УДК 519.767.2
DOI: 10.15827/0236-235X.128.547-555
Дата подачи статьи: 12.08.19
2019. Т. 32. № 4. С. 547–555

# *Верификация моделей систем на базе эквациональной характеристики формул CTL*

*Ю.П. Кораблин* [1,2]*, д.т.н., профессор, y.p.k@mail.ru*
*А.А. Шипов* [3]*, к.т.н., старший инженер-программист, a-j-a-1@yandex.ru*

[1] *Российский технологический университет – МИРЭА, г. Москва, 119454, Россия*
[2] *Национальный исследовательский университет «МЭИ», г. Москва, 111250, Россия*
[3] *РСК Технологии, г. Москва, 121170, Россия*

В статье предложена и рассмотрена RTL-нотация, основанная на системах рекурсивных уравнений и привычных семантических определениях логики линейного времени LTL и логики ветвящегося времени CTL. В предыдущих работах авторов, когда данная нотация еще называлась RLTL-нотацией, было показано, что с ее помощью можно легко формулировать и верифицировать свойства логики линейного времени, в том числе и относительно моделей систем, заданных с помощью той же нотации. Затем были расширены возможности RLTL-нотации, благодаря чему с ее помощью стало возможным формулировать выражения не только логики LTL, но и логики ветвящегося времени. В результате этого появилась первая версия RTL-нотации.

В данной статье представлена вторая версия RTL как результат доработки и упрощения семантических определений нотации, позволивших повысить наглядность и читаемость ее выражений.

Целью статьи является демонстрация возможности использования RTL-нотации в качестве инструмента для формулировки и верификации свойств, задаваемых формулами обеих логик, на базе единых аксиом и правил. Это дает возможность RTL выступать в роли единой универсальной нотации данных логик. При этом за счет незначительных дополнений ее базовых определений нотация способна включать в себя выразительные особенности и других временных логик, что в перспективе позволит RTL стать полноценной универсальной временной логикой, обладающей всеми необходимыми инструментами и средствами для реализации всех этапов верификации.

***Ключевые слова:*** *верификация, Model Checking, эквациональная характеристика RTL, формулы временной логики, LTL, CTL, системы рекурсивных уравнений.*

В данной статье авторы продолжают развивать теоретическую и инструментальную базу RTL-нотации, преследуя цель создания самостоятельной, независимой (от сторонних алгоритмов и моделей данных) и единой системы средств и методов верификации. Концептуально RTL выстроена таким образом, что выразительные свойства любой существующей или возникающей временной логики либо уже, либо без особого труда могут быть имплементированы в качестве некоторой ее части или разновидности. Такая способность RTL к инкапсуляции свойств временных логик стала возможной благодаря лежащей в ее основе теории систем рекурсивных уравнений и оператору продолжения. Однако, несмотря на рекурсивную природу, RTL является весьма интуитивно понятной и читаемой нотацией.

Ранее в работе [1] авторами были подробно рассмотрены инструменты и средства унифицированного представления формул логик LTL и CTL в виде систем рекурсивных уравнений. Было показано, что RTL-нотация является достаточно выразительной, чтобы с ее помощью задавать и модели систем, и верифицирующие их свойства (на базе логики как линейного времени, так и ветвящегося). Данная статья логически развивает эту тему и посвящена решению естественно возникшей проблемы, касающейся процесса верификации моделей систем, или Model Checking [2, 3], относительно свойств логики ветвящегося времени, заданных на базе RTL.

За счет небольших (по сравнению с [1]) изменений в синтаксисе, существенно упростивших восприятие RTL-формул, а также введения ряда аксиом авторам удалось сформировать весьма понятный и наглядный алгоритм верификации, который основан на таком стандартном и всем известном методе, как метод построения синхронной композиции. Разработанный алгоритм подробно описывается в статье, а его работа (для самопроверки) наглядно демонстрируется на нескольких доступных и известных примерах, взятых из работы [2]. Так, забегая вперед, можно сказать, что, помимо

теоретической обоснованности алгоритма, на практике удалось проверить и подтвердить его работоспособность, получив аналогичные результаты верификации. В то же время нужно отметить, что в данной статье авторами не исследовалась возможность верификации больших распределенных программных систем. В основном речь идет о создании инструмента унифицированного представления формул логик LTL и CTL и использовании его для верификации моделей систем.

## RTL-нотация

В основе RTL-нотации лежат семантические определения временных логик LTL и CTL [4–7] и системы рекурсивных уравнений [8]. Для упрощения задания в RTL семантических значений кванторов CTL расширим семантическое определение оператора продолжения по сравнению с тем, как это сделано в [9]. Оператор продолжения «∘», как и ранее, выступает в качестве связующего звена (перехода) между текущим состоянием и следующим. Для задания моделей систем это будет его единственным назначением. Для верифицируемого CTL-свойства оператор продолжения необходимо наделить возможностью характеризовать как квантор существования E, так и квантор всеобщности A. В связи с этим примем следующие обозначения:

«∘» – применяется как для моделей систем, так и для верифицируемых свойств (для моделей систем задает и описывает переходы состояний, а для верифицируемых свойств гарантирует проверку выполнимости хотя бы для одной из ветвей);

«•» – применяется только для верифицируемых свойств и гарантирует проверку выполнимости по всем ветвям.

Данное расширение оператора продолжения является по сути единственным необходимым изменением (по сравнению с [1]), которое дает нотации возможность формулировать выражения логики CTL. В таблице 1 приведены рекурсивные представления базовых операторов CTL в RTL-нотации с учетом использования операторов продолжения, соответствующих различным кванторам. Если формулы LTL задают лишь некоторый порядок наступления каких-либо событий в целом, то формулы CTL эти события соотносят с альтернативными путями развития вычислительного процесса. Это требуется, когда верифицируемая система переходит в различные режимы работы (в зависимости от исходных данных или условий), работает в многопоточном режиме [10] или результаты ее работы носят вероятностный характер [11].

