

-
13. Savchenko T. V., Kalenkova O. N. i dr. Kul'turno-jazykovaja adaptatsija detej migrantov sredstvami obrazovaniya: Metodika obucheniya russkomu jazyku // Etnojekspress. 2005.
14. Usha T. Ju. Natsional'naja rossijskaja shkola — polietnicheskaja i polikul'turnaja // Izvestija Ross. gos. ped. universiteta im. A. I. Gertsena. 2012. № 148. S. 125–133.
15. Usha T. Ju. Russkij jazyk v polijetnicheskoj shkole // Nachal'naja shkola. 2007. № 12.
16. Usha T. Ju. Russkij jazyk v shkole kak metodicheskaja problema // Vestnik Baltijskogo federal'nogo universiteta im. I. Kanta. 2013. № 13. S. 28–33.
17. Usha T. Ju. Testirovanie v polikul'turnoj shkole kak sredstvo jazykovej identifikatsii inofonov // izvestija vysshih uchebnyh zavedenij. Povolzhskij region: Gumanitarnye nauki. 2013. № 4 (28). S. 224–232.
18. Usha T. Ju. Jazykovaja identifikatsija uchashchih-sja-inofonov // Vestnik BFU im. I. Kanta. Ser.: Pedagogicheskie nauki. Vyp. 2. Kaliningrad: Izd-vo BFU im. I. Kanta, 2012. S. 100–105.
19. Killough E. <http://vk.com/id39866416>. Obrashchenie: 10.05.14.
20. <http://www.bildungserver.de/> Obrashchenie: 10.05.14.
21. <http://www.kindersache.de/> Obrashchenie: 10.05.14.
22. <http://www.kmk.org/> Obrashchenie: 10.05.14.
23. <http://www.3sat.de> Obrashchenie: 10.05.14.

A. G. Rakitin

ИНТЕРПРЕТАТОР ДНК-ВЫЧИСЛЕНИЙ НА ФУНКЦИОНАЛЬНОМ ЯЗЫКЕ ПРОГРАММИРОВАНИЯ

В связи с возрастающим интересом к применению молекулярных вычислений для решения различных типов задач появляется необходимость в создании интерпретаторов неклассических вычислительных моделей, которые опираются на процессы, происходящие в природе. В будущем молекулярные вычисления могут не только повлиять на развитие современной науки и образования, но и конкурировать с существующими в настоящее время классическими вычислительными моделями.

Ключевые слова: биокомпьютеры, язык программирования Haskell, ДНК-вычисления, операции над цепочками ДНК.

A. Rakitin

INTERPRETER OF DNA-COMPUTING IN A FUNCTIONAL PROGRAMMING LANGUAGE

In connection with the increasing interest in the application of molecular computing for solving various types of problems there is a need in creating interpreters of non-classical computational models which rely on the processes taking place in nature. In future molecular calculations can not only affect the development of modern science and education, but also compete with the existing currently classical computational models.

Keywords: Bio-computers, computer language Haskell, DNA-computing, operations over strands of DNA.

Вычисления на ДНК являются одним из возможных направлений развития биоинформатики. ДНК-вычисления принципно

отличаются от вычислений, основанных на принципах фон Неймана. За счет принципиально нового варианта структуры

хранения и обработки данных появляются некоторые особенности.

К особенностям ДНК-вычислений и ДНК-компьютеров можно отнести массовый параллелизм вычислений, возможность хранения значительно большего количества данных по сравнению с кремниевым компьютером при одинаковом объеме пространства, низкие затраты «энергии» для проведения вычислений. Перечисленные свойства позволяют рассматривать науки, изучающие ДНК-вычисления и биокомпьютеры, как перспективные области, которые смогут дать толчок для развития информационных технологий в будущем.

В настоящее время опубликованы опыты, которые были осуществлены с цепочками ДНК в лабораторных условиях [3; 4; 5]. В таких опытах исходные данные и результат были закодированы последовательностями нитей ДНК.

К таким опытам можно отнести:

(1) опыт Эдельмана, позволивший решить задачу коммивояжера (о поиске пути в графе);

(2) опыт Шапиро, позволивший создать конечный автомат;

(3) опыт Винфри, позволивший синтезировать различные поверхности при помощи ДНК.

