РОССИЙСКАЯ АКАДЕМИЯ НАУК

ВСЕРОССИЙСКИЙ ИНСТИТУТ НАУЧНОЙ И ТЕХНИЧЕСКОЙ ИНФОРМАЦИИ
(ВИНИТИ)

НАУЧНО • ТЕХНИЧЕСКАЯ ИНФОРМАЦИЯ

Серия 2. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ И СИСТЕМЫ

ЕЖЕМЕСЯЧНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ СБОРНИК

Издается с 1961 г. № 2 Москва 2012

# ИНФОРМАЦИОННЫЙ АНАЛИЗ

УДК 001.102 : [002-027.21 : 316.324.8]

В. А. Касумов, К. П. Касумова (Азербайджан)

## Методы определения тематики и содержания индивидуальных и коллективных информационных потребностей

*Рассматриваются понятия «информационная потребность» и «субъект деятельности», предлагаются модели деятельности субъекта и информационной потребности. Описываются этапы формирования информационной потребности. Предложены классификация информационных потребностей, методы определения индивидуальных и коллективных информационных потребностей, а также тематических направленностей и профилей информационных ресурсов и информационных потребностей.*

**Ключевые слова:** *субъект, деятельность субъекта, информационная потребность, тематический профиль, потребители информации, информационный потенциал, реальная информационная потребность, осознанная информационная потребность, выраженная информационная потребность, индивидуальная информационная потребность, коллективная информационная потребность*

### 1. ВВЕДЕНИЕ

Нет необходимости доказывать, что при любой профессиональной или непрофессиональной деятельности человека возникает потребность в информации. При этом содержание информационной потребности зависит от области деятельности, личных, профессиональных и деловых интересов и потенциалов людей. Так, любой человек помимо профессиональной деятельности может заниматься учебой, общественно-политической и спортивной деятельностью, интересоваться социально-бытовыми делами, иметь хобби и т. д. Исходя из этого можно сказать, что информационная потребность, как правило, состоит из многочисленных

компонентов, отличающихся друг от друга по характеру и содержанию [1–5].

Во многих случаях потребность в информации в некоторой степени удовлетворяется семьей, окружающей средой, школой, высшими учебными заведениями, соответствующими организациями и т. д. Однако эти знания не всегда являются достаточными в реальной жизни. Часто для выполнения конкретной задачи, в том числе для осознания и решения проблемы, принятия решений, наряду с существующими знаниями необходимы дополнительные сведения и знания.

На первый взгляд может показаться, что в настоящее время спектр ресурсов и служб, осуществляющих информационную поддержку, является весьма широким, что позволяет достаточно точно и полно обеспечить удовлетворение существующих информационных потребностей, а также профессиональных интересов разных групп потребителей. Однако практика показывает: вопрос качественного удовлетворения информационных потребностей пользователей до сих пор остается нерешенным.

Для улучшения качества функционирования систем информационного обеспечения в разных сферах деятельности, в том числе библиотечно-информационных систем, значительное место в комплексе осуществляемых мероприятий должно занимать развитие спектра информационных ресурсов и служб, направленных на удовлетворение профессиональных интересов потребителей информации. Именно поэтому в настоящее время большое значение имеет изучение информационных потребностей [6].

Следует отметить, что информационные потребности могут быть рассмотрены в нескольких аспектах [7]:

1. Содержание информационных потребностей.
2. Тематика информационных потребностей.
3. Форма и причина возникновения информационных потребностей.
4. Интенсивность возникновения информационных потребностей.
5. Динамика информационных потребностей и т. д.

Однако с научной точки зрения наиболее актуальным является исследование содержания и тематики (тематической направленности, профиля) информационных потребностей. С учетом этих обстоятельств в настоящей статье исследованы этапы формирования информационных потребностей, представлена их классификация, предложены модели информационной потребности и порождающей её деятельности, разработаны методы определения содержания и тематики индивидуальных и коллективных информационных потребностей, а также тематической направленности и профилей информационных ресурсов и информационных потребностей вне зависимости от области деятельности.

## 2. МОДЕЛИРОВАНИЕ ДЕЯТЕЛЬНОСТИ СУБЪЕКТОВ, ПОРОЖДАЮЩЕЙ ИНФОРМАЦИОННУЮ ПОТРЕБНОСТЬ

Без предварительного определения основных групп потребителей информации, в профессиональной деятельности которых могут появляться информационные потребности, невозможно организовать информационную службу на требуемом уровне. Несомненно, на информационную потребность субъектов влияют их социально-демографические, личные и деловые качества (уровень образования, трудовой стаж, возраст, жизненный опыт и т. д.). Поэтому при организации информационного обслуживания необходимо учитывать сферы профессиональной деятельности, специальности, должностные статусы, функциональные обязанности и уровни профессиональной подготовленности субъектов.

Для классификации потребителей информации используются разные подходы. Прежде всего, потребителей информации можно разделить на две категории: реальные и потенциальные. Вообще, к потребителям информации, в профессиональной деятельности которых в некоторой степени появляется информационная потребность, можно отнести следующих: ученые, ведущие исследования определенных научных проблем; специалисты, работающие в разных сферах производства; научно-педагогический состав и студенты учебных заведений; руководители организаций; участники, координаторы и руководители проектов; менеджеры, бизнесмены, коммерсанты; служащие, рабочие и т. д.

Однако на практике при осуществлении информационного обслуживания выделяют четыре основные группы потребителей информации [1–5, 8–10].

В первую группу входят *все руководители независимо от уровней*. Они являются более высококвалифицированными специалистами и обладают правами принятия решений в условиях нехватки времени. Естественно, оптимальность принимаемых решений сильно зависит от информированности руководителей. Чем выше уровень руководителя, тем шире круг влияния принимаемых им решений. Информационная потребность руководителей высокого уровня отличается многоаспектностью и широким диапазоном тематики.

Вторую группу потребителей информации составляют *специалисты*. К этой категории относятся все лица, работающие в разных сферах деятельности, в том числе в области науки, образования, культуры и т. д. В профессиональной деятельности специалистов часто возникает потребность в научно-технической и других видах информации. Как правило, тематику информационных потребностей специалистов определяют их специальности. На содержание информационной потребности этой группы также значительно влияют сферы деятельности (наука, образование, производство и т. д.), место работы, функциональные обязанности. Например, специалистов можно разделить на следующие подгруппы:

– научно-исследовательские работники (ученые, педагогические работники – учителя, преподаватели);

– специалисты, работающие в разных сферах деятельности (врачи, экономисты, агрономы, зоотехники и др.);

– администраторы, занимающиеся управленческой деятельностью и т. д.

В рамках специальности потребителей информации можно классифицировать по характеру выполняемых работ, занимаемой должности и исполняемым функциям.

В третью группу потребителей информации входят *инженерно-технические работники*, непосредственно работающие на производстве. Они составляют основную часть потребителей информации. Информационные потребности этой категории характеризуются относительной стабильностью. Однако постоянный научно-технический прогресс, особенно обновление технологических процессов в промышленности, замена устаревшего оборудования, машин, устройств, принадлежностей новыми, применение новых информационных технологий, электронной техники, значительно влияет на содержание информационных потребностей инженерно-технических работников, что одновременно стимулирует их деятельность.

Четвертую группу потребителей информации составляют *бизнесмены, предприниматели, коммерсанты*. У них часто появляется потребность в фактографической информации. Однако информационный интерес лиц, входящих в эту категорию, может значительно отличаться по диапазону.

Вообще, любую деятельность субъекта можно выразить через следующую четверку:

$$F = \{S, P, A, V\}, \quad (1)$$

где $S$ – субъект, выполняющий деятельность, $P$ – объект или предметная область деятельности, $A$ – цель деятельности, $V$ – средства и механизмы осуществления деятельности.

Под субъектом здесь понимаются отдельные лица, коллективы (группы людей, объединенных определенной областью деятельности или интересов), в деятельности которых возникает информационная потребность, независимо от ее формы, типа, предметной области, охвата и назначения [1, 3–5].

Связи между элементами, входящие в модель (1), осуществляются через субъект деятельности: субъект реализует цель своей деятельности в объекте (предметной области) с использованием определенных механизмов, системы действий и операций.

Прежде чем осуществлять свою деятельность, субъект, как правило, строит ее модель в своем сознании. Эта модель носит информационный характер и формируется в сознании субъекта на основе имеющейся у него системы знаний. В такой модели предварительное отображение объективных условий и построение субъективного образа деятельности обеспечивают целенаправленное управление ею. Поэтому удачное завершение деятельности и достижение поставленной цели непосредственно зависит от степени информированности субъекта, другими словами, от его потенциала. В результате сознание человека превращается в главный механизм управления деятельностью, а также выполняет такие функции, как принуждение, регулирование и контролирование.

В большинстве случаев построение в сознании субъекта полной модели деятельности сталкивается с определенными трудностями. Его также невозможно осуществлять без соответствующего информационного обеспечения и дополнительного знания. В этой связи субъекту необходимо принимать меры с целью получения недостающей информации.

## 3. КЛАССИФИКАЦИЯ ИНФОРМАЦИОННЫХ ПОТРЕБНОСТЕЙ

Информационные потребности можно классифицировать по многочисленным критериям, но наиболее часто встречаются следующие подходы [1, 8–11]:

**1. По степени охвата информационной потребности:**

– персональные информационные потребности – информационные потребности отдельных лиц;

– коллективные информационные потребности – информационные потребности, сформированные в коллективном сознании групп людей, объединенных вокруг определенных интересов, таких, как трудовая деятельность, наука, образование, искусство, хобби и т. д.

Коллективные информационные потребности формируются на основе информационных потребностей отдельных лиц, образующих коллектив. Однако их нельзя понимать как простое суммирование (объединение) персональных информационных потребностей, поскольку коллективные информационные потребности формируются в результате суммирования только той части персональных информационных потребностей, которая соответствует общим целям и функциям коллектива.

**2. По уровню реализации информационной потребности:**

– реальная информационная потребность – информация (знание), необходимая для решения некоторых задач или выполнения должностных обязанностей;

– осознанная информационная потребность – информация (знание), требуемая субъектом для решения некоторой задачи после понимания им этой задачи.

**3. По степени выражения информационной потребности:**

– пассивные (не выраженные) информационные потребности;

– активные (выраженные) информационные потребности.

**4. По времени существования информационной потребности:**

– временные информационные потребности;

– постоянные информационные потребности.

**5. По сфере формирования информационной потребности:**

– профессиональные информационные потребности;

– непрофессиональные информационные потребности.

**6. По тематике требуемой информации:**

– потребность в узкопрофильной информации;

– потребность в широкопрофильной информации.

**7. По виду требуемой информации** можно выделить информационные потребности:

– в фактографической информации,

– в концептуальной информации,

– в научной информации,

– в методической информации,

– в инструктивной (руководящей) информации и так далее.

**8. По типу требуемого документа:**

– книги,

– журналы,

– специальные научно-технические документы,

– отчеты и т. д.

**9. По потребителю информации:**

– руководители;

– ученые, учащиеся;

– инженеры;

– предприниматель, бизнесмен, коммерсант и т. д.

## 4. ЭТАПЫ ФОРМИРОВАНИЯ ИНФОРМАЦИОННОЙ ПОТРЕБНОСТИ

Как было отмечено выше, информационная потребность возникает в процессе некоторой деятельности и на ее содержание влияют личные, деловые и социально-демографические качества (образование, мировоззрение, трудовая практика, жизненный опыт, личные интересы, хобби и т. д.) субъектов, осуществляющих эту деятельность. Поэтому с точки зрения организации работы информационного обеспечения очень важно учитывать области профессиональной деятельности, должностные положения, функциональные обязанности, профессиональные и другие качества субъектов.

Необходимо подчеркнуть, что не всякая деятельность субъекта приводит к формированию у него информационной потребности. Это зависит, прежде всего, от характера, объекта, цели деятельности и информационного потенциала субъекта. Если субъект для реализации поставленной цели повторяет ранее осуществленную деятельность или объект деятельности полностью (или частично) ему известен, тогда необходимость получения дополнительной информации уменьшится или же в течение определенного периода времени вообще перестанет существовать. Иными словами, если некоторая профессиональная или непрофессиональная деятельность повторяется, основывается на ранее полученной информации и жизненном опыте, субъект обладает достаточным знанием или опытом работы, то в таком случае потребность в получении новой информации будет минимальной.

При организации информационного обслуживания должны быть учтены следующие факторы:

1. Очень сложно описать, особенно измерить, информационную потребность, которая является психосоциологическим и психологическим феноменом. Люди часто затрудняются сформулировать или не могут достаточно точно определить свою информационную потребность. Лексические и семантические неопределенности, допускаемые при оформлении поисковых запросов, в значительной степени влияют на точность и полноту результата поиска (найденных данных).

2. В настоящее время информационная потребность характеризуется многоаспектностью и частотой изменения. Например, информационные потребности ученых и специалистов по содержанию и количеству являются различными, с развитием научно-технического прогресса их содержание меняется.

3. Сложность получения точной информации об информационных потребностях обусловлена их прогнозным характером. Часто эта проблема возникает при формировании информационной потребности в условиях объективной неопределенности из-за низкого уровня информированности.

4. Особенности информационно-поисковых языков поисковых систем отрицательно влияют на качество выражения информационной потребности в виде поисковых запросов, в результате в значительной степени снижается точность поиска. Так, во время формирования поисковых запросов при устранении неопределенностей лексического (неоднозначность терминов), семантического (неточность выражения содержания темы на информационно-поисковом языке) и метаинформационного характера возникают объективные трудности.

5. В связи с беспрерывным ростом объема информационных ресурсов, появлением широких возможностей в сфере производства и потребления информации в настоящее время наряду с существующими проблемами возникают достаточно сложные проблемы, связанные с современным эколого-информационным состоянием данной области.

Исходя из вышесказанного отметим, что в формировании информационной потребности главную роль играют три фактора [1, 8]:

- существование потребности – возникновение нужды в знаниях, необходимых для осуществления деятельности, ощущение нехватки информации для формирования требуемого знания и попытка ликвидации этой нехватки;
- содержание потребности – владение сведениями о тематике, содержании, форме, количестве, а также о существовании необходимой информации;
- информационный потенциал – осознание субъектом факта нехватки информации, владение необходимым знанием для определения нужной (недостающей) информации.

Представление субъекта о необходимой информации формируется на основе существующей у него системы знаний об объекте, т.е. информационного потенциала. Информационный потенциал влияет, прежде всего, на содержание информационной потребности, связанной с объектом деятельности. Существует прямая зависимость между информационным потенциалом и профилем деятельности субъекта: с ростом информационного потенциала субъекта растет и его информационная потребность, но после достижения определенного уровня информационного потенциала потребность в дополнительной информации уменьшается, а после образования развитой (совершенной) системы знаний информационная потребность в данной области деятельности перестает существовать (рис. 1).


Рис. 1. Зависимость информационной потребности субъекта от его информационного потенциала

Отсутствие соответствующего информационного потенциала у субъекта может привести к формированию неопределенного и неадекватного представления о необходимой (недостающей) информации, в результате чего у субъектов с разными информационными потенциалами, занимающихся одинаковой деятельностью, могут формироваться разные по содержанию информационные потребности [7, 10].

В целом процесс формирования информационной потребности включает в себя четыре этапа (рис. 2).

На первом этапе существует *реальная информационная потребность*, т.е. та информационная потребность, которая должна формировать системы знаний, необходимых для решения поставленной задачи или выполнения функциональных обязанностей. Она может быть не осознана субъектом до конца.

На втором этапе, после понимания предстоящей задачи, в сознании субъекта формируется *осознанная информационная потребность*. Как правило, на практике реальная информационная потребность отличается от осознанной информационной потребности. Иными словами, в зависимости от степени информированности (потенциала) субъекта в соответствующей области может возникнуть одно из следующих состояний:

- субъект не знает, какая информация необходима для решения задачи;
- часть нужной информации уже имеется у субъекта;
- субъект владеет некоторыми знаниями в данной области, он является высококвалифицированным специалистом или опытным работником.

