9. Прессли Э., Сигал Г. Группы петель / пер. с англ. А. В. Зелевинского, А. О. Радула. М.: Мир, 1990. 456 с. ISBN 5-03-001331-8.

10. Пугачев В. С., Синицын И. Н. Теория стохастических систем. М.: Логос, 2000. 1000 с.

ЯНИШЕВСКАЯ Анна Генриховна, доктор технических наук, профессор (Россия), профессор кафедры «Инженерная геометрия и САПР» Омского государственного технического университета.
Адрес для переписки: anna-yanish@mail.ru
СКОРОБОГАТОВ Роман Юрьевич, аспирант, старший преподаватель кафедры «Системы автоматизированного проектирования» Сибирского государственного университета телекоммуникаций и информатики (СибГУТИ).
Адрес для переписки: ro-m-a-n@yandex.ru
МИХАЙЛОВ Сергей Борисович, старший преподаватель кафедры «Высшая математика» СибГУТИ.
Адрес для переписки: sergmikh4@gmail.com
СЕДИНИН Валерий Иванович, доктор технических наук, профессор (Россия), заведующий кафедрой «Системы автоматизированного проектирования» СибГУТИ.
Адрес для переписки: sedvi@bk.ru

Для цитирования

Янишевская А. Г., Скоробогатов Р. Ю., Михайлов С. Б., Сединин В. И. Линейность математической модели механического движения антропоморфного робота // Омский научный вестник. 2018. № 1(157). С. 83—87. DOI: 10.25206/1813-8225-2018-157-83-87.

Статья поступила в редакцию 07.12.2017 г.

УДК 004:[004.02+004.588] 377
DOI: 10.25206/1813-8225-2018-157-87-91

А. Е. УЛЬТАН
Н. В. АБРАМЧЕНКО
Н. А. МЕЩЕРЯКОВА
Е. А. МЕЩЕРЯКОВ

Финансовый университет при Правительстве Российской Федерации, г. Омск

АЛГОРИТМ НАХОЖДЕНИЯ НЕРАВЕНСТВ-СЛЕДСТВИЙ

Цель работы состоит в разработке алгоритма, позволяющего в коллекции неравенств отыскивать неравенства-следствия. Такой алгоритм нужен для программы искусственного интеллекта, позволяющей, с одной стороны, самостоятельно решать школьные уравнения, неравенства и системы, а с другой — проверять правильность самостоятельного решения школьником этих задач. Все это поспособствует компьютеризации школьного образования.

Ключевые слова: компьютеризация образования, применение искусственного интеллекта в обучении, алгоритмы решения уравнений и неравенств, моделирование системы неравенств, нахождение неравенств-следствий.

В последнее время при программировании все чаще приходится сталкиваться с задачами искусственного интеллекта [1—7]. С этим мы сталкиваемся при создании компьютерной обучающей системы «Элементарная алгебра», которая должна не только сама решать школьные задачи по алгебре, но и проверять правильность решения ученика [8—10].

При создании такой системы необходимо помнить, что при решении алгебраических систем типа «И» основным является утверждение, представленное на рис. 1. Смысл этого утверждения в том, что изображенные на рисунке системы типа «И» эквивалентны (имеют одно и то же множество решений). Это означает, что для получения решения системы типа «И» можно поступать двояко:

1) можно решить A, решить B, решить C и полученное пересечь;

2) если C следует из A, B, то C можно просто удалить не решая и в дальнейшем решать и пересекать только A и B. Этот вариант значительно проще.

Как видим, если мы хотим, чтобы компьютерная программа не только сама решала системы типа «И», но и смогла проверить любое решение школьника и объяснить ему все способы решения, то необходимо иметь алгоритмы, позволяющие находить следствия в системах типа «И».

В частности, необходимо иметь алгоритм, находящий среди неравенств неравенства-следствия. Этому алгоритму и посвящена данная работа.