Были опубликованы и различные теоретические алгоритмы, наиболее известный из них — алгоритм взлома криптосистемы DES, основанный на понятии «стикеры» [1].

В связи с возрастающими потребностями обучения студентов технических специальностей перспективным технологиям был создан интерпретатор на функциональном языке программирования Haskell, моделирующий операции на ДНК. В 2012 году данный интерпретатор успешно был применен в обучении новым вычислительным моделям студентов информационных технологий на базе кафедры ин-

формационных систем и программного обеспечения.

Обучение состояло из изучения студентами теоретических понятий и последующего выполнения заданий, связанных с операциями над цепочками ДНК. Студентам предлагалось выполнить операции без использования компьютера с последующей возможностью проверить результат с использованием интерпретатора на языке программирования Haskell.

Интерпретатор позволяет выполнить следующие команды [1]:

- Ввести пробирку с клавиатуры;
- Ввести случайную пробирку;
- Выполнить операцию «Слить»;
- Выполнить операцию «Размножить»;
- Выполнить операцию «Обнаружить»;
- Выполнить операцию «Разделить» («извлечь»);
- Выполнить операцию «Выделить по длине»;
- Выполнить операцию «Выделить по префиксу»;
- Выполнить операцию «Выделить по суффиксу»;
- Просмотреть содержимое всех пробирок;
- Получить справку по командам.

Кроме указанных операций есть возможность выполнять следующие команды для работы со стикерами [1]:

- Ввести пробирку с клавиатуры;
- Ввести случайную пробирку;
- Выполнить операцию «Слить»;
- Выполнить операцию «Разделить»;
- Выполнить операцию «Включить»;
- Выполнить операцию «Очистить»;
- Посмотреть содержимое всех пробирок;
- Получить справку по командам для работы со стикерами.

Команды в интерпретаторе вводятся с клавиатуры.

После запуска программы в интерпретаторе HUGS-98 необходимо вызывать главную функцию «main», ввод случайных пробирок 1 и 2 можно осуществить командами: «0 N1» и «0 N2». После номера команды «0», указывается специальный символ «N» и номер пробирки. Результат выполнения работы программы после ввода команд представлен на рисунке 1.

При работе со случайными пробирками на экране выводится информация о случайном числе, на основе которого сгенерирована пробирка. На следующей строке выводится последовательность символов, моделирующая цепочку ДНК.

Пробирки представлены массивом строк ДНК. Цепочки ДНК представлены элементами массива строк. Строки состоят из символов «А», «Т», «G», «С», кодирующих нуклеотиды аденин, тимин, гуанин и цитозин соответственно.

Для выполнения операции «Объединить» необходимо набрать номер команды

«1», затем номера пробирок через пробел вместе со спецсимволом «N». Получившиеся значения элементов массива можно увидеть, набрав команду «show». Результат выполнения команды «объединить» и команды «show» представлен на рисунке 2.

Аналогичные команды используются при работе со стикерами. Однако при вводе с клавиатуры для операций «разделить», «включить» и «очистить» необходимо указывать вместе с номерами пробирок и номер стикера, для которого осуществляется операция.

Результат выполнения следующих операций представлен на рисунке 3:

1. Ввести случайную ДНК в пробирку под номером один (длина стикера равна пяти, длина запоминающей цепочки равна пяти).
2. Показать содержимое всех пробирок.
3. Включить в первую пробирку второй стикер.
4. Показать содержимое всех пробирок.

```

Enter a command (help - h): 0 N1
Command: "0 N1", result:
"Random number: "
592148115
["TTTTCT", "TCCAAGAA", "CTGCTCTAGAG", "TTCATGCC", "AGAAAGCGTTTG"]
"The tube is entered with number:"
"1"
Enter a command (help - h): 0 N2
Command: "0 N2", result:
"Random number: "
-1366864315
["AT", "ACGGTGA", "AGGATCGCA", "CATAGGTTG", "GGG"]
"The tube is entered with number:"
"2"