На третьем этапе в результате выражения субъектом осознанной информационной потребности с помощью естественного языка формируется *выраженная информационная потребность*.

На четвертом этапе на основе выраженной на естественном языке информационной потребности с помощью формальных поисковых средств (информационно-поискового языка) составляется поисковый запрос, который передается информационной службе или информационно-поисковой системе. Этот поисковый запрос является *сформулированной информационной потребностью*.

Таким образом, под **информационной потребностью** понимается осознанная потребность субъекта в информации, необходимой для получения недостающих знаний в процессе выполнения деятельности. Эту потребность в общем можно выразить как разность между реальной информационной потребностью и знаниями, имеющимися у субъекта:

$$IT=TB-MB, \qquad (2)$$

где $IT$ – осознанная информационная потребность, $TB$ – знания, необходимые для выполнения функциональных обязанностей в области деятельности (реальная информационная потребность), $MB$ – знания предметной области, имеющиеся у субъекта.

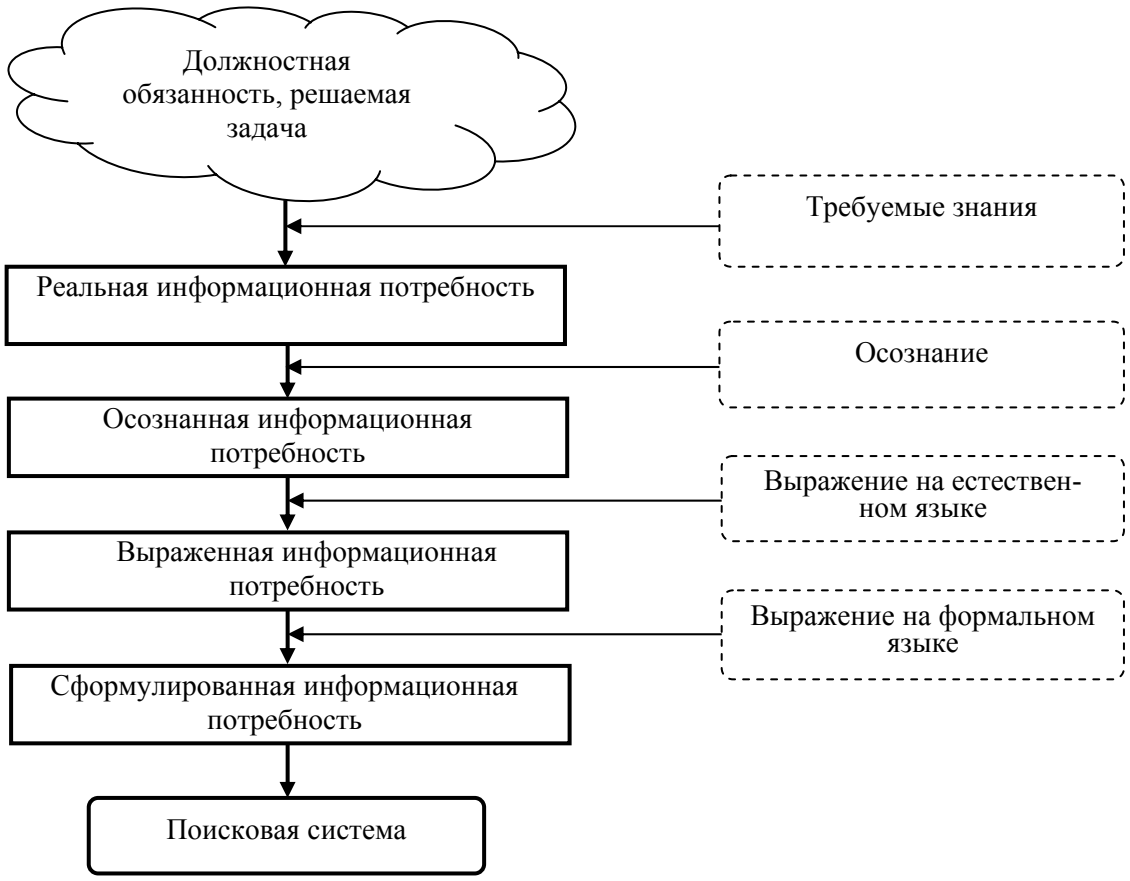


Рис. 2. Этапы формирования информационной потребности

## 5. МОДЕЛЬ ИНФОРМАЦИОННОЙ ПОТРЕБНОСТИ

Учитывая вышесказанное, модель информационной потребности можно представить в следующем виде [1]:

$$IT = \{S, P, R, R^S\}. \quad (3)$$

Здесь $S = \{S_1, S_2, ..., S_n\}$ – множество субъектов (пользователей), выполняющих деятельность, $P = \{P_1, P_2, ..., P_m\}$ – множество предметных областей (профилей) деятельности субъектов, $R = \{R_1, R_2, ..., R_l\}$ – множество информационных ресурсов (в том числе документов, аудио- и видеоматериалов и т. д.), относящихся к предметной области (профилю) и необходимых для осуществления деятельности, $R^S = \{R_1^S, R_2^S, ..., R_l^S\}$ – множество информационных ресурсов, имеющихся у субъектов.

Согласно модели (3) информационную потребность конкретного субъекта $S_j$ в предметной области $P_k$ можно представить в виде:

$$IT_j = \{S_j, P_k, R_k, R_{kj}^S\}, k = \overline{1,m}. \quad (4)$$

Здесь $R_k = \{(r_i, \mu_k(r_i)) \mid r_i \in R, i = \overline{1,l}\}, k = \overline{1,m}$ – множество информационных ресурсов, относящихся к предметной области (профилю) $P_k$; $\mu_k(r_i): R \to [0,1]$ является функцией принадлежности множества $R_k$ и показывает степень вхождения информационного ресурса $r_i$ в предметную область $P_k$. Вычисление значений этой функции можно осуществить путем экспертной оценки. Отметим, что множество $R_k$ представляет собой реальную информационную потребность.

$R_{kj}^S = \{(r_i, \varphi_{kj}(r_i)) \mid r_i \in R, i = \overline{1,l}\}, j = \overline{1,n}, k = \overline{1,m}$ – множество информационных ресурсов, существующих у субъекта $S_j$ по предметной области $P_k$; $\varphi_{kj}(r_i): R \to [0,1]$ является функцией принадлежности множества $R_{kj}^S$ и показывает степень соответствия информационного ресурса $r_i$, имеющегося у субъекта $S_j$, предметной области $P_k$. Значения ее определяются субъектом.

## 6. ОПРЕДЕЛЕНИЕ ТЕМАТИЧЕСКОГО ПРОФИЛЯ ИНФОРМАЦИОННЫХ ПОТРЕБНОСТЕЙ

Пусть $T = \{t_1, t_2, ..., t_f\}$ – множество признаков (терминов, ключевых слов), определяющих тематические профили информационных ресурсов и предметных областей. Тогда тематический профиль предметной области $P_k$ деятельности можно описать следующим образом [1]:

$$P_k \Rightarrow T_k^P = \{(t_q, \omega_k^P(t_q)) \mid t_{q,} \in T, q = \overline{1,f}\}, k = \overline{1,m}. \quad (5)$$

Здесь $\omega_k^P(t_q): T \to [0,1]$ – функция принадлежности множества $T_k^P$, она определяет степень принадлежности термина $t_q$ профилю предметной области $P_k$,, $T_k^P$ – множество терминов, определяющих профиль предметной области $P_k$. Значения функции принадлежности $\omega_k^P(t_q)$ вычисляются заранее при определении профилей предметных областей.

Аналогично, можно описать информационную потребность субъекта $S_j$ элементами множества $T$ следующим образом:

$$IT_j \Rightarrow T_j^{IT} = \{(t_q, \eta_j(t_q)) \mid t_{q,} \in T, q = \overline{1,f}\}, j = \overline{1,n}. \quad (6)$$

Здесь $\eta_j(t_q): T \to [0,1]$ – функция принадлежности множества $T_j^{IT}$, которая показывает степень важности термина $t_q$ для профиля информационной потребности субъекта $S_j$. Ее значения задаются субъектом при описании своей потребности. $T_j^{IT}$ – множество терминов, определяющих профиль информационной потребности субъекта $S_j$. Это множество формируется на основе введенных субъектом ключевых слов путем анализа функциональных областей, предметных областей решаемых им задач, круга интересов, указанных в анкетах субъекта.

С учетом вышесказанного рассмотрим задачу определения тематического профиля информационной потребности. Суть задачи заключается в определении тематических профилей, релевантных информационной потребности субъекта, из множества заданных тематических профилей.

Для этого сначала в отношении отдельных терминов $t_q, q = \overline{1,f}$ вычисляются степени релевантности для каждой пары $IT_j, j = \overline{1,n}$ и $P_k, k = \overline{1,m}$. С этой целью нужно найти пересечение множеств $T_j^{IT}$ и $T_k^P$ [1]:

$$T_j^{IT} \cap T_k^P, j = \overline{1,n}, k = \overline{1,m}.$$

Иными словами:

$$\mu_{jk}(t_q) = \eta_j(t_q) \times \omega_k^P(t_q), j = \overline{1,n}, k = \overline{1,m}, q = \overline{1,f}. \quad (7)$$

Следует отметить, что по значениям функции $\mu_{jk}(t_q)$ для отдельных терминов трудно сказать что-нибудь о степенях релевантности между тематикой информационной потребности и профилем предметной области. Так как степени важности (значительности) отдельных терминов для информационных потребностей и профилей предметных областей отличаются. Только учитывая степени важности всех терминов и разности между этими степенями, можно определять средневзвешенную степень релевантности между профилями информационных потребностей и предметных областей.

С этой целью для предметной области $P_k, k = \overline{1,m}$ для каждой пары терминов $t_q$ и $t_{q'}$ определяются нечеткие отношения предпочтительности $\lambda_k(t_q, t_{q'})$:

$$\lambda_k(t_q, t_{q'}): T_k^P \times T_k^P \to [0,1]. \quad (8)$$

Значения этой функции показывают, насколько термин $t_q$ предпочтительнее термина $t_{q'}$ для $P_k$, и вычисляются на основе весовых коэффициентов терминов для предметных областей следующим образом:

$$\lambda_k(t_q, t_{q'}) = \begin{cases} 1 - [\omega_k^P(t_q) - \omega_k^P(t_{q'})], & \text{если } \omega_k^P(t_{q'}) < \omega_k^P(t_q) \\ 1, & \text{в обратном случае} \end{cases}. \quad (9)$$

Учитывая вышеотмеченное, а также все термины и нечеткие отношения предпочтительности между ними, можно определять степени релевантности между тематикой информационной потребности и профилями предметной области следующим образом:

$$\mu_k(IT_j) = \max_{t_q}\left\{\max_{t_{q'}}\left\{\min\left\{\mu_{jk}(t_q), \mu_{jk}(t_{q'}), \lambda_k(t_q, t_{q'})\right\}\right\}\right\}, \quad (10)$$

где $j = \overline{1,n}, k = \overline{1,m}, t_q \neq t_{q'}$.

Полученные значения функции $\mu_k(IT_j)$ показывают степень близости информационной потребности субъекта $S_j$ к профилю предметной области $P_k$.

Таким образом, профиль информационной потребности субъекта $S_j$ может быть релевантным профилю всех предметных областей $P_k$, удовлетворяющих условию $\mu_k(IT_j) > \varepsilon$, для наименьшего значения некоторой граничной величины $\varepsilon > 0$.

## 7. ОПРЕДЕЛЕНИЕ ПРОФИЛЯ ИНФОРМАЦИОННОГО РЕСУРСА

Согласно формулам (5) и (6) информационный ресурс $r_i$ можно описать через элементы множества $T$ следующим образом:

$$r_i \Rightarrow T_i^r = \left\{\left(t_q, \rho_i(t_q)\right) \middle| t_q, \in T, q = \overline{1,f}\right\}, i = \overline{1,l}. \quad (11)$$

Здесь $\rho_i(t_q) : T \to [0,1]$ – функция принадлежности термина $t_q$ множеству $T_i^r$, описывающая информационный ресурс $r_i$, которая показывает степень важности термина $t_q$ для информационного ресурса $r_i$. Ее значения можно вычислить с помощью методов автоматического индексирования. Для этого разрабатывается специальное программное обеспечение, например, программы робота, паук и т. д.

Учитывая вышесказанное, для определения степени близости (релевантности) тематики информационного ресурса $r_i, i = \overline{1,l}$ к профилю предметной области $P_k, k = \overline{1,m}$, т.е. степени принадлежности $r_i, i = \overline{1,l}$ множеству $P_k$ можно пользоваться нижеследующим методом.

Сначала вычисляются степени релевантности профилей для каждой пары $r_i, i = \overline{1,l}$ и $P_k, k = \overline{1,m}$ в отношении отдельных терминов $t_q, q = \overline{1,f}$. Для этого следует найти пересечение множеств $T_k^P$ и $T_i^r$:

$$T_k^P \cap T_i^r, i = \overline{1,l}, k = \overline{1,m}.$$

Другими словами:

$$\gamma_{ik}(t_q) = \omega_k^P(t_q) \times \rho_i(t_q), i = \overline{1,l}, k = \overline{1,m}, q = \overline{1,f}. \quad (12)$$

Отметим, что аналогично предыдущему разделу для определения средневзвешенной степени релевантности между предметными областями и информационными ресурсами в целом можно использовать нечеткие отношения предпочтительности $\lambda_k(t_q, t_{q'})$.

С учетом вышесказанного можно определять степени релевантности информационного ресурса $r_i$ профилю предметной области $P_k$ следующей формулой:

$$\gamma_k(r_i) = \max_{t_q}\left\{\max_{t_{q'}}\left\{\min\left\{\gamma_{ik}(t_q), \gamma_{ik}(t_{q'}), \lambda_k(t_q, t_{q'})\right\}\right\}\right\}, \quad (13)$$

где $i = \overline{1,l}, k = \overline{1,m}, t_q \neq t_{q'}$.

Если $\varepsilon \in [0,1]$, тогда все предметные области $P_k$, удовлетворяющие условию $\gamma_k(r_i) > \varepsilon$, могут быть релевантными профилю информационного ресурса $r_i$.

## 8. ОПРЕДЕЛЕНИЕ ИНДИВИДУАЛЬНОЙ ИНФОРМАЦИОННОЙ ПОТРЕБНОСТИ СУБЪЕКТА ДЛЯ ОТДЕЛЬНОЙ ПРЕДМЕТНОЙ ОБЛАСТИ

Учитывая вышесказанное, для определения информационной потребности субъекта $S_j$ по предметной области $P_k$ нужно найти пересечение множеств $R_k$ и $\overline{R_{jk}^S}$:

$$R_{jk}^T = R_k \cap \overline{R_{jk}^S}, i = \overline{1,l}, j = \overline{1,n}, k = \overline{1,m}. \quad (14)$$

Здесь $\overline{R_{jk}^S}$ – дополнение множества $R_{jk}^S$. Функция принадлежности множества $\overline{R_{jk}^S}$ имеет вид:

$$\overline{\varphi_{jk}(r_i)} = 1 - \varphi_{jk}(r_i). \quad (15)$$

Если

$$R_{jk}^T = \left\{\left(r_i, \alpha_{jk}(r_i)\right) \middle| r_i \in R, i = \overline{1,l}\right\}, k = \overline{1,m}, j = \overline{1,n}, \quad (16)$$

где $\alpha_{jk}(r_i) : R \to [0,1]$ является функцией принадлежности множества $R_{jk}^T$ и показывает потребность субъекта $S_j$ в информационном ресурсе $r_i$, относящемся к предметной области $P_k$. Согласно правилам пересечения нечетких множеств ее значения можно вычислить по следующей формуле:

$$\alpha_{jk}(r_i) = \min_{i=1,l}\left\{\mu_k(r_i), \left(1 - \varphi_{kj}(r_i)\right)\right\}, j = \overline{1,n}, k = \overline{1,m}. \quad (17)$$

Если $\varepsilon \in [0,1]$, тогда все информационные ресурсы $r_i$, удовлетворяющие условию $\alpha_{jk}(r_i) > \varepsilon$, могут быть приняты в качестве потребности субъекта $S_j$ по предметной области $P_k$ и предоставлены ему.

