

FLIDE

FLOGOL Integrated Development Environment

Система
функционально-логического
программирования
на языке S-FLOGOL

Язык S-FLOGOL

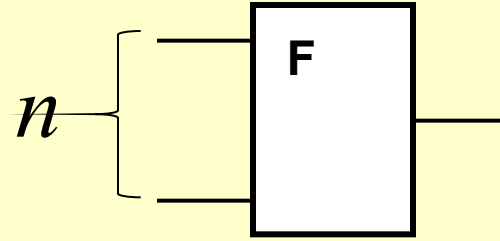
- Основан на теории направленных отношений (НО) (Фальк В.Н., Кутепов В.П.).
- Имеет развитые средства схемного описания НО.
- Допускает использование:
 - индексированных имен НО,
 - параметризованных НО.
- Поддерживает объектно-ориентированный стиль программирования.
- Позволяет строить многомодульные программы.
- Обладает средствами ограничения области видимости определяемых отношений (Private-домены).

Среда программирования

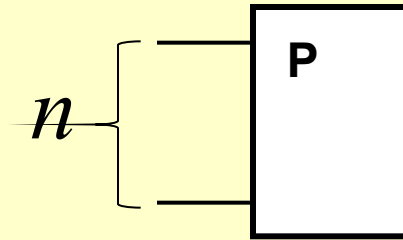
- Проектный подход к организации работы с комплексами программ.
- Оригинальные технологии графического и структурированного текстового построения программ.
- Интегрированные средства разработки и отладки.
- Собственное сетевое ядро вычислений и компилятор запросов на языке S-FLOGOL.
- Возможность ограниченного импорта программ, написанных на языке Пролог.
- Реализация в среде C++ Builder 5.0.