*Таблица 1*

**Рекурсивные представления операторов CTL в RTL-нотации**

*Table 1*

**Recursive representations of CTL statements in RTL notation**

| CTL | RTL |
|---|---|
| $AX\varphi$ | $\Delta \bullet \varphi$ |
| $EX\varphi$ | $\Delta \circ \varphi$ |
| $AF\varphi = \varphi \vee AX\ AF\varphi$ | $F = \varphi + \Delta \bullet F$ |
| $EF\varphi = \varphi \vee EX\ EF\varphi$ | $F = \varphi + \Delta \circ F$ |
| $AG\varphi = \varphi \wedge AX\ AG\varphi$ | $F = \varphi \bullet F$ |
| $EG\varphi = \varphi \wedge EX\ EG\varphi$ | $F = \varphi \circ F$ |
| $A(\varphi_1 U \varphi_2) = \varphi_2 \vee \varphi_1 \wedge AX\ A(\varphi_1 U \varphi_2)$ | $F = \varphi_2 + \varphi_1 \bullet F$ |
| $E(\varphi_1 U \varphi_2) = \varphi_2 \vee \varphi_1 \wedge EX\ E(\varphi_1 U \varphi_2)$ | $F = \varphi_2 + \varphi_1 \circ F$ |

**Аксиомы RTL:**

A1. $\neg F\gamma = G\neg\gamma$,
A2. $\neg G\gamma = F\neg\gamma$,
A3. $\neg A\gamma = E\neg\gamma$,
A4. $\neg E\gamma = A\neg\gamma$,
A5. $\neg\Delta = \Delta$,
A6. $\Delta^\omega \circ \gamma = \Delta^\omega$,
A7. $\Delta^\omega = \Delta \circ \Delta^\omega$,
A8. $\gamma_1 + \gamma_2 = \gamma_2 + \gamma_1$,
A9. $\gamma + \gamma = \gamma$,
A10. $(\gamma_1 + \gamma_2) + \gamma_3 = \gamma_1 + (\gamma_2 + \gamma_3)$,
A11. $\gamma_1 \circ (\gamma_2 \circ \gamma_3) = (\gamma_1 \circ \gamma_2) \circ \gamma_3$,
A12. $(\gamma_1 + \gamma_2) \circ \gamma_3 = \gamma_1 \circ \gamma_3 + \gamma_2 \circ \gamma_3$,
A13. $\gamma_1 \circ (\gamma_2 + \gamma_3) = \gamma_1 \circ \gamma_2 + \gamma_1 \circ \gamma_3$,
A14. $\gamma_1 \bullet (\gamma_2 + \gamma_3) = \gamma_1 \bullet \gamma_2 + \gamma_1 \bullet \gamma_3$.

Сформулируем основные аксиомы, необходимые для построения синхронной композиции двух систем рекурсивных уравнений:

S1. $\varphi_1 \otimes \varphi_2 = \varphi_1$, если $\varphi_1 \subseteq \varphi_2$,
$\varphi_2$, если $\varphi_2 \subseteq \varphi_1$,
$\perp$ в противном случае.
S2. $\varphi \otimes \perp = \perp$,
S3. $\mu \circ \perp = \perp \circ \mu = \perp$,
S4. $\varphi \otimes \Delta = \varphi$,
S5. $(\varphi_1 \circ \mu_1) \otimes (\varphi_2 \circ \mu_2) = (\varphi_1 \otimes \varphi_2) \circ (\mu_1 \otimes \mu_2)$,
S6. $(\varphi_1 \circ \mu_1 + \varphi_2 \circ \mu_2) \otimes (\varphi_3 \bullet \mu_3) =$
$\{(\varphi_1 \otimes \varphi_3) \circ (\mu_1 \otimes \mu_3), (\varphi_2 \otimes \varphi_3) \circ (\mu_2 \otimes \mu_3)\}$,
S7. $(\varphi_1 \circ \mu_1 + \varphi_2 \circ \mu_2) \otimes (\varphi_3 \circ \mu_3) =$
$(\varphi_1 \otimes \varphi_3) \circ (\mu_1 \otimes \mu_3) + (\varphi_2 \otimes \varphi_3) \circ (\mu_2 \otimes \mu_3)$,
S8. $\{\varphi_1 \circ \mu_1, \varphi_1 \circ \mu_2\} = \varphi_1 \circ \{\mu_1, \mu_2\}$,
S9. $\{\mu_1, \mu_2\} = \{\mu_2, \mu_1\}$,
S10. $\{\mu, \perp\} = \perp$,

S11. $(\varphi_1 \circ \mu_1) \otimes (\varphi_2 \circ (\mu_2 + \mu_3)) = (\varphi_1 \otimes \varphi_2) \circ (\mu_1 \otimes \mu_2 + \mu_1 \otimes \mu_3)$,

S12. $(\varphi_1 \circ \mu_1) \otimes (\varphi_2 \bullet (\mu_2 + \mu_3)) = (\varphi_1 \otimes \varphi_2) \circ (\mu_1 \otimes \mu_2 + \mu_1 \otimes \mu_3)$,

S13. $\{\mu_1, \mu_2 + \mu 3\} = \{\mu_1, \mu_2\} + \{\mu_1, \mu_3\}$,

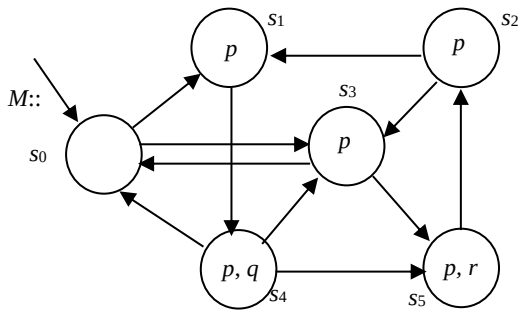
S14. $\Delta \circ \perp = \perp$.

В записи вида $\{\mu_1, \mu_2\}$ фигурные скобки означают связанное продолжение, что эквивалентно выполнению формулы $\mu_1 \,\&\, \mu_2$, то есть связанное продолжение выполняется тогда и только тогда, когда выполняется каждое продолжение, входящее в его состав.

## Верификация CTL-свойств с помощью RTL

Поскольку верификация LTL-свойств была подробно рассмотрена в работе [9], данная статья посвящена лишь верификации CTL-свойств, заданных с помощью RTL-нотации. Применение предложенного метода наглядно и всесторонне демонстрируется на типовых примерах задания верифицируемых свойств для систем небольших размеров, что является стандартным и общепринятым способом иллюстрации возможностей метода.

С помощью вышеописанных аксиом на базе RTL рассмотрим пример, представленный в [1] (см. рисунок).




*Верифицируемая модель M на базе структуры Крипке*

*Verifiable model M based on the Kripke structure*

В RTL-нотации данная модель M примет следующий вид:

$S_0 = \{\} \circ (S_1 + S_3)$,

$S_1 = p \circ S_4$,

$S_2 = p \circ (S_1 + S_3)$,

$S_3 = p \circ (S_0 + S_5)$,

$S_4 = \{p, q\} \circ (S_0 + S_3 + S_5)$,

$S_5 = \{p, r\} \circ S_2$.

В качестве верифицируемого свойства используем CTL-свойство $\Phi = E((EX]p) \,U\, AF(q \vee r))$.

Выполним преобразование свойства Φ к RTL-виду:

$\Phi_0 = E(\Phi_1 \,U\, \Phi_2) = \Phi_2 + \Phi_1 \circ \Phi_0$,

$\Phi_1 = EX]p = \Delta \circ ]p$,

$\Phi_2 = AF(q \vee r) = (q + r) + \Delta \bullet \Phi_2$

$\Leftrightarrow$

$\Phi_0 = (q + r) + \Delta \bullet \Phi_2 + \Delta \circ ]p \circ \Phi_0$

$\Leftrightarrow$

$F_0 = (q + r) + \Delta \bullet F_1 + \Delta \circ F_2$,

$F_1 = (q + r) + \Delta \bullet F_1$,

$F_2 = ]p \circ F_0$.

