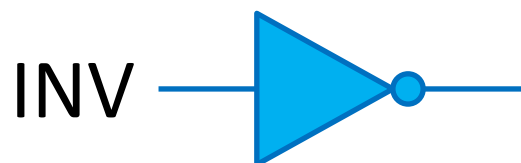


Поиск структурных соответствий.

Осень 2015

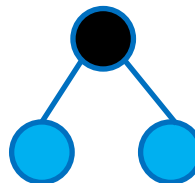
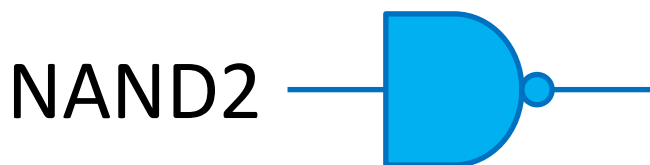


Представление библиотеки



I1v

t1.1

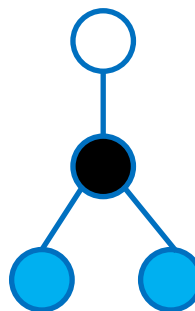
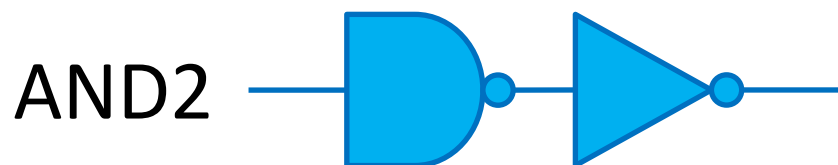


N1v

t2.1

N2v

t2.2




I1N1v

t3.1

I1N2v

t3.2

INV(I) 

NAND2(N) 

Terminal(v) 

Рекурсивный алгоритм поиска структурных соответствий

Match (u, v)

```
if u.isLeaf() return True
```

```
else
```

```
    if v.isLeaf() return False
```

```
    if u.degree() != v.degree() return False
```

```
    if v.degree() == 1
```

```
        u_c := u.get_child()
```

```
        v_c := v.get_child()
```

```
        return Match(u_c, v_c)
```

```
    else
```

```
        u_l := u.get_left_child()
```

```
        u_r := u.get_right_child()
```

```
        v_l := v.get_left_child()
```

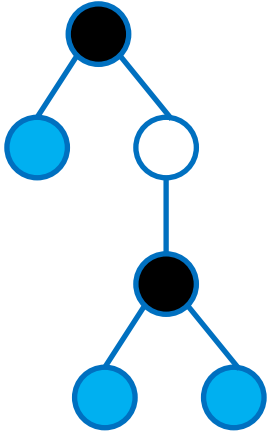
```
        v_r := v.get_right_child()
```

```
        m_1 := Match(u_l, v_l) and Match(u_r, v_r)
```

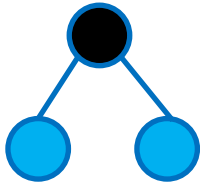
```
        m_2 := Match(u_l, v_r) and Match(u_r, v_l)
```

```
        return m_1 or m_2
```

Построение автомата



N1v	t1.1
N2I1N1v	t2.2
N2I1N2v	t2.2



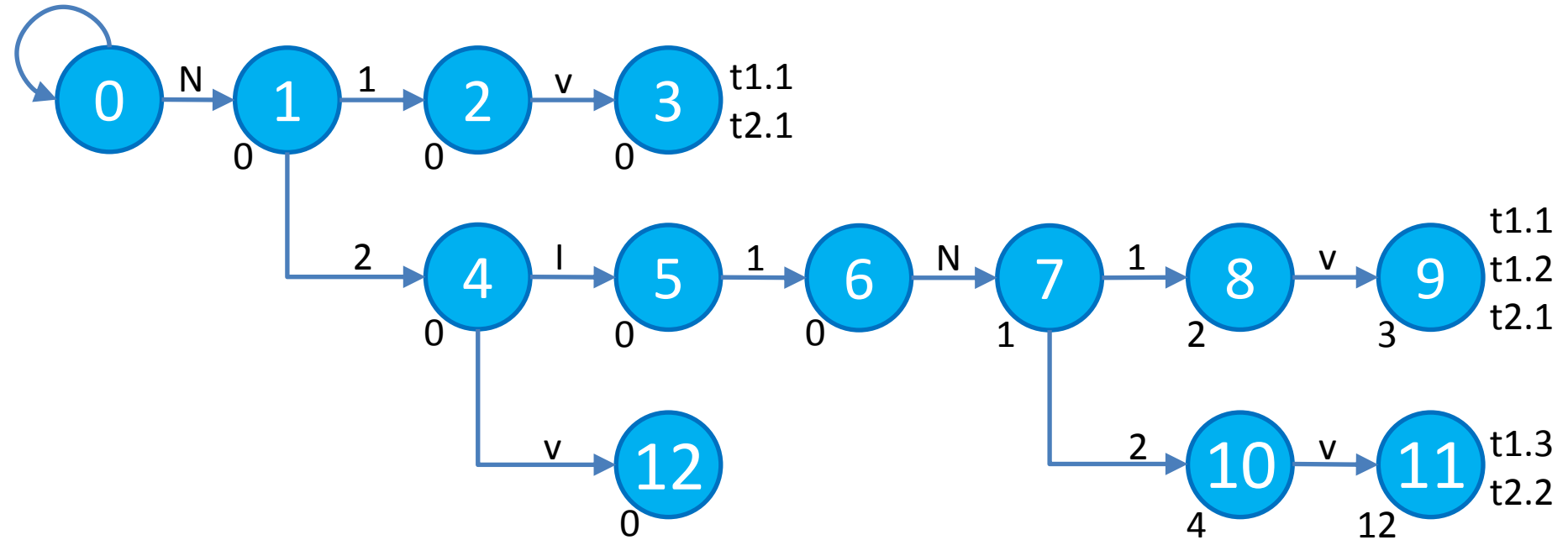
N1v	t2.1
N2v	t2.2

INV(I) ○

NAND2(N) ●

Terminal(v) ●

Построение автомата



N1v t1.1
 N2l1N1v t2.2
 N2l1N2v t2.2

N1v t2.1
 N2v t2.2

Обход целевого дерева

```
Visit_Node (G, n)
  if n - корень дерева
    n.state:=Next(0, n.symbol)
  else
    n.state:=Next(Next(n.parent.state,k),n.symbol)
    //вершина n является k-ым потомком родительской
    //вершины n.parent
  foreach потомка c вершины n
    Visit_Node(G, c)
  Post_Process(G,n)
```

Обход целевого дерева

```
Post_Process (G, n)
  n.b(i):=0
  if n.state == "ACCEPT"
    Set_Partial(G, n, n.state)
  foreach шаблона  $t_i$ 
    n.b(i):=n.b(i) or  $\prod_{c \in C(n)} c.b_i/2$ 
    //C(n) – множество всех дочерних вершин вершины n
  Do_Reduce(G, n)
```

```
Set_Partial(G, n, s)
  foreach строки  $t_i$  длины  $2^{j+1}$  обнаруженных в s
    n.b(i):=n.b(i) or  $2^j$ 
```