gyartosor*, *atrak2_kezd* és *atrak3_kezd* átmenetek prioritását a CPNTools standard prioritásait felhasználva magasra (P_HIGH) állítottam, a többit normál (P_NORMAL) értéken hagytam.

Elsőként egy olyan modellt készítettem el, amelyben a munkadarabok tetszőleges sorrendben mehetnek végig egy vagy két gyártási folyamaton. A valóságban azonban nem jellemző, hogy teljesen a gyártórendszerre bizzuk azt, hogy az milyen termékből mennyit gyárt. A munkadarabokat ezért címkével látjuk el, ami kijelöli, hogy mely folyamatokon kell végigmenniük. Ennek megfelelően én is így módosítottam a modellt. A munkadarabok a következő információkat hordozzák:

1. azonosító (*id*): a munkadarab sorszáma, amellyel azonosíthatjuk;
2. címke (*cim*): azt mutatja meg, hogy a munkadarabnak mely folyamatokon kell keresztülmennie. A címke értékei a következők lehetnek:
 - *m1*: a munkadarab csak az 1-es gyártósoron megy végig;
 - *m2*: a munkadarab csak a 2-es gyártósoron megy végig;
 - *m12*: a munkadarab először az 1-es, azután a 2-es gyártósoron megy végig;
 - *m21*: a munkadarab először a 2-es, azután az 1-es gyártósoron megy végig;
3. számláló (*a*): azt számolja, hogy hány gyártási folyamaton ment végig a munkadarab.

Az így létrehozott színezett Petri háló most már jól leírja a rendszer hibamentes működését. (lásd **Melléklet 2. ábra.**)

4.3. Technológiai rendszerben előforduló hibalehetőségek beépítése a modellbe

Az előző fejezetben bemutatott modell a gyártórendszer hibamentes működését írja le. A valóságban azonban egy gyártási folyamat során előfordulhatnak hibák is. A hibalehetőségét korábbi kutatásokból ismert módon építettem be [3][13]. Feltételeztem

egy olyan hibalehetőséget, hogy a gyártás során a munkadarab címkéje megsérül, így elveszik a további folyamatokra vonatkozó információ. A munkadarabot ekkor selejtesnek minősítjük és kivesszük a folyamatból. Mivel a hiba a gyártósorokon következhet be, ezért az azokat jelképező átmenethez egy olyan függvényt rendeltem hozzá, amelynek visszatérési értéke meghatározza, hogy az adott folyamat során történt-e hiba. A függvény egy valószínűségi változót tartalmaz, amellyel a hibák bekövetkezésének valószínűségét állítani lehet. Továbbá a *cim* változó színhalmazának elemeit kiegészítettem egy *m0* értékkel, amit akkor vesz fel a változó, ha a munkadarab címkéje megsérül. A hiba bekövetkezése után rögtön a folyamat végére ugrunk, és a hibás munkadarabokat a gyártósorról közvetlenül a folyamat végét jelentő kimenő tárgyasztalra juttatjuk. Ehhez a művelethez nincs szükség a robotra. A hibás munkadarabokat célszerű elkülönítve kezelni, ezért a kimenő tárgyasztalt két részre osztottam. A Petri hálóban a hibás munkadarabok gyűjtőhelyét egy új hely jelképezi (*OUT_hiba*), a modell többi része megfelel a hiba nélküli esetnél leírtaknak (lásd **Melléklet 3. ábra**). Látható, hogy a beépített hibalehetőségek nem növelték meg lényegesen a hibamentes működést leíró háló méretét, továbbá az így kialakított modell hordozza a hibaesemények bekövetkezési valószínűségére vonatkozó fontos információt is. Technológiai rendszerekben a hibaesemények bekövetkezési valószínűségére megfigyelésekből, mérési adatokból állhatnak rendelkezésre tapasztalati értékek.

4.4. Időzítés beépítése a színezett Petri háló modellbe

A rendszer időbeli működését eddig az események egymásutáni bekövetkezése fejezte ki, mivel minden átmenetet pillanatszerűen lejátszódónak tekintettünk. Érdekes megvizsgálni azonban, hogy mi történik akkor, ha az események tényleges lejátszódási idejét is figyelembe vesszük, hogyan befolyásolja ez a rendszer működését. Az időzítés bevezetésekor feltételeztem, hogy a vizsgált rendszerben a műveletek lejátszódási ideje ismert, előre meghatározott. Ennek megfelelően a meglévő színezett Petri hálóban az átmenetekhez diszkrét lejátszódási időt rendeltem hozzá, valamint a munkadarabokat jelképező tokeneknek időbélyegeket adtam. Így megkaptuk a vizsgált rendszer időzített színezett Petri háló modelljét (lásd **Melléklet 4. ábra**). Megjegyzem, hogy bár a definíció szerint az időbélyegek nemnegatív valós számok lehetnek, a CPN Tools-ban csak egész értékeket tudunk beállítani.

4.5. Színezés megváltoztatása

Az előzőekben bemutatott modellekben minden egyes munkadarab az azonosítója és a címkéje alapján az egész gyártási folyamat során egyértelműen azonosítható. Sok esetben – például tömegtermékek esetén – nincs szükség arra, hogy az azonos címkéjű munkadarabokat a gyártási folyamat során megkülönböztessük egymástól. Ha egyedi azonosítókat használunk, akkor a technológia során előfordulnak olyan állapotok, ahol a munkadarabok címke szerinti eloszlása azonos, és csak az azonosítóiban különböznek egymástól. Például legyen két *m1* címkéjű munkadarabunk amelyek azonosítója 1, illetve 2. Vegyük azokat az állapotokat, amikor a gyártási folyamat végén az egyik hibás, a másik pedig hibátlan. Jelen esetben ez két különböző állapotnak felel meg, attól

függően, hogy az 1-es, vagy 2-es sorszámú munkadarab sérült-e meg. Ha a munkadaraboknak nincs azonosítója, akkor ezek az állapotok nem különböztethetők meg. Célszerűnek látszott tehát a korábbi Petri háló modellek olyan változatát is elkészíteni, ahol a színezés módosításaként a munkadarabok egyéni azonosítóját elhagytam. Valamennyi eddigi Petri háló modellnek elkészítettem az azonosító nélküli változatát is. Ezek közül a **Melléklet 5. ábrán** az időzítés nélküli, hibalehetőséget tartalmazó, a **Melléklet 6. ábrán** az időzített, hibalehetőséget tartalmazó esetek azonosító nélküli változatai láthatóak.

A hibalehetőséget tartalmazó Petri háló azonosító nélküli változatával vizsgálhatjuk például azt, hogy adott darabszámú és címkézésű munkadarabok gyártása során milyen arányban keletkeznek hibátlan, illetve hibás termékek.

Azt, hogy a gyártórendszer elkészített színezett Petri háló modelljei közül mikor melyiket érdemes használni, a modellezési cél, a gyártórendszer működésének vizsgálni kívánt aspektusa dönti el.

4.6. Szimulációs vizsgálatok

A CPNTools-ban többféle szimulációs lehetőség közül választhatunk, amelyek segítségével vizsgálhatjuk a létrehozott modell működését. Lépésről lépésre történő szimuláció esetén kiválaszthatjuk, hogy az engedélyezett átmenetek közül melyik játszódjon le. A lekötéseket manuálisan, vagy automatikusan állíthatjuk be. Az automatikus szimulációnál erre nincs lehetőség, itt a program véletlenszerűen választ az engedélyezett átmenetek, illetve a lehetséges lekötések közül. A szimulációt elsősorban arra használjuk, hogy a modell létrehozása után meggyőződjünk annak helyes működéséről. Érdekes több szimulációt is lefuttatni, mert előfordulhat, hogy így találunk technológiai holtpontot a rendszerben. A vizsgált gyártórendszernél technológiai holtpont úgy alakulhat ki, hogy a gyártási folyamatokban egy vagy több munkadarab „megakad”, a kívánt gyártási folyamat nem fejeződik be.

A gyártórendszer működését a fejlesztett színezett Petri hálók szimulációjával hibamentes és hibalehetőséget tartalmazó esetekben, valamint időzítés nélküli és időzített esetben, egyedi azonosítókkal és azonosítók nélküli változatban is vizsgáltam, különböző számú és címkézésű munkadarabok gyártását feltételezve. A szimulációk során az egyik legfontosabb észrevételem az volt, hogy időzített esetben, bizonyos időbeállításoknál 4 különböző munkadarab legyártása esetén már létrejöhet technológiai holtpont a gyártórendszer működése során.

Ha 4 különböző munkadarabot helyezünk a bemenő tárgyasztalra, továbbá az átrakás ideje kisebb, vagy egyenlő a gyártósorok idejénél, akkor létrejöhet az a helyzet, hogy először két olyan munkadarab kerül a két gyártósorra, amelyeknek utána még a másikon is végig kell haladniuk. Amíg az egyik gyártósoron haladnak végig, addig a robot a gyártósorok bemenetére rakja a maradék két munkadarabot. Mikor az első munkadarabok a gyártósorok kimenetére kerülnek, akkor a rendszer holtpontra jut (lásd **Melléklet 7. ábra**), mivel mindkét gyártósor kimenete és bemenete is foglalt, feltételnek pedig azt adtam meg, hogy egy munkadarab akkor kerülhet rá a gyártósorra, ha annak üres a kimenete, valamint egy gyártósor bemenetén egyszerre csak egy munkadarab lehet. Ha egy átrakás ideje nagyobb, mint amennyi idő alatt a gyártósoron végigmegy egy munkadarab, akkor nem lesz fennakadás a rendszerben, mivel amíg az áthelyezés történik, addig a gyártási folyamat lezajlik, így nem történhet meg az, hogy amíg az egyik gyártósoron munkadarab van, addig a másik bemenetére munkadarab kerül. Ha a két gyártósor ideje különböző, akkor a holtpont kialakulásának lényegesen kisebb a valószínűsége.

Több munkadarab esetén az összes lehetséges lejátszódási sorozat már nagyon sokféle lehet. Ezek egyenkénti, lépésenkénti szimulációja igen sok ideig tartana, automatikus szimulációval pedig nem lehet garantálni, hogy az egymás utáni szimulációk különbözőek lesznek. Emiatt célszerű egy hatékonyabb, az elérhetőségi gráf elemzésén alapuló módszert alkalmazni a gyártórendszer vizsgálatára.

4.7. Vizsgálat az elérhetőségi gráf alapján a CPNTools-ban

A CPNTools-ban lehetőség van az elérhetőségi gráf előállítására. A gyártósoron bekövetkező hiba valószínűségi változóval történő megadását az elérhetőségi gráf számításánál a CPNTools nem kezeli helyesen, mert csak h egy értékére értékeli ki az

átmenet korlátfüggvényét. Így az elérhetőségi gráfban nem jelenik meg az összes lehetséges állapot. A helyes elérhetőségi gráf létrehozásához a korlátfüggvényből ki kellett vennem a valószínűségi változót, és helyette egy *bool* típusú változót alkalmaztam a helyes vagy hibás működés eldöntéséhez. Ezzel ugyan elveszítettem a hiba bekövetkezésének valószínűségére vonatkozó információt, de az összes állapot megjelenik a gráfban.

A CPNTools-ban generált elérhetőségi gráfokat több szempont szerint elemeztem. Vizsgáltam, hogy hogyan változik a gráfok mérete, szerkezete, jelennek-e meg holtpontok, miközben változtatjuk a munkadarabok számát és színét, valamint az időzítéseket.

Elsőként az időzítést nem tartalmazó színezett Petri háló elérhetőségi gráfját vizsgáltam, abban az esetben, amikor a munkadarabok rendelkeznek azonosítóval és címkével is. A hibamentes és a hibás működéshez tartozó elérhetőségi gráfok jellemzőit a **2. táblázat** tartalmazza. A hibamentes esetben a gráfban található elágazások minden esetben technológiai elágazások miatt jönnek létre. Technológiai elágazáson azt értjük, amikor a rendszer egy adott állapotából több következő állapotba juthatunk el, amelyek mindegyike a gyártórendszer hibamentes működésének megfelelő állapot. Ilyen elágazások jöhetnek létre például a gyártás kezdeténél, amikor a bemenő tárgyasztalon lévő munkadarabok közül a robot véletlenszerűen kiválasztja az első feldolgozásra kerülő munkadarabot. A hibalehetőség beépítésével azt tapasztaltam, hogy az elérhetőségi gráfok mérete jelentősen megnőtt. Ennek oka, hogy a hibaesemények bekövetkezése miatt újabb elágazások jelentek meg a gráfban a technológiai

2. táblázat: Az elérhetőségi gráfok jellemzői azonosítóval, időzítés nélküli esetben.

munka- darabok száma	címkézés (azonosító, címke)	időzítés nélkül, hiba nélkül			időzítés nélkül, hibával		
		csúcs	él	vég- pont	csúcs	él	vég- pont
1	(1,m1)	7	6	1	8	7	2
1	(1,m12)	10	9	1	12	11	3
2	(1,m1)++(2,m2)	29	32	1	42	50	4
2	(1,m1)++(2,m12)	37	40	1	60	70	6
2	(1,m2)++(2,m12)	39	41	1	62	72	6
2	(1,m12)++(2,m21)	42	44	1	82	95	9
3	(1,m1)++(2,m2)++(3,m12)	145	173	1	306	409	12
3	(1,m2)++(2,m12)++(3,m21)	165	189	1	414	536	18
3	(1,m1)++(2,m1)++(3,m1)	107	123	1	210	273	8
4	(1,m1)++(2,m2)++(3,m12)++(4,m21)	631	772	1	2063	2900	36
4	(1,m1)++(2,m1)++(3,m1)++(4,m1)	340	404	1	949	1300	16
5	(1,m2)++(2,m12)++(3,m21)++(4,m1)++(5,m2)	2208	2797	1	9813	14420	72
5	(1,m21)++(2,m12)++(3,m21)++(4,m12)++(5,m21)	3259	3839	1	23709	32939	243
5	(1,m2)++(2,m1)++(3,m2)++(4,m1)++(5,m2)	1621	2165	1	5252	8051	32
6	(1,m2)++(2,m12)++(3,m21)++(4,m1)++(5,m2)++(6,m12)	8698	10980	1	62751	93323	216
6	(1,m21)++(2,m12)++(3,m21)++(4,m12)++(5,m21)++(6,m12)	12184	14562	1	146992	208602	729

elágazásokon kívül. A hibaesemények miatt létrejövő elágazások abban különböznek a technológiai elágazásoktól, hogy ilyenkor az a technológia szempontjából kedvezőtlen állapotok is létrejönnek. Láthatjuk továbbá, hogy amíg a hibamentes működésnél minden esetben egy végpontja volt a gráfnak (hiszen a gyártás eredménye csak egyféle lehet – minden munkadarab hibátlan), addig a hibát is tartalmazó gráfokban újabb végpontok jelentek meg, melyek azokat az állapotokat jelölik, amikor egy vagy több

munkadarab hibás lett. A hibamentes és a hibát tartalmazó Petri hálókhoz tartozó elérhetőségi gráfok méretének eltérése főleg több munkadarab esetén jelentős, mivel itt többször fordulhat elő hiba, ami az elérhetőségi gráfban újabb csúcsok és élek megjelenését okozza. A gráf méretét a munkadarabok címkézése is jelentősen befolyásolja, hiszen az olyan esetekben, amikor a munkadarabnak mindkét gyártósoron végig kell mennie, nyilvánvalóan nagyobb gráfot kapunk, mintha az csak egy gyártósoron menne végig.

A színezett Petri háló elérhetőségi grájának vizsgálatával kerestem a rendszerben előforduló végpontokat, vagyis azokat az állapotokat, amelyek esetén nincs további engedélyezett átmenet, azaz a jelzőponteloszlás halott. Az elérhetőségi gráfban megjelenő végpont jelentheti azt, hogy a gyártási folyamat véget ért és minden munkadarab a kimenő tárgyasztalok valamelyikére került. Ezek a végpontok a technológia szempontjából kedvező állapotokat jelölnek. Az ettől eltérő végpontok megjelenése azt jelenti, hogy a folyamatban valahol egy vagy több munkadarab „megakadt”, és ez a gyártórendszer leállását eredményezi. Az ilyen holtpontok adódhatnak például a nem megfelelő tervezésből. Feloldásuk külső beavatkozással, vagy a tervezési paraméterek módosításával lehetséges. A gyártórendszer tervezésénél a cél az ilyen holtpontok elkerülése.

A gyártórendszer színezett Petri háló modelljének elérhetőségi gráfja alapján az időzítés nélküli esetben a végpontok mindig olyan állapotokat jelölnek, amelyek a technológia szempontjából kedvező végállapotok, tehát minden munkadarab a kimenő tárgyasztalra került (hibás vagy hibátlan). Nem fordul elő olyan végpont, ami holtpontnak felel meg, tehát a rendszer ebből a szempontból megbízhatóan működik. Hibalehetőséget nem tartalmazó elérhetőségi gráf látható a **Melléklet 8. ábrán**, 3 darab *m2*, *m12*, *m21* címkéjű munkadarab gyártását feltételezve. Ugyanezen munkadarabokra vonatkozó hibalehetőséget is tartalmazó elérhetőségi gráf látható a **Melléklet 9. ábrán**.

