АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

УДК 681.142.2

Ю. О. Герман, О. В. Герман
Белорусский государственный технологический университет

О НЕСООТВЕТСТВИИ МЕЖДУ РАСПОЗНАВАНИЕМ ЯЗЫКА ЗА ПОЛИНОМИАЛЬНОЕ ВРЕМЯ И ТОТАЛЬНОЙ ПОЛИНОМИАЛЬНОСТЬЮ РАСПОЗНАВАТЕЛЯ

Приводимые в статье рассуждения устанавливают разницу между тотальной полиномиальностью распознающей машины и полиномиальной сложностью реализуемого ею алгоритма. Проблема связана с отождествлением выводов и доказательств, что не всегда допустимо. Кроме того, нужно иметь в виду, что доказательство становится некорректным, если оно доказывает предикат, являющийся характеристической функцией изменяющегося в процессе доказательства множества. Таким образом, множество доказательств, вообще говоря, не следует отождествлять с множеством корректно построенных выводов. Применительно к теории вычислительной сложности заметим, что трудности доказательства P ≠ NP могут быть напрямую связаны с рассматриваемыми в данной статье проблемами теории доказательств. Возникает идея сведения языка, распознаваемого некоторой детерминированной машиной M за полиномиальное время, при условии, что для M нет доказательства тотальной полиномиальности, к языку Выполнимость. Эта идея, по сути, является наиболее существенной точкой роста для последующих работ. Другой важной точкой роста является развитие концепции доказательства как формального математического объекта.

Ключевые слова: алгоритм, система доказательств, проблема выполнимости, полиномиальная вычислительная сложность, финитность распознавателя.

Yu. O. German, O. V. German
Belarusian State Technological University

ABOUT DISCREPANCY BETWEEN POLYNOMIAL TIME LANGUAGE RECOGNITION ALGORITHM AND TOTAL POLYNOMIALITY OF THE RECOGNIZER

The reasoning given in the paper establishes distinction between total polynomiality of the recognizing machine and polynomial complexity of the algorithm realized by it. The problem stems from the incorrect identifying the proofs as inferences what may not take place. Besides, one should bear in mind the proof becoming incorrect if it proves the predicate representing a characteristic function of some set which changes due to the given proof. Thus, the set of proofs in general cannot be identified with a set of all correctly built inferences. Concerning the computation complexity theory, let us note that the obstacles in proving P ≠ NP may be directly connected to the difficulties in the proof theory considered in the paper. The idea arises to reduce a language recognized for polynomial time by some deterministic machine M, provided that there is no proof of M total polynomiality, to Satisfiability language. This idea, by essence, is the most important “point of growth” for future research alongside with the construction of a new concept of the proof as a formal mathematical object.

Key words: algorithm, proof system, satisfiability problem, polynomial computational complexity, recognition procedure finiteness.

Введение. Принимается, что доказательства алгоритмически распознаваемы. Доказательства используют некоторые множества объектов, которые должны оставаться неизменными. Поэтому если доказательство само использует некоторое множество доказательств, которому оно a priori не принадлежит, то оно не должно попадать в это множество a posteriori. Исходя из этой позиции, мы ставим цель продемонстрировать, что распознавание языка за полиномиальное время и тотальная полиномиальность распознавателя не всегда тождественны.

Основная часть. Рассмотрим язык LAR* [1], содержащий слова y, такие что

$$y = y_1 @ y_2.$$

i. y_1 представляет набор правил (спецификацию) произвольной машины Тьюринга MT_ε;

ii. y_2 является доказательством формулы, утверждающей, что MT_ε со спецификацией y_1 финитна на каждом входе x;

iii. MT_ε отклоняет слово $y = y_1 @ y_2$.

Символ-демаркатор @ имеет чисто техническое значение и может быть опущен.

Замечание. Доказательства y_2 представляют корректные выводы в некоторой метатеории *Ps* с точно определенными понятиями машины Тьюринга, финитности, отклоняющего (принимающего) вычисления и др. Теорию *Ps* можно противопоставить объектной теории *Th*, в которой определен универсум рассуждений относительно MT_ε. В *Ps* с четко определенной системой аксиом *Ax* и алгоритмически проверяемым синтаксисом доказательств имеем: доказательство произвольной формулы ε есть последовательность [2]:

$$R_1(\phi_1^1,\ \phi_2^1,\ ...,\ \phi_{k-1}^1 \mid \phi_k^1),$$
$$R_2(\phi_{k+1}^2,\ \phi_{k+2}^2,...,\ \phi_{l-1}^2 \mid \phi_l^2),\ ...,$$
$$R_v(\phi_w^v,\ \phi_{w+1}^v,\ ...,\ \phi_z^v \mid \varepsilon),$$

где R_i правила вывода с посылками, указанными слева от вертикальной черты в скобках и заключением, указанным справа. Посылки являются либо аксиомами *Ax*, либо заключениями ранее использованных правил в последовательности вывода. По тексту доказательства всегда можно установить, какая формула доказана и корректен ли вывод.

