

Одесская национальная академия связи им. А.С. Попова

<http://onat.edu.ua>



LVEE 2019

Linux Vacation / Eastern Europe



Программные генераторы моделей в форме сетей Петри

Татьяна Шмелёва

t.shmeleva@onat.edu.ua

Методы моделирования дискретных систем

- Автоматы и их расширения: временные, взаимодействующие
- Логические системы: нечеткая, темпоральная, модальная
- Алгебры и исчисления процессов
- Сети Петри и их расширения: временные, стохастические, раскрашенные

Практическое применение сетей Петри (СП)

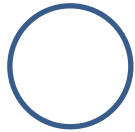
- Язык параллельного программирования:
экспериментальные разработки
программирование PLC (лог. контроллеры)
- Схемы параллельных программ UML/MDD
- Моделирование производственных систем:
схема организации автоматизированных
производств
- Транспортные системы
- Верификация сетевых протоколов

Инструментарий для сетей Петри

- Tina <http://projects.laas.fr/tina>
- CPN Tools <http://cpntools.org>
- Petri Nets World <http://www.informatik.uni-hamburg.de/TGI/PetriNets>

- Опера-Топаз
- Невод
- Deborah, Adriana, ParAd for Tina
<http://daze.ho.ua>

Элементы сети Петри



позиция (place)



переход (transition)



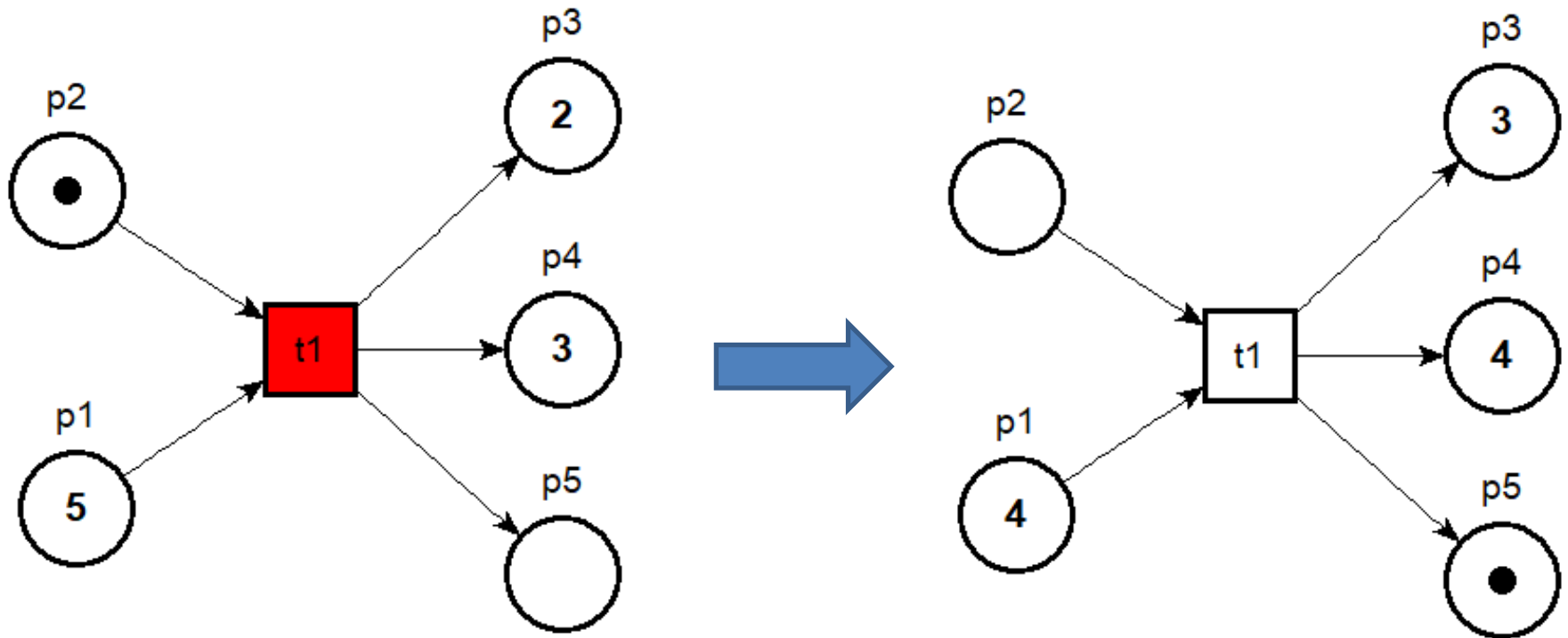
дуга (arc)



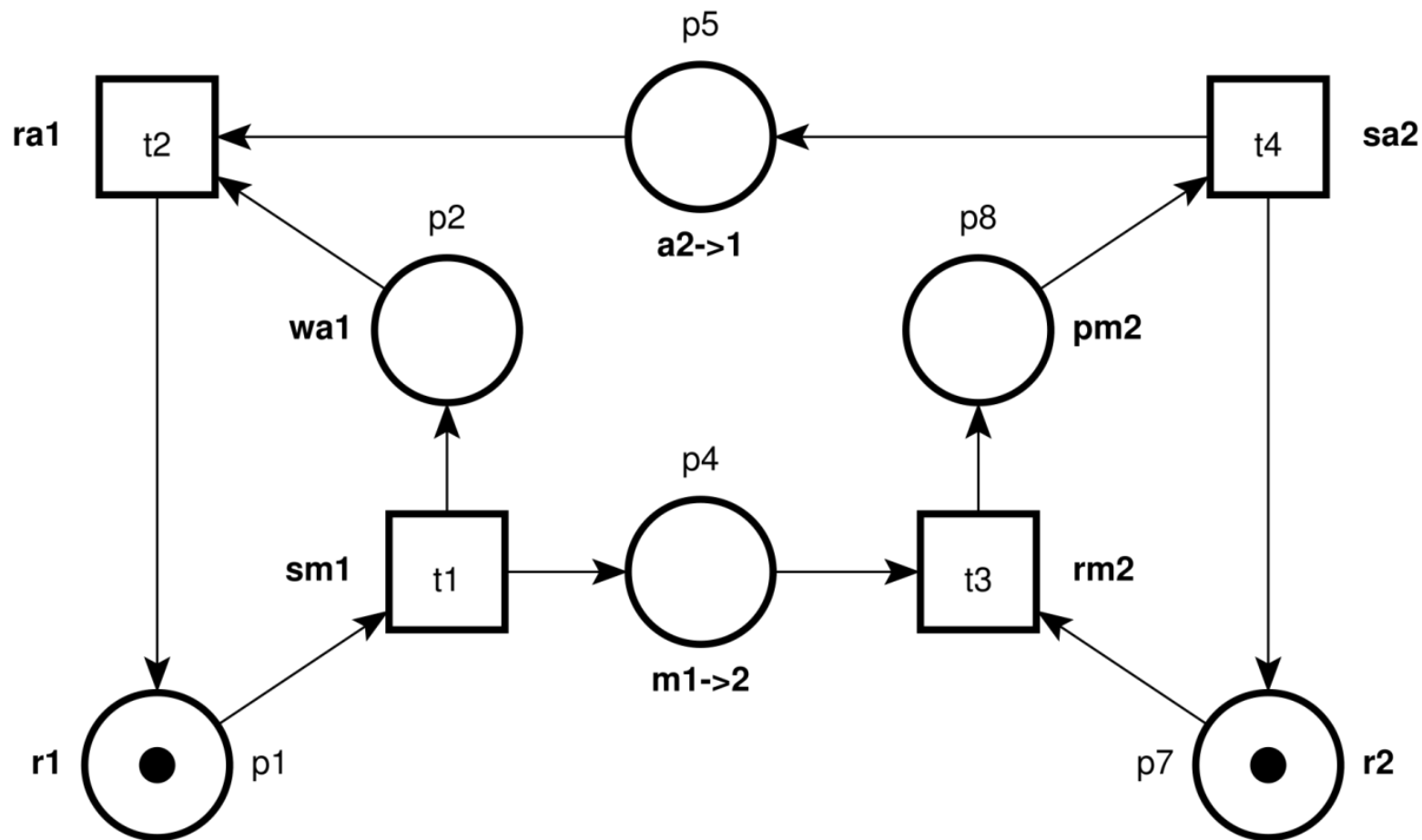
фишка (token)



