ВЕРИФИКАЦИЯ И ГЕНЕРАЦИЯ ГИБРИДНЫХ АВТОМАТОВ: ЗАДАЧИ И МЕТОДЫ

Овсянникова П. А., аспирант факультета ИТиП университета ИТМО,
polina.ovsyannikova@corp.ifmo.com

Аннотация

Гибридный автомат – математическая модель, описывающая поведение гибридных систем, то есть систем, обладающих одновременно дискретной и непрерывной динамикой. В данной работе приводится обзор существующих решений основных задач, связанных с данной предметной областью, а именно, моделирования, верификации и генерации гибридных автоматов.

Введение

Окружающая действительность наполнена динамическими гибридными системами, сочетающими в себе дискретную и непрерывную динамику. Для предсказания поведения таких систем и их верификации необходим инструмент, в полной мере отображающий специфику их функционирования. Одним из таких инструментов являются гибридные автоматы (ГА). В дискретных состояниях ГА (*locations*) с помощью обыкновенных дифференциальных уравнений (ОДУ) указываются законы изменения непрерывных переменных с течением времени (*flows*). При передаче управления между областями (*jumps*) значения переменных автомата обновляются в соответствии с условиями на ребрах. Формально ГА можно определить следующим образом.

Определение [1]: *N*-мерный ГА – это дискретно-непрерывная математическая модель, содержащая следующие компоненты:

- конечное множество вещественных переменных $X = \{x_1, .., x_n\}$;
- конечный ориентированный мультиграф (V, E), где V – множество вершин или областей, E – множество ребер, с помощью которых происходит передача управления;
- три функции пометок вершин: $init(v)$, $inv(v)$, $flow(v)$;
- функция пометок ребер (функция переходов) $jump(e)$;
- конечное множество событий (условий переходов), каждое ребро должно быть отмечено одним из них.

Пример: ГА, отображающий поведение системы «прыгающий мячик» (Рис. 1). Он содержит одну область l_0, множество переменных $X = \{x, v\}$. Функции пометок выражаются следующими предикатами:

- $flow(l_0) := \dot{x} = v \wedge \dot{v} = -g$;
- $init(l_0) := x = x_0 \wedge v = 0$;
- $inv(l_0) := x \geq 0$.

Функция пометок ребра выражается через предикат $jump(e) := x = 0 \wedge v \leq 0 \rightarrow v = -cv$. Таким образом, с помощью данного ГА моделируется изменение скорости v и координаты x мячика, сброшенного с высоты x_0 с нулевой начальной скоростью вертикально вниз. Во время удара о поверхность направление скорости меняется на противоположное, и численно она становится равной cv, где c – коэффициент упругости.

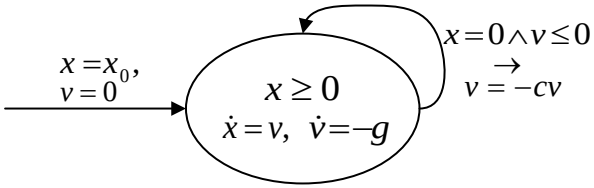

Рисунок 1: Гибридный автомат для системы «прыгающий мячик»

Применение

ГА находят применение в самых разных областях знаний. Так, например, в [2] описано моделирование человеческих внутренних органов для последующей верификации имплантируемых медицинских устройств; в промышленной автоматике с их помощью, например, может быть представлен один из трендов в области проектирования энергетических систем – умная сеть электроснабжения (smart grid) [3]. Далее, ГА может отражать поведение химической установки [4] и Т-лимфоцита [5].

Верификация

Несмотря на то, что ГА удобны для моделирования многих динамических систем во многих областях знаний, они сложны для верификации, так как обладают бесконечным пространством состояний. В связи с этим зачастую для анализа достижимости применяются способы,

связанные с аппроксимацией достижимых состояний сверху [6]. Также существуют более узкие классы ГА, где задача анализа достижимости имеет решение, например, ГА со значениями производных в функциях пометок, выражающихся с помощью интервалов (так называемые прямоугольные гибридные автоматы) [7]. Пример прямоугольного автомата представлен на Рис. 2. Рассмотрим несколько методов верификации.

