

Discrete and continuous dynamic systems

PIPE

Petri net editor and analysis tool

Katalin Hangos, Anna Ibolya Pózna

University of Pannonia
Faculty of Information Technology
Department of Electrical Engineering and Information Systems
`pozna.anna@virt.uni-pannon.hu`

March 2020

PIPE - Platform Independent Petri Net Editor

- Source
 - <https://sourceforge.net/projects/pipe2/files/PIPEv4/PIPEv4.3.0/> (latest stable version)
- Constructing Petri nets
 - simple Petri nets
 - places with capacity
 - arc weights, inhibitor arcs
 - timed, prioritized transitions
 - "colored" tokens - NOT CPN!
- Simulation
 -
- Analysis
 - Incidence matrix, invariant analysis
 - Reachability graph
 - State space analysis (boundedness, safeness, deadlocks)
- Documentation:
http://sarahtattersall.github.io/PIPE/user_guide.html

User interface

PIPE: Platform Independent Petri Net Editor v4.3.0: Petri net 1

File Edit View Draw Animate Help

Analysis Module Manager

- Available Modules
 - GSPN Analysis
 - Performance Query Editor
 - Invariant Analysis
 - Minimal Siphons And Minimal Traps
 - Comparison
 - Steady State Analysis
 - Reachability/Coverability Graph
 - Simulation
 - State Space Analysis
 - Tagged Net Converter
 - Classification
 - Response Time Analysis
 - Incidence & Marking
 - Find IModule

Petri net 1

simulation mode

add place

add arcs (normal, inhibitor)

add/delete tokens

add transitions (immediate, timed)

Incidence matrix, place and transition invariants

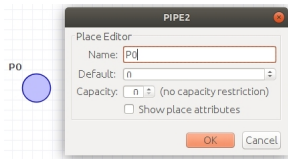
Compute reachability/coverability graph

Boundedness, safeness, deadlock

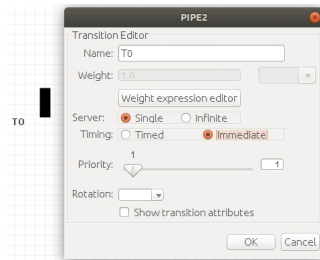
Backward/forward incidence matrix, marking vector

Select Mode: Click/drag to select objects; drag to move them

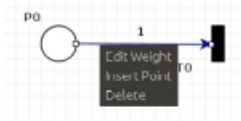
Constructing a Petri net



properties of a place:
name, default tokens, capacity



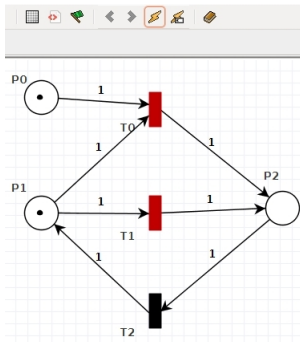
properties of a transition:
name, (weight), timing, priority



properties of an arc:
weight, bend points

Simulation

- Enabled transitions are highlighted with red color
- Simulation modes:
 - manually fire a transition by clicking on it
 - randomly fire a transition
 - fire a given number of transitions



Consider the example on the previous slide:

- Invariants

The screenshot shows a window titled "Invariant Analysis". At the top, there is a "Source net" section with a checked box for "Use current net" and a "Filename:" field with a "Browse" button. Below this is a "Results" section containing the following text:

Petri net invariant analysis results

T-Invariants

T0	T1	T2
0	1	1

The net is not covered by positive T-Invariants, therefore we do not know if it is bounded and live.

P-Invariants

P0	P1	P2
0	1	1

The net is not covered by positive P-Invariants, therefore we do not know if it is bounded.

P-Invariant equations

$$M(P1) + M(P2) = 1$$

Analysis time: 0.001s

At the bottom of the window, there are "Copy" and "Save" buttons, and a prominent orange "Analyse" button.

- Reachability graph

Reachability/Coverability Graph

Source net

Use current net Filename: Browse

Results

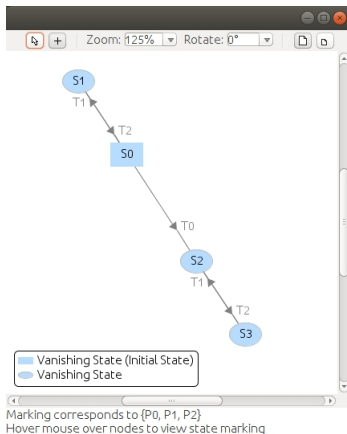
Reachability/Coverability Graph Results

Generating Reachability graph took 0.105s
Constructing it took 0.294s
Total time was 0.399s

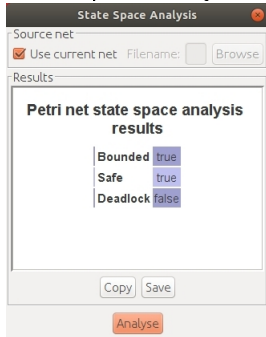
Copy Save

Generate Reachability/Coverability Graph

Display initial state(S0) in a different shape



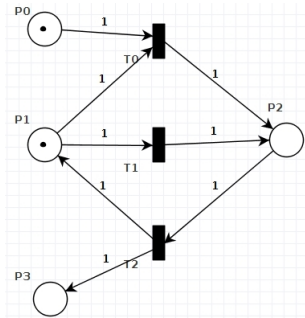
- State space analysis



- Conservativity? Liveness?

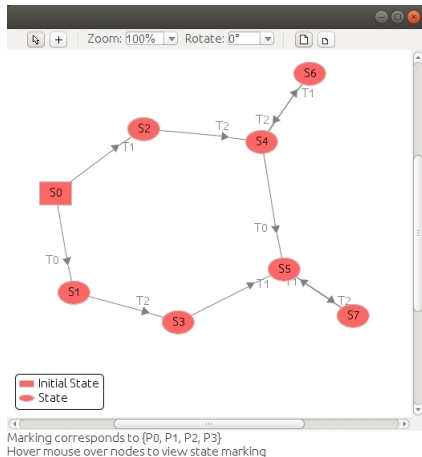
Bugs... I

Consider the following unbounded net:



- Generate the reachability/coverability graph

Bugs... II



- Hover the mouse over the nodes. The unbounded places have large number of tokens at the beginning...

Task

- Construct the Petri net of the coffee automaton, given in the tutorials. Take care of the place capacities!
- Simulate the model! Try different initial markings!
- Analyse the model (boundedness, safety, deadlock, reachability graph)