

PETRIDOTNET 1.3 FELHASZNÁLÓI ÚTMUTATÓ

Utolsó módosítás: 2011. július 11.
Aktuális programverzió: 1.3.4209

I. Bevezető

A PetriDotNet program Petri-hálók szerkesztésére, szimulációjára és analízisére szolgál. A BME Méréstechnika és Információs Rendszerek tanszékének szakmai segítségével készült azzal a céllal, hogy az oktatásban is használható, könnyen kezelhető és bővíthető program legyen.

II. Rendszerkövetelmények

A PetriDotNet program futtatásához a Microsoft .NET (3.5 vagy újabb) keretrendszer szükséges Windows (XP vagy újabb) operációs rendszer alatt. A fejlesztés közben arra törekedtünk, hogy a program a Mono futtatókörnyezettel is kompatibilis maradjon, így Linux és Mac OS X platformokon is futtatható legyen. Azonban a Mono környezet nem tesztelt, így nem is támogatott.

Kisebb méretű (néhány száz csomópontot tartalmazó) Petri-hálók esetén a program memóriaigénye nem jelentős (10–20 MB nagyságrendű), ugyanakkor nagy (egy alhálóban több ezer elem feletti) hálók esetén már jelentős az erőforrásigény, illetve ilyen méretű hálók gyors megjelenítésére a program egyelőre nincs optimalizálva.

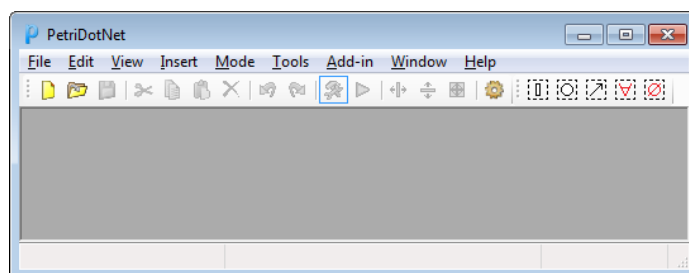
A program nem igényel telepítést. A tömörített állományból kibontva, tetszőleges helyre másolva futtatható.

III. A program használata

Igyekeztünk olyan programot készíteni, aminek használata könnyen és gyorsan elsajátítható. Az alábbiakban rövid betekintést nyújtunk a program kezelésébe, az egyes funkciók működésébe, a program különböző kezelőfelületeinek használatába.

III.1. Indítás

A PetriDotNet program indítása után egy üres szerkesztőablak fogad:



A programot a főmenüből, illetve az eszközsávról kiválasztott parancsokkal vezérelhetjük. Indítás után választhatunk, hogy egy új üres hálót készítünk (*File » New* parancs) vagy egy korábban elmentett hálót töltünk be (*File » Open* parancs).

III.2. Betöltés, mentés

A program jelenleg kétféle fájlformátumot használ:

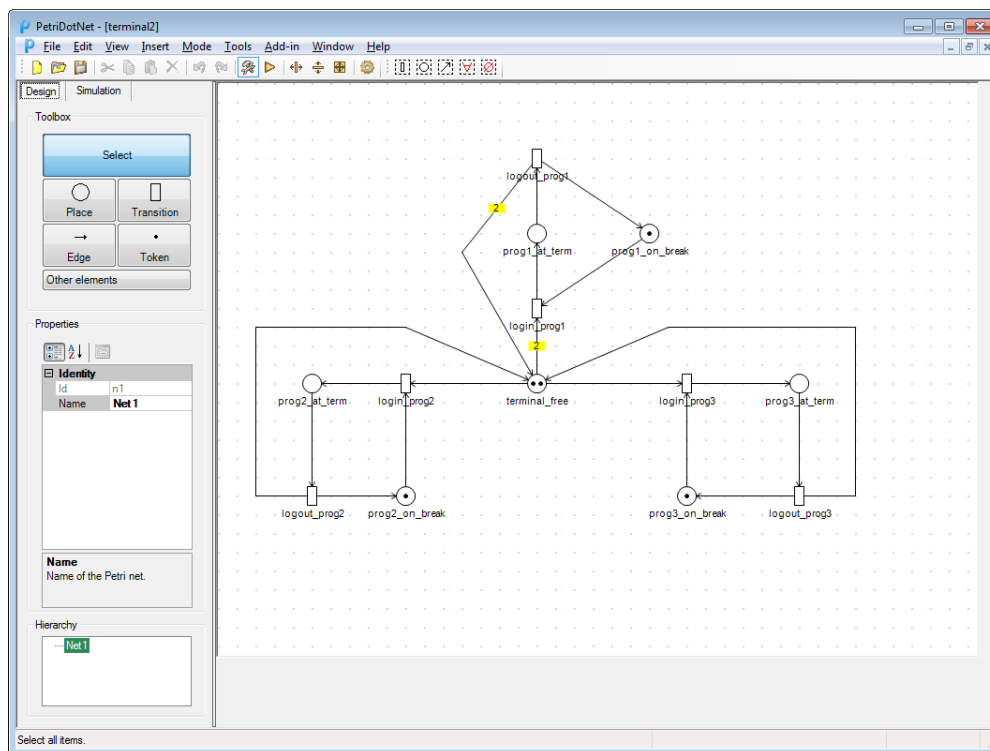
- .pnml: XML alapú szöveges fájlformátum. A PNML a Petri-hálók szabványos tárolási formátuma, más Petri-háló programok is támogatják. Ennek a használata javasolt.
- .pn: Bináris fájlformátum. Gyorsabb a betöltése, viszont az olvashatósága csak egy programverzió belül garantált.

Ezen felül jelenleg egy további fájlformátum érhető el beépülő kiterjesztés (add-in) formában (az *Add-in » Export* menüpont alatt):

- .pnt: az INA analízis program által elfogadott formátum. Ebbe a formátumba mentve tudjuk a hálót az INA programmal beolvastatni és a tulajdonságait elemezni.

III.3. Petri-hálók létrehozása, szerkesztése

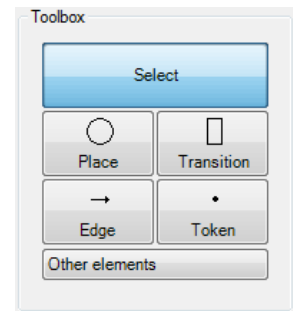
A PetriDotNet program több ablakos (MDI) szerkesztést tesz lehetővé. Azaz egyszerre több Petri-hálót nyithatunk meg és szerkeszthetünk, a hozzájuk tartozó szerkesztőablakokat pedig a *Window* paranccsal tetszés szerint váltogathatjuk, illetve egymáshoz képest függőlegesen és vízszintesen elrendezhetjük.



A szerkesztőmező (ahova a hálót rajzolhatjuk) mérete más Petri-hálós eszközöktől eltérően rögzített. Három paranccsal változtathatjuk meg: a *View » Fit to net* (Alt+F7) parancs az éppen szerkesztés alatt álló Petri-háló befoglaló méretéhez igazítja a szerkesztőmezőt, míg a

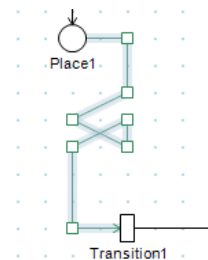
View » Grow width (Alt+Jobbra) parancs szélességében, a *View » Grow height (Alt+Le)* parancs pedig magasságában növeli meg a szerkesztőmezőt. (Ezek a parancsok az eszköztárról is elérhetők.) A szerkesztőmenő azonban automatikusan növekszik görgetés hatására, illetve ha egy elemet a szerkesztőmező szélére húzunk.

Alaphelyzetben a szerkesztő (azaz *Design*) mód van kiválasztva. Ebben a módban a képernyő bal oldalán levő oldalsávban láthatók a szerkesztési eszköztárak, ezekből választhatók ki a háló szerkesztéséhez szükséges eszközök. A *Toolbox*ban található a beszúrni kívánt elemek: hely, tranzíció, él, token, helyettesítő tranzíció (coarse transition – erről egy külön fejezetben részletesebben is írunk).



