КОМПЬЮТЕРНЫЕ НАУКИ

УДК 004.82:16

МОДИФИКАЦИЯ ОПЕРАТОРОВ CTL ЛОГИКИ И ИХ АЛГОРИТМЫ ПРОВЕРКИ НА СТРУКТУРЕ КРИПКЕ

Ю.А. Мазуров

Московский государственный социальный университет
129226, г. Москва, ул. Вильгельма Пика, д. 4, стр. 1

Аннотация. В настоящей статье предложено расширение некоторых темпоральных операторов CTL логики, позволяющее явно указывать количество состояний системы или количество путей, в которых выполняется заданное свойство системы. Дается семантика, формальная грамматика, а так же соответствующие алгоритмы проверки выполнимости на структуре Крипке, расширенных темпоральных операторов.

Ключевые слова: верификация, Model Checking, CTL логика, структура Крипке.

Введение

Уже не одно десятилетие ведутся интенсивные исследования по развитию и применению аппарата темпоральных логик для спецификации и верификации параллельных распределенных систем. На заре развития темпоральной логики, стояла проблема описания требований к программным системам, относительно цепочки состояний структуры Крипке. Решение этой проблемы А. Пнуэли, привело к созданию LTL (Linear Temporal Logic) логики [1]. При этом у LTL логики, была проблема, состоящая в том, что LTL логика не различала точек ветвления. То есть в терминах LTL логики можно сделать утверждения обо всех возможных вычислениях, не позволяя при этом выразить те свойства системы, для выявления которых необходимо только существование вычисления, удовлетворяющего определенным требованиям. Попытка решить эту проблему, привела к созданию CTL* (Extended Computational Tree Logic)[2]. Введение правила, согласно которому любая темпоральная формула должна начинаться с квантора пути E (Exist) привело к созданию логики ветвящегося времени - CTL (Computational Tree Logic)

Вследствие того, что спецификация требований к системе, записанная с помощью CTL логики, нередко бывает достаточно объемной, представляется актуальным выполнить модификацию темпоральной логики, направленную на увеличение компактности спецификаций.

Рассмотрим следующее требование к системе: если пин-код банковской карточки, ввели неправильно N раз, то карточка блокируется.

Выполним декомпозицию моделируемой системы. В рассматриваемой нами системе, для нас важны только следующие состояния системы:

1. осуществляется ввод пин кода (Состояние S_1);

2. система готова к началу выполнения операции по карточке (состояние S_2);
3. система не может начать выполнение операции по карточке (состояние S_3).

Далее, среди всех событий, которые могут происходить в системе, нас будут интересовать только следующие:

1. пин код введен - верно;
2. пин код введен - неверно;
3. карточка успешно прошла авторизацию;
4. карточка заблокирована.

Структура Крипке для программной системы, осуществляющей работу с банковской карточкой:

$S = \{S_1, S_2, S_3\}$;

S_1- это начальное состояние;

$AP = \{\gamma, \alpha, \beta, \delta\}$- это множество атомарных предикатов,

где γ – правильный ввод пин кода, α - неправильный ввод пин кода, β - блокировка карточки, δ - карточка успешно прошла авторизацию.

Для того, что бы мы могли ответить на вопрос о том, выполняется ли на модели Крипке требование, предъявленное к системе, требования необходимо переписать, используя CTL логику. При N = 3, наша спецификация будет выглядеть так: $EX(\alpha \wedge EX(\alpha \wedge AX(\beta))))$<1>. Запись <1> уже достаточно объемная. Хотя, в требовании сказано, только о том, что атомарное высказывание α должно повториться три раза. Запись <1> стала бы еще длиннее, если высказывание α должно было бы выполняться большее количество раз. Или если бы в требовании было бы сказано, о том, что не одно, а целое множество атомарных высказываний, должно повторяться определенное количество раз.

На практике встречаются требования к системам, в которых есть явное упоминание на то, что какое то свойство или группа свойств системы, должна выполняться заранее заданное количество раз. Требования к системе, записанные с помощью темпоральных операторов <1>, в явном виде не содержит информацию о том, что какое то атомарное высказывание должно выполняться N раз. Эту информацию можно получить, только выполнив анализ спецификации <1> к системе. Приведенная спецификация, достаточно мала и не вызывает затруднений для анализа, на практике же могут встречаться более объемные требования, анализ которых может потребовать значительных навыков от исследователя.

