

Секция «Дискретная математика и математическая кибернетика»
С-подобная версия синтаксиса языка ЛОС

Перпер Евгений Михайлович

Аспирант

Московский государственный университет имени М.В.Ломоносова,
Механико-математический факультет, Кафедра математической теории
интеллектуальных систем, Москва, Россия

E-mail: e_m_perper@mail.ru

В работе рассматривается алгоритмический язык ЛОС (Логический Описатель Ситуаций), созданный А.С. Подколзиным для записи приёмов Компьютерного решателя математических задач [1].

ЛОС — достаточно мощный язык; многие программы на нём записываются значительно короче, чем, например, на С. Но основное значение этого языка заключается в том, что на этом языке написан решатель математических задач — программа, аналогов которой нет в мире, но которая, тем не менее, мало известна из-за трудностей, возникающих при её изучении. В том числе и язык ЛОС не слишком прост для изучения. Также большие проблемы возникают при необходимости добавить в язык новые операторы. Авторы работы задались следующей целью: сделать синтаксис ЛОСа более похожим на синтаксисы языков, которые уже известны человеку, решившему изучать Решатель и язык ЛОС. При этом требовалось полностью сохранить функциональность языка.

В изначальной версии языка ЛОС программа представляет из себя множество функций, каждая из которых привязана к *логическому символу* и видна из всех остальных процедур. Структура программы на ЛОСе представляет собой дерево. Каждая ветвь этого дерева являются последовательностью операторов. Результат выполнения любого оператора — истина либо ложь, причём переход к выполнению следующего по порядку оператора происходит только в том случае, если этим результатом оказалась истина. Исключение — оператор *иначе(N)*, который выполняется только тогда, когда предыдущий оператор выдал ложь. С помощью оператора *иначе(N)*, а также оператора *ветвь(N)*, осуществляется переход на новую ветвь (N - номер ветви, на которую происходит переход).

Каждый оператор ЛОСа — это терм, записанный в скобочном виде. Вот пример оператора: *равно(x1 плюс(x2 x3))*. Если значение $x1$ уже было определено перед вызовом этого оператора, то он возвращает истину, когда $x1 = x2 + x3$, и ложь в противном случае. Если же значение $x1$ не было определено, то переменной $x1$ присваивается сумма $x2$ и $x3$, а оператор возвращает истину. Редактор программ на ЛОСе встроен в Решатель; каждая ветвь программы показывается в отдельном окне. Операторы, принадлежащие одной ветви, записываются подряд через пробел. При этом программа хранится в бинарном формате, из-за чего нет возможности ее редактировать в обычных редакторах, а также полноценно использовать системы контроля версий.

В результате, чтобы сделать язык ЛОС более привычным, авторы разработали для него С-подобный синтаксис. Можно выделить, например, следующие отличия новой версии синтаксиса от той, что используется в Решателе.

- 1) Оператор теперь отделён от следующего оператора из последовательности точкой с запятой.
- 2) Для многих операторов были предложены альтернативные названия на английском языке (при этом названия этих операторов на русском языке использовать по-прежнему можно).
- 3) Переменные в этой версии можно именовать любым набором из букв и цифр, начинающимся с буквы (изначально в ЛОСе переменные называются $x1$, $x2$, $x3$ и т.д.,

причём для новой переменной всякий раз крайне желательно использовать наименьший из незанятых номеров);

- 4) Там, где есть возможность использовать знаки арифметических операций и отношений, используются именно они. Например, упомянутый выше терм *равно*($x1$ *плюс*($x2$ $x3$)) записывается как $x1 = x2 + x3$.
- 5) Любая программа на ЛОСе со всеми её ветвями целиком записывается в одном файле, причём этот файл можно редактировать в любом стандартном текстовом редакторе. Ветвь, окружённая фигурными скобками, записывается после соответствующего оператора *ветвь* или *иначе* (сами эти операторы теперь называются *branch* и *if_false*).
- 6) Добавлен оператор *if*($f1$) $f2$ *else* $f3$. Здесь $f2$ и $f3$ — составные операторы, записывающиеся как последовательность операторов, окружённая фигурными скобками (если последовательность состоит из одного оператора, фигурные скобки можно не писать). Фактически он представляет собой имеющийся в ЛОСе оператор *альтернатива*($f1$ $g2$ $g3$), который выполняет оператор $g2$, если оператор $f1$ истинен, и выполняет $g3$ в противном случае. Здесь $g2$ — это конъюнкция всех операторов из последовательности, соответствующей $f2$, а $g3$ — конъюнкция всех операторов из последовательности, соответствующей $f3$.
- 7) Оператор *длялюбого*($x1...xn$ *если* $f1...fm$ *то* $f0$), который истинен, если при любых значениях переменных $x1, \dots, xn$, для которых истинны операторы $f1, \dots, fm$, истинен и оператор $f0$, теперь можно записать как оператор цикла *foreach*($x1, \dots, xn$: $f1, \dots, fm$) $f0$, где $f0$ может быть составным оператором, как в языке С.
- 8) Теперь можно создавать локальные функции, видимые только в том файле, где они объявлены.
- 9) Для добавления нового оператора в интерпретатор ЛОСа теперь достаточно написать обычную функцию на С++, реализующую этот оператор, и зарегистрировать её одной командой. В изначальной версии пришлось бы вручную извлекать из битового представления типы и поля объектов, изменять указатель на вершину стека интерпретатора и производить множество других действий, в которых легко ошибиться, но очень трудно потом найти ошибку.

Все синтаксические правила новой версии ЛОСа были описаны формальной грамматикой, и с помощью утилит *flex* и *bison* для неё были созданы лексический и синтаксический анализаторы. Далее для этой версии был написан интерпретатор, а также программа, переводящая программы Решателя из бинарного формата в новый синтаксис ЛОСа. Все программы на ЛОСе, в том числе скомпилированные приемы, были переведены в новый формат. Новая версия ЛОСа была успешно протестирована на математических задачах из базы данных Решателя. Таким образом, изменения в языке ЛОС, сделавшие его внешне похожим на С, сохранили его функциональность. Тем самым, основные задачи, стоявшие перед авторами, были выполнены.

Источники и литература

- 1) Подколзин А.С. Компьютерное моделирование логических процессов. Москва, 2007.

Слова благодарности

Автор выражает благодарность Глебу Вячеславовичу Калачёву как соавтору проекта, Александру Сергеевичу Подколзину за помощь в изучении языка ЛОС и Эльяру Эльдаровичу Гасанову за полезные идеи.