Подход к верификации прямоугольных автоматов был предложен в [7]. Он состоит из трех шагов. На первом прямоугольный автомат преобразуется в сингулярный. Сингулярный автомат отличается от прямоугольного тем, что значения производных в функциях пометок не принадлежат какому-либо интервалу, а являются конкретными действительными числами. На втором этапе сингулярный автомат, полученный на первом шаге, преобразуется во временной автомат, после чего, на третьем шаге решается задача анализа достижимости для временного автомата.

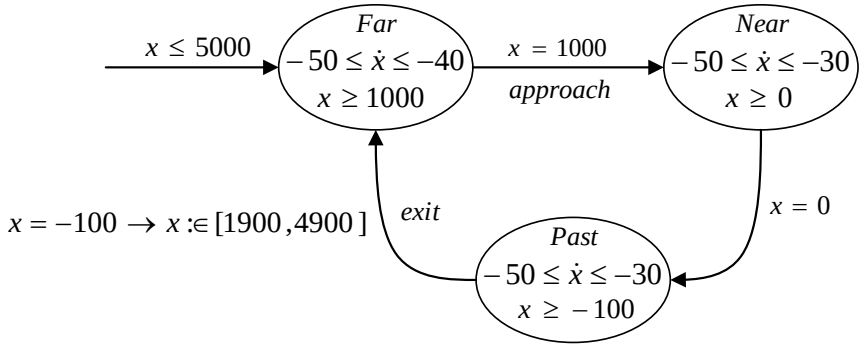

Рисунок 2: Пример прямоугольного автомата [1]

Второй подход, представленный в [6], предполагает итеративное уточнение абстракций и вычисление достижимых состояний при помощи аппроксимации сверху. Рассмотрена замкнутая система, состоящая из программируемого логического контроллера (ПЛК) и управляемого устройства. Производится верификация контроллера, соединенного с неточной моделью устройства. Если результат верификации отрицательный, выдается контрпример, то есть последовательность состояний, приводящая в указанное в свойстве множество нежелательных состояний (опасный регион). После чего ГА управляемого устройства уточняется при помощи добавления ОДУ вдоль последовательности состояний контрпримера. Таким образом, если опасный регион достижим

на максимально конкретизированном ГА, результат верификации отрицательный, иначе – положительный.

Многие подходы были оформлены в виде программных инструментов. На данный момент можно выделить следующие:

- SpaceEx [8]. Сочетает в себе подходы, использующие многогранники и представление непрерывных множеств с помощью их поддерживающих функций для аппроксимации сверху области достижимых состояний.
- Flow* [9]. Для аппроксимации достижимых состояний сверху здесь использована техника интегрирования тейлоровских моделей. Возможна проверка моделей, содержащих линейные, полиномиальные и неполиномиальные ОДУ.
- HyComp [10]. Инструмент для проверки моделей гибридных систем, выполненный как надстройка над верификатором NuXmv, основан на методах решения задачи выполнимости формул в теориях (SMT). ГА в данном случае задается с помощью символьного языка HyDI. Одним из его преимуществ является возможность обработки спецификаций, заданных в форме линейной темпоральной логики.

Синтез

Задача верификации трудная, но к настоящему времени было разработано некоторое число программных средств, позволяющих выполнять проверку различных ГА на соответствие различным свойствам. Однако, для решения описанной задачи необходим ГА, корректно отображающий работу системы. Но что делать, если представленная для моделирования система слишком сложна, чтобы создать ГА вручную, с учетом взаимодействия всех ее переменных? Такая ситуация характерна, например, для киберфизических систем (КФС). В КФС входные сигналы физического компонента (объекта управления) формируются программным компонентом (контроллером) на основе выходных сигналов первого. В этом случае сначала нужно решить задачу синтеза ГА по примерам поведения КФС. На текущий момент существует не так много способов автоматической генерации ГА. Рассмотрим некоторые из них.

В подходе [11] синтез ГА осуществляется по трассировкам. Во входных данных также необходимо предоставить шаблоны *flow* предикатов. На первом этапе алгоритма происходит разбиение примеров поведения на области и синтез *flow* предикатов на основе шаблонов. Далее, используя уравнение нормализованной точечной взаимной информации, выводятся условия переходов.

