# ПРОМЕЖУТОЧНОЕ ПРЕДСТАВЛЕНИЕ ФУНКЦИОНАЛЬНОГО ЯЗЫКА ПОТОКОВОГО ПРОГРАММИРОВАНИЯ

**А.В. Редькин**

*КГТУ, Красноярск*

Язык Пифагор является развивающимся языком функционального параллельного программирования. Он создан на основе функциональной потоковой модели [1], которая обеспечивает поддержку вычислений по готовности данных.

Модель вычислений, лежащая в основе языка Пифагор предполагает гибкое изменение стратегии управления для интерпретации программ в зависимости от условий, предоставляемых вычислительной и программной архитектурой. С другой стороны, информационный граф программы, его структуры и основные принципы выполнения по готовности данных не изменяются. Таким образом, для интерпретации программы необходимо выделить ее смысловую часть, описываемую моделью вычислений, а для вычислений использовать стратегию интерпретации, наиболее подходящую для решения поставленной задачи.

Промежуточное представление языка Пифагор разрабатывалось с целью четкого выделения информационного графа и всего необходимого для последующего наложения управляющего графа и интерпретации или компиляции. Каждая функция транслируется в промежуточное представление отдельно и для нее формируется массив внешних связей с типами и функциями. Промежуточное представление функции и ее связи помещаются в репозитарий, откуда среда интерпретации или компоновщик может запросить нужную функцию со всеми ее связанными функциями и типами (рис. 1).

На этапе трансляции функция проходит проверку на синтаксическую и семантическую правильность. На этапе помещения в репозитарий происходит проверка на наличие всех необходимых для выполнения функций. Таким образом, системе интерпретации не нужно делать лишних проверок, что экономит ценное время в ходе сборки и интерпретации программы.

Информационный граф для функции формируется в виде таблицы вершин, для каждой вершины в этой таблице существует таблица связанных с ней вершин, то есть вершин, результаты вычисления которых используются данной вершиной. Соответственно, константы не имеют свя-

занных вершин, узлы интерпретации имеют две связанные вершины, списки в качестве связанных вершин имеют составляющие их элементы.

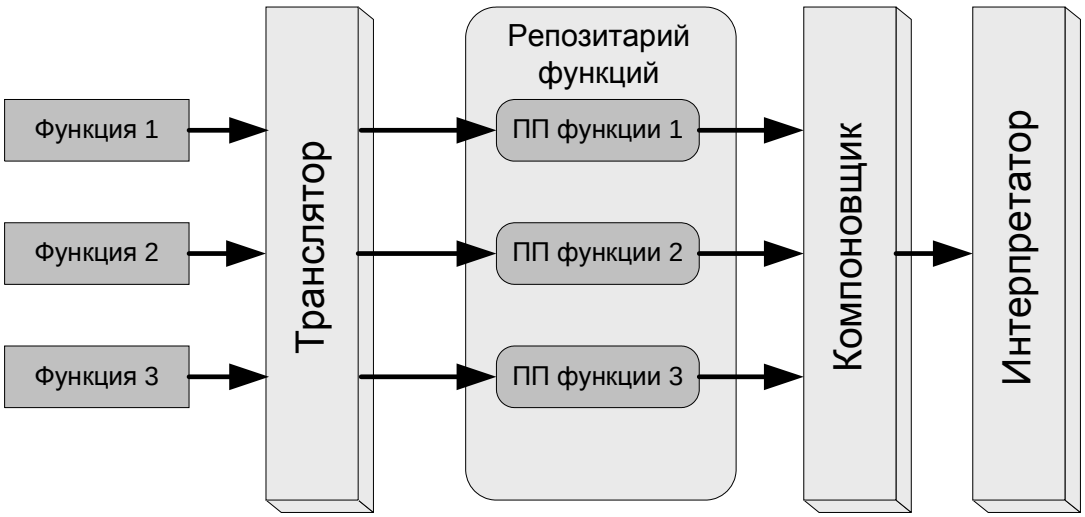


Рис.1. Схема использования промежуточного представления

Связывание вершин производится на основе уникальных идентификаторов. На рис. 2 представлен информационный граф выражения [(x,1),(x,2)]:+, а на рис. 3 схематично представлено, как будет выглядеть промежуточное представление такого графа.

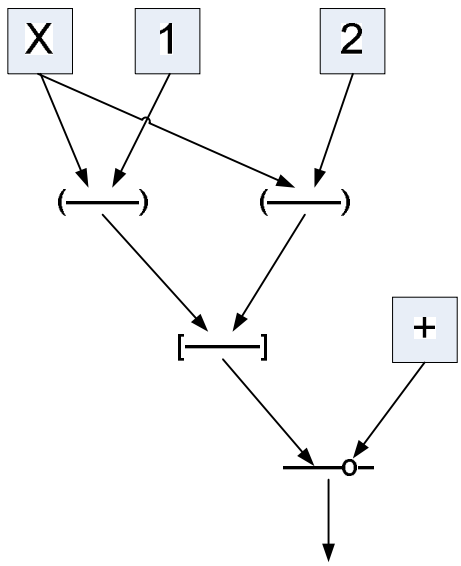


Рис. 2. Информационный граф
выражения [(x,1),(x,2)]:+

Создание промежуточного представления языка Пифагор является первым шагом к поддержке модульных программ. Также данная разработ-

ка предполагает создание трансляторов и систем исполнения программ, совместимых на уровне промежуточного представления.