Семантические объекты

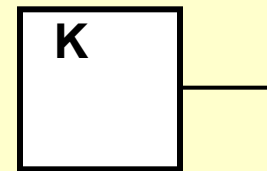
- Функция $F^{(n,1)}$



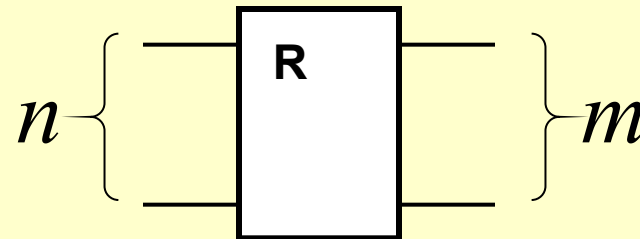
- Предикат $P^{(n,0)}$



- Константа $K^{(0,1)}$

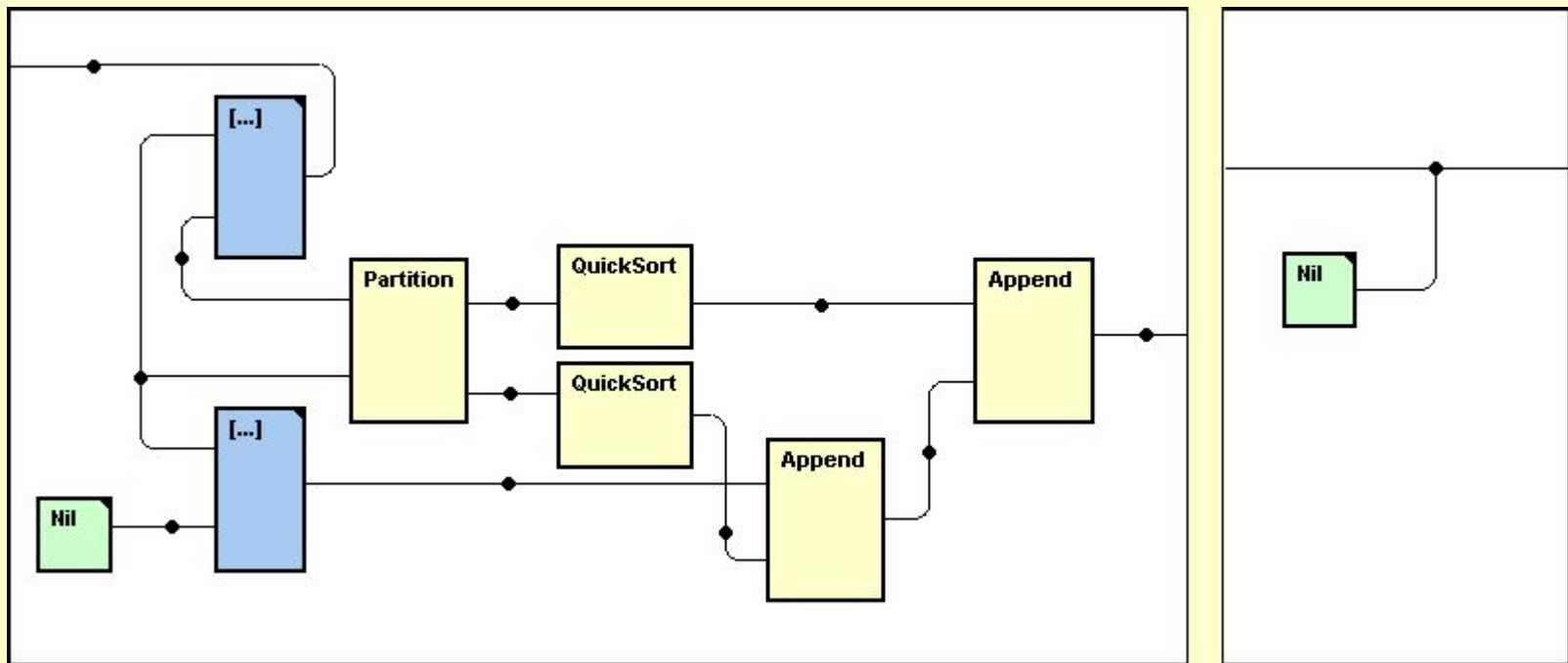


- НО общего вида $R^{(n,m)}$

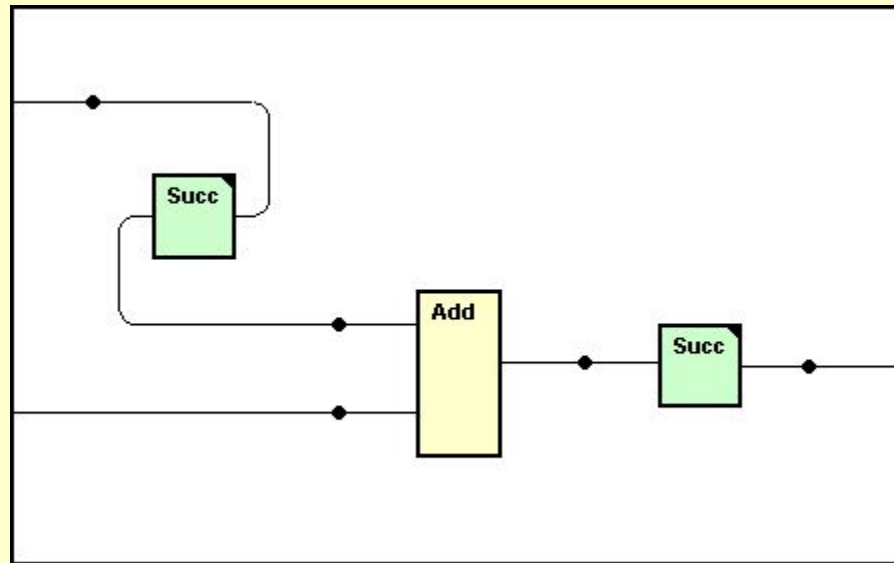
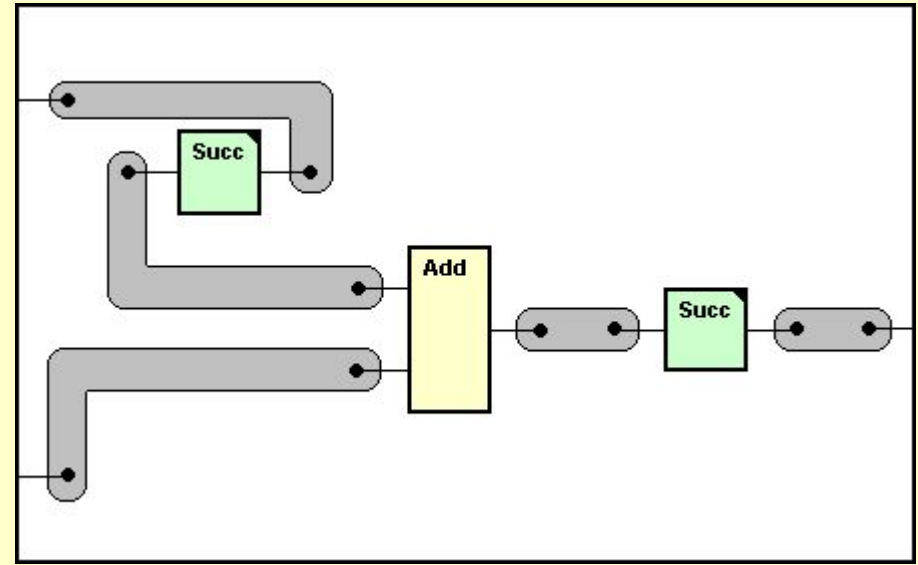
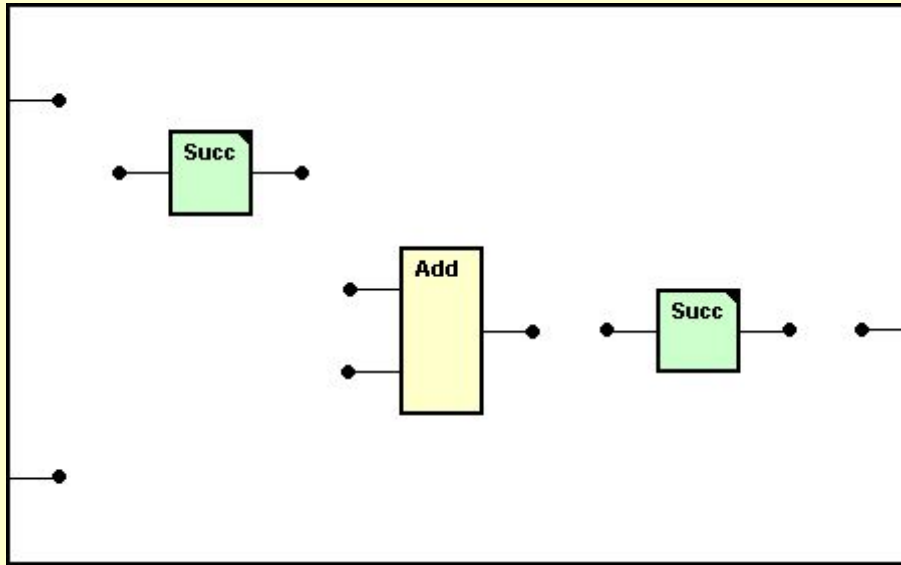