## 9. ОПРЕДЕЛЕНИЕ ИНДИВИДУАЛЬНОЙ ИНФОРМАЦИОННОЙ ПОТРЕБНОСТИ СУБЪЕКТА ДЛЯ ЕГО ДЕЯТЕЛЬНОСТИ В ЦЕЛОМ

Для определения общей информационной потребности субъекта $S_j$ по всем предметным областям, относящимся к его деятельности, необходимо найти объединение всех множеств $R_{jk}^T, j = \overline{1,n}$, содержащих информационные потребности по отдельным предметным областям:

$$R_j^{ST} = \bigcup_{k=1}^{m} R_{jk}^T, j = \overline{1,n}. \quad (18)$$

Если

$$R_j^{ST} = \left\{\left(r_i, \alpha_j^S(r_i)\right) \middle| r_i \in R, i = \overline{1,l}\right\}, j = \overline{1,n},$$

тогда значения функции принадлежности $\alpha_j^S(r_i): R \to [0,1]$ множества $R_j^{ST}$ можно вычислять следующим образом:

$$\alpha_j^S(r_i) = \max_{k=1,m} \alpha_{jk}(r_i), j = \overline{1,n},$$

т.е.

$$\alpha_j^S(r_i) = \max_{k=1,m} \min_{i=1,l} \{\mu_k(r_i), (1-\varphi_{kj}(r_i))\}, j = \overline{1,n}. \quad (19)$$

Если $\varepsilon \in [0,1]$, тогда все информационные ресурсы $r_i$, удовлетворяющие условию $\alpha_j^S(r_i) > \varepsilon$, могут быть предоставлены субъекту $S_j$ в качестве информационных ресурсов, соответствующих его потребности.

## 10. ОПРЕДЕЛЕНИЕ ИНФОРМАЦИОННЫХ ПОТРЕБНОСТЕЙ ВСЕХ СУБЪЕКТОВ, ЗАНИМАЮЩИХСЯ НЕКОТОРОЙ ПРЕДМЕТНОЙ ОБЛАСТЬЮ

Для определения общей информационной потребности по некоторой предметной области $P_k$ иными словами, информационной потребности всех субъектов $S_j, j = \overline{1,l}$, занимающихся этой предметной областью, необходимо найти объединение множеств $R_{jk}^T, j = \overline{1,n}$, содержащих информационные потребности отдельных субъектов по данной предметной области:

$$R_k^{PT} = \bigcup_{j=1}^{n} R_{jk}^T, k = \overline{1,m}.$$

Тогда значения функции принадлежности $\alpha_k^P(r_i): R \to [0,1]$ множества

$$R_k^{PT} = \left\{ \left( r_i, \alpha_k^P(r_i) \right) \middle| \, r_i \in R, i = \overline{1,l} \right\}, k = \overline{1,m} \quad (20)$$

можно вычислить по следующей формуле:

$$\alpha_k^P(r_i) = \max_{j=1,n} \alpha_{jk}(r_i), k = \overline{1,m}, \quad (21)$$

т.е.

$$\alpha_k^P(r_i) = \max_{j=1,m} \min_{i=1,l} \{\mu_k(r_i), (1-\varphi_{kj}(r_i))\}, k = \overline{1,m}. \quad (22)$$

Если $\varepsilon \in [0,1]$, тогда все информационные ресурсы $r_i$, удовлетворяющие условию $\alpha_k^P(r_i) > \varepsilon$, составляют множество информационных ресурсов, соответствующих общей информационной потребности по предметной области $P_k$ Это множество может быть представлено для удовлетворения информационной потребности при информационном обеспечении деятельности по предметной области $P_k$.

## 11. ОПРЕДЕЛЕНИЕ КОЛЛЕКТИВНОЙ ИНФОРМАЦИОННОЙ ПОТРЕБНОСТИ

Для определения общей (коллективной) информационной потребности $R^T = \left\{ \left( r_i, \alpha(r_i) \right) \middle| \, r_i \in R, i = \overline{1,l} \right\}$ всех субъектов, занимающихся в организации всеми предметными областями, необходимо найти объединение множеств $R_j^{ST}$ для $j = \overline{1,l}$ или множеств $R_k^{PT}$ для $k = \overline{1,m}$:

$$R^T = \bigcup_{j=1}^{n} R_j^{ST}, \quad (23)$$

т.е.

$$\alpha(r_i) = \max_{j=1,n} \alpha_j^S(r_i), \quad (24)$$

или

$$R^T = \bigcup_{k=1}^{m} R_k^{PT}, \quad (25)$$

т.е.

$$\alpha(r_i) = \max_{k=1,m} \alpha_k^P(r_i). \quad (26)$$

Здесь $\alpha(r_i): R \to [0,1]$ – функция принадлежности множества $R^T$. Для вычисления ее значений получим следующее выражение:

$$\alpha(r_i) = \max_{j=1,n} \max_{k=1,m} \min_{i=1,l} \{\mu_k(r_i), (1-\varphi_{kj}(r_i))\} \quad (27)$$

Если $\varepsilon \in [0,1]$, тогда все информационные ресурсы $r_i$, удовлетворяющие условию $\alpha(r_i) > \varepsilon$, составляют множества информационных ресурсов, соответствующих общей информационной потребности организации. Это принимается руководством в качестве коллективной информационной потребности организации и используется для удовлетворения информационных потребностей отдельных субъектов, коллективов субъектов и организации в целом.

## ЗАКЛЮЧЕНИЕ

Предлагаемые модели и методы, а также разработанные на их основе методы определения тематики и профилей информационных ресурсов и информационных потребностей обеспечивают научно обоснованный и наиболее целенаправленный подход к удовлетворению потребностей людей в информации, возникающих в процессе их деятельности. Так, определение содержания и тематической направленности информационной потребности, а затем выбор соответствующей предметной области позволяют наиболее точно обеспечить ее. При этом формирование более полноценного (всеохватывающего) множества $T_j^{IT}$ позволяет оптимизировать значения функции принадлежности $\mu_k(IT_j)$. Методы определения индивидуальных и коллективных информационных потребностей позволяют выбирать необходимые (недостающие) информационные ресурсы из множества существующих.

## СПИСОК ЛИТЕРАТУРЫ


1. Касумов В. А., Касумова К. П. Определение тематических профилей информационных потребностей // Информационные технологии моделирования и управления. – 2011. – №1(66). – С. 4 – 10.
2. Касумов В. А., Касумова К. П. Информационные потребности учебной деятельности // ИКТ в образовании. – Баку, 2011. – №1. – С. 33 – 41.
3. Уханов В. А., Коган В. З. Человек: информация, потребность, деятельность. – Томск : Изд-во Томского ун-та, 1991. – 192 с.
4. Уханов В. А. Информационная потребность и деятельность человека. – М., 1993. – 25 с. – Деп. в ИНИОН РАН 19.10.93, № 47585.
5. Гиляревский Р. С., Маркусова В. А., Черный А. И. Научные коммуникации и проблема

информационной потребности // НТИ. Сер. 1. – 1993. – № 9. – С. 1 – 7.
6. Ступкин В. В. Прогнозирование информационно-потребительской ситуации и качество функционирования систем библиотечно-информационного обеспечения // 15-я Юбилейная Международная конференция «Крым 2008. Библиотеки и информационные ресурсы в современном мире науки, культуры, образования и бизнеса» (Судак, 2008) [Электрон. ресурс]. – URL: http://www.gpntb.ru/win/inter-events/crimea2008/disk/93.pdf
7. Блюменау Д. И. К уточнению исходных понятий теории информационных потребностей // НТИ. Сер. 2. – 1986. – № 2. – С. 7 – 12.
8. Дрешер Ю. Н., Атланова Т. А. Изучение информационных потребностей и информационного поведения специалистов в структуре деятельности по обеспечению комфортной информационной среды // Научно-технические библиотеки. – 2005. – № 11; см. также: http://dlib.eastview.com/browse/doc/10057585
9. Евтюхина Е. А. Специфика формирования информационных потребностей (биопсихологический аспект) : дис. … канд. пед. наук. – М., 1998. – 126 с.
10. Коготков С. Д. Формирование информационных потребностей // НТИ. Сер. 2. – 1986. – № 2.– С. 1 – 7.
11. Зенашвили Н. Г. К вопросу о классификации потребностей // Проблемы формирования социогенных потребностей. – Тбилиси, 1981. – С. 28 – 29.

*Материал поступил в редакцию 22.11.11.*

**Сведения об авторах**

**КАСУМОВ Вагиф Алиджавад оглы** – доктор технических наук, профессор, начальник учебного отдела Академии МНБ им. Гейдара Алиева Азербайджанской Республики, Баку
E-mail: gasumov@yahoo.com

**КАСУМОВА Кифаят Паша кызы** – зав. сектором Института информационных технологий Национальной Академии наук Азербайджана, Баку
E-mail: kqasimova@yahoo.com

УДК 004.891:025.321-047.44

М. Л. Халабия

# Анализ форматов библиографической записи на повторяемость элементов данных

*Анализируются форматы семейства MARC для библиографических данных на повторяемость описательных элементов. С помощью статистических ранговых распределений графически показано распределение полей форматов по частоте их использования в библиографической записи.*

**Ключевые слова:** *машиночитаемая каталогизация, элементы библиографических данных*

Значительным вкладом в развитие мировой электронной каталогизации стал формат машиночитаемой каталогизации MARC, разработанный в 1961 г. специалистами Библиотеки Конгресса США под руководством системного аналитика Генриетты Авраам. Первоначально, формат USMARC был создан для того, чтобы распределять информацию, отраженную на издании, в машиночитаемой форме и выводить ее посредством печатного устройства на каталожную карточку. Следует отметить, что для первых двух десятилетий существования формата, его основным результатом являлась печать созданных библиографических описаний на каталожную карточку [1]. Из этих чисто утилитарных целей в США он стал национальным стандартом ввода библиографической информации, ее хранения и коммуникации.

В 1999 г. было объявлено о создании новой системы форматов для библиографических данных MARC 21. Эта система образовалась в результате согласования и слияния форматов США и Канады. С этого момента библиографирующие учреждения, которые были ориентированы в своей работе на формат USMARC, должны были создавать свои библиографические записи в MARC 21 и отслеживать все последующие изменения, включая и дополнения к нему.

Форматы семейства MARC используются для самых разных функций: ввода, отображения и обмена библиографическими записями, представленными в электронных каталогах. В соответствии с трехуровневой моделью ANSI / SPARC, концептуальная схема является «сердцем» электронного каталога. Она отражает все внешние представления пользователя о данных, а сама поддерживается средствами внутренней схемы. Однако эта внутренняя схема является лишь физическим представлением концептуальной схемы. Таким образом, именно концептуальная схема призвана быть полным и точным представлением требований к элементам библиографических данных формата MARC 21. По мнению Г. Лизера форматы семейства MARC, не имеют четко выраженной концептуальной схемы. Она частично отображена в документации, посвященной форматам, частично отображена на страницах профессиональной печати, и в какой-то мере представлена в национальных правилах каталогизации [2]. Однако, на наш взгляд, правила каталогизации не могут содержать признаки концептуальной схемы, так как они основаны на национальных традициях каталогизации. Применение «международного стандартного библиографического описания» (ISBD) как фундамента для концептуальной схемы формата MARC 21 может являться обоснованным решением. Они регламентируют элементы библиографических данных и определяют форму ввода для каждого элемента библиографической записи, включенного в машиночитаемую запись.

Концептуальная схема выступает как основа для создания формата, предназначенного для формирования библиографической записи. Следствием высокой стоимости ее создания является избыточность представленных в MARC 21 элементов данных.

Отсутствие задокументированной концепции элементов данных отрицательно сказывается на функциях полей переменной длины в формате MARC 21 для библиографических данных. Из-за их большого количества, представленных в формате MARC 21 для библиографических данных очень сложно детализировать функции полей, содержащихся в библиографической записи. Представляется возможным идентифицировать функции тех полей, которые реализуют основные цели электронного каталога.

Г. Лизер в [2] и Р. Теннант в [3] отмечают, что главными недостатками USMARC и MARC 21 являются сложность и избыточность элементов, представленных в форматах для библиографических данных.

В лингвистике избыточностью является повторная передача одной и той же информации как буквально, так и косвенно. В последнем случае избыток информации обусловлен традицией либо увеличением надежности передачи сообщения. В каталогизации документов под традицией будем понимать представление библиографической информации, содержащейся на каталожной карточке, в машиночитаемом виде. Отсюда следует, что неизбежным является «наличие элементов или свойств, которые в каком-либо определенным смысле, не являются необходимыми».

Нашел свое подтверждение и тот факт, что избыточность элементов библиографических данных в формате MARC 21 происходит из USMARC. В руководстве по применению библиографических данных можно найти целый ряд примеров повторения одной и той же информации в разных полях машиночитаемой библиографической записи [4].

*Таблица 1*

**Частота появления элементов данных**

| Категория поля | Количество повторений |
|---|---|
| Дата публикации | 18 |
| Место издания | 18 |
| Издание | 14 |
| Характеристики файла | 3 |
| Математические данные | 2 |

Наличие большого количества повторений, которые содержатся в формате для машиночитаемых элементов библиографических записей, говорит о слабой концептуальной схеме форматов семейства MARC 21. Избыточность данных и их линейная структура подтверждается и нашим исследованием, суть которого заключается в определении ядра полей библиографической записи при помощи многопараметрических обобщенных распределений.

В информатике и математической лингвистике для описания ранговых статистических распределений используются такие математические модели как законы Ципфа, Брэдфорда и др. Однако практика показала, что эти законы не могут описывать с достаточной точностью статистические распределения. Для этих целей наиболее пригодны многопараметрические (обобщенные) распределения В. Нешитого [5], с помощью которых мы хотели бы показать, что при библиографическом описании документов следует более внимательно относиться к полям описательных данных MARC-форматов, имеющих огромное значение для пользователя.

Для статистического исследования библиографического описания были отобраны машиночитаемые записи, которые используются в национальных библиотеках Российской Федерации и описывают разные типы и виды документов. Записи, представленные в ЭК РГБ, описаны при помощи формата для библиографических данных MARC 21; соответственно в ЭК РНБ они (записи) представлены в российском коммуникативном формате RUSMARC. При изучении библиографических описаний крупнейших библиотек РФ четко прослеживаются две тенденции: с одной стороны, применение при библиографическом описании формата MARC 21, созданного Библиотекой Конгресса США; с другой – использование в практике каталогизации российского коммуникативного формата RUSMARC. Сложившаяся ситуация отрицательно сказывается на остальных российских библиотеках, которые при внедрении в практику работы новых информационных технологий ориентируются на РГБ и РНБ, в том числе и в вопросах каталогизации документных ресурсов. Таким образом, российские библиотеки используют при создании библиографических записей как формат MARC 21, так и формат RUSMARC. Словом, «битва форматов» представления элементов библиографических данных продолжается. Поэтому для анализа нами была сделана выборка из записей, содержащих элементы описательных данных как формата MARC 21, так и RUSMARC.

Здесь под выборкой будем понимать множество случаев (испытуемых, объектов, событий, образцов), которые с помощью определенной процедуры выбраны из генеральной совокупности для участия в исследовании.

Было отобрано из ЭК РНБ 120 записей в формате RUSMARC на разные типы и виды документов и столько же записей из каталога РГБ в формате MARC 21 и проанализирована частота использования описательных полей библиографической записи при создании библиографического описания.