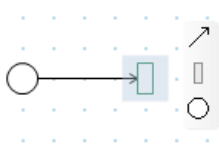
Minden szerkesztési művelethez először az adott elemet kiválasztjuk a *Toolbox*ban. Csomópontok (hely, tranzíció) rajzolásakor a kiválasztott csomópontot a szerkesztőmező megfelelő pontjára kattintva helyezhetjük le. A *Token* eszközzel a kezdő token eloszlás állítható be: az adott helyre kattintva a helyben levő tokenek száma eggyel nő.

Él behúzásához először a kezdő csomópontra, majd a vég csomópontra kell kattintani (ez kicsit eltér a más eszközökben gyakori „kezdőpontra kattintás majd húzás” művelettől). Él rajzolásakor lehetőség van töréspontok beiktatására is, ehhez a kezdő csomópontra kattintás után sorban a szerkesztőmező megfelelő pontjaira kell rákattintanunk. A készülő él a töréspontokkal együtt folyamatosan látható. Az utolsó töréspont berajzolása után kattintsunk a vég csomópontra, ezzel elkészül az él.



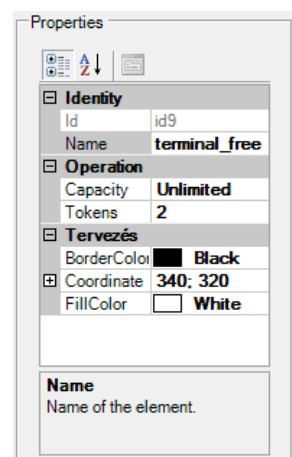
A *Select* mód kiválasztásával az egyes elemek (hely, tranzíció, él) kiválaszthatók és tulajdonságaik szerkeszthetők. Élek kiválasztásakor az élet alkotó töréspontok is megjelennek és szerkeszthetővé válnak. Ha egy töréspont rossz helyen van, akkor az él kiválasztása és a töréspontok megjelenése után az adott pontot egyszerűen húzzuk odébb. Ha egy töréspontot törölni szeretnénk, akkor ehhez a helyi menüből (az él kiválasztása után az adott töréspontra jobb egérgombbal kattintva megjelenő menüből) a *Delete edge point* parancsot válasszuk. Töréspont utólagos felvételére is lehetőség van *Select* módban: a *Ctrl* módosító gomb megnyomása mellett kattintsunk az adott él megfelelő pontjára, így ott egy új töréspont keletkezik (ezt a funkciót a kurzor megváltozása is jelzi).

Ha a beállítások közt a *View* fülön bejelöljük az *Enable edit helper tool* lehetőséget, akkor egy hely vagy tranzíció kijelölése során kapunk egy szerkesztést könnyítő eszköztárt, mely közvetlenül a kijelölt elem mellett jelenik meg.



A tranzíció vagy a hely ikonra kattintva a kijelölt helyhez kapcsolódó tranzíció ill. hely szűrhető be, az él ikonnal pedig az adott elem-elemből kiinduló él készíthető egyszerűen.

A kiválasztott elemek tulajdonságai az oldalsávon elhelyezkedő *Properties* részen szerkeszthetők. Az egyes tulajdonságokról rövid leírást kapunk a *Properties* doboz alján. Az elemek tulajdonságai szerkeszthetők a helyi menü segítségével is. Ehhez az adott elemre jobb gombbal kattintva megjelenő menüből válasszuk ki a *Properties*

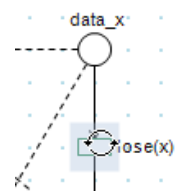


menüpontot.

Egyes tulajdonságokat anélkül meg tudunk változtatni, hogy megnyitnánk a *Properties* lapot. Ilyen lehetőség a már említett *Token* eszköz, amivel a helyekben levő tokenek számát módosíthatjuk. Ezt a lehetőséget gyorsbillentyűk segítségével is használhatjuk: *Select* módban a kívánt helyet kiválasztva a *Ctrl+Num+* és *Ctrl+Num-* billentyűkombinációkkal növelhetjük illetve csökkenthetjük a hely jelölését.

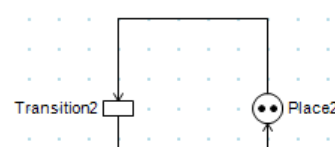
Ugyanez a két billentyűkombináció élek esetén is működik: *Select* módban a kívánt élet kiválasztva a *Ctrl+Num+* növeli, míg a *Ctrl+Num-* csökkenti az él súlyát.

Másik ilyen egyszerűsítés a tranzíciók orientációjának (függőleges vagy vízszintes irány) megváltoztatása. Ehhez a *Select* mód kiválasztása után a *Ctrl* módosító gomb megnyomása mellett kattintsunk az adott tranzícióra, aminek az iránya függőlegesből vízszintesbe vált és a következő kattintásra vissza (ezt a funkciót a kurzor megváltozása is jelzi).



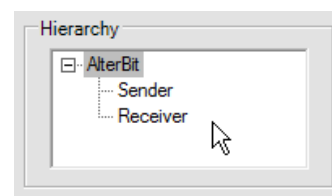
Helyek és tranzíciók létrehozásakor az új csomópont neve nem látható. Ez a beállítás az áttekinthetőséget szolgálja. A nevek megjelenítését a *Properties* szerkesztődoboz *Name Visible* tulajdonsága vezérli, ennek átállításával engedélyezhetjük. Kényelmi funkció, hogy ha megváltoztatjuk egy csomópont nevét (a *Properties* dobozban a *Name* tulajdonságnál), akkor a név kijelzése automatikusan engedélyeződik. A nevek megjelenítésének átállítását gyorsbillentyű is segíti: a *Select* mód kiválasztása után az *Alt* módosító gomb megnyomása mellett kattintsunk az adott helyre vagy tranzícióra, ennek hatására az adott csomópont neve megjelenik illetve a következő kattintásra megint eltűnik.

Helyek és tranzíciók esetén lehetőségünk van a csomópont nevének elhelyezkedését (a csomópont felett, alatt, tőle balra vagy jobbra) megváltoztatni a *Properties* dobozban a csomópont *Text Align* tulajdonságának átállításával (négy lehetőség közül választhatunk egy legördülő listából: Top, Bottom, Left, Right). Természetesen ez is elvégezhető egyszerűbben gyorsbillentyűk segítségével: *Select* módban a kívánt csomópontot kiválasztva a *Ctrl+Fel*, *Ctrl+Le*, *Ctrl+Balra*, *Ctrl+Jobbra* billentyűkombinációkkal rendre a csomópont fölé, alá, tőle balra vagy jobbra helyezhetjük át a név kijelzését.



IV. Hierarchikus hálók kezelése

Hierarchikus háló esetén a *Hierarchy* dobozban látható a háló felépítése és itt a megjelenítendő alhálót is ki lehet választani. A program ún. helyettesítő tranzíciók (coarse transition, CT) segítségével kezeli az alhálókat. A helyettesítő tranzíció lehetőséget ad arra, hogy egy modellezett tevékenységet vagy eseményt (amit eredetileg egy tranzíció modellezett) finomítsunk, azaz egy alhálóvá, egy alacsonyabb hierarchia szinten megjelenő Petri-háló részletté bontsunk ki.