Можно построить машину MT_{LAR^*} (обозначим ее спецификацию $y_1^{LAR^*}$), распознающую LAR*. Принимаем практически очевидные факты о наличии финитных процедур для проверки пунктов i, ii, iii определения языка LAR*. Заметим, что верификация п. iii заключается в моделировании работы MT_ε на данном входном слове. Процесс моделирования финитен в силу доказательства y_2. Итак, MT_{LAR^*} распознает для каждого входа x, имеет ли место $x \in$ LAR* или $x \notin$ LAR* за финитное время.

***Теорема* 1.** Нельзя доказать финитность машины MT_{LAR^*}.

Доказательство. Пусть такое доказательство есть. Обозначим его D_{LAR}. Рассмотрим слово $y_1^{LAR^*}$@D_{LAR}. Тогда MT_{LAR^*} не может за финитное время ни отклонить его, ни принять, что противоречит D_{LAR}.

***Теорема* 2.** MT_{LAR^*} распознает язык LAR* за финитное время.

Доказательство. Ясно, что LAR* не пуст: он включает хотя бы одно слово для машины, отклоняющей все слова за один такт. По сути, машина MT_{LAR^*} должна проверить текст доказательства y_2. Доказательство состоит из однотипных блоков, каждый блок содержит посылки применяемого правила вывода, само правило вывода и заключение правила вывода. Проверка посылок сводится к выяснению, что это либо аксиома, либо полученное ранее заключение. Таким образом, проверка доказательств сводится к повторению однотипного действия – выяснению того, что некоторое слово содержится в другом слове. Если доказательство состоит из m слов, то проверок будет не более m^2. Проверка вхождения слова в другое слово выполняется на основе полиномиального детерминированного алгоритма за время порядка N, где N – наибольшая длина слова. Отсюда общее время проверки доказательства может быть оценено как $O(m^2 \cdot N)$ или $O(L^3)$, L есть длина доказательства. Процесс моделирования завершается за гарантированно финитное время в силу предъявленного доказательства. Поэтому наличие доказательства y_2 гарантирует распознавание результата работы машины y_1 на входе $y = y_1 @ y_2$. Однако финитность процедуры распознавания не дает оснований считать распознающую машину доказательно финитной в силу теоремы 1.

Рассмотрим несколько измененный язык PLAR, состоящий из слов $y = y_1y_2$, где:

1) y_1 представляет спецификацию произвольной детерминированной машины Тьюринга MT_η;

2) y_2 есть доказательство того, что MT_η, определенное словом y_1, отклоняет любое слово y, начинающееся с y_1 (т. е. $y = y_1Z$ с произвольным словом Z), за время, ограниченное некоторым фиксированным полиномом от длины слова $y = y_1Z$.

Ясно, что PLAR не пуст: он включает хотя бы одно слово для машины, отклоняющей все слова за один такт. Построим машину MT_{PLAR} с полиномиальным временем работы [3] для распознавания PLAR. По сути, эта машина должна проверить доказательство y_2. Доказательства распознаются за полиномиальное время. Моделировать теперь не надо. Следовательно, если доказательство проверено и оно верно, то машина MT_{PLAR} принимает слово $y = y_1 @ y_2$. Мы получаем таким образом следующий результат.

***Теорема* 3.** MT_{PLAR} распознает язык PLAR за полиномиальное время.

***Теорема* 4.** Нельзя доказать полиномиальную сложность машины MT_{PLAR}.

Доказательство. Оно состоит из двух частей. Пусть y_1^{LAR} есть спецификация MT_{PLAR}.

1. Доказываем, что MT_{PLAR} не принимает ни одного слова вида $y_1^{LAR}Z$. Если допустить противное (слово $y_1^{LAR}Z$ принимается), то в силу Z машина MT_{PLAR} отклоняет $y_1^{LAR}Z$, что дает противоречие. Заметим, что «не принимает» не означает отклоняет (допускается работа без останова). Поэтому данная часть доказательства не входит в множество доказательств того, что машина Тьюринга отклоняет все слова, начинающиеся с ее спецификации, за полиномиальное время. Это обеспечивает корректность данного доказательства.