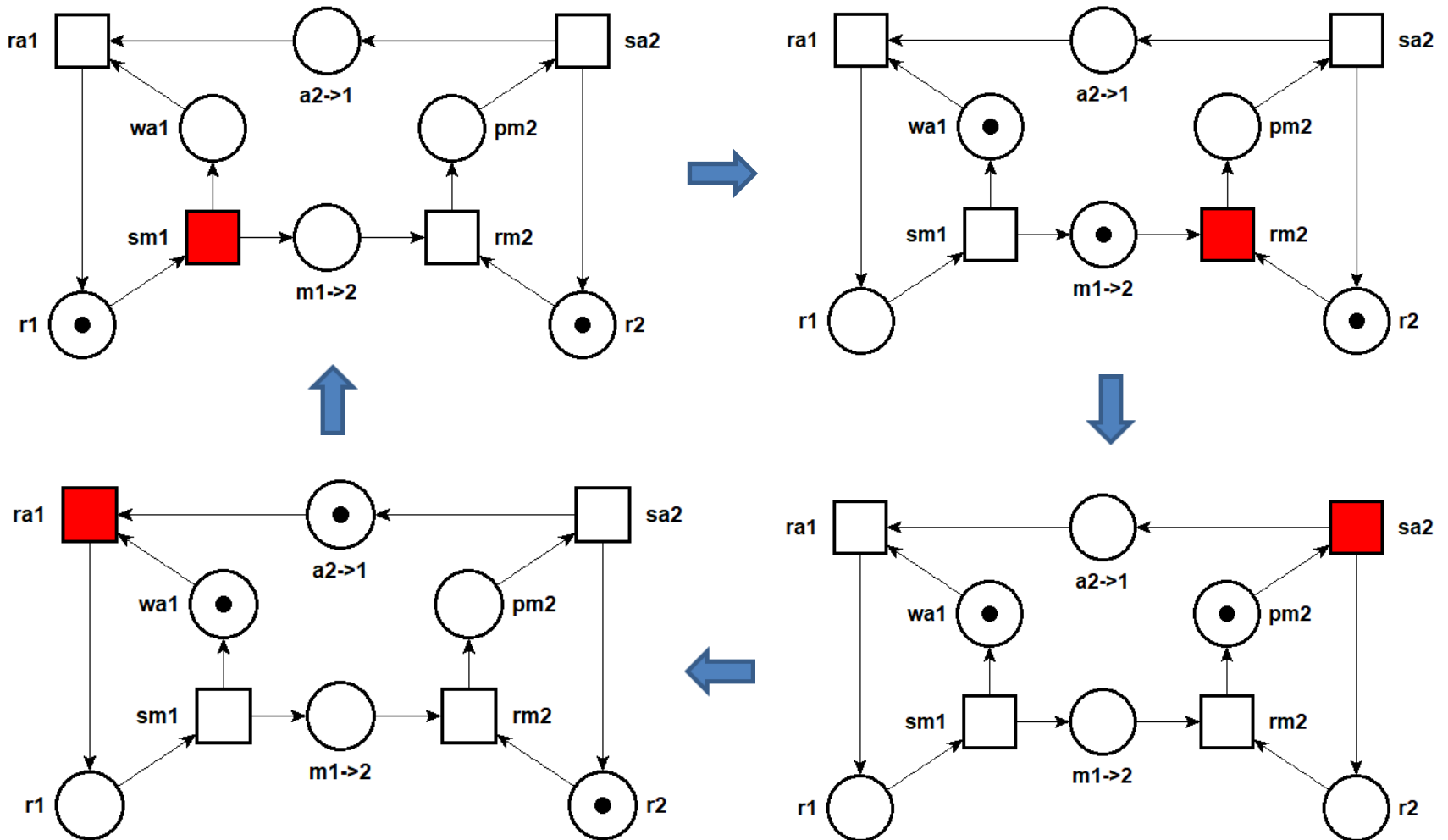
Срабатывание перехода сети Петри



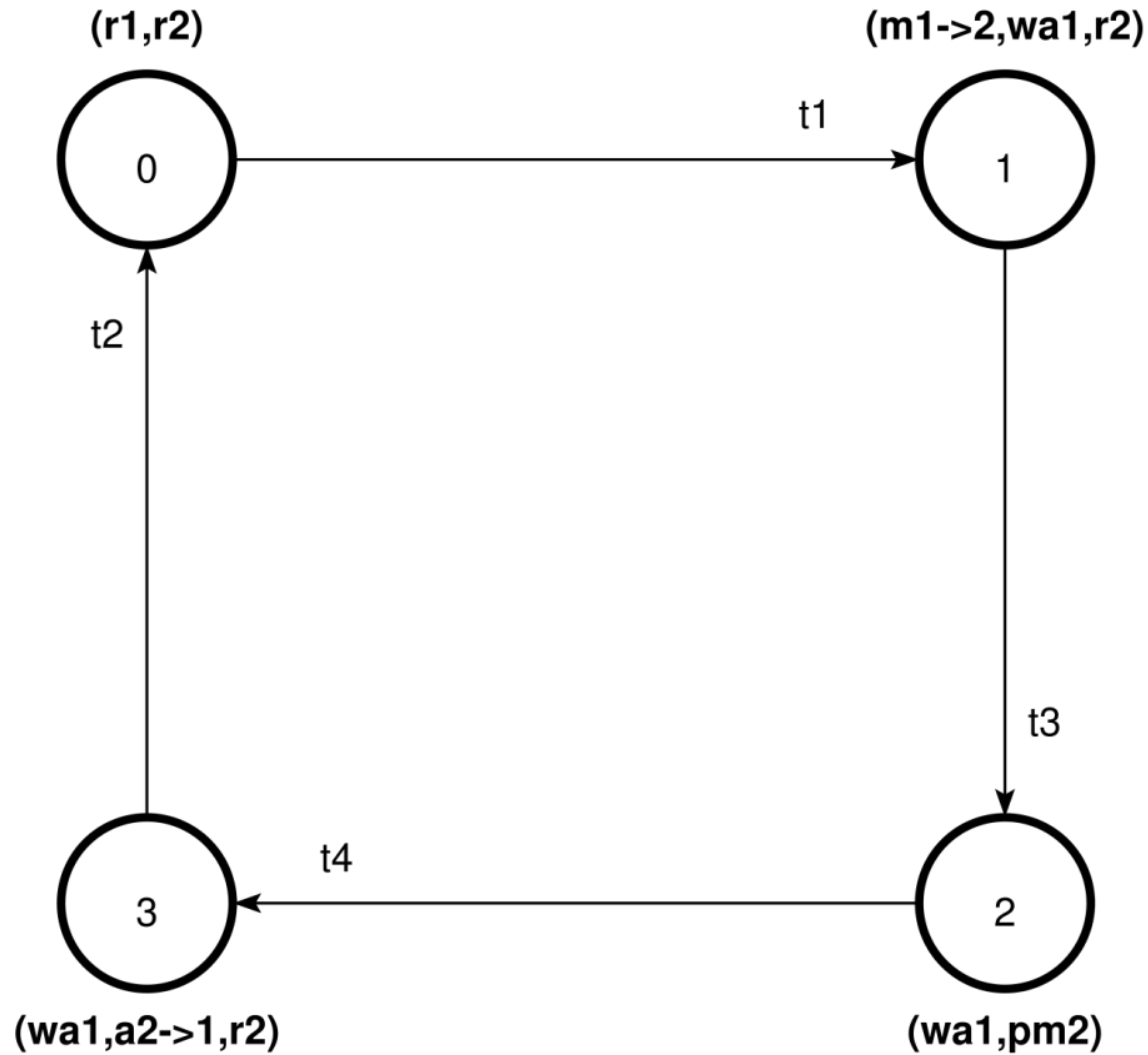
Протокол односторонней передачи сообщений с подтверждениями