Точная постановка задачи следующая: дана коллекция неравенств, например, изображенная на рис. 2. За буквами, обозначающими правые и левые части неравенств, скрываются числа (возможно, в виде числовых выражений), имена величин или алгебраические выражения.

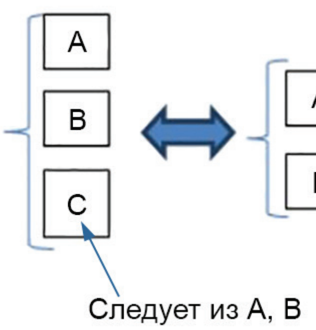

Рис. 1. Эквивалентные И-системы

$$a < d,$$

$$b < d,$$

$$c < e,$$

$$d < g,$$

$$d < f,$$

$$e < g,$$

$$a < g.$$

Рис. 2. Коллекция неравенств

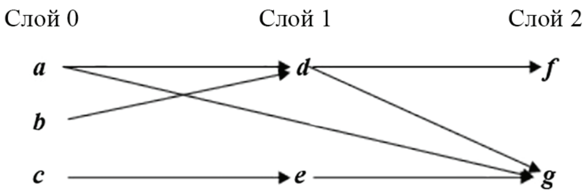

Рис. 3. Граф, изображающий систему неравенств

Найти неравенства-следствия.

В данном примере таким является неравенство $a < g$. Оно следует из неравенств $a < d$ и $d < g$. Все неравенства одного типа не случайны. Если появится неравенство другого типа, то мы его приведем к рассматриваемому типу. Как видим, одна и та же величина может несколько раз встречаться как в левой, так и в правой части неравенств.

Ясно, что данную систему неравенств можно представить в виде графа, изображенного на рис. 3.

Стрелка между величинами означает знак « < » или « ≤ ». В первый слой мы поместили величины, для которых не указаны меньшие предшественники, а в каждый последующий слой мы поместили величины, имеющие предшественников **обязательно в предыдущем слое и, возможно, в других предшествующих слоях. Тогда для каждого элемента номер слоя, в который его необходимо поместить, совпадает с длиной самого длинного пути из 0-слоя в рассматриваемый элемент** (длина пути измеряется количеством входящих в него стрелок). **Послойный граф, удовлетворяющий данному требованию, будем называть устроенным правильно.**

Величины, соединенные путями из стрелок, будут сравнимыми **в рамках нашей системы неравенств,** а те, для которых таких путей нет, будут **в рамках нашей системы неравенств** несравнимыми. Глядя на такой граф, легко увидеть существующие в рамках нашей системы неравенств неравенства-следствия. Например, стрелка, ведущая из a в g даст неравенство-следствие, **т. к. из a в g есть более длинный путь** ($a < d$ и $d < g$).

Однако если за некоторыми нашими буквами скрываются числа (возможно, в виде сравнимых числовых выражений), то их можно сравнивать **вне рамок нашей системы неравенств** и учитывать эти сравнения при построении (рис. 3). Это приведет к тому, что **в каждом слое, помимо несравнимых величин, сможет находиться лишь одно число (или сравнимое числовое выражение). При этом большее число будет находиться в слое с большим номером.**

Итак (рис. 3), визуализирующий систему неравенств, можно делать либо только **в рамках заданной системы неравенств,** и тогда даже числа могут получиться несравнимыми, либо **учитывая возможную дополнительную сравнимость элементов.** Исходя из рис. 3 можно делать разные выводы об изображенных на нем величинах и соотношениях между ними.

Из изложенного выше следует следующий алгоритм решения нашей проблемы.

Алгоритм создания правильного графа, моделирующего систему неравенств, учитывающий не только заданные неравенства, но и возможную дополнительную (не заданную) сравнимость элементов неравенств. Смысл алгоритма в организации пошагового процесса. **Будем полагать, что неравенства поступают и обрабатываются по одному. При этом смысл обработки каждого нового неравенства состоит в том, чтобы на его основе внести изменения в построенный ранее правильный граф, аналогичный графу, изображенному на рис. 3, получая при этом опять правильный граф.** Первоначально этот граф состоит только из пустого 0-слоя.