Таким образом, по мнению автора, проблема увеличения выразительности и ясности темпоральных формул является актуальной задачей. В рамках решения поставленной проблемы, в этой статье предложены следующие расширенные темпоральные операторы:

1. E_e (Extended Existential Path Quantifier);
2. F_e (Extended Future Time Operator);
3. G_e (Extended Global Time Operator);
4. U_e (Extended Until Time Operator).

Дана семантика и формальная грамматика указанных расширенных темпоральных операторов. Так же в статье предложены вычислительные процедуры доказательства корректности расширенных темпоральных операторов на структуре Крипке, которые позволяют вычислять все состояния системы, в которых выполняется заданное высказывание.

Общие понятия

Определение 1. Формулами состояния будем называть те темпоральные формулы, которые способны обращаться в истину в некотором состоянии [3].

Определение 2. Формулами пути будем называть те темпоральные формулы, которые способны быть истинными на протяжении некоторого пути [3].

Определение 3. Вычислениями будем называть последовательность состояний S_1 … S_n, которые может проходить система, в процессе своего функционирования.
Вычисления будем обозначать следующим образом: υ_i. Здесь индекс указывает на то, что у системы существует множество возможных путей, по которым может происходить вычисление в системе.
Пример вычислений: $\upsilon_1 = S_1S_3S_5S_7S_9$;
$\upsilon_2 = S_2S_4S_6S_8S_{10}$.

Определение 4. Структура Крипке[4] – это пятерка M = (S, S_0, R, AP, L), где:

S - это конечное непустое множество состояний;

$S_0 \in S$ - непустое множество начальных состояний;

$R \subseteq S \times S$ - тотальное отношение на S, т.е. множество переходов удовлетворяющих требованию: $(\forall s \in S)(\exists s' \in S)(s, s') \in R$ (из любого состояния есть переход);

AP – конечное множество атомарных предикатов;

$L{:}S \rightarrow 2^{AP}$ - функция пометок (каждому состоянию отображение L сопоставляет множество истинных в нем атомарных предикатов).

Замечание 1. запись M, $s \models \varphi$ означает, что формула φ истинна в состоянии s структуры Крипке M.

Определение 5. Атомарное высказывание[4] – это утверждение, которое может принимать истинное или ложное значение и от структуры которого мы абстрагируемся.

Определение 6. Операторы F_e, G_e, U_e являются расширениями соответственно следующих темпоральных операторов: F, G, U.

Определение 7. Квантор E_e является расширением следующего квантора пути: E.

Определение 8. Истинность оператора U_e на вычислении υ_i, будем обозначать $\upsilon_i \models U_e$.

Для примера вычислений, приведенного в определении 3, запись $\upsilon_i \models U_e$ будет означать, что оператор U_e принимает истинное значение в состояниях: $S_1S_3S_5S_7S_9$. Запись же вида $S_1 \models U_e$ будет иметь следующую семантику: оператор U_e принимает значение “true” в состоянии S_1.

Формальная грамматика и семантика расширенного квантора пути и расширенных темпоральных операторов.

Формальная грамматика: формальную грамматику запишем, используя расширенную форму Бэкуса-Наура:

$$E_e ::= (E_e^N)(X \mid F_e \mid G_e);$$
$$F_e ::= (F_e^N)(\varphi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi);$$
$$G_e ::= (G_e^N)(\varphi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi);$$

$$U_e ::= (E_e)(leftPart)U_e(E_e)(rightPart);$$
$$leftPart ::= p \mid \neg p \mid p \wedge p \mid p \vee p;$$
$$rightPart ::= p \mid \neg p \mid p \wedge p \mid p \vee p,$$

где X – next time operator, F_e - extended future time operator, G_e - extended global time operator, φ – это формула CTL_e логики, p – это атомарное высказывание.
Определение 9. Семантикой темпоральной логики являются интерпретации данных формальных утверждений.

Семантика.
Формула $E_e^{\ N} X\varphi$ означает: существует N путей, начинающихся из состояния S_t и у которых в состоянии S_{t+1} истинна формула φ.