Для частотного анализа в информатике и математической лингвистике, как правило, применяются две модели ранговых распределений:

1) Дж. К. Ципфа

$$P_r = kr^{-\gamma} \; (k \approx 0{,}1, \gamma = 1) \qquad (1.1)$$

2) С. Брэдфорда (в т.н. ранговой интегральной форме)

$$F(r) = \sum_{i=1}^{r} P_i = \sum_{i=1}^{r} \frac{k}{i} \approx k \ln r + c, \qquad (1.2)$$

где $r$ - ранг поля;

$P_r$ - относительная частота поля (доля полей из общего их числа, найденных в каталоге с рангом $r$). Поля упорядочены по невозрастанию относительных частот.

Эмпирическая кривая рассеяния информации Брэдфорда

$$F^i(r) = f(\ln r)$$

на границе зоны ядра переходит в прямую, при этом число полей в ядре и последующих зонах находятся в отношении $1 : N : N^2$. В ядре и каждой зоне рассеяния содержится равное число библиографических описаний, использующих эти поля.

Выражение (1.2) при логарифмировании преобразуется в прямую:

$$\ln p_r = \ln k - \gamma \ln r, \qquad (1.3)$$

которая утвердилась как одна из основных форм представления ранговых распределений.

Однако график этой зависимости, построенный по опытным данным, близок к прямой лишь в средней части. Наличие кривизны в областях низких и высоких рангов принуждает исследователей либо вводить поправки в модель Ципфа-Брэдфорда, либо искать новые, более подходящие модели.

Естественно, есть работы посвященные исследованию распределений по частоте отдельных лингвистических единиц. Для их описания можно использовать законы Пуассона, нормальный закон Гаусса и логнормальный, однако, для некоторой группы полей библиографической записи статистические распределения не демонстрируют согласия ни с одним из этих трех законов, так как:

1) выбранная форма представления ранговых распределений (в виде прямой (1.3)), не отражает в должной мере их характерные особенности;

2) не располагают достаточно широком набором распределений, предназначенных для описания лингвистических, в том числе ранговых распределений, которые не объединены в системы, что сильно затрудняет подбор выравнивающего распределения;

3) невозможно построить универсальную модель для аппроксимации любых, в том числе ранговых распределений, а еще лучше - во взаимосвязи их с дискретными распределениями и кривой роста новых событий с целью их обобщения;

4) частотный словарь, как правило, представляет собой неоднородную совокупность элементов и, следовательно, ранговое распределение слов не может быть описано никаким одним распределением.

5) исследование проводятся при ограниченном объеме выборки.

Поэтому в работе используется усовершенствованная модель В. Нешитого [5] Ципфа-Брэдфорда, на основании анализа которой можно сделать вывод, что есть единое уравнение (распределение), с достаточной точностью описывающего все многообразие статистических ранговых распределений библиографических форматов. Следовательно, предлагается другая форма их представления:

$$rp_r = f(lnr)$$

По оси ординат будем откладывать произведение ранга на относительную частоту слова с данным рангом, а по оси абсцисс - натуральный логарифм ранга. Такой график имеет принципиальные преимущества перед традиционной формой представления ранговых распределений. Во-первых, он представляет собой кривую распределения, площадь под которой равна единице. Во-вторых, на такой кривой видны колебания самих частот (по оси ординат), а не их логарифмов. В-третьих, статистические ранговые распределения однородных случайных величин имеют одновершинную кривую распределения.

В итоге, на основании собранных данных построены две таблицы статистических ранговых распределений.

Для построения универсальной вероятностной модели аппроксимации статистических распределений обычно применяют дифференциальное уравнение К. Пирсона [6]. Существенным недостатком семейства распределений К. Пирсона является отсутствие обобщенной плотности, представленной в явном виде, что сильно ограничивает ее использование для достоверной оценки. В принципе, можно использовать метод наибольшего правдоподобия Р. Фишера, но для его расчета необходимо знать тип выравнивающего распределения, к сожалению, критерии для его установления неизвестны. Кроме того, метод Фишера [6] требует совместного решения весьма сложных уравнений правдоподобия, число которых равно числу оцениваемых параметров. Таким образом, чтобы оценить необходимые параметры, была применена теория обобщенных распределений В. Нешитого [5]. Причем, для нашего случая подходит его вторая система непрерывных распределений, которая описывается уравнением обобщенной плотности:

$$p(t) = Nt^{k\beta-1}(1-\alpha u t^{\beta})^{\frac{1}{u}-1}, \qquad (1.4)$$

где $\alpha, \beta. k, u$ - параметры, которые вычисляются по статистическим ранговым распределениям; $N$ – нормирующий множитель, зависящий от параметров.

Уравнение 1.4 включает как частные случаи, так и большое количество известных распределений, в том числе закон Ципфа (при $\beta < 0, u = 1$) Следовательно, график плотности (1.4) даже в случае рангового (т.е. убывающего) распределения, заданный в виде зависимости $tp(t) = f(lnt)$, представляет собой выпуклую колоколообразную кривую. Она имеет три характерные точки: моду $lnt_c$ и две точки перегиба $lnt_a$ и $lnt_b$. Координаты этих точек приняты в качестве границ зон рангового распределения (например, ядра и зон рассеяния).

С помощью компьютерной программы SNR1V97 были вычислены аппроксимирующие распределения. В обоих случаях - это распределения 3-го типа (см. [5]) (параметр u < 0).

В случае формата MARC 21 распределение имеет параметры:

$$\alpha u = -1.296172E-03,\ \beta = 2.505966,$$
$$\gamma = k\beta = 1.1804,\ u = -2.038108.$$

Нормирующий множитель $N = 5.211085E-2$.

В случае формата RUSMARC:

$$\alpha u = -4.48807E-8,\ \beta = 7.13074,$$
$$\gamma = k\beta = 1.080418,\ u = -1.730465.$$

Нормирующий множитель $N = 0.0561917$.

На рис. 1 и рис. 2 в координатах $rp_r = f(lnr)$ представлены кривые распределения: отдельными точками – статистические данные и штриховой линией – рассчитанные по программе. Там же отмечены (в логарифмическом масштабе) три характерные точки A, C, B.

В первом случае абциссы характерных точек (пересчитанные путем потенцирования) равны:

$$r_a = 5.5;\ r_c = 10.4;\ r_b = 19.7$$

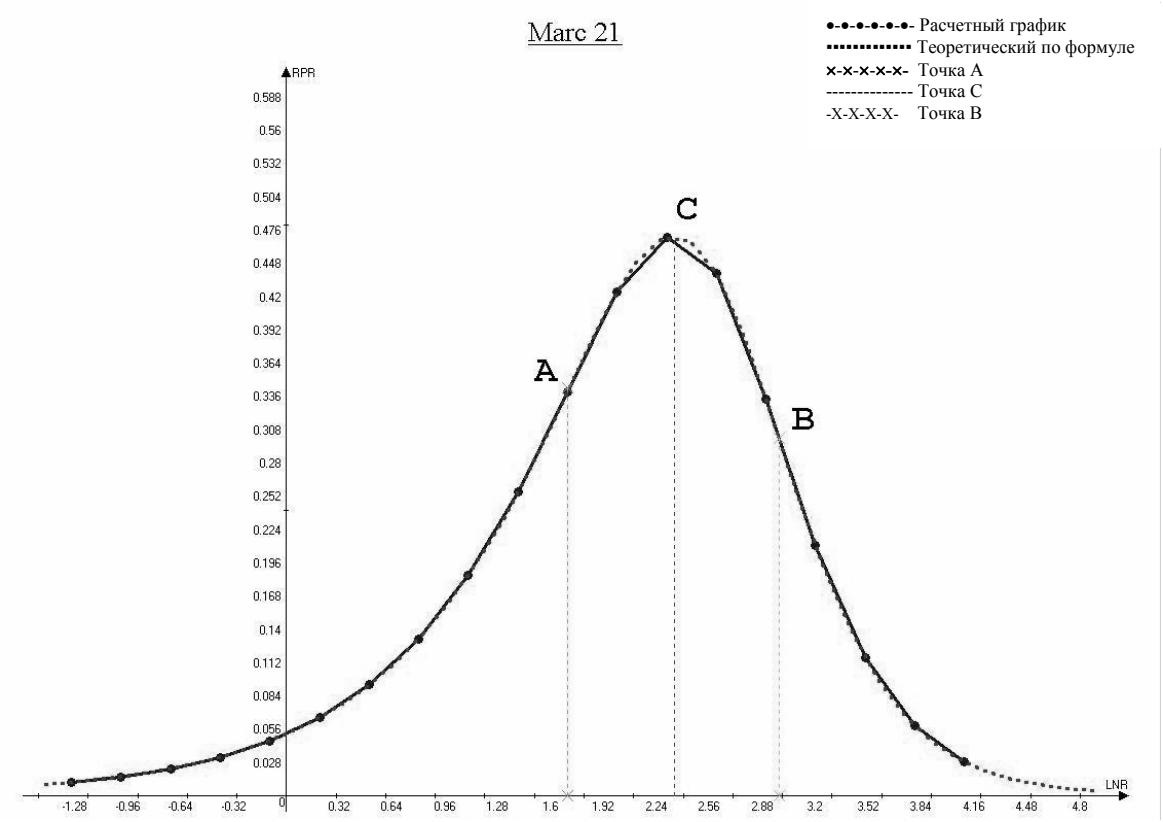


Рис. 1. Ранговое распределение полей Marc 21, полученное теоретическим и расчетным путем (показаны разными линиями)

Во втором случае: $r_a = 7$; $r_c = 9.9$; $r_b = 14$.

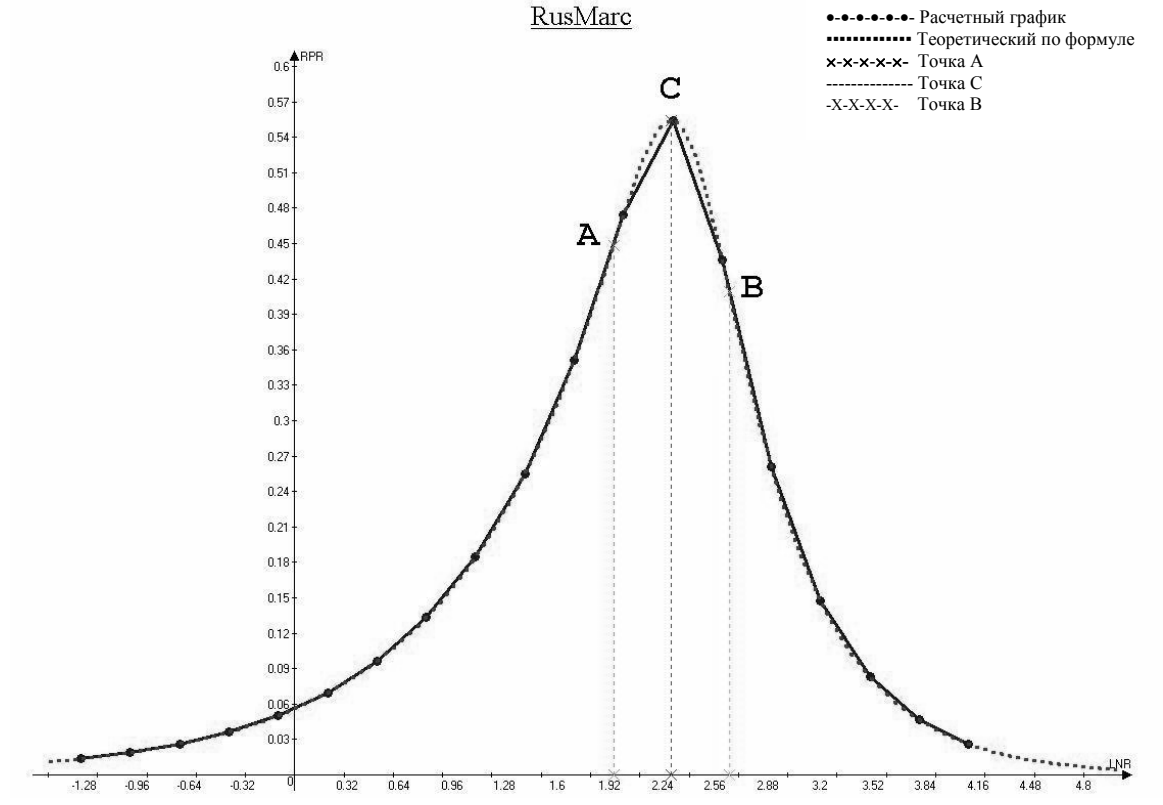


Рис. 2. Ранговое распределение полей RusMarc, полученное теоретическим и расчетным путем (показаны разными линиями)

Поля до точки А составляют ядро. Поля от точки А до точки С образуют первую зону рассеяния; от С до В - вторую зону рассеяния.

Таким образом, можно дать определение понятию «ядро библиографической записи»: часть библиографической записи, включающая наиболее ценные для пользователя описательные поля библиографической записи. Его состав, на наш взгляд, определяется типом и видом документа и содержащейся в нем информации (носителя). Поля, относящиеся к ядру и первой зоне рассеяния, представлены в табл. 2.

*Таблица 2*

**Поля ядра и первой зоны рассеивания**

| MARC 21 | Частота использования | RUSMARC | Частота использования |
|---|---|---|---|
| 008 – Элементы данных фиксированной длины | 120 | 801 – Источник записи | 120 |
| 245 – Область заглавий и сведений об ответственности | 120 | 100 – Данные общей обработки | 120 |
| 041 – Код языка | 120 | 101 – Язык документа | 120 |
| 040 – Организация-создатель записи | 119 | 200 – Заглавие и сведения об ответственности | 120 |
| 260 – Область выходных данных | 118 | 899 – Данные о местонахождении (устар.) | 114 |
| 300 – Область количественной характеристики | 113 | 215 – Физическая характеристика | 113 |
| 084 – Индексы других классификации / Индекс ББК | 105 | 102 – Страна публикации | 112 |
| 017 – Номер регистрации авторского права или обязательного экземпляра | 100 | 105 – Поле кодированных данных: текстовые материалы | 112 |
| | | 210 – Публикация, распространение | 109 |
| | | 686 – Индексы другой классификации | 90 |

Из проведенного исследования видно, что на ядро библиографической записи приходится около 10 описательных полей. Как показывает практика каталогизации документов, большая часть данных в определенных полях является очень важной для пользователя. Отношение каталогизатора к ядру библиографической записи, исходя из вышесказанного, не должно быть поверхностным. Необходимо избегать ошибок в этих полях.

## СПИСОК ЛИТЕРАТУРЫ


1. Leazer G., Smiraglia R. P. Toward the Bibliographic Control of Works: Derivative Bibliographic Relationships in the Online Union Catalog // OCLC Research Bulletin. – 1994. – P. 56 – 59.
2. Leazer G. An examination of data elements for bibliographic description: toward a Conceptual Schema for the USMARC Formats // Library Resources & Technical Services. – 1992. – Vol. 35. – P. 189 – 208.
3. Tennant R. MARC must die // Library Journal. – 2002. – Vol. 127, № 17. – P. 20 – 25.
4. Формат MARC 21 для библиографических данных / Адаптированный пер. с англ. с руководством по применению в российских библиотеках: Рос. гос. б-ка; – М. : Пашков дом, 2002. – 454 с.
5. Нешитой В. В. Элементы теории обобщенных распределений / Учреждение образования «Белорус. гос. ун-т культуры и искусств». – Минск : Респ. ин-т высш. шк., 2009. – 202 с.
6. Каазик Ю. Я. Математический словарь. – М. : Физматлит, 2007. – 334 с.



