



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Факультет компьютерных наук
Департамент программной инженерии
Курсовая работа
Программа поиска ловушек сетей Петри
логическими методами

Выполнил студент группы 151 ПИ
Абдукеримов Тимур Русланович
Научный руководитель:

Старший преподаватель департамента программной инженерии
Дворянский Леонид Владимирович

Предметная область

Сети Петри – популярный математический аппарат для моделирования математических дискретных систем.

Постановка задачи

Создать программу, которая позволит находить все ловушки сети Петри, используя логические методы.

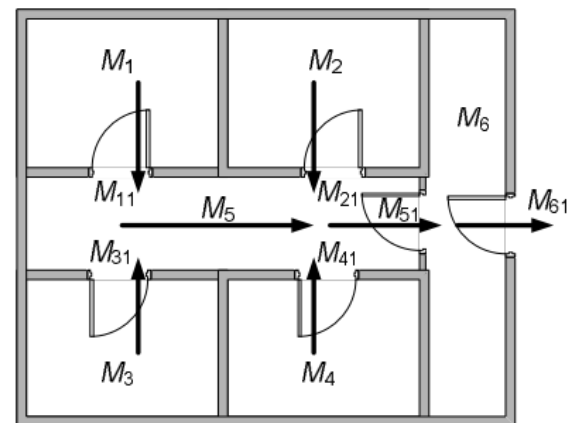


Рис. 1. Пример плана эвакуации из здания

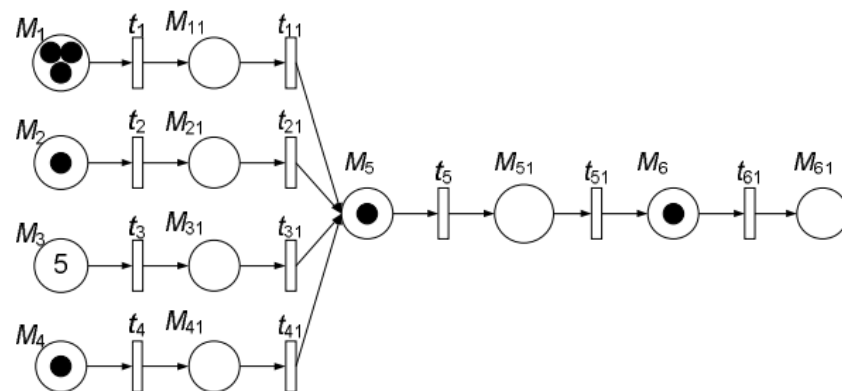


Рис. 2. Пример потоковой сети на основе плана эвакуации

Основные определения

Ловушка – Набор позиций такой, что каждый переход, который берет токен из набора, возвращает его обратно.

Это означает, что если ловушка получит токен, то в одной из позиций набора всегда будет находиться токен.

Булева функция $g(x_1, \dots, x_n)$ называется импликантой булевой функции $f(x_1, \dots, x_n)$, если для любого набора переменных, на котором $g=1$, справедливо $f=1$.

Импликанта g булевой функции f , являющаяся элементарной конъюнкцией, называется простой, если никакая часть импликанты g не является импликантой функции f .

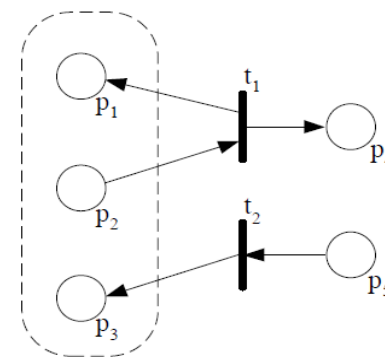


Fig. 3. Example of a trap $Q = \{p_1, p_2, p_3\}$,
 $Q^\bullet = \{t_1\}$, ${}^\bullet Q = \{t_1, t_2\}$.

Обоснование актуальности работы

В наши дни существуют технологии для разработки сети малых систем, способных взаимодействовать друг с другом в заданной области и решать глобальные задачи совместными усилиями.

Так как поиск ловушек имеет очень значимую роль в анализе, то программа по нахождению всех ловушек принесет значительную пользу многоагентной системе анализа сетей Петри.

Theorem 6 (Trap Theorem). Let N be an elementary system net with an initially marked trap $Q = \{q_1, \dots, q_r\}$. Then the following inequality holds in N :

$$q_1 + \dots + q_r \geq 1. \quad (4)$$

Цели и задачи работы

Цель работы:

Создать программу, которая будет осуществлять поиск ловушек сети Петри построенной пользователем или открытой из файла.

