

Обнаружение недостижимых и конечных уроков в компьютерном учебном курсе

О. С. Естехин

Компьютерный учебный курс можно представить в виде набора уроков, каждый из которых обладает определенным состоянием, например результатом прохождения этого урока пользователем. В таком случае состояние всего учебного курса является вектором $P = (p_1, p_2, \dots, p_n)$, где $p_i \in E^k$ — состояние отдельного урока, принимающее одно из k значений. Исходное состояние учебного курса $P_0 = (0, 0, \dots, 0)$.

Существует несколько вариантов задания последовательности прохождения учебного курса. В общем случае можно считать, что для конкретного учебного курса существует функция переходов $T(P) = \{P_i\}$, которая возвращает множество состояний, в которые можно перейти из указанного. Причем любое следующее состояние P_i отличается от исходного значением ровно одной компоненты, т. е. значением состояния одного урока.

В такой постановке компьютерный учебный курс представляет собой конечный автомат.

При разработке учебного курса, кроме непосредственного создания учебных материалов, необходимо так же проверить правильность задания правил прохождения. Двумя возможными проверками являются проверки, что все уроки достижимы, т. е. для каждого урока существует хотя бы один вариант прохождения учебного курса, при котором этот урок проходит пользователем, и что все варианты прохождения учебного курса завершаются уроком, принадлежащим заданному множеству конечных уроков.

Для нахождения как недоступных, так и конечных уроков можно применить следующий алгоритм, основанный на алгоритме двустекового моделирования недетерминированного конечного автомата:

Вход: Функция переходов $T(P)$, множество исходных состояний учебного курса S (обычно $S = \{P_0\}$).

Выход: Множество конечных уроков L_f и множество недостижимых уроков L_n .

Метод: На каждом шаге алгоритма необходимо вычислить множество следующих состояний (обозначим его N), доступных из какого-либо состояния из множества текущих состояний (обозначим его C). Изначально $C = S$, а L_n содержит все уроки. Конечные состояния заносятся в изначально пустое множество F .

Одна внутренняя итерация алгоритма состоит из следующих действий:

- Если C пустое, то итерации завершены.
- $N = \{\}$.

- Для каждого состояния курса $P_c \in C$.
 - Все уроки, состояние которых (компонент p_i из вектора P_c) отличается от исходного, удаляются из множества L_n .
- Для каждого состояния курса $P_c \in C$ вычисляется $T(P_c) = N_c$.
 - Если N_c пустое, то состояние курса P_c конечное. Добавляем P_c в F .
 - Если N_c не пустое, то добавляем все его компоненты в N .
- $C = N$.

После завершения итераций во множестве L_n останутся те уроки, состояние которых не менялось. Недостижимые уроки указывают на то, что при составлении правил прохождения учебного курса допущена ошибка. Зная, какой именно урок недостижим, разработчик учебного курса может вручную найти и исправить ошибку в правилах перехода между состояниями учебного курса. Автоматическое обнаружение подобных ошибок является предметом дальнейшего исследования.

Во множестве F будут находиться такие состояния учебного курса, из которых невозможно перейти в какие-либо другие состояния. Если при внесении какого-либо состояния курса P_f во множество F также сохранять предыдущее состояние P_p (т. е. $P_f \in T(P_p)$), то можно определить, изменение состояние какого именно урока привело к переходу в конечное состояние курса. Таким образом, можно заполнить множество L_f . Разработчик учебного курса может проверить, во-первых, что все уроки, которые завершают учебный курс, действительно находятся во множестве конечных уроков, а во-вторых, что не существует тупиковых уроков, которые не завершают учебный курс, но из которых пользователь тем не менее не может продолжить изучение курса.

К недостаткам предложенного алгоритма относится невозможность использования для учебного курса, содержащего циклы. Модификация алгоритма для обработки подобных учебных курсов является предметом дальнейшего исследования.

Литература

1. *Естехин О. С.* Математическая модель учебного курса // Труды XLVIII науч. конф. МФТИ «Современные проблемы фундаментальных и прикладных наук». Ч. VIII. МФТИ, 2005.
2. *Aho, Sethi, Ullman.* Compilers: Principles, Technologies and Tools. Addison Wesley, 1986.