*Материал поступил в редакцию 14.12.11.*


### Сведения об авторе


**ХАЛАБИЯ Мария Леонидовна** – аспирантка кафедры информационных технологий и электронных библиотек Московского государственного университета культуры и искусств, старший научный сотрудник Российской государственной библиотеки
E-mail: forsite80@mail.ru

# ИНФОРМАЦИОННЫЕ ЯЗЫКИ

УДК 004.434:54

Т. Г. Петров, А. А. Андриянец-Буйко, С. В. Мошкин

# Двупараметрический алфавит для кодирования структурно-химической информации и ее систематизации (на примере турмалина)

*Предлагается способ построения двупараметрического «кристаллохимического алфавита» для кодирования кристаллохимических формул (КХФ) с целью их систематизации. Сформулированы общие принципы кодирования кристаллохимических формул. Двупараметрический алфавит представляет собой упорядоченную совокупность пар символов, в которых зафиксированы сведения о позиции и элементе (ПЭ), присущих данному структурному типу минерала. Стехиометрические коэффициенты элементов в позициях конкретной КХФ позволяют строить ранговые кристаллохимические формулы – последовательности пар ПЭ по снижению их коэффициентов. Совокупность таких ранговых формул упорядочивается по словарному принципу в соответствии с ПЭ в предложенном алфавите, что обеспечивает получение иерархической классификации кодов КХФ. Построение ранговых формул КХФ дает возможность расчетов энтропийных характеристик получаемых кодов при изучении процессов изменений структурно-химического состояния вещества. Описание способа приведено на примере минерала турмалина.*

**Ключевые слова:** *кодирование, кристаллохимический алфавит, позиция-элемент, ранговая кристаллохимическая формула, иерархическая классификация, родство кристаллохимических формул, турмалин*

## ВВЕДЕНИЕ

Проблема единообразного упорядочения химических составов минералов и других геологических объектов была решена ранее [1-6] созданием информационного языка-метода ***RHA***, предназначенного для описания составов любой природы. Здесь ***R***- ранговая формула – последовательность символов химических элементов по мере снижения их атомных содержаний в составе объекта; ***H*** – мера сложности состава – отрицательная сумма произведений содержаний на их логарифмы (энтропия К. Шеннона); ***A*** – мера чистоты состава – отрицательная сумма логарифмов содержаний – анэнтропия. Упорядочение ранговых формул как «слов», в которых символы элементов приняты за «буквы», производится лексикографически (словарно). В качестве алфавита используется Периодическая система элементов. В результате получается иерархическая периодическая классификация составов [6]. Использование ранговых формул химических составов минералов позволило создать «***R***-словарь-каталог химических составов минералов» [7]. В нем в виде ранговых формул представлено все множество химических составов минералов из известных минералогических баз данных Интернета, а также некоторых иных источников. Собрания составов нескольких групп минералов с энтропийными характеристиками (***H*** и ***A***) представлены в статьях: скаполиты [8], эвдиалиты [9, 10], турмалины [11], а также в Интернете: слюды, гранаты, турмалины (по адресу: http://geology.spbu.ru/department/scientific/rha-language-method).

Стандартное написание химических составов минералов не отражает их структурных особенностей, знания которых для минералога являются чрезвычайно важными. Эта информация содержится в кристаллохимических формулах минералов (КХФМ). Поэтому приведение таких формул, параллельно с химическими анализами минералов, стало обычным в геологической литературе. Однако систематизированных собраний КХФМ не существует. Это затрудняет обзор больших массивов данных, идентификацию минералов по их составу, нахождение сходных КХФ, осмысление их разнообразия, оценку распространенности или оригинальности данной КХФ, выявление новых разновидностей и новых минералов. Не существует возможности отразить на плоскости процессы структурно химических изменений минералов, как это можно делать с процессами химических изменений, используя энтропийные характеристики. Отсутствие систематизированных собраний КХФ показывает, что они не могут использоваться для упорядочения без каких-либо изменений их изображения.

Разнообразие структур минералов не позволяет решить задачу выработки единой системы для описания кристаллохимических формул ***всех*** минералов, но не закрывает возможности решения частных задач. Среди них – систематизация материалов с возможностью создания банков данных кристаллохимических формул отдельных групп изоструктурных минералов, а также решение генетических задач, аналогичных уже описанным для химических составов [3, 5, 12, 13]. Кроме того, должно достигаться упрощение визуального восприятия сложных, плохо воспринимаемых последовательностей знаков в кристаллохимических формулах - символы химических элементов с индексами, знаки «+», «–», скобки, коэффициенты при последних.

Опыт работы с химическими составами, с применением метода ***RHA*****,** показал его широкие возможности для систематизации информации [7, 14]. Это побудило разработать систему преобразования (кодирования) и упорядочения коллекций кристаллохимических формул на базе метода ***RHA***. Для такого кодирования авторы считают необходимым и достаточным использование трех параметров: символа позиции, символа элемента и количества сочетаний позиция-элемент в конкретной структуре. То есть, в качестве первого шага требуется разработка способа однозначного построения комплексного - учитывающего и позицию и элемент, алфавита для конкретной группы минералов. Под алфавитом понимается знаковая система с фиксированным значением символов и однозначным линейным их упорядочением.

В настоящее время не существует жестких, общепринятых правил расчета кристаллохимических формул минералов со сложными структурами даже для минералов одной группы. Так, только для турмалина существует 8 (восемь) вариантов «пересчета» КХФ и соответственно их изображения [15]. Это ставит отдельную задачу поиска оптимального способа расчета и представления кристаллохимических формул с выходом за пределы обсуждения вопросов для отдельных групп минералов.

Цель статьи: предложить способ кодирования двупараметрических описаний объектов, в частности, кристаллохимических формул минералов, позволяющий создавать фактографические информационные системы, т. е. позволяющие систематизировать и искать аналоги по свойству, а не по имени объекта, работать с теоретическими и конкретными КХФ, принадлежащими отдельным группам минералов. В качестве примера предложен вариант алфавита для кодирования кристаллохимических формул группы турмалина. Имея в виду, что фактически излагается алгоритм для кодирования кристаллохимической информации для любых минералов, мы не будем обсуждать дискуссионные положения, касающиеся конкретной структуры и распределения ионов в турмалине, выбранном в качестве образца, а также стремиться к полноте самого алфавита. Последнее – отдельная задача.

## ПРИНЦИПЫ И СПОСОБ ПОСТРОЕНИЯ АЛФАВИТА ДЛЯ КОДИРОВАНИЯ КРИСТАЛЛОХИМИЧЕСКИХ ФОРМУЛ

Под термином «алфавит» обычно понимается общепризнанная последовательность символов звуков речи в естественных языках. Однако только в информатике выделены 10 видов алфавитов [16], кроме того, среди наиболее известных существует 8 видов «новых» алфавитов [17]. Алфавиты довольно распространенное средство для фиксации, упорядоченного хранения, кодирования и передачи информации разного рода.

Сформулируем Принципы построения ***кристаллохимического алфавита***, которые должны быть положены в основу системы кодирования двупараметрических кристаллохимических формул минералов (КХФМ), и их упорядочения. На этой же логической базе возможно построение трехпараметрических (и более) алфавитов.

**I. Принцип Соответствия**

Собрание символов алфавита должно соответствовать набору выделенных структурных единиц минерала – позиций – и их заселенности химическими элементами, например, в КХФ «справочного» типа. Под «справочным» понимается тип кристаллохимических формул в наиболее крупных словарях и справочниках [18, 19].

**II. Принцип Содержания**

Комплексный символ (код) должен отражать фиксируемую в КХФ ***позицию*** иона (элемента) и ***символ*** иона (элемента), сокращенно: «ПЭ» (позиция-элемент).

**III. Принцип Порядка**

Порядок символов позиций в создаваемом алфавите должен соответствовать первому алфавиту - наиболее известному (распространенному в публикациях) порядку символов позиций в КХФ. Пары ПЭ ***при одинаковости символов позиций*** упорядочиваются согласно расположению во втором алфавите, за который принята Периодическая система химических элементов.

**IV. Принцип Простоты**

В кодировке КХФ должны быть: а) минимум общепринятой информации о КХФ; б) минимум информации, новой по форме, т.е. того, что требует особых пояснений.

**V. Принцип Минимизации условностей**

При формировании алфавита количество условностей должно быть минимально. К условностям относятся: замена валентности 2 на “k”, валентности 3 на “l” , 4 на “m” и так далее.

**VI. Принцип Ограничения**

Код КХФ не должен отражать в явном виде структуру кристалла. (Для этого существуют другие средства.) Так, все скобки раскрываются и отбрасываются. Сочетания знаков типа $Si_4O_{10}$, $BO_3$ расчленяются.

Согласно особенностям используемого программного обеспечения (Petros3), а) максимальное количество символов в коде не должно превышать четырех знаков, б) в символы алфавита не включается цифровая информация. Таким образом, эти два последних ограничения являются техническими.

Итак, для построения алфавита необходимо и достаточно иметь:

1) упорядоченный перечень позиций;

2) упорядоченный перечень элементов, занимающих эти позиции.

Возможны два варианта построения алфавита, кодирующего связку химического элемента и его позиции в структуре минерала:

1. Позиция-элемент, где на первом месте в коде находятся символы позиций, а на втором – символы химических элементов.

2. Элемент-позиция, где на первом месте в коде стоят символы элементов, а на втором – символы позиций.

Варианты равновозможны. Однако степень определенности существования элементов и существования позиций в кристалле существенно различна. Так, кристалл как твердое тело в первую очередь характеризуется структурой, т. е. некоторым ограниченным набором возможных позиций для элементов. Что же касается химических элементов и ионов, которые могут входить в состав данного минерала, то существуют неопределенности и длины списка элементов (в турмалине обнаружено свыше 50), и количественных соотношений между ними. Поэтому представляется более естественным использовать именно вариант «позиция-элемент» (ПЭ). Такой подход позволит создавать алфавиты и, соответственно, систематизированные базы данных для групп изоструктурных и иных родственных по структуре веществ. При построении подобных алфавитов для других типов объектов естественно придерживаться подобного же обоснования расположения символов в коде, т. е. на первое место ставить символ более инертного параметра, более важного.

Для создания «идеально полного» алфавита для группы минералов требуется знание всех позиций и всех элементов, установленных для каждой позиции. Столь жесткое требование практически невыполнимо из-за отсутствия полной информации о распределении атомов в структуре. Оно также невыполнимо, из-за невозможности предвидеть обнаружение новых элементов в минерале при повышении чувствительности аналитических методов и при обнаружении новых разновидностей. Поэтому для иллюстрации способа построения алфавита (согласно принципу соответствия) остановимся на форме представления КХФ, которая является наиболее обычной, общепринятой по набору позиций и элементов.

Построение кристаллохимического алфавита производится следующим образом.

Первый член алфавита позиция (П) последовательно, без пробелов, объединяется с символами элементов (Э), встречающихся в данной позиции, в порядке, отвечающему Периодической системе элементов. Получаем, например, $П_1Э_1$, $П_1Э_3$, $П_1Э_8$.

Когда будут исчерпаны варианты элементов, которые могут находиться в $П_1$, начинаем строить участок алфавита, описывающий связки второй позиции с элементами (которые могут находиться в ней). Соответственно, получим ряд, например: $П_2Э_3$, $П_2Э_9$, $П_2Э_{25}$ и так далее до исчерпания пар символов «позиция-элемент».

Отметим, что члены алфавита являются комплексными (составными) символами, имеющими определенный смысл, как и символы Периодической системы элементов. Далее мы будем их называть «***буквами***» ***кристаллохимического алфавита,*** из которых будут составляться ***слова информационного языка описания КХФМ,*** а именно ***ранговые кристаллохимические формулы минералов – R-КХФМ.***

## ПОСТРОЕНИЕ АЛФАВИТА «ПОЗИЦИЯ-ЭЛЕМЕНТ» ДЛЯ КРИСТАЛЛОХИМИЧЕСКИХ ФОРМУЛ ТУРМАЛИНА

В качестве опорных были взяты работы [18, 20, 21]. Мы ограничились этими работами, полагая, что на способы построения алфавита и ранговых формул увеличение количества выделяемых символов и видов минерала не влияет.

Итак, имеем теоретическую формулу минералов группы турмалина: $X\ Y_3\ Z_6\ [T_6O_{18}]\ [BO_3]_3\ V_3\ W$. В этой формуле выделены восемь различаемых в справочной литературе позиций: X, Y, Z, T, B, O, V, W. Поскольку символы Y, O, V и W являются и символами химических элементов в Периодической системе, следует признать неудачным использование их для обозначения позиций в ***R***-КХФ. Чтобы не порождать излишних дискуссий, ограничимся лишь констатацией этого.

Как видим, в теоретической формуле турмалина не различаются и не указываются несколько существующих позиций кислорода, связанного с Si и с B. По литературным данным [21] кислород имеет позиции O4, O5, O6, O7 в группировке с Si и O2, O8 в группировке с B.

Согласно [18], приведем список ионов в теоретических кристаллохимических формулах турмалина:

$Na^+$, $K^+$, $Ca^{2+}$, вакансия (G); $Li^+$, $Mg^{2+}$, $Al^{3+}$, $Ti^{4+}$, $V^{3+}$, $Cr^{3+}$, $Mn^{2+}$, $Mn^{3+}$, $Fe^{2+}$, $Fe^{3+}$, $Cu^{2+}$, $Zn^{2+}$; Mg, Al, Si, $B^{3+}$, $O^{2-}$, $O_{Si}$, $O_B$, $(OH)^-$, $F^-$.

Строим комплексный алфавит, приписывая к символу каждой позиции, свойственный ей ион. Получаем ряд:

$XNa^+$, $XK^+$,$XCa^{2+}$, X(G), $YLi^+$, $YMg^{2+}$, $YAl^{3+}$, $YTi^{4+}$, $YV^{3+}$, $YCr^{3+}$, $YMn^{2+}$, $YMn^{3+}$, $YFe^{2+}$, $YFe^{3+}$, $YCu^{2+}$, $YZn^{2+}$, ZMg, ZAl, $ZV^{3+}$; $ZCr^{3+}$, $ZFe^{3+}$; $T(B^{3+})$, TAl, TSi, $BB^{3+}$, B(G ), $V(OH)^-$, $VO^{2-}$; $W(OH)^-$, $WF^-$, $WO^2$,$O_{Si}$O4, $O_{Si}$O5, $O_{Si}$O6, $O_{Si}$O7, $O_B$O2 и $O_B$O8.

Это ***комплексный кристаллохимический*** алфавит для описания КХФ турмалина, представленный 37 комплексными символами. Приведенную символику алфавита, согласно принципу **IV**, следует упростить. В тех случаях, когда ион имеет постоянный заряд, знак заряда убирается. Валентность 2 заменяется на “к”, валентность 3 заменяется на “l” и так далее, согласно обычной математической символике в случаях перечисления не с начала натурального ряда. Символом VWOH обозначим ион OH, различение позиций V и W для которого в рамках кристаллохимической формулы не произведено. Символом OB – обозначим атомы кислорода, входящие в комплексный ион $BO_3$; OSi – атомы кислорода, входящие в комплекс $Si_6O_{18}$.

Таким образом, получаем упрощенный алфавит:

XNa, XK, XCa, XG, YLi, YMg, YAl, YTi, YV, YCr, YMnk, YMnl, YFek, YFel, YCu, YZn, ZMg, ZAl, ZV, ZCr, ZFel, TB, TAl, TSi, OSi, BB, BG, OB, VOH, VO, WOH, VWOH, WO, WF.

Здесь 34 символа. Для наглядности сведем полученные варианты алфавита в таблицу (табл. 1).