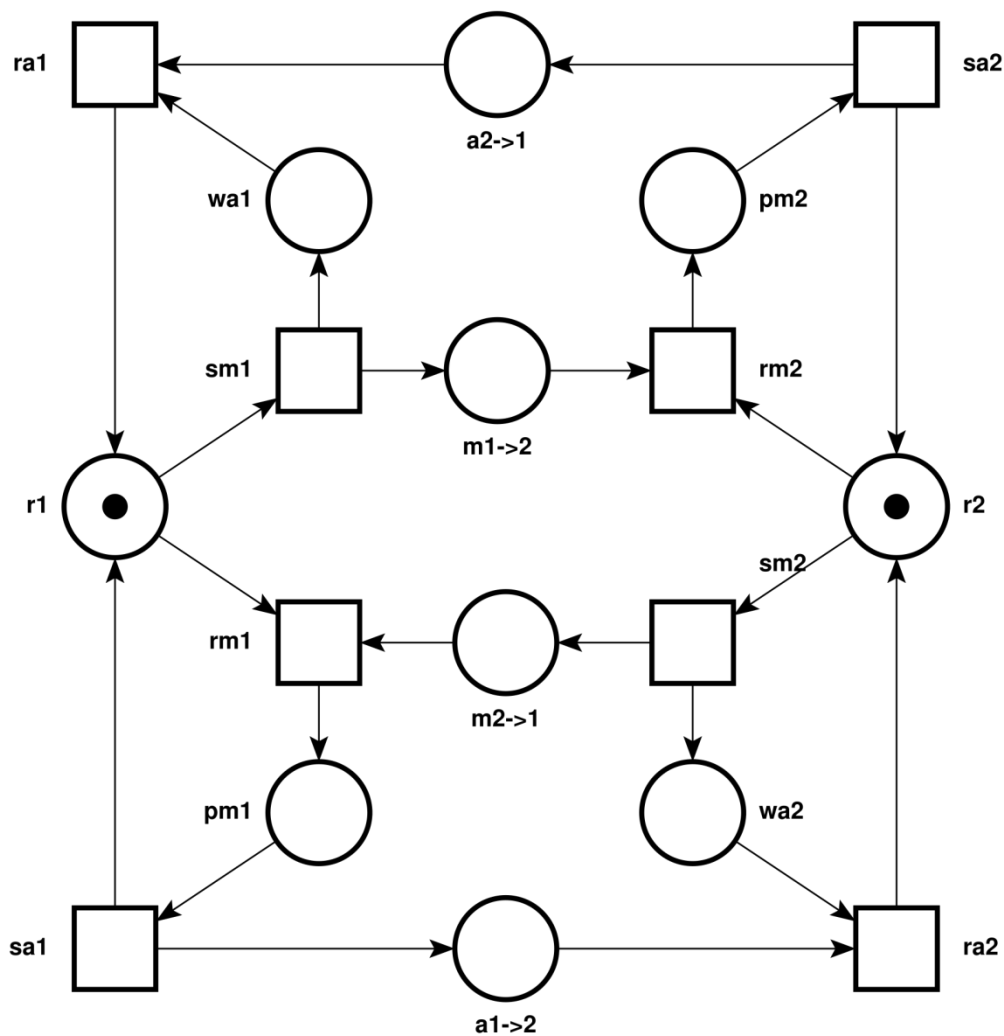
Динамика сети



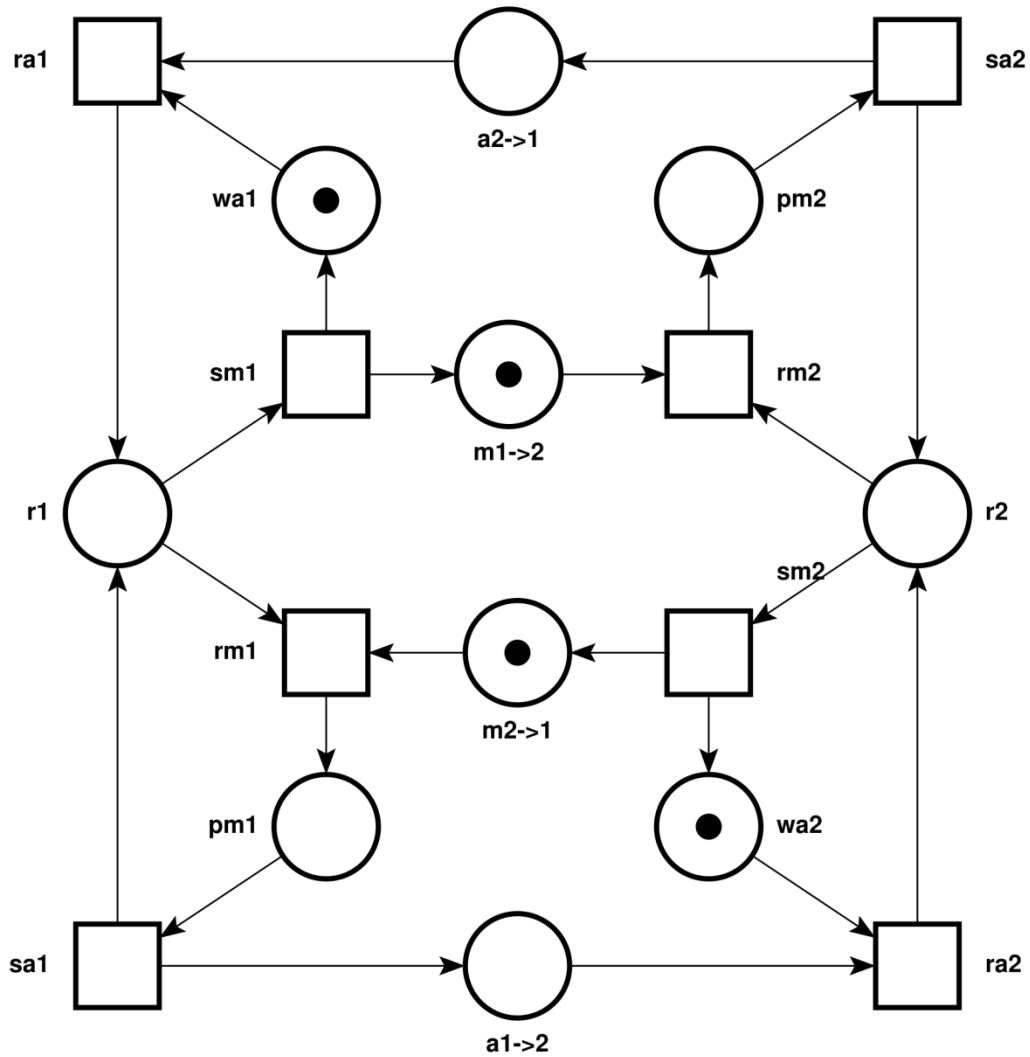
Граф достижимых маркировок



Протокол двусторонней передачи