

достижимо в автомате $\mathbf{A}$ из состояния s , если найдется входное слово $p \in X^*$, такое, что $\delta(s, p) = t$. Записывая это в виде $(s, t) \in \tau$, вводим отношение достижимости τ в автомате $\mathbf{A}$.

Симметричная часть $\sigma = \tau \cap \tau^{-1}$ отношения достижимости называется отношением взаимной достижимости в $\mathbf{A}$. Классы этой эквивалентности называют слоями автомата $\mathbf{A}$. Каркасом автомата $\mathbf{A}$ назовем упорядоченное множество $F(\mathbf{A}) = (S/\sigma, \tau^{-1})$. Его элементами являются слои автомата $\mathbf{A}$, а порядком — отношение, обратное достижимости, перенесенное на слои: $(\sigma(t), \sigma(s)) \in \tau^{-1}$ равносильно тому, что $(s, t) \in \tau$.

Теорема 1. Каждое конечное упорядоченное множество изоморфно каркасу подходящего автомата с двумя входными сигналами.

Теорема 2. Конечное упорядоченное множество тогда и только тогда изоморфно каркасу автономного автомата, когда у каждого его элемента имеется не более чем один нижний сосед.

Теорема 3. Если $\mathbf{A}$ и $\mathbf{B}$ — произвольные автоматы, то $\text{Sub}\mathbf{A} \cong \text{Sub}\mathbf{B}$ тогда и только тогда, когда $F(\mathbf{A}) \cong F(\mathbf{B})$.

Теорема 4. Если φ — вложение автомата $\mathbf{A}$ в автомат $\mathbf{B}$ и $F(\mathbf{A}) \cong F(\mathbf{B})$, то φ — изоморфизм $\mathbf{A}$ на $\mathbf{B}$.

Следствие 1. Если $\mathbf{A}'$ — подавтомат автомата $\mathbf{A}$ и $F(\mathbf{A}') \cong F(\mathbf{A})$, то $\mathbf{A}' = \mathbf{A}$.

Следствие 2. Если φ — эндоморфизм автомата $\mathbf{A}$ и $F(\varphi(\mathbf{A})) \cong F(\mathbf{A})$, то φ — автоморфизм.

Теорема 5. Если θ — конгруэнция автомата $\mathbf{A}$, то $F(\mathbf{A}/\theta) \cong F(\mathbf{A})$ тогда и только тогда, когда $\theta \subseteq \sigma$.

ЛИТЕРАТУРА

1. Салий В. Н. Автоматы, у которых все конгруэнции — внутренние // Изв. вузов. Математика. 2009. № 9. С. 36–45.

УДК 519.5

КЛАССЫ ГРАФОВ, ВОССТАНАВЛИВАЕМЫЕ С ЛИНЕЙНОЙ ВРЕМЕННОЙ СЛОЖНОСТЬЮ

Е. А. Татаринов

Рассматривается задача [1] восстановления конечного связного неориентированного графа G без петель и кратных ребер при помощи агента, который перемещается по ребрам графа G , считывает и изменяет метки на вершинах и инциденторах. На основе собранной информации агент строит граф H , изоморфный графу G с точностью до меток на вершинах и инциденторах графов. Необходимо найти метод обхода и разметки графа G с целью его восстановления.

Известен ряд методов восстановления графа [2, 3], которые используют не более четырех различных красок, однако имеют верхнюю оценку временной сложности восстановления, равную квадратичной и кубической функцией от числа вершин в восстанавливаемом графе соответственно. Для каждого метода нижней оценкой временной сложности восстановления является линейная функция от числа вершин в графе G . Данная работа посвящена выделению классов графов, для которых верхняя

оценка временной сложности восстановления является линейной функцией, и построению операций над графами, порождающих новый граф, для которого верхняя оценка временной сложности не ухудшается.

Методы основаны на обходе графа «в глубину». Для восстановления ребра агент вычисляет M — номера вершин [4], которым оно инцидентно. Для этого агент выполняет некоторое количество переходов по ранее посещенным вершинам. Этот отход порождает нелинейность верхней границы временной сложности восстановления. Для некоторых классов количество и длина таких отходов могут быть заранее известны и ограничены. Очевидны следующие утверждения.

Теорема 1. Для графа вида «дерево» верхняя оценка временной сложности восстановления является линейной функцией от числа вершин G .

Теорема 2. Для графа вида «кольцо» порождается один отход при восстановлении графа G .

Найдены операции над графами, которые не ухудшают верхнюю оценку $T(G)$ временной сложности восстановления графа.

Теорема 3. Операции добавления в граф висячей вершины и введения вершины в ребро графа не ухудшают верхнюю оценку временной сложности восстановления.