*Таблица 1*

**Варианты алфавита: исходный (I) и сокращенный (II)**

| | I | II | | I | II | | I | II |
|---|---|---|---|---|---|---|---|---|
| 1 | $XNa^{+}$ | XNa | 14 | $YFe^{3+}$ | YFeI | 27 | $O_{Si}O6$ | OSi |
| 2 | $XK^{+}$ | XK | 15 | $YCu^{2+}$ | YCu | 28 | $O_{Si}O7$ | |
| 3 | $XCa^{2+}$ | XCa | 16 | $YZn^{2+}$ | YZn | 29 | $BB^{3+}$ | BB |
| 4 | X(G) | XG | 17 | ZMg | ZMg | 30 | B(G ) | BG |
| 5 | $YLi^{+}$ | YLi | 18 | ZAl | ZAl | 31 | $O_BO2$ | OB |
| 6 | $YMg^{2+}$ | YMg | 19 | $ZV^{3+}$ | ZV | 32 | $O_BO8$ | |
| 7 | $YAl^{3+}$ | YAl | 20 | $ZCr^{3+}$ | ZCr | 33 | $V(OH)^{-}$ | VOH |
| 8 | $YTi^{4+}$ | YTi | 21 | $ZFe^{3+}$ | ZFeI | 34 | $VO^{2-}$ | VO |
| 9 | $YV^{3+}$ | YV | 22 | $T(B^{3+})$ | TB | 35 | $W(OH)^{-}$ | WOH |
| 10 | $YCr^{3+}$ | YCr | 23 | TAl | TAl | 36 | (OH) | VWOH |
| 11 | $YMn^{2+}$ | YMnk | 24 | TSi | TSi | 37 | $WO^{2-}$ | WO |
| 12 | $YMn^{3+}$ | YMnl | 25 | $O_{Si}O4$ | OSi | 38 | $WF^{-}$ | WF |
| 13 | $YFe^{2+}$ | YFek | 26 | $O_{Si}O5$ | | 39 | | |

Примечание. Кодировки $O_{Si}O4$ – $O_{Si}O7$ и $O_BO2$, $O_BO8$ обозначают позиции кислорода в группировке с кремнием $Si_6O_{18}$ и с бором $BO_3$ соответственно. Код VW обозначает неразделенные позиции, т. е. отсутствие информации по разделению этих двух позиций.

Анализ кислорода требует решения специальной структурной задачи. Она решается в единичных работах [21]. Кислород будет всегда занимать первые места в кристаллохимической ранговой формуле (из-за его большого содержания), и потому, видимо, не будет влиять на различение видов турмалина. Согласно принципу соответствия при кодировании кислорода учитывается только его принадлежность к центральным ионам: кремнию и бору.

## ПОЛУЧЕНИЕ РАНГОВОЙ КРИСТАЛЛОХИМИЧЕСКОЙ ФОРМУЛЫ МИНЕРАЛА (R-КХФМ)

Построение ранговой формулы с использованием сформированного алфавита ПЭ заключается в следующем:

1) Определяются количества атомов в каждой позиции. Это можно производить непосредственно по исходной кристаллохимической формуле.

2) Все установленные в конкретной КХФМ пары ПЭ располагаются в соответствии со снижением их количеств. (Эти количества отбрасываются на данном этапе работы). В результате получаем ***ранговую кристаллохимическую формулу*** – «слово» кристаллохимического языка. Это слово, «буквы» которого разделены пробелами, может иметь следующий вид:

**OSi OB ZAl = TSi VWOH YMg = BB XNa**

Знаки равенства ставятся и при равенстве коэффициентов при ПЭ, и при их различии не более чем на 15 относительных процентов, т.е. на величину не более 1.15 при делении большей величины на меньшую.

Расположение полученных отображений ***R***-КХФ, т. е. полученных «слов», производится по вертикали с записью символов одного ранга друг под другом. Упорядочение производится по словарному (алфавитному) принципу. В качестве алфавита для упорядочения «слов» используется последовательность пар ПЭ в полученном кристаллохимическом алфавите.

В результате алфавитного упорядочения ранговых КХФ формул автоматически получается линейная иерархическая периодическая ***R***-классификация кристаллохимических формул. Такая классификация, опираясь на количественные соотношения встречаемости сочетаний позиция-элемент, на первых местах в РФ всегда будет иметь наиболее значимые в количественном отношении пары ПЭ, то есть выделять каркас структуры минерала. Поэтому форма исходной КХФ и ее отображение в виде ранговой кристаллохимической формулы будут отличаться.

## КОДИРОВАНИЕ ТЕОРЕТИЧЕСКИХ КРИСТАЛЛОХИМИЧЕСКИХ ФОРМУЛ МИНЕРАЛОВ ГРУППЫ ТУРМАЛИНА И ПОСТРОЕНИЕ ИХ R-КХФ КЛАССИФИКАЦИИ

В качестве исходных материалов были приняты теоретические кристаллохимические формулы всех известных реальных и гипотетических видов турмалина из справочника [18]. Они приведены в табл. 2.

В табл. 2 материал расположен в соответствии с его порядком в исходном издании. Упорядочение такого типа влечет обычное отсутствие какой-либо связи между соседними записями. Это связано с тем, что буквы естественного языка, в отличие от символов химических элементов, не несут смысловой информации [7]. Поэтому во всех словарях, опирающихся на алфавиты естественных языков, между словами, одинаково начинающимися и располагающимися близко и рядом, обычно отсутствуют связи, кроме сходства изображения слов (*кровать, кровь, крокодил, крокус, кролик, крон, крона, кросс*).

С другой стороны, совокупность символов, например, OSiAl четко ориентирует интуицию профессионала, даже не знакомого с информационным языком ***RHA***, на химический тип геологического объекта.

Преодоление показанного свойства обычного способа упорядочения совокупностей названий минералов с их КХФ достигнуто следующим образом.

Для каждого вида турмалина проведено ранжирование комплексных символов ПЭ по снижению их коэффициентов, что сформировало их ранговые формулы (***R***-КХФ). В качестве примера рассмотрим построение ***R***-КХФ увита, теоретическая кристаллохимическая формула которого приведена в табл.3 - строка 1. Позиции указаны в строке 2 и соответствующие им элементы в строке 3. Символы ПЭ в строке 4. Коэффициенты при ПЭ находятся в строке 5. В строке 6 названа процедура их упорядочения. В 7 строке имеем построенную кристаллохимическую ранговую формулу (***R***-КХФ) увита и ниже соответствующие коэффициенты пар ПЭ. Как видим, вся процедура кодирования предельно проста и может быть выполнена вручную.

Полученная совокупность *кристаллохимических ранговых формул* была упорядочена по словарному принципу в соответствии с приведенным выше позиционно-элементным алфавитом. В результате возникла линейная иерархическая ***R***-классификация кристаллохимических формул турмалина (табл. 4).

*Таблица 2*

**Номенклатура и кристаллохимические формулы турмалина**

| | | |
|---|---|---|
| 1 | Бюргерит | $NaFe_3Al_6(BO_3)_3(Si_6O_{18})O_3F$ |
| 2 | Ванадиодравит | $NaMg_3V_6(Si_6O_{18})(BO_3)_3(OH)_4$ |
| 3 | Гидроксилиддикоатит_теор | $Ca(Li_2Al)Al_6(BO_3)_3(Si_6O_{18})(OH)_3O$ |
| 4 | Гидроксиувит _теор | $CaMg_3(Al_5Mg)(BO_3)_3(Si_6O_{18})(OH)_3$ |
| 5 | Дравит | $NaMg_3Al_6\ (BO3)_3(Si_6O_{18})(OH)_4$ |
| 6 | Лиддикоатит | $Ca(Li_2Al)Al_6(BO_3)_3(Si_6O_{18})(OH)_3F$ |
| 7 | Магнезиофойтит _теор | $\square(Mg_2Al)Al_6(Si_6O_{18})(BO_3)_3(OH)_4$ |
| 8 | Оксидравит_теор | $Na(Al_2Mg)(Al_5Mg)(Si_6O_{18})(BO_3)_3(OH)_3O$ |
| 9 | Оксилиддикоатит_теор | $Ca(Li_{1.5}Al_{1.5})Al_6(Si_6O_{18})(BO_3)_3(OH)_3O$ |
| 10 | Оксироссманит _теор | $\square(Li_{0.5}Al_{2.5})Al_6(Si_6O_{18})(BO_3)_3O(OH)_3$ |
| 11 | Оксиувит _теор | $Ca(Al_2Mg)(Al_4Mg_2)(Si_6O_{18})(BO_3)_3(OH)_3O$ |
| 12 | Оксиферрифойтит _теор | $\square(Fe^{3+}{}_2Fe^{2+})(Fe^{3+})_6(Si_6O_{18})(BO_3)_3(OH)_3O$ |
| 13 | Оксиферувит _теор | $Ca(Al_2Fe^{2+})(Al_4Mg_2)(Si_6O_{18})(BO_3)_3(OH)_3O$ |
| 14 | Оксифойтит _теор | $\square(Al_2Fe^{2+})Al_6(Si_6O_{18})(BO_3)_3(OH)_3O$ |
| 15 | Оксихромдравит_теор | $Na(Cr_2Mg)(Cr_5Mg)(Si_6O_{18})(BO_3)_3(OH)_3O$ |
| 16 | Оксишерл _теор | $Na(Al_2Fe^{2+})(Al_5Fe^{2+})(Si_6O_{18})(BO_3)_3(OH)_3O$ |
| 17 | Оксиэльбаит_теор | $Na(Al_2Li)Al_6(Si_6O_{18})(BO_3)_3(OH)_3O$ |
| 18 | Оленит | $NaAl_3Al_6(BO_3)_3(Si_6O_{18})O_3(OH)$ |
| 19 | Повондраит | $NaFe_3Fe_6(BO_3)_3(Si_6O_{18})(OH)_4$ |
| 20 | Россманит | $\square(LiAl_2)Al_6(Si_6O_{18})(BO_3)_3(OH)_4$ |
| 21 | Увит | $CaMg_3(Al_5Mg)(BO_3)_3(Si_6O_{18})(OH)_3F$ |
| 22 | Ферриувит _теор | $Ca(Fe^{2+})_3Al_5Mg)(BO_3)_3(Si_6O_{18})_3(OH)_3O$ |
| 23 | Ферриферувит _теор | $Ca(Fe^{3+}{}_2Fe^{2+})(Fe^{3+}{}_4Mg_2)(Si_6O_{18})_3(BO3)_3\ (OH)_3$ |
| 24 | Фойтит | $\square(Fe_2Al)Al_6(BO_3)_3(Si_6O_{18})(OH)_4$ |
| 25 | Фтордравит _теор | $NaMg_3Al_6\ (BO_3)_3(Si_6O_{18})(OH)_3F$ |
| 26 | Фтороленит | $NaAl_3Al_6(BO_3)_3(Si_6O_{18})O_3F$ |
| 27 | Фторроссманит _теор | $\square(LiAl_2)Al_6(Si_6O_{18})(BO_3)_3(OH)_3F$ |
| 28 | Фторфойтит _теор | $\square(Fe_2Al)Al_6(BO_3)_3(Si_6O_{18})(OH)_3F$ |
| 29 | Фторхромдравит_теор | $NaMg_3Cr_6(BO_3)_3(Si_6O_{18})(OH)_3F$ |
| 30 | Фторшерл _теор | $NaFe_3Al_6(BO_3)_3(Si_6O_{18})(OH)_3F$ |
| 31 | Фторэльбаит _теор | $Na(Li_{1.5}Al_{1.5})Al_6(BO_3)_3(Si_6O_{18})(OH)_3F$ |
| 32 | Хромдравит | $NaMg_3Cr_6(BO_3)_3(Si_6O_{18})(OH)_4$ |
| 33 | Шерл | $NaFe_3Al_6(BO_3)_3(Si_6O_{18})(OH)_4$ |
| 34 | Эльбаит | $Na(Li_{1.5}Al_{1.5})Al_6(BO_3)_3(Si_6O_{18})(OH)_4$ |

Примечание. Аббревиатура «теор» относится к видам, выделенным как крайние члены гипотетических изоморфных рядов. В природе эти минералы пока не встречены.

*Таблица 3*

**Преобразование кристаллохимической формулы увита в ранговую кристаллохимическую формулу увита**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | **КХФ увита** | $CaMg_3(Al_5Mg)(BO_3)_3(Si_6O_{18})(OH)_3F$ | | | | | | | | | |
| 2 | Позиции | X | Y | Z | Z | B | OB | T | OSi | V | W |
| 3 | Элементы | Ca | Mg | Al | Mg | B | O | Si | O | OH | F |
| 4 | Символы ПЭ | XCa | YMg | ZAl | ZMg | BB | OB | TSi | OSi | VOH | WF |
| 5 | Коэффициенты ПЭ | 1 | 3 | 5 | 1 | 3 | 9 | 6 | 18 | 3 | 1 |
| 6 | упорядочение ПЭ по снижению их коэффициентов | | | | | | | | | | |
| 7 | ***R*-КХФ увита** (Коэффициенты ПЭ) | **OSi OB TSi ZAl YMg= BB= VOH XCa= ZMg= WF** ( 18 9 6 5 3 = 3 = 3 1 = 1 = 1 ) | | | | | | | | | |