3. táblázat: Az elérhetőségi gráfok jellemzői azonosítóval, időzített esetben.

munka- darabok száma	címkézés (azonosító, címke)	időzített, hiba nélkül			időzített, hibával		
		csúcs	él	végpont/ holtpont	csúcs	él	végpont/ holtpont
1	(1,m1)	7	6	1 / 0	8	7	2 / 0
1	(1,m12)	10	9	1 / 0	12	11	3 / 0
2	(1,m1)++(2,m2)	23	22	2 / 0	41	40	8 / 0
2	(1,m1)++(2,m12)	29	28	2 / 0	59	58	12 / 0
2	(1,m2)++(2,m12)	29	28	2 / 0	58	57	12 / 0
2	(1,m12)++(2,m21)	35	34	2 / 0	87	86	18 / 0
3	(1,m1)++(2,m2)++(3,m12)	73	74	4 / 0	264	263	48 / 0
3	(1,m2)++(2,m12)++(3,m21)	86	85	4 / 0	392	391	72 / 0
3	(1,m1)++(2,m1)++(3,m1)	85	84	6 / 0	262	261	48 / 0
4	(1,m1)++(2,m2)++(3,m12)++(4,m21)	173	172	8 / 2	1473	1472	258 / 2
4	(1,m1)++(2,m1)++(3,m1)++(4,m1)	365	364	24 / 0	2121	2120	384 / 0
5	(1,m2)++(2,m12)++(3,m21)++(4,m1)++(5,m2)	574	573	24 / 4	9320	9375	1582 / 8
5	(1,m21)++(2,m12)++(3,m21)++(4,m12)++(5,m21)	154	153	24 / 24	12076	12075	2196 / 60
5	(1,m2)++(2,m1)++(3,m2)++(4,m1)++(5,m2)	514	513	24 / 0	5152	5247	780 / 0
6	(1,m2)++(2,m12)++(3,m21)++(4,m1)++(5,m2)++(6,m12)	1725	1732	80 / 40	71817	72336	12300 / 116
6	(1,m21)++(2,m12)++(3,m21)++(4,m12)++(5,m21)++(6,m12)	325	324	72 / 72	45541	45540	8280 / 432
6	(1,m2)++(2,m1)++(3,m2)++(4,m1)++(5,m2)++(6,m1)	2341	2412	72 / 0	39349	41076	5760 / 0

időzítések minden esetben:

atrak1,2,3_kezd, atrak1,2,3_vege

gyartosor

@+1

@+5

Az időzítés bevezetésével az elérhetőségi gráfok jellemzői is megváltoztak (lásd **3. táblázat**), mivel a jelzőpontoknak időbélyegeik is vannak. A nem időzített esetben egy adott tokeneloszláshoz akár többféle lejátszódási sorozattal is eljuthattunk. Az átmenetek időzítéseinek figyelembevételével viszont ezekhez a lejátszódási sorozatokhoz különböző tokeneloszlások fognak tartozni, mivel a tokenek időbélyegei különbözőek lesznek. Az időzítés nélkül azonosnak tekinthető tokeneloszlásokat tehát megkülönböztetjük egymástól. Emiatt az időzítés nélküli esethez képest több végpont jelenik meg a gráfban, amelyek között vannak olyanok, amelyek csak a tokenek időbélyegeiben térnek el, de ugyanazok a munkadarabok hibásak/hibátlanok. A **Melléklet 10. ábrán** látható az előbbi 3 munkadarabos eset időzített Petri hálójának elérhetőségi gráfja, akkor, ha a gyártósorok lejátszódási idejét 5 időegységnek, az $atrak\{1,2,3\}_{kezd,vege}$ átmeneteket pedig 1 időegységnek választjuk.

További eltérés az időzítés nélküli estekhez képest, hogy abban az esetben ha a gyártósorok időzítése nagyobb, mint az átrakások ideje, akkor megjelenhetnek technológiai holtpontok a rendszerben. A holtpont megjelenése a munkadarabok számától és címkézésétől is függ. Ha a munkadaraboknak csak egy gyártósoron kell végigmenniük, akkor nem alakulhat ki holtpont, bármilyen sok munkadarabunk van. A holtpont kialakulásának feltétele a címkézés és a munkadarabok száma alapján az, hogy legalább négy munkadarabunk legyen, ezek közül legalább kettő olyan, amelynek mindkét gyártósoron végig kell mennie, és a többi munkadarab között legyen két olyan, amelyeknek két különböző gyártósoron kell először végigmenniük. Például holtpont alakul ki $(m12, m21, m1, m2)$, vagy $(m12, m21, m12, m21)$ címkézéskor, de $(m12, m21, m1, m1)$ vagy $(m12, m21, m21, m21)$ címkézéskor nem. Előfordul olyan címkézés is, amely esetén a gyártórendszer hibamentes működése mellett a rendszer soha nem jut technológiailag kedvező végállapotba, hanem minden esetben holtpontra jut. Ez az eset áll fenn például akkor, ha legalább 4 munkadarabot felváltva $m12$ és $m21$ címkével

látunk el. A hibalehetőség bevezetése az időzített esetben is növeli a gráf méretét. Fontos észrevétel, hogy időzített esetben az eltérő időbélyegek miatt az elérhetőségi gráf gyakran fa szerkezetű.

Azonos címkézések mellett érdemes megvizsgálni, hogy az időzítéseket változtatva hogyan változnak az elérhetőségi gráfok jellemzői. Az időzítések hatását 3,4,5 munkadarabra, rögzített címkézés mellett vizsgáltam, az eredmények a **4. táblázatban** láthatóak. A modellben összesen 7 átmenet időzítését lehet változtatni, de a könnyebb kezelhetőség miatt az átrakásokhoz tartozó átmenetek időzítését minden esetben egyformának választottam. Az időzítések változtatásakor azt tapasztaltam, hogy elérhetőségi gráfok szerkezete bizonyos beállítások mellett fa, más beállítások esetén nem fa lett. Abban az esetben, ha az időzítési beállítások olyanok, hogy az elérhetőségi gráfban nincs két olyan csúcs, amelyeknek a tokeneloszlása és a tokenek időbélyegei is megegyezne, az elérhetőségi gráf fa lesz. Olyan időzítési beállításoknál, ahol létrejöhetnek olyan csúcsok, amelyeknél a tokeneloszláson kívül tokenek időbélyegei is megegyeznek, az elérhetőségi gráf szerkezete már nem lesz fa. Az időzítések változtatása a holtpontok megjelenésére is hatással van. Holtpontok akkor jelentek meg, ha a gyártósorok lejátszódási idejét nagyobbra állítottam, mint az átrakások ideje

4. táblázat: Időzítések hatásának vizsgálata.

munkadarabok száma	címkézés (azonosító, címke)	elérhetőségi gráf jellemzői (időzített, hibával)				időzítések
		csúcs	él	végpont/holtpont	szerkezet (fa/nem fa)	
3	(1,m2)++(2,m12)++(3,m21)	392	391	72 / 0	fa	atrak{1,2,3}_{kezd,vege}: @+1, gyartosor: @+5
3	(1,m2)++(2,m12)++(3,m21)	395	400	72 / 0	nem fa	atrak{1,2,3}_{kezd,vege}: @+2, gyartosor: @+3
4	(1,m2)++(2,m12)++(3,m21)++(4,m1)	1473	1472	258 / 2	fa	atrak{1,2,3}_{kezd,vege}: @+1, gyartosor: @+5
4	(1,m2)++(2,m12)++(3,m21)++(4,m1)	1523	1534	270 / 2	nem fa	atrak{1,2,3}_{kezd,vege}: @+2, gyartosor: @+3
4	(1,m2)++(2,m12)++(3,m21)++(4,m1)	1653	1704	288 / 0	nem fa	atrak{1,2,3}_{kezd,vege}: @+2, gyartosor: @+2
5	(1,m2)++(2,m12)++(3,m21)++(4,m1)++	9320	9375	1582 / 8	nem fa	atrak{1,2,3}_{kezd,vege}: @+1, gyartosor: @+5
5	(1,m2)++(2,m12)++(3,m21)++(4,m1)++	9592	9635	1644 / 6	nem fa	atrak{1,2,3}_{kezd,vege}: @+2, gyartosor: @+3
5	(1,m2)++(2,m12)++(3,m21)++(4,m1)++	11283	11606	2000 / 0	nem fa	atrak{1,2,3}_{kezd,vege}: @+2, gyartosor: @+2

Ezután azt vizsgáltam, hogy ha a munkadaraboknak nincs azonosítója, csak címkéje, akkor milyen változások történnek az elérhetőségi gráfban. Legszembetűnőbb változás az elérhetőségi gráf méretének drasztikus csökkenése, főleg több munkadarab esetén. (lásd **5. és 6. táblázat**). Ennek oka, hogy az azonos címkéjű munkadarabokat most nem különböztetjük meg azonosítóval, így azok az állapotok, amelyeket az azonosító miatt különbözőnek tekintettünk, azonosító nélkül összevonhatóak. Másrészt a technológiai elágazásoknál az adott állapotból elérhető új állapotok száma lényegesen kevesebb, mert azt nem a munkadarabok száma, hanem a különböző címkézések száma határozza meg.

5. táblázat: Elérhetőségi gráfok jellemzői, azonosító nélkül, időzítés nélkül.

munka- darabok száma	címkézés	időzítés nélkül, hiba nélkül			időzítés nélkül, hibával		
		csúcs	él	végpont	csúcs	él	végpont
1	m1	7	6	1	8	7	2
1	m12	10	9	1	12	11	3
2	m1++m2	29	32	1	42	50	4
2	m1++m12	37	40	1	60	70	6
2	m2++m12	39	41	1	62	72	6
2	m12++m21	42	44	1	81	94	8
3	m1++m2++m12	145	173	1	306	409	12
3	m2++m12++m21	165	189	1	407	530	16
3	3`m1	27	28	1	49	57	4
4	m1++m2++m12++m21	631	772	1	2023	2852	32
4	4`m1	37	39	1	83	100	5
5	2`m2++m12++m21++m1	1283	1604	1	5431	7890	48
5	3`m21++2`m12	440	504	1	2573	3492	42
5	3`m2++2`m1	230	290	1	693	1001	12
6	2`m2++2`m12++m21++m1	2870	3588	1	18709	27518	90
6	3`m21++3`m12	759	878	1	6347	8782	64

6. táblázat: Elérhetőségi gráfok jellemzői, azonosító nélkül, időzítéssel.

munka- darabok száma	címkézés	időzített, hiba nélkül			időzített, hibával		
		csúcs	él	végpont/ holtpont	csúcs	él	végpont/ holtpont
1	m1	7	6	1 / 0	8	7	2 / 0
1	m12	10	9	1 / 0	12	11	3 / 0
2	m1++m2	23	22	2 / 0	40	40	7 / 0
2	m1++m12	29	28	2 / 0	58	58	11 / 0
2	m2++m12	29	28	2 / 0	57	57	11 / 0
2	m12++m21	35	34	2 / 0	83	86	14 / 0
3	m1++m2++m12	73	74	4 / 0	256	259	44 / 0
3	m2++m12++m21	86	85	4 / 0	376	379	66 / 0
3	3`m1	19	18	1 / 0	49	48	8 / 0
4	m1++m2++m12++m21	173	172	8 / 2	1287	1312	204 / 2
4	4`m1	25	24	1 / 0	102	101	16 / 0
5	2`m2++m12++m21++m1	302	301	12 / 2	3863	3954	603 / 4
5	3`m21++2`m12	26	25	2 / 2	1025	1024	183 / 5
5	3`m2++2`m1	56	55	2 / 0	448	455	65 / 0
6	2`m2++2`m12++m21++m1	461	462	20 / 10	13142	13446	2082 / 29
6	3`m21++3`m12	27	26	2 / 2	1269	1292	206 / 12
6	3`m2++3`m1	83	84	2 / 0	1118	1168	157 / 0

Időzítések minden esetben: atrak{1,2,3}_ {kezd,vege}: @+1
 gyartisor @+5

Érdemes az azonosítót tartalmazó és azonosítót nem tartalmazó modellek elérhetőségi gráfját összehasonlítani 3 darab *m1* címkéjű munkadarab esetén (lásd **Melléklet 11.** és **12. ábra**) Ha a munkadarabok egyedi azonosítóval rendelkeznek, akkor az elérhetőségi gráfban technológiai elágazás található például a 4. csúcsnál. Az ennek megfelelő állapotban a robot véletlenszerűen választ a három munkadarab közül, azaz a 4-es csúcsnak három kimenő éle van az elérhetőségi gráfon. Ezzel szemben ha a munkadaraboknak nincs egyedi azonosítójuk, akkor az azonos *m1* címkék miatt egyáltalán nem lesznek technológiai elágazások az elérhetőségi gráfon. Így ennek mérete lényegesen kisebb.

A 4. és 5. táblázat adatait összevetve az 1. és 2. táblázat adataival, megfigyelhetjük, hogy kis számú (2-4) különböző címkéjű munkadarab esetén a két esetben a gráfok mérete nem tér el jelentősen. Az időzítés nélküli, hibamentes esetben a gráfok megegyeznek, mivel ekkor a címke egyértelműen azonosítja a munkadarabot. A hibát is tartalmazó, nem időzített esetében az azonosító nélküli gráf csúcsainak száma valamivel kisebb lesz, mert a címkék csak addig különböztetik meg a munkadarabokat, amíg nem történt hiba. Ha a gyártósoron hiba következik be, akkor a hibás munkadarabok egyforma *m0* címkét kapnak, és ezután már nem különböztethetők meg. Ha azonosítójuk is van, akkor a hibás munkadarabok is megkülönböztethetők, mivel a hiba bekövetkezése az azonosítót nem befolyásolja. Összefoglalva megállapítható, hogy a színezés kialakítása (egyedi azonosító alkalmazása vagy nem alkalmazása) jelentősen befolyásolja az elérhetőségi gráf méretét.

4.8. Rendszerállapotok bekövetkezési valószínűségeinek számítása az elérhetőségi gráf élsúlyozásával

A modellezés során a CPNTools-ban sikeresen beépítettem a gyártórendszer hibalehetőségeit és azok bekövetkezési valószínűségeit a modellbe, de ez utóbbi információ az elérhetőségi gráfok generálásakor elveszik. A gyártórendszer vizsgálatakor azonban nagyon hasznos információ, hogy a hibák mekkora valószínűséggel következnek be, mert ezek hatással vannak a rendszerben létrejövő állapotok bekövetkezési valószínűségeire is. Mivel az elérhetőségi gráf a rendszer működése során létrejövő összes lehetséges állapotot tartalmazza, ezért célszerű ezt felhasználni a bekövetkezési valószínűségek számításához. Feltételezzük, hogy a gyártórendszer adott indulóállapotából generált elérhetőségi gráfja véges és nem tartalmaz körutat. Az általam elkészített Petri hálók elérhetőségi gráfjára ezek a feltételek teljesülnek. A továbbiakban ismertetem az elérhetőségi gráf élsúlyozásán alapuló módszert, amellyel meghatározható a rendszer összes állapotának bekövetkezési valószínűsége.

Első lépésben az elérhetőségi gráf élsúlyozását végezzük el. Ekkor minden élhez hozzárendeljük azt a valószínűséget, amellyel az átmenet lejátszódhat az élnak megfelelő lekötésben. A súlyokat az alábbi szabályok alapján határozzuk meg:

1. Ha az elérhetőségi gráfban egy csúcsnak egy kimenő éle van, akkor ez azt jelenti, hogy a csúcsnak megfelelő állapotból egyértelműen következik a következő állapot. Ennek megfelelően az élsúly ebben az esetben 1.
2. Ha a csúcshoz több kimenő él tartozik, akkor ez azt jelenti, hogy a rendszer ebből az állapotból több különböző állapotba juthat el. Ekkor a valószínűségek a hibamentes és a hibát is tartalmazó esetben a következőképpen számíthatók:
 - a) A hibamentes működést leíró Petri háló elérhetőségi gráfjában az elágazások minden esetben technológiai elágazások miatt jönnek létre. Például, ha a bemenő tárgyasztalon több munkadarab van, akkor ezek közül a robot véletlenszerűen választja ki a következő feldolgozandó munkadarabot. A technológiai elágazások egymást kölcsönösen kizáró eseményeknek felelnek meg, és minden eseménynek megadható a bekövetkezési valószínűsége, melyek összege 1 (azaz az események teljes eseményrendszert alkotnak). Általában feltételezzük, hogy minden esemény egyforma valószínűséggel következhet be. Ekkor az élsúly a kimenő élek számának reciprokával egyezik meg.
 - b) A hibalehetőség beépítésével a technológiai elágazásokon kívül új, hibaeseményekhez kapcsolódó elágazások jelennek meg az elérhetőségi gráfban. Ezeket az elágazásokat a jelzőpont-eloszlások, illetve lekötéselemek segítségével azonosítjuk és elkülönítjük a technológiai elágazásoktól. Tegyük fel, hogy egy t átmenet lejátszódása során a normál lejátszódás mellett k -féle hibás lejátszódás is lehetséges. Ekkor az elérhetőségi gráf megfelelő csúcsához $k+1$ kilépő él tartozik, melyek közül egy a hibátlan működés, k db a hibás működés miatt jön létre. Jelölje a hibaeseményeket $f_1, f_2, \dots, f_k$, és a bekövetkezési valószínűségeiket $P_1, P_2, \dots, P_k$. Ekkor a hibamentes működéshez tartozó él súlya $1-(P_1+P_2+\dots+P_k)$, az f_i hibaeseményhez tartozó él súlya P_i , ($i=1, \dots, k$).

Az élsúlyozás elvégzése után meghatározhatjuk a rendszerállapotok bekövetkezési valószínűségét. Ha feltételezzük, hogy az esetlegesen egymás után bekövetkező hibaesemények független események, akkor egy adott állapot bekövetkezési valószínűsége a következő módon számítható ki:

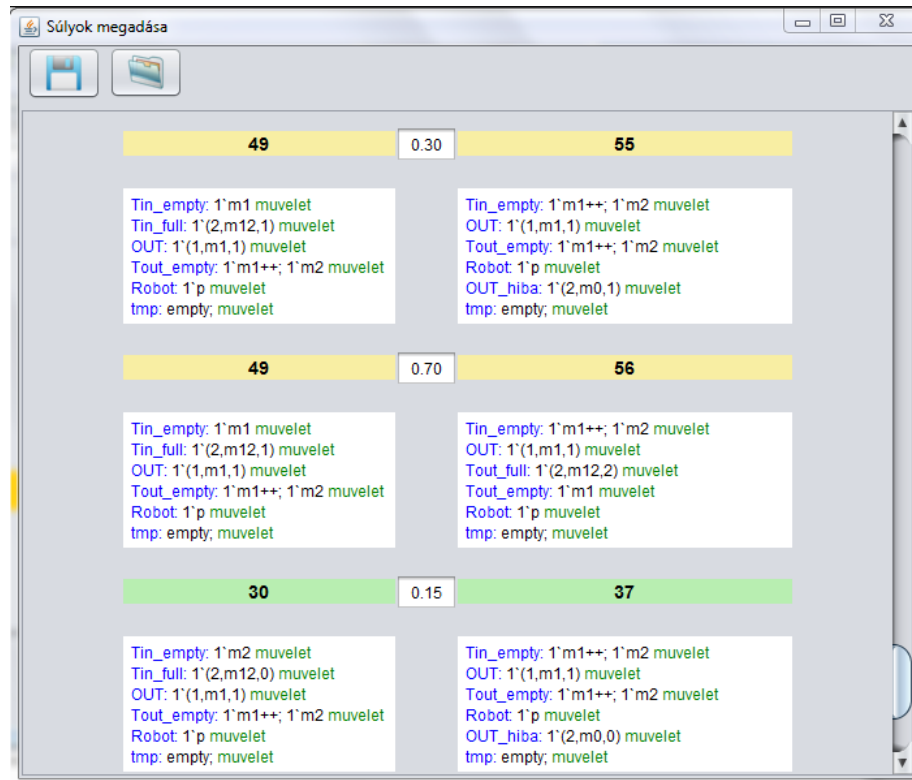
1. Ha az elérhetőségi gráf fa, akkor az adott állapot bekövetkezési valószínűsége megegyezik a kiinduló csúcsból az adott állapotnak megfelelő csúcsba vezető út mentén vett élsúlyok szorzatával.
2. Ha az elérhetőségi gráf nem fa, akkor a kiinduló csúcsból egy adott csúcsba több út vezet. Ekkor az adott állapot bekövetkezési valószínűsége megegyezik a kiinduló csúcsból az adott állapotnak megfelelő csúcsba vezető utak mentén vett élsúlyok szorzatának összegével.