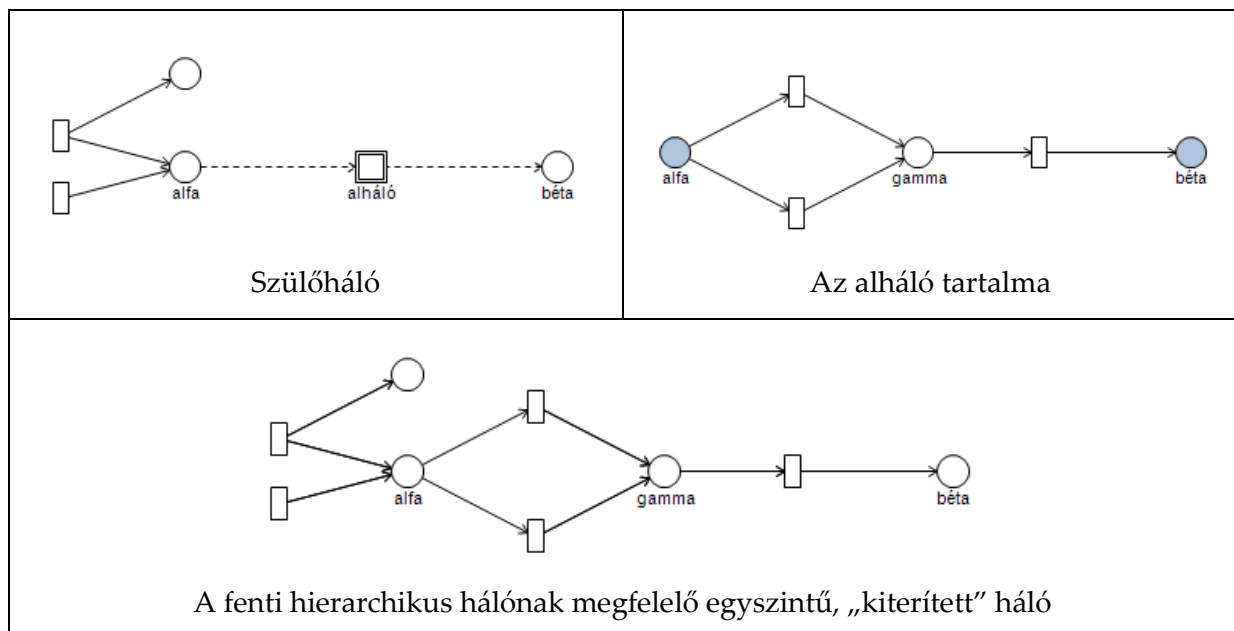
Ebből következik, hogy minden alhálót egy-egy CT reprezentál a szülőhálóban. Természetesen egy ilyen helyettesítő tranzíció „szintaktikailag” is tranzícióként viselkedik, azaz

csak helyekhez kapcsolódhat mind bemenő, mind kimenő éleivel. Ezért amikor egy élet húzunk egy szülőhálóbeli hely és a CT közé, a helyből képződik egy referenciapéldány a CT által reprezentált alhálóban. A referencialhelyek pontosan ugyanúgy viselkednek, mint szülőjük. (A referencialhelyeket a program késsel jelöli alapértelmezés szerint.)

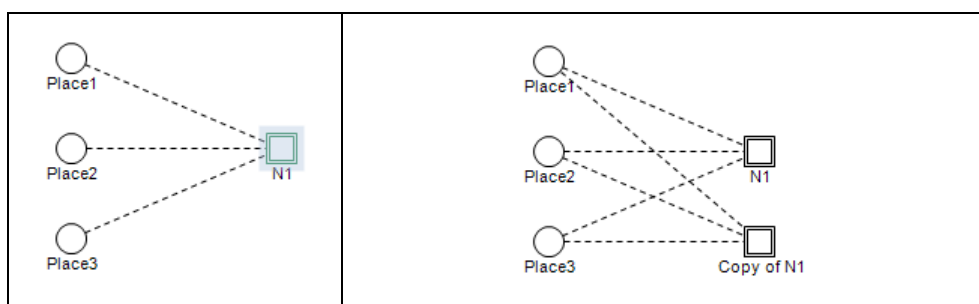
Ha egy szülőhálóbeli helyet a CT-hez kötve nem szeretnénk új referenciapéldányt létrehozni az alhálóban, hanem egy már létező alhálóbeli helyet szeretnénk referencialhelyé alakítani, akkor a szülőhálóbeli hely és a CT közti él behúzása közben tartjuk lenyomva a *Ctrl* gombot. Ennek hatására az él behúzása után egy listából kiválaszthatjuk, hogy mely alhálóbeli helyet szeretnénk referencialhelyé alakítani.

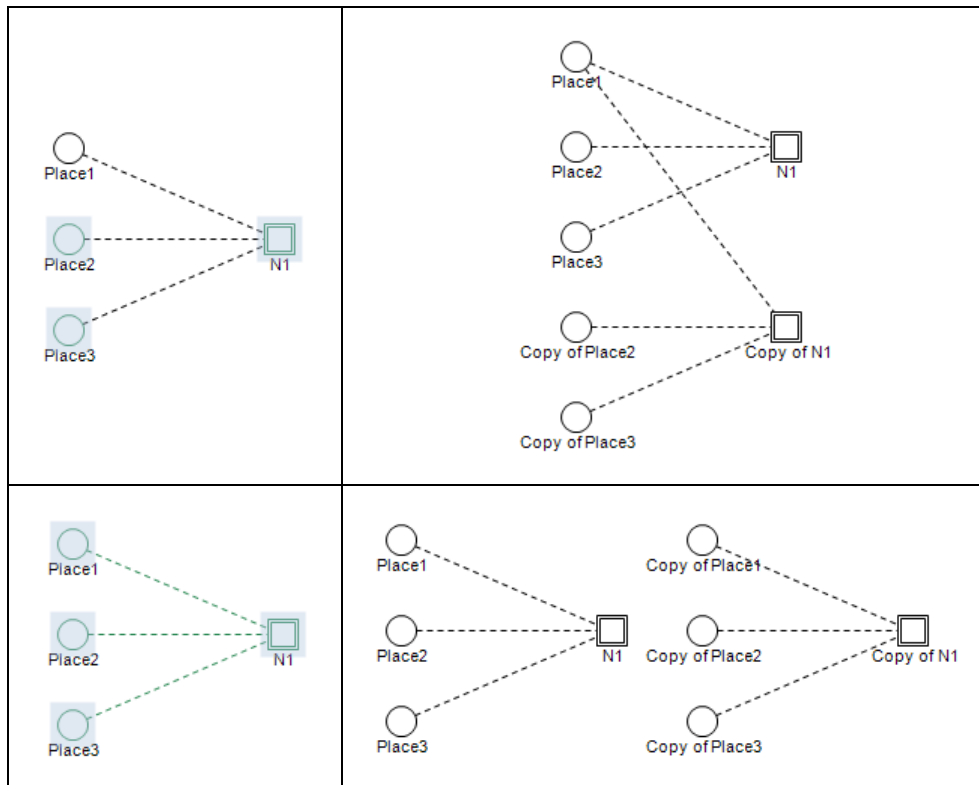
A szülőhálóban a helyet és a CT-t összekötő él csak egy kapcsolatot vizualizál, nem valódi él, ezért szaggatott vonallal van feltüntetve és nincs multiplicitása. A végén lévő nyilak jelölik, hogy az alhálóban a referencialhelyhez milyen élek kapcsolódnak.

Például ha a szülőháló és az alháló a fenti kettőnek megfelelő, akkor ez ekvivalens az alul látható hálóval.



A helyettesítő tranzíciók másolhatók. Amennyiben egy másolatot hozunk létre egy CT-ből, másolat képződik a hozzá tartozó alhálóról is. Ez az alháló ugyanazokkal a referencialhelyekkel fog rendelkezni, mint az eredeti, kivéve, ha a másolásakor a referencialhelyek szülőhelyeiből is kijelöltünk. Ilyenkor a szülőhelyek másolataira fognak az adott referencialhelyek mutatni. Ezt mutatja a következő példa is (a bal oldalon látható, hogy melyik elemek voltak a másolásnál kijelölve, a jobb oldalon pedig a beillesztés utáni eredmény látható):