*Таблица 4*

**Иерархическая *R*-классификация кристаллохимических ранговых формул минералов группы турмалина**

| Ранговые кристаллохимические формулы | | | | | | | | | | | Кристаллохимические формулы |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | |
| OSi | OB | YCr = | TSi | VWOH | YMg = | BB | XNa | | | | 32_$NaMg_3\mathbf{Cr_6}(BO_3)_3(\mathbf{Si_6}O_{18})(OH)_4$ |
| OSi | OB | ZAl = | TSi | YMg = | BB= | VOH | XNa= | WF | | | **25**_$NaMg_3\mathbf{Al_6}(BO_3)_3(\mathbf{Si_6}O_{18})(OH)_3F$ |
| OSi | OB | ZAl = | TSi | YAl = | BB= | VO | XNa= | WF | | | 26_$NaAl_3\mathbf{Al_6}(BO_3)_3(\mathbf{Si_6}O_{18})O_3F$ |
| OSi | OB | ZAl = | TSi | YAl = | BB= | VO | XNa= | WOH | | | 18_$NaAl_3\mathbf{Al_6}(BO_3)_3(\mathbf{Si_6}O_{18})O_3(OH)$ |
| OSi | OB | ZAl = | TSi | YFek= | BB= | VOH | XNa= | WF | | | **30**_$NaFe_3\mathbf{Al_6}(BO_3)_3(\mathbf{Si_6}O_{18})(OH)_3F$ |
| OSi | OB | ZAl = | TSi | YFek= | BB= | VO | XNa= | WF | | | 1_$NaFe_3\mathbf{Al_6}(BO_3)_3(\mathbf{Si_6}O_{18})O_3F$ |
| OSi | OB | ZAl = | TSi | BB= | VOH | YLi | XCa= | YAl= | WF | | 6_$Ca(Li_2Al)\mathbf{Al_6}(BO_3)_3(\mathbf{Si_6}O_{18})(OH)_3F$ |
| OSi | OB | ZAl = | TSi | BB= | VOH | YLi | XCa= | YAl= | WO | | **3**_$Ca(Li_2Al)\mathbf{Al_6}(BO_3)_3(\mathbf{Si_6}O_{18})(OH)_3O$ |
| OSi | OB | ZAl = | TSi | BB= | VOH | YLi = | YAl | XNa= | WF | | **31**_$Na(Li_{1.5}Al_{1.5})\mathbf{Al_6}(BO_3)_3(\mathbf{Si_6}O_{18})(OH)_3F$ |
| OSi | OB | ZAl = | TSi | BB= | VOH | YLi = | YAl | XCa= | WO | | **9**.$Ca(Li_{1.5}Al_{1.5})\mathbf{Al_6}(\mathbf{Si_6}O_{18})(BO_3)_3(OH)_3O$ |
| OSi | OB | ZAl = | TSi | BB= | VOH | YAl | XNa= | YLi= | WO | | **17**_$Na(Al_2Li)\mathbf{Al_6}(\mathbf{Si_6}O_{18})(BO_3)_3(OH)_3O$ |
| OSi | OB | ZAl = | TSi | BB= | VOH | YAl | XG= | YLi= | WF | | **27**_ $\square(LiAl_2)\mathbf{Al_6}(\mathbf{Si_6}O_{18})(BO_3)_3(OH)_3F$ |
| OSi | OB | ZAl = | TSi | BB= | VOH | YAl | XG= | YFek= | WO | | **14**_$\square(Al_2Fe^{2+})\mathbf{Al_6}(\mathbf{Si_6}O_{18})(BO_3)_3(OH)_3O$ |
| OSi | OB | ZAl = | TSi | BB= | VOH | YFek | XG= | YAl= | WF | | **28**_$\square(Fe_2Al)\mathbf{Al_6}(BO_3)_3(\mathbf{Si_6}O_{18})(OH)_3F$ |
| OSi | OB | ZAl = | TSi | BB= | WOH | YAl | XG= | VO= | YLi | | **10**_$\square(Li_{0.5}Al_{2.5})\mathbf{Al_6}(\mathbf{Si_6}O_{18})(BO_3)_3O(OH)_3$ |
| OSi | OB | ZAl = | TSi | VWOH | YMg = | BB | XNa | | | | 5_$NaMg_3\mathbf{Al_6}(BO_3)_3(\mathbf{Si_6}O_{18})(OH)_4$ |
| OSi | OB | ZAl = | TSi | VWOH | YFek= | BB | XNa | | | | 33_$NaFe_3\mathbf{Al_6}(BO_3)_3(\mathbf{Si_6}O_{18})(OH)_4$ |
| OSi | OB | ZAl = | TSi | VWOH | BB | YLi = | YAl | XNa | | | 34_$Na(Li_{1.5}Al_{1.5})\mathbf{Al_6}(BO_3)_3(\mathbf{Si_6}O_{18})(OH)_4$ |
| OSi | OB | ZAl = | TSi | VWOH | BB | YMg | XG= | YAl | | | **7**_$\square(Mg_2Al)\mathbf{Al_6}(\mathbf{Si_6}O_{18})(BO_3)_3(OH)_4$ |
| OSi | OB | ZAl = | TSi | VWOH | BB | YAl | XG= | YLi | | | 20_ $\square(LiAl_2)\mathbf{Al_6}(\mathbf{Si_6}O_{18})(BO_3)_3(OH)_4$ |
| OSi | OB | ZAl = | TSi | VWOH | BB | YFek | XG= | YAl | | | 24_$\square(Fe_2Al)\mathbf{Al_6}(BO_3)_3(\mathbf{Si_6}O_{18})(OH)_4$ |
| OSi | OB | ZV= | TSi | VWOH | YMg = | BB | XNa= | | | | 2_$NaMg_3\mathbf{V_6}(\mathbf{Si_6}O_{18})(BO_3)_3(OH)_4$ |
| OSi | OB | ZCr = | TSi | YMg = | BB= | VOH | XNa= | WF | | | **29**_$NaMg_3\mathbf{Cr_6}(BO_3)_3(\mathbf{Si_6}O_{18})(OH)_3F$ |
| OSi | OB | ZFel= | TSi | BB= | VOH | YFek= | XG= | YFel= | WO | | **12**_$\square(\mathbf{Fe^{3+}}_2Fe^{2+})(\mathbf{Fe^{3+}})_6(\mathbf{Si_6}O_{18})(BO_3)_3(OH)_3O$ |
| OSi | OB | ZFel= | TSi | VWOH | YFek= | BB | XNa | | | | 19_$NaFe_3\mathbf{Fe_6}(BO_3)_3(\mathbf{Si_6}O_{18})(OH)_4$ |
| OSi | OB | TSi | ZAl | YMg = | BB= | VOH | XCa | ZMg | WF | | 21_$CaMg_3(\mathbf{Al_5}Mg)(BO_3)_3(\mathbf{Si_6}O_{18})(OH)_3F$ |
| OSi | OB | TSi | ZAl | YMg = | BB= | VWOH | XCa= | ZMg | | | **4**_$CaMg_3(\mathbf{Al_5}Mg)(BO_3)_3(\mathbf{Si_6}O_{18})(OH)_3$ |
| OSi | OB | TSi | ZAl | YFek= | BB= | VOH | XCa= | ZMg= | WO | | **22**_$CaFe_3^{2+}(\mathbf{Al_5}Mg)(BO_3)_3(\mathbf{Si_6}O_{18})_3(OH)_3O$ |
| OSi | OB | TSi | ZAl | BB= | VOH | YAl | XNa= | YMg= | ZMg= | WO | **8**_$Na(Al_2Mg)(\mathbf{Al_5}Mg)(\mathbf{Si_6}O_{18})(BO_3)_3(OH)_3O$ |
| OSi | OB | TSi | ZAl | BB= | VOH | YAl | XNa= | YFek | ZFel | WO | **16**_$Na(Al_2Fe^{2+})(\mathbf{Al_5}Fe^{2+})(\mathbf{Si_6}O_{18})(BO_3)_3(OH)_3O$ |
| OSi | OB | TSi | ZAl | BB= | VOH | YAl = | ZMg= | XCa= | YMg= | WO | 11_$Ca(Al_2Mg)(\mathbf{Al_4}Mg_2)(\mathbf{Si_6}O_{18})(BO_3)_3(OH)_3O$ |
| OSi | OB | TSi | ZAl | BB= | VOH | YAl = | ZMg | XCa= | YFek= | WO | **13**_$Ca(Al_2Fe^{2+})(\mathbf{Al_4}Mg_2)(\mathbf{Si_6}O_{18})(BO_3)_3(OH)_3O$ |
| OSi | OB | TSi | ZCr | BB= | VOH | YCr | XNa= | YMg= | ZMg= | WO | **15**_$Na(Cr_2Mg)(\mathbf{Cr_5}Mg)(\mathbf{Si_6}O_{18})(BO_3)_3(OH)_3O$ |
| OSi | OB | TSi | ZFel | BB= | VWOH | ZFel | ZMg | XCa= | YFek | | **23**_$Ca(\mathbf{Fe^{3+}}_2Fe^{2+})(\mathbf{Fe^{3+}}_4Mg_2)(\mathbf{Si_6}O_{18})_3(BO3)_3(OH)_3$ |
| 1 | 1 | 6 | 8 | 15 | 18 | 25 | 28 | 32 | 34 | 34 | |

Примечание. Номера кристаллохимических формул соответствует описанию и номерам в Таблице 2. 1 – Бюргерит, 2 – ванадиодравит, 3 – гидроксиливдикоатит, 4 – гидроксиувит, 5 – дравит, 6 – лиддикоатит, 7 – мегнезиофойтит, 8 –оксидравит, 9 – оксилиддикоатит, 10 – оксиросманит, 11 – оксиувит, 12 – оксиферрифойтит, 13 – оксиферувит, 14 – оксифойтит, 15 – оксихромдравит, 16 – оксишерл, 17 – оксиэльбаит, 18 – оленит, 19 – повондраит, 20 – россманит, 21 – увит, 22 – ферриувит, 23 – ферриферувит, 24 – фойтит, 25 – фтордравит, 26 – фтороленит, 27 – фторроссманит, 28 – фторфойтит, 29 – фторхромдравит, 30 – фторшерл, 31 – фторэльбаит, 32 – хромдравит, 33 – шерл, 34 - эльбаит. Жирными шрифтом в номерах кристаллохимических формул выделены теоретические члены группы турмалина. В *R*-КХФ-лах жирным выделены элементы 3-4 рангов как важнейшие после кислорода каркаса турмалина, выделяющие 8 классов по соотношениям элементов в позициях T и Z . В нижней строке таблицы приведено число различающихся RKXФ при разных длинах ранговой формулы (NR).

## ОБСУЖДЕНИЕ

Как видим, ранговые формулы всех КХФ различаются. Это означает, что при кодировании кристаллохимических формул способом, который был использован, их разнообразие (а именно 34 разновидности) не уменьшилось. Это могло бы произойти, если бы не был выдержан принцип соответствия (**I**), или произошло злоупотребление принципом простоты (**IV**).

В нижней строке табл. 4 приведено число различающихся ***R***-КХФ при разных длинах ранговой формулы. Как видим, с 10-го ранга увеличение разнообразия ранговых формул прекращается. Это говорит о достаточности выделенных пар позиция – элемент для полного описания всего набора существующих сегодня теоретических составов турмалина. Понятно, что детальность ранговых формул ПЭ для ***реальных составов***, также как и для создания новых теоретических, величиной 10 не будет ограничиваться.

Общими для всего множества кристаллохимических ранговых формул являются два первых (главенствующих) ранга, кодирующих кислородный каркас структуры организованный и «сцементрированный» кремнием и бором. Третий ранг выделяет 6 классов – видов турмалина. В 4-м ранге восемь ***R***-КХФ – классов, в 5-м ранге 15 и так далее. Так проявляется иерархичность структуры ***R***-классификации кристаллохимических формул. Горизонтальные разделители делают эту структуру легко воспринимаемой. Формально же, иерархия заключается в том, что каждая ранговая формула длиной ***n*** входит в один единственный класс длиною ***n***-1.

Иерархическая структура классификации выявила резкую неравномерность расчленения поля кристаллохимических составов на разновидности. Так, если ***R***-формула OSi>OB>ZAl имеет 20 вариантов продолжений – видов турмалина, то ***R***-формулам OSi>OB>YCr, OSi>OB>ZV, OSi>OB>ZCr соответствуют единичные виды турмалина, а формуле OSi>OB>ZFel принадлежат два. Этому соответствует и то, что хромдравит, ванадиодравит, фтрохромдравит, оксиферрифойтит, повондраит, по своему «кристаллохимическому составу» резко отличаются от большинства остальных, в то время как между лиддикоатитом и гидроксилиддикоатитом, а также оксиувитом и оксиферувитом отличия минимальны. Так визуально проявились различия в степени родства между видами. Чем длиннее общая часть ранговой кристаллохимической формулы, чем короче разделители, тем более близки между собой виды и разновидности турмалина. Такие же соотношения будут между любыми объектами, свойства которых будут описаны с помощью алфавитов, типа представленного здесь.

Степень различия между минералами обычно практически никак не проявляется в названиях и с трудом – в самих кристаллохимических формулах. Ранговая КХФ формула показывает структурную значимость каждого элемента. В этом отношении интересно отметить, что ранги видообразующих элементов широко варьируют. Сам «группообразующий» бор занимает 5-7 ранги, количественно уступая многим элементам. Это обстоятельство можно считать проявлением его высокой структурообразующей роли. Следует также обратить внимание на выделение восьми классов по соотношению заполнений рангов 3 и 4 элементами в позициях T и Z (в табл. 4 выделены жирным шрифтом). Было бы интересно выявить связь между различиями физических свойств и степенью родства по кристаллохимическим свойствам групп турмалина, выделяемых описанным методом.

Перечисленные Принципы-положения, приняты нами как интуитивно ясные и ориентирующие на получение ***первого*** исходного результата с определенными свойствами – алфавита. В свою очередь, он используется для достижения ***второго*** – построения «слов» – ранговых кристаллохимических формул. Последний, в свою очередь, используется для построения конечного результата – иерархической классификации КХФ. Качество алфавита, ранговых КХФ и «наглядность», «удобство» их использования, ценяваются по результатам использования всей системы. Так, бессмысленно оценивать простоту, удобство и полезность обозначений: «три», «3», «I I I», «$^3$» - вне контекста их использования. Кроме того, необходимо и обучение оперированию с системой, а освоение нового, в той или иной степени, затруднено всегда [22, 23].

Разработанный алфавит учитывает только позиции и элементы кристаллохимических формул, относящихся к ограниченному списку гипотетических и теоретических составов турмалина, однако, способ и предложенные принципы не имеют каких-либо ограничений на использование их при работе с реальными составами. Это потребует расширения алфавита. Для этого необходимо и достаточно новые символы ПЭ вставить на соответствующие места алфавита, не меняя относительного положения остальных символов. Пополнение алфавита может потребоваться и при появлении дополнительной структурной информации о позициях отдельных атомов кислорода и группы гидроксила.

Известно, что предметная – содержательная систематизация существенно облегчает работу с материалами, позволяет лучше видеть и достоинства и недостатки материалов исследования. При больших объемах данных кодирование и упорядочение кристаллохимических формул, вместе с соответствующими названиями минералов, позволит однозначно классифицировать данные, наглядно отражать полноту собраний данных, оригинальность и банальность отдельных объектов, производить идентификацию видов внутри групп минералов. Создание банков данных для групп минералов с помощью предлагаемого метода кодирования позволит анализировать собрания минералов с более высоким уровнем определенности, чем это происходит при учете только их химического состава. Представляется, что новый вид систематизации материалов при выявлении аномалий в больших банках данных будет способствовать уточнению позиций элементов в кристаллической структуре и открытию новых минералов.

Важно отметить, что получение ранговых кристаллохимических формул – первой составной части метода ***RHA*** – открывает возможности использования энтропии и анэнтропии как наиболее адекватных характеристик изменения составов при процес-

сах смешения и разделения [24]. К последним относится кристаллизация [25]. Известно, что чем больше параметров сопоставляемых систем учтено, тем более различимы системы. Распределения частот ПЭ более информативны по сравнению с распределениями химических элементов. Поэтому использование энтропийных характеристик кристаллохимических составов, повысит различимость объектов на диаграммах ***HA*** в совокупностях точек, будь то отображениях разновидностей минерала, или процессов их образования [12]. В качестве первого опыта, по материалам, приведенным в [26], были рассчитаны энтропийные характеристики химических и кристаллохимических составов хлоритов (для которых алфавит иной) разных зон рудного месторождения. становлено, что поля точек ***HA*** кристаллохимических составов зон месторождения разделены, в то время как поля ***HA*** химических составов тех же хлоритов перекрываются.

Как полагают авторы, принципы, приведенные в статье, могут быть положены в основу создания аналогичных алфавитов для описания не только других групп минералов, но и комплексных алфавитов для систематизации аналогичных аналитических материалов в иных отраслях знания.

## ЗАКЛЮЧЕНИЕ

Следует подчеркнуть, что предложенный способ кодирования кристаллохимических формул не есть замена самих формул, точно так же, как кодовое имя, название или описание не заменяет самого предмета и на него не похоже. Код КХФ - только форма отображения КХФ. Поэтому достоинства и недостатки предложенного кода могут сопоставляться только с другими вариантами кодов КХФ, но не с самими кристаллохимическими формулами. Главное – открывает ли какие-либо новые возможности использование кода по сравнению с обычным использованием исходных данных. Некоторые, уже найденные возможности, упомянуты выше.

Используя двупараметрические алфавиты можно создавать Базы данных с иерархической, периодической структурой, объединяющие разные виды сведений по социально-экономическим и другим парам свойств сложных систем.

Например:

Алфавит1 – специальности, по которым происходит обучение в вузах; Алфавит2 –специальности, по которым действительно работают их выпускники.

Алфавит1 – национальности; Алфавит2 – профессии, или виды преступлений, или области деятельности за рубежом.

Алфавит1 – страны; Алфавит2 – виды продуктов на экспорт, или области деятельности граждан за рубежом, или области деятельности, за которые жители этой страны удостоены Нобелевских премий.

Символы первого параметра (Алф1) – отвечают дискретным, неизменным за время изучения (фиксации), сущностям. Символы второго (Алф2), также дискретны, но отвечают параметрам переменным по интенсивности, частоте встречаемости.

Ранговые формулы строятся с учетом частот совместных событий. Далее возможны расчеты информационной энтропии и анэнтропии, как мер сложности и чистоты системы, использующихся при изучении процессов эволюции составов систем.

Таким образом, двупараметрические алфавиты применимы для организации информации той, которая изучается статистикой с использованием коэффициентов корреляции, но в которых тонет конкретная информация. Здесь – в базе - она становится легко обозримой и разнообразно осознаваемой, имея в виду, что ее представление открывает возможности использования языка-метода ***RHA***, который и был положен в основу разработки представляемого алфавита.

Предлагаемый метод кодирования кристаллохимических формул, формирование алфавита нового типа, а также построение самих кристаллохимических формул и их упорядочение с получением иерархической классификации ***R***-КХФ, насколько нам известно, не имеет аналогов в существующий литературе.

Расчеты и упорядочение ранговых формул кристаллохимического состава производятся с помощью программного комплекса Petros3, составленного С.В. Мошкиным [27, 28], что обеспечивает высокую скорость и точность обработки материалов.