Ha a hibaesemények nem függetlenek, akkor a valószínűségek hasonlóan számíthatóak, de feltételes valószínűségeket alkalmazva.

A módszer alkalmazását egy egyszerű példán keresztül mutatom be. Az élsúlyozáshoz és a rendszerállapotok bekövetkezési valószínűségeinek számításához a kutatócsoportban kifejlesztett OGAnalyzer szoftvert használok [14].

Az OGAnalyzer először megjeleníti a CPNTools-ban generált elérhetőségi gráfokat. Ezután elvégezhetjük a gráf élsúlyozását. Az OGAnalyzer a technológiai elágazásokhoz automatikusan hozzárendeli az élsúlyokat, így a továbbiakban már csak a hibaesemények miatt létrejövő elágazásoknál kell megadni az élsúlyozást. A gyártórendszerre kifejlesztett modelljeinkben a gyártósorokon egyféle hibaesemény következhet be (megsérül a munkadarab címkéje), így az elérhetőségi gráfon a hibaeseményekhez köthető elágazásoknál mindig két kimenő él található, annak megfelelően, hogy történt, vagy nem történt hiba. A jelzőponteloszlások ismeretében azonosítani tudjuk, hogy a hiba melyik gyártósoron következett be, és az ennek megfelelő súlyt rendeljük hozzá az élhez (lásd **2. ábra**). Az OGAnalyzer ezután megjeleníti az élsúlyokat (az 1-es élsúlyokat a szoftver nem írja ki), majd a következő lépésben kiszámítja a csúcsokhoz tartozó valószínűségeket. Az OGAnalyzer a csúcsokon századokra kerekítve jeleníti meg a számolt értékeket, emiatt lehet eltérés a pontos bekövetkezési valószínűség és a megjelenített érték között, illetve nagyon kis ($<0,01$) valószínűség esetén 0,00 jelenik meg a gráfban.

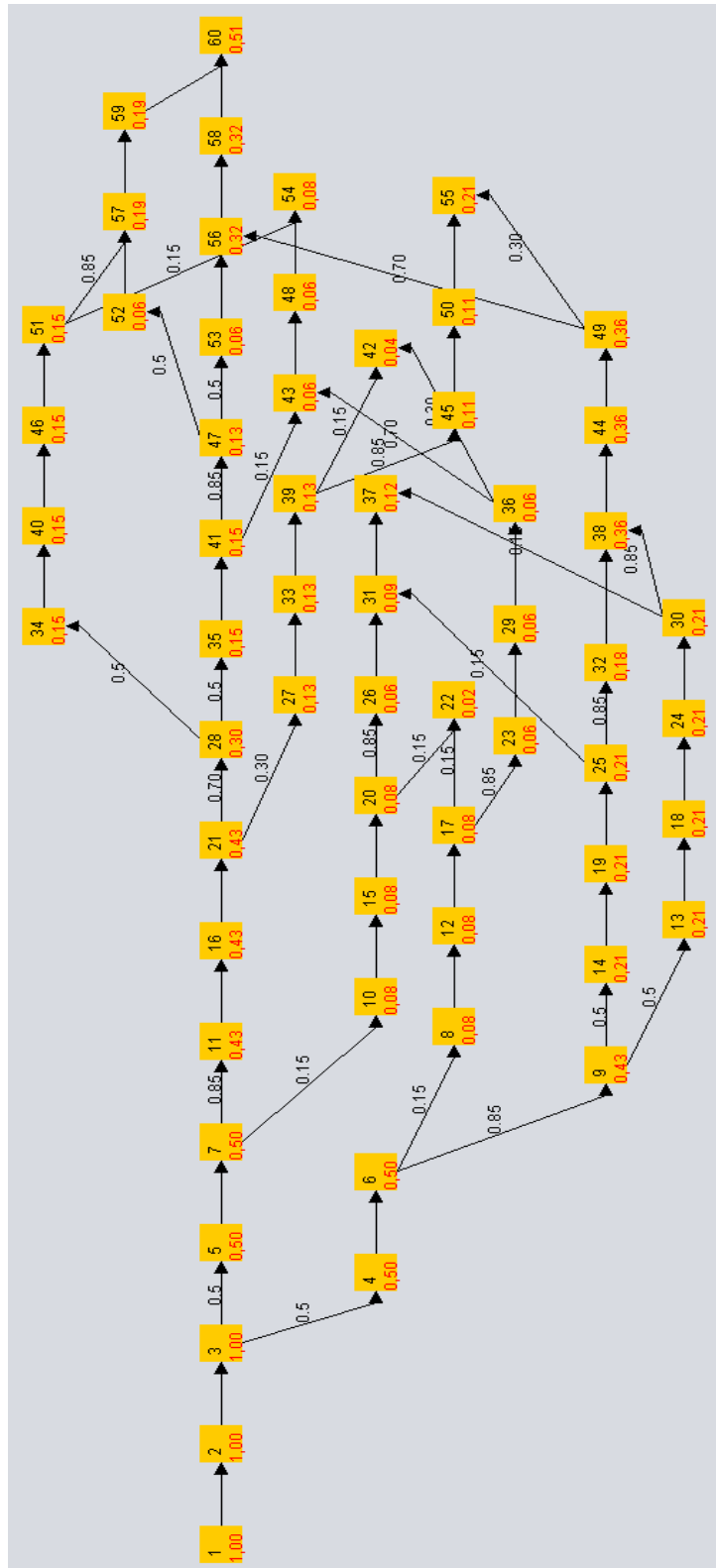
Legyen 2 munkadarabunk, amelyek címkézése $m1$, illetve $m12$, tehát az első munkadarabnak az első, a másodiknak először az első, majd a második gyártósoron kell végigmennie. A munkadarabok rendelkeznek azonosítóval is. Legyen a hiba bekövetkezési valószínűsége az első gyártósoron 0,15 a másodikon 0,3.



2. ábra: Élsúlyok megadása az OGAnalyzer-ben.

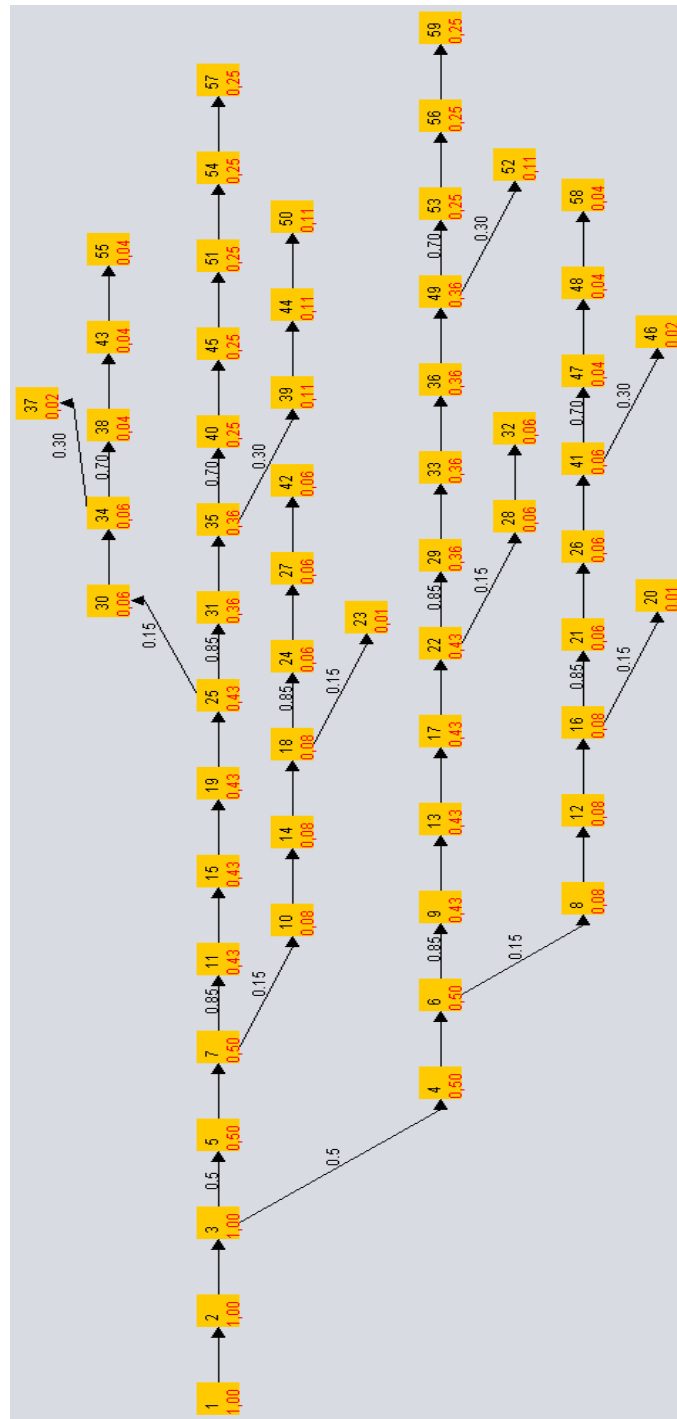
Az időzítés nélküli Petri hálózathoz tartozó súlyozott elérhetőségi gráf a számolt valószínűségekkel a **3. ábrán** látható. A gráfon például a 3-as csúcsnál technológiai elágazás van (a robot véletlenszerűen választ a két munkadarab közül), ezért itt a kimenő élek súlyai 0,5 és 0,5. A 7-es csúcsnál az *m12* címkéjű munkadarab került az 1-es gyártósorra, ahol a meghibásodás valószínűsége 0,15, így a kimenő élek súlyai 0,85 és 0,15.

Mivel a gráf szerkezete nem fa, így a csúcsokhoz tartozó valószínűségek kiszámításakor több utat is figyelembe kell venni. Vegyük például azt az állapotot, amikor a gyártás során nem keletkezett hibás munkadarab. Ennek az állapotnak a 60-as sorszámú csúcs felel meg, melynek az OGAnalyzer által számolt bekövetkezési valószínűsége 0,51. A 22-es és 42-es csúcsok azokat az állapotokat jelölik, amikor a gyártási folyamat eredménye két hibás munkadarab. Ezeknek a bekövetkezési valószínűsége 0,02 illetve 0,04 tehát annak a valószínűsége, hogy mindkét munkadarab gyártása során hiba történik 0,06.



3. ábra: A súlyozott elérhetőségi gráf a bekövetkezési valószínűségekkel, időzítés nélküli esetben.

Az időzítést tartalmazó színezett Petri háló elérhetőségi gráfja jelen beállítások mellett fa szerkezetű lesz. Ekkor minden állapot bekövetkezési valószínűségének számításakor csak egy utat kell figyelembe venni. Az időzített esethez tartozó súlyozott elérhetőségi gráf a számolt valószínűségekkel a **4. ábrán** látható. Az időbélyegek miatt két olyan csúcs van (az 57-es ill. 59-es sorszámú), amely annak az állapotnak felel meg, amikor nem történt hiba a gyártósorokon. Mindkét állapot bekövetkezési valószínűsége 0,25, így annak a valószínűsége, hogy a gyártási folyamat eredményeként mindkét munkadarabunk hibátlan lesz 0,5.



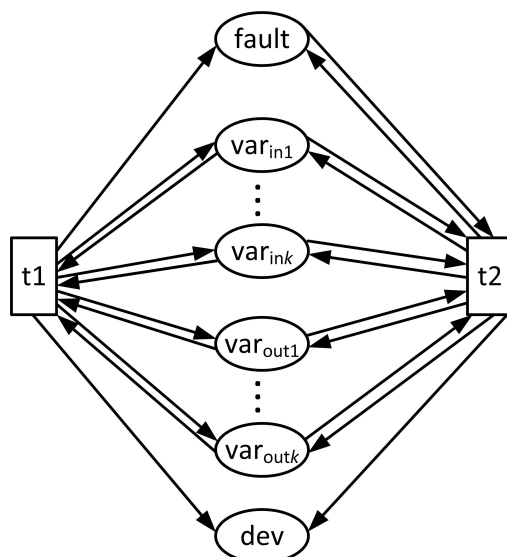
4. ábra: A súlyozott elérhetőségi gráf a bekövetkezési valószínűségekkel, időzített esetben.

5. Technológiai folyamatok kvalitatív modell alapú diagnosztikája színezett Petri háló használatával

5.1. Eseménysorok reprezentációja színezett Petri hálóval

A színezett Petri háló felépítése

Diszkrét eseményű rendszerek működése sok esetben leírható a ki- és bemenő változók aktuális értékeit tartalmazó eseménysorokkal. Ahogy ezt a 3.1 fejezetben bemutattam, megadhatjuk a működés leírását normál (hibamentes) esetben és különböző hibák fellépése esetén is. A színezett Petri háló elkészítése során is ezt a megközelítést alkalmaztam. Ennek alapján a bemenő és a kimenő változóknak helyek feleltethetők meg. A változók által felvehető kvalitatív értékeket az ezekhez a helyekhez kapcsolódó színhalmazok tartalmazzák. A színhalmazokat kvalitatív értékkészletként definiáltam. Ezen kívül, mivel egy eseményben a változók értékein kívül az esemény bekövetkezésének időpontja is szerepel, ezért időzített színhalmazok használata szükséges. Így egy változó aktuális értékét az adott időpontban a változónak megfelelő helyen lévő jelzőpont színe adja meg. A diagnosztikai célú modell elkészítéséhez a be- és kimenő változókon kívül egy-egy hely szükséges a hiba, valamint az eltérések tárolására is (lásd 5. ábra). A hibának megfelelő hely (*fault*) a bekövetkezett hibát tartalmazza, pl. a hiba típusát leíró sztring formájában. Az eltéréshez tartozó helyen (*dev*) a szimuláció során keletkezett eltéréseket gyűjtjük össze, a 3.1 fejezetben ismertetett *kulcsszó(nominális esemény_i)* formátumban. Így egy szimuláció futtatása után ezen a helyen az összes keletkezett eltérés szerepelni fog. A technológiai folyamat modellezésére két átmenetet használtam. Az első (*t1*) átmenet feladata a (normál vagy hibás) működési mód generálása, és a kapott működési módnak megfelelően az 1. időponthoz tartozó értékek hozzárendelése a helyekhez. A második (*t2*) átmenet feladata a változók értékének frissítése az időzítéseknek és a működési módnak megfelelően, valamint hibás működés esetén az eltérések hozzárendelése az erre szolgáló helyhez.



5. ábra: Egy technológiai egység általános színezett Petri háló modellje.

A színezett Petri háló működése

Kezdetben a bemenő és kimenő változóknak megfelelő helyen egy-egy inicializáló token van, melyek időbélyege 0. Ekkor a *t1* átmenet engedélyezett. Az átmenetbe beépített *fault* függvény véletlenszerűen generálja a normál vagy különböző hibás működési módokat. Az átmenet tüzelése után a *fault* helyen megjelenik egy token, a *fault* függvény által visszaadott értékkel. Ha a rendszer hibamentesen működik, akkor a *fault* helyen egy *norm* színnel rendelkező token jelenik meg, egyébként pedig a hibára vonatkozó sztring (pl. *leak*, *pos_bias*). A többi változó lekötése szintén az átmenetben történik meg, a kapott működési módnak megfelelően. A *t1* átmenet tüzelése tehát meghatározza a hibamentes vagy hibás működést, valamint ennek megfelelően beállítja a változók 1. időponthoz tartozó értékeit. Ezután a *t2* átmenet válik engedélyezetté, és annyiszor tüzel, ahány esemény szükséges az adott technológiai folyamat lejátszódásához. A *t2* átmenet minden lejátszódásakor felhasználja a *fault* helyen lévő token, és ennek megfelelően frissíti a változók értékeit. Az átmenet lejátszódási idővel is rendelkezik, így a tokenek időbélyegei minden lejátszódás után változnak. Hibás működés esetén az átmenet a karakterisztikus és a nominális trace között az aktuális lépésben létrejövő egy vagy több eltérést is hozzárendeli a *dev* helyhez.

5.2. Az elérhetőségi gráfon alapuló diagnosztikai módszer

Ebben a részben azt mutatom be, hogy hogyan lehet az eseménysorokat modellező színezett Petri háló elérhetőségi gráfját felhasználni a technológiai rendszer diagnosztikai célú vizsgálatához.

A technológiai egység előzőekben bemutatott modellje tartalmazza a normál működés mellett a különböző hibás működési módokat is. Az elérhetőségi gráfja egy adott kezdőállapotból kiindulva a rendszer működése során bekövetkező összes lehetséges állapotot tartalmazza. Ebből következik, hogy a technológiai rendszer előzőekben ismertetett felépítésű színezett Petri háló modelljének elérhetőségi grájában a normál és a hibás működésekhez tartozó állapotok egyaránt szerepelnek. A gráfban az induló állapot után biztosan lesz egy elágazás a *t1* átmenetben szereplő *fault* függvény kiértékelése miatt (a különböző működési módokra a függvény különböző értékeket ad). Mivel a hibát a *fault* helyen lévő token színe tartalmazza, és ez a szimuláció során végig változatlan marad, emiatt a különböző működési módokhoz tartozó állapotok egyértelműen megkülönböztethetők egymástól. Például, ha két különböző működés során előfordul olyan állapot, amikor a változók értékei és az esetleges eltérések is megegyeznek, az állapotok a hiba színe miatt akkor is különbözni fognak. Ez azt jelenti, hogy az elérhetőségi gráf fa szerkezetű, és az egyes működési módoknak különböző utak fognak megfelelni.

