Применение методов поискового конструирования к разработке алгоритмов поиска числового ключа в файлах.

Кащеев Д.И., Костерин В.В.

Волгоградский государственный технический университет

Актуальность

Параллельное развитие теории программирования и теории проектирования сделало актуальным их системное исследование.

В настоящее время программирование трансформировалось в целую индустрию производства программных изделий. Основные должности программистов: техник-программист, инженер-программист (третьей, второй и первой категорий). Следовательно, профессиональный разработчик программных изделий должен владеть теорией проектирования, методами активизации мышления.

Поэтому представляется актуальным исследование возможности применения методов системного анализа, в частности, поискового конструирования, в задачах программирования.

Цель исследования.

Применить методы поискового конструирования при решении задачи разработки алгоритма поиска ключа в упорядоченном файле.

Основная часть

При успешном решении какой-либо творческой задачи, человек получает два результата: это само решение поставленной задачи и методический опыт, т.е. уяснение процесса решения данной конкретной задачи. Такие методические правила называют эвристическими приемами.

Эвристический прием – способ разрешения определённого противоречия. Но проблема заключается в том, что решение одной задачи нельзя просто перенести на решение другой. Поэтому только после решения определенного числа задач у человека появлялся набор правил, указаний или приемов решения той или иной задачи.

Методы технического (инженерного) творчества подразделяются на две группы:

1) эвристические методы технического творчества - основаны на использовании достаточно четко описанных методик и правил поиска новых технических решений.

2) компьютерные методы поискового конструирования - основаны на использовании ЭВМ в решении творческих инженерных задач.

Значительной популярностью среди эвристических методов пользуются метод морфологических таблиц и методы аналогий.

Рассмотрим возможность применения методов аналогий и метода морфологических таблиц при построении новых алгоритмов поиска в файле. Будем считать файл отсортированным по численному ключу (что справедливо в большинстве практически важных приложений).

Можно выделить такие свойства алгоритма поиска:

1) детерминированность/стохастичность - каждое действие выполняется по определенным правилам либо случайно выбирается из совокупности правил;

2) метод простой/составной – является ли он самостоятельным или использует другие методы поиска;

3) обучаемость - метод учитывает результативность предыдущих попыток или не учитывает.

За критерий эффективности алгоритма выберем количество операций доступа к файлу при выполнении поиска ключа.

Рассмотрим целесообразность выбора каждого из качеств алгоритма. Очевидно, что правильно построенный обучаемый алгоритм с большей вероятностью будет более эффективным, чем алгоритм, не учитывающий предыстории. Обучаемость алгоритма означает, что алгоритм будет использовать ту или иную последовательность действий в зависимости от предыдущих попыток. Естественно ожидать, что в качестве последовательностей действий следует выбрать группу наиболее эффективных из известных алгоритмов, т.е., метод следует построить составным. Процесс обучения (ведения истории) может происходить по детерминированному закону или стохастично. Однако, так как заранее неизвестны параметры файла, целесообразно создать алгоритм, подстраивающийся под конкретный файл. Автору представляется естественным использовать такое стохастическое правило: вызывать каждый метод случайным образом, но вероятность вызова распределять так, чтобы более быстрый для данного файла метод вызывался чаще.

Таким образом, используя морфологический синтез, автор построил комбинированный метод, суть которого состоит в следующем:

1) сначала полагаем, что вероятности вызова всех методов равны;

2) при каждой операции поиска в файле данного объема накапливаем статистику;

3) Для обеспечения приоритетности более быстрого метода по сравнению с остальными следует пересчитывать вероятности;

Проведенные численные эксперименты доказали, что при различных распределениях значений ключей в файле алгоритм действительно на этапе выбора метода выбирает наискорейший для данного файла.

Заключение

Методы поискового конструирования находят широкое применение, как при решении задач программирования, так и при решении научных задач. На конкретном примере доказана эффективность применения методов поискового конструирования к решению задач программирования, созданный метод поиска ключа показал эффективность при имитационном моделировании.