

УДК 519.688

РЕГУЛЯРНЫЕ ВЫРАЖЕНИЯ В РАСЧЕТАХ ЯДЕРНОЙ ТРАНСМУТАЦИИ

А. Р. Белозёрова

(ГНЦ НИИАР, г. Димитровград)

Математическое моделирование процессов ядерных превращений (трансмутаций) в веществе, происходящих под воздействием нейтронного облучения, позволяет изучить влияние облучения на ядерно-физические свойства и радиационные характеристики реакторных материалов. В качестве инструментального средства пользователя применяется Универсальная программа метролога (УПМ) под OS Windows 9x/2000/XP/NT.

Настоящая версия программы обеспечивает графический интерфейс для баз данных и имеет расширенное меню для расчетов ядерной трансмутации в конструкционных материалах, наведенной активности конструкционных материалов при реакторном облучении с использованием нейтронно-физических характеристик, химического состава материала и разветвленных схем нуклидных превращений, для восстановления нейтронного спектра по экспериментальным данным. В расчетах ядерной трансмутации и при обработке текстов с результатами расчетов широко используются регулярные выражения.

Ключевые слова: регулярные выражения, регулярный язык, ядерная трансмутация, формат ENDF/B-6, библиотека оцененных ядерных данных.

Введение

Процессы ядерных превращений (трансмутаций) в веществе, происходящие под воздействием нейтронного облучения, представляются особенно важными при выборе материалов, которые используют в качестве конструкционных для узлов и устройств ядерных реакторов различного типа. Ядерные трансмутации приводят к изменению химического и изотопного состава; происходит накопление радиоактивных и стабильных продуктов, содержание которых определяет сроки безопасной эксплуатации устройств, как радиационной, так и функциональной [1, 2], в ядерно-энергетических установках.

Моделирование процесса в рамках модели изонуклидной трансмутации позволяет прогнозировать динамику таких изменений [3]. Для непрерывного решения линейной системы обыкновенных дифференциальных уравнений ядерной трансмутации решается задача дискретной оптимизации схемы нуклидных превращений с применением эвристических алгоритмов из методов дискретной глобальной и локальной оптимизации с элементами математической теории принятия решений и теории искусственного интеллекта [4, 5].

В качестве инструментального средства пользователя под OS Windows 9x/2000/XP/NT разработана Универсальная программа метролога (УПМ). Настоящая версия УПМ предоставляет пользователю графический интерфейс для доступа к хранящимся в базе данных (БД) нейтронно-физическим характеристикам, данным по химическому составу материалов и схемам нуклидных превращений для расчетов эффектов ядерной трансмутации, повреждающей дозы в конструкционных материалах при реакторном облучении.

Нейтронно-физические данные в созданной для УПМ БД занимают 1,1 Мб. Они состоят из 162 записей по энергетическим спектрам нейтронов, измеренным или рассчитанным для различных точек

и конфигураций исследовательских реакторов. Данные по химическому составу конструкционных материалов сформированы на основе реально используемых в облучательных устройствах, конструкторских элементах исследовательских реакторов (чехлы экранных сборок, оболочки тепловыделяющих элементов и т. д.). Данные по разветвленным блочным схемам нуклидных превращений обновляются на основе проводимых расчетов ядерной трансмутации.

В расчетах ядерной трансмутации и при обработке текстов с результатами расчетов широко используются регулярные выражения. Регулярные выражения предоставляют мощный, гибкий и эффективный метод обработки текста. Широкие возможности сопоставления шаблонов, предоставляемые регулярными выражениями, позволяют быстро анализировать большие объемы текста, проверять текст на соответствие шаблонам (например форматам интегрированных библиотек оцененных ядерных данных по сечениям реакций взаимодействия нейтронов с ядрами атомов ADL-3 [6], FENDL-2.0 [7], ENDF/B-VII.0 [8] и JEFF-3.1.1 [9]¹), извлекать, изменять, заменять или удалять подстроки текста, а также добавлять извлеченные строки в коллекцию для формирования отчетов.