Сетевое представление НО

Пример: НО $QuickSort^{(1,1)}$ (быстрая сортировка)

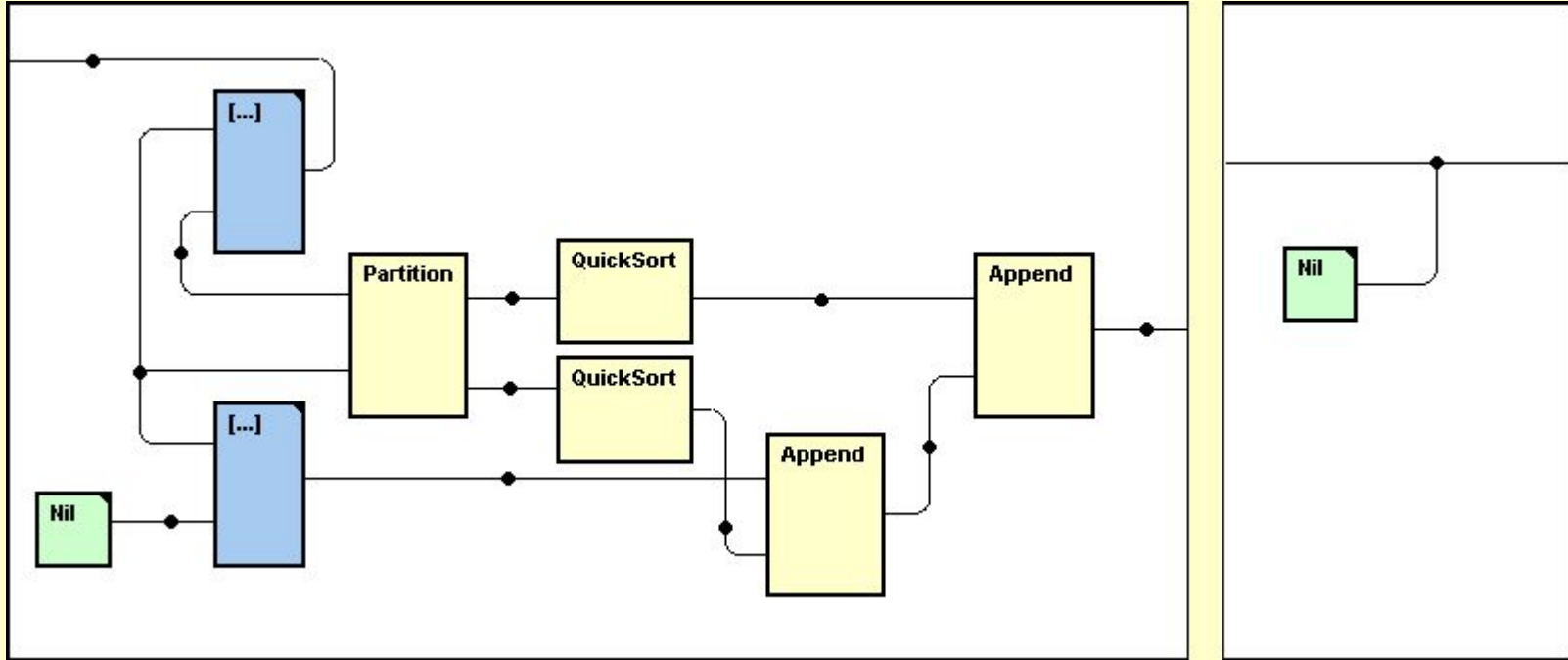


Графическое построение сети



Текстовое представление НО

Пример: НО *QuickSort*^(1,1)



...

QuickSort=

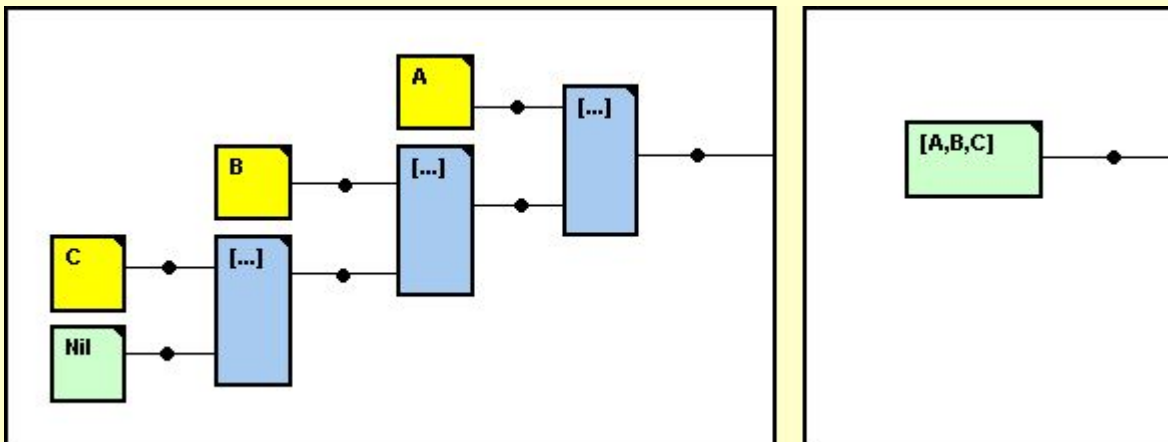
```
{LCons(x,xs):Append(@ls,Append(LCons(x,Nil),@bs))  
  ?Partition(xs,x)=ls,bs};
```

```
QuickSort={Nil:Nil};
```

Системные типы данных

- Натуральные числа (0, 1, 2, ...).
- Списки термов ([], [A(B), [D]]).
- Строки ('Some text').

Пример записи списка [A,B,C] в сетевой форме при помощи конструктора списка и системного генератора списков:



Общий вид интерфейса системы

The screenshot displays the FLIDE software interface, which is used for developing and testing logic programs. It is divided into several main sections:

- Top Panel:** Contains the menu bar (Файл, Правка, Вид, Действия, Справка) and the title bar for the active window, "FLIDE - [Arithm.FLT]".
- Left Panel:** A tree view of "Конструкторы" (Constructors) and "Правила" (Rules). Under "Правила", various rules are listed, including "Add.1", "Add.2", "Mult.1", "Mult.2", "Nat0", "Nat2", "Nat3", "Power.1", and "Power.2".
- Center Panel:** A text editor showing the source code of a module named "Arithm". The code defines several rules and constants, such as:

```
MODULE Arithm=  
  (0:1)Null;  
  (1:1)Succ;  
  Nat3={:Succ(Succ(Succ(Null)))};  
  Nat2={:Succ(Succ(Null))};  
  Nat0={:Null};  
  Add={Null,x:x};  
  Add={Succ(x),y:Succ(@x,y)};  
  Mult={Null,x:Null};  
  Mult={Succ(x),y:Add(@x,y)};  
  Power={x,Null:Succ(Null)};  
  Power={x,Succ(y):Mult(x,@(x,y))};  
  QAdd={:Add(Nat2,Nat3)};  
  Mult={:Mult(Nat2,Nat3)}.
```
- Right Panel:** A project browser showing a hierarchy of modules under "Demo":
 - Сетевые модули (Network modules): Arithm, ColorMap, Hanoi, Lists, Queens, Triggers.
 - Дедуктивные модули (Deductive modules): Chang&Yong, Simple, SteamRoller.
 - Текстовые модули (Textual modules): Arithm, ColorMap, Hanoi, Lists, Queens, RelComp, Triggers.
- Bottom Panel:** A diagram editor showing a flow graph. The graph consists of nodes and directed edges. The nodes are labeled "Succ" (green), "Mult" (purple), and "Add" (yellow). The flow starts from a source node, goes through "Succ", then "Mult", and finally "Add".

Графический редактор

The screenshot shows the FLIDE graphical editor interface. The title bar reads "FLIDE - [Hanoi.FLN]". The toolbar contains various icons for file operations, editing, and execution. The left sidebar displays a project tree with the following structure:

- Конструкторы
 - [...] (+2+:+1)
 - A (+0+:+1)
 - B (+0+:+1)
 - C (+0+:+1)
 - Nil (+0+:+1)
 - Null (+0+:+1)
 - Succ (+1+:+1)
 - to (+2+:+1)
- Константы
- Системные
- Правила
 - append.1
 - append.2
 - HANOI.1
 - HANOI.2**
- Запросы
 - Ханой(11)
 - Ханой(2)
- Результат
- Отладка

The main workspace shows a flowchart on a grid background. The flowchart consists of the following elements:

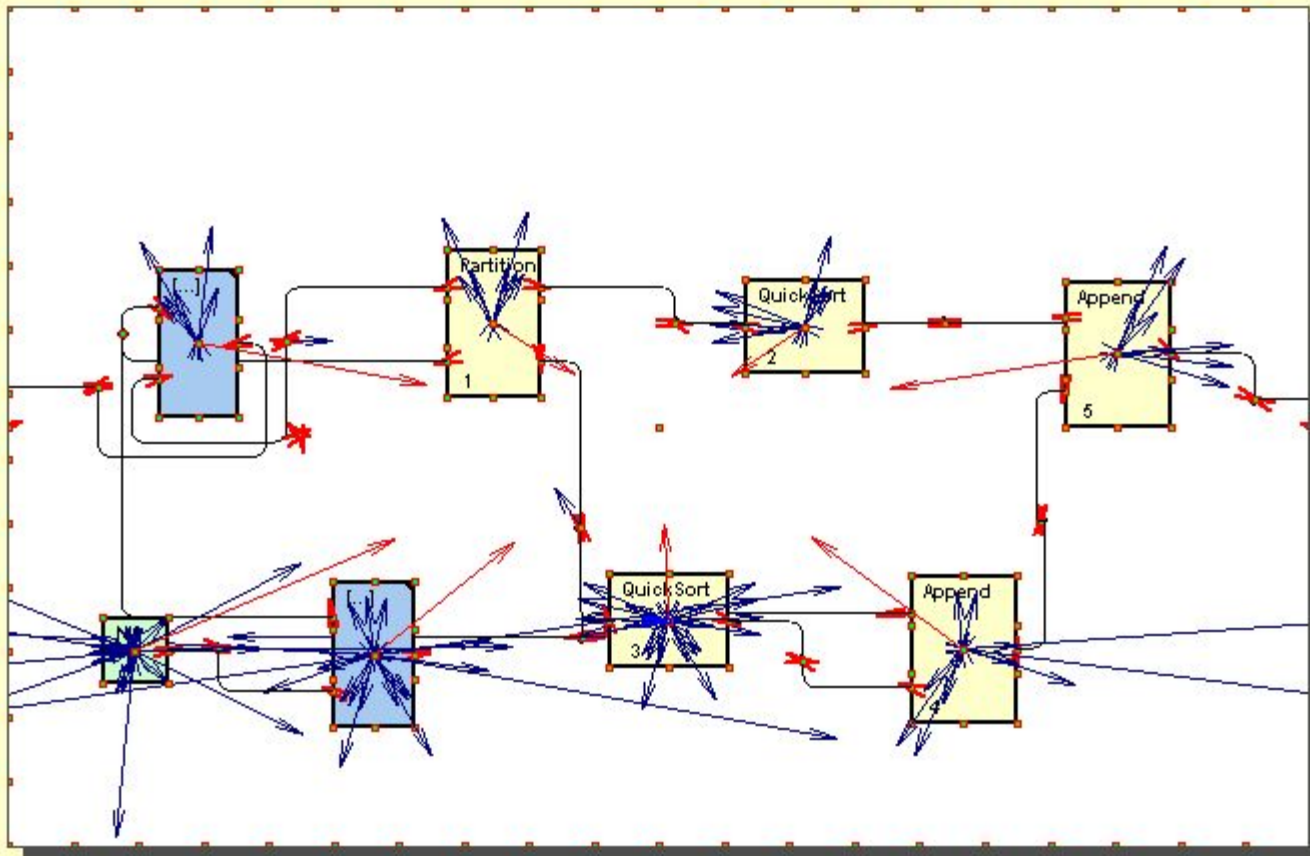
- A green box labeled "Succ" at the top left.
- Two pink boxes labeled "HANOI" with sub-labels "1" and "2" in the center.
- A blue box labeled "to" on the left side.
- A blue box labeled "[...]" on the right side.
- A yellow box labeled "append" with the number "3" below it on the far right.

