# Параллельное построение пространства состояний конечной системы

Коротков Иван Андреевич,
учащийся магистратуры кафедры ИУ7 МГТУ им. Баумана;
руководитель В.А. Крищенко;
e-mail: korotkov2@mail.ru

При создании параллельных алгоритмов, протоколов взаимодействия возникает необходимость их верификации — проверки на соответствие определенным условиям целостности, например, отсутствие взимоблокировок.

На данный момент основным программным средством для такой проверки путем полного перебора пространства состояний является SPIN, использующий язык описания PROMELA. Существенным недостатком SPIN является невозможность использовать вычислительные ресурсы более чем одной машины, в первую очередь память. Количество состояний растет экспоненциально, что приводит к крайне длительной проверки моделей даже среднего размера, а модели большого размера требуют

В данной работе в качестве решения предлагается параллельный перебор множества состояний. На данный момент имеется лишь один экспериментальный аналог — DiVinE.

PROMELA позволяет осуществлять проверку как LTL-формул на пространстве состояний модели, так и более простых условий целостности (assert'ов) в определенных состояниях.

Во втором случае подходит перебор состояний в произвольном порядке, что позволяет использовать параллельный поиск в ширину. Для первого необходим поиск циклов, требующий поиска в глубину, нереализуемого в чистом виде параллельно. В [1,2] было предложение несколько различных алгоритмов, представляющих из себя совмещение поиска в ширину с последующим поиском циклов.

Возможны две модели распределения вычислительных работ между узлами — симметричная, когда каждый узел одновременно используется для хранения найденных состояний и поиска новых, и асимметричная, когда один или несколько узлов занимаются поиском, а остальные используются как распределенное хранилище. В асимметричной модели при генерации каждого нового состояния надо делать удаленный вызов для проверки, есть ли такое состояние в хранилище. В симметричной модели при правильном распределении состояний между узлам можно добиться значительного сокращения удаленных вызовов за счет локализации смежных состояний на одном узле. Одной из главных задач, таким образом, является нахождение удачного распределения состояний, которое бы обеспечило минимальное

количество удаленных вызовов при равномерном использовании памяти узлов; некоторые из таких распределений предложены в [3].

Дополнительной оптимизацией, позволяющей сократить обмен требуемой для хранения состояний памяти, является так называется бит-хэширование состояний [4], когда каждое состояние представляется в хэш-таблице единственным битом, без проверки коллизий.

В результате экспериментов выявлено, что подбором удачного распределения состояний между узлами можно сократить количество удаленных вызовов в 3-10 раз (в зависимости от задачи и объема пространства состояний) при ухудшении равномерности использования памяти до 10-20%. При этом время, уходящее на верификацию, примерно пропорционально количеству удаленных вызовов, которые являются узким местом алгоритма.

## Список литературы


1. Jiri Barnat, Lubos Brim. Parallel Breadth-First Search LTL Model-Checking.
2. Jiri Barnat, Lubos Brim. Distributed LTL Model-Checking in SPIN.
3. Flavio Lerda, Riccardo Sisto. Distributed-Memory Model Checking with SPIN.
4. Gerard Holzmann. An Analysis of Bitstate Hashing.