

Подсистема автоматизации формализации НПА будет включать следующие компоненты:

- модуль ввода НПА, которая позволяет вводить в систему текст НПА, с указанием ссылки на него;

- модуль формализации НПА на основе технологий обработки естественных языков, который по введенному тексту НПА строит его формальную модель;

- модуль корректировки формальной модели НПА, который позволяет оператору проверить правильность этой модели и при необходимости внести в неё изменения;

- база знаний, построенная на основе формальных моделей, введенных в систему НПА и прошедших корректировку (этот компонент является общим для обеих подсистем);

Подсистема поддержки принятия решений состоит из следующих компонентов:

- база знаний, построенная на основе формальных моделей, введенных в систему НПА и прошедших корректировку (этот компонент является общим для обеих подсистем);

- модуль принятия заявлений, который на основе базы знаний и запроса оператора на добавление абитуриента формирует список документов, которые этот абитуриент должен предоставить с указанием ссылок на НПА и их части, определяющие этот список;

- база данных абитуриентов, которая хранит информацию об абитуриентах, поданных ими заявления и предоставленных документах, а также в этой базе фиксируется факт зачисления абитуриента;

- модуль зачисления, который автоматически формирует списки зачисленных на основе базы знаний и данных об абитуриентах;

- модуль ручной проверки зачисления, который позволяет просматривать сформированные предыдущим модулем списки зачисленных и при необходимости вносить в них изменения;

- модуль документального оформления принятых решений, который формирует необходимые документы для фиксации принятых приемной комиссией решений о зачислении абитуриентов.

Наличие последнего модуля обусловлено тем фактом, что за принятые решения ответственность несет комиссия, и, следовательно, недопустимо, чтобы ошибки программной системы, которые неизбежны, приводили к принятию ошибочных решений.

Возможный подход по интеграции описанной системы поддержки принятия решений с внешними системами может состоять в генерации модулей этих внешних систем, соответствующих модулям принятия заявлений, зачисления, его ручной проверки и документального оформления на основе сформированной базы знаний. Этот подход также предполагает совместное использование базы данных абитуриентов. Альтернативный подход может заключаться в организации взаимодействия указанных модулей и соответствующих им модулей внешней системы.

Заключение. В работе изложена структура системы поддержки принятия решений приемной комиссии учреждения высшего образования на основе формализованных нормативно-правовых актов, основные аспекты её функционирования и возможные подходы к её интеграции с другими системами.

1. Белов В. М. Что такое юрисметрия? // Тенденции формирования науки нового времени : сборник статей Международной научно-практической конференции (18 октября 2014 г., г. Уфа). – Уфа : РИО МЦИИ ОМЕГА САЙНС, 2014. – С. 122–124.
2. Курейчик, В. М. Курейчик Виктор Михайлович Особенности построения систем поддержки принятия решений // Известия ЮФУ. Технические науки. – 2012 – №7 – С. 92–98.

ОПИСАНИЕ РЕГУЛЯРНОГО ВЫРАЖЕНИЯ СТАТИЧЕСКОЙ ИНФОРМАЦИЕЙ О ТИПЕ НА ЯЗЫКЕ C++

*С.В. Сергеев
Витебск, ВГУ имени П.М. Машерова*

При выделении в тексте на искусственном языке элементарных лексических единиц (лексем) широкое распространение получили регулярные языки и описывающие их регулярные выражения [1].

Цель исследования – разработать набор шаблонов классов, которые позволяют описать статической информацией о типе структуру регулярного выражения. Такое описание с помощью параметров шаблона указывает из каких подвыражений оно состоит, а с помощью имени

шаблона – какой операцией оно формируется. То есть, предполагается построение описания не с помощью объекта, хранимого в памяти в процессе выполнения, а с помощью экземпляра шаблонного типа, формальные параметры которого описывают подвыражения.

Материал и методы. При проведении исследования использовались различные источники по регулярным выражениям и языку C++. Поставленная цель достигается средствами обобщенного программирования посредством шаблонов в языке программирования C++. Кроме того, были использованы методы математического моделирования и общенаучные методы.

Результаты и их обсуждение. Для описания регулярных выражений были написаны шаблоны классов, соответствующие основным операциям над регулярными языками [2].