2. Пусть имеется доказательство D_{polLAR} тотальной полиномиальности MT_{PLAR}. Тогда п. 1 означает, что MT_{PLAR} отклоняет каждое слово вида $y_1^{LAR}Z$ за полиномиальное время (работа без останова или с неполиномиальным временем теперь отпадают).

Пункты 1, 2 дают доказательство ϑ того, что каждое слово $y_1^{LAR}Z$ отклоняется за полиномиальное время. Поэтому MT_{PLAR} должна принять слово $y_1^{LAR}\vartheta$, что опять невозможно. Следовательно, допущение о наличии D_{polLAR} ложно.

Итак, мы столкнулись со следующей проблемой: нельзя доказать полиномиальность (финитность) некоторой машины Тьюринга как таковой, но можно доказать полиномиальность (финитность) реализуемого ею алгоритма. Для уяснения сути мы сошлемся на следующий наглядный парадокс. *Пловцом считается тот, кто умеет перебраться через реку. Пловцы получают сертификат пловца. Считается, что пловец, не прибегающий к услугам других, может переправить на себе только одного другого человека. Некто X переплавляется через реку только на сертифицированных пловцах. Причем ни один сертифицированный пловец не может отказать X. Следует ли этому X вручить сертификат?*

Сертификат пловца означает способность перебраться с одного берега на другой (неважно, каким способом). Следовательно, формально X и каждый такой X должен быть сертифицирован. Однако если такой сертифицированный пловец, теперь уже с удостоверением пловца, встретится другому такому же пловцу X, то оба пойдут ко дну либо их переправа вовсе не состоится. Здесь причиной казуса является нефиксированность ситуации. Пусть ситуация с сертифицированными пловцами зафиксирована, и среди них нет ни одного X. Тогда X не может стать сертифицированным пловцом, ибо множество сертифицированных пловцов зафиксировано, а если его изменить, то это уже будет другая ситуация. Первым сертифицированным пловцом как раз является не X. Таким образом, некое множество сертифицированных пловцов не содержит ни одного X. Еще раз: проблема в том, что X получает сертификат за счет некоего фиксированного множества сертифицированных пловцов, как только X получит сертификат, это множество сертифицированных пловцов будет уже иным (ситуация изменится).

Формальные выкладки. Теперь мы формально докажем, что машина MT_{LAR*} распознает язык LAR* за финитное время. Введем следующие предикаты:

prffin(z, x) – z есть доказательство финитности машины со спецификацией x;

fin(x) – утверждает, что машина Тьюринга со спецификацией x тотально финитна;

accept(x, y) – машина Тьюринга со спецификацией x принимает слово y;

decline(x, y) – машина Тьюринга со спецификацией x отклоняет слово y;

$\forall y$ *accept*(x, y) $\vee$ decline(x, y) – утверждает, что машина x распознает слова y за финитное время;

m – спецификация машины MT_{LAR*};

$f_1(w)$ – представляет рекурсивную функцию, которая возвращает левую часть от символа @ входного слова w и $f_2(w)$, соответственно, возвращает правую от символа @ часть слова (если нечего возвращать, то возвращается пустая строка).

Рассматриваем аксиомы *Ps*, для MT_{LAR*}:

1. $\forall x \forall y\ accept(x, y) \rightarrow \neg decline(x, y)$;

2. $\forall w\ \neg prffin(f_2(w), f_1(w)) \rightarrow$

$\rightarrow decline(m, w)$;

3. $\forall w\ prffin(f_2(w), f_1(w))\ \&$

$decline(f_1(w), w) \rightarrow accept(m, w)$;

4. $\forall w\ prffin(f_2(w), f_1(w))\ \&$

$accept(f_1(w), w) \rightarrow decline(m, w)$;

5. $\forall w\ accept(m, w) \rightarrow$

$\rightarrow prffin(f_2(w), f_1(w))\ \&\ decline(f_1(w), w)$;

6. $\forall w\ prffin(f_2(w), f_1(w)) \rightarrow$

$\rightarrow accept(f_1(w), w) \vee decline(f_1(w), w)$.

Требуется доказать:

7. $\forall y\ accept(m, y) \vee decline(m, y)$.

Доказательство. Основанный на резолюциях вывод состоит из следующих шагов. Записываем отрицание доказываемой формулы:

8. $\neg accept(m, c)$ /c – некоторая константа/;

9. $\neg decline(m, c)$.

Из 2, 9:

10. $prffin(f_2(c), f_1(c))$.

Из 3, 8:

11. $\neg decline(f_1(c), c) \vee \neg prffin(f_2(c), f_1(c))$.