сообщений с подтверждениями



Тупик



Расширенный протокол: распознавание коллизий

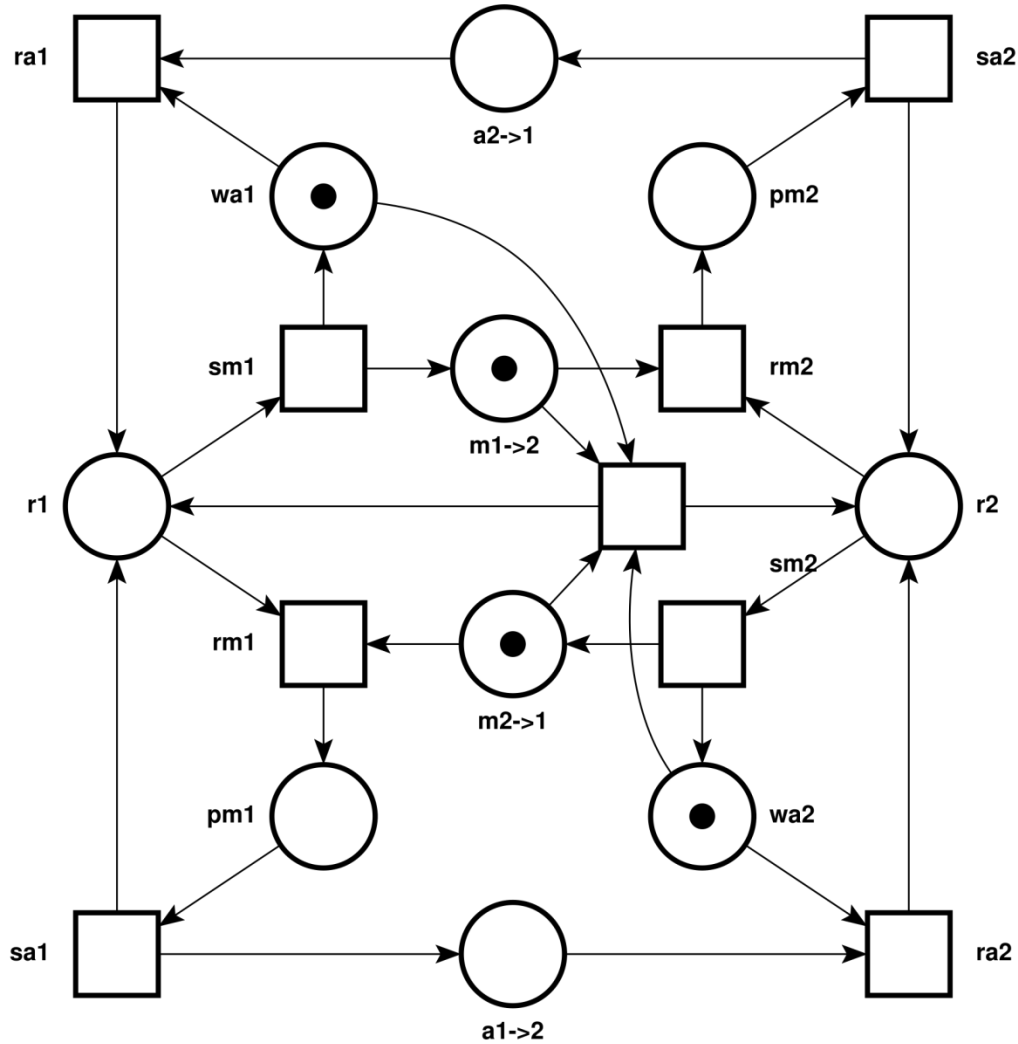
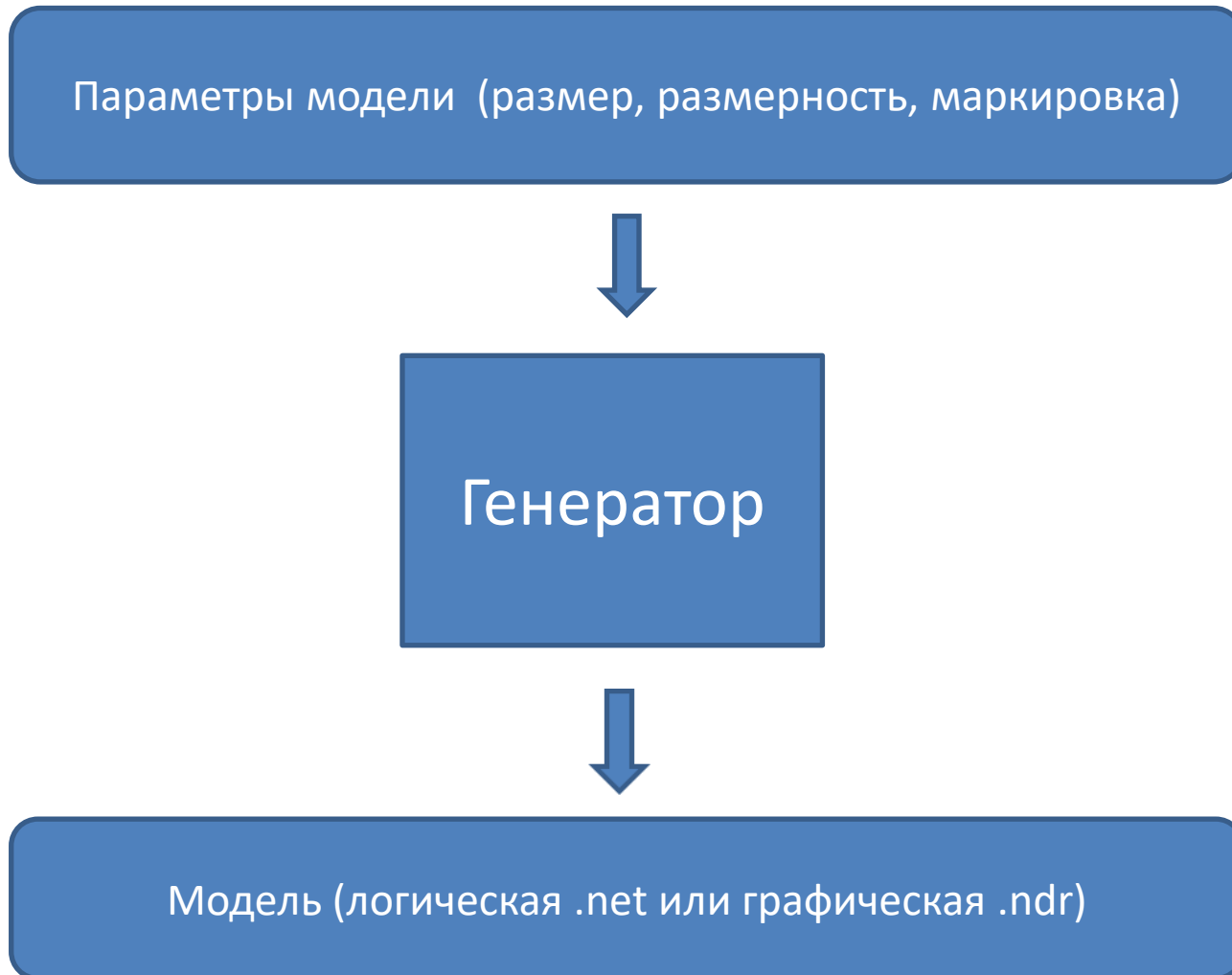