Для простоты изложения элементы неравенств (*м*, *б*), обладающие возможной дополнительной (не заданной) сравнимостью, будем называть числами (числа такой возможностью всегда обладают), а не обладающие — будем называть не числами. Это не совсем так, т. к. мы знаем, что, например, выражения тоже могут обладать возможной дополнительной сравнимостью. Например, среднее арифметическое $a > 0$ и $b > 0$ всегда будет больше, либо равно их среднему геометрическому, и это необходимо учитывать, даже если это явно не задано.

Опишем действия по обработке очередного неравенства $м < б$ или $м \le б$.

Сначала ищем в уже построенном графе *м* и *б*. Дальнейшее поведение зависит от результатов поиска.

Если *б* нашли в *пб*-слое и *м* нашли в *пм*-слое, то:
если *пм* < *пб*, то:
если *м* и *б* еще не соединены, то:
соединяем *б* и *м*
иначе (*м* и *б* уже соединены)
ничего не делаем

если *пм* ≥ *пб*, то:
если *м* и *б* еще не соединены, то:
переносим последовательно *б* и затем послойно всех его последователей вправо за *пм*-слой.
иначе (*м* и *б* соединены и значит уже есть утверждение, что *б* < *м* или *б* ≤ *м*):
если знак добавляемого неравенства строгий (*м* < *б*), то:
объявляем всю систему неравенств несовместной (т. к. *б* < *м* или *б* ≤ *м* и *м* < *б*).
если же неравенство нестрогое (*м* ≤ *б*), то:
если путь из *б* в *м* состоит только из нестрогих неравенств, то:
признаем *б* = *м*
переносим последовательно *б* и затем послойно всех его последователей вправо за *пм*-слой.
иначе:
объявляем всю систему неравенств несовместной.

Если *б* не нашли и *м* не нашли, то:

если *м* не число, то:
помещаем *м* (с пустой коллекцией цепочек-предшественников) в 0-слой.
пм приравниваем к 0
если же *м* — число, то:
если ближайшее меньшее к *м* число отсутствует, то:
помещаем *м* (с пустой коллекцией цепочек-предшественников) в 0-слой
пм приравниваем к 0
иначе (ближайшее меньшее к *м* число присутствует)
помещаем *м* (с пустой коллекцией цепочек-предшественников) в слой, добавляемый за слоем, содержащим ближайшее меньшее число
пм приравниваем к номеру добавленного слоя

рэ (рассматриваемый элемент) = *б*
рэ помещаем в 0-слой
п = *пм*

если рэ не число, то:
переносим рэ в слой, следующий за *п*-слоем (уже существующий или добавляемый нами в противном случае)
соединяем рэ и *м*

если рэ — число, то:
находим *п1* — номер слоя, содержащего ближайшее меньшее к рэ число, или 0, если такое отсутствует
находим *п2*= max{*п1, п*}
если слой, следующий за *п2*-слоем, существует и не содержит число, то:
помещаем рэ в слой, следующий за *п2*-слоем
если рэ = *б* и *п2* = *п*, то
соединяем рэ и *м*
иначе
соединяем рэ и число из предыдущего слоя
иначе (если слой, следующий за *п2*-слоем, не существует или содержит число)
помещаем рэ в слой, добавленный за *п2*-слоем
если рэ = *б* и слой, следующий за *п2*-слоем, не существует, то
соединяем рэ и *м*
иначе:
соединяем рэ и число из предыдущего слоя

Если *б* нашли в *пб*-слое, а *м* не нашли, то:

если *м* не число, то:
помещаем *м* (с пустой коллекцией цепочек-предшественников) в 0-слой.
пм приравниваем к 0
если же *м* — число, то:
если ближайшее меньшее к *м* число отсутствует, то:
помещаем *м* (с пустой коллекцией цепочек-предшественников) в 0-слой
пм приравниваем к 0
иначе (ближайшее меньшее к *м* число присутствует)
помещаем *м* (с пустой коллекцией цепочек-предшественников) в слой, добавляемый за слоем, содержащим ближайшее меньшее число
пм приравниваем к номеру добавленного слоя