1. Формула $E_e^{\ N} F_e^{\ M} \varphi$ означает: существует N путей, начинающихся из состояния S_t, на которых в будущем есть M состояний (не включая состояние S_t), в которых истинна формула φ.
2. Формула $E_e^{\ N} G_e^{\ M} \varphi$ означает: существует N путей, начинающихся из состояния S_t, таких, что в каждом пути, существует M состояний, начиная с состояния S_t, в каждом из которых будет истинна формула φ.
3. Формула $F_e^{\ N} \varphi$ означает: начиная из состояния S_t, существует всего N состояний (не включая состояние S_t), в которых атомарное высказывание φ обращается в истину.
4. Формула $G_e^{\ N} \varphi$ означает: начиная с некоторого состояния S_t, в N состояниях, идущих подряд, формула φ обращается в истину.
5. Левая часть $([E_e^{\ N}][p^I])$ оператора U_e означает, что существует N путей, ведущих в состояние S_t, таких что на каждом пути, в "I" состояниях, идущих подряд, но не включающих состояние S_t, будет истинна формула φ.
6. Правая часть $([E_e^{\ M}][p^J])$ оператора U_e означает, что существует M путей, ведущих из состояния S_t, таких что в каждом пути, начиная с состояния S_t, в "J" состояниях, идущих подряд, будет истинна формула φ.

Формальная грамматика, формальная семантика модифицированной CTL логики (CTL_e).

Формальную грамматику CTL_e формул запишем, используя расширенную форму Бэкуса-Наура:

$$\varphi ::= p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid E_e \chi U_e E_e \chi \mid E_e X\varphi \mid E_e F_e \varphi \mid E_e G_e \varphi;$$

$$\chi ::= p \mid \neg\chi \mid \chi \wedge \chi \mid \chi \vee \chi,$$

где p и χ – это атомарные высказывания, φ – это формула модифицированной CTL_e логики.

Формальную семантику модифицированной CTL_e логики, зададим относительно модели Крипке.

Истинность формулы φ из логики CTL_e в состоянии s структуры Крипке M определяется по индукции по структуре формулы φ:

1. $s \models p \equiv p \in L(s)$;
2. $s \models p \equiv s \not\models \varphi$;
3. $s \models \varphi_1 \wedge \varphi_2 \equiv s \models \varphi_1 \wedge\ s \models \varphi_2$;
4. $s \models \varphi_1 \vee \varphi_2 \equiv s \models \varphi_1 \vee\ s \models \varphi_2$;
5. $s \models E_{ei=1}^{N} X\varphi \equiv (\exists S_i, i = \overline{1,N} : (\exists s \to s_i) \wedge (\forall i, i = \overline{1,N} \Rightarrow s_i \models \varphi))$;

6. s $\models E_{ei=1}^{N} F_{ej=1}^{M} \varphi \equiv \exists$ N путей, таких что на каждом пути $\exists$ M состояний, в которых истинна формула φ;
7. s $\models E_{e}^{N} \chi_{l}^{M} U_{e} E_{e}^{K} \chi_{r}^{L} \rightarrow \equiv \exists$ N-путей, ведущих в состояние s: в каждом пути $\exists$ M состояний идущих подряд, не включая состояние s, в которых будет истинно атомарное высказывание χ_l и существует K путей, таких что в каждом из K путей, начинающихся из состояния s, будет идти подряд L состояний, начиная с состояния s, в которых будет истинна формула φ.

Алгоритм проверки оператора U_e на структуре Крипке.

Совершенно очевидно, что после расширения оператора U(until time operator)из CTL логики, к расширенному оператору U_e не применим существующий алгоритм проверки CTL-формул. Поэтому возникает необходимость разработки отдельного алгоритма для проверки оператораU_e на структуре Крипке (рис.1 и 2).

Алгоритм проверки формулы $E_{e}^{N} X\varphi$ на структуре Крипке.

Шаги алгоритма:

1. находим все состояния, в которые есть непосредственный переход из состояния S_0 и в которых выполняется высказывание φ.
2. Если количество найденных состояний меньше N, то $S_0 \not\models E_{e}^{N} X\varphi$. Иначе $S_0 \models E_{e}^{N} X\varphi$.

Алгоритм проверки формулы $E_{e}^{N} G_{e}^{M} \varphi$ на структуре Крипке.