V. Színezett hálók szerkesztése

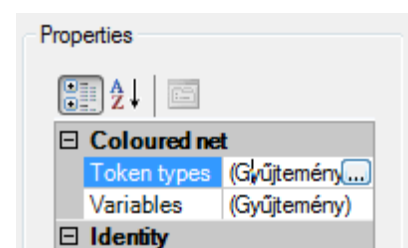
V.1. Háló szerkesztése

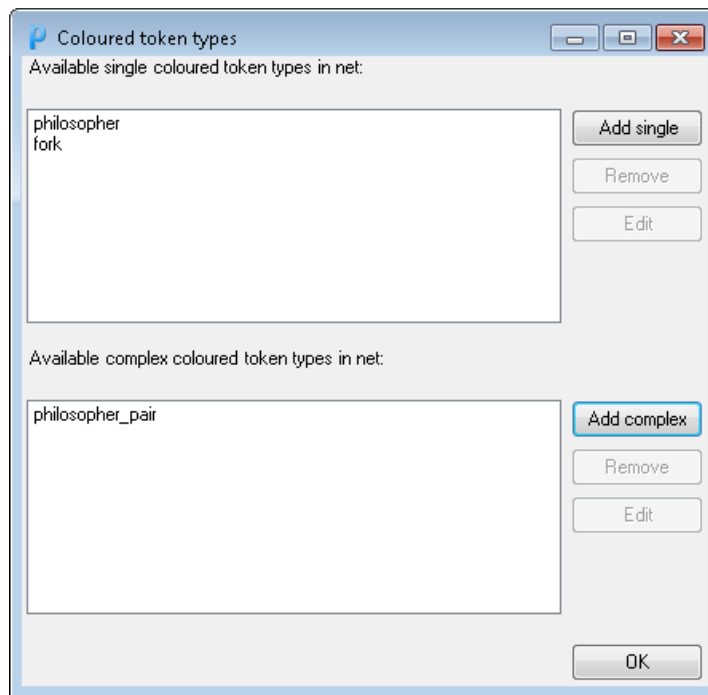
Színezett hálók szerkesztésére a színezetlen hálókhoz hasonló módon van lehetőség. Színezett hely vagy színezett tranzíció beszúrásához a *Toolbox*ban az *Other elements* menüből válasszuk ki a *Coloured Place* vagy a *Coloured Transition* opciót. A színezett helyek és tranzíciók közt ugyanúgy az *Edge* eszközzel tudunk éleket létrehozni.

V.2. Színosztályok kezelése

A színezett hálók kezeléséhez szükséges színosztályok definíciója is. Ezt a *CPN » Token types* menüpontban vagy a háló tulajdonságai közt tudjuk módosítani.

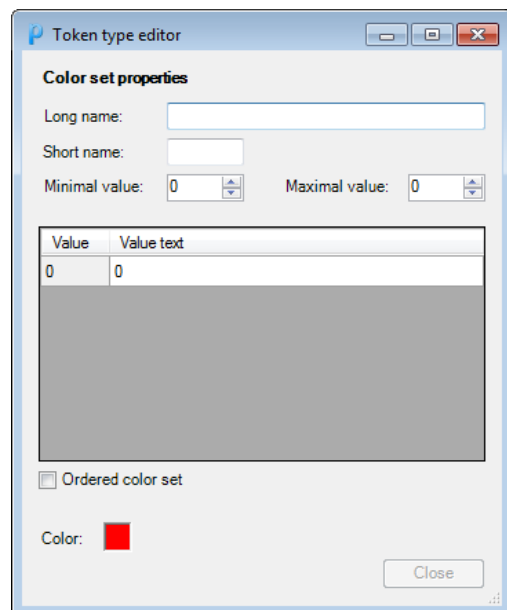
Az így megjelenő ablakban láthatók a definiált színosztályok. Kétféle színosztályt különböztetünk meg: az egyszerű (single) osztályokat és az összetett (complex) osztályokat. Az egyszerű osztályok enumerációk, az összetett osztályok pedig egyszerű osztályok Descartes-szorzataként előálló színosztályok.



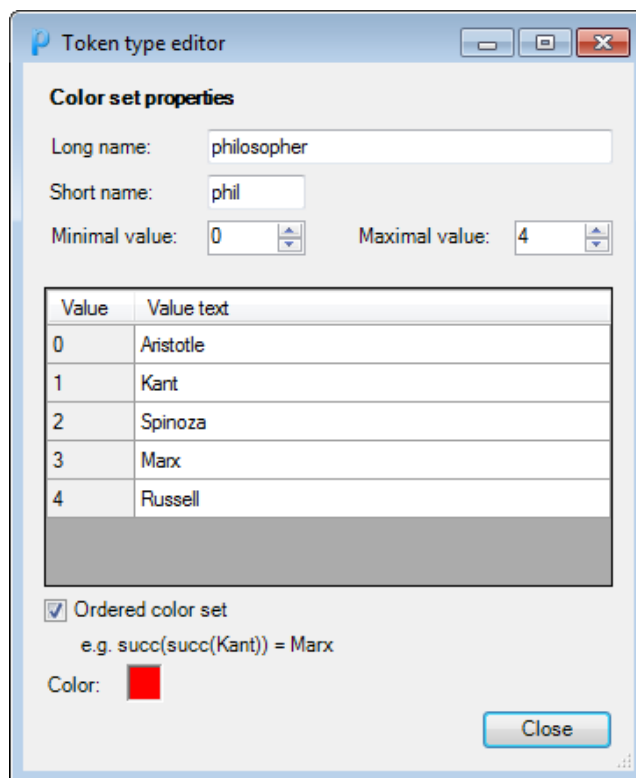


Az *Add single* gombra kattintva új színosztály hozható létre, melyhez az alábbi adatok megadása szükséges:

- Long name: A színosztály hosszú neve.
- Short name: A színosztály rövid neve (ez jelenik meg a helyek mellett szerkesztéskor).
- Minimal value: A színosztályba tartozó legkisebb érték.
- Maximal value: A színosztályba tartozó legnagyobb érték.
- Ordered color set: Bejelölése esetén a színosztály elemeire értelmezett az „előző” és a „rákövetkező” reláció.
- Color: A megjelenítéshez használt szín.



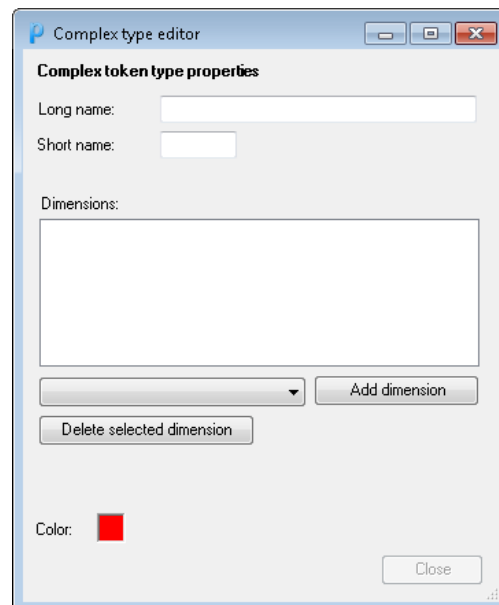
Lehetőség van továbbá a táblázatban az egyes értékeket elnevezni, így enumeráció típusú színosztályok is létrehozhatók. Például Dijkstra híres étkező filozófusainak színosztálya a következőképp néz ki:



A Coloured Token Types ablakban az Add complex gombra kattintva új összetett típus hozható létre. Ehhez az alábbi adatok megadása szükséges:

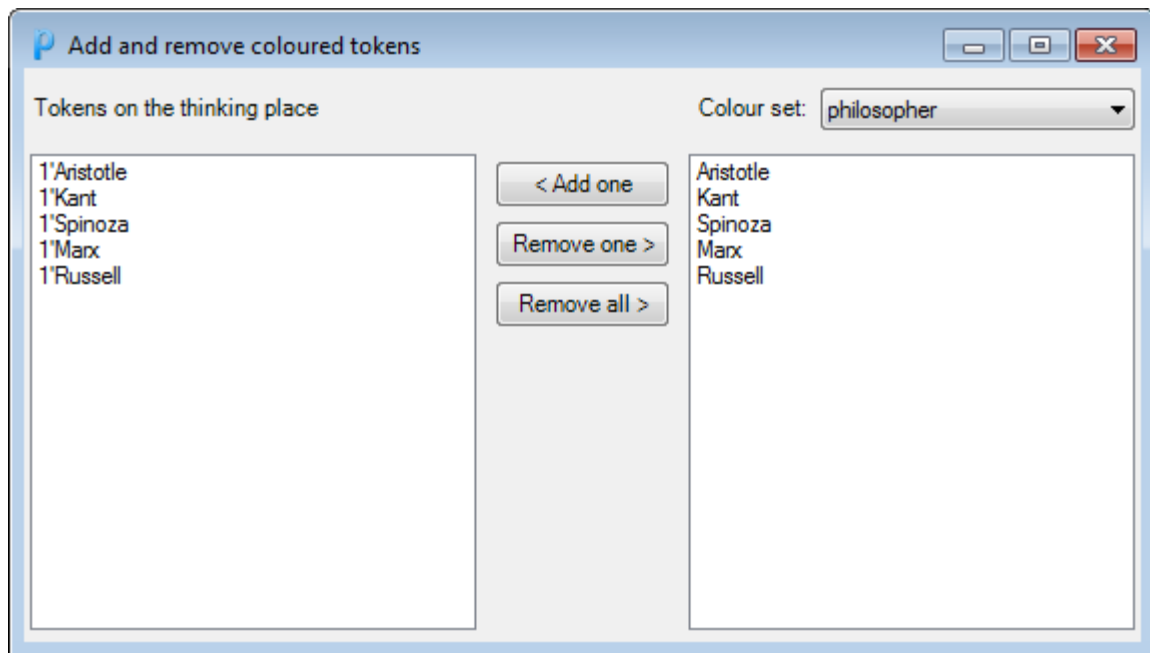
- Long name: A színosztály hosszú neve.
- Short name: A színosztály rövid neve (ez jelenik meg a helyek mellett szerkesztéskor).
- Dimensions: Az összetett színosztályt alkotó egyszerű színosztályok (sorrendezett) felsorolása. Új színosztályt az *Add dimension* gombbal lehet felvenni. Az aktuálisan kijelölt színosztályt a *Delete selected dimension* gombbal lehet kitörölni.
- Color: A megjelenítéshez használt szín.