если *пм* < *пб*, то:
если *м* и *б* еще не соединены, то:
соединяем *б* и *м*
иначе (*м* и *б* уже соединены)
ничего не делаем

если *пм* ≥ *пб*, то:
если *м* и *б* еще не соединены, то:
переносим последовательно *б* и затем послойно всех его последователей вправо за *пм*-слой.
иначе (*м* и *б* соединены и значит уже есть утверждение, что *б* < *м* или *б* ≤ *м*):
если знак добавляемого неравенства строгий (*м* < *б*), то:
объявляем всю систему неравенств несовместной (т. к. *б* < *м* или *б* ≤ *м* и *м* < *б*).
если же неравенство нестрогое (*м* ≤ *б*), то:
если путь из *б* в *м* состоит только из нестрогих неравенств, то:
признаем *б* = *м*
переносим последовательно *б* и затем послойно всех его последователей вправо за *пм*-слой.
иначе:
объявляем всю систему неравенств несовместной.

Если *б* не нашли, а *м* нашли в *пм*-слое, то:

рэ (рассматриваемый элемент) = *б*
п = *пм*

если рэ не число, то:
переносим рэ в слой, следующий за *п*-слоем (уже существующий или добавляемый нами в противном случае)
соединяем рэ и *м*

если рэ — число, то:
находим *п1* — номер слоя содержащего ближайшее меньшее к рэ число или 0, если такое отсутствует
находим *п2* = max{*п1, п*}
если слой, следующий за *п2*-слоем, существует и не содержит число, то:
помещаем рэ в слой, следующий за *п2*-слоем
если рэ = *б* и *п2* = *п*, то
соединяем рэ и *м*
иначе
соединяем рэ и число из предыдущего слоя
иначе (если слой, следующий за *п2*-слоем, не существует или содержит число)
помещаем рэ в слой, добавленный за *п2*-слоем
если рэ = *б* и слой, следующий за *п2*-слоем, не существует, то

соединяем рэ и *м*

иначе

соединяем рэ и число из предыдущего слоя

Теперь опишем процедуру

переносим последовательно рэ (рассматриваемый элемент) и затем послойно всех его последователей вправо за *n*-слой:

если рэ признан равным, то:

рэ переносим в *n*-слой

всех предшественников и последователей *м* добавляем к предшественникам и последователям рэ

иначе

если рэ не число, то:

переносим рэ в слой, следующий за *n*-слоем (уже существующий или добавляемый нами в противном случае)

соединяем рэ и *м*

если рэ — число, то:

находим *n1* —номер слоя, содержащего ближайшее меньшее к рэ число или 0, если такое отсутствует

находим $n2=\max\{n1, n\}$

если слой, следующий за *n2*-слоем, существует и не содержит число, то:

помещаем рэ в слой, следующий за *n2*-слоем

если рэ $=$ *б* и $n2 = n$, то

соединяем рэ и *м*

иначе

соединяем рэ и число из предыдущего слоя

иначе (если слой, следующий за *n2*-слоем, не существует или содержит число)

помещаем рэ в слой, добавленный за *n2*-слоем

если рэ $=$ *б* и слой, следующий за *n2*-слоем, не существует, то

соединяем рэ и *м*

иначе

соединяем рэ и число из предыдущего слоя

рэ помещаем в коллекцию «Перемещенные»

nm (номер текущего слоя) $= n + 1$

пока $nm \le$ количество слоев -1 (т. к. нумерация слоев начинается с 0)

для каждого рэт (рассматриваемого элемента текущего слоя) будем делать следующее:

если рэт отсутствует в коллекции «Перемещенные», то

$k = \max$ номер слоя предшественников рэт присутствующих в коллекции «Перемещенные» или -1, если такие предшественники отсутствуют