Задачи:

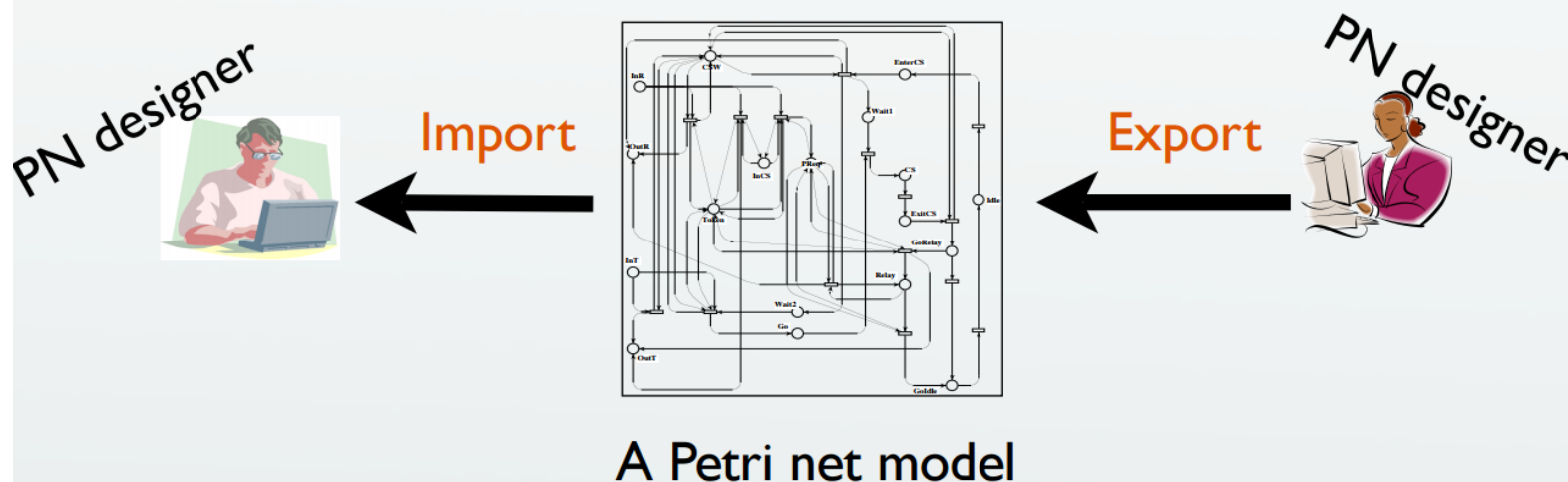
- 1) Создание и редактирование новой сети Петри используя панель инструментов;
- 2) Открытие ранее построенной сети Петри из файла;
- 3) Сохранение построенной или измененной сети Петри в исходный или новый файл;
- 4) Нахождение всех ловушек данной сети Петри;
- 5) Обеспечить наглядное визуальное выделение набора ловушек в пользовательском интерфейсе;
- 6) Импортирование сетей Петри из других редакторов(PNML);

Формат .pnml

Petri Net Markup Language (PNML) – представляет собой формат обмена, направленный на предоставление разработчикам и пользователям возможности обмена готовыми моделями сетей.

Основы и методики данного формата находятся в открытом доступе по официальной ссылке:

<http://www.pnml.org/>



Язык C#. Среда программирования: Visual Studio 2015



Перевод сети в логическую формулу:

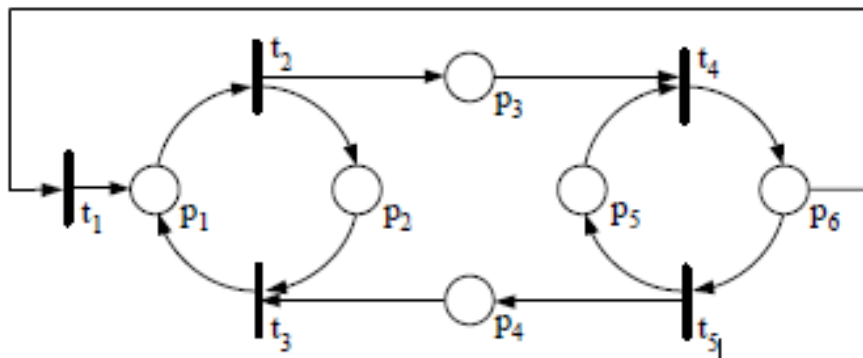


Fig. 6. Petri net (Zakrevskij, 1999).

