А.А. Вылиток

Об алгоритмической полноте некоторых языков программирования

В данной заметке рассматривается вопрос, являются ли алгоритмически полными следующие языки программирования:

1) машинный язык для УМ-3 (трехадресная учебная машина);

2) язык ассемблера (например, MASM без макросов);

3) язык Паскаль (стандарт ISO);

4) язык Си (стандарт ISO).

Алгоритмическая полнота языка (или полнота по Тьюрингу) предполагает, что на нем можно выразить любой алгоритм, представимый машиной Тьюринга. Для доказательства полноты языка по Тьюрингу достаточно показать, как на данном языке смоделировать работу машины Тьюринга. Отметим, что кроме моделирования функциональных возможностей машины по чтению, записи, сдвигу головки на ленте и смены состояния, важно, чтобы язык никак не ограничивал потенциальную длину ленты, поскольку известно, что существуют задачи сколь угодно ёмкие по памяти (хотя конкретная реализация языка вправе накладывать ограничения, но это будут ограничения реализации, а не самого языка). Иными словами, в языке должна присутствовать абстракция потенциальной бесконечности.

Машинный язык для УМ-3 не является алгоритмически полным. Это следует из того, что формат команд УМ-3 предполагает адреса фиксированного размера, а значит размер адресного пространства (памяти) ограничен самим языком и потенциально бесконечную ленту в памяти смоделировать невозможно. Можно добавить в язык команды ввода-вывода, что принципиально ничего не изменит, поскольку вычислитель УМ-3 не контролирует внешние данные (не управляет устройствами, на которых хранятся эти данные). Чтобы достичь полноты, можно превратить УМ-3 в разновидность машины Тьюринга, добавив в архитектуру УМ-3 потенциально бесконечную внешнюю ленту и команды для движения головки по ленте, чтению и записи значений в ячейки ленты из памяти и обратно в память.

Ситуация с языком ассемблера аналогична — он рассчитан на архитектуру с конечной памятью.

Язык Паскаль позволяет смоделировать ленту машины Тьюринга с помощью двунаправленного списка из переменных, создаваемых оператором ***new***, семантика которого не предполагает отказа в создании переменной. (Также с помощью списков можно смоделировать сколь угодно большие числа.) Стандарт не накладывает никаких ограничений: указательный тип абстрактен (слово «адрес» не встречается в тексте Стандарта ни разу), множество значений указательного типа языком не ограничено. В Паскале есть еще один тип данных с неограниченным множеством значений, файловый, также пригодный для моделирования ленты машины Тьюринга и представления больших чисел. Следовательно, язык Паскаль алгоритмически полон.

В языке Си нет высокоуровневого понятия переменной (в смысле Паскаля), есть объекты (***object***), хранящиеся в памяти как последовательно расположенные байты,

имеющие адрес (байты в свою очередь состоят из неадресуемых битов). Целые типы ограничены (конечное множество значений), указатель отождествляется с адресом, постулируется возможность хранить адрес в целочисленной переменной (***int*** или ***long*** — зависит от реализации), откуда следует ограниченность множества значений указателей, а стало быть, и ограниченность адресного пространства Си-машины. То есть язык Си, как и язык ассемблера, ориентирован на архитектуру с конечной памятью. Файл не является типом данных языка Си, в отличие от Паскаля. Это вещь из окружения, для работы с которой есть операции над потоками в виде набора библиотечных функций. Тип *fpos_t*, принятый в стандарте Си для позиционирования файлов, постулируется как «отличный от массива тип данных (***object type***)». Следовательно, множество значений этого типа конечно, а значит, максимальная длина файла в языке Си ограничена сверху.

Для иллюстрации понятия алгоритмического языка в курсе «Алгоритмы и алгоритмические языки» более подходит язык Паскаль, ибо он не ограничен какой-либо машинной архитектурой (и его алгоритмическая полнота не вызывает сомнений), в отличие от Си, явно нацеленного на архитектуру современных вычислительных машин, обусловливающую семантические ограничения множества значений типов данных. Язык Си по своей сути — «высокоуровневый ассемблер»: по форме он похож на Паскаль, а по смыслу — на язык ассемблера. Такой язык хорошо подходит для задач курса «Операционные системы», когда студенты уже знакомы с «Архитектурой ЭВМ и языком ассемблера».

Можно указать и другие алгоритмически полные языки, например, Лисп, Пролог, Рефал, Хаскель и им подобные, «пришедшие из математики».