Введем еще одно состояние, чтобы модель верифицирующего свойства описывала реагирующую систему, которое и станет допускающим состоянием (выделено жирным шрифтом):

$F_0 = (q + r) \bullet \mathbf{F_3} + \Delta \bullet F_1 + \Delta \circ F_2$,

$F_1 = (q + r) \bullet \mathbf{F_3} + \Delta \bullet F_1$,

$F_2 = ]p \circ F_0$,

$\mathbf{F_3} = \Delta \circ \mathbf{F_3}$.

Переход в состояние $\mathbf{F_3}$ задан оператором «•», то есть должен выполняться для всех путей, поскольку данное состояние наступает только после выполнения AF(q ∨ r). Иными словами, только если все переходы состояния модели помечены q или r, можно выполнить переход в $\mathbf{F_3}$.

Правило выполнимости формулы Φ на модели M: Φ выполняется на M тогда и только тогда, когда из начального состояния синхронной композиции $M \otimes \Phi$ существует хотя бы одна нетерминальная ветвь, ведущая в одно из ее допускающих состояний.

Выполним построение синхронной композиции $M \otimes \Phi$.

В дальнейшем для краткости будем обозначать синхронную композицию $(S_i \otimes F_j)$ через $C_{i,j}$. Допускающими состояниями композиции являются состояния, которые содержат допускающие состояния верифицирующего свойства (выделено жирным шрифтом). Поскольку попадание в состояние $C_{i,3}$ $(i = 0 \div 5)$ порождает лишь переходы в состояния $C_{j,3}$ $(j = 0 \div 5)$, все соответствующие уравнения могут быть сразу же заменены на одно $\mathbf{C_e} = \Delta \circ \mathbf{C_e}$ и при появлении в правой части уравнений состояний $C_{i,3}$ их можно сразу заменять на $\mathbf{C_e}$. Получим следующую цепочку равенств:

$C_{0,0} = S_0 \otimes F_0 =$

$[\{\} \circ (S_1 + S_3)] \otimes [(q + r) \bullet \mathbf{F_3} + \Delta \bullet F_1 + \Delta \circ F_2] =$

$\{\} \circ (S_1 + S_3)] \otimes [(q + r) \bullet \mathbf{F_3}] + [\{\} \circ (S_1 + S_3)] \otimes [\Delta \bullet F_1] + [\{\} \circ (S_1 + S_3)] \otimes [\Delta \circ F_2] =$

$[\perp \circ \{S_1 \otimes \mathbf{F_3}, S_3 \otimes \mathbf{F_3}\}] + [\{\} \otimes \Delta \circ \{S_1 \otimes F_1, S_3 \otimes F_1\}] + [\{\} \otimes \Delta \circ (S_1 \otimes F_2 + S_3 \otimes F_2)] =$

$\{\} \circ \{C_{1,1}, C_{3,1}\} + \{\} \circ (C_{1,2} + C_{3,2})$,

$C_{1,1} = S_1 \otimes F_1 =$
$[p \circ S_4] \otimes [(q + r) \bullet \mathbf{F_3} + \triangle \bullet F_1] =$
$[p \circ S_4] \otimes [(q + r) \bullet \mathbf{F_3}] + [p \circ S_4] \otimes [\triangle \bullet F_1] =$
$[\perp \circ S_4 \otimes \mathbf{F_3}] + [p \otimes \triangle \circ S_4 \otimes F_1] = p \circ \{C_{4,1}\},$
$C_{3,1} = S_3 \otimes F_1 =$
$[p \circ (S_0 + S_5)] \otimes [(q + r) \bullet \mathbf{F_3} + \triangle \bullet F_1] =$
$[p \circ (S_0 + S_5)] \otimes [(q + r) \bullet \mathbf{F_3}] + [p \circ (S_0 + S_5)] \otimes [\triangle \bullet F_1] = [\perp \circ \{S_0 \otimes \mathbf{F_3}, S_5 \otimes \mathbf{F_3}\}] +$
$[p \circ \{S_0 \otimes F_1, S_5 \otimes F_1\}] = p \circ \{C_{0,1}, C_{5,1}\},$
$C_{1,2} = S_1 \otimes F_2 = [p \circ S_4] \otimes []p \circ F_1] = \perp,$
$C_{3,2} = S_3 \otimes F_2 = [p \circ (S_0 + S_5)] \otimes []p \circ F_1] = \perp,$
$C_{4,1} = S_4 \otimes F_1 =$
$[\{p, q\} \circ (S_0 + S_3 + S_5)] \otimes [(q + r) \bullet \mathbf{F_3} + \triangle \bullet F_1] =$
$[\{p, q\} \circ (S_0 + S_3 + S_5)] \otimes [(q + r) \bullet \mathbf{F_3}] +$
$[\{p, q\} \circ (S_0 + S_3 + S_5)] \otimes [\triangle \bullet F_1] =$
$[q \circ \{S_0 \otimes \mathbf{F_3}, S_3 \otimes \mathbf{F_3}, S_5 \otimes \mathbf{F_3}\}] +$
$[\{p, q\} \circ \{S_0 \otimes F_1, S_3 \otimes F_1, S_5 \otimes F_1\}] =$
$q \circ \{\mathbf{C_{0,3}}, \mathbf{C_{3,3}}, \mathbf{C_{5,3}}\} + \{p, q\} \circ \{C_{0,1}, C_{3,1}, C_{5,1}\} =$
$q \circ \{\mathbf{C_e}\} + \{p, q\} \circ \{C_{0,1}, C_{3,1}, C_{5,1}\},$
$C_{0,1} = S_0 \otimes F_1 =$
$[\{\} \circ (S_1 + S_3)] \otimes [(q + r) \bullet \mathbf{F_3} + \triangle \bullet F_1] =$
$[\{\} \circ (S_1 + S_3)] \otimes [(q + r) \bullet \mathbf{F_3}] +$
$[\{\} \circ (S_1 + S_3)] \otimes [\triangle \bullet F_1] =$
$[\perp \circ \{S_1 \otimes \mathbf{F_3}, S_3 \otimes \mathbf{F_3}\}] + [\{\} \circ \{S_1 \otimes F_1, S_3 \otimes F_1\}] =$
$\{\} \circ \{C_{1,1}, C_{3,1}\},$
$C_{5,1} = S_5 \otimes F_1 =$
$[\{p, r\} \circ S_2] \otimes [(q + r) \bullet \mathbf{F_3} + \triangle \bullet F_1] =$
$[\{p, r\} \circ S_2] \otimes [(q + r) \bullet \mathbf{F_3}] + [\{p, r\} \circ S_2] \otimes [\triangle \bullet F_1] =$
$[r \circ S_2 \otimes \mathbf{F_3}] + [\{p, r\} \circ S_2 \otimes F_1] =$
$r \circ \mathbf{C_{2,3}} + \{p, r\} \circ C_{2,1} = r \circ \mathbf{C_e} + \{p, r\} \circ C_{2,1},$
$C_{2,1} = S_2 \otimes F_1 =$
$[p \circ (S_1 + S_3)] \otimes [(q + r) \bullet \mathbf{F_3} + \triangle \bullet F_1] =$
$[p \circ (S_1 + S_3)] \otimes [(q + r) \bullet \mathbf{F_3}] + [p \circ (S_1 + S_3)] \otimes [\triangle \bullet F_1] =$
$[\perp \circ \{S_1 \otimes \mathbf{F_3}, S_3 \otimes \mathbf{F_3}\}] + [p \circ \{S_1 \otimes F_1, S_3 \otimes F_1\}] =$
$p \circ \{C_{1,1}, C_{3,1}\},$
$\mathbf{C_e} = \triangle \circ \mathbf{C_e}.$