Шаги алгоритма:

A.While (i<=M) do…..

1. j = 0. // флаг того, что $S_0 \models E_{e}^{N} G_{e}^{M} \varphi$.
2. Если ($S_0 \models \varphi$) $\wedge$ (N=1) $\wedge$ (M=1) то выходим из алгоритма с результатом $S_0 \models E_{e}^{N} G_{e}^{M} \varphi$

   j=1
3. i:=1
4. Если i = 1, то находим все состояния, в которые есть непосредственный переход из состояния S_0 и в которых в истину обращается высказывание φ. Количество таких состояний будет N_i
5. Находим все те состояния, в которые есть переход из состояний, найденных на шаге i-1 и в которых в истину обращается атомарное высказывание φ. Количество новых состояний будет N_i
6. Если N_i = 0, то выходим из алгоритма с результатом $S_0 \not\models E_{e}^{N} G_{e}^{M} \varphi$
7. Если i=1 и N_i меньше общего количества состояний, в которые есть прямой переход из состояния S_0, то выходим из алгоритма с результатом $S_0 \not\models E_{e}^{N} G_{e}^{M} \varphi$.

8. Если N_i меньше общего количества тех состояний, в которые есть переход из состояний, найденных на шаге i-1, то выходим из алгоритма с результатом $S_0 \not\models E_e^N G_e^M \varphi$.
9. Проверяем, если (i<=N) $\wedge$ (i=M), то выходим из алгоритма с результатом $S_0 \models E_e^N G_e^M \varphi$

j=1
10. i :=i+1.
od

B. Если j=0, то завершаем алгоритм с результатом $S_0 \not\models E_e^N G_e^M \varphi$

Алгоритм проверки формулы $E_e^N F_e^M \varphi$ на структуре Крипке.

Шаги алгоритма:
A. While (i<=M)
do

1. j := 0. // флаг того, что $S_0 \models E_e^N F_e^M \varphi$
2. k:=0 // флаг для подсчета количества состояний, в которых вып. высказывание φ
3. i:=1
4. Если i=1, то ищем все состояния, в которые есть переход из состояния S_0 и в которых обращается в истину атомарное высказывание φ. Количество таких состояний будет N_i
5. Находим все те состояния, в которые есть переход из состояний, найденных на шаге i-1 и в которых в истину обращается атомарное высказывание φ. Количество новых состояний будет N_i
6. Если N_i>0, то k:=k+1
7. Если (N_i <=N) $\wedge$ (k=M), то выходим из алгоритма с результатом $S_0 \models E_e^N F_e^M \varphi$
   j =0
8. i:=i+1

od

B. Если все состояния системы перебраны, при этом j=0, то завершаем алгоритм с результатом $S_0 \not\models E_e^N F_e^M \varphi$

Пример

Рассмотрим такое требование к системе: если пин-код банковской карточки ввели неправильно десять раза, то карточка блокируется.

Пусть γ – правильный ввод пин кода, α – неправильный ввод пин кода, β – блокировка карточки. Тогда спецификация к системе, записанная с использованием расширенного темпорального оператора U_e будет выглядеть так: $(E\alpha^{10})U_e(A\beta)$