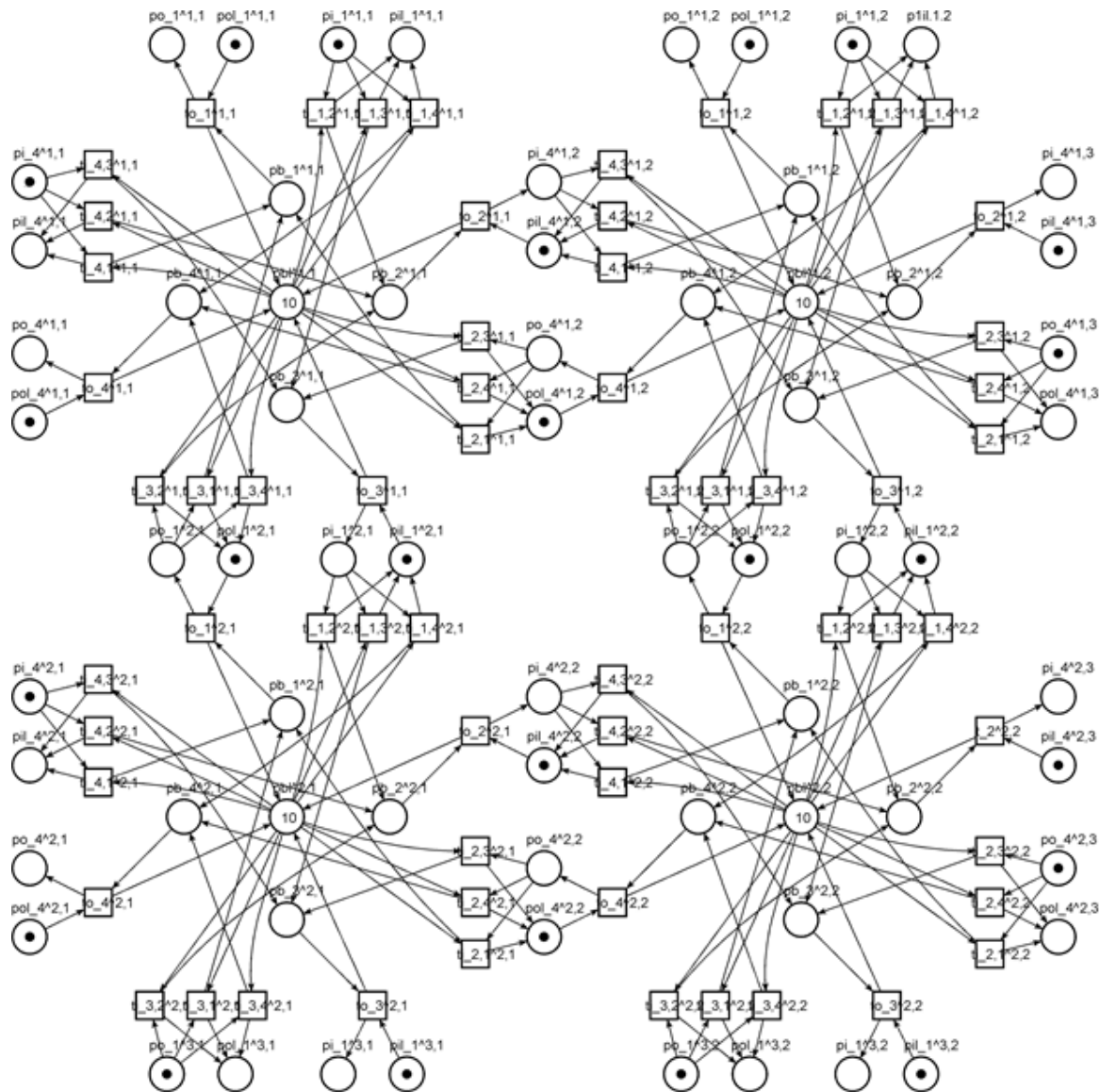
Схема работы генератора модели



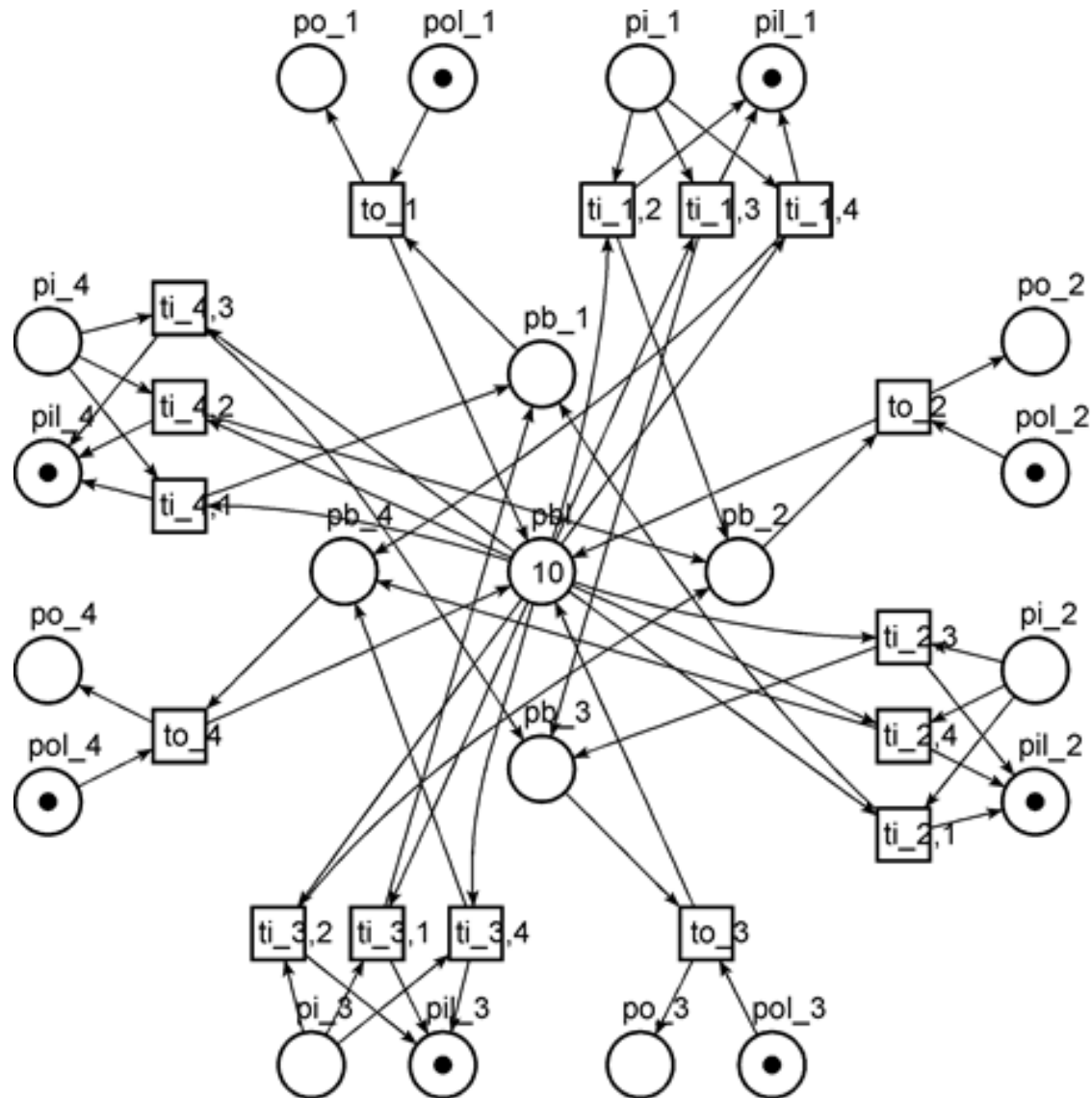
Обзор разработанных генераторов

Решетка	Размещение
Квадратная	https://github.com/dazeorgacm/sq
Треугольная	https://github.com/tishtri/g3a
Шестиугольная	https://github.com/tishtri/g6a
Гиперкуб с краевыми условиями	https://github.com/tishtri/hcgen
Гипертор	https://github.com/dazeorgacm/htgen
Гиперкуб с окрестностью Зайцева	https://github.com/dazeorgacm/hmn