Egy hely színosztályát egyszerűen beállítani a *Properties* » *Coloured set* részen lehet, ahol egy legördülő menüből választható ki a kívánt színosztály. Emellett a következő fejezetben ismertetésre kerülő módon, a helyen lévő tokenek szerkesztésénél is meg lehet változtatni a hely színosztályát.



V.3. Tokenek kezelése

Egy hely tokenjeinek megváltoztatására a *Properties* » *Initial tokens* részen, a ... gombra kattintva van lehetőség.



A megjelenő ablak bal oldalán felsorolásra kerülnek a kiválasztott helyen szereplő tokenek, multiplicitásukkal együtt. A jobb oldalon láthatók a hely színosztályába tartozó tokenek. Új token hozzáadásához válasszunk ki egy vagy több token-t a jobb oldali listából, majd nyomjuk meg az *< Add one* gombot. Ennek hatására az összes kijelölt tokenből 1-1 darabot adhatunk az aktuális helyhez.

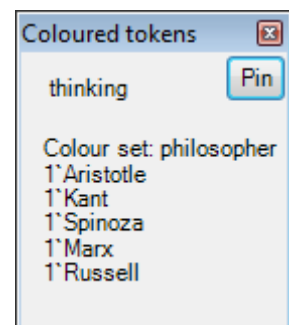
Token törléséhez a bal oldali listából válasszuk ki a törölni kívánt tokeneket. A *Remove one >* gombra kattintva a kiválasztott tokenekből egy-egy darabot törölhetünk, míg a *Remove all >* gombbal az adott token összes példányát törölhetjük a helyről.

Egy hely tokenjeit a Properties részen is megadhatjuk szövegesen *<számosság> <érték_vektor>* formában. (Az *<érték_vektor>* definícióját lásd V.4. fejezetben.) Emellett használható az *all* kifejezés is, amely a helyhez rendelt színosztály összes tokenjét jelenti.

Példák helyes tokenmegadásra: *1`Aristotle* vagy *2`[0,3]++1`[0,0]* vagy *all*

Lehetőség van ebben az ablakban a hely színosztályának megváltoztatására is a *Colour set* legördülő menüben. Figyelem, a színosztály megváltoztatása törli az adott helyen lévő összes token-t!

Az egyes helyeken lévő tokeneknek csak a számossága látható a hálóban. Bővebb információhoz válasszuk ki a *View » Show coloured tokens form* parancsot. Az így megjelenő ablakban láthatóvá válnak annak a helynek a tokenjei, amelyre az egérrel rámutatunk. Az ablak *Design* és *Simulation* módban egyaránt használható.



V.4. Élkifejezések megadása, változók kezelése

A színezett hálók dinamikus működéséhez szükséges az egyes élekre élkifejezéseket írni. Ezek a tokenek áramlását szabályozzák.

Egy kiválasztott él élkifejezését a *Properties / Arc expression* részen lehet megadni. Egy élen kizárólag változók vagy konstansok szerepelhetnek. Amennyiben több változó is szerepel, azokat ++ karakterekkel kell elválasztani. Azaz ilyen éleken az alábbi *ArcExpr* kifejezések állhatnak:

```
<ArcExpr> ::= <érték_vektor> | <ArcExpr> ++ <érték_vektor>

<érték_vektor> ::= [<érték>, <érték>, ...]
                A kapcsolódó helyhez rendelt színosztály dimenziójának megfelelő
                számú értéknek kell szerepelnie. Egyszerű színosztály esetén a szögletes
                zárójelek elhagyhatók.

<érték> ::= <változó> | <konstans>
```

Példák helyes élkifejezésre: `ph` vagy `fb++fj` vagy `[p,q]++[r,s]`

Élkifejezéseket megadása előtt állítsuk be a kapcsolódó hely színosztályát. Ellenkező esetben a program hibát jelez az élkifejezésre, bár amennyiben a szimuláció megkezdése előtt a hely színosztálya beállításra kerül, ez semmilyen gondot nem okoz.

A tokenmozgás szabályozásának másik eszköze a színezett tranzíciókhoz rendelhető őrfeltétel, melyet a *Properties / Guard* részen lehet megadni. Ennek formája a következő:

```
<Guard> ::= <kifejezés> | <Guard> <bool_operátor> <Guard>

<bool_operátor> ::= '&&' | '||'

<relációs_jel> ::= > | < | >= | <= | = | != | <>

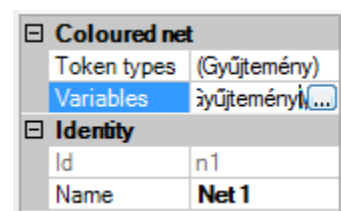
<kifejezés> ::= <érték> <relációs_jel> <egész_számb> |
               <érték> <relációs_jel> <érték>

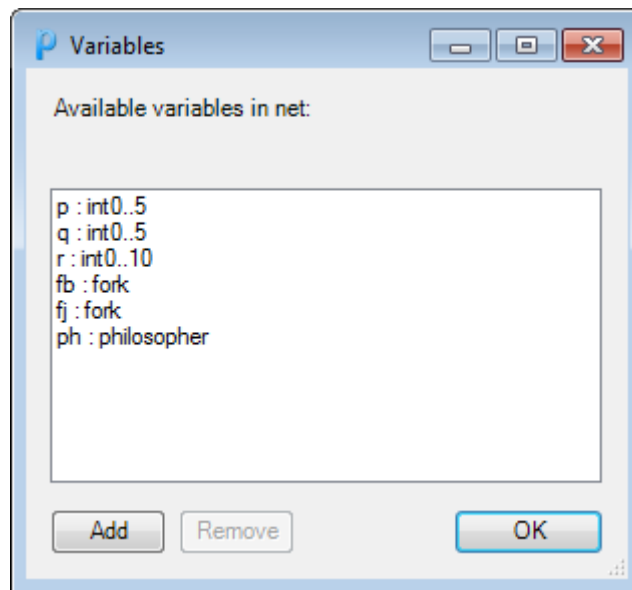
<érték> ::= <változó> | succ <változó>
```

Példák helyes őrfeltételre: `fb=succ fj` vagy `fb > 2` vagy `fb>2 && fj<1`

A kifejezésekben szereplő változók típusát előre definiálni kell. Ezt a CPN » *Variables* menüpontra kattintva vagy a háló tulajdonságainál lehet megtenni.

A megjelenő ablakban felsorolásra kerülnek a definiált változók típusukkal együtt. Ezeket szerkeszteni (*Edit*) és törölni (*Remove*) lehet, illetve lehetőség van új változó felvételére is (*Add*). Változók kizárólag egyszerű színosztályokhoz vehetők fel.