Miután létrehoztuk a vizsgálandó rendszer színezett Petri háló modelljét, célszerű szimulációval meggyőződhetünk a működés helyességéről. Ha a modell megfelelő, akkor előállíthatjuk az elérhetőségi gráfot, adott indulóállapotot felvéve. Feltételezzük, hogy csak előre ismert hibák következhetnek be, amelyeket beépítettünk a modellbe és ezeknek ismerjük ezen hibák hatását a technológiai rendszer működésére. Továbbá feltételezzük, hogy a hibák közül egyszerre csak egy következhet be, valamint a hiba a vizsgálat kezdetekor fennáll és nem változik a folyamat lejátszódása során. Ebben az esetben az elérhetőségi gráf minden olyan állapotot tartalmaz, ami a kezdőállapotból

elérhető. A gráf végpontjai megfelelnek a technológiai rendszer végállapotainak. A végállapotokban a *dev* helyen lévő minden egyes token egy eltérésnek felel meg. Tehát minden végponthoz tartozik egy, az adott működésre jellemző eltérés lista. Tegyük fel, hogy rendelkezésünkre áll a technológiai rendszer működéséből származó teljes karakterisztikus trace, és az eltérések listáját is ismerjük, amit a nominális trace-szel való összehasonlításból kaptunk. Ekkor a hiba meghatározásához ezt az eltérés listát kell összehasonlítani a gráf végpontjainak *dev* helyén szereplő tokenek listájával. Ha a két lista megegyezik, akkor a bekövetkezett hiba kiolvasható a *fault* helyen lévő token színéből. Ha az elérhetőségi gráf végpontjai között több olyan is van, amelyeknél az eltérések listája megegyezik a technológiai rendszerből származó karakterisztikus trace alapján meghatározott eltérés listával, akkor a fennálló hiba nem azonosítható egyértelműen, hanem csak a lehetséges hibakok halmazát tudjuk megadni. Ilyen esetben a hiba azonosításához további vizsgálat szükséges.

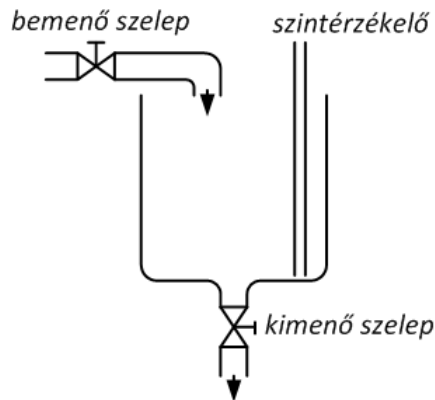
Most tegyük fel, hogy nem ismerjük a teljes karakterisztikus trace-t, például azért, mert a technológiai folyamat még nem fejeződött be. Ekkor csak egy adott időpontig ismerjük a bekövetkezett eseményeket és az eltéréseket. Ebben az esetben ezeket az eltéréseket azoknak a csúcsoknak az eltéréseivel kell összehasonlítani, amelyeknek az időbélyege megegyezik az adott időponttal. Ha csak egy olyan csúcsot találunk, ahol az eltérések megegyeznek, akkor a hibát egyértelműen meghatároztuk. Ha több csúcsot is találunk, amelyekre teljesül az egyezés, akkor az ezekhez tartozó hibák közül bármelyik bekövetkezhetett. Ebben az esetben nem dönthető el egyértelműen a ténylegesen bekövetkezett hiba, viszont az összes hiba halmazát leszűkítettük a lehetséges esetekre.

5.3. Egyszerű technológiai rendszer diagnosztikai vizsgálata

Az általam vizsgált technológiai rendszer több hasonló tartályból épül fel, melyek különböző módon kapcsolódhatnak egymáshoz. A vizsgált technológiai folyamat a tartályok valamilyen folyadékkal való feltöltése.

Egy tartály feltöltési folyamatának modellje

Először egy tartály feltöltési folyamatának modelljét készítettem el, ezt később felhasználtam a különböző kapcsolások kialakításakor. A tartály egy töltő és egy ürítő vezetékekkel rendelkezik, és az ezeken áthaladó folyadékáram egy-egy kétállású szeleppel (pl. mágnesszeleppel) szabályozható. A tartályban egy szintérzékelő is található, mely az operátor felé jelzi a folyadék szintjét (lásd **6. ábra**).



6. ábra: A vizsgált technológiai egység.

A technológiai folyamat a következő lépésekből áll:

0. kezdeti állapot, a szelepek zárva vannak és a tartály üres
1. kinyitjuk a bemenő szelepet
2. két periódus alatt a tartály feltöltődik
3. kinyitjuk a kimenő szelepet, ezzel a feltöltési folyamat befejeződik

A rendszerben a következő hibák valamelyikének előfordulását feltételezzük:

- *leak* – a tartály kilyukadása, melynek hatására a tartályban a szint nulla marad a feltöltés során
- *neg_bias* – a szintmérő szenzor negatív irányban egy kvalitatív egységet téved
- *pos_bias* – a szintmérő szenzor pozitív irányban egy kvalitatív egységet téved

A modellezés és a diagnosztikai vizsgálat során feltesszük, hogy a hiba a technológiai folyamat indításakor már fennáll, valamint az előre ismert lehetséges hibák közül egyszerre csak egy következik be.

A modellezési folyamat első lépéseként meghatároztam a bemenő és kimenő változókat. Példánkban a bemenő változók a szelepek állásai lesznek, a kimenő változó pedig a szenzor által mért folyadékszint. A bemenő és kimenő változókhoz a következő kvalitatív értékhalmozokat határoztam meg:

- $Q_{sz} = \{opn, cl\}$ a szelepek lehetséges állása: *opn* – nyitott, *cl* – zárt
- $Q_e = \{e0, null, L, N, H, e1\}$ a szint lehetséges értékei: *e0* – a folyadékszint az alsó méréshatár alatt van, *null* – üres, *L* – alacsony, *N* – normál, *H* – magas, *e1* – a szint meghaladja a felső méréshatárt

A színezett Petri hálóban ezeknek a változóknak egy-egy helyet (*in*, *out*, *level*) feleltettem meg. A helyekhez tartozó színhalmazok a változók lehetséges értékeit tartalmazzák, kvalitatív értékkészletet feltételezve. A technológiai rendszer vizsgálata során a változók értéke mellett az is fontos, hogy ezt az értéket mely időpontban veszi fel, ezért az előbbi színhalmazok egyben időzítettek is. Így minden token a szín mellett egy időbélyeggel is rendelkezik. A **7. táblázatban** található az egyes működési módokhoz tartozó események. A táblázat egy sorának eseményeiből álló sorozat az adott működést leíró trace-nek felel meg.

7. táblázat: Különböző működési módokhoz tartozó trace-ek.

Működési módok	trace		
<i>norm</i>	<i>(1, opn,cl,null)</i>	<i>(2,opn,cl,L)</i>	<i>(3,opn,opn,N)</i>
<i>leak</i>	<i>(1,opn,cl,null)</i>	<i>(2,opn,cl,null)</i>	<i>(3,opn,opn,null)</i>
<i>neg_bias</i>	<i>(1,opn,cl,e0)</i>	<i>(2,opn,cl,null)</i>	<i>(3,opn,opn,L)</i>
<i>pos_bias</i>	<i>(1,opn,cl,L)</i>	<i>(2,opn,cl,N)</i>	<i>(3,opn,opn,H)</i>

Kezdő állapotban a szelepek zárva vannak, a tartály pedig üres. Az aktuális normál vagy hibás működési módot a *t1* átmenetbe beépített *fault* függvény állítja elő, a 4 lehetséges érték közül véletlenszerűen választva egyet. A kapott működési mód tárolására egy új helyet is létrehoztam, *fault* névvel. Erre azért van szükség, hogy a folyamat lejátszódása során minden lépésben tudjuk, hogy mi az aktuális mód, ami szerint működik a rendszer. A *t1* átmenet tüzelése tehát előállítja az aktuális működési módot, melyet egy token színében sztring formában tárolunk a *fault* helyen. Ezentúl a *fault* függvény eredményének megfelelően hozzárendeli az *in*, *out*, *level* helyekhez az 1. időponthoz tartozó értékeket. Ezután a *t2* átmenet engedélyezetté válik, és addig tüzelőképes is marad, amíg a technológiai folyamat véget nem ér. A *t2* átmenet tüzelése minden lépésben frissíti a változók értékeit. A változók értékeinek beállítása az adott működési módnak megfelelő trace felhasználásával történik. Feltételezzük, hogy a különböző ismert működési módokhoz tartozó trace-ek rendelkezésre állnak. Ekkor a hiba típusa alapján kiválasztjuk a hozzá tartozó trace-t, és egy adott időpontban a trace megfelelő eseményéből kiolvassuk a változók aktuális értékeit. Mindkét átmenet lejátszódási ideje 1 időegység.

Az alkalmazni kívánt diagnosztikai eljáráshoz a hibás működések esetén szükség van az eltérések előállítására is. Az eltérések tárolására létrehoztam egy új helyet *dev* névvel. A hely színhalmazában *kulcsszó(nominális esemény)* típusú értékek szerepelhetnek, ahol a kulcsszó a *GRE* (*greater*), *SML* (*smaller*), *LAT* (*later*), *EAR* (*earlier*), *NH* (*never happened*) valamelyike lehet, a normál esemény pedig (*idő*, *in*, *out*, *level*) formátumban adott. Az eltérések előállítására két módszert mutatok be.

Amennyiben rendelkezésre áll a rendszer P-HAZID táblázata, akkor tudjuk azt, hogy az egyes hibák bekövetkezései milyen eltéréseket okoznak a normál működéshez képest. Ekkor minden lépésben a P-HAZID táblázat alapján hozzárendelhetjük a megfelelő eltérést a *dev* helyhez. Ennél a megoldásnál minden lépésben egyesével kell az eltéréseket az adott időponthoz rendelni, ami nagyobb rendszerek esetében nem hatékony.

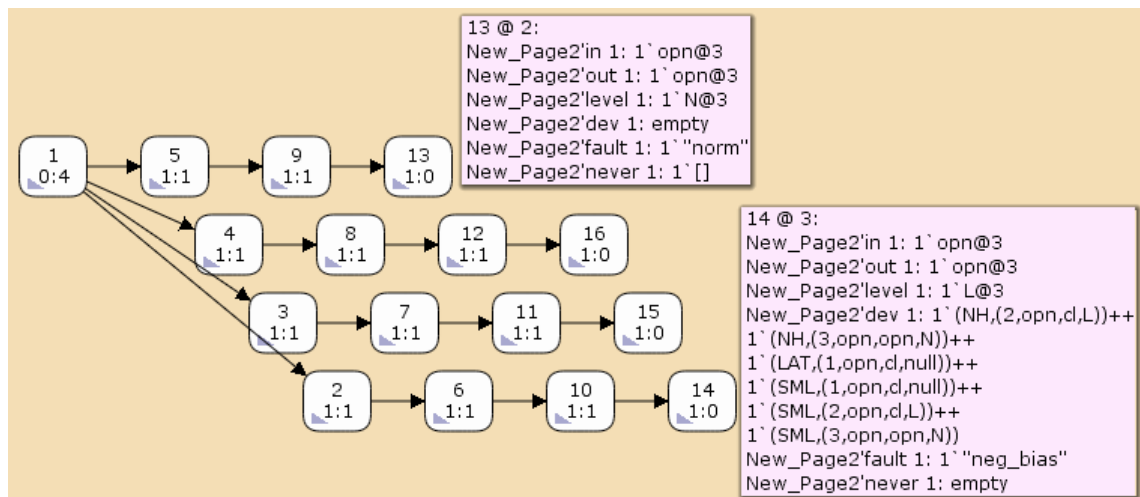
A másik lehetőség az eltérések előállítására a rendszer aktuális állapotának és a normál trace-nek az összehasonlításával történik. Az eltérések generálása több lépésből áll. A mennyiségi eltérések a normál trace adott időponthoz tartozó eseményével való összehasonlításból határozhatók meg. Az összehasonlítás során egyedül a kimenet (vagyis a szint) térhet el, az esemény többi paraméterének meg kell egyeznie. Erre a feladatra létrehoztam egy függvényt *quan* névvel. Az időbeli eltérések közül az *earlier* és *later* típusúak meghatározásához az aktuális eseményt a normál trace összes eseményével össze kell hasonlítani. Ebben az esetben csak az időbélyeg lehet különböző, az esemény többi paraméterének meg kell egyeznie. Ezt a feladatot *chron* függvény végzi el. A *never happened* típusú eltéréseket külön kell kezelni, mivel ezek

ahol a lehetséges értékek jelentése: *l0* – a tartály üres, *l10* – a szint a teljes tartomány 10%-a, *l20* – a szint 20%, ..., *l100* – a szint 100%, *e0* és *e1* továbbra is az alsó, illetve felső méréshatáron kívüli értékeket jelenti.

A feltöltési folyamat diagnosztikai vizsgálata az elérhetőségi gráf használatával

A feltöltési folyamatot leíró Petri háló elérhetőségi gráfja a **8.ábrán** látható. A gráf az időbélyegek és a hiba típusának tárolása miatt fa struktúrával rendelkezik. A fa négy ága a négy különböző (egy normál és három hibás) működési módnak felel meg. A *never happened* típusú eltérések generálása miatt a hibás működésekhez tartozó ágak hossza eggyel nagyobb, mint a normál működéshez tartozó ág hossza. Az elérhetőségi gráf minden csúcsához tartozik egy időbélyeg és egy tokeneloszlás, amely a rendszer egy adott állapotát írja le. A diagnosztikai szempontjából a *dev* hely tokeneloszlása érdekes, mert ez a hely tartalmazza az adott időpontig keletkezett eltéréseket.

Tegyük fel, hogy a rendszer működésének teljes karakterisztikus trace-e, és az ez alapján számított eltérések rendelkezésre állnak, azaz a technológiai folyamat befejeződött. Az ilyen az állapotoknak az elérhetőségi gráf végpontjai felelnek meg. A négy végpont az adott működési mód során létrejött összes eltérést tartalmazza. A keresést ilyen esetben elég a gráf végpontjai között elvégezni. Ennél az egyszerű folyamatnál az egyes működési módokhoz tartozó eltérések listái különbözőek, tehát a technológiai rendszerből származó teljes eltérés listát összehasonlítva a végpontok eltéréseivel, a bekövetkezett hiba egyértelműen azonosítható. A **8. ábrán** a 13-as csúcs a normál működés végállapotának felel meg, így itt az eltéréseket tartalmazó *dev* hely üres. A 14, 15, 16 sorszámmal rendelkező csúcsok a három különböző hibás működéshez tartozó végállapotok. Például a 14-es csúcs a szenzor negatív bias típusú meghibásodása esetén létrejött végállapotot jelenti. Látható, hogy ennél a csúcsnál a *dev* helyen megjelentek a működés során generált eltérések.

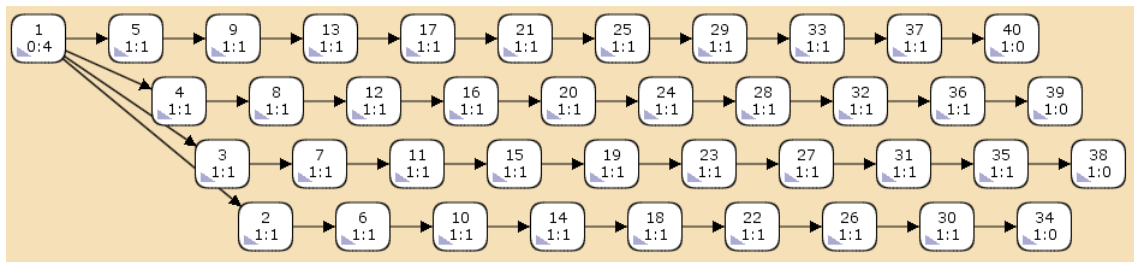


8. ábra: Tartály töltési folyamatához tartozó elérhetőségi gráf.

On-line diagnosztika esetében csak egy adott időpontig ismerjük a technológiai folyamatból származó trace-t és az eltéréseket. Ekkor az adott időbélyegű csúcsokhoz tartozó eltéréslistákat hasonlítjuk össze a trace által meghatározott eltérésekkel. Ebben a példában, ha az első időpontban keletkezett eltérés, akkor ez csak a szenzor negatív vagy pozitív bias hibája miatt jöhetett létre. Az eltérés típusából a bekövetkezett hiba egyértelműen azonosítható. Ha az első időpontban nem keletkezett eltérés, akkor két

eset lehetséges: a rendszer normál módon működik, vagy pedig lyukas a tartály. A két eset elkülönítése a második időpontban létrejött eltérések alapján lehetséges. Ha a második időpontban keletkezett eltérés, akkor a tartály lyukadása következett be, ellenkező esetben nem következett be hiba, és a rendszer normál módon működik.

Ha a szenzor értéktartományának felbontását finomítjuk, akkor a tartály feltöltése is hosszabb eseménysorral írható le. Ez azt jelenti, hogy az elérhetőségi gráfban az egyes működésekhez hosszabb utak tartoznak. Az **9. ábrán** erre láthatunk egy példát, itt a tartályt normál működés szerint az *l80* (azaz 80%-os) szintig töltjük.