Шаблон класса Concatenation обозначает операцию конкатенации произвольного количества регулярных выражений. Этот шаблон имеет параметр C – обозначающий тип используемый для представления отдельных символов, а также пакет параметров args, каждый из которых обозначает очередное подвыражение.

Шаблон класса Union обозначает операцию объединения произвольного количества регулярных выражений. Этот шаблон имеет параметр C – обозначающий тип используемый для представления отдельных символов, а также пакет параметров args, каждый из которых обозначает очередное подвыражение.

Шаблон класса Iteration обозначает операцию итерации некоторого регулярного выражения. Этот шаблон имеет параметр C – обозначающий тип используемый для представления отдельных символов, а также параметр arg, который обозначает итерируемое подвыражение.

```
template<class C, class T> struct is_RE
    : std::false_type {};

template <class C, class ... args>
struct Concatenation {
    typedef std::tuple<args...> subexprs;
    typedef typename std::enable_if<
        std::conjunction_v<is_RE<C, args>...>
        >::type enabled_t;
};

template <class C, class ... args>
struct Union {
    typedef std::tuple<args...> subexprs;
    typedef typename std::enable_if<
        std::conjunction_v<is_RE<C, args>...>
        >::type enabled_t;
};

template <class arg>
struct Iteration {
    typedef arg subexpr;
    typedef typename std::enable_if<
        is_RE<C, arg>::value
        >::type enabled_t;
};

template <class C, char c>
struct Character {
    static const C value = c;
    typedef void enabled_t;
};

struct Empty {
    typedef void enabled_t;
};
```

```

struct Null {
    typedef void enabled_t;
};

template<class C, class ... args>
struct is_RE<C, Concatenation<args...>>
    : std::conjunction<is_RE<C, args>...> {};

template<class ... args>
struct is_RE<Union<args...>>
    : std::conjunction<is_RE<C, args>...> {};

template<class arg>
struct is_RE<Iteration<arg>> : is_RE<C, arg> {};

template<class C, C c>
struct is_RE<C, Character<C, c>> : std::true_type {};

template<>
struct is_RE<C, Empty> : std::true_type {};

template<>
struct is_RE<C, Null> : std::true_type {};

```

Для того, чтобы предотвратить указание в качестве подвыражения типа, не описывающего регулярное выражение, были использованы частичные и полные специализации шаблона класса `is_RE`.

Разработанный набор шаблонов классов позволяет описывать регулярные выражения с помощью статической информации о типе, что позволяет осуществлять обработку регулярных выражений, структура которых описана с помощью этого набора шаблонов, на этапе компиляции.

Заключение. Был разработан набор классов, который позволяет описывать структуру регулярного выражения с помощью статической информации о типе. Была реализована проверка того, что все подвыражения описываемого выражения являются регулярными выражениями. Показано, что данный подход обладает некоторым преимуществом: позволяет обрабатывать структуру регулярного выражения на этапе компиляции.

1. Фрилл, Дж. Регулярные выражения / Дж. Фрилл. – СПб : «Питер», 2001. – 352 с.
2. Компиляторы: принципы, технологии и инструментарий / А. В. Ахо, М. С. Лам, Р. Сети, Дж. Д. Ульман. – 2-е изд. : Пер. с англ. – М. : ООО «И. Д. Вильямс», 2008. – 1184 с.

ИССЛЕДОВАНИЕ МЕХАНИЗМОВ РОСТА И РАСПРЕДЕЛЕНИЯ ПРИМЕСИ В ЛЕГИРОВАННЫХ КРИСТАЛЛАХ TGS – TGS +Cr

А.Л. Толстихина¹, Б.С. Роцин¹, В.Н. Шут^{2,3}, С.Е. Мозжаров³, И.Ф. Кашевич²

¹Москва, Институт кристаллографии имени А.В. Шубникова РАН

²Витебск, ВГУ имени П.М. Машерова

³Витебск, ИТА НАН Беларуси

Практически важные свойства сегнетоэлектрических материалов связаны с характером и конфигурацией их доменной структуры. В связи с этим в последние годы в области теоретического и практического исследования сегнетоэлектриков большое внимание уделяется развитию доменной инженерии, целью которой является управление свойствами таких материалов путем создания заданных доменных структур.

Одним из направлений доменной инженерии является получение кристаллов с регулярной доменной структурой (РДС), которая формируется как результат проявления микрозонарного строения примесной подструктуры при выращивании кристаллов из расплава, содержащего