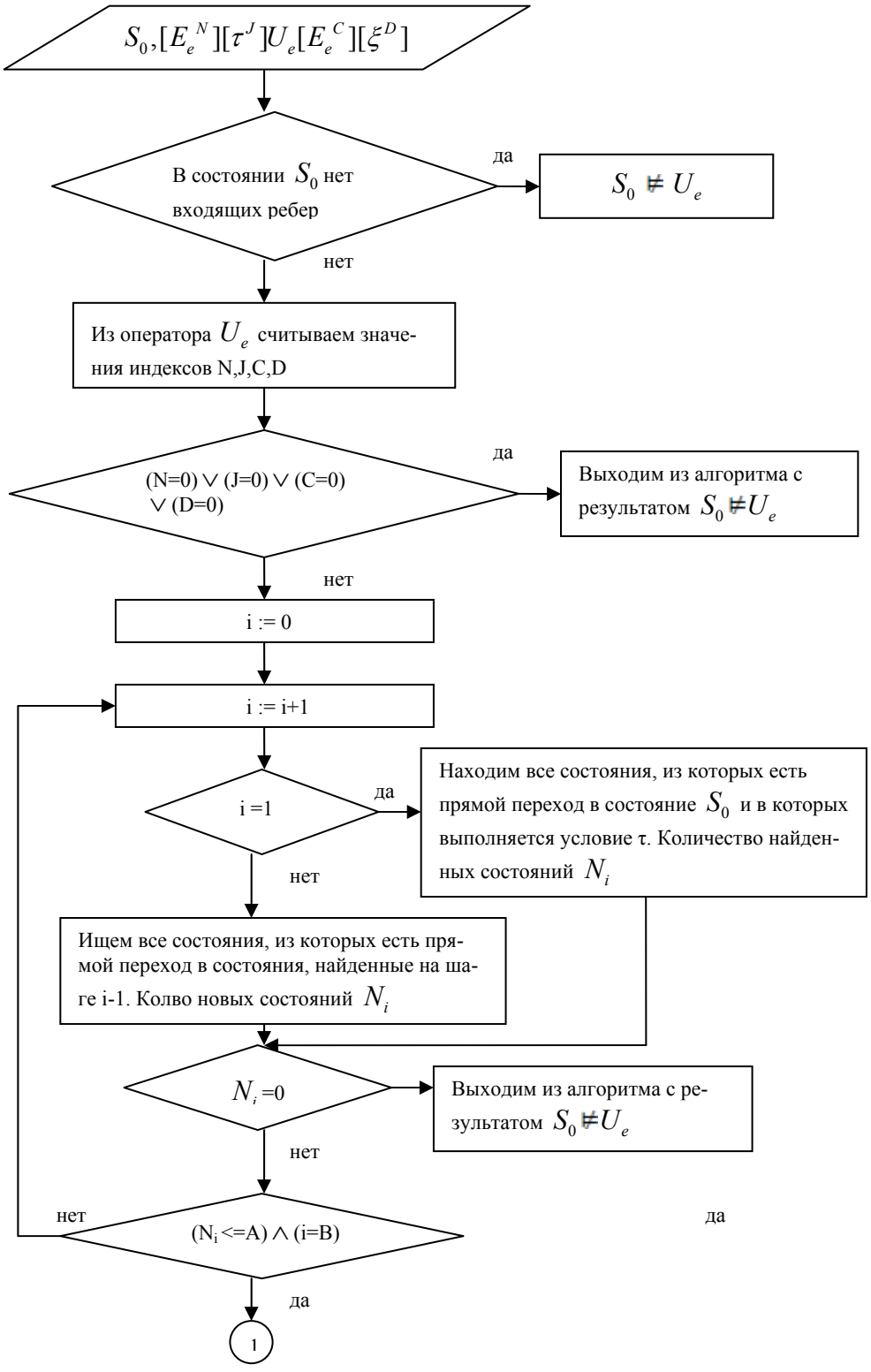

Рис. 1. Алгоритм проверки оператора U_e на структуре Крипке.