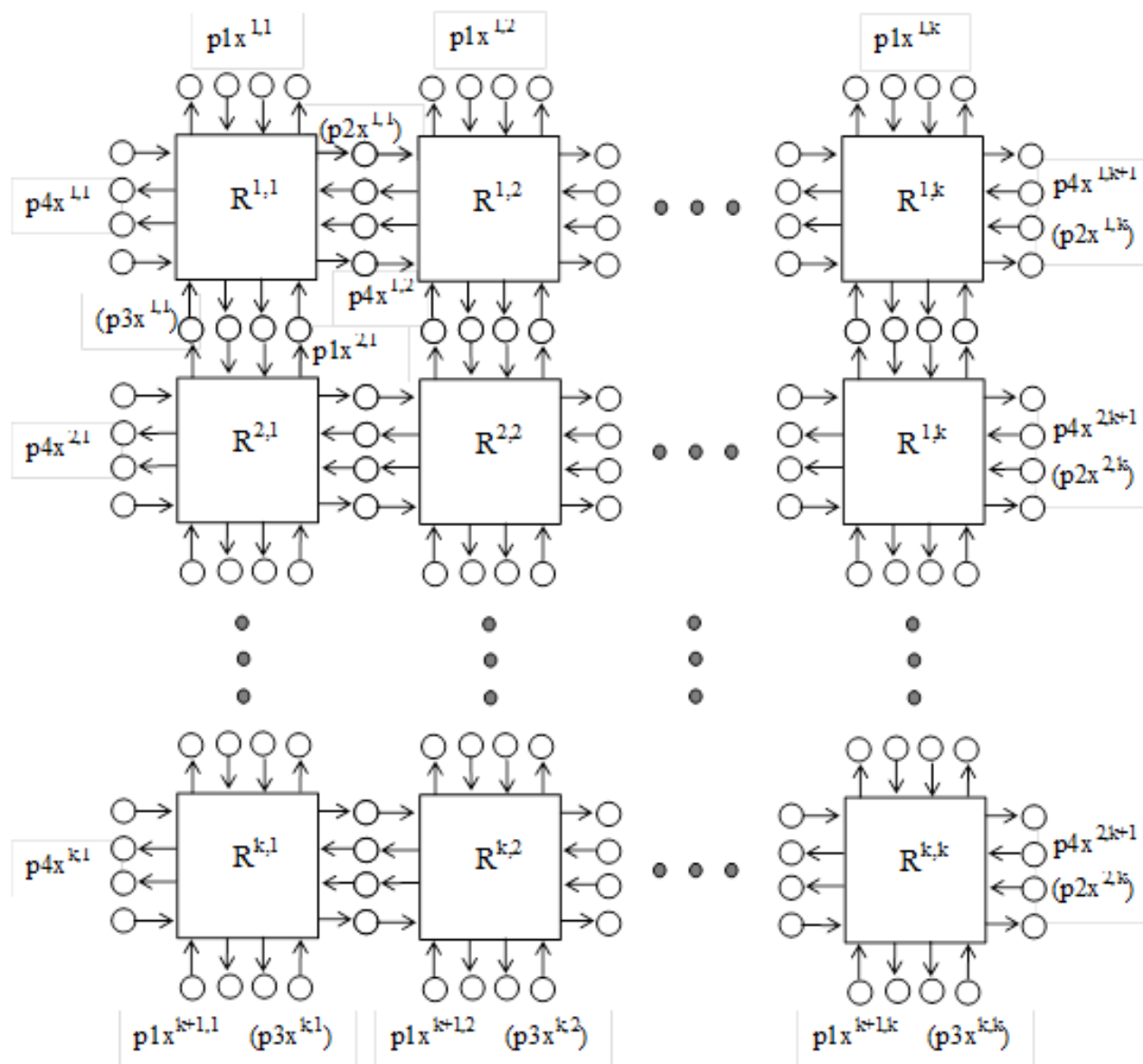
Пример сгенерированной решетки



Квадратная решетка: устройство



Квадратная решетка: композиция



Формат файлов Tina

- Логический

“tr” {name} {pi_1} ... {pi_k} -> {po_1} ... {po_l}

“pl” {name} (marking)

- Графический

“p” x y {name} marking

“t” x y {name}

“e” {name1} {name2}

Решетка на плоскости

```
for(i=1;i<=k;i++) // row
  for(j=1;j<k;j++) // column
  {
    off_x=i*STEP_i;
    off_y=j*STEP_j;
    print_cell(i,j,off_x,off_y);
  }
```