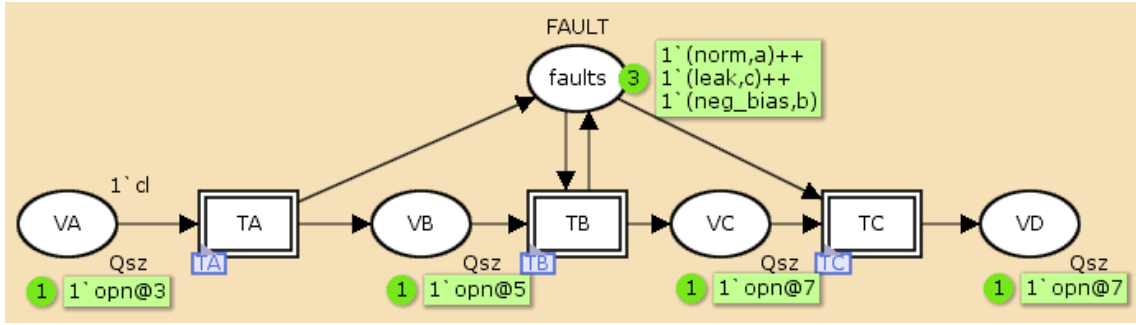
9. ábra: Tartály töltési folyamatához tartozó elérhetőségi gráf, a szenzor értéktartományának finomabb felbontása mellett.

5.4. Összetett technológiai rendszerek diagnosztikai vizsgálata

Miután elkészítettem egy tartály színezett Petri háló modelljét, ezt felhasználva több tartályból álló rendszereket hoztam létre. Az összetett rendszerek modellezésére hierarchikus Petri hálókat használtam. A hierarchikus Petri háló legfelső szintje a logikai kapcsolatokat írja le, egy-egy technológiai egység modelljét pedig az alsó szintű hálók tartalmazzák.

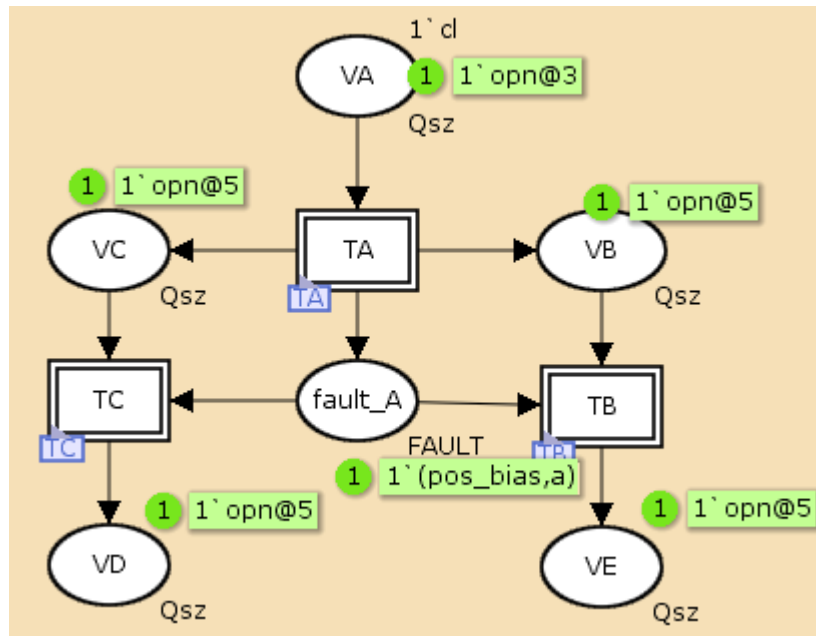
Először három egyforma tartály sorba kapcsolását modelleztem. Az első tartály (TA) kimenő szelepe (VB) egyben a második tartály (TB) bemenő szelepe, a VC szelep pedig hasonló módon kapcsolja össze a második (TB) és a harmadik (TC) tartályt. A technológiai folyamat célja mindhárom tartály feltöltése. Az egyes tartályok töltési folyamata hasonló, mint az egy tartályos példánál, azzal a különbséggel, hogy egy tartály töltése akkor kezdődik, amikor az előtte lévő tartály kimeneti szelepét kinyitjuk. (Megjegyzés: hasonlóan az egy tartályos példához, a tartály feltöltése az idő alapján történik, nem a szintjelző visszajelzése alapján.) A rendszer felső szintű Petri hálója a **10. ábrán** látható. Az alsó szintű hálók megfelelnek az egy tartályos példánál leírtaknak. Feltételezzük, hogy bármelyik tartály feltöltése során bekövetkezhet a korábban ismertetett háromféle hiba (lyukadás, érzékelő negatív vagy pozitív bias hibája) valamelyike, de egy technológiai egységnél csak egyféle hiba fordulhat elő. A felső szintű hálóban az előző tartályok hibáinak figyelembe vételére szolgál a *faults* hely. Feltételezzük, hogy a különböző tartályokban a hibák egymástól függetlenül következnek be. Mivel minden tartály négyféleképpen működhet, így összesen $4 \cdot 4 \cdot 4 = 64$ -féle végállapot jöhet létre, a teljes elérhetőségi gráf pedig 408 csúcsot tartalmaz (lásd **Melléklet 13. ábra**). Ezek között vannak olyanok, amelyeknél a bemeneti és kimeneti változók értékei és az eltérések listája azonos, noha nem ugyanazok a hibák következtek be a rendszer működése során. Például, ha a második tartálynál lyukadás történt, a harmadik viszont normál módon működik, akkor az utolsó két tartály üres marad, de ugyanez a helyzet áll elő akkor is, ha az utolsó két tartály mindegyike lyukas. Ebben az

esetben az eltérések vizsgálata alapján csak a lehetséges hibák halmaza határozható meg.



10. ábra: Három tartály sorba kapcsolása.

A második példa egy nagyobb gyűjtőtartályból (TA) és a hozzá csatlakozó két kisebb tartályból (TB, TC) áll. Tételezzük fel, hogy a TA tartályt kétszer akkora térfogatárammal töltjük, mint a TB és TC tartályokat, így a feltöltés ideje a három tartálynál azonos. A rendszer felső szintű Petri hálója a **11. ábrán** látható. A párhuzamos kapcsolat következményeként a TB és TC átmenetei egyszerre engedélyezettek. Ekkor a következő átmenet kiválasztása véletlenszerűen történik, ami azt jelenti, hogy egy állapotot akár többféle tüzelési sorozattal is el lehet érni. Emiatt a rendszer elérhetőségi gráfja nem lesz fa struktúrájú. Ezt a problémát úgy küszöböltem ki, hogy a TB tartály átmeneteinek magasabb prioritást adtam, így egy időpontban először a TB-hez, majd a TC-hez tartozó átmenet fog lejátszódni. Ezáltal az átmenetek lejátszódási sorrendje egyértelmű, az elérhetőségi gráf pedig fa lesz. A soros esethez hasonlóan a működési módoktól függően itt is 64-féle végállapot jöhet létre, a teljes elérhetőségi gráfnak pedig 453 csúcsa van (lásd **Melléklet 14. ábra**). A rendszer diagnosztikájára vonatkozóan ebben az esetben is hasonló megállapítások tehetők, mint a soros kapcsolás esetén.



11. ábra: Három tartály párhuzamos kapcsolása.

5.5. Összetett technológiai rendszerek diagnosztikája strukturális dekompozícióval

Az előzőekben bemutatott viszonylag kis méretű összetett rendszerek példáján is jól látszik az elérhetőségi gráfok méretének robbanásszerű növekedése. Már három tartály esetén is a teljes rendszerhez tartozó elérhetőségi gráfok több száz csúcsot tartalmaznak. A szenzor értéktartományának finomításával a gráfok még nagyobbak lesznek, mert egy-egy működéshez hosszabb ágak tartoznak. A gráfok méretének növekedésével a keresésekhez szükséges számításigény is nő. Ezért a diagnosztikai eljárás gyakorlati használhatósága szempontjából kulcsfontosságú a dekompozíció lehetősége.

Az összetett rendszerek felbonthatók az őket alkotó technológiai egységekre. Ezt kihasználva a diagnosztikai vizsgálatot célszerű technológiai egységenként külön-külön elvégezni. Ennek megvalósításához első lépésben a teljes rendszerhez tartozó trace-ből ki kell nyerni azokat az információkat, amelyek a vizsgált egységre vonatkoznak. Ehhez először megkeressük az adott egység tényleges működéséhez tartozó időpontokat. Az ezekhez tartozó eseményekben szereplő változók közül kiválasztjuk azokat, amelyek a vizsgált egységre vonatkoznak, a többi változót pedig elhagyjuk. Ezután az események bekövetkezésének időpontjait eltoljuk úgy, hogy az első eseményhez tartozó időpont 1 legyen. Így az összetett rendszer működését leíró trace-t feldaraboljuk az egyes alrendszerek működését leíró eseménysorokra. Ezután generáljuk az eltéréseket egy-egy technológiai egységre vonatkozóan, majd az adott technológiai egység színezett Petri hálójához tartozó elérhetőségi gráf alapján végezzük el az alrendszer diagnosztikáját.

Több tartály összekapcsolását leíró modell elkészítésénél figyelembe kell venni, hogy az előző tartályokban bekövetkezett hiba milyen hatással van az aktuális tartály működésére. Ezért egy tartály színezett Petri hálójában a fault helyen az aktuális és az előző tartályok működési módjait tároltam. Hogy megkülönböztessem őket, a működési módot leíró szint kiegészítettem egy plusz attribútummal. Például (*norm*, *act*) jelentése: az aktuális tartály normál módon működik, (*leak*, *prev*) jelentése: az előző tartály lyukas.

Ha az első tartály analízisével a kapott eredmény 1 darab azonosított hiba, akkor a második technológiai egység vizsgálatánál ezt az eredményt – mint előző hibát – hozzáadjuk a *fault* helyhez a korábban ismertetett módon. Ezután előállítjuk a második technológiai egységhez tartozó elérhetőségi gráfot és az eltéréseket az erre az egységre vonatkozó trace felhasználásával. Az így kapott elérhetőségi gráf azokat az állapotokat és eltéréslistákat tartalmazza, amelyek az előző egységben történt hiba figyelembevételével jöhetnek létre. Ezen a gráfon is elvégezzük a diagnosztikai vizsgálatot, a kapott eredményt pedig ismét felhasználjuk a következő egység vizsgálatához, és így tovább. Ennek a módszernek az előnye, hogy kisebb gráfon és kevesebb összehasonlítandó eltéréssel kell elvégezni a keresést, mint ha a teljes rendszert vizsgálnánk.

Tekintsük a három sorba kapcsolt tartály példáját. A teljes rendszerhez tartozó nominális trace a **8. táblázatban** látható. A trace 1., 2., 3. időponthoz tartozó sorának VA, VB és TA oszlopai alkotják az első tartályhoz tartozó trace-t, a 3., 4., 5. sorok VB, VC, és TB oszlopai a második tartályhoz tartozó trace-t, és az 5., 6., 7. sorok VC, VD és TC oszlopai a harmadik tartályhoz tartozó trace-t.

8. táblázat: A soros rendszerhez tartozó teljes nominális trace és az egyes egységekhez tartozó trace darabok. Piros: első tartály, kék: második tartály, zöld: harmadik tartály.

idő	Bemeneti változók				Kimeneti változók		
	VA	VB	VC	VD	TA	TB	TC
1	opn	cl	cl	cl	null	null	null
2	opn	cl	cl	cl	L	null	null
3	opn	opn	cl	cl	N	null	null
4	opn	opn	cl	cl	N	L	null
5	opn	opn	opn	cl	N	N	null
6	opn	opn	opn	cl	N	N	L
7	opn	opn	opn	opn	N	N	N

Tegyük fel, hogy a technológiai rendszerből a **9. táblázatban** látható karakterisztikus trace nyerhető ki.

9. táblázat: A soros rendszerhez tartozó karakterisztikus trace.

idő	Bemeneti változók				Kimeneti változók		
	VA	VB	VC	VD	TA	TB	TC
1	opn	cl	cl	cl	L	null	null
2	opn	cl	cl	cl	N	null	null
3	opn	opn	cl	cl	H	null	null
4	opn	opn	cl	cl	H	null	null
5	opn	opn	opn	cl	H	null	null
6	opn	opn	opn	cl	H	null	null
7	opn	opn	opn	opn	H	null	null

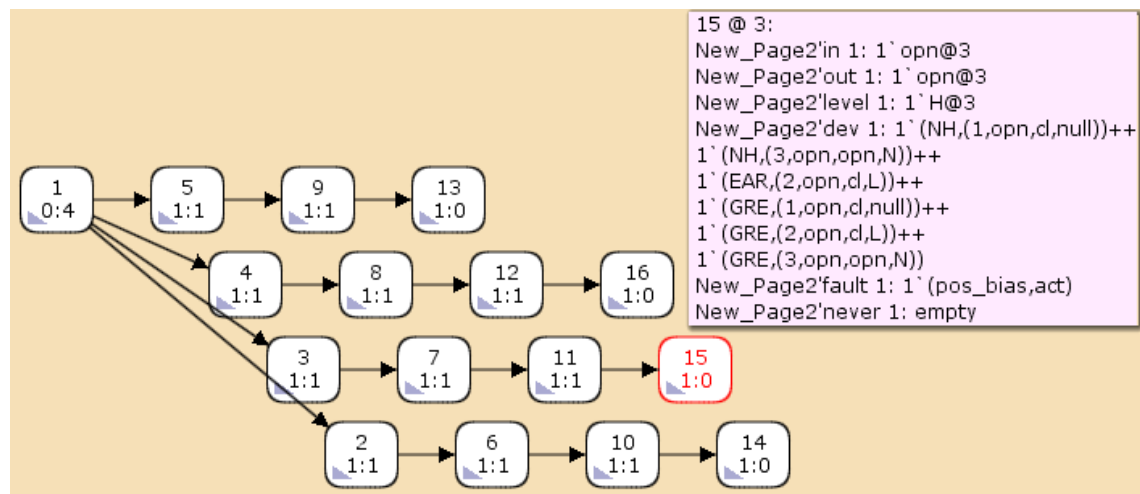
A karakterisztikus trace-ből kivágjuk az egyes tartályokhoz tartozó trace darabokat. Az egyes tartályokhoz tartozó eseménysorok a **10. táblázatban** láthatók.

10. táblázat: Az egyes tartályokhoz tartozó trace-ek.

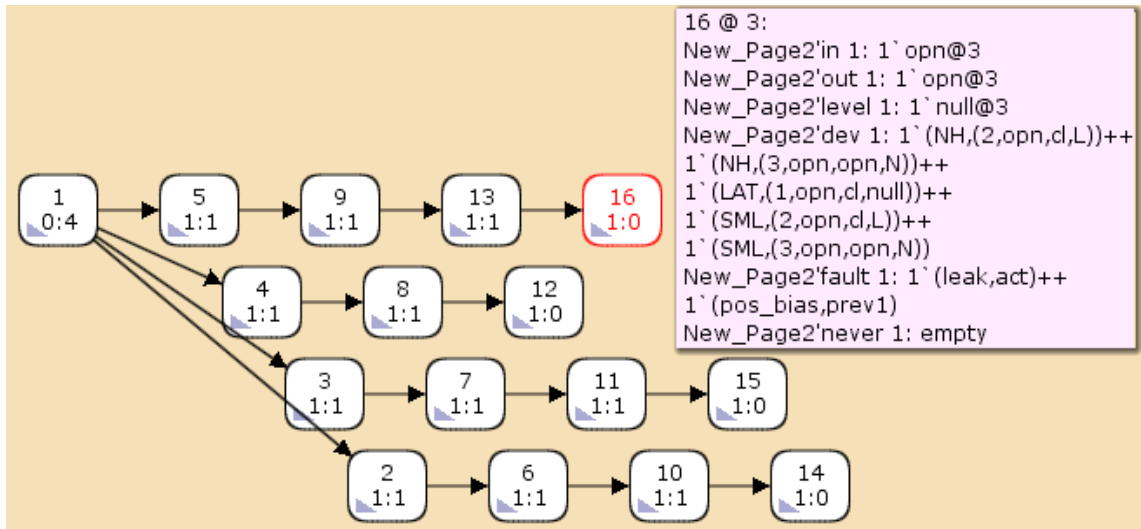
tartály	trace		
TA	(1,opn,cl,L)	(2,opn,cl,N)	(3,opn,opn,H)
TB	(1,opn,cl,null)	(2,opn,cl,null)	(3,opn,opn,null)
TC	(1,opn,cl,null)	(2,opn,cl,null)	(3,opn,opn,null)

A vizsgálatot az első tartállyal kezdjük. Előállítjuk a TA tartály működését leíró trace darab (**10. táblázat** első sora) alapján előálló eltéréseket, és ezeket összevetjük az egy tartály feltöltéséhez tartozó elérhetőségi gráf (**12. ábra**) végpontjaival. A 15. sorszámú

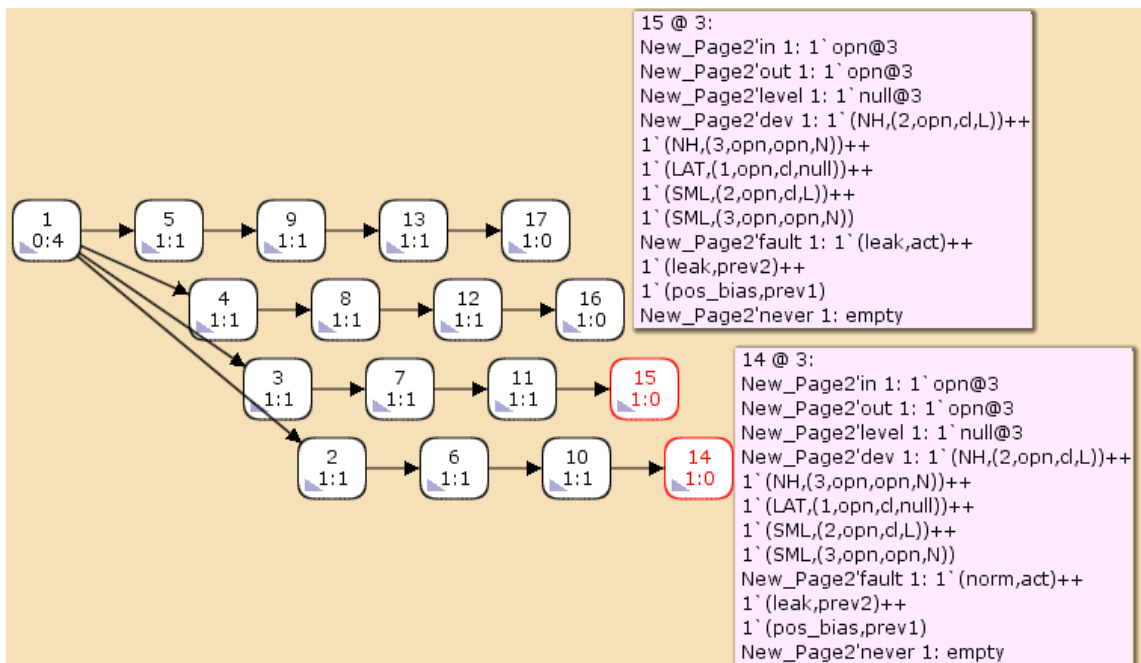
csúcs eltérés listája tartalmazza pontosan ezeket az eltéréseket, így a *fault* hely tokenje alapján megállapítható, hogy az első tartályban a szintérzékelőnek pozitív bias hibája van. Ezt az információt felhasználjuk a következő tartály vizsgálatakor, azaz a *fault* helyhez egy (*pos_bias*, *prev1*) értékű tokenet adunk. Ennél a tartálynál is generáljuk az elérhetőségi gráfot. A kapott gráf a **13. ábrán** látható, és azt mutatja meg, hogy milyen állapotokat kaphatunk a második tartály feltöltésekor, ha az előző tartályban az érzékelő pozitív bias hibával működött. Ezután előállítjuk a TB tartály működését leíró trace (**10. táblázat** második sora) alapján adódó eltéréseket, és megkeressük az ennek megfelelő végpontot az elérhetőségi gráfon. A hiba, vagyis a tartály lyukadása ebben az esetben is egyértelműen azonosítható (16-os csúcs). A harmadik tartály modelljében a *fault* helyhez mindkét előző hibát hozzáadjuk egy (*pos_bias*, *prev1*) valamint egy (*leak*, *prev2*) értékű token formájában. Ezután generáljuk a harmadik tartály elérhetőségi gráfját is. A kapott gráf a **14. ábrán** látható. Ez a gráf megadja a harmadik tartály feltöltésekor előálló állapotokat, ha az előző tartályokban pozitív bias hiba, illetve lyukadás fordult elő. A TC tartály feltöltésekor kapott karakterisztikus trace darab alapján ismét generáljuk az előálló eltéréseket és ezeket összevetjük az elérhetőségi gráf végpontjaival. Az eltérések alapján két lehetséges működési mód azonosítható: a harmadik tartály vagy normál módon működik (14-es csúcs), vagy lyukas (15-ös csúcs).