$$(x_6 \rightarrow x_1)(x_1 \rightarrow x_2 \vee x_3)(x_2 \vee x_4 \rightarrow x_1)(x_3 \vee x_5 \rightarrow x_6)(x_6 \rightarrow x_4 \vee x_5)$$

$$(\overline{x_6} \vee x_1)(\overline{x_1} \vee x_2 \vee x_3)(\overline{x_2} \vee x_1)(\overline{x_4} \vee x_1)(\overline{x_3} \vee x_6)(\overline{x_5} \vee x_6)(\overline{x_6} \vee x_5 \vee x_4)$$

Описание алгоритма: Thelen's Prime Implicant Algorithm

- Основан на построении поискового дерева, каждый уровень которого соответствует выражению КНФ. А ребра выходящие из узлов соответствуют константам дизъюнкции.
- При поиске используют алгоритм поиска в глубину.
- Для минимизации дерева существуют три правила:
- 1. Дуга отсекается, если предшествующий ей узел имеет дополнение константы.
- 2. Дизъюнкция отбрасывается, если содержит константу, которая появляется в предшествующем узле.
- 3. Дуга отсекается, если на более высоком уровне есть еще не раскрытая дуга с той же константой.

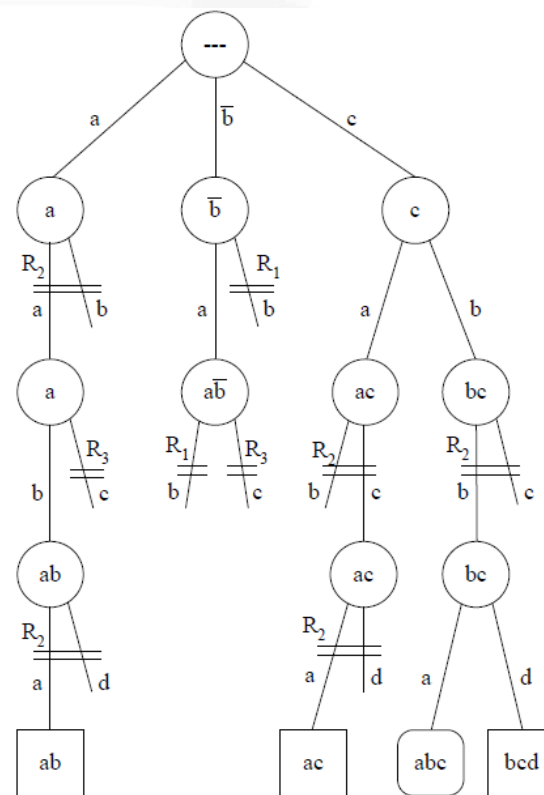


Fig. 4. Tree example $f = (a \vee \bar{b} \vee c) \wedge (a \vee b) \wedge (b \vee c) \wedge (a \vee d)$.

$$(\overline{x_6} \vee x_1)(\overline{x_1} \vee x_2 \vee x_3)(\overline{x_2} \vee x_1)(\overline{x_4} \vee x_1)(\overline{x_3} \vee x_6)(\overline{x_5} \vee x_6)(\overline{x_6} \vee x_5 \vee x_4)$$

- **Heuristic 1. (Sort by Length)**

Использование дизъюнкций с наименьшим числом констант.

$$(\overline{x_6} \vee x_1)(\overline{x_2} \vee x_1)(\overline{x_4} \vee x_1)(\overline{x_3} \vee x_6)(\overline{x_5} \vee x_6)(\overline{x_1} \vee x_2 \vee x_3)(\overline{x_6} \vee x_5 \vee x_4)$$

- **Heuristic 2. (Sort by Variables)**

Выбор дизъюнкций с наименьшим количеством констант, которые не появлялись в выбранных ранее дизъюнкциях.

$$(\overline{x_6} \vee x_1)(\overline{x_2} \vee x_1)(\overline{x_4} \vee x_1)(\overline{x_3} \vee x_6)(\overline{x_5} \vee x_6)(\overline{x_1} \vee x_2 \vee x_3)(\overline{x_6} \vee x_5 \vee x_4)$$

- **Heuristic 3. (Reordering Literals)**

Деление набора констант каждой дизъюнкции D_i на две части. Первая часть формируется из констант, которые появляются в любом выражении $D_{i+1} \dots D_k$. Вторая часть содержит остальные константы.

$$(\overline{x_6} \vee x_1)(\overline{x_1} \vee x_2 \vee x_3)(\overline{x_2} \vee x_1)(\overline{x_4} \vee x_1)(\overline{x_3} \vee x_6)(\overline{x_5} \vee x_6)(\overline{x_6} \vee x_5 \vee x_4)$$

