DOI 10.58351/2949-2041.2026.40.11.009

Дейнеко Максим Дмитриевич,
Магистрант, МГТУ им. Н. Э. Баумана
Deyneko Maxim Dmitrievich, Master's student
Bauman Moscow State Technical University

ИНТЕГРАЦИЯ ТЕМПОРАЛЬНОГО ВЕРИФИКАТОРА В КОНТУР АДАПТИВНОГО УПРАВЛЕНИЯ ДОРОЖНЫМ ДВИЖЕНИЕМ
INTEGRATION OF A TEMPORAL VERIFIER INTO AN ADAPTIVE ROAD TRAFFIC CONTROL LOOP

Аннотация. Предложена схема включения темпорального верификатора в контур адаптивного управления дорожным движением. Внешний модуль формирует кандидатное действие, а верификатор проверяет состояние, границы длительностей, совместимость фаз и временные инварианты. Исполнительный модуль получает только разрешенные действия; при отказе сохраняется безопасное состояние и формируется диагностическое сообщение.

Abstract. A method for integrating a temporal verifier into an adaptive road traffic control loop is proposed. An external module produces a candidate action, whereas the verifier checks the state, phase-duration bounds, transition compatibility, and temporal invariants. Only permitted actions are passed to the execution module; otherwise, a safe state and a diagnostic message are retained.

Ключевые слова: Адаптивное управление, темпоральная верификация, дорожное движение, управляющее действие, временные инварианты, диагностика.

Keywords: Adaptive control, temporal verification, road traffic, control action, temporal invariants, diagnostics.

Введение

Адаптивное управление получает данные о дорожной сети и формирует воздействие. Кандидатное действие может предложить аналитический алгоритм, обучаемая модель или операторский модуль, однако численный ответ не гарантирует исполнимости: он способен нарушить длительность фазы, порядок переходов или баланс цикла. Поэтому генерацию решения и его допуск следует разделять.

Темпоральный верификатор рассматривается как независимый шлюз между модулем принятия решения и исполнителем. Он не оценивает оптимальность алгоритма, а устанавливает допустимость конкретного действия в текущем состоянии и на ближайшем временном горизонте. Формальное описание свойств последовательностей состояний опирается на темпоральную логику [1]; распределение длительностей фаз может формироваться из локальных очередей [2].

Состояние контура и формат управляющего сообщения

Для изолированного двухфазного дорожного узла в момент t используется состояние, содержащее измеренные потоки, длины очередей, активную фазу, время ее выполнения и параметры цикла. Наблюдение передается в верификатор вместе с идентификатором узла и отметкой времени.

$$x\ (t)\ =\ [q\ (t), l\ (t), p\ (t), \tau\ (t), C, L, gmin, m\ (t)] \quad (1)$$

где x (t) – состояние узла в момент t; q (t) – интенсивности потоков по направлениям, авт./ч; l (t) – длины очередей, авт.; p (t) – активная фаза; τ (t) – время, прошедшее с ее начала, с; C – длительность цикла, с; L – суммарное потерянное время, с; gmin – минимальная допустимая длительность фазы, с; m (t) – метка полноты и давности измерений.

Кандидатное действие имеет вид пары целевых длительностей и признака переключения: a (t) = {gNS, gEW, switch}. Здесь gNS и gEW задают зеленое время для двух взаимно конфликтующих групп направлений, a switch определяет, нужно ли запускать переход к следующей группе. Полезное зеленое время вычисляется до формирования действия.

$$G\ =\ C\ -\ L \quad (2)$$

где G – полезное зеленое время, с; C – длительность цикла, с; L – потерянное время, с. Если G < 2gmin, минимальное обслуживание двух групп невозможно, поэтому кандидат отклоняется.

Состав обмена приведен в таблице 1; исполнительный модуль получает только одобренную команду.

Таблица 1

Сообщения в контуре темпоральной верификации

Сообщение	**Передаваемые данные**	**Назначение**
Наблюдение	x (t), идентификатор узла, отметка времени	Фиксирует исходное состояние для проверки
Кандидат	gNS, gEW, switch, источник решения	Передает действие, которое еще не является командой
Вердикт и диагностика	допуск/отказ, нарушенные условия, фактические значения	Отделяет исполнимое действие и объясняет отказ
Команда	одобренное действие, остаток текущей фазы	Передается исполнительному компоненту

Логика допуска управляющего действия

Верификатор вычисляет четыре группы условий: полноту и диапазоны входных данных; баланс gNS + gEW = G и g ≥ gmin; соответствие действия текущей фазе; временные инварианты перехода. Конфликтующие направления не получают разрешение одновременно, а переходные состояния не пропускаются.

$$V\ (x, a)\ =\ vdata\ \wedge\ vcycle\ \wedge\ vphase\ \wedge\ vtime \qquad (3)$$

где V (x,a) – итоговый вердикт для состояния x и действия a; vdata – достоверность и допустимость измерений; vcycle – выполнение ограничений цикла и минимального обслуживания; vphase – соответствие действия текущей фазе; vtime – выполнение временных инвариантов. Символ ∧ означает логическое «и»: команда допускается только при истинности всех четырех составляющих.

