

Н. А. Б у р я к о в а, А. В. Ч е р н о в (Ростов-на-Дону, РГСУ). **Методология обнаружения заикливания фрагментов программ на основе верификации моделей.**

Метод верификации моделей (или проверки на модели) — это метод, при помощи которого желаемые свойства поведения реагирующей системы проверяются на заданной системе (модели) путем исчерпывающего перебора (явного или не явного) всех состояний, достижимых системой, и всех поведений, проходящих через эти состояния [1].

Существуют известные способы верификации моделей, рассматриваемые в работах Куршана [2], которая привела к разработке известной системы верификации COSPAN [3]–[5]; Варди и Волпера, которые предложили применять ω -автоматы (автоматы над бесконечными словами) для автоматической верификации [6].

В докладе рассматривается новый подход, основной целью которого является разработка методов и моделей диагностики программного обеспечения, которые позволяют выделять циклические конструкции из кода программы, классифицировать выделенные циклы и оценивать вероятность заикливания программ. В связи с поставленной задачей на первом этапе (выделение циклических структур) определяется область программного кода, содержащая операторы цикла. На втором этапе (классификация) оценивается возможность сокращения числа найденных циклических структур, исключением заведомо не проблемных, т. е. базирующихся на основе обычных циклов со счетчиком. На последнем этапе (оценка вероятности) строятся двоично-разрешающие диаграммы (BDD-diagram), формируются наборы входных данных и рассчитывается вероятность заикливания по BDD-диаграммам с учетом полученных на выходе результатов.

Классификация циклов осуществляет разделение всех найденных циклов на две группы: простые, т. е. циклы, не посещающие один и тот же узел дважды; сложные, повторяющие некоторые группы операторов неограниченное число раз.

Первую из представленных групп исключаем из дальнейшего рассмотрения.

Далее строим дерево булевой функции — дерево решений (BDD-diagram). Для заданного распределения входных логических значений (интерпретаций) переменных, путь в дереве, который отвечает этому распределению, ведет от корня дерева к вершине-листочку, обозначение которой есть значение функции. Сравнивая частоту появления среди выходных данных каждого из значений нашей функции, вычисляем вероятность безотказной незаикливаемой работы проверяемой программы.

СПИСОК ЛИТЕРАТУРЫ

1. *Кларк Э. М., Гамбург О., Пелед Д.* Верификация моделей программ: Model Checking. М.: МЦНМО, 2002, 416 с.
2. *Aggarwal S., Kurshan R. P., Sabnani K.* A calculus for protocol specification and validation. — In: H. Rudin and C. H. West, eds., Protocol Specification, Testing and Verification, North Holland, 1983, p. 19–34.
3. *Hardin R., Har'El Z., Kurshan R. P.* COSPAN. — In: Proceedings of the 1996 Workshop of Computer-Added Verification, Springer: LNCS, 1996, p. 423–427.
4. *Har'El Z., Kurshan R. P.* Software for analytical development of communications protocols. — ATUT Technical Journal, v. 69, № 1, p. 45–59.
5. *Kurshan R. P.* Computer-Aided Verification of Coordinating Processes: The Automata Theoretic Approach. Princeton University Press, 1994, p. 170–174.
6. *Vardi M. Y., Wolper P.* An automata-theoretic approach to automatic program verification. — In LICS86 Journal of the ACM, v. 20, № 1, p. 332–334.