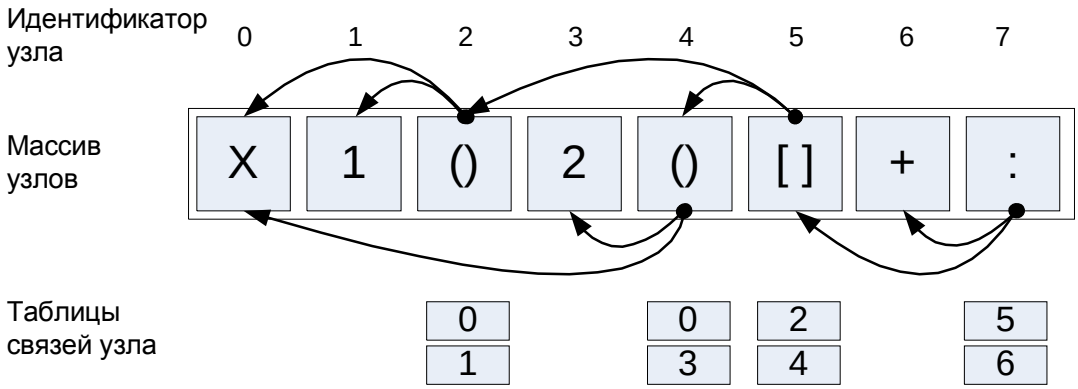


Рис. 3. Представление информационного графа в виде массива узлов

Современная поддержка модульного программирования, как правило, включает возможности использования глобальной коллекции функций или классов, доступных компилятору и системе интерпретации. Популярные системы переносимого программирования, такие как Java и Microsoft.NET, предоставляют программистам такие возможности. Использование глобального хранилища функций на компьютере или в сети позволяет упростить разработку, компиляцию и исполнение модульных программ.

Репозитарий функций в общем случае позволяет организовать модульность программ независимо от используемых систем трансляции и интерпретации и исполнения программ, что схематично изображено на рис. 4.

В качестве эксперимента разработаны структуры промежуточного представления и библиотека для работы с ним, разработана структура репозитария функций, реализован транслятор из языка Пифагор. Во всех программных разработках использован стандартный язык C++, что позволяет говорить о хорошей переносимости программ на многие архитектуры.

Промежуточное представление позволяет упростить создание систем интерпретации, а также является основой для поддержки модульного программирования. Работа в дальнейшем позволит более детально описать модель вычислений, ее динамику функционирования и стратегии управления.

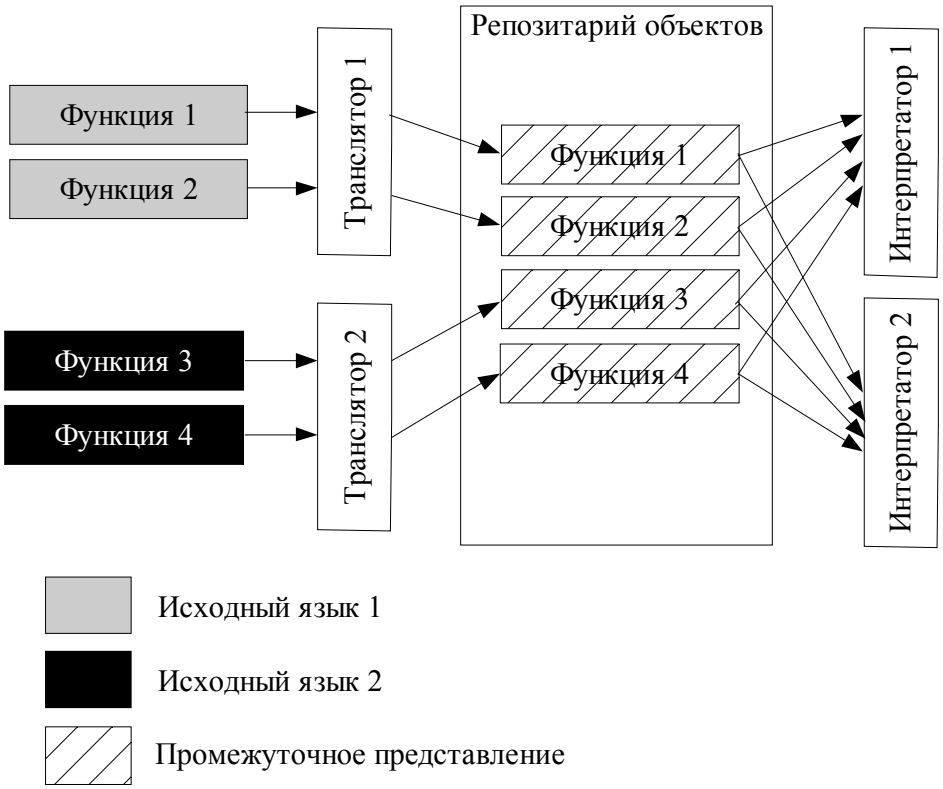


Рис. 4. Использование репозитария

В настоящее время работа над промежуточным представлением потокового языка параллельного программирования продолжается, разрабатывается представление таких элементов, как асинхронный список, которые позволят более точно описать информационный граф.

*Литература*

1. Легалов А.И. Функциональный язык для создания архитектурно-независимых параллельных программ // Вычислительные технологии. − 2005. − Т.10, № 1. − С. 71−89.
2. Редькин А.В. Промежуточное представление информационного графа для языка Пифагор // Информатика и информационные технологии: Материалы межвузовской научной конференции. − Красноярск: ИПЦ КГТУ, 2004. − С. 52.