Connections between boxes are made using black lines with small circular nodes at the connection points. The flow starts from the left, goes through the "to" box, then splits to connect to both "HANOI 1" and "HANOI 2". From "HANOI 1", the flow goes to "Succ" and then to "append". From "HANOI 2", the flow goes to "[...]" and then to "append".

The bottom status bar displays the following text:

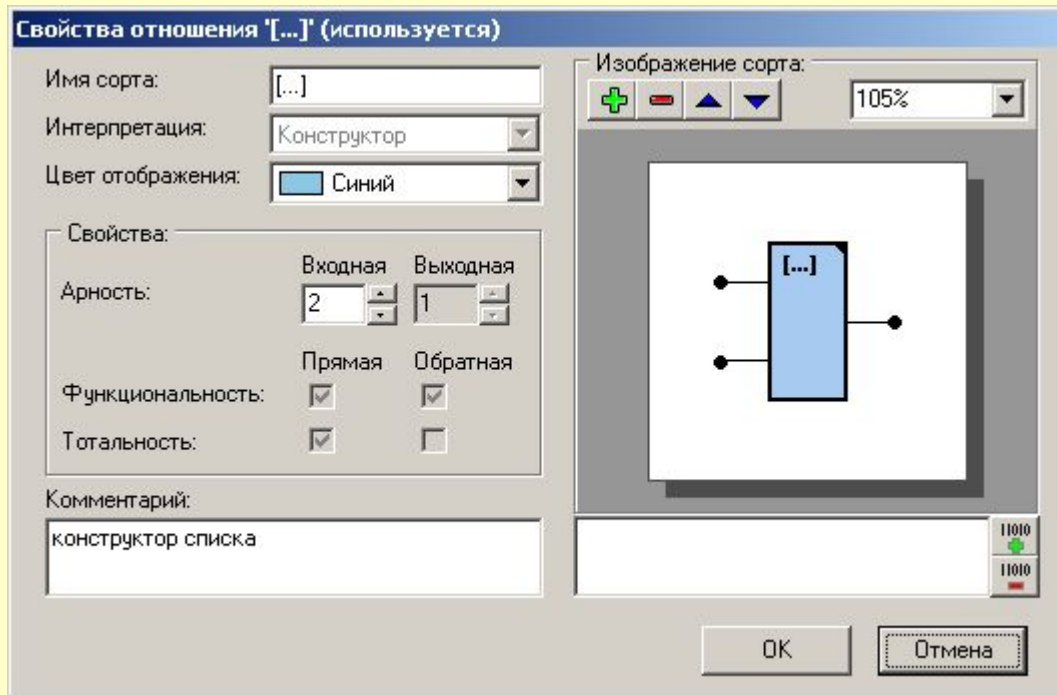
Начало вычисления...
Результат 1:
Выход 1: [to(A,C),to(A,B),to(C,B)]
Вычисление завершено за 0.005 секунд (9 шагов)