V.5. Ismert limitációk

A színezett hálók kezelése fejlesztés alatt áll. Az alábbi funkciók hiányossága ismert, implementálásuk folyamatban van.

- Jelenleg nem lehet színezett hierarchikus hálót létrehozni.
- Nem köthetők össze színezetlen helyek és színezett tranzíciók.

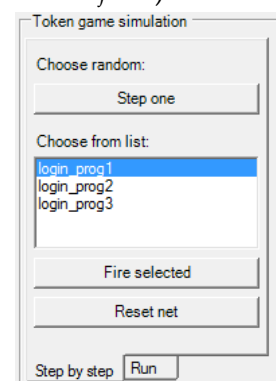
VI. Petri-hálók ellenőrzése

Petri-hálók ellenőrzésére alapvetően kétféle lehetőséget kínálkozik: a szimuláció (token játék) és a háló analízise (dinamikus/viselkedési tulajdonságok vizsgálata, invariánsok számítása, strukturális tulajdonságok felderítése, stb.). Ezek közül beépítve a PetriDotNet program a szimuláció eszközt kínálja fel, az analízis elvégzésére pedig a külön telepíthető kiterjesztések (add-in) támogatásával ad lehetőséget.

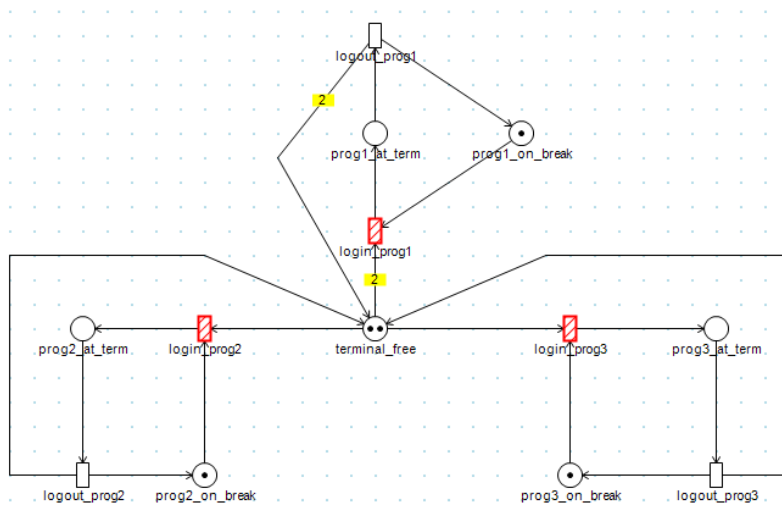
VI.1. Szimuláció, token játék

Szimuláció módban lehet a háló működését ellenőrizni. A szimuláció a programban két-féle formában érhető el: az *interaktív, lépésenkénti szimuláció* (más néven *token játék*) és a *nem interaktív szimuláció* avagy *futtatás*.

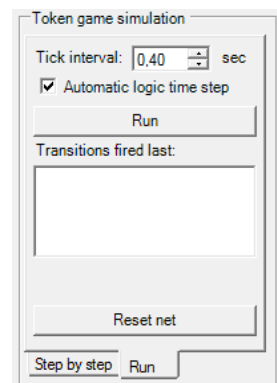
Token játék esetén (*Step by step* üzemmód) az adott állapotban engedélyezett tranzíciók a jó láthatóság kedvéért piros kerettel és piros sraffozással vannak jelölve. Ebben a lépésenkénti szimulációs módban egyesével tüzelhetjük az engedélyezett tranzíciókat a kiválasztott tranzícióra kattintva. Ugyanezt érhetjük el a kívánt tranzíció nevét a lenti *Choose from list:* listából kiválasztva és a *Fire selected* gombra kattintva. A *Step one* gombbal véletlenszerűen választunk egyet tüzelésre az engedélyezett tranzíciók közül. A *Reset net* gomb



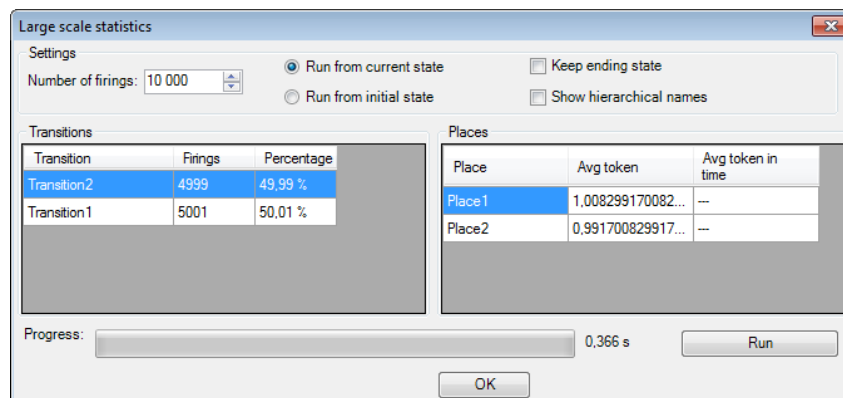
alaphelyzetbe állítja a hálót, azaz visszaállítja az a kezdő tokeneloszlást (állapotot), ami a szimuláció indításakor volt érvényes.



A szimulációs szerkesztődoboz alsó *Run* fülével válthatunk nem interaktív vagy futtatási üzemmódba. Ebben a módban (adott időközönként) ismétlődő tüzelést indíthatunk a *Run* gombbal. A háló a *Tick interval*-ban beállított időközönként lép a következő állapotba (azaz tüzel egyet). A konfliktusokat a program véletlen választással oldja fel, tehát ebben a módban az állapottér egy trajektóriáját járjuk be. A *Transitions fired last* mezőben az utolsó néhány lépésben tüzelt tranzíciók nevei láthatók. Ha az *Automatic logic time step* be van jelölve, a hálóbeli idő úgy működik, mint egy logikai óra. Ha nincs bejelölve, akkor a hálóbeli idő a *Tick interval*-ban beállított időközönként halad előre.



A *Large scale statistics* nevű kiegészítő (add-in) beépülő modul segítségével szintén nem interaktív szimulációt futtathatunk a hálón, majd a futás eredményei alapján készült statisztikát láthatjuk a modul ablakában:



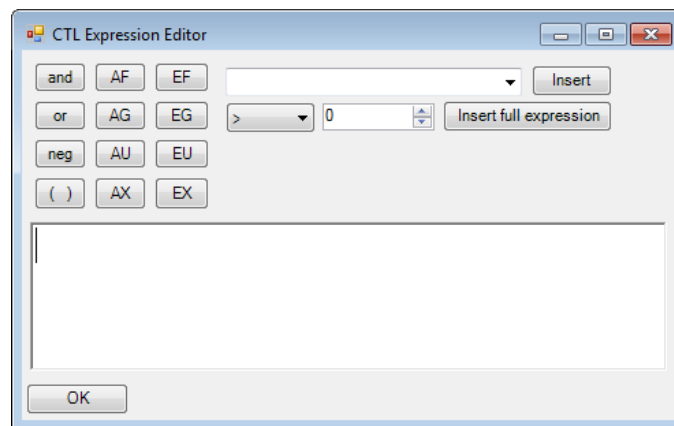
VI.2. Analízis

A hálók analízisére két lehetőség van. Egyrészt használható a berlini Humbolt Egyetem Informatikai Intézete által fejlesztett INA (Integrated Net Analyzer, honlapja:

<http://www2.informatik.hu-berlin.de/lehrstuehle/automaten/ina/>) program. Ez .pnt formátumú fájl bemenetet vár, amelybe az aktuális hálót a PetriDotNet exportálni tudja az *Add-in » Export » .pnt export* menüpont segítségével.

Másrészt a programhoz mellékelve vannak bizonyos analízis modulok. Az *Add-in » Analysis » Invariant analysis* menüpont segítségével **P-invariánsok** és **T-invariánsok** kereshek a Martinez-Silva algoritmus segítségével.

Ezen kívül lehetőség van **modellellenőrzésre** az *Add-in » Analysis » Model checking* menüpontban, *amennyiben ez a beépülő modul külön telepítésre került*. Ebben a modulban az állapottér szimbolikus feltérképezése után egy CTL kifejezés adható meg az alábbi ablakban. Ezt a program kiértékeli és igaz/hamis választ ad.



