

Опубликовано в материалах 2-й межвузовской научной конференции по проблемам информатики СПИСОК-2011, с. 366-367.

Д. А. Парашенко, А. С. Станкевич

*Санкт-Петербургский государственный университет
информационных технологий, механики и оптики*

Суффиксные автоматы с сохранением промежуточных версий и их приложения

Структуры данных, которые хранят свои промежуточные версии, имеют более широкий круг приложений, чем их варианты без такой возможности [1–4]. Классическим примером является задача нахождения области, содержащей заданную точку [1]. Она может быть элегантно решена с помощью применения дерева поиска с сохранением промежуточных версий.

Не существует общего механизма, позволяющего после совершения какие-либо модификаций над обычной структурой данных вернуться к ее предыдущей версии. Структуры данных, хранящие свою хронологию, называются персистентными.

Существует несколько видов персистентных структур данных [1]. Частичная персистентность позволяет модифицировать последнюю версию структуры данных и делать запросы к любой из промежуточных версий. Полная персистентность позволяет совершать запросы и модификации с любой версией структуры данных. При таком типе персистентности версии образуют не линейную, а древовидную структуру. В обоих случаях доступ может осуществляться по номеру требуемой версии.

В настоящее время для решения множества строковых задач используются суффиксные деревья, суффиксные массивы и суффиксные автоматы [5–9].

Также известны методы, позволяющие делать суффиксные деревья персистентными. В свете работы [10] возникает вопрос о возможности эффективной реализации персистентных суффиксных автоматов.

Рассмотрены существующие методы модификаций структур данных с целью добавления в них возможности сохранения промежуточных версий применительно к суффиксному автомату. Предложен метод, позволяющий с небольшими дополнительными затратами сделать суффиксный автомат персистентным.

Свойство персистентности позволяет суффиксному автомату поддерживать помимо операции добавления символа операцию удаления последнего символа. При этом последняя операция не вызывает никаких дополнительных временных затрат.

Кроме того, персистентность позволяет использовать суффиксные автоматы в многопоточных приложениях, в которых предъявляются особые требования к быстродействию. Применение персистентной структуры данных позволяет выполнять операции по модификации суффиксного автомата без блокировки операций чтения. В случае суффиксных автоматов это является достоинством, так как модификация суффиксного автомата может занимать до $O(n)$ времени [6].

В работе выполнен анализ существующих методов построения персистентных структур данных, а также разработан эффективный метод построения персистентного суффиксного автомата. При использовании предлагаемого метода доступ к прежним версиям суффиксного автомата в большинстве случаев не требует значительных дополнительных временных затрат, однако, для некоторых видов строк верхняя оценка времени выполнения запросов может увеличиться в $O(\log n)$ раз.

Предложен метод, позволяющий поддерживать множество допускающих состояний суффиксного автомата в процессе его построения. При этом требуется $O(\log n)$ дополнительного времени и $O(1)$ дополнительной памяти на каждый символ базовой строки суффиксного автомата. В связи с тем, что описанный метод основывается на использовании древовидной структуры данных, можно без дополнительных затрат хранить промежуточные версии множества допускающих состояний суффиксного автомата, тем самым, сделав упомянутое множество персистентным.

Источники

1. *Karger D.* Persistent Data Structures. Advanced Algorithms: Lecture 2. September 9, 2005.
2. *Bentley J. L., Saxe J. B.* Decomposable searching problems I: Static-to-dynamic transformations // J. Algorithms. 1980. Vol 1, pp. 301-358.
3. *Chazelle B.* Filtering search: A new approach to query-answering // SIAM J. Comput. 1986. Vol. 15, pp. 703-724.
4. *Chazelle B.* How to search in history // Inform. and Control. 1985. Vol 77, pp. 77-99.
5. *Гасфилд Д.* Строки, деревья и последовательности в алгоритмах. Информатика и вычислительная биология. СПб.: Невский диалект; БХВ-Петербург, 2003.
6. *Lothaire M.* Applied Combinatorics on Words // Encyclopedia of Mathematics and its Applications, 2005. Vol. 90. Cambridge University Press, Cambridge.
7. *Хопкрофт Дж., Мортвани Р., Ульман Дж.* Введение в теорию автоматов, языков и вычислений. М.: Вильямс, 2002.
8. *Кормен Т., Лейзерсон Ч., Ривест Л.* Алгоритмы: построение и анализ. М.: МЦНМО, 2000.
9. *Sartaj Sahni Dr.* Data Structures, Algorithms, \& Applications in Java. Suffix Trees. CISE Department Chair

at University of Florida

<http://www.cise.ufl.edu/~sahni/dsaaj/enrich/c16/suffix.htm>

10. *Паращенко Д.* Обработка строк на основе суффиксных автоматов. Бакалаврская работа. СПбГУ ИТМО. 2007.