12. ábra: Az első tartály elérhetőségi gráfja.



13. ábra: A második tartályhoz tartozó elérhetőségi gráf, ha az első tartályban pozitív bias hiba történt.



14. ábra: A harmadik tartályhoz tartozó elérhetőségi gráf, ha a második tartály lyukas.

IV. Összefoglalás

Munkám során elkészítettem egy két gyártósorból és az azokat kiszolgáló robotból felépülő gyártórendszer működését leíró színezett Petri háló modellt. A gyártósoron hibaesemények bekövetkezését feltételezve a működés során előforduló hibalehetőségeket beépítettem a rendszer modelljébe. Ezzel megmutattam, hogy a korábbi kutatásink során alkalmazott hibabeépítési eljárás akkor is működik, ha a hibamentes működést leíró modell nem alacsony szintű, hanem színezett Petri háló. Az elkészített színezett Petri háló modelleknél időzítést is alkalmaztam a valós működés jobb leírására. Elkészítettem a gyártórendszer fenti színezett Petri háló modelljeit olyan változatban is, amikor a munkadarabok nem rendelkeznek egyedi azonosítóval. A vizsgált gyártórendszer különböző Petri háló alapú modelljeinek elkészítésére a CPNTools szoftvert használtam.

Megvizsgáltam a hiba beépítésének, illetve az időzítés bevezetésének a hatását az elérhetőségi gráf jellemzőire (méret, szerkezet, végpontok száma, holtpontok megjelenése). Megállapítottam, hogy a modellezés során a háló színezésének kialakítása (egyedi azonosító alkalmazása/nem alkalmazása) jelentősen befolyásolja az elérhetőségi gráf méretét.

Az elérhetőségi gráfon élsúlyozást bevezetve meghatározhatóak az egyes rendszerállapotok bekövetkezési valószínűségei. Ezt az OGAnalyzer szoftver felhasználásával konkrét példákon mutattam be.

Egyszerű és összetett technológiai folyamatok példáján bemutattam, hogy a diszkrét, vagy kvalitatív változókat tartalmazó eseménysorok hogyan modellezhetőek színezett Petri hálókkal. Vizsgáltam, hogy a technológiai folyamat valós eseménysorát ismerve a Petri háló elérhetőségi gráfja miként használható az esetlegesen előforduló hibák azonosítására.

Összetett folyamatok vizsgálatánál bemutattam, hogy strukturális dekompozícióval hogyan lehet a teljes folyamat diagnosztikáját a részrendszerek vizsgálatára visszavezetni.

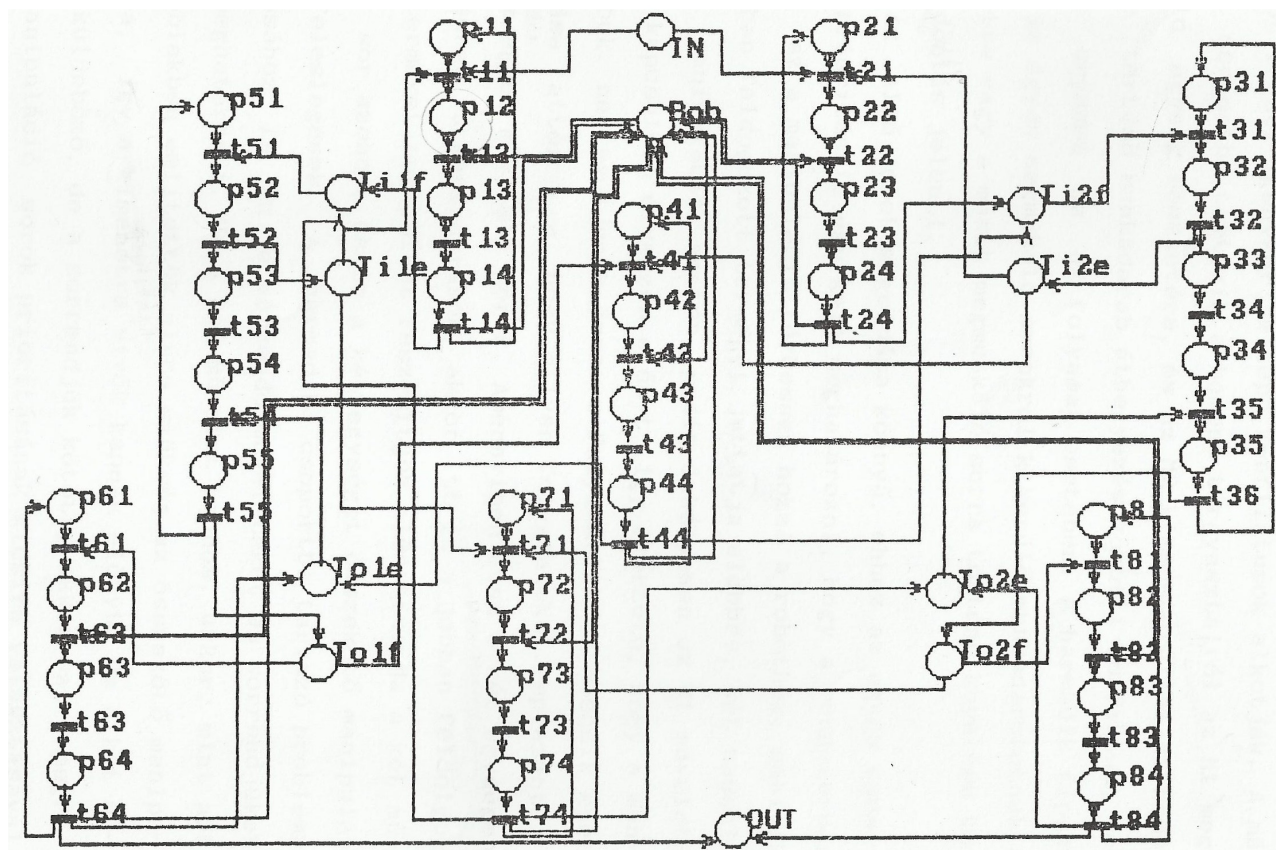
A dolgozatban bemutatott eredményeket felhasználtuk a [15], [16], [17] közleményekben.

A dolgozat a TÁMOP-4.2.2.A-11/1/KONV-2012-0072 projekt keretében Magyarország és az Európai Unió közös támogatásával készült.

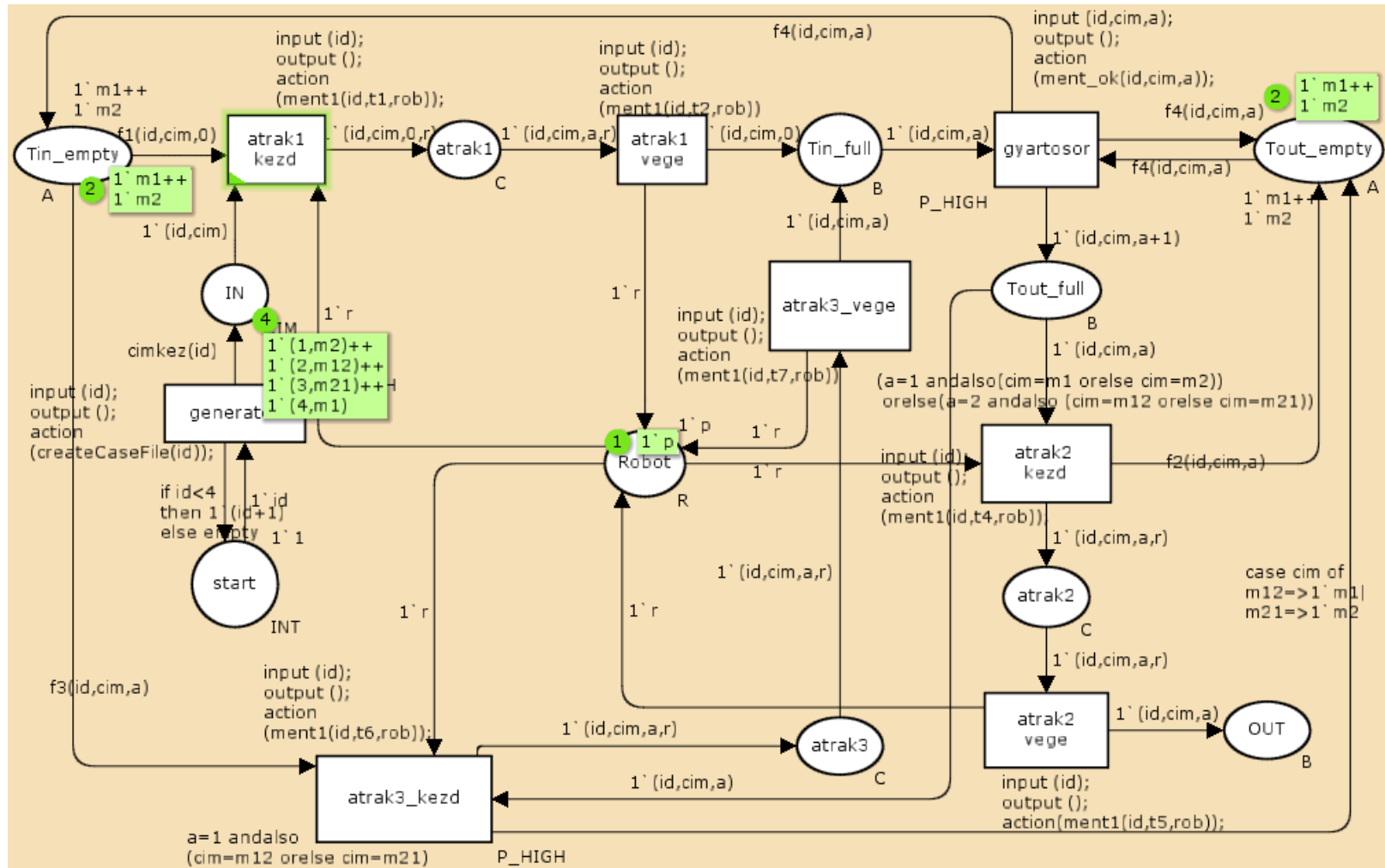
V. Irodalomjegyzék

- [1] **C. Cassandras, S. Lafortune:** *Introduction to Discrete Event Systems*, Kluwer Academic Publishers, **1999**
- [2] **M. Zhou, K Venkatesh:** *Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach*, World Scientific, **1999**
- [3] **M. Gerzson, B. Márczi, A. Leitold:** *Diagnosis of Technological Systems based on their Coloured Petri Net Model*, ARGESIM Report, no. S38, pp. 358/1-6., **2012**
- [4] **L. Yaqiong, C. K. M. Lee, H. K. Chan, W. H. Ip:** *Rfid-based colored petri net applied for quality monitoring in manufacturing system*, The International Journal of Advanced Manufacturing Technology, vol. 60, pp. 225-236, **2012**
- [5] **A. Tóth, Á. Werner-Stark, K. M. Hangos:** *HAZID information based operational procedure diagnosis method*, 12th International PhD Workshop on Systems and Control, Veszprém, (A. Magyar, ed.), 1–6 (on CD), **2012**
- [6] **T. Murata:** *Petri Nets: Properties, Analysis and Applications*, Proceedings of the IEEE, 77, no. 4, **1989**
- [7] **K. Jensen:** *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*, Springer-Verlag, **1997**
- [8] **A. Høyland, M. Rausand:** *System Reliability Theory: Models and Statistical Methods*, John Wiley & Sons, **2004**
- [9] **T. A. Kletz:** *Hazop and Hazan*, The Institution of Chemical Engineers, **2006**
- [10] **Center for Chemical Process Safety:** *Guidelines for Hazard Evaluation Procedures*, John Wiley & Sons, **2008**
- [11] **M. Gerzson:** *Petri háló alkalmazása technológiai rendszerek vizsgálatára*, (egyetemi doktori értekezés), Veszprém, **1989**
- [12] <http://CPN Tools.org> (2012. december 8.)
- [13] **B. Márczi:** *Technológiai rendszerek diagnosztikai vizsgálata folyamatinformációk alapján generált Petri hálókkal*, (tudományos diákköri dolgozat), Veszprém, **2011**
- [14] **B. Márczi, A. I. Pózna:** *Technológiai rendszerek vizsgálata időzített, színezett Petri háló segítségével*, X. Jedlik Ányos Szakmai Napok, Veszprém, április 11-13., **2013**
- [15] **M. Gerzson, B. Márczi, A. I. Pózna, A. Leitold:** *Investigation of Manufacturing Systems Using Timed Coloured Petri Net*, Factory Automation Conference, Veszprém, Hungary, **2013**
- [16] **A. Leitold, B. Márczi, A. I. Pózna, M. Gerzson:** *Monitoring and Diagnosis of Manufacturing Systems Using Timed Coloured Petri Net*, Hungarian Journal of Industry and Chemistry, vol. 41, no 1., pp. 17-26, **2014**
- [17] **A. Leitold, M. Gerzson, A. I. Pózna, K. M. Hangos:** *On-line qualitative model-based diagnosis of technological systems using colored Petri nets*, European Simulation and Modelling Conference, pp. 332-336, Porto, Portugal, **2014**