## СПИСОК ЛИТЕРАТУРЫ


1. Петров Т. Г. Обоснование варианта общей классификации геохимических систем // Вестник ЛГУ. – 1971. – № 18. – С. 30 – 38.
2. Петров Т. Г. Информационный язык для описания составов многокомпонентных объектов // НТИ. Сер. 2. – 2001. – № 3. – С. 8 – 18.
3. Петров Т. Г. Рангово-энтропийный подход к описанию составов геологических объектов и их изменений (на примере геологической ценологии) // Общая и прикладная ценология. – 2007. – № 5. – С. 27 – 33.
4. Краснова Н. И., Петров Т. Г., Ретюнина А. В. Практические аспекты использования метода RHA для классификации минералов группы слюд // Вестник СПбГУ. Сер. 7. – 2008. – № 2. – С. 3 – 19.
5. Петров Т. Г., Фарафонова О. И. Информационно-компонентный анализ. Метод RHA. – СПб. : Изд-во СПбГУ, 2005. – 168 с.
6. Петров Т. Г. Иерархическая периодическая система химических составов и ее связь с периодической системой элементов // Вестник СПбГУ. Сер. 7. – 2009. – Вып. 2. – С. 21 – 28.
7. Петров Т. Г., Краснова Н. И. R-словарь-каталог химических составов минералов. – СПб. : Наука, 2010. – 152 с.
8. Золотарев А. А., Петров Т. Г., Мошкин С. В. Особенности химического состава минералов группы скаполита // Записки ВМО. – 2003. – № 6. – С. 63 – 84.
9. Булах А. Г., Петров Т. Г. Химическое разнообразие минералов группы эвдиалита, ранговые формулы их состава, химические и химико-структурные разновидности // Записки ВМО. – 2003. – № 4. – С. 1 – 17.

10. Bulakh A. G., Petrov T. G. Chemical variability of eudialyte-group minerals and their sorting // Neues Jahrb. Min. – 2004. – № 3. – P. 127 – 144.
11. Андриянец-Буйко А. А., Краснова Н. И., Петров Т. Г. Разнообразие состава турмалинов и их химическая классификация на основе метода RHA // Записки РМО. – 2007. – № 1. – С. 25 – 41.
12. Петров Т. Г., Фарафонова О. И., Соколов П. Б. Информационно-энтропийные характеристики состава минералов и горных пород как отражение напряженности процесса кристаллизации // Записки ВМО. – 2003. – № 2. – С. 33 – 40.
13. Мухамеджанов А. Ф., Петров Т. Г. Рассмотрение дифференцированных интрузий с помощью химико-информационного метода RHA // Структура, вещество, история литосферы Тимано-Северо-Уральского сегмента. – Сыктывкар, 2006. – Вып. 15. – С. 124 – 125.
14. Петров Т. Г. Метод *RHA* как решение проблемы систематизации аналитических данных о вещественном составе геологических объектов // Отечественная геология. – 2008. – № 4. – С. 98 – 105.
15. Кривовичев В. Г., Золотарев А. А. Химический состав минералов и графические способы его изображения. – СПб., 2002. – 84 с.
16. Першиков В. И., Савинков В. М. Толковый словарь по информатике. – М. : Финансы и статистика, 1991. – 540 с.
17. Уникальная иллюстрированная энциклопедия в таблицах и схемах. – М. : АСТ, 2006.
18. Кривовичев В. Г. Минералогический словарь. – СПб. : Изд-во СПбГУ, 2008. – 556 с.
19. Минералогическая энциклопедия / ред. К. Фрей. – Л. : Недра, 1985. – 512 с.
20. Hawthorne F. C., Henry D. J. Classification of the minerals of the tourmaline group // Eur. J. Mineral. – 1999. – Vol. 11. – P. 201 – 215.
21. Рождественская И. В., Франк-Каменецкая О. В., Золотарев А. А., Бронзова Ю. М., Баннова И. И. Уточнение кристаллических структур трех фторсодержащих эльбаитов // Кристаллография. – 2005. – Т. 50, № 6. – С. 981.
22. Салямон Л. С. О некоторых факторах, определяющих восприятие нового слова в науке // Научное открытие и его восприятие / ред. С. Р. Микулинский, М. Г. Ярошевский. – М. : Наука, 1971. – С. 95 – 114.
23. Иванов С. А. 1000 лет озарений. – М. : Слово, 2002. – 220 с.
24. Шурубор Ю. В. Об одном свойстве меры сложности геохимических систем // Доклады АН СССР. – 1972. – Т. 205, № 2. – С. 453 – 456.
25. Петров Т. Г. Проблема разделения и смешения в неорганических системах // Геология / ред. В. Т. Трофимов. – М. : МГУ, 1995. – Т. 2. – С. 181 – 186.
26. Барсуков В. Л., Волосов А. Г., Козеренко С. В., Сущевская Т. М., Баранова Н. Н. Научные основы геохимических методов поисков глубокозалегающих рудных месторождений. – Иркутск, 1970. – С. 79 – 98.
27. Мошкин С. В., Шелемотов А. С., Богачев В. А., Иваников В. В., Петров Т. Г., Филиппов Н. Б., Франк-Каменецкий Д. А. «PETROS» – новый программный комплекс для обработки и анализа петрогеохимической информации // Петрография на рубеже XXI века. Итоги и перспективы. – Сыктывкар, 2000. –Т. 1: Общие проблемы петрографии. – С. 285 – 287.
28. Петров Т. Г., Мошкин С. В. Метод RHA и его реализация в программном комплексе Petros-3 // Вычисления в геологии. – 2011. – №1. – С. 50 – 53.

*Материал поступил в редакцию 29.11.11.*

**Сведения об авторах**

**ПЕТРОВ Томас Георгиевич** – доктор геолого-минералогических наук, профессор, главный научный сотрудник Санкт-Петербургского государственного университета
E-mail: tomas_petrov@rambler.ru

**АНДРИЯНЕЦ-БУЙКО Алиса Александровна** – аспирант Санкт-Петербургского государственного университета
E-mail: a-baa@mail.ru

**МОШКИН Сергей Владимирович** – кандидат геолого-минералогических наук, начальник Отдела ООО "INFOCOMM", Санкт-Петербург
E-mail: svmoshkin@rambler.ru

# АВТОМАТИЗАЦИЯ ОБРАБОТКИ ТЕКСТА

УДК 81’322.3

В. А. Лапшин

# Грамматический анализ языков с неоднозначными лексикой и синтаксисом

*Рассматривается проблема синтаксического анализа языков с неоднозначной лексикой. Предложен алгоритм лексического анализа, позволяющий корректно обрабатывать различного рода неоднозначности. Для рассматриваемого алгоритма лексического анализа предложен алгоритм синтаксического анализа, позволяющий корректно обрабатывать как лексические, так и синтаксические неоднозначности.*

**Ключевые слова:** *лексический анализ, синтаксический анализ, токен Шредингера, контекстно-свободные грамматики, алгоритм Эрли*

## 1. ПОСТАНОВКА ПРОБЛЕМЫ

В данной работе рассматривается проблема синтаксического анализа языков с неоднозначной лексикой. Здесь очевидно подразумевается, что анализ языка разбивается по крайней мере на два этапа:

- Разбиение входного текстового потока на *токены*, представляющие собой строки текста, принадлежащие одному или более типам. Обычно этот этап разбора называют лексическим анализом.
- Синтаксический анализ потока токенов, сконструированного на предыдущем этапе. Синтаксис языков, разбираемых на этом этапе, обычно может быть описан контекстно-свободной грамматикой Хомского [1].

Традиционно лексический анализатор представляется в виде программного модуля, предоставляющего процедуру GetToken, для получения очередного токена из входного потока. Данная процедура выделяет с текущей позиции строку текста и присваивает этой строке некоторый тип (в дальнейшем такие типы мы будем называть лексическими типами). Пара (строка, тип) и представляет токен, выделенный из текста. Таким образом, при однозначном разбиении входного текста на токены синтаксическому анализатору передается на вход поток токенов, выстроенный в линейно упорядоченную последовательность.

Для языков с неоднозначной лексикой ситуация усложняется. Усложнение может происходить в двух направлениях:

- Присваивание более чем одного типа выделенной из символьного потока строке.
- Выделение из символьного потока более чем одной строки.

Первая ситуация часто встречается в программах анализа естественных языков с достаточно высокой флективностью. Рассмотрим, например, фразу русского языка «мама мыла раму». При выделении токенов из текста фразы будут выделены три строки: «мама», «мыла», «раму». Для первой строки будет возвращен токен (мама, существительное)[1]. Но для второй строки будут возвращены два токена: (мыло, существительное) и (мыть, глагол). Здесь проявляется лексическая неоднозначность первого рода, т.е. неоднозначность функции присваивания выделенной из текста строке лексического типа. Эта ситуация схематически изображена на рис. 1.

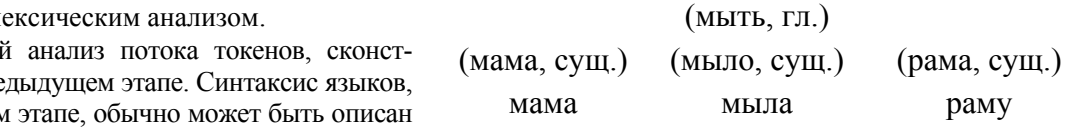


Рис. 1. Результат работы лексического анализатора на тексте «мама мыла раму»

Неоднозначности второго рода часто возникают в искусственных языках. Так, выражение языка программирования C++ «a>>» можно рассматривать двояко:

1) `a` является именем переменной целочисленного типа, `a >>` представляет операцию правого побитового сдвига значения переменной `a`, например, в выражении `a >> 6`.

2) Выражение `>>` представляет собой составное выражение, состоящее из двух подряд идущих операторов `>`, завершающих списки шаблонных параметров. Эта синтаксическая конструкция возникает при объявлении шаблонных параметров шаблона, например, в выражении `template<typename T, template<int a>>`.

В зависимости от синтаксического контекста, каждой из описанных ситуаций соответствуют различные способы разделения текста программы на строки. В первом случае будет выделен поток токенов:

[1] Предполагается, что в качестве лексических типов здесь выступают части речи русского языка. Для упрощения изложения другие грамматические атрибуты слов не рассматриваются.

(a, целочисленная переменная), (>>, оператор правого сдвига). Во втором случае будет выделен поток токенов: (a, целочисленная переменная), (>, конец списка шаблонных параметров), (>, конец списка шаблонных параметров). Выделение потока токенов схематически изображено на рис. 2.

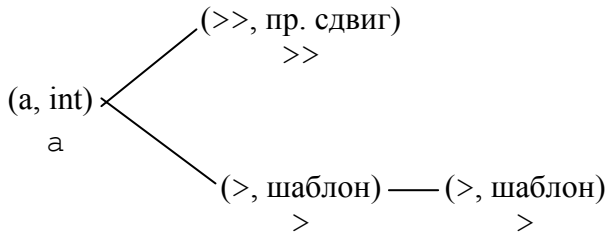


Рис. 2. Результат работы лексического анализатора на выражении a>>

Наибольшие трудности возникают при анализе языков, в которых неоднозначности первого и второго родов смешаны друг с другом. При выделении строк из символьного потока различным образом этим строкам присваивается также более одного лексического типа. В этом случае результатом работы лексического анализатора является не поток токенов (может быть, где-то образующий «петли» в случае неоднозначности), но древовидная структура, в которой дочерние узлы представляют собой токены, идущие за токеном, представляющим в дереве родительский узел. Вершиной дерева можно считать псевдотокен, соответствующий началу чтения символьного потока.

Возникает законный вопрос: может ли вообще иметь право на существование, т.е. быть осознан носителями, язык с настолько запутанной лексикой, что лексический анализатор будет генерировать такие ветвистые древовидные структуры для текстов этого языка? Ответ на этот вопрос положительный. Примером такого языка может быть любой язык с открытой лексикой. Такие языки возникают, в частности, в системах коллективного построения онтологий с открытым языком представлений и запросов. Так, в системе ЭЗОП [2,3] множество лексических типов описывается регулярными выражениями, на основе которых конструируются конечные автоматы, используемые для выделения токенов из символьного потока. Поэтому на этапе написания алгоритма лексического анализа множество лексических типов неизвестно или до конца не определено. Конечные автоматы лексических типов могут выделять строки из символьного потока совершенно произвольно, следовательно, результат исполнения алгоритма лексического анализа в общем случае будет представляться древовидной структурой.

В данной статье будут рассмотрены способы организации работы синтаксического анализатора с такими древовидными структурами. Будет представлен алгоритм синтаксического анализа, который позволяет разрешать лексическую неоднозначность на синтаксическом уровне и представлять результаты, даже если лексическая неоднозначность не была разрешена. Алгоритм обладает хорошей производительностью и может быть применим также для анализа естественных флективных языков, в которых лексическая омонимия является ординарным явлением.

## 2. МЕТОДЫ РАБОТЫ С ЛЕКСИЧЕСКОЙ НЕОДНОЗНАЧНОСТЬЮ

Обычно лексическую неоднозначность пытаются разрешить на уровне лексического же анализа. Достаточно популярны методы разрешения лексической омонимии на основе статистики встречаемости словоформ друг с другом в уже имеющихся текстах данного языка. На основе такой статистики строят вероятностные гипотезы о принадлежности выделенной из текста строки тому или иному лексическому типу. Статистика обычно подсчитывается на основе контекстов соседних словоформ. На основании этой статистики и контекста нахождения словоформы среди соседних словоформ в тексте создается гипотеза о том, какой токен представляет данная словоформа. Изучению этого направления посвящено достаточно много работ, из последних следует упомянуть публикации [4-6].

Конечно, статистические методы разрешения неоднозначности в их традиционном применении обращены прежде всего на неоднозначности первого рода. Неоднозначности второго рода, хотя и могут быть разрешены на основе вероятностных гипотез, тем не менее часто требуют более точных методов. Так, для выражения a>> из примера, приведенного в предыдущем разделе, необходимо разрешать лексическую омонимию на основе синтаксического контекста, т.е. на основе точных правил грамматики разбираемого языка. В этом смысле можно сказать, что вероятностные методы разрешения лексической неоднозначности приносят точность разрешения на алтарь производительности алгоритма грамматического анализа, тогда как методы разрешения омонимии на основе синтаксического контекста требуют более обстоятельного и потому медленного рассмотрения. В языках программирования, у которых лексические и синтаксические неоднозначности носят характер скорее исключения, нежели правила, резонно передать ответственность за разрешение омонимии на уровень синтаксического анализа.

Традиционно лексический и синтаксический анализаторы взаимодействуют друг с другом на основе конвейерной модели: лексический анализатор читает символьный поток и выдает синтаксическому анализатору поток токенов. Неоднозначности обычно разрешаются на уровне лексического анализа, поэтому результатом работы лексического анализатора является цепочка токенов, рассматриваемая синтаксическим анализатором как *поток*, т.е. последовательность токенов, которая может быть прочитана только в одном направлении (слева направо). Это существенно упрощает[2] жизнь синтаксическому анализатору, вычислительную слож-

[2] В смысле производительности алгоритма синтаксического анализа. Если алгоритмы лексического анализа обычно имеют линейную сложность относительно длины входного текста, выраженной в символах, то алгоритмы синтаксического анализа часто имеют экспоненциальную сложность относительно длины входного потока токенов.

ность алгоритма которого обычно стараются сделать хотя бы квадратичной от длины потока токенов, выдаваемых лексическим анализатором.

Таким образом, взаимодействие лексического и синтаксического модулей грамматического анализа строго направлено от лексического анализатора к синтаксическому. Выражение `a>>` в этом смысле не вписывается в традиционную модель: необходимо передать информацию о синтаксическом контексте в лексический анализатор от синтаксического. Лексическому анализатору необходимо знать, читает ли он сейчас список параметров шаблона или арифметическое выражение, чтобы разрешить лексическую омонимию. В данном конкретном случае синтаксический контекст легко передать, устанавливая в синтаксическом анализаторе глобальный флаг синтаксического контекста, который может быть прочитан лексическим анализатором. Но что делать, если язык таков, что подобные случаи