Авторы работы [12] предлагают алгоритм генерации линейных гибридных автоматов по трассировкам. Для определения охранных условий на переходах между областями применяется неравенство Крамера Рао, также оно используется для вывода предиката *flow*. Построенный автомат верифицируется при помощи тестового набора трассировок и уточняется в случае необходимости.

Метод [13] предполагает конструирование ГА для встроенных систем по предварительно сегментированным примерам поведения. Сначала трассировки кластеризуются, далее по ним с помощью алгоритма, близкого к L* [14], генерируется автомат Мили. Затем предлагается эвристика для вывода *init* и *flow* предикатов.

Несмотря на то, что работа в этом направлении ведется, на данный момент, насколько известно автору, в свободном доступе все еще не представлено ни одного программного средства для синтеза ГА.

Заключение

ГА – мощный инструмент моделирования работы дискретно-непрерывных динамических систем, применяемый в разнообразных областях знаний. ГА могут быть использованы для визуализации работы процессов и систем, предсказания их будущих состояний, верификации. Существует множество исследований, решающих рассмотренные задачи, и все они требуют наличия ГА, отображающего работу моделируемой системы. Однако в случае сложных систем, построение такого ГА вручную может оказаться непростой задачей. Кроме того, в процессе разработки системы необходимо поддерживать в актуальном состоянии и ее формальную модель (ГА), что зачастую сложно и трудозатратно. Следовательно, задача синтеза гибридных автоматов актуальна, однако, будучи малоизученной, открывает автору простор для исследований.

Литература

1. Henzinger T. The theory of hybrid automata // Verification of Digital and Hybrid Systems. — Springer, Berlin, Heidelberg, 2000. — P. 265–292.
2. Wang L. et al. A Formal Approach for Scalable Simulation of Gastric ICC Electrophysiology // IEEE Transactions on Biomedical Engineering. — 2019.
3. Martins J., Platzer A., Leite J. Statistical model checking for distributed probabilistic-control hybrid automata with smart grid applications //

International Conference on Formal Engineering Methods. — Springer, Berlin, Heidelberg, 2011. — P. 131–146.

4. Engell S. et al. Continuous-discrete interactions in chemical processing plants // Proceedings of the IEEE. — 2000. — Vol. 88., no. 7. — P. 1050–1068.
5. Milutinovic D. et al. A hybrid automata model of TCR triggering dynamics // Proceedings of the 11th Mediterranean Conference on Control and Automation MED. — 2003. — P. 18–20.
6. Nellen J., Ábrahám E., Wolters B. A CEGAR tool for the reachability analysis of PLC-controlled plants using hybrid automata // Workshop on Formal Methods Integration. — Springer, Cham, 2014. — P. 55–78.
7. Henzinger T. A. et al. What's decidable about hybrid automata? // Journal of computer and system sciences. — 1998. — Vol. 57., no. 1. — P. 94–124.
8. Frehse G. et al. SpaceEx: Scalable verification of hybrid systems // International Conference on Computer Aided Verification. — Springer, Berlin, Heidelberg, 2011. — P. 379–395.
9. Chen X., Ábrahám E., Sankaranarayanan S. Flow*: An analyzer for non-linear hybrid systems // International Conference on Computer Aided Verification. — Springer, Berlin, Heidelberg, 2013. — P. 258–263.
10. Cimatti A. et al. HyComp: An SMT-based model checker for hybrid systems // International Conference on Tools and Algorithms for the Construction and Analysis of Systems. — Springer, Berlin, Heidelberg, 2015. — P. 52–67.
11. Summerville A., Osborn J., Mateas M. Charda: Causal hybrid automata recovery via dynamic analysis // arXiv preprint arXiv:1707.03336. — 2017.
12. Lamrani I., Banerjee A., Gupta S. K. S. HyMn: mining linear hybrid automata from input output traces of cyber-physical systems // 2018 IEEE Industrial Cyber-Physical Systems (ICPS). — IEEE, 2018. — P. 264–269.
13. Medhat R. et al. A framework for mining hybrid automata from input/output traces // Proceedings of the 12th International Conference on Embedded Software. — IEEE Press, 2015. — P. 177–186.
14. Angluin D. Learning regular sets from queries and counterexamples // Information and computation. — 1987. — Vol. 75., no. 2. — P. 87–106.