Из 4, 9:

12. $\neg accept(f_1(c), c) \vee \neg prffin(f_2(c), f_1(c))$.

Из 10, 11, 12:

13. $\neg decline(f_1(c), c)$.

14. $\neg accept(f_1(c), c)$.

Из 6, 13, 14:

15. $\neg prffin(f_2(c), f_1(c))$.

10, 15 дают □ (противоречие).

Совместность аксиом устанавливается без труда. Приведенное доказательство устанавливает финитность процедуры распознавания, реализуемой машиной MT_{LAR*}. Докажем теперь, что для MT_{LAR*} нет доказательства финитности:

1. $\forall x \forall y\ accept(x, y) \rightarrow \neg decline(x, y)$;

2. $\forall w\ \neg prffin(f_2(w), f_1(w)) \rightarrow decline(m, w)$;

3. $\forall w\ prffin(f_2(w), f_1(w)) \&$

$decline(f_1(w), w) \rightarrow accept(m, w)$;

4. $\forall w\ prffin(f_2(w), f_1(w)) \&$

$accept(f_1(w), w) \rightarrow decline(m, w)$;

5. $\forall w\ accept(m, w) \rightarrow prffin(f_2(w), f_1(w)) \&$

$decline(f_1(w), w)$;

6. $\forall w\ prffin(f_2(w), f_1(w)) \rightarrow$

$\rightarrow accept(f_1(w), w) \vee$

$decline(f_1(w), w)$.

Требуется доказать

7. $\forall y\ \neg prffin(m, y)$.

Пусть вопреки утверждению теоремы есть такое доказательство. Обозначим его d. Тогда истинен предикат

8. $prffin(m, d)$.

Обозначим слово $a = m@d$. Тогда в силу наших обозначений $f_1(a) = m$, $f_2(a) = d$. Резольвентой дизъюнктов 6 и 8 является дизъюнкт

9. $accept(m, a) \vee decline(m, a)$.

Резольвентой 4, 8 является дизъюнкт

10. $\overline{accept}(m, a) \vee decline(m, a)$.

Резольвентой 9, 10 является дизъюнкт

11. $decline(m, a)$.

Из 1, 11 получим

12. $\overline{accept}(m, a)$.

Из 3, 8 получим

13. $accept(m, a) \vee \overline{decline}(m, a)$.

Противоречие дают 11, 12, 13.

Таким образом следует признать, что в общем случае не имеет места импликация

$\forall y\ accept(m, y) \vee decline(m, y) \rightarrow fin(m)$,

где $fin(m)$ утверждает финитность машины m, так как имеет значение смысл слова y.

Теоремы 3 и 4 доказываются аналогично. Ниже под $declinep(x, y)$ понимаем предикат, утверждающий, что машина x отклоняет слово y за время, ограниченное некоторым фиксированным полиномом; $prffinp(y_2, y_1)$ утверждает, что y_2 есть доказательство того, что машина y_1 отклоняет любое слово y_1Z за время, ограниченное некоторым фиксированным полиномом; $struct(x, y_1, y_2)$ означает, что слово x есть результат конкатенации слов y_1 и y_2.

Имеем систему аксиом:

1. $\forall x \forall y\ accept(x, y) \rightarrow \neg decline(x, y)$;

2. $\forall x \forall y_1 \forall y_2\ (struct(x, y_1, y_2)\ \&$
$prffinp\ (y_2, y_1) \rightarrow accept(m, x))$;

3. $\forall x \forall y_1 \forall y_2\ ((\neg struct(x, y_1, y_2) \vee$
$\neg prffinp\ (y_2, y_1)) \rightarrow decline(m, x))$;

4. $\forall w\ accept(m, w) \rightarrow prffinp(f_2(w), f_1(w))$;

5. $\forall x \forall y_1 \forall y_2\ (struct(x, y_1, y_2)\ \&$
$prffinp\ (y_2, y_1)) \rightarrow decline\ (y_1, x))$.

Теорема о том, что машина MT_{PLAR} со спецификацией y_1^{LAR} не принимает ни одного слова вида $y_1^{LAR}Z$:

6. $\forall x \forall z\ struct(x, m, z) \rightarrow \overline{accept}(m, x)$.

Отрицание теоремы дает два дизъюнкта:

7. $accept(m, c)$;

8. $struct(c, m, f_2(c))$.

Обозначим слово $c = m@d$. Тогда в силу наших обозначений $f_1(c) = m$, $f_2(c) = d$.

Из 1 и 7:

9. $\overline{decline}(m, c)$.