A kifejezésekben CTL temporális operátorok és a Petri-háló helyeinek tokenszámaira felírt kifejezések szerepelhetnek. Ezek tetszőlegesen zárójelezhetők, összekapcsolhatók ÉS és VAGY operátorokkal, valamint egymásba ágyazhatók.

Az érvényes kifejezések formális definíciója:

```

<CTL_kifejezés> ::= <operátor>(<kifejezés>) |
                  <bin_operátor>(<kifejezés> <bin_elválasztó> <kifejezés>)

<kifejezés>      ::= <CTL_kifejezés> |
                  <hely_kifejezés> |
                  !<kifejezés> |
                  <kifejezés> <bool_operátor> <kifejezés> |

<operátor>       ::= EX | EF | EG | AX | AF | AG

<bin_operátor>   ::= EU | AU | EW | AW | ER | AR

<bin_elválasztó> ::= u | w | r

<hely_kifejezés> ::= <hely_ID> <relációs_jel> <egész_számb>

<relációs_jel>   ::= > | < | >= | <= | = | != | <>

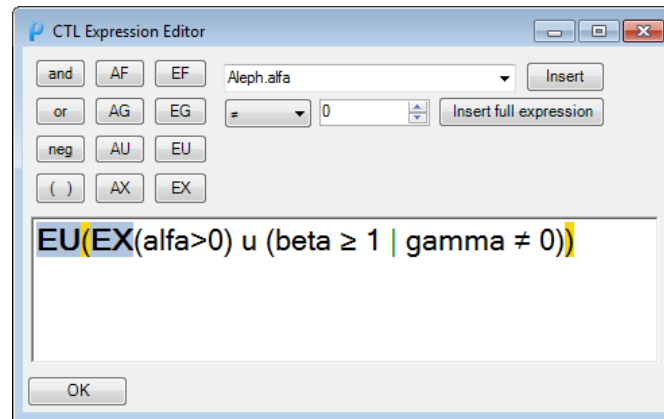
<bool_operátor>  ::= '&' | '|'

```

A <hely_ID> egy Petri-hálóbeli hely ID-ja vagy neve. Ezek az ablak tetején látható listából kiválaszthatók és az *Insert* gombbal be is szúrhatók a kurzor aktuális pozíciójába. Amennyiben egy relációs jelet és egy számot is kiválaszt, majd az *Insert full expression* gomb-

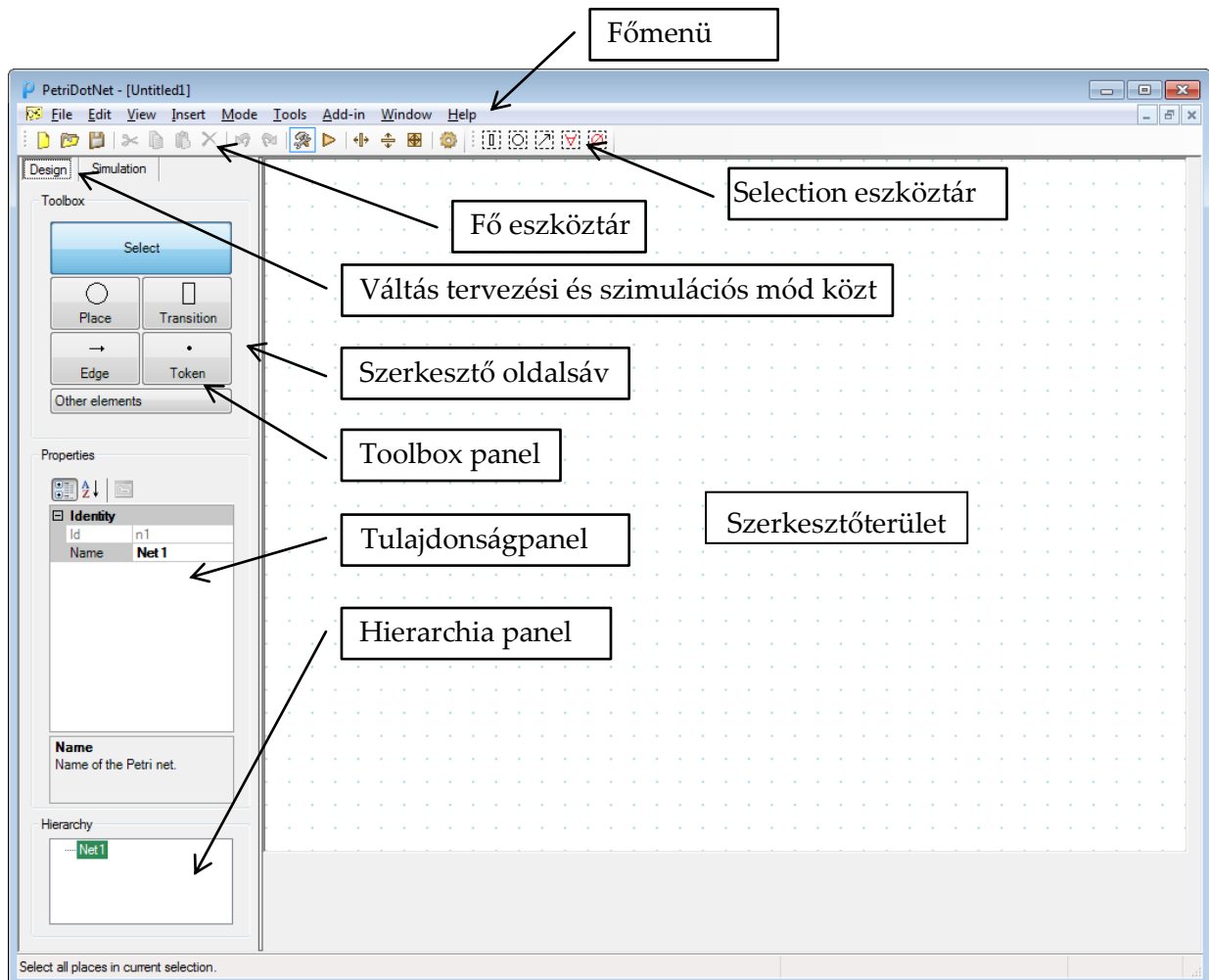
ra kattint, egy teljes <hely_kifejezés> szúrható be a kifejezésbe. A szerkesztést szintaxiskiemelés is segíti.

Egy érvényes kifejezés látható a következő képernyőképen:

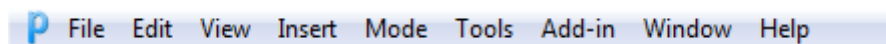


VII. A kezelői felület részletes leírása

VII.1. A PetriDotNet főablakának felépítése



VII.2. Főmenü



A menü felépítése

– File

- New: Új hálót hoz létre.
- Open: Megnyit egy hálót fájlból.
- Close: Bezárja az aktuális hálót.
- Save: Menti az aktuális hálót.
- Save As: Menti az aktuális hálót más néven vagy más formátumban.
- Save to bitmap: Elmenti a hálót .png formátumban.