После исключения из композиции терминальных ветвей, ведущих в состояния $C_{12}$ и $C_{32}$, синхронная композиция примет вид

$C_{0,0} = \{\} \circ \{C_{1,1}, C_{3,1}\},$
$C_{1,1} = p \circ \{C_{4,1}\},$
$C_{3,1} = p \circ \{C_{0,1}, C_{5,1}\},$
$C_{4,,1} = q \circ \mathbf{C_e} + \{p, q\} \circ \{C_{0,1}, C_{3,1}, C_{5,1}\},$
$C_{0,1} = \{\} \circ \{C_{1,1}, C_{3,1}\},$
$C_{5,1} = r \circ \mathbf{C_e} + \{p, r\} \circ C_{2,1},$
$C_{2,1} = p \circ \{C_{1,1}, C_{3,1}\},$
$\mathbf{C_e} = \triangle \circ \mathbf{C_e}.$

### Алгоритм маркировки состояний синхронной композиции

С помощью алгоритма маркировки состояний проверим выполнимость Ф на М.

*Шаг 1.* Пометим маркером **F** все допускающие состояния композиции: F: $\mathbf{C_e}$.

*Шаг 2.* Далее в цикле будем добавлять к ним те состояния, из которых есть переходы в уже помеченные состояния:

F: $\mathbf{C_e}$, $C_{4,1}$, $C_{5,1}$, так как в $\mathbf{C_e}$ есть переходы из $C_{4,1}$, $C_{5,1}$;

F: $\mathbf{C_e}$, $C_{4,1}$, $C_{5,1}$, $C_{1,1}$, так как из $C_{1,1}$ есть переход в уже помеченное состояние $C_{4,1}$.

Больше никакие состояния не могут быть помечены. В частности, состояние $C_{3,1}$ нельзя отметить маркером F, поскольку оно содержит связанное продолжение $\{C_{0,1}, C_{5,1}\}$, а $C_{0,1}$ не было отмечено F.

Так как начальное состояние $C_{0,0}$ не было помечено F, допускающее состояние из него недостижимо. А это означает невыполнение Ф на М, что совпадает с результатами, представленными в работе [2].

### Проверка альтернативного свойства

Для этого же примера проверим выполнимость другого свойства $\Phi' = E((EX]p) \; U \; EF(q \vee r))$, которое, согласно представленному в работе [2] алгоритму верификации, будет выполняться на модели М.

Преобразуем свойство Ф' к RTL-виду:
$\Phi'_0 = E(\Phi'_1 \; U \; \Phi'_2) = \Phi'_2 + \Phi'_1 \circ \Phi'_0,$
$\Phi'_1 = EX]p = \triangle \circ ]p,$
$\Phi'_2 = EF(q \vee r) = (q + r) + \triangle \circ \Phi'_2$
$\Leftrightarrow$
$\Phi'_0 = [(q + r) + \triangle \circ \Phi'_2] + \triangle \circ ]p \circ \Phi'_0$
$\Leftrightarrow$
$F_0 = [(q + r) + \triangle \circ F_1] + \triangle \circ F_2,$
$F_1 = (q + r) + \triangle \circ F_1,$
$F_2 = ]p \circ F_0.$

Чтобы модель верифицирующего свойства описывала реагирующую систему, введем еще одно состояние, которое и станет допускающим (выделено жирным шрифтом):

$F_0 = (q + r) \circ \mathbf{F_3} + \triangle \circ (F_1 + F_2),$
$F_1 = (q + r) \circ \mathbf{F_3} + \triangle \circ F_1,$
$F_2 = ]p \circ F_0,$
$\mathbf{F_3} = \triangle \circ \mathbf{F_3}.$

Поскольку $\mathbf{F_3}$ наступает только после выполнения $EF(q \vee r)$, переход в это состояние задается оператором «∘». Необходимо, чтобы хотя бы один переход состояния модели был помечен q или r.

Выполним построение синхронной композиции $M \otimes \Phi'$.

В дальнейшем для краткости будем обозначать синхронную композицию $(S_i \otimes F_j)$ через $C_{ij}$, а $(S_i \otimes \mathbf{F_3})$ через $\mathbf{C_e}$:

$C_{0,0} = \{\} \circ (C_{1,1} + C_{3,1})$,
$C_{1,1} = p \circ C_{4,1}$,
$C_{3,1} = p \circ (C_{0,1} + C_{5,1})$,
$C_{4,1} = q \circ \mathbf{C_e} + \{p, q\} \circ (C_{0,1} + C_{3,1} + C_{5,1})$,
$C_{0,1} = \{\} \circ (C_{1,1} + C_{3,1})$,
$C_{5,1} = r \circ \mathbf{C_e} + \{p, r\} \circ C_{2,1}$,
$C_{2,1} = p \circ (C_{1,1} + C_{3,1})$,
$\mathbf{C_e} = \triangle \circ \mathbf{C_e}$.

Рассмотрим по шагам работу алгоритма маркировки:

1. F: $\mathbf{C_e}$,
2. F: $\mathbf{C_e}$, $C_{4,1}$, $C_{5,1}$,
3. F: $\mathbf{C_e}$, $C_{4,1}$, $C_{5,1}$, $C_{1,1}$, $C_{3,1}$,
4. F: $\mathbf{C_e}$, $C_{4,1}$, $C_{5,1}$, $C_{1,1}$, $C_{3,1}$, $C_{0,0}$, $C_{0,1}$, $C_{2,1}$.

Состояние $C_{0,0}$ входит во множество помеченных F, следовательно, Φ' выполняется на M, что совпадает с результатами, которые можно получить, используя алгоритм проверки, приведенный в работе [2].

### Задача о козе, капусте и волке

***Условия задачи.*** Лодочнику необходимо перевезти с левого берега на правый трех пассажиров: волка, козу и капусту. В любой момент времени лодочник может перевозить только одного пассажира. Главным условием задачи является то, что лодочнику нельзя оставлять без присмотра козу с капустой и волка с козой на любом из берегов, поскольку капуста и коза могут быть съедены.