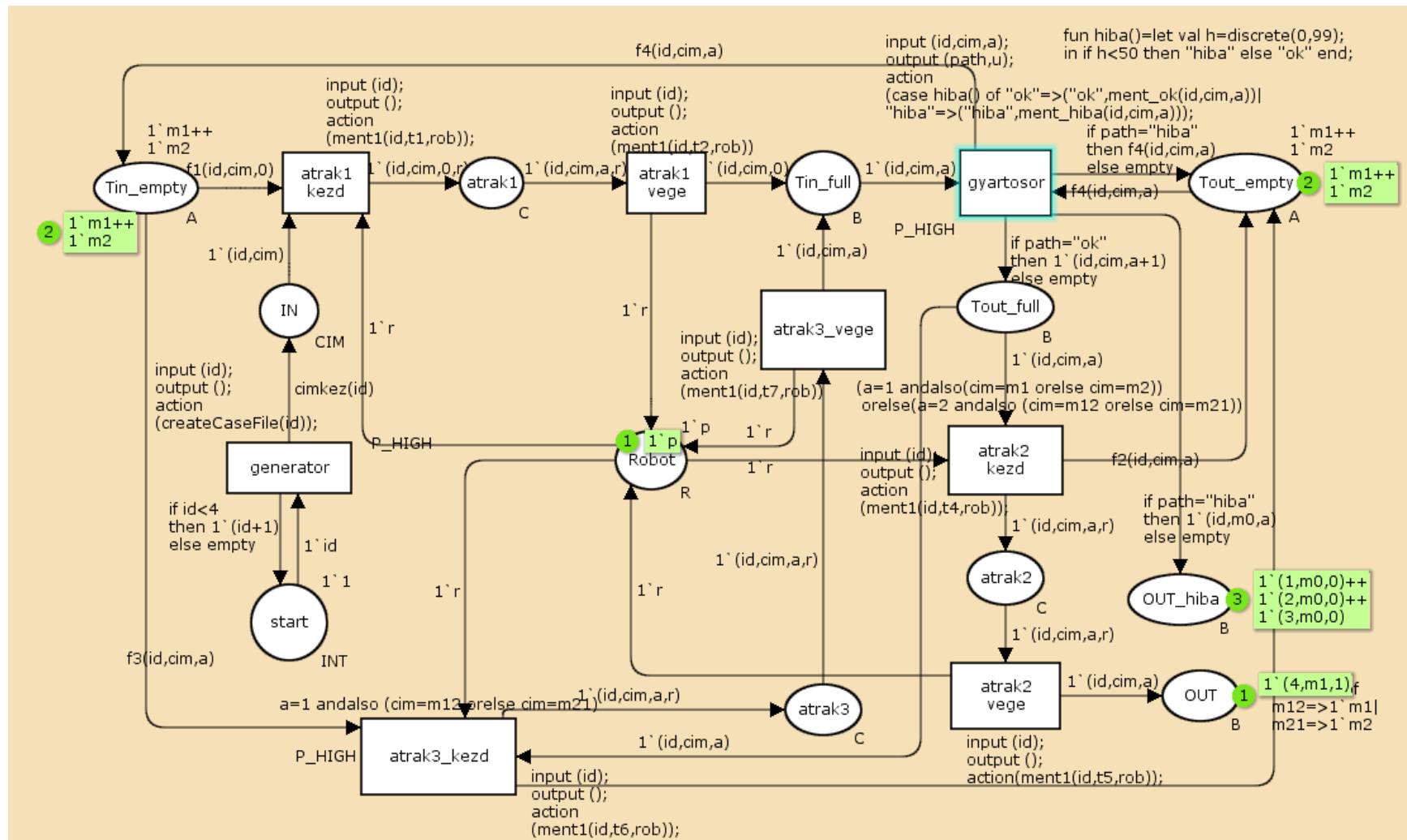
VI. Mellékletek



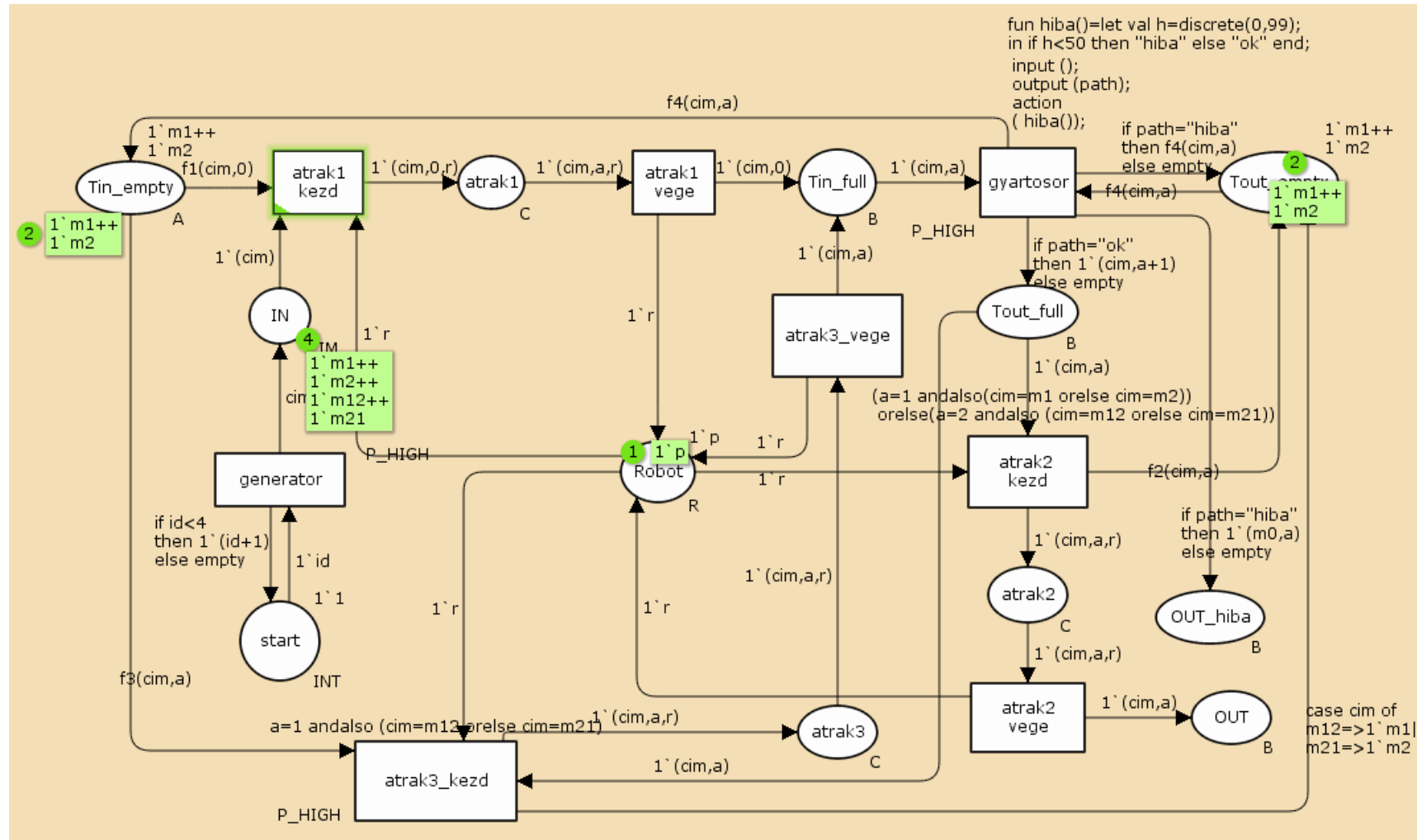
Melléklet 1. ábra: A gyártórendszer alacsony szintű Petri hálója.



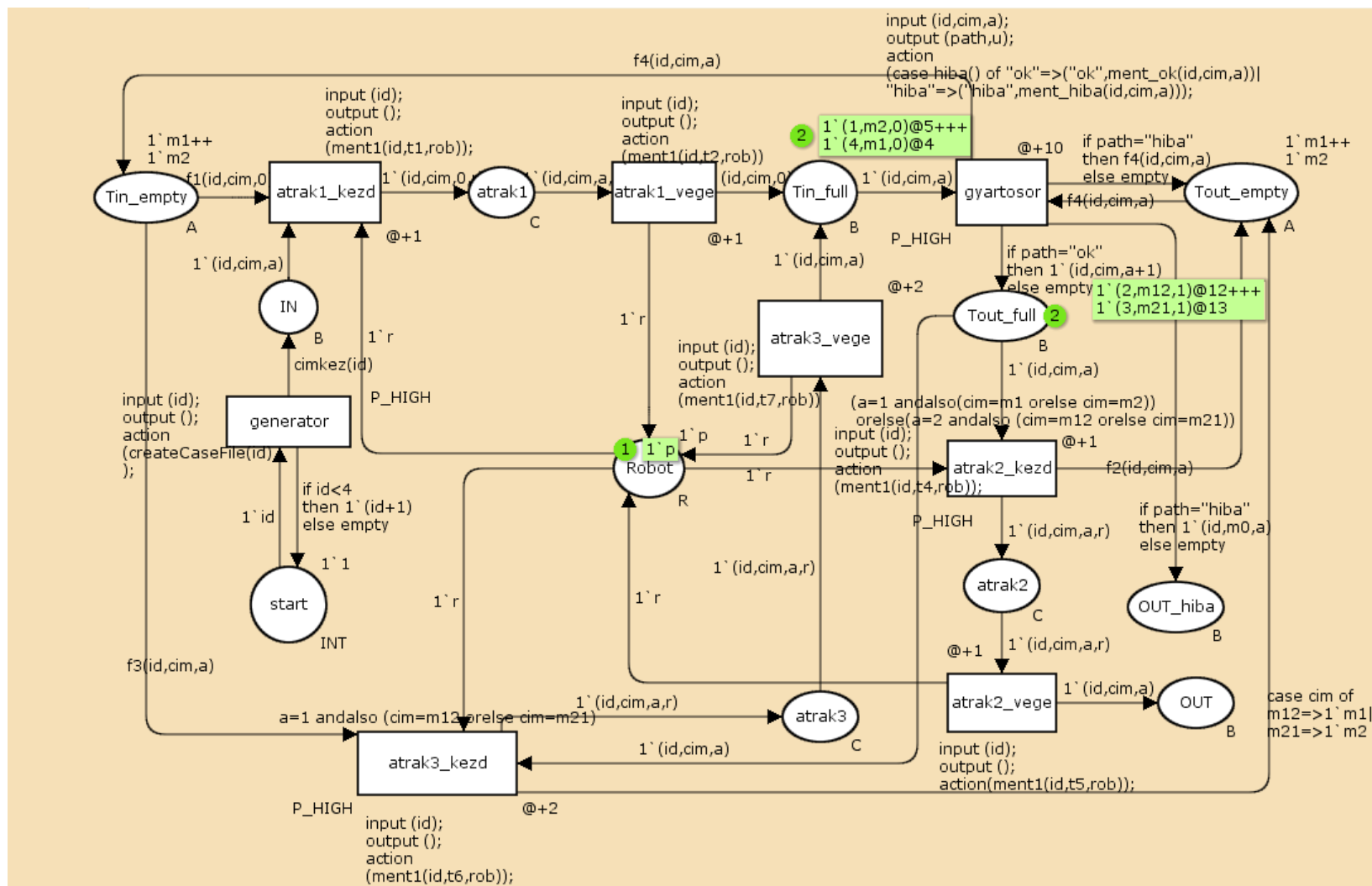
Melléklet 2. ábra: A gyártórendszer hibamentes működését leíró színezett Petri háló.

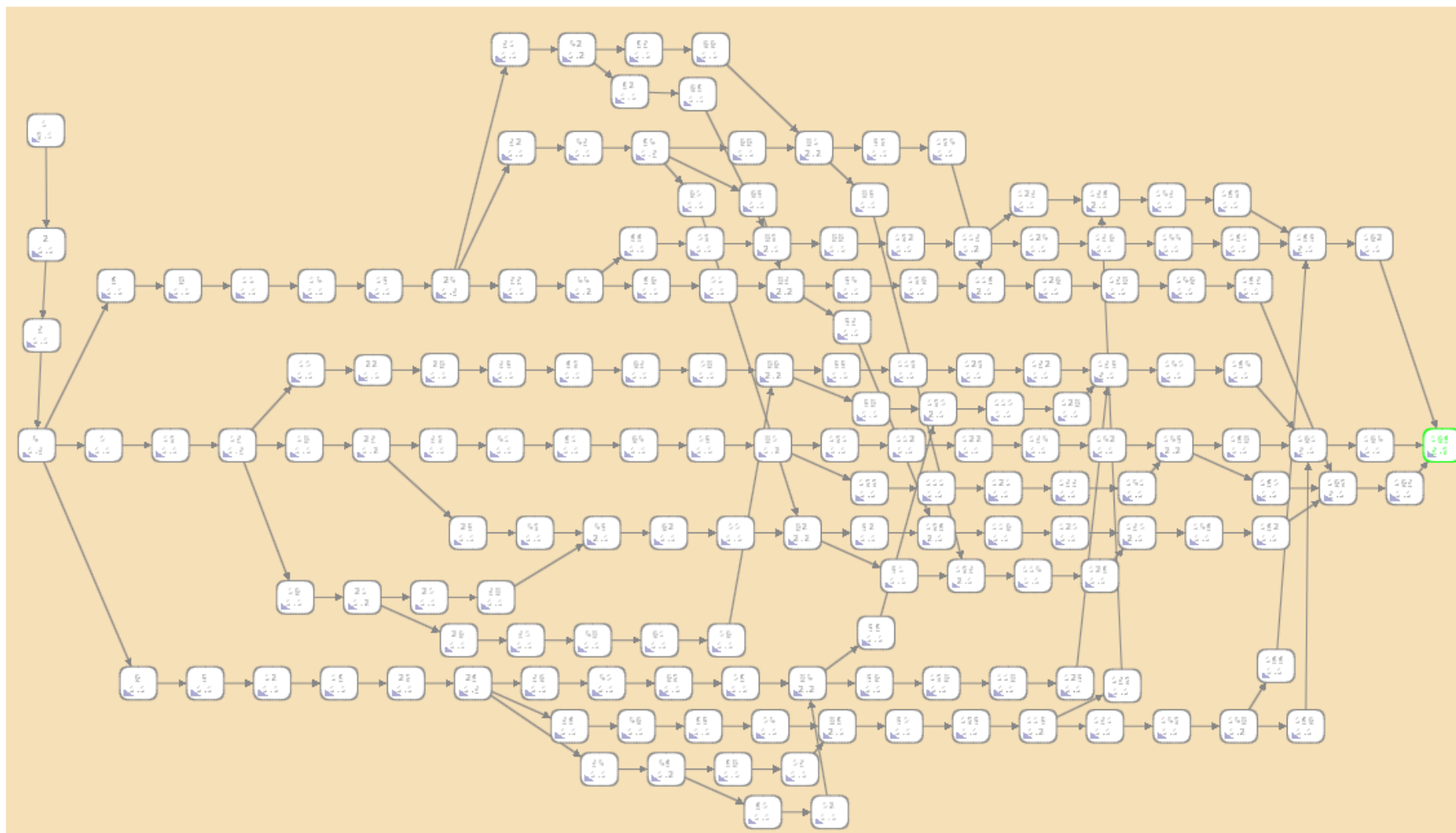


Melléklet 3. ábra: A gyártórendszer hibalehetőséget is tartalmazó színezett Petri háló modellje.

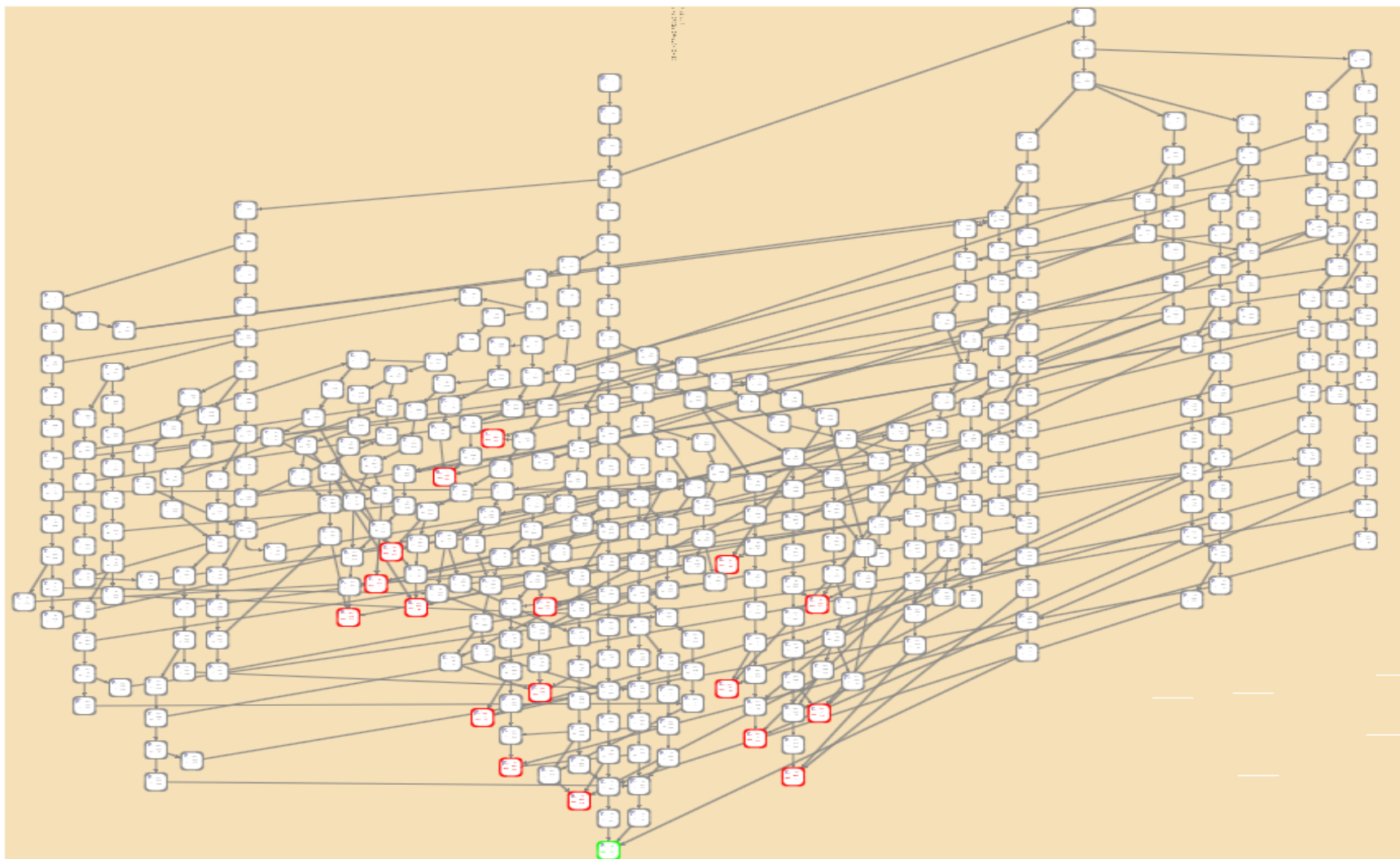


Melléklet 5. ábra: A gyártórendszer színezett Petri hálója, ha a munkadaraboknak nincs azonosítója.