если $k \neq -1$, то

если рэ не число, то:

переносим рэ в слой, следующий за *n*-слоем (уже существующий или добавляемый нами в противном случае)

соединяем рэ и *м*

если рэ — число, то:

находим *n1* — номер слоя содержащего ближайшее меньшее к рэ число или 0, если такое отсутствует

находим $n2=\max\{n1, n\}$

если слой, следующий за *n2*-слоем, существует и не содержит число, то:

помещаем рэ в слой, следующий за *n2*-слоем

если рэ $=$ *б* и $n2 = n$, то

соединяем рэ и *м*

иначе

соединяем рэ и число из предыдущего слоя

иначе (если слой, следующий за *n2*-слоем, не существует или содержит число)

помещаем рэ в слой, добавленный за *n2*-слоем

если рэ $=$ *б* и слой, следующий за *n2*-слоем, не существует, то

соединяем рэ и *м*

иначе

соединяем рэ и число из предыдущего слоя

рэт помещаем в коллекцию «Перемещенные»

$nт = nт + 1$

Слова «соединяем *эл*1 и *эл*2» означают, что в коллекцию последователей *эл*2 добавляется ссылка на *эл*1, а в коллекцию предшественников *эл*1 добавляются цепочки из коллекции предшественников *эл*2, пополненные справа именем *эл*2. Если при этом образуется цепочка, состоящая только из имени *эл*2, то при наличии других более длинных цепочек неравенство *эл*2 < *эл*1 будет следствием и его необходимо поместить в коллекцию следствий.

Обоснованием этого алгоритма является тот факт, что для каждого элемента номер слоя, в который его необходимо поместить, совпадает с длиной самого длинного пути из 0-слоя в рассматриваемый элемент.

Таким образом, построенный алгоритм позволяет в системе неравенств найти неравнества-следствия, что позволит правильно решать уравнения, неравенства и системы.

Библиографический список

1. Адаменко А. Н., Кучуков А. М. Логическое программирование и Visual Prolog. СПб.: БХВ-Петербург, 2003. 990 с. ISBN 5-94157-156-9.
2. Корнеев В. В., Гараев А. Ф., Васютин С. В. [и др.]. Базы данных. Интеллектуальная обработка информации. М.: Нолидж, 2000. 352 с. ISBN 5-89251-089-1.
3. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. СПб.: Питер, 2000. 384 с. ISBN 5-272-00071-4.
4. Нильсон Н. Принципы искусственного интеллекта / пер. с англ. Р. М. Абдусаматова; под ред. В. Л. Стефанюка. М.: Радио и связь, 1985. 373 с.
5. Lenat D. B. EURISKO: A Program That Learns New Heuristics and Domain Concepts: The Nature of Heurictics III: Program Design and Results // Artificial Intelligence. 1983. Vol. 21, Issue 1-2. P. 61–98. DOI: 10.1016/S0004-3702(83)80005-8.
6. Newell A. The Knowledge Level // Artificial Intelligence. 1982. Vol. 18, Issue 1. P. 87–127. DOI: 10.1016/0004-3702(82)90012-1.
7. Schach S. R. Software Engineering. 2nd ed. CRC Press Publ., 1993. 400 p. ISBN: 0256129983; 978-0256129984.
8. Ультан А. Е., Закандырин В. А. Разработка обучающей программы «Решение текстовых задач по алгебре» // Омский научный вестник. 2009. № 6 (82). С. 200–201.
9. Ультан А. Е., Петров Е. С. Разработка архитектуры комплекса обучающих программ // Омский научный вестник. 2010. № 6 (92). С. 190–193.
10. Ультан А. Е., Кравцов Д. А. Разработка обучающей информационной системы «Алгебра» // Омский научный вестник. 2011. № 3 (98). С. 169–173.