```

Рис. 1. Ввод случайных пробирок с клавиатуры

```

Enter a command (help - h): 1 N1 N2
Command: "1 N1 N2", result:
"Multiply tubes"
Enter a command (help - h): show
Command: "show", result:
"-----"
["TTTTCT", "TCCAAGAA", "CTGCTCTAGAG", "TTCATGCC", "AGAAAGCGTTTG"], ["TTTTCT", "TCCAAGAA", "CTGCTCTAGAG", "TTCATGCC", "AGAAAGCGTTTG"], [], [], [], [], [], []
"Show tubes"

```

Рис. 2. Ввод команд «объединить» и команды «show»

```

Enter a command (help - h): 0 N1 5 5
Command: "0 N1 5 5", result:
"Random number: "
-1077238442
"k=5"
"m=5"
["ATTAT-CCATT-TTTCA-AGTCC-CCGCT"]
"The tube is entered with number: 1"
Enter a command (help - h): s
Command: "s", result:
"-----"
[["ATTAT-CCATT-TTTCA-AGTCC-CCGCT"],[" - - - - "] , [ ], [ ], [ ], [ ], [ ]
, [ ], [ ], [ ], [ ], [ ], [ ], [ ], [ ], [ ], [ ], [ ], [ ], [ ], [ ]
"Show tubes"
Enter a command (help - h): 3 N1 2
Command: "3 N1 2", result:
"Include j sticker in a tube"
Enter a command (help - h): s
Command: "s", result:
"-----"
[["ATTAT-CCATT-TTTCA-AGTCC-CCGCT"],[" -GGTAA- - - - "] , [ ], [ ], [ ], [ ], [ ]
, [ ], [ ], [ ], [ ], [ ], [ ], [ ], [ ], [ ], [ ], [ ], [ ], [ ]
"Show tubes"

```

Рис. 3. Результат выполнения операций со стикерами

Цепочки ДНК представлены строками, стикеры разделены между собой символом «←→». Если стикер включен, то на соответствующем месте в четной пробирке стоят символы, которые по смыслу будут компонентами соответствующей цепочки со стикером.

Разработанный интерпретатор позволяет наглядно продемонстрировать вы-

полнение операций над цепочками ДНК, а также выполнять операции над цепочками ДНК, моделирующими стикеры. В связи с этим рассматриваемый интерпретатор может являться одним из средств обучения неклассическим вычислительным моделям студентов технических специальностей.

СПИСОК ЛИТЕРАТУРЫ

1. Паун Г. ДНК-компьютер. Новая парадигма вычислений / Г. Паун, Г. Розенберг, А. Саломая. М.: Мир, 2004. 528 с.
2. Ракитин А. Г. Интерпретатор языка пробирок для ДНК-вычислений / А. Г. Ракитин, Т. С. Стефанова, М. В. Швецкий // Научное мнение. 2013. № 10. С. 214–219.
3. Adleman L. M. Computing with DNA, Scientific American, August 1998. P. 34–41.
4. Benenson Y., Paz-Elizur T., Adar R., Keinan E., Livneh Z. & Shapiro E. Programmable and autonomous computing machine made of biomolecules. Nature 414, 430–434 (2001).
5. Rothmund P. W. K., Papadakis K. & Winfree E. Algorithmic self-assembly of DNA Sierpinski triangles. PLoS Biol. 2, e424 (2004).

REFERENCES

1. Paun G. DNK-komp'yuter. Novaja paradigma vychislenij / G. Paun, G. Rozenberg, A. Salomaa. M.: Mir, 2004. 528 s.
2. Rakitin A. G. Interpretator jazyka probirok dlja DNK-vychislenij. / A. G. Rakitin, T. S. Stefanova, M. V. Shveckij // Nauchnoe mnenie. 2013. № 10. S. 214–219.
3. Adleman L. M. Computing with DNA, Scientific American, August 1998. P. 34–41.
4. Benenson Y., Paz-Elizur T., Adar R., Keinan E., Livneh Z. & Shapiro E. Programmable and autonomous computing machine made of biomolecules. Nature 414, 430–434 (2001).
5. Rothmund P. W. K., Papadakis K. & Winfree E. Algorithmic self-assembly of DNA Sierpinski triangles. PLoS Biol. 2, e424 (2004).