

2022 International Conference on Information Technologies (InfoTech-2022),  
Proceedings of the IEEE Conference, 15-16 September 2022, Bulgaria

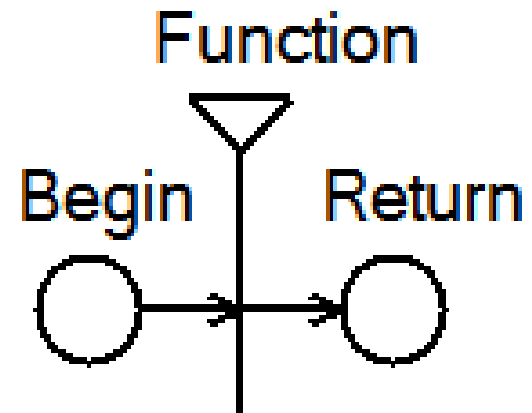
# Generalized Nets Representing C - based Programming Constructs

Valeri Petrov Gochev<sup>1</sup> and Nikolay Lyuboslavov Hinov<sup>2</sup>

<sup>1</sup>Department of Telecommunications, Faculty of Telecommunications and Management  
University of Telecommunications and Posts  
1 Acad. Stefan Mladenov str., 1700, Sofia, Bulgaria

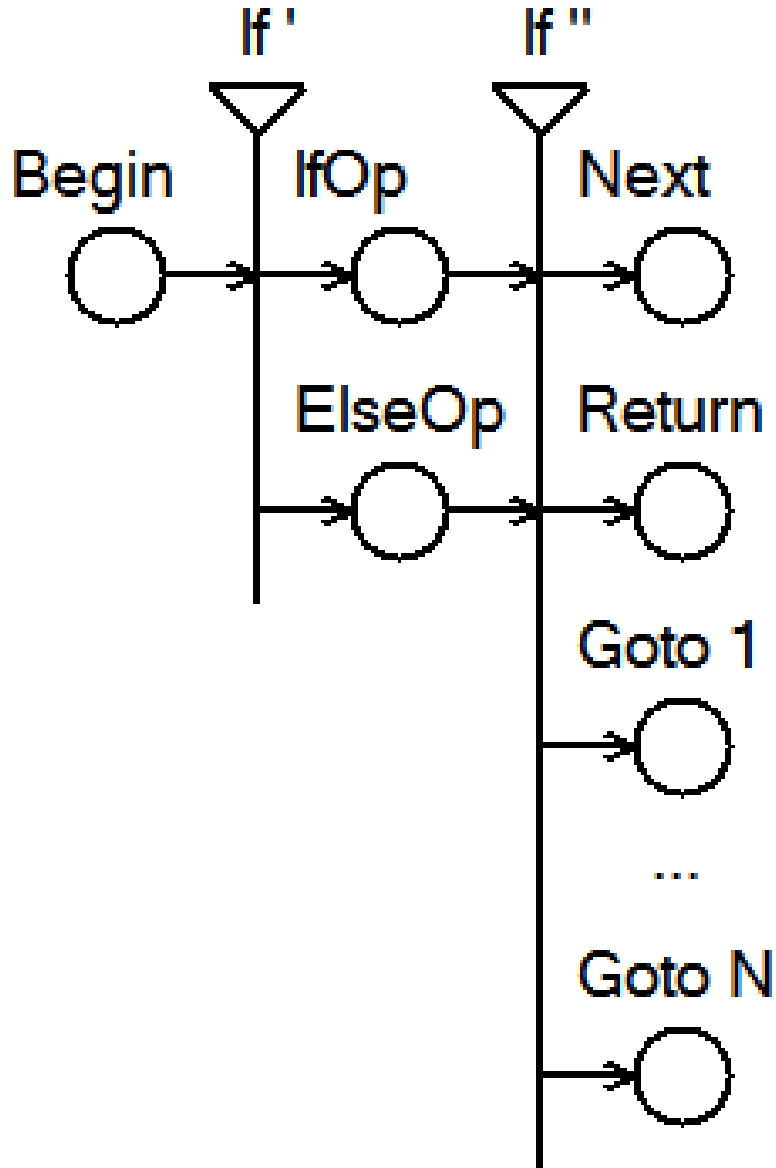
<sup>2</sup>Department of Power Electronics, Faculty of Electronic Engineering and Technologies  
Technical University of Sofia  
8 Kliment Ohridski blvd., 1000 Sofia, Bulgaria  
[hinov@tu-sofia.bg](mailto:hinov@tu-sofia.bg)