Таким образом, если граф G восстанавливается за линейное время, то и граф, полученный из него при помощи этих операций, восстанавливается за линейное время.

Найдена бинарная операция, в результате которой получается граф, для которого верхняя оценка временной сложности восстановления не превосходит суммы верхних оценок временной сложности восстановления исходных графов.

Теорема 4. Если граф G получен путем соединения двух графов G_1 и G_2 через вершину сочленения, то $T(G) \leq T(G_1) + T(G_2)$.

Данная операция допускает естественное обобщение.

Теорема 5. Если граф G получен путем древовидного соединения графов $G_1, \dots, G_i$ через вершины сочленения, то $T(G) \leq T(G_1) + \dots + T(G_i)$.

Если $G_1, \dots, G_i$ восстанавливаются за линейное время, то G , полученный при помощи этой операции, также восстанавливается за линейное время.

В результате найдены частные случаи классов графов, для которых верхняя оценка временной сложности является линейной функцией от числа вершин в восстанавливаемом графе. Найдены операции, которые не ухудшают верхнюю оценку временной сложности восстановления, и бинарная операция, которая строит граф, для которого верхняя оценка временной сложности не превосходит суммы верхних оценок временной сложности исходных графов.

ЛИТЕРАТУРА

1. Dudek G., Jenkin M. Computational principles of mobile robotic. Cambridge Univ. Press. 2000.
2. Грунский И. С., Татаринев Е. А. Распознавание конечного графа блуждающим по нему агентом // Вестник Донецкого университета. Сер. А. Естественные науки. 2009. Вып. 1. С. 492–497.
3. Грунский И. С., Татаринев Е. А. Алгоритм распознавания графов // Труды Четвертой Междунар. конф. «Параллельные вычисления и задачи управления» РАСО '2008. М.: Институт проблем управления им. В. А. Трапезникова РАН, 2008. С. 1483–1498.

4. Касьянов В. Н., Евстигнеев В. А. Графы в программировании, визуализация и применение. СПб.: Петербург, 2003.

УДК 519.713

ПЕРЕСТРАИВАЕМЫЕ АВТОМАТЫ С ОБЩЕЙ ПАМЯТЬЮ¹

В. Н. Тренькаев

Одним из требований к современным цифровым устройствам является гибкость, т.е. возможность внесения изменений в алгоритм функционирования, реализуемая с помощью соответствующей настройки устройства [1, 2]. В работе рассматривается функциональная настройка, когда не изменяются связи между элементами перестраиваемого устройства, но изменяется их функциональность. Предполагается, что часть устройства реализована на «жесткой» логике, а часть — на многофункциональных настраиваемых элементах. Например, к данному классу перестраиваемых устройств можно отнести автоматные шифраторы с ключевой информацией в виде подмножества переходов. Поведение таких устройств предлагается моделировать с помощью совместной работы двух базовых автоматов, один из которых имеет жестко фиксированное поведение, а поведение другого может изменяться (задаваться пользователем). Предложена логическая структура перестраиваемого устройства (рис.1), поведение которого является объединением поведения составляющих его базовых автоматов.

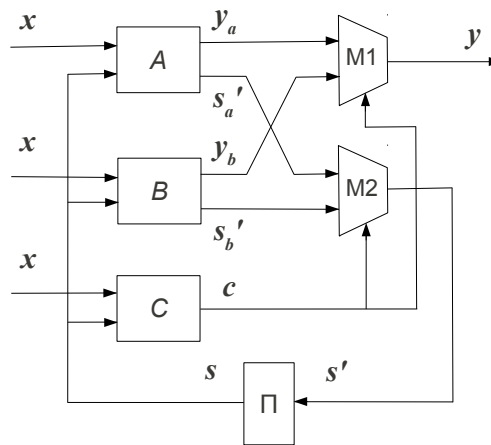


Рис. 1. Структура перестраиваемого устройства

Далее в работе «гибкие» цифровые устройства описываются моделью перестраиваемого автомата. Перестраиваемым автоматом Q называется шестерка $(S, X, Y, K, \psi, \varphi)$, где S, X, Y, K — конечные множества, называемые соответственно множеством состояний, входным алфавитом, выходным алфавитом, множеством настроек, а $\psi : S \times X \times K \rightarrow S$ и $\varphi : S \times X \times K \rightarrow Y$ — функции переходов и выходов соответственно. Таким образом, перестраиваемый автомат Q задает множество автоматов Мили $\{A_k = (S, X, Y, \psi_k, \varphi_k) : k \in K\}$, где $\psi_k(s, x) = \psi(s, x, k)$, $\varphi_k(s, x) = \varphi(s, x, k)$ для всех $s \in S$, $x \in X$ и $k \in K$.

¹Работа выполнена в рамках реализации ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг. (гос. контракт № П1010).