Расстановка объектов сети

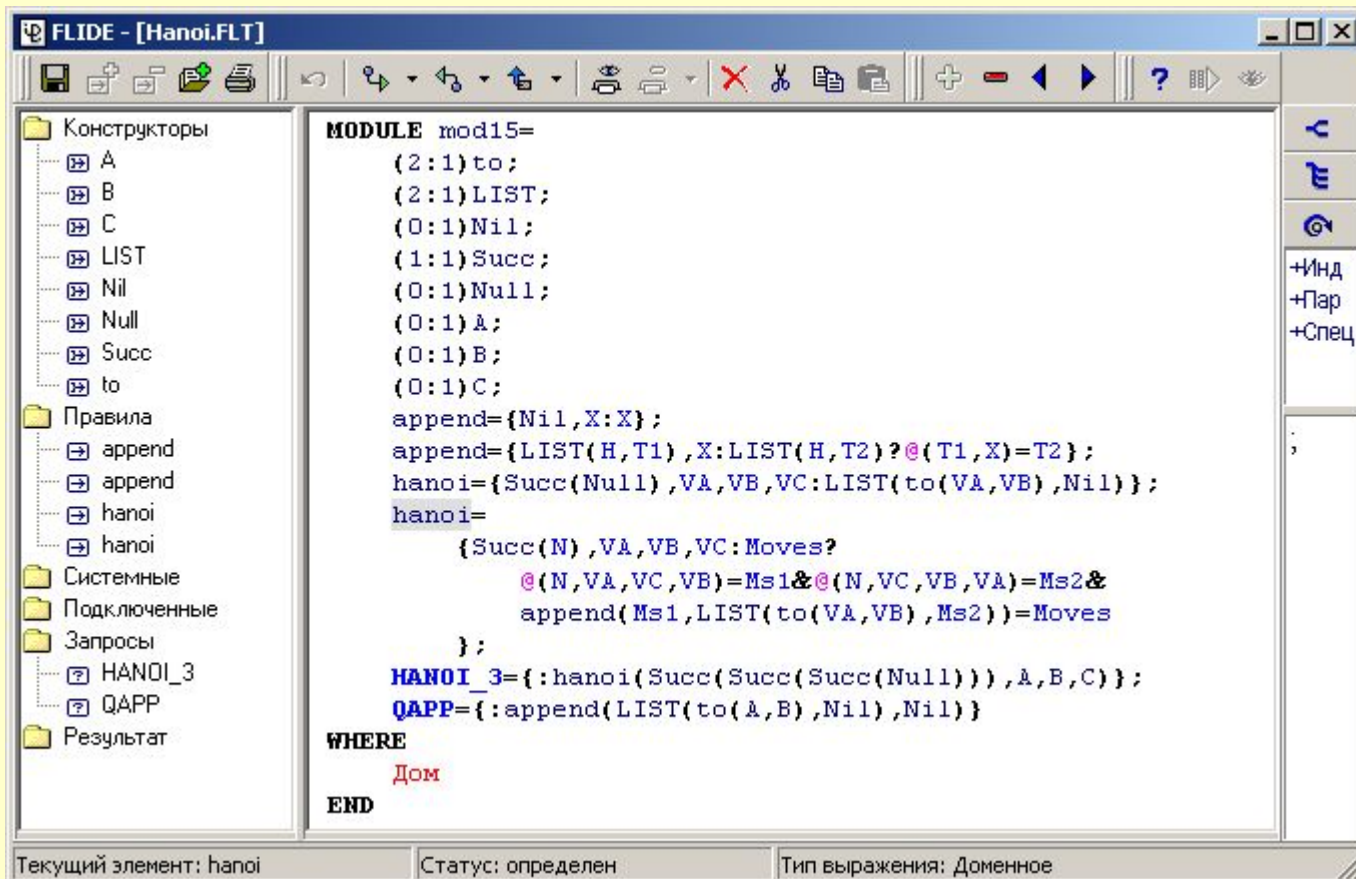


Определение свойств НО

Пример: конструктор списка [...] ^(2,1)



Текстовый редактор



Инструменты ввода

The screenshot displays the FLIDE software interface with the following elements:

- Left Panel (Tree View):**
 - Конструкторы
 - Null
 - Succ
 - Правила
 - Add.1
 - Add.2
 - Mult.1
 - Mult.2
 - Nat0
 - Nat2
 - Nat3
 - Power.1
 - Power.2
 - Системные
 - Подключенные
 - Запросы
 - Nat10
 - QAdd
 - QMult
 - QPower
 - Результат
- Main Editor (Code):**

```

MODULE Arithm=
(0:1)Null;
(1:1)Succ;
Nat3={:Succ(Succ(Succ(Null)))};
Nat2={:Succ(Succ(Null))};
Nat0={:Null};
Add={Null,x:x};
Add={Succ(x),y:Succ(@(x,y))};
Mult={Null,x:Null};
Mult={Succ(x),y:Add(Терм)};
Power={x,Null:Succ(Null)};
Power={x,Succ(x):Mult(x,@(x,y))};
WHEN
END
          
```
- Context Menu (Over 'Терм'):**
 - Переменная терма
 - Анонимная переменная
 - Вызов отношения** (highlighted)
 - Скрыть
 - Скрыть все
 - Свернуть
 - Очистить
 - Конструкция IF
 - Конструкция CASE
 - Сетка
- Callout Boxes and Tools:**
 - Списки:** Points to the top toolbar.
 - IF:** Points to the IF button in the right toolbar.
 - Свертка:** Points to the macro button in the right toolbar.
 - Опции:** Points to the +Инд, +Мод, +Тар, +Спец, -Терм buttons in the right toolbar.
 - Операции:** Points to the main editing tools in the right toolbar.
 - Выбор альтернативы:** Points to the context menu.
- Status Bar (Bottom):**
 - Текущий элемент: Терм
 - Статус: не определен
 - Тип выражения: Терм

Ввод выбором альтернатив

Текст программы:

```
MODULE Common=  
  Дом  
END
```

Выбор альтернативы:

Определение:

Спец [СпИнд]ИдОтн[СпПар]=Рел

Конструктор:

Спец [СпИнд]ИдОтн

```
MODULE Common=  
  Спец [СпИнд]ИдОтн[СпПар]=Рел  
END
```

Вызов отношения:

ИмяОтн

График:

{Терм:Терм?Формула}

```
MODULE Common=  
  Спец [СпИнд]ИдОтн[СпПар]={Терм:Терм?Формула}  
END
```

Ввод идентификаторов

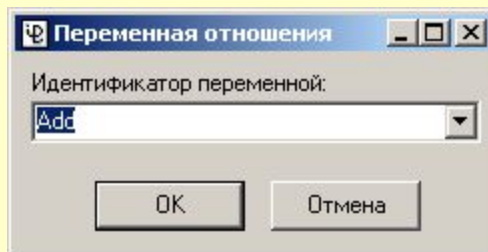
Текст программы:

```
MODULE Common=  
  Спец [СпИнд]ИдОтн[СпПар]=Рел  
END
```

Действие:

Раскрыть:
Текст

Форма ввода:

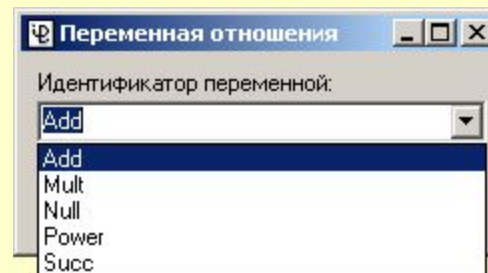


Переменная отношения

Идентификатор переменной:

Add

OK Отмена



Переменная отношения

Идентификатор переменной:

Add

- Add
- Mult
- Null
- Power
- Succ

```
MODULE Common=  
  Спец [СпИнд]Add[СпПар]=Рел  
END
```


Ввод выражений: операции

Текст программы:

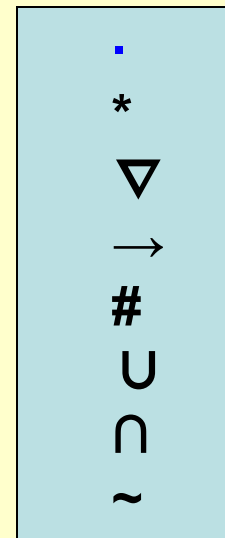
```
MODULE Common=  
  (0:1)[1]Nat=Рел  
END
```

```
MODULE Common=  
  (0:1)[1]Nat=Рел·Рел  
END
```

...

```
MODULE Common=  
  (0:1)[1]Nat=Null·Succ  
END
```

Операция:



Ввод выражений: IF, Свертка.

Текст программы:


```
MODULE Common=  
  (0:1)R=Succ  
END
```

```
MODULE Common=  
  (0:1)R=IF Лог THEN Рел ELSE Succ  
END
```


```
MODULE Common=  
  (0:1)R=(ИдСв=СпЗнач)Succ  
END
```

Действие:

Условная
конструкция IF

Кнопка: 

Свертка

Кнопка: 

Автоструктурирование текста