***Решение.*** Введем следующие обозначения: b – лодочник (boatman), w – волк (wolf), g – коза (goat) и c – капуста (cabbage). Запись вида ⌉x будет означать нахождение соответствующего субъекта на левом берегу, а запись x – на правом берегу, где $x \in \{b, w, g, c\}$. Таким образом, возможны 16 различных состояний.

Начальным состоянием $S_0$ является состояние {⌉b, ⌉g, ⌉w, ⌉c}, когда все находятся на левом берегу, а заключительным состоянием $S_{15}$ – {b, g, w, c}, когда все находятся на правом берегу.

Вначале на базе RTL-нотации построим модель M, описывающую все возможные варианты развития событий для поставленной задачи, включая те, которые противоречат ее главному условию. При появлении новых состояний будем записывать их в левой колонке:

$S_0 = \{\} \circ S_1$,
{⌉b, ⌉g, ⌉w, ⌉c}: $S_1 = \{b, g\} \circ S_2 + \{b, w\} \circ S_3 + \{b, c\} \circ S_4 + \{b\} \circ S_5$,
{b, g, ⌉w, ⌉c}: $S_2 = \{g\} \circ S_6 + \{\} \circ S_1$,
{b, ⌉g, w, ⌉c}: $S_3 = \{w\} \circ S_7 + \{\} \circ S_1$,
{b, ⌉g, ⌉w, c}: $S_4 = \{c\} \circ S_8 + \{\} \circ S_1$,
{b, ⌉g, ⌉w, ⌉c}: $S_5 = \{\} \circ S_1$,
{⌉b, g, ⌉w, ⌉c}: $S_6 = \{b, g, w\} \circ S_9 + \{b, g, c\} \circ S_{10} + \{b, g\} \circ S_2$,
{⌉b, ⌉g, w, ⌉c}: $S_7 = \{b, g, w\} \circ S_9 + \{b, w, c\} \circ S_{11} + \{b, w\} \circ S_3$,
{⌉b, ⌉g, ⌉w, c}: $S_8 = \{b, g, c\} \circ S_{10} + \{b, w, c\} \circ S_{11} + \{b, c\} \circ S_4$,
{b, g, w, ⌉c}: $S_9 = \{w\} \circ S_7 + \{g\} \circ S_6 + \{g, w\} \circ S_{12}$,
{b, g, ⌉w, c}: $S_{10} = \{c\} \circ S_8 + \{g\} \circ S_6 + \{g, c\} \circ S_{13}$,
{b, ⌉g, w, c}: $S_{11} = \{c\} \circ S_8 + \{w\} \circ S_7 + \{w, c\} \circ S_{14}$,
{⌉b, g, w, ⌉c}: $S_{12} = \{b, g, w, c\} \circ S_{15} + \{b, g, w\} \circ S_9$,
{⌉b, g, ⌉w, c}: $S_{13} = \{b, g, w, c\} \circ S_{15} + \{b, g, c\} \circ S_{10}$,
{⌉b, ⌉g, w, c}: $S_{14} = \{b, g, w, c\} \circ S_{15} + \{b, w, c\} \circ S_{11}$,
{b, g, w, c}: $S_{15} = \{w, c\} \circ S_{14} + \{g, c\} \circ S_{13} + \{g, w\} \circ S_{12} + \{g, w, c\} \circ S_{16}$,
{⌉b, g, w, c}: $S_{16} = \{b, g, w, c\} \circ S_{15}$.

Выполним проверку CTL-свойства Φ, целью которого является доказательство того, что задача не имеет решения:

$\Phi = A][(g \equiv c \vee g \equiv w) \rightarrow (g \equiv b)\ \mathbf{U}\ (b \wedge g \wedge w \wedge c)]$.

«Среди всех возможных путей не существует такого, на котором все переправятся на правый берег при условии, что, когда коза находится на одном берегу с капустой или волком, рядом с ней находится лодочник».

Преобразуем условие Φ, воспользовавшись следующим правилом: $⌉(\varphi_1 U \varphi_2) = ⌉\varphi_1 R ⌉\varphi_2$,

$\Phi = A[(g \equiv c \vee g \equiv w) \wedge ⌉(g \equiv b)\ \mathbf{R}\ (⌉b \vee ⌉g \vee ⌉w \vee ⌉c)]$.

Получим рекурсивное определение оператора **R** через оператор **U**:

$⌉(\varphi_1 U \varphi_2) = ⌉(\varphi_2 \vee \varphi_1 \wedge X(\varphi_1 U \varphi_2)) \equiv$
$⌉(\varphi_1 U \varphi_2) = ⌉\varphi_2 \wedge (⌉\varphi_1 \vee X⌉(\varphi_1 U \varphi_2)) \equiv$
$⌉(\varphi_1 U \varphi_2) = ⌉\varphi_2 \wedge ⌉\varphi_1 \vee ⌉\varphi_2 \wedge X⌉(\varphi_1 U \varphi_2)$.

$\varphi_1 R \varphi_2 = \varphi_2 \wedge \varphi_1 \vee \varphi_2 \wedge X(\varphi_1 R \varphi_2)$.

В RTL-нотации данный оператор задается следующим образом: $F = \{\varphi_1, \varphi_2\} + \varphi_2 \circ F$.

Выполним преобразование свойства Φ к RTL-виду:

$\Phi = A(\alpha\ R\ \beta) = \{\alpha, \beta\} + \beta \bullet \Phi_0$,

$\alpha = (g \equiv c \vee g \equiv w) \wedge ⌉(g \equiv b) = (g \equiv c \vee g \equiv w) \wedge (g \equiv ⌉b) = (g \equiv c \equiv ⌉b) + (g \equiv w \equiv ⌉b)$,

$\beta = ⌉b + ⌉g + ⌉w + ⌉c$.

Чтобы модель верифицирующего свойства описывала реагирующую систему, введем допускающее состояние $\mathbf{F_1}$ (выделено жирным шрифтом):

$F_0 = \{\alpha, \beta\} \bullet \mathbf{F_1} + \beta \bullet F_0$,

$\mathbf{F_1} = \triangle \circ \mathbf{F_1}$.

Переход в $\mathbf{F_1}$ задается оператором «•», поскольку свойство считается выполненным лишь при его выполнении по всем возможным путям.

***Замечание.*** Такой же результат получим, если осуществим непосредственный переход от выражения $⌉(\varphi_1 U \varphi_2)$ к RTL-виду.

Выполним построение синхронной композиции $M \otimes \Phi$.

В дальнейшем для краткости будем обозначать синхронную композицию $(S_i \otimes F_j)$ через $C_{i,j}$.