УЛЬТАН Александр Ефимович, кандидат технических наук, доцент кафедры «Высшая математика и информатика».
Адрес для переписки: ultan_ae@mail.ru

АБРАМЧЕНКО Нина Владимировна, кандидат педагогических наук, доцент кафедры «Высшая математика и информатика».
Адрес для переписки: abram4enko44@gmail.ru
МЕЩЕРЯКОВА Наталия Ананьевна, кандидат педагогических наук, доцент кафедры «Высшая математика и информатика».
Адрес для переписки: mna1961@gmail.ru
МЕЩЕРЯКОВ Евгений Александрович, кандидат физико-математических наук, доцент кафедры «Высшая математика и информатика».
Адрес для переписки: mechtch@mail.ru

Для цитирования

Ультан А. Е., Абрамченко Н. В., Мещерякова Н. А., Мещеряков Е. А. Алгоритм нахождения неравенств-следствий // Омский научный вестник. 2018. № 1 (157). С. 87–91. DOI: 10.25206/1813-8225-2018-157-87-91.

Статья поступила в редакцию 27.12.2017 г.

УДК 519.7
DOI: 10.25206/1813-8225-2018-157-91-94

А. Н. ФЛОРЕНСОВ

Омский государственный технический институт, г. Омск

ПРИМЕНЕНИЕ КРИТЕРИЯ ТЬЮРИНГА ДЛЯ ИССЛЕДОВАНИЯ ДИНАМИКИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Классический тест Тьюринга для определения искусственного интеллекта можно переформулировать с учетом текущей реальности. Вместо его применения к изолированному компьютеру предложено рассматривать глобальные поисковые системы из Интернета. Результат теста оказывается зависимым не только от двух объектов оценивания потенциального интеллекта, но и от интеллекта оценивающего субъекта. Для более строгого описания процедуры Тьюринга использовано понятие меры разума. Модифицированный тест, названный критерием, позволяет определять и оценивать текущую динамику искусственного интеллекта Интернета и сравнивать его с интеллектуальным уровнем человека, действующего в качестве опорного субъекта. В качестве вывода из формализмов теста показывается, что современная система Интернета обладает искусственным интеллектом при использовании в качестве опорного узла субъекта с современным образованием.

Ключевые слова: разум, искусственный интеллект, тест Тьюринга, взаимодействие, информация, компьютер, мера разума.

Введение. Научное понятие искусственного интеллекта было предложено английским математиком А. Тьюрингом. С учетом очевидной расплывчатости общечеловеческих представлений о разуме и интеллекте человека, Тьюринг ввел формальную систему определения наличия интеллекта у субъекта, с которым происходит общение [1]. Эта система больше известна под названием теста Тьюринга. Содержание теста состоит в сравнении текстовых взаимодействий человека с двумя субъектами, один из которых техническое устройство, второй — обычный человек. Сравнивающий человек не знает заранее, какой из этих субъектов общения настоящий человек, а какой — искусственное устройство. Критерий признания интеллекта за искусственным субъектом общения заключается в том, что при длительном таком двустороннем общении проверяющий человек не может определить, какая сторона является техническим субъектом, а какая — натуральным. Частным случаем такого заключения является вариант приписывания человеческого разума не той стороне, где действительно находится он, а искусственной системе. Иначе говоря, в ходе такого взаимодействия — общения человек ошибается и приписывает недостаточные проявления разума на самом деле человеку, а техническую сторону начинает считать разумной.

В своей классической форме тест Тьюринга описывался как взаимодействие с одним компьютером и одним человеком. Нетрудно видеть, что технический прогресс сделал взаимодействие человека с одним компьютером почти анахронизмом.

Действительно, современный пользователь компьютера в подавляющем большинстве случаев информационно взаимодействует не только с программным обеспечением одного компьютера, а посредством его с мировой информационной сетью Интернет. Отклонение от такой ситуации наблюдаются только для специальных компьютерных систем ограниченного пользования. Поэтому в определении теста Тьюринга в его научном применении следует изменить термин компьютер на более общее понятие технической системы, что и указано выше.