При V (x,a) = ложь кандидат не передается исполнителю. Безопасное действие – продолжение текущей фазы до ближайшего допустимого момента; ответ содержит нарушенное условие, фактические значения τ (t), gmin и остаток времени. Диагностика служит оператору и меткой для обучаемого модуля.

Преобразование одобренного решения в исполнимую команду

После допуска команда переводится в операции исполнительного интерфейса. При продлении активной фазы вычисляется ее оставшаяся длительность, а при смене группы направлений запускается заранее заданная последовательность переходных состояний. Такой порядок исключает прямое присваивание конфликтующих состояний и сохраняет желтые и всекрасные интервалы, определенные в модели дорожного узла [4].

$$drem\ =\ gtarget\ -\ \tau\ (t) \qquad (4)$$

где drem – оставшаяся длительность активной фазы, с; gtarget – одобренная целевая длительность этой фазы, с; τ (t) – время, уже прошедшее с начала фазы, с. Если drem ≤ 0, продление не выполняется: модуль формирует событие перехода и повторно проверяет его допустимость.

В экспериментальном контуре исполнительный интерфейс может подключаться к микроскопической модели. Она задает программы фаз, возвращает состояния направлений и изменяет длительность текущей фазы из внешнего процесса [3, 4]; ее трассы и метрики независимы от решения верификатора.

Пример обработки кандидата

Для узла с потоками север–юг 1210 авт./ч и восток–запад 790 авт./ч примем C = 90 с, L = 8 с, gmin = 15 с и τ (t) = 15 с. Кандидат: gNS = 50 с, gEW = 32 с. G = 82 с; сумма длительностей и минимальный таймер соблюдены, поэтому допускается продление фазы.

Результат проверки приведен в таблице 2. Пример иллюстрирует логику обмена сообщениями и не является оценкой транспортного эффекта на реальной сети.

Таблица 2

Обработка кандидатного действия

Проверка	**Значения**	**Результат**
Полезное время	G = 90 − 8 = 82 с	Выполнено
Минимальные длительности	gNS = 50 с; gEW = 32 с; gmin = 15 с	Выполнено
Баланс цикла	50 + 32 = 82 с	Выполнено
Временное предусловие	τ (t) = 15 с; gmin = 15 с	Переключение разрешено
Остаток фазы	drem = 50 − 15 = 35 с	Передать команду продления

Журналирование и применение вербальной обратной связи

Журнал хранит идентификатор сценария, время, хеш наблюдения, кандидатное действие, vdata, vcycle, vphase, vtime, вердикт и команду. Хеш связывает объяснение с измерениями. Одних признаков «разрешено» или «отказано» недостаточно: без порогов и фактических значений результат невоспроизводим.

Вербальная обратная связь строится по шаблону из полей журнала. При отказе по циклу указываются C, L, gNS, gEW и отклонение от G; при отказе по времени – p (t), τ (t) и gmin. Текст служит объяснением и меткой ошибки, не влияя на решение верификатора.

Заключение

Темпоральный верификатор отделяет генерацию кандидатного действия от исполнения и проверяет ограничения цикла, фазы и временные инварианты. При отказе сохраняется безопасное состояние, а журнал формирует воспроизводимую диагностику и вербальную обратную связь

Список литературы:

1. Pnueli A. The Temporal Logic of Programs // Proceedings of the 18th Annual Symposium on Foundations of Computer Science. 1977. P. 46–57. DOI: 10.1109/SFCS.1977.32.

2. Varaiya P. Max Pressure Control of a Network of Signalized Intersections // Transportation Research Part C: Emerging Technologies. 2013. Vol. 36. P. 177–195. DOI: 10.1016/j.trc.2013.08.014.

3. Krajzewicz D., Erdmann J., Behrisch M., Bieker L. Recent Development and Applications of SUMO – Simulation of Urban MObility // International Journal On Advances in Systems and Measurements. 2012. Vol. 5. No. 3–4. P. 128–138.

4. Traffic Lights – SUMO Documentation [Электронный ресурс]. URL: https://eclipse.dev/sumo/docs/Simulation/Traffic_Lights.html (дата обращения: 17.11.2026)