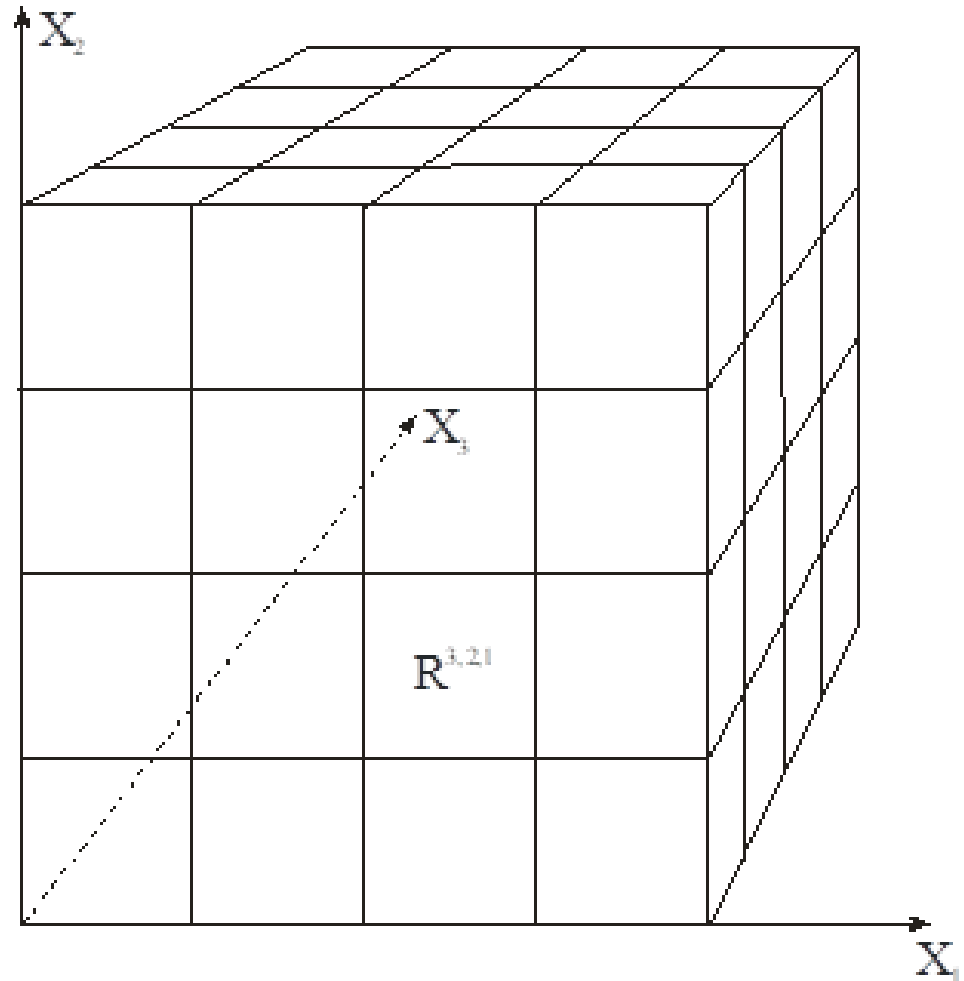
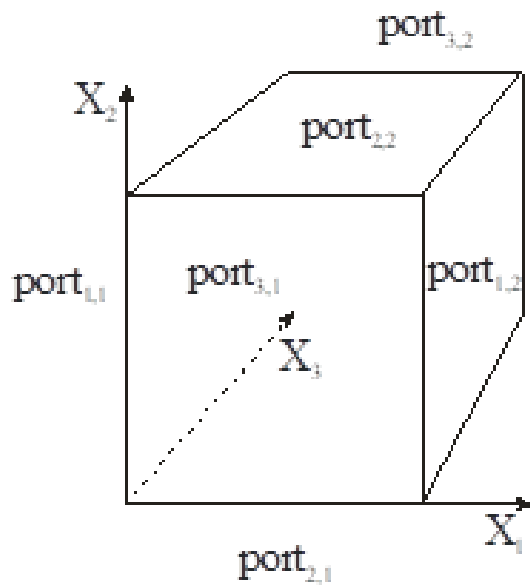
Фрагмент программы

```
for(i=0; i<=k; i++)
  for(j=0; j<=k; j++)
  {
    if(i<k)
    {
      printf("p %.1f %.1f {pol_1^%d.%d} 1 n\n",
        i*DI+280.0, j*Dj+50.0, j+1, i+1);
      ...
    }
    ...
    if(i<k && j<k)
    {
      ...
      printf("t %.1f %.1f {to_1^%d.%d} 0 w n\n",
        i*DI+240.0, j*Dj+130.0, j+1, i+1);
      ...
      printf("e {to_1^%d.%d} {po_1^%d.%d} 1 n\n",
        j+1, i+1, j+1, i+1);
      ...
    }
  }
```