Common generalized net, representing execution of a function



# Generalized nets, representing conditional operators (1)

```
if(boolExpression)  
    ifOperator;  
else  
    elseOperator
```



Generalized nets,  
 representing  
 conditional  
 operators (2)

$$R_{If'} = \begin{array}{c|cc} & IfOp & ElseOp \\ \hline Begin & r_{If} & \neg r_{If} \end{array}$$

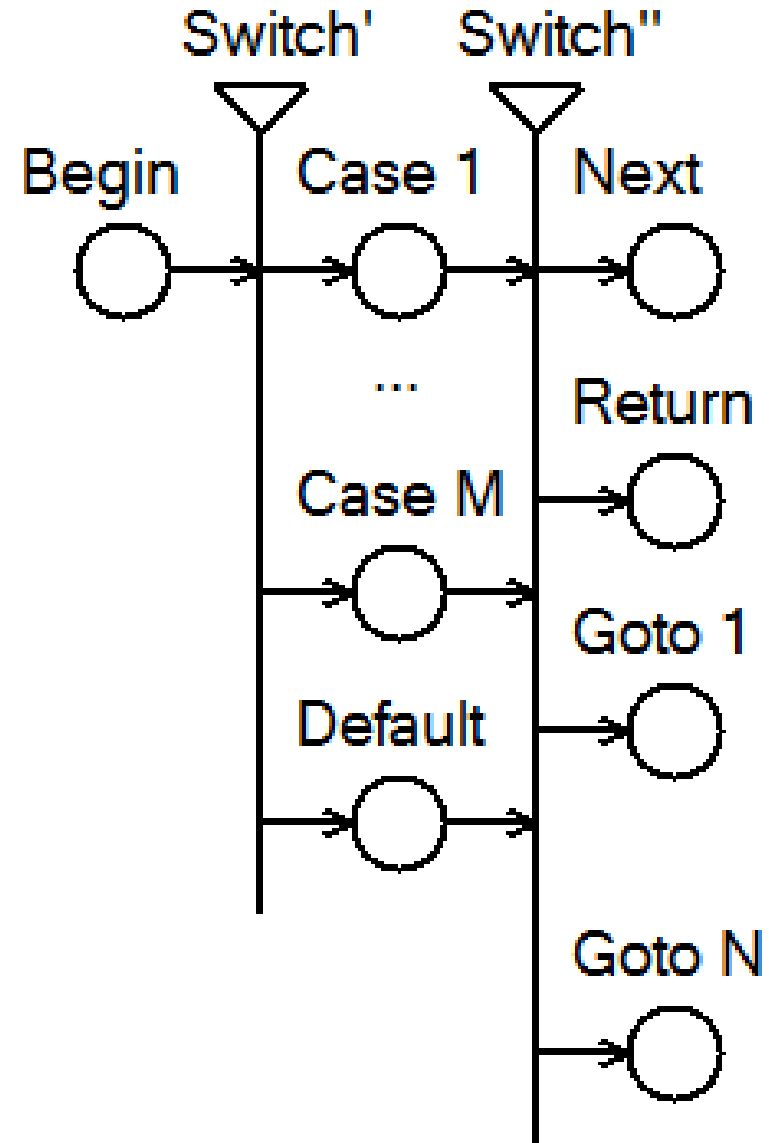
	<i>Next</i>	<i>Return</i>	<i>Goto 1</i>	...	<i>Goto N</i>
<i>IfOp</i>	<i>r Next</i>	<i>r Return</i>	<i>r Goto 1</i>	...	<i>r Goto N</i>
<i>ElseOp</i>	<i>r Next</i>	<i>r Return</i>	<i>r Goto 1</i>	...	<i>r Goto N</i>