Из 5, 8, 9:

10. $\overline{prffinp}(f_2(c), m)$.

Из 4 и 10:

11. $\overline{accept}(m, c)$.

7 и 11 дают противоречие.

С учетом доказанной теоремы убедиться в том, что нет доказательства тотальной полиномиальной сложности машины MT_{PLAR}, не составляет труда. Допустим противное и расширим систему аксиом:

1. $\forall x \forall y\ accept(x, y) \rightarrow \neg decline(x, y)$;

2. $\forall w\ \neg prffinp(f_2(w), f_1(w)) \rightarrow$
$\rightarrow decline(m, w)$;

3. $\forall w\ prffinp(f_2(w), f_1(w)) \rightarrow$
$\rightarrow accept(m, w)$;

4. $\forall w\ accept(m, w) \rightarrow prffinp(f_2(w), f_1(w))$;

5. $\forall w\ accept(m, w) \rightarrow decline(f_1(w), w)$;

6. $pf(m)$ // доказательство полиномиальности MT_{PLAR};

7. $\forall w\ pf(m) \rightarrow accept(m, w) \vee decline(m, w)$;

8. $\forall w\ pf(m)\ \&\ \overline{accept}(m, w) \rightarrow$
$\rightarrow prffinp(d, m)$.

Убеждаемся, что эта система противоречива. Обозначим слово $a = m@d$. В силу наших обозначений $f_1(a) = m$, $f_2(a) = d$.

Из 6, 7 получим

9. $accept(m, a) \vee decline(m, a)$.

Из 5, 9 получим

10. $decline(m, a)$.

Резольвентой 1, 10 является дизъюнкт

11. $\overline{accept}(m, a)$.

Из 6, 8, 11:

12. $prffinp(d, m)$.

Из 3, 11:

13. $\overline{prffinp}(d, m)$.

12 и 13 дают противоречие.

Вместе с тем без $pf(m)$ последняя система аксиом непротиворечива (можно убедиться, построив выполняющую интерпретацию).

Заключение. Основной вывод статьи такой: машина Тьюринга может реализовать некий алгоритм за полиномиальное время и в то же время не быть тотально полиномиальной. Этот вывод позволяет по-новому взглянуть на проблему P ≠ NP и, возможно, открывает дорогу новым подходам к ее решению.

Литература

1. Герман О. В. Обучение основной теореме теории вычислительной сложности // Новые информационные технологии в образовании: материалы II Междунар. конф., Минск, 12–14 нояб. 1996 г.: в 2 т. / Объед. ин-т проблем информатики Нац. акад. наук Респ. Беларусь. Минск, 1996. Т. 1. С. 99–109.

2. Расева Е., Сикорски Р. Математика метаматематики. М.: Наука, 1972. 590 с.

3. Гэри М., Джонсон Д. С. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 410 с.

References

1. German O. V. Learning the main theorem in computational complexity theory. *Materialy II Mezhdynarodnoy konferentsii "Novye informatsionnye tekhnologii v obrazovanii: v 2 tomakh* [Materials of the 2nd International Conference "New information technologies in education": in 2 vol.]. Minsk, 1996, vol. 1, pp. 99–109 (In Russian).

2. Raseva E., Sikorski R. *Matematika metamatematiki* [Mathematics of Metamathematics]. Moscow, Nauka Publ., 1972. 590 p.

3. Geri M., Johnson D. S. *Vychislitel'nye mashiny i trudnoreshaemye zadachi* [Computers and intractable Problems]. Moscow, Mir Publ., 1982. 410 p.

Информация об авторах

Герман Юлия Олеговна – старший преподаватель кафедры информационных систем и технологий. Белорусский государственный технологический университет (220006, г. Минск, ул. Свердлова, 13а, Республика Беларусь). E-mail: juliagerman@tut.by

Герман Олег Витольдович – кандидат технических наук, доцент кафедры информационных систем и технологий. Белорусский государственный технологический университет (220006, г. Минск, ул. Свердлова, 13а, Республика Беларусь). E-mail: ovgerman@tut.by

Information about the authors

German Yulia Olegovna – senior lecturer, the Department of the Information Systems and Technologies. Belarusian State Technological University (13a, Sverdlova str., 220006, Minsk, Republic of Belarus). E-mail: juliagerman@tut.by

German Oleg Vitol'dovich – PhD (Engineering), Assistant Professor, the Department of Information Systems and Technologies. Belarusian State Technological University (13a, Sverdlova str., 220006, Minsk, Republic of Belarus). E-mail: ovgerman@tut.by

Поступила 15.05.2018