Допускающими состояниями композиции являются те, которые содержат допускающие состояния верифицирующего свойства. По-

поскольку попадание в состояние $C_{i,1}$ (i = 0÷5) порождает лишь переходы в состояния $C_{j,1}$ (j = 0÷5), все соответствующие уравнения могут быть сразу же заменены на одно уравнение $\mathbf{C_e} = \triangle \circ \mathbf{C_e}$ и при появлении в правой части уравнений состояний $C_{i,1}$ их можно сразу заменять на $\mathbf{C_e}$:

$C_{0,0} = \{\} \circ C_{1,0}$,

$C_{1,0} = \{\perp \circ \mathbf{C_e}, \{b, w\} \circ \mathbf{C_e}, \{b, c\} \circ \mathbf{C_e}, b \circ \mathbf{C_e}\} + \{\{b, g\} \circ C_{2,0}, \{b, w\} \circ C_{3,0}, \{b, c\} \circ C_{4,0}, b \circ C_{5,0}\}$,

$C_{2,0} = \{g \circ C_{6,0}, \{\} \circ C_{1,0}\}$,

$C_{3,0} = \{w \circ C_{7,0}, \{\} \circ C_{1,0}\}$,

$C_{4,0} = \{c \circ C_{8,0}, \{\} \circ C_{1,0}\}$,

$C_{5,0} = \{\} \circ C_{1,0}$,

$C_{6,0} = \{\{b, g, w\} \circ C_{9,0}, \{b, g, c\} \circ C_{10,0}, \{b, g\} \circ C_{2,0}\}$,

$C_{7,0} = \{\perp \circ \mathbf{C_e}, \perp \circ \mathbf{C_e}, \{b, w\} \circ \mathbf{C_e}\} + \{\{b, g, w\} \circ C_{9,0}, \{b, w, c\} \circ C_{11,0}, \{b, w\} \circ C_{3,0}\}$,

$C_{8,0} = \{\perp \circ \mathbf{C_e}, \perp \circ \mathbf{C_e}, \{b, c\} \circ \mathbf{C_e}\} + \{\{b, g, c\} \circ C_{10,0}, \{b, w, c\} \circ C_{11,0}, \{b, c\} \circ C_{4,0}\}$,

$C_{9,0} = \{\perp \circ \mathbf{C_e}, \perp \circ \mathbf{C_e}, \{g, w\} \circ \mathbf{C_e}\} + \{\{w\} \circ C_{7,0}, \{g\} \circ C_{6,0}, \{g, w\} \circ C_{12,0}\}$,

$C_{10,0} = \{\perp \circ \mathbf{C_e}, \perp \circ \mathbf{C_e}, \{g, c\} \circ \mathbf{C_e}\} + \{\{c\} \circ C_{8,0}, \{g\} \circ C_{6,0}, \{g, c\} \circ C_{13,0}\}$,

$C_{11,0} = \{\{c\} \circ C_{8,0}, \{w\} \circ C_{7,0}, \{w, c\} \circ C_{14,0}\}$,

$C_{12,0} = \{\perp \circ C_{15,0}, \{b, g, w\} \circ C_{9,0}\}$,

$C_{13,0} = \{\perp \circ C_{15,0}, \{b, g, c\} \circ C_{10,0}\}$,

$C_{14,0} = \{\perp \circ C_{15,0}, \{b, w, c\} \circ C_{11,0}\}$,

$C_{15,0} = \{\perp \circ \mathbf{C_e}, \{g, c\} \circ \mathbf{C_e}, \{g, w\} \circ \mathbf{C_e}, \{g, w, c\} \circ \mathbf{C_e}\} + \{\{w, c\} \circ C_{14,0}, \{g, c\} \circ C_{13,0}, \{g, w\} \circ C_{12,0}, \{g, w, c\} \circ C_{16,0}\}$,

$C_{16,0} = \perp$,

$\mathbf{C_e} = \triangle \circ \mathbf{C_e}$.

После исключения всех терминальных ветвей синхронная композиция примет вид $C_{0,0} = \perp$, откуда следует тривиальное решение о невыполнимости Ф на М.

## Проверка альтернативного свойства

Пусть теперь $\Phi' = E[(g \equiv c \vee g \equiv w) \rightarrow (g \equiv b)\ \mathbf{U}\ (b \wedge g \wedge w \wedge c)]$ – «Среди всех возможных путей существует хотя бы один такой, на котором все переправятся на правый берег при условии, что, когда коза находится на одном берегу с капустой или волком, то рядом с ней находится лодочник».

Выполним преобразование свойства Ф' к RTL-виду:

$\Phi' = E(\alpha\ U\ \beta) = \beta + \alpha \circ \Phi'$,

$\alpha = (g \equiv c \vee g \equiv w) \rightarrow (g \equiv b) = (g \equiv \rceil c \wedge g \equiv \rceil w) \vee (g \equiv b) = \{g \equiv \rceil c \equiv \rceil w\} + (g \equiv b)$,

$\beta = (b \wedge g \wedge w \wedge c) = \{b, g, w, c\}$.

Введем еще одно состояние, чтобы модель верифицирующего свойства описывала реагирующую систему, которое и станет допускающим состоянием (выделено жирным шрифтом):

$F_0 = \beta \circ \mathbf{F_1} + \alpha \circ F_0$,

$\mathbf{F_1} = \triangle \circ \mathbf{F_1}$.

Переход в $\mathbf{F_1}$ задается оператором «∘», поскольку верифицирующее свойство выполняется для некоторого состояния модели М тогда и только тогда, когда β выполнится хотя бы для одного из его переходов.

Выполним построение синхронной композиции $M \otimes \Phi'$ с учетом принятых обозначений:

$C_{0,0} = \{\} \circ C_{1,0}$,

$C_{1,0} = \{b, g\} \circ C_{2,0}$,

$C_{2,0} = \{g\} \circ C_{6,0} + \{\} \circ C_{1,0}$,

$C_{6,0} = \{b, g, w\} \circ C_{9,0} + \{b, g, c\} \circ C_{10,0} + \{b, g\} \circ C_{2,0}$,

$C_{9,0} = \{w\} \circ C_{7,0} + \{g\} \circ C_{6,0}$,

$C_{10,0} = \{c\} \circ C_{8,0} + \{g\} \circ C_{6,0}$,

$C_{7,0} = \{b, g, w\} \circ C_{9,0} + \{b, w, c\} \circ C_{11,0}$,

$C_{8,0} = \{b, g, c\} \circ C_{10,0} + \{b, w, c\} \circ C_{11,0}$,

$C_{11,0} = \{c\} \circ C_{8,0} + \{w\} \circ C_{7,0} + \{w, c\} \circ C_{14,0}$,

$C_{14,0} = \{b, g, w, c\} \circ \mathbf{C_e} + \{b, w, c\} \circ C_{11,0}$,

$\mathbf{C_e} = \triangle \circ \mathbf{C_e}$.