Сравнение скорости поиска эвристик

Table 2. Results of computer experiments

Net Param.	PI	No Sort		Sort by Length			Sort by Variables			Reordering Literals			SV + RL		
		TS	NPI	TS	NPI	%	TS	NPI	%	TS	NPI	%	TS	NPI	%
20x15	144	42194	6336	7680	1296	18.2	7522	1296	17.8	2536	0	6.0	702	0	1.7
20x18	145	7755	420	9504	420	122.6	6391	420	82.4	2596	0	33.5	2191	0	28.3
20x18	12	1422	150	645	24	45.4	599	24	42.1	247	0	17.4	283	0	19.9
20x18	105	2486	319	3166	319	127.4	3166	319	127.4	702	0	28.2	1067	0	42.9
20x18	70	3024	317	1543	125	51.0	1435	125	47.5	816	0	27.0	738	0	24.4
20x18	28	1346	168	671	84	49.9	635	84	47.2	333	0	24.7	247	0	18.4
20x20	36	4356	377	1638	48	37.6	1575	48	36.2	705	0	16.2	654	0	15.0
20x20	16	2650	92	660	56	24.9	826	56	31.2	881	0	33.2	391	0	14.8
20x20	117	3418	406	4181	406	122.3	4181	406	122.3	773	0	22.6	1139	0	33.3
25x20	946	51275	6111	26030	6111	50.8	26075	4895	0.9	6365	0	12.4	2910	0	5.7
25x20	144	55608	9026	4222	540	7.6	1770	540	3.2	2283	0	4.1	438	0	0.8
25x20	560	63234	5622	22826	6226	36.1	18831	6226	29.8	4170	0	6.6	2125	0	3.4
25x20	91	6838	288	4603	288	67.3	4109	288	60.1	1917	0	28.0	970	0	14.2
Average:						58.5			53.7			20.0			17.1

Пример преобразования полученных из листьев дерева простых импликант в тернарную матрицу:

$$\bar{x}_1 \bar{x}_2 \bar{x}_3 x_5 x_6$$

$$\bar{x}_2 \bar{x}_3 x_1 x_5 x_6$$

$$x_1 x_2 x_3 x_6$$

$$x_1 x_2 x_5 x_6$$

$$x_1 x_3 x_4 x_6$$

$$x_1 x_4 x_5 x_6$$

	p_1	p_2	p_3	p_4	p_5	p_6
$\bar{x}_1 \bar{x}_2 \bar{x}_3 x_5 x_6$	0	0	0	—	1	1
$\bar{x}_2 \bar{x}_3 x_1 x_5 x_6$	—	0	0	1	1	1
$x_1 x_2 x_3 x_6$	1	1	1	—	—	1
$x_1 x_2 x_5 x_6$	1	1	—	—	1	1
$x_1 x_3 x_4 x_6$	1	—	1	1	—	1
$x_1 x_4 x_5 x_6$	1	—	—	1	1	1

Сифоны

Сифон – Набор позиций такой, что каждый переход, который кладет token в набор, также забирает token.

Это означает, что сифон либо никогда не имеет в наборе токенов, либо имеет константное число.

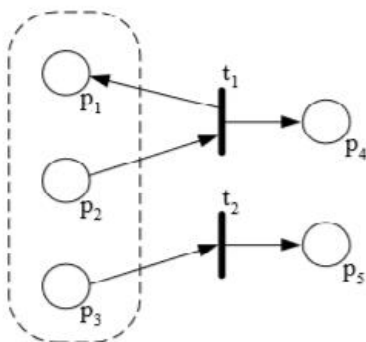


Fig. 2. Example of a deadlock $S = \{p_1, p_2, p_3\}$,
 $\bullet S = \{t_1\}$, $S^\bullet = \{t_1, t_2\}$.

Сифоны являются, по сути, противопоставлением ловушкам. Они также активно используются в анализе сетей Петри.

Демонстрация программы

Выводы по работе

Практическая значимость:

Программа будет использоваться для верификации распределенных систем.

Также может выступать в роли агента многоагентной системы.

Список использованных источников

- Сеть Петри. [Электронный ресурс]// URL: <https://ru.wikipedia.org/wiki/СетьПетри> (Дата обращения: 23.03.2016, режим доступа: свободный).
- Питерсон Дж. Теория сетей Петри и моделирование систем. — М: Мир, 1984. — 264 с.
- Котов В. Е. Сети Петри. — М: Наука, 1984. — 160 с.
- Reisig W.: Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies. Springer 2013.
- A. Węgrzyn, A. Karatkevich, and J. Bieganowski. Detection of deadlocks and traps in Petri nets by means of Thelen's prime implicant method. Applied Mathematics and Computer Science, 14(1):113–121, 2004.



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Спасибо за внимание!

Абдукеримов Тимур Русланович
trabdukerimov@edu.hse.ru
www.hse.ru