# Generalized nets, representing conditional operators (3)

```
switch(x)
{
  case a_1:
    operator_1;
    break;

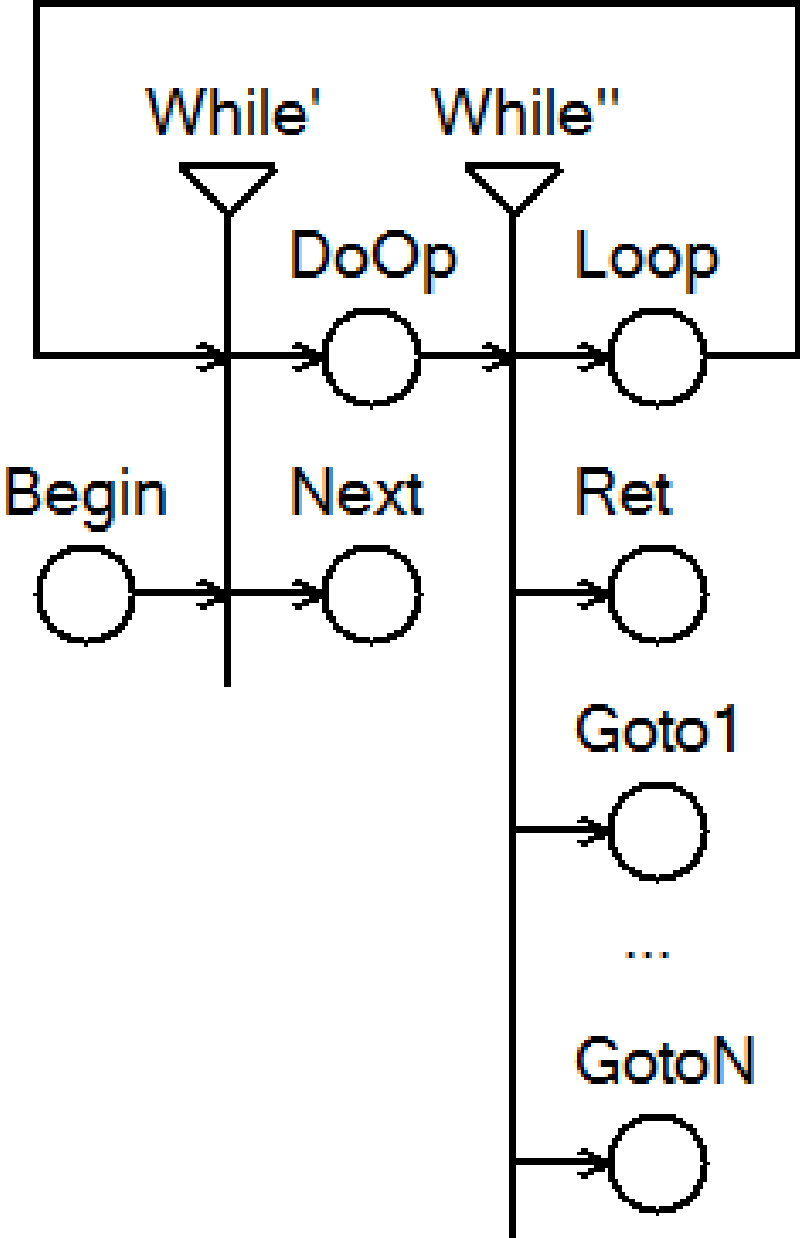
  ...

  case a_M:
    operator_M;
    break;
  default:
    operator_D;
    break;
}
```



# Generalized nets, representing loops (1)

```
while(boolExpression)
{
  doOperator;
}
```



# Generalized nets, representing loops (2)

$$R \textit{ While } ' = \begin{array}{c|cc} & \textit{DoOp} & \textit{Next} \\ \hline \textit{Begin} & \textit{true} & \neg \textit{r If} \\ \textit{Loop} & \textit{r If} & \neg \textit{r If} \end{array}$$

	<i>Loop</i>	<i>Return</i>	<i>Goto 1</i>	...	<i>Goto N</i>
<i>DoOp</i>	<i>r Next</i>	<i>r Return</i>	<i>r Goto 1</i>	...	<i>r Goto N</i>

```

static long fact(int n)
{
  if (n < 0 || n > 20)
  {
    return -1;
  }
  int i = 1;
  long p = 1;
  while(i <= n)
  {
    p *= i;
    i++;
  }
  return p;
}

```

Generalized nets, representing execution of a program code