- Save to SVG: Elmenti a hálót vektorgrafikus (Scalable Vector Graphics, .svg) formátumban.
- Exit: Bezárja a programot.
- Edit
 - Undo: Visszavonja az utolsó műveletet.
 - Redo: Megismétli az utolsó visszavont műveletet.
 - Cut: Kivágja a háló kijelölt elemeit.
 - Copy: Kimásolja a háló kijelölt elemeit.
 - Paste: Beilleszti a vágólapra másolt elemeket.
 - Delete: Kitörli a háló kijelölt elemeit.
 - Select All: Kijelöli a háló összes elemét.
- View – ebben a menüben lehet az oldalsávban levő szerkesztődobozok láthatóságát és a szerkesztőmező méretét beállítani
 - Toolbar: Mutatja vagy elrejtja a fő eszköztárat
 - Selection toolbar: Mutatja vagy elrejtja a Selection eszköztárat
 - Status bar: Mutatja vagy elrejtja a státuszsort
 - Toolbox: Mutatja vagy elrejtja a Toolbox-ot
 - Property pane: Mutatja vagy elrejtja a Properties dobozt
 - Token game pane: Mutatja vagy elrejtja a tokenjáték dobozát
 - Overview: Mutatja vagy elrejtja a háló áttekintő képét
 - Design area size: Beállítja a háló alapterületét.
 - Grow width: Megnöveli a háló alapterületét vízszintes irányban.
 - Grow height: Megnöveli a háló alapterületét függőleges irányban.
 - Fit to net: A háló alapterületét az elemekhez igazítja.
- Insert
 - Place: Beszúr egy helyet a következő egérekattintás pozíciójába.
 - Transition: Beszúr egy tranzíciót a következő egérekattintás pozíciójába.
 - Edge: Beszúr egy élet a következő két egérekattintással kijelölt elem közé.
 - Token: Eggyel növeli az egérekattintással kijelölt helyen levő tokenek számát.
 - Default cursor: Kijelölés módra vált.
- CPN
 - Token types: A színezett háló színosztályai adhatók meg.
 - Variables: A színezett hálóban használható változók adhatók meg.
- Mode
 - Design mode: Szerkesztés módba vált.
 - Token game simulation: Token játék (szimuláció) módba vált.
- Tools
 - Options: A program működését itt állíthatja be.

- Add-in – ebben a menüben jelennek meg a program kiegészítői, amelyek a program melletti *add-in* könyvtárban találhatók.
- Window – ebben a menüben találhatók a megnyitott hálók ablakainak átrendezésére szolgáló menüpontok
- Help
 - Send error feedback: Visszajelzést küldhet egy hibáról.
 - About: Itt tájékozódhat a program verziószámáról.

VII.3. Fő eszköztár



	Új
	Megnyitás
	Mentés
	Kivágás
	Másolás
	Beillesztés
	Törlés
	Visszavonás
	Újra
	Szerkesztés módba váltás
	Token játék (szimuláció) módba váltás
	Megnöveli a háló alapterületét vízszintes irányban.
	Megnöveli a háló alapterületét vízszintes irányban.
	A háló alapterületét az elemekhez igazítja.
	Beállítások

VII.4. Selection eszköztár



	Az aktuális kijelölésből kiválasztja a tranzíciókat.
	Az aktuális kijelölésből kiválasztja a helyeket.
	Az aktuális kijelölésből kiválasztja az éleket.
	Kijelöli az aktuális háló összes elemét.
	Megszünteti a kijelölést.

VII.5. Tulajdonságpanel (Property pane)

A tulajdonságpanelen az alábbi tulajdonságok állíthatók be az egyes elemekhez:

	Háló				Tulajdonság leírása
BorderColor		+	+	+	Az elem körvonalának színe.
Capacity		+			A hely kapacitása.
Coordinate		+	+		Az elem pozíciója
Distribution			+		A tranzíció tüzeléséhez rendelt valószínűségi eloszlása.
FillColor		+	+	+	Az elem kitöltőszíne.
Id	+	+	+	+	Az elem egyedi azonosítója.
Name	+	+	+		Az elem neve.
Name visible		+	+		A csomópont nevének láthatósága.
Orientation			+		A tranzíció iránya (függőleges vagy vízszintes).
Points				+	Az él közbenső pontjainak koordinátája.
Priority			+		A tranzíció prioritása. (Alapértelmezés szerint 100, minél nagyobb szám, annál előbb tüzel.)
Prohibitive				+	True, ha az él tiltó él.
Text align		+	+		A csomópont nevének elhelyezkedése.
Tokens		+			A hely tokenszáma a szimuláció kezdetén.
Weight				+	Az él súlya (multiplicitása).

Az elemekre jobb gombbal kattintva előhívható tulajdonságablakban ugyanezen tulajdonságok állíthatók be. Ez az ablak a *Pin* gombra kattintva rögzíthető a kényelmesebb szerkesztés érdekében. Többszörös kijelölésnél azok a tulajdonságok láthatók, mellyel az összes kijelölt elem rendelkezik.

VII.6. Beállítások

A beállításokban a program számos tulajdonsága testre szabható. A beállítások automatikusan mentésre kerülnek.

– Editor

- Show element names: Ki- vagy bekapcsolható az elemek nevének kiírása.

- Edge sections arrow cap: Beállítható, hogy a törtvonalú éleknél minden töréspontnál legyen nyílhegy vagy sem.
 - More visible selection: Ki- vagy bekapcsolható a jobb láthatóságú kijelölés. Bekapcsolva az elemek háttére lesz színes kijelöléskor, kikapcsolva csak a körvonaluk színeződik át.
 - Edge arrow size: Beállítható, hogy mekkora legyen a nyílhegyek mérete.
 - Align to grid: Ki- vagy bekapcsolható, hogy az elemek illeszkedjenek-e a segédhálózathoz.
 - Show grid: Ki- vagy bekapcsolható a segédháló megjelenítése.
 - Grid style: Beállítható a segédháló típusa (négyzetrács vagy pontok).
 - Grid width: Beállítható, hogy milyen távol legyenek egymástól a segédháló pontjai vízszintesen.
 - Grid height: Beállítható, hogy milyen távol legyenek egymástól a segédháló pontjai függőlegesen.
 - Add multiple elements: Beállítható, hogy egy elem beszúrása után maradjon-e az editor beszúrás módban vagy visszatérjen kijelölés módra.
 - Default transition orientation: Beállítható, hogy egy beszúrt tranzíció alapértelmezés szerint függőleges vagy vízszintes irányú legyen.
 - Cut/copy edges'connecting nodes also: Ki- vagy bekapcsolható, hogy mi történjen élek másolásakor (másolódjanak-e a végpontok is).
 - Delete edges becoming orphan also: Ki- vagy bekapcsolható, hogy mi történjen élek végpontjának törlésekor.
- View
 - Desing area: Itt beállítható, hogy a szerkesztőterület alapértelmezett mérete mekkora legyen, illetve mekkora egységekben lehessen növelni.
 - Panels: Itt beállítható, hogy alapértelmezés szerint mely panelok legyenek láthatók a program indulásakor.
 - Token game
 - Default tick interval: Beállítható a folyamatos tüzelés esetén a két tüzelés közt eltelt időintervallum alapértelmezett hossza.
 - Default token game mode: Beállítható, hogy alapértelmezés szerint melyik tokenjáték mód legyen aktív.
 - Disk, memory
 - Default file type: Megadható, hogy alapértelmezés szerint bináris vagy XML formátumba mentsen a program.
 - Memory: Itt korlátozható a visszavonás és újra műveletek száma.

VIII. Add-inek

A program támogatja add-inek (kiegészítő modulok) használatát. Az ilyen kiegészítőket a program add-in könyvtárában kell elhelyezni.

IX. Hibák bejelentése

A program folyamatos fejlesztés alatt áll, így sajnos hibák is előfordulhatnak benne. Köszönettel vesszük, ha a felmerülő hibákat jelzi a program *Help » Send error feedback* felületén keresztül vagy a *petridotnet@inf.mit.bme.hu* e-mail címre. Ugyanerre az e-mail címre várjuk a javaslatokat, ötleteket is.

Tartalom

I. Bevezető.....	1
II. Rendszerkövetelmények.....	1
III. A program használata.....	1
III.1. Indítás.....	1
III.2. Betöltés, mentés	2
III.3. Petri-hálók létrehozása, szerkesztése.....	2
IV. Hierarchikus hálók kezelése	4
V. Színezett hálók szerkesztése	6
V.1. Háló szerkesztése.....	6
V.2. Színosztályok kezelése	6
V.3. Tokenek kezelése	8
V.4. Élkifejezések megadása, változók kezelése.....	9
V.5. Ismert limitációk	11
VI. Petri-hálók ellenőrzése	11
VI.1. Szimuláció, token játék.....	11
VI.2. Analízis	12
VII. A kezelői felület részletes leírása.....	15
VII.1. A PetriDotNet főablakának felépítése.....	15
VII.2. Főmenü.....	15
VII.3. Fő eszköztár	17
VII.4. Selection eszköztár.....	18
VII.5. Tulajdonságpanel (Property pane).....	18
VII.6. Beállítások	18
VIII. Add-inek	20
IX. Hibák bejelentése	20