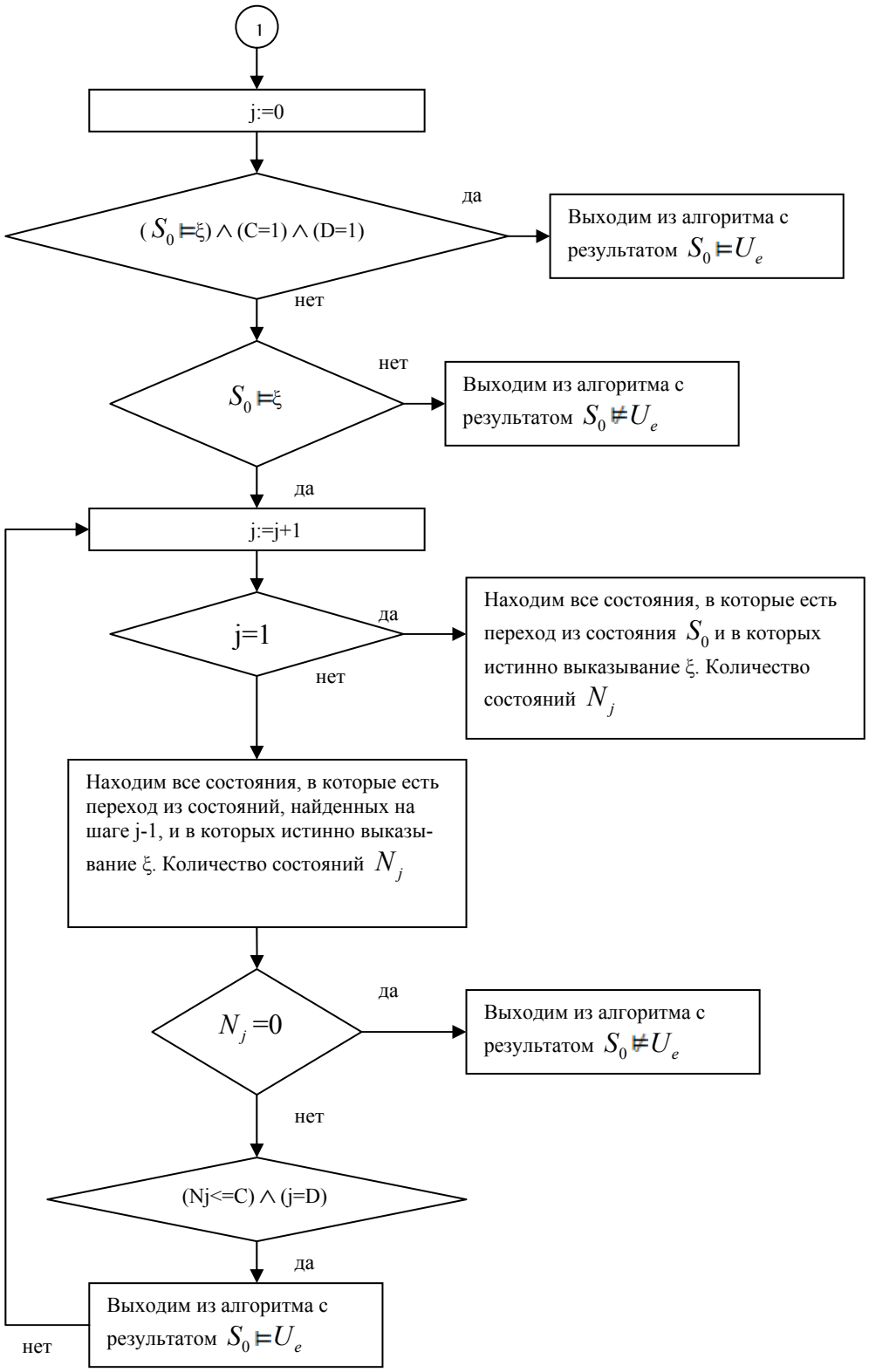

Рис. 2. Продолжение алгоритма проверки оператора U_e на структуре Крипке

Выводы

Расширение таких темпоральных операторов, как U,F,G, а так же квантора пути E, позволяет в спецификациях систем явно указывать на количество. Что в свою очередь облегчает анализ спецификаций.

ЛИТЕРАТУРА

1. Pnueli, A.. The temporal logic of programs [Text]. //Proc. of 18-th IEEE Symposium on Foundations of Computer Science, pp. 46-57, 1977.
2. Clarke, J.E.M., Emerson E.A. Design and synthesis of synchronization skeletons using branching time temporal logic [Text]. //A Series of Lecture Notes in Computer Science. – V. 131, pp.52-71, Springer, 1981.
3. Кларк, Э., Грамберг, О., Пелед, Д. Верификация моделей программ: Model Checking [Текст]. – М.: МЦНМО, 2002, с. 56-61.
4. Карпов, Ю.Г. Model Checking. Верификация параллельных и распределенных программных систем[Текст]. – СПб.: БХВ-Петербург, 2010, с. 41-73.

UPDATING OF CTL OPERATORS AND THEIR CHECK ALGORITHMS ON KRIPKE STRUCTURE

Y. Mazurov

Moscow State Social University
h.4, b.1, Vilgelma Pika st., Moscow, 105005, Russia

Abstract. In this article, extension of some temporal CTL logics operators is proposed, which enables explicit indication of the number of system states or the number of ways wherein the preset feature of the system is realized. Semantics, formal grammar, as well as relevant Kripke structure and extended temporal operators based feasibility check algorithms are given.

Key words: verification, Model Checking, CTL logics, Kripke structure.