Генераторы решеток в многомерных пространствах

- Перечисление узлов решетки для заданных размерности и размера
- Нумерация портов, расположенных на гранях гиперкуба единичного размера
- Именованное совмещенных контактных позиций
- Реализация разнообразных краевых условий: открытая, замкнутая, усеченное устройство, оконечное устройство
- Реализация различных окрестностей: фон Неймана, Мура, Зайцева

Нумерация портов и композиция решетки



Перечисление узлов

```
i = (int *) malloc( (d+1)* sizeof( int ) );  
...  
while( loop ){  
    // print current cell  
    go = 1;  
    j=d;  
    while( go ){  
        (i[j])++;  
        if( i[j] > k ){  
            if( j == 1 ) { loop=0; go=0; }  
            else {  
                i[j]=1;  
                j--;  
            }  
        }  
        else go=0;  
    } /* while go */  
} /* while loop */
```


Печать текущей ячейки

```
#define PUTINDX { for( u=1; u<=d; u++)  
                printf( "%.d", i[u] ); }  
  
#define PUTINDXNEXT( r )  
    { for( u=1; u<=d; u++)  
        printf( "%.d", (u!=(r))? i[u] : (i[u]<k)? i[u]+1 : 1 ); }  
  
...  
for( j=1; j<=d; j++ ) {  
    printf( "tr {to.d%d.n1", j );  
    PUTINDX;  
    printf( "} {pol.d%d.n1", j );  
    PUTINDX;  
    printf( "} {pb.d%d.n1", j );  
    PUTINDX;  
    printf( "} -> {po.d%d.n1", j );  
    PUTINDX;  
    printf( "} {pbl" );  
    PUTINDX;  
    printf( "}\n" );  
    ...  
}
```

Бесконечные сети Петри

гипертор

$$\left(\begin{array}{l} (to_{j,1}^i : pb_{j,1}^i, pol_{j,1}^i \rightarrow po_{j,1}^i, pbl^i), i_j \neq 1 \\ (ti_{j,1,j',n'}^i : pi_{j,1}^i, pbl^i \rightarrow pb_{j',n'}^i, pil_{j,1}^i), \\ i_j \neq 1 \wedge (n' \neq 1 \vee i_j \neq 1) \wedge (n' \neq 2 \vee i_j \neq k_{j'}) \\ (to_{j,2}^i : pb_{j,2}^i, pil_{j,1}^{xi(i,j)} \rightarrow pi_{j,1}^{xi(i,j)}, pbl^i), i_j \neq k_j \\ (ti_{j,2,j',n'}^i : po_{j,1}^{xi(i,j)}, pbl^i \rightarrow pb_{j',n'}^i, pol_{j,1}^{xi(i,j)}), \\ i_j \neq k_j \wedge (n' \neq 1 \vee i_j \neq 1) \wedge (n' \neq 2 \vee i_j \neq k_{j'}) \end{array} \right) \begin{array}{l} i = (i_1, \dots, i_d), 1 \leq i_j \leq k_j, \\ 1 \leq j \leq d, \\ 1 \leq j' \leq d, \\ 1 \leq n' \leq 2, (j', n') \neq (j, n). \end{array}$$

$$xi(i, j) = i', \quad i'_l = \begin{cases} i_l, & l \neq j, \\ i_l + 1, & l = j. \end{cases}$$

Свойства бесконечных сетей

- Сгенерировать последовательность моделей увеличивающегося размера и размерности
- Найти свойства моделей
- Описать свойства моделей в параметрическом виде
- Доказать свойства модели
- Сделать вывод о свойствах модели произвольного размера и размерности

Выводы

- Разработано программное обеспечение с открытым кодом для генерации квадратных, треугольных и шестиугольных решеток на плоскости, гиперкуба и гипертора в многомерных пространствах
- Генераторы использованы для доказательства свойств бесконечных сетей Петри
- Области применения: верификация коммуникационных протоколов, параллельное программирование и другие

Недавние публикации

- Dmitry A. Zaitsev, Tatiana R. Shmeleva and Jan Friso Groote, Verification of Hypertorus Communication Grids by Infinite Petri Nets and Process Algebra, [IEEE/CAA Journal of Automatica Sinica, 6\(3\), 2019, 733-742. DOI: 10.1109/JAS.2019.1911486](#)
- Dmitry A. Zaitsev, Ivan D. Zaitsev and Tatiana R. Shmeleva. Infinite Petri Nets: Part 2, Modeling Triangular, Hexagonal, Hypercube and Hypertorus Structures, Complex Systems, 26(4), 2017, 341-371. DOI: 10.25088/ComplexSystems.26.2.341
- Dmitry A. Zaitsev, Ivan D. Zaitsev and Tatiana R. Shmeleva. Infinite Petri Nets: Part 1, Modeling Square Grid Structures, [Complex Systems, 26\(2\), 2017, 157-195. DOI: 10.25088/ComplexSystems.26.2.157](#)