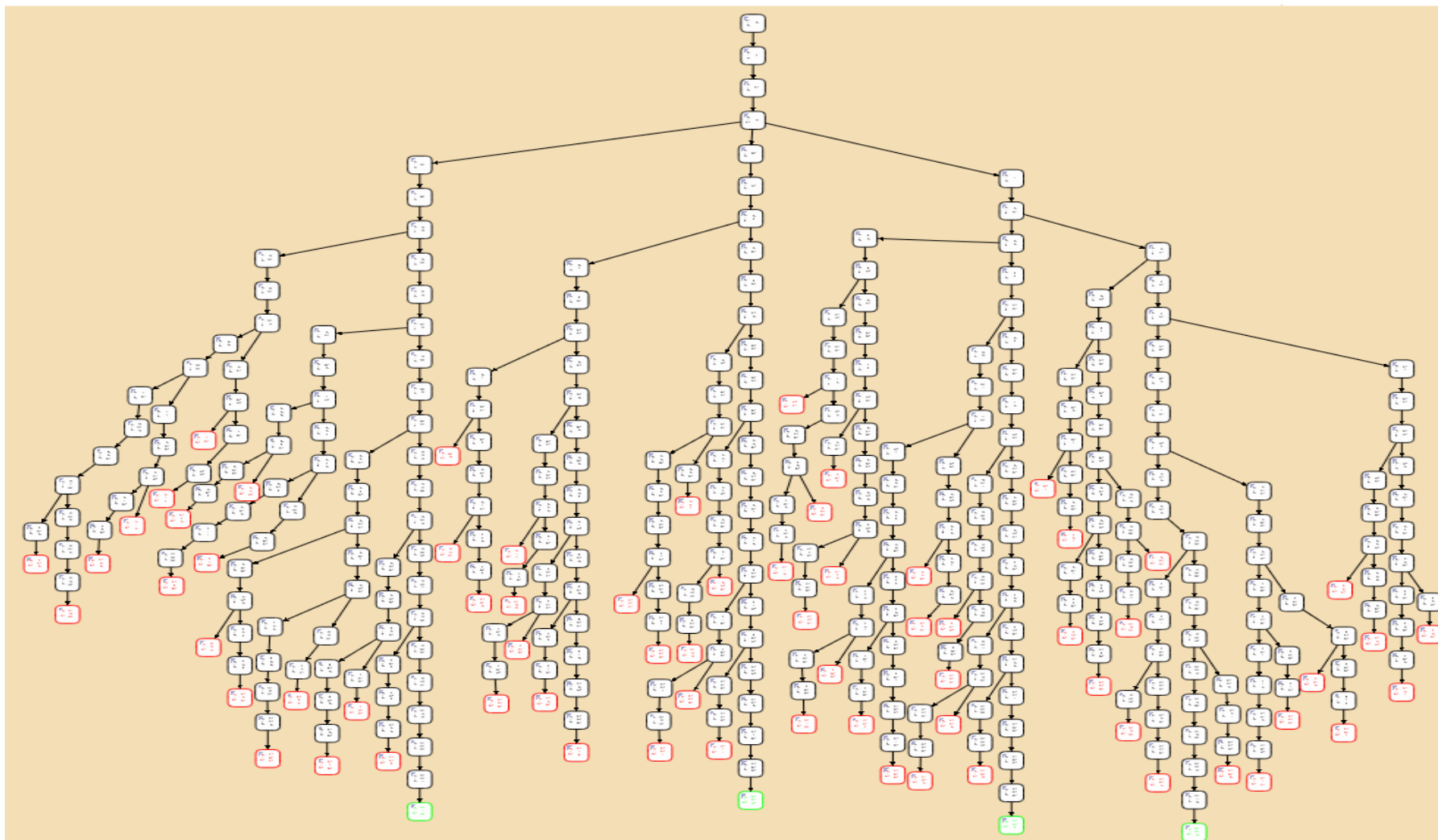




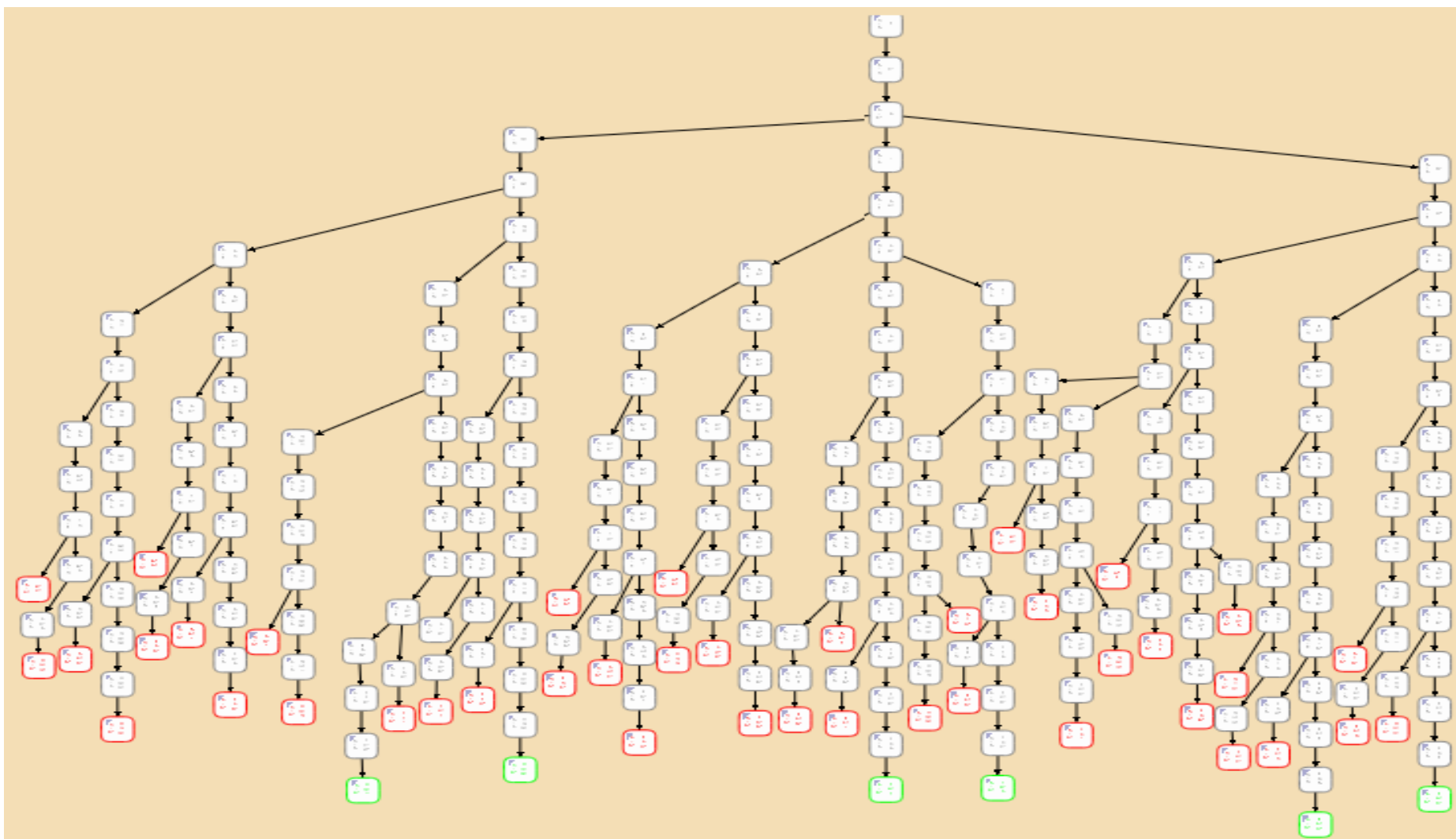
Melléklet 8. ábra: A gyártórendszer hibamentes működéséhez tartozó Petri háló elérhetőségi gráfja. (3 munkadarab m2, m12, m21 címkével)



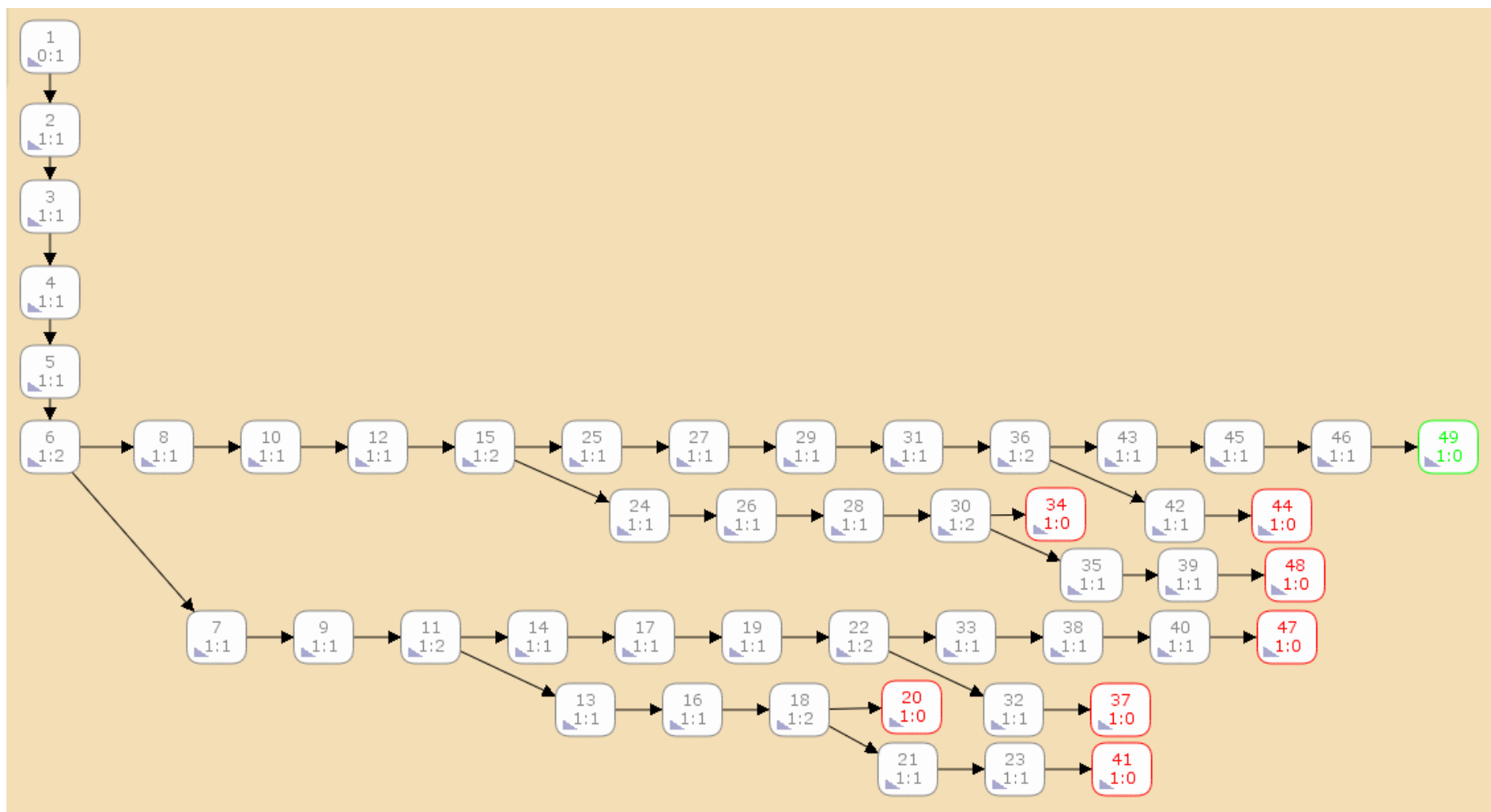
Melléklet 9. ábra: Hibalehetőséget is tartalmazó színezett Petri háló elérhetőségi gráfja. (3 munkadarab m2, m12, m21 címkével)



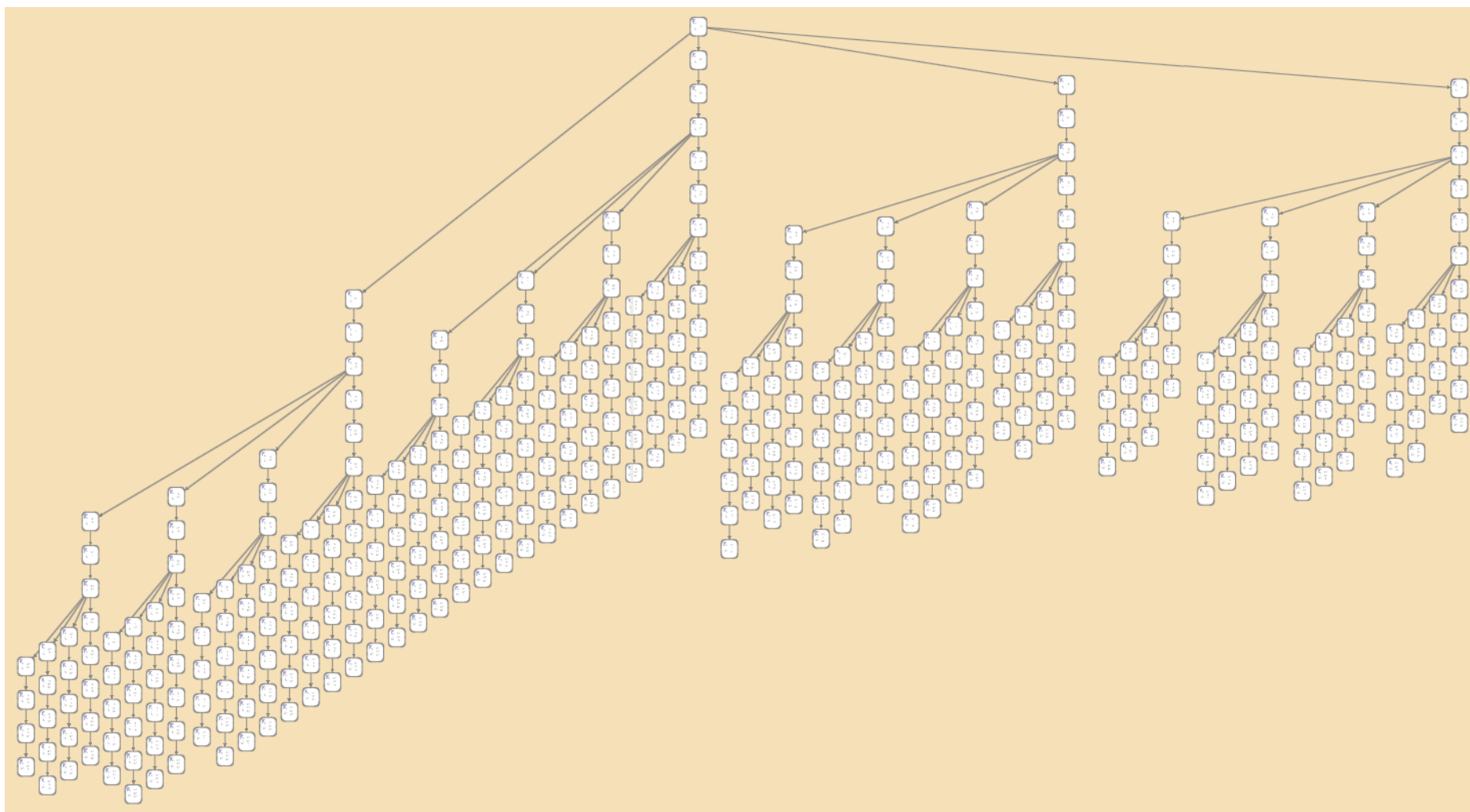
Melléklet 10. ábra: Példa elérhetőségi gráfra időzített esetben. (3 munkadarab m2, m12, m21 címkével, időzítések: atrak{1,2,3}_{kezd,vege}:@+1, gyartoros:@+5)



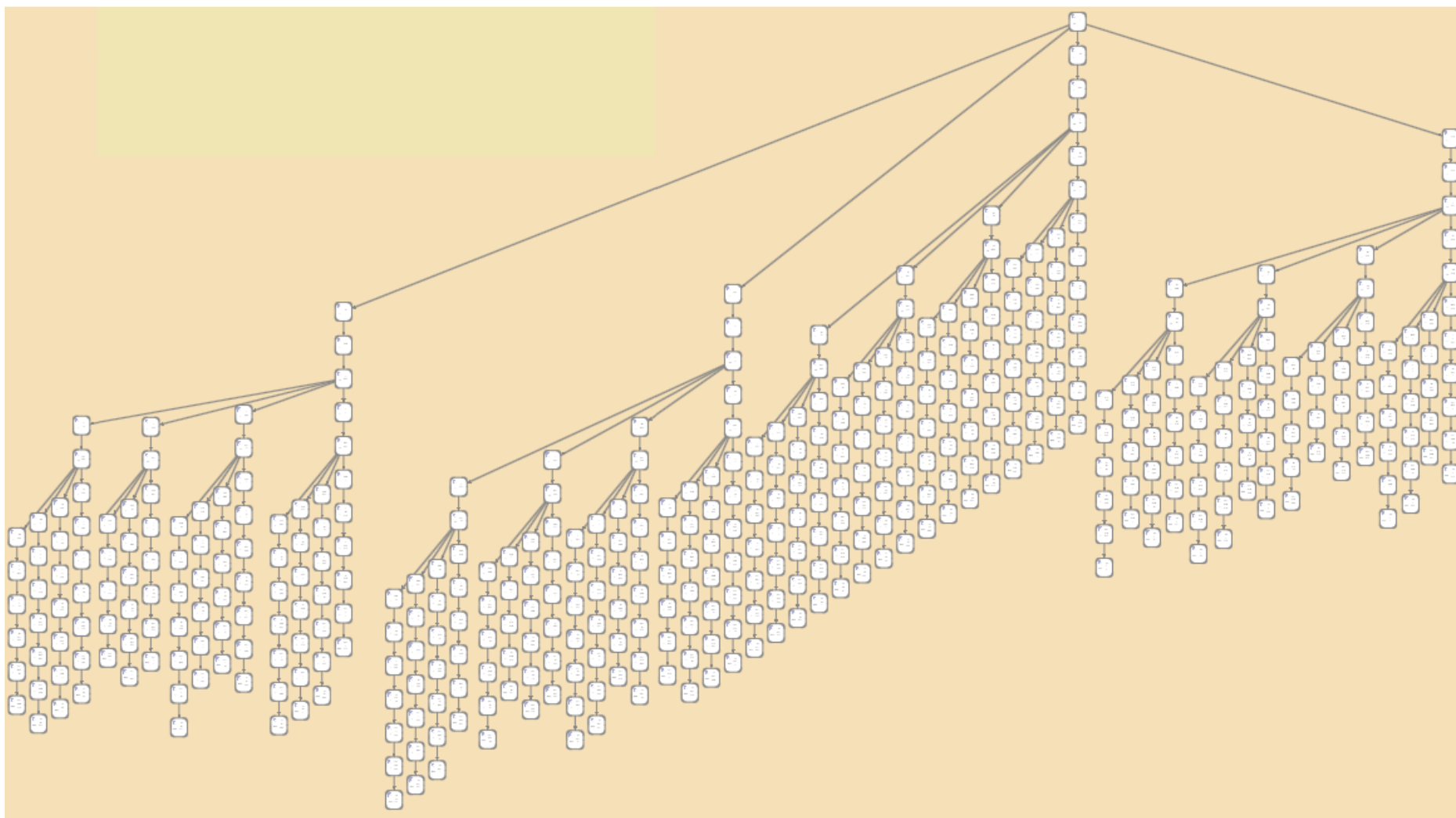
Melléklet 11. ábra: Az időzített színezett Petri háló elérhetőségi gráfja, 3 db m1 címkéjű, azonosítóval rendelkező munkadarab gyártása esetén.



Melléklet 12. ábra: Az időzített színezett Petri háló elérhetőségi gráfja, 3 db m1 címkéjű, azonosító nélküli munkadarab gyártása esetén.



Melléklet 13. ábra: A soros rendszer elérhetőségi gráfja.



Melléklet 14. ábra: A párhuzamos rendszer elérhetőségi gráfja.