Цель статьи — показать эффективность применения современных технологий программирования, используемых, как правило, для коммерческих, экономических, обучающих программ, на примере регулярных выражений, реализованных в УПМ для расчета ядерной трансмутации.

Применение регулярных выражений в УПМ

Общие сведения об УПМ. УПМ — это приложение с графическим интерфейсом пользователя, предполагающее псевдореально-мультизадачный режим выполнения. Программа разработана по принципу модульного проектирования, содержит пакет утилит PREPRO 2007 [11] и ряд самостоятельных подпрограмм в виде dll-файлов динамической библиотеки и exe-файлов. Интерфейсная часть УПМ и алгоритмы расчетных процедур написаны на языке C++ .NET в среде разработки Microsoft Visual Studio.NET [12, 13]. Структура БД создана средствами управления БД Microsoft SQL Server 2000 [14]. Неоднократное тестирование используемого набора модулей подтвердило адекватность реализации каждого из них.

Пакет утилит PREPRO 2007 используется для предварительной обработки ядерных данных в формате ENDF/B. Это модульный набор компьютерных кодов, каждый из которых обеспечивает чтение и запись оцененных ядерных данных в формате ENDF. Каждый код исполняет одну или несколько независимых операций над данными. Предварительная обработка ENDF-форматированных данных необходима для последующего использования в приложениях. Утилиты пакета PREPRO 2007 являются независимыми и совместимыми с современными операционными системами любого типа компьютеров — от большого универсального до малых персональных компьютеров, таких как IBM-PC и Power MAC.

Общие сведения о регулярных выражениях [15]. Понятие регулярного языка играет важную роль в современной информатике, причем как в теоретических, так и в практических ее разделах. Регулярные языки являются важным классом формальных языков. Формальные языки классифицируются в соответствии с типами грамматик, которыми они задаются. Сложность языка определяется его типом. Наиболее сложные языки — с фразовой структурой (сюда можно отнести естественные языки), далее — контекстно-зависимые, контекстно-свободные языки и самые простые — регулярные языки.

Удобный и компактный способ конечного описания формального языка в виде регулярных выражений находит практическое применение во многих компьютерных приложениях, таких как текстовые редакторы, интерпретаторы командной строки и автоматические генераторы лексических анализаторов, информационно-поисковые системы.

Краткие сведения о формате ENDF/B-6. ENDF/B-6 является международным форматом для хранения оцененных ядерных данных (Приложение 1). Однажды подготовленные в форма-

¹Константное обеспечение комплекса состоит из указанных интегрированных библиотек, а также файла данных по распадам радионуклидов из библиотеки FENDL-2 [10].

те ENDF наборы оцененных данных могут преобразовываться в формы, пригодные для тестирования и непосредственных приложений с помощью обрабатывающих программ. Разработаны обрабатывающие программы, которые получают усредненные по группам сечения из библиотеки ENDF для использования в нейтронных расчетах. Эти программы выполняют такие функции, как восстановление детального хода сечений по параметрам резонансов, доплеровское уширение резонансных линий, усреднение по заданным энергетическим группам, и/или перевод в специальные интерфейсные форматы.

Описательная информация дается в виде серии карточных образов, каждая карта содержит в себе до 66 символов Холлерита [16].

Последняя оценка стандартов формата ENDF была выполнена в 2005 г.

Примеры регулярных выражений из УПМ

Подстроки. Для разработчиков приложений на C++ имеется возможность обращений к обработчику регулярных выражений, который содержится в платформе .NET Framework и представляется классом `System.Text.RegularExpressions.Regex` [17, 18]. Объект типа `Regex` инициализируется передачей строки в конструктор (см. ниже выделенную строку листинга). Тривиальным случаем регулярного выражения является строка, состоящая из одного символа алфавита регулярного языка, например, "=", "-", "\\n", "\\t"². Приведенный ниже пример из УПМ иллюстрирует процесс поиска подстроки "CROSS" в другой строке:

```
DirectoryInfo* di = new DirectoryInfo(Environment::CurrentDirectory);
DirectoryInfo* fi[] = di->GetDirectories();
String* l_strContents, *l_strSourceFileName;
Regex* myDirectory = new Regex(S"CROSS");
for(int i=0; i<fi->Length; i++)
{
    l_strContents=String::Copy(fi->GetValue(i)->ToString());
    if(myDirectory->IsMatch(l_strContents)) break;
}
```

С помощью метода `IsMatch()` определяется, обнаружено ли в указанной входной строке соответствие образцу.

Приведенный образец поиска относится к простым регулярным выражениям, так как это уровень строчных методов. В таблице приведены примеры других подстрок для организации поиска в строках. Во фрагменте листинга для поиска подстроки "\\n" встречается обращение к методу `Split()`. Данный метод разделяет входную строку в позициях, которые определяются сопоставлением регулярного выражения с образцом.

Обработка образцов строк "SKALE", "Chart", "8457" аналогична обработке "CROSS".

Примеры поиска подстрок

Подстроки	Фрагменты листинга
"\\n"	<code>Regex* myEqual = new Regex(S"\\n");</code> <code>String* sim[] = myEqual->Split(l_strContents);</code>
"->"	<code>Regex* FirstRegex = new Regex(S"->");</code> <code>if(FirstRegex->IsMatch(words)){...}</code>
"ppm"	<code>Regex* myOneRegex = new Regex(S"ppm");</code> <code>if(!myOneRegex->IsMatch(m_ptrLVMaterial->get_Items()->get_Item(MaterialIndex)->get_SubItems()->get_Item(j)->get_Text()->ToString())){...}</code>

²Соответствие escape-символов: \t — знак табуляции, \n — знак новой строки, \s — знак пробела, \d — любая десятичная цифра, \w — любой алфавитно-цифровой знак.

Классы символов. Класс символов определяет набор символов, хотя бы один из которых должен содержаться в строке ввода, чтобы поиск соответствия выполнялся успешно. Данный набор символов можно задать в виде группы в квадратных скобках, например [mM], [MNO]³, или/и как диапазон [*первый символ-последний символ*]⁴, например [0-9], [A-Z].

Классы символов состоят из языковых элементов. Класс символов соответствует какому-либо одному набору символов. Ниже приведены примеры регулярных выражений с классами символов:

```
"[\\b\\n\\t\\r]+";5
"\\d\\.\\w-+|[\\s\\t]{1}";
"\\w+|[\\d]+[mM]?[\\d]?";
"[0-9]{1}\\.[0-9]+[-+]{1}[0-9]+";
"=[\\s]+";
"$(<=\\b|v|=)\\d\\.+-\\w{8}[\\s\\t]{1}";
"(<=\\b|~|=)\\d\\.+-\\w{8}[\\s\\t]{1}$";
"[M,m]?"
"$[0-9][.]?[0-9]+[eE]?[-+]?[0-9]-$";
"\\s\\b\\t+";
"\\d\\.\\w-+";
```

Для проверки действительного числа с плавающей точкой в УПМ используется регулярное выражение вида "\$[0-9][.]?[0-9]+[eE]?[-+]?[0-9]-\$", где знак ?⁶ имеет смысл *0 или 1*, знак + (плюс) имеет смысл *один или несколько*, знак - (минус) имеет смысл *0 или более*.

Ограничим образец символами " и \$ (в начале и в конце образца). Эти специальные символы указывают, в какой части текста располагается образец. Символ " означает, что образец должен находиться в начале текста, а символ \$ — что образец должен быть в конце. Одновременное использование обоих символов исключает возможность наличия любого другого текста в проверяемой строке. Такой метод проверки очень полезен.

Теперь рассмотрим два регулярных выражения для проверки нуклидов:

$$" [\\w] + [-] { 1 } [\\d] + [mM] ? [\\d] ? " ; \tag{1}$$

$$" [0 - 9] { 1 , 3 } - [A - Z a - z \\s] { 2 } - [0 - 9 \\s] { 3 } [MNO] { 0 , 1 } " . \tag{2}$$

Здесь знак перехода \w указывает на любой из символов, составляющих слово, такой как буква, цифра или символ подчеркивания. Предполагается, что обозначение нуклида состоит из одного или двух символов, разделенных одним дефисом, после чего следуют символы десятичных цифр, за которыми может быть знак изомерности данного нуклида в виде символа m или M в случае (1) (соответствует формату ENDF/B-6, диалект INT для ADL-3) или символа M, N или O в случае (2) (соответствует формату ENDF/B-6 для остальных библиотек), а далее могут следовать символы десятичных цифр, которые указывают на номер уровня изомерности нуклида.

В соответствии с (1) корректными будут нуклиды He-4, Co-60, Co-60m, Zn-61m2, Zn-61m3. В соответствии с (2) корректными будут нуклиды 2-He-4, 27-Co-60, 27-Co-60M, 30-Zn-61N, 30-Zn-61O.

Конструкции группирования. Конструкции группирования с помощью регулярных выражений отображают части выражений и обычно *захватывают* части входной строки. Эти конструкции состоят из языковых элементов. Так, конструкция группирования (*часть выражения*)

³Группа символов может включать любую комбинацию одного или нескольких литеральных символов, escape-символов или классов символов.

⁴Диапазон символов — это непрерывная последовательность знаков, которая задается указанием первого и последнего символов последовательности и дефисом между ними. Два символа являются непрерывными, если они имеют соседние кодовые точки в Юникоде.

⁵Здесь и далее знаки ";" (точка с запятой) и "." (точка) после символа " к регулярным выражениям не относятся.

⁶Метасимвол (символ-джокер) — символ, который используется для замены других символов или их последовательностей, приводя, таким образом, к символьным шаблонам. Чаще всего употребляются два метасимвола: * для замены любой строки символов или пустой строки и ? для замены любого одиночного или отсутствующего символа.

захватывает соответствующую *часть выражения* — любой допустимый шаблон регулярного выражения. Используемые скобки нумеруются автоматически слева направо, основываясь на порядке открывающихся скобок в регулярном выражении, начиная с 1. *Захват* с номером 0 — это текст, соответствующий всему шаблону регулярного выражения.

Ранее уже была описана конструкция для проверки действительного числа с плавающей точкой. Представим второй вариант:

```
"([\d\.\w-]+\s\t{1})"
```

Данный образец регулярного выражения захватывает соответствующую часть выражения `[\d\.\w-]+\s\t{1}`, где класс символов `[\d\.\w-]` позволяет проверять одну или более стандартных десятичных цифр с возможным вхождением символа экспоненты и знака порядка числа, а класс символов `[\s\t]` позволяет проверять строго один символ пробела или табуляции. Здесь используется конструкция группирования, так как в обрабатываемом тексте каждая строка представлена двумя действительными значениями, между которыми встречается символ пробела либо символ табуляции.

Значения двумерного массива для описания нейтронного спектра обрабатываются и используются при составлении рабочих файлов пакета утилит PREPRO 2007 [11] для предварительной обработки ядерных данных в формате ENDF/B-6. Текстовый формат рабочего файла GROUPIE.LST представлен в виде фрагмента в Приложении 2.

Перечислим примеры конструкций группирования, реализованных в УПМ:

```
"([\b\n\t\r]+)";
"([\w]+[-]{1}[\d]+[mM]?[\d]?)";
"([\d\.\w-]+\s\t{1})";
"(\b)";
"(\n)";
"(\t)";
"()";
"([\d\.\w-]+)";
"(\d+)";
"([A-Z,a-z,A-Я,a-я,0-9,\-\,]+)";
"([\.\w+])";
"([\w+[-\.\.]+\d+)";
"([\w+[-\.\.]?[\d*\w?[-+]?[\d*])";
"([\w-]+)$";
"([\d]{3}-[\w]{2}-[\d]{3})"
```

Заключение

Некоторые регулярные выражения, используемые в УПМ, показывают возможности манипулирования произвольным текстовым форматом, предоставляемые инструментарием регулярных выражений. Регулярные выражения позволяют быстро анализировать текст и находить определенные шаблоны символов для извлечения, редактирования, замены или удаления текстовых подстрок, а также для добавления извлеченных строк в коллекцию при создании отчета. Эффективность применения регулярных выражений, в отличие от строковых типов, особенно наглядна в реализации для формата ENDF/B-6 оцененных ядерных данных при использовании интегрированных библиотек, таких как ADL-3, FENDL-2.0, ENDF/B-VII.0, JEFF-3.1.1.

Работа выполнена при финансовой поддержке РФФИ (грант 08-08-97000 р_поволжье_a).

Приложение 1. Файл данных по распадам радионуклидов decay-ENDF-VII0.endf

```

..... Первые 53 строки файла вырезаны.....
1.001000+3 9.991673-1 -1 0 0 1 2 1451 1
0.000000+0 0.000000+0 0 0 0 6 2 1451 2
0.000000+0 0.000000+0 0 0 4 7 2 1451 3
0.000000+0 0.000000+0 0 0 16 2 2 1451 4
1-H - 1 BNL EVAL-NOV05 A.A. Sonzogni (from NWC) 2 1451 5
/ENSDF/ DIST-DEC06 2 1451 6
—ENDF/B-VII MATERIAL 2 2 1451 7
—RADIOACTIVE DECAY DATA 2 1451 8
—ENDF-6 FORMAT 2 1451 9
***** Begin Description ***** 2 1451 10
** ENDF/B-VII RADIOACTIVE DECAY DATA FILE ** 2 1451 11
** Produced at the BNL from the Nuclear Wallet Cards database ** 2 1451 12
** Author: J.K. Tuli ** 2 1451 13
** Translated into ENDF format by A.A. Sonzogni, November 2005 ** 2 1451 14
***** 2 1451 15
Parent Excitation Energy: 0.0000 2 1451 16
Parent Spin & Parity: 1/2+ 2 1451 17
Parent half-life: STABLE 2 1451 18
Abundance: 99.985% 1 2 1451 19
***** End Description ***** 2 1451 20
1 451 22 0 2 1451 21
8 457 5 0 2 1451 22
0.000000+0 0.000000+0 0 0 0 0 2 1 099999
0.000000+0 0.000000+0 0 0 0 0 2 0 0 0
1.001000+3 9.991673-1 0 0 1 0 2 8457 1
0.000000+0 0.000000+0 0 0 6 0 2 8457 2
0.000000+0 0.000000+0 0.000000+0 0.000000+0 0.000000+0 0.000000+0 2 8457 3
5.000000-1 1.000000+0 0 0 6 0 2 8457 4
0.000000+0 0.000000+0 0.000000+0 0.000000+0 0.000000+0 0.000000+0 2 8457 5
0.000000+0 0.000000+0 0 0 0 0 2 8 099999
..... Остальные 450532 строки файла вырезаны.....

```

Приложение 2. Рабочий файл GROUPIE.LST

```

Multi-Band Library Identification
-----
Groupie Test Run
-----
MAT/MF/MT Ranges
-----
Minimum Maximum
MAT MF MT MAT MF MT
-----
1 1 1 9999 99 999
-----
Group Energy Boundaries
-----
Energy-eV Energy-eV Energy-eV Energy-eV Energy-eV Energy-eV

```

```

.100000000 .105000000 .110000000 .115000000 .120000000 .127500000
.135000000 .142500000 .150000000 .160000000 .170000000 .180000000
.190000000 .200000000 .210000000 .220000000 .230000000 .240000000
.255000000 .270000000 .280000000 .300000000 .320000000 .340000000
.360000000 .380000000 .400000000 .425000000 .450000000 .475000000
.500000000 .525000000 .550000000 .575000000 .600000000 .630000000
.660000000 .690000000 .720000000 .760000000 .800000000 .840000000
.880000000 .920000000 .960000000 1.00000000 1.05000000 1.10000000

```

..... Эта часть файла вырезана.....

```

15700000.0 15800000.0 15900000.0 16000000.0 16100000.0 16200000.0
16300000.0 16400000.0 16500000.0 16600000.0 16700000.0 16800000.0
16900000.0 17000000.0 17100000.0 17200000.0 17300000.0 17400000.0
17500000.0 17600000.0 17700000.0 17800000.0 17900000.0 18000000.0
18100000.0 18200000.0 18300000.0 18400000.0 18500000.0 18600000.0
18700000.0 18800000.0 18900000.0 19000000.0

```

ENDF/B Tape Label

Список литературы

1. Конструкционные материалы ядерных реакторов. Ч. 2: Структура, свойства, назначение / Под ред. Н. М. Бескоровайного. М.: Атомиздат, 1977.
2. *Solonin M. I., Chernov V. M., Gorokhov V. A. et al.* Present status and future prospect of the Russian program for fusion low-activation materials // J. Nucl. Mater. 2000. Vol. 283—287. P. 1468—1472.
3. *Markina N. V., Shimansky G. A.* TRANS_MU computer code for computation of transmutant formation kinetics in advanced structural materials for fusion reactors // Ibid. 1999. Vol. 271—272. P. 30—34.
4. *Белозёрова А. Р., Шиманский Г. А.* Оптимизация схемы расчета трансмутации методом ветвей и границ // Сб. рефератов семинара "Физическое моделирование изменения свойств реакторных материалов в номинальных и аварийных условиях". Димитровград, 5—6 апреля 2004 г. Димитровград: ФГУП "ГНЦ РФ НИИАР", 2005. С. 75—77.
5. *Белозёрова А. Р., Мельников Б. Ф.* Применение комплекса эвристик в задаче составления схемы нуклидных превращений // Сб. тр. Второй всерос. науч. конф. "Методы и средства обработки информации". Москва, 5—7 октября 2005 г. М.: МГУ им. М. В. Ломоносова, 2005. С. 208—214.
6. *Грудзевич О. Т., Зеленецкий А. В., Игнатюк А. В., Пащенко А. Б.* Библиотека ядерно-физических данных для расчетов активации и трансмутации // Атомная энергия. 1994. Т. 76. Вып. 2. С. 124—130.
7. FENDL-2, May 1998 Version. CD ROM Edition, Request 2378.01. International Atomic Energy Agency, Division of Physical and Chemical Sciences, Nuclear Data Section. Austria. <http://www-nds.iaea.org/fendl/index.html>.
8. Special issue on ENDF/B-VII.1 library // Nuclear Data Sheets. December 2011. Vol. 112, Issue 12. P. 2887—2996. <http://www.nndc.bnl.gov/exfor/endl00.jsp>.
9. *Sublet J-Ch., Koninkl A. J., Forrest R. A., Kopecky J.* The JEFF-3.0/A Neutron Activation File — EAF-2003 into ENDF-6 format // JEFDOC-982. Saint Paul Lez Durance, France. November 2003. P. 34. http://www.nea.fr/dbforms/data/eva/evatapes/jeff_31/.

10. *Firestone R. B.* Table of Isotopes. CD ROM Edition, Version 1.0, March 1996. Office of Energy Research, Office of High Energy and Nuclear Physics, Nuclear Physics Division of the US Department of Energy.
11. PREPRO 2007. ENDF/B Pre-processing Codes (ENDF/B-VII Tested) Owned, Maintained and Distributed by the Nuclear Data Section International Atomic Energy Agency. Vienna, Austria. <http://www-nds.iaea.org/ndspub/endf/prepro2007/>.
12. *Торстеинсон П., Оберг Р.* Архитектура .NET и программирование на Visual C++: Пер. с англ. М.: Вильямс, 2002.
13. *Саранг П. Г., Корера А., Фрейзер С. и др.* Visual C++.NET: Пособие для разработчиков C++: Пер. с англ. М.: ЛОРИ, 2003.
14. *Артёмов Д. В.* Microsoft SQL Server 2000. Новейшие технологии. М.: Русская редакция, 2001.
15. *Мельников Б. Ф.* Недетерминированные конечные автоматы. Тольятти: ТГУ, 2009.
16. *Кит С.* Рид-Грин. История переписи населения в США и обработки ее данных // В мире науки. 1989. № 4. С. 70—76.
17. Microsoft, 2012. <http://msdn.microsoft.com/ru-ru/library/system.text.regularexpressions.regex.aspx>.
18. *Бекер П.* Регулярные выражения / Журнал для профессионалов. Программирование на C/C++. ООО "Инфопресс", 2006. № 10 (34). С. 1—7.

Статья поступила в редакцию 22.07.11.