Применив алгоритм маркировки, получим следующий результат:

1. F: $\mathbf{C_e}$,
2. F: $\mathbf{C_e}$, $C_{14,0}$,
3. F: $\mathbf{C_e}$, $C_{14,0}$, $C_{11,0}$,
4. F: $\mathbf{C_e}$, $C_{14,0}$, $C_{11}$, $C_{7,0}$, $C_{8,0}$,
5. F: $\mathbf{C_e}$, $C_{14,0}$, $C_{11}$, $C_{7,0}$, $C_{8,0}$, $C_{9,0}$, $C_{10,0}$,
6. F: $\mathbf{C_e}$, $C_{14,0}$, $C_{11}$, $C_{7,0}$, $C_{8,0}$, $C_{9,0}$, $C_{10,0}$, $C_{6,0}$,
7. F: $\mathbf{C_e}$, $C_{14,0}$, $C_{11}$, $C_{7,0}$, $C_{8,0}$, $C_{9,0}$, $C_{10,0}$, $C_{6,0}$, $C_{2,0}$,
8. F: $\mathbf{C_e}$, $C_{14,0}$, $C_{11}$, $C_{7,0}$, $C_{8,0}$, $C_{9,0}$, $C_{10,0}$, $C_{6,0}$, $C_{2,0}$, $C_{1,0}$,
9. F: $\mathbf{C_e}$, $C_{14,0}$, $C_{11}$, $C_{7,0}$, $C_{8,0}$, $C_{9,0}$, $C_{10,0}$, $C_{6,0}$, $C_{2,0}$, $C_{1,0}$, $C_{0,0}$.

Состояние $C_{0,0}$ входит во множество состояний, помеченных F, следовательно, Ф' выполняется на М. Так, взяв все возможные пути переходов из конечного состояния в начальное (не проходящими несколько раз через одно и то же состояние), получим все возможные решения данной задачи:

$C_{0,0} \rightarrow C_{1,0} \rightarrow C_{2,0} \rightarrow C_{6,0} \rightarrow C_{9,0} \rightarrow C_{7,0} \rightarrow C_{11,0} \rightarrow C_{14,0} \rightarrow \mathbf{C_e}$,

$C_{0,0} \rightarrow C_{1,0} \rightarrow C_{2,0} \rightarrow C_{6,0} \rightarrow C_{10,0} \rightarrow C_{8,0} \rightarrow C_{11,0} \rightarrow C_{14,0} \rightarrow \mathbf{C_e}$.

Обозначая положительными значениями предикатов факт переезда соответствующих субъектов на правый берег, а отрицательными значениями предикатов – факт переезда на левый берег, получим следующие последовательности действий, являющиеся решениями задачи:

$bg \Rightarrow \rceil b \Rightarrow bw \Rightarrow \rceil b \rceil g \Rightarrow bc \Rightarrow \rceil b \Rightarrow bg$,

$bg \Rightarrow \rceil b \Rightarrow bc \Rightarrow \rceil b \rceil g \Rightarrow bw \Rightarrow \rceil b \Rightarrow bg$.

## Оценка сложности

Верификация данным методом включает в себя два отдельных алгоритма:

– алгоритм построения синхронной композиции,

– алгоритм маркировки состояний.

Для оценки сложности метода в целом нужно оценить сложность каждого из алгоритмов отдельно. Произведем вначале оценку сложности алгоритма построения синхронной композиции. Обозначим через $m$ число уравнений (состояний) верифицируемой модели, через $p$ – число уравнений верифицирующего свойства, через $n$ – максимальное число переходов любого из уравнений верифицируемой модели, а через $t$ – максимальное число переходов любого из уравнений верифицирующего свойства. Оценка сложности алгоритма синхрокомпозиции в таком случае может быть задана выражением $O(mpnt)$.

Оценим сложность алгоритма маркировки состояний. Поскольку общее число операций данного алгоритма зависит от максимального числа состояний синхрокомпозиции и не может превысить это значение, оценка сложности есть $O(mp)$.

Таким образом, суммарная сложность верификации данным методом составляет $O(mpnt + mp) \equiv O(mp(nt + 1)) \cong O(mpnt)$. Проанализируем, как именно следует понимать данную оценку. Положим $m \sim p$ и $n \sim t$. Это означает, что размеры верифицируемой модели и верифицирующего свойства изменяются пропорционально. В этом случае оценка примет вид $O(m^2n^2)$. На первый взгляд, может показаться, что в таком виде оценка способна стать серьезной преградой на пути верификации моделей больших систем. Однако при верификации больших систем размеры верифицирующих свойств, как правило, остаются крайне небольшими (не сопоставимыми с размерами проверяемых моделей), а следовательно, в таком виде оценка может быть применена лишь для систем небольших размеров. Принимая во внимание тот факт, что для больших систем $m \gg p$ и $n \gg t$, оценка примет вид $O(Cmn) \cong O(mn)$, где $C = pt$ есть константа, обозначающая предельное значение произведения переменных $p$ и $t$. Таким образом, оценка сложности верификации данным методом напрямую зависит от условий взаимосвязей переменных $m$, $p$, $n$ и $t$ между собой. В таблице 2 представлены различные вариации данной оценки для различных условий.

*Таблица 2*

**Вариации оценки сложности метода верификации CTL-свойств на базе RTL-нотации**

*Table 2*

**Variations in assessing the complexity of the CTL properties verification method based on RTL notation**

| Вариация | Условие | Описание |
|---|---|---|
| $O(m^2n^2)$ | $m \sim p$, $n \sim t$ | Число состояний и переходов верифицируемой модели и верифицирующего свойства изменяется пропорционально. Данная оценка применима для моделей систем небольших размеров |
| $O(mn)$ | $m \gg p$, $n \gg t$ | Число состояний и переходов верифицируемой модели значительно превосходит число состояний и переходов верифицирующего свойства. Данная оценка применима для моделей систем больших размеров |
| $O(m)$ | $m \gg p$, $n \gg t$, $m \sim n$ | Число состояний и переходов верифицируемой модели значительно превосходит число состояний и переходов верифицирующего свойства, при этом число переходов модели изменяется пропорционально числу ее состояний. Данная оценка применима для моделей систем очень больших размеров |

## Заключение

Представленные в статье результаты в полной мере подтверждают идею авторов о возможности использования RTL-нотации в качестве средства для верификации CTL-свойств. Более того, несмотря на кажущуюся громоздкость приведенных в статье примеров, выполненная оценка сложности метода подтверждает возможность его использования для верификации систем больших размеров.

Так, обновленное в данной статье синтаксическое определение нотации позволило проще, понятнее и единообразно задавать выражения обеих логик: LTL и CTL, а описанные аксиомы построения синхронной композиции дают возможность легко осуществлять проверку их выполнимости относительно произвольных моделей систем, также заданных с помощью RTL.

Алгоритм маркировки дополнил процесс верификации свойств логики ветвящегося времени, упростив и формализовав этап анализа синхрокомпозиции.

Таким образом, к настоящему времени авторам удалось разработать нотацию, которая обладает всеми необходимыми средствами и методами для выполнения полного цикла верификации, начиная от построения моделей систем, заканчивая проверкой выполнимости свойств на этих моделях, единообразно для логик линейного и ветвящегося времени. Благодаря этим возможностям RTL-нотация является более гибким и мощным инструментом, чем каждая из двух логик в отдельности.

Представленные в статье результаты подтверждают идею о возможности унифицированного представления формул темпоральных логик LTL и CTL для анализа свойств моделей систем.

### *Литература*

1. Кораблин Ю.П., Шипов А.А. Унифицированное представление формул логик LTL и CTL системами рекурсивных уравнений // Программные продукты и системы. 2019. Т. 32. № 1. С. 20–25. DOI: 10.15827/0236-235X.125.020-025.

2. Карпов Ю.Г. Model Checking. Верификация параллельных и распределенных программных систем. СПб: Изд-во БХВ-Петербург, 2010. 552 с.

3. Кларк Э.М., Грамберг О., Пелед Д. Верификация моделей программ. Model Checking. М.: Изд-во МЦНМО, 2002. 416 с.

4. Pnueli. The temporal logic of program. Proc. 18th Annyv. Symp. on Foundation of Computer Science. 1977, pp. 46–57.

5. Manna Z., Pnueli A. The temporal logic of reactive and concurrent systems: Specification. 1992, 427 p.

6. Kroger F., Merz S. Temporal logic and state systems. Springer Publ., 2008, 436 p.

7. Huth M., Ryan M. Logic in computer science: modelling and reasoning about systems. Cambridge Univ. Press, 2004, 443 p.

8. Хоар Ч. Взаимодействующие последовательные процессы; [пер. с англ.]. М.: Мир, 1989. 264 с. DOI: 10.1145/359576.359585.

9. Кораблин Ю.П., Шипов А.А., Кочергин А.С. Верификация моделей систем на базе эквациональной характеристики формул LTL // Программные продукты и системы. 2017. № 3. С. 456–460. DOI: 10.15827/0236-235X.119.456-460.

10. Olderog E.-R., Apt K.R. Fairness in parallel programs, the transformational approach. ACM TOPLAS, 1988, vol. 10, iss. 3, pp. 420–455. DOI: 10.1145/44501.44504.

11. Li Y.M., Li Y., Ma Zh. Computation tree logic model checking based on possibility measures. Fuzzy Sets and Systems. 2015, no. 1, vol. 262, pp. 44–59.

Software & Systems
DOI: 10.15827/0236-235X.128.547-555

Received 12.08.19
2019, vol. 32, no. 4, pp. 547–555

## Systems model verification based on equational characteristics of CTL formulas

***Yu.P. Korablin*** [1,2]*, Dr.Sc. (Engineering), Professor, y.p.k@mail.ru*
***A.A. Shipov*** [3]*, Ph.D. (Engineering), Senior Engineer-Programmer, a-j-a-1@yandex.ru*

[1] *Russian Technological University – MIREA, Moscow, 119454, Russian Federation*
[2] *National Research University MPEI, Moscow, 111250, Russian Federation*
[3] *RSC Technologies Group, Moscow, 121170, Russian Federation*

**Abstract.** The paper proposes and examines the RTL notation based on systems of recursive equations and standard Linear Temporal Logic (LTL) semantic definitions and the Computational Tree Logic (CTL). When this notation was still called RLTL, the previous works of the authors showed that it enables easy formulation and verifying of LTL properties with respect to system models, even with those that are also specified using the RLTL notation. Then the authors expanded the capabilities of the RLTL notation, so it has become possible to formulate LTL and CTL expressions. Therefore, the first version of the RTL notation was created.

This article presents the second version of the RTL, which was the result of refinement and simplification of notation semantic definitions, which allowed increasing the visibility and readability of its expressions.

The purpose of the article is to demonstrate the possibility of using the RTL notation as a tool to formulate and verify properties defined by formulas of both LTL and CTL logics using common axioms and rules. This lets RTL to become a single and universal notation for these logics. At the same time, it is possible for RTL to include expressiveness of other temporal logics too by minor additions to its basic definitions. It means that in future it is possible for RTL to become a full-fledged universal temporal logic that has all of the necessary tools and means for implementing all stages of verification.

**Keywords:** verification, Model Checking, equational characteristic of RTL, temporal logic formulas, LTL, CTL, recursive equation systems.

### *References*

1. Korablin Yu.P., Shipov A.A. Unified representation of LTL and CTL logic formulas by recursive equation systems. *Software & Systems*. 2019, vol. 32, no. 1, pp. 20–25. DOI: 10.15827/0236-235X.125.020-025 (in Russ.).
2. Karpov Yu.G. *Model Checking. Verification of Parallel and Distributed Software Systems*. St. Petersburg, BHV Peterburg Publ., 2010, 552 p.
3. Klark E.M., Gramberg O., Peled D. *Program Model Verification. Model Checking*. Moscow, MTsNMO Publ., 2002, 416 p.
4. Pnueli A. The temporal logic of program. *Proc. 18th Annyv. Symp. on Foundation of Computer Science*. 1977, pp. 46–57.
5. Manna Z., Pnueli A. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. 1992, 427 p.
6. Kroger F., Merz St. *Temporal Logic and State Systems*. Springer, 2008, 436 p.
7. Huth M., Ryan M. *Logic in Computer Science: Modelling and Reasoning About Systems*. Cambridge Univ. Press, 2004, 443 p.
8. Hoare C.A.R. Communicating sequential processes. *Comm. of the ACM*. 1978, vol. 21, no. 8, pp. 666–677. DOI: 10.1145/359576.359585.
9. Korablin Yu.P., Shipov A.A., Kochergin A.S. Verification of system models based on the equational characteristics of LTL formulas. *Software & Systems*. 2017, vol. 30, no. 3, pp. 456–460 (in Russ.).
10. Olderog E.-R., Apt K.R. Fairness in parallel programs, the transformational approach. *ACM TOPLAS*. 1988, vol. 10, iss. 3, pp. 420–455. DOI: 10.1145/44501.44504.
11. Li Y.M., Li Y., Ma Zh. Computation tree logic model checking based on possibility measures. *Fuzzy Sets and Systems*. 2015, vol. 262, pp. 44–59.

**Для цитирования**

Кораблин Ю.П., Шипов А.А. Верификация моделей систем на базе эквациональной характеристики формул CTL // Программные продукты и системы. 2019. Т. 32. № 4. С. 547–555. DOI: 10.15827/0236-235X.128.547-555.

**For citation**

Korablin Yu.P., Shipov A.A. Systems model verification based on equational characteristics of CTL formulas. *Software & Systems*. 2019, vol. 32, no. 4, pp. 547–555 (in Russ.). DOI: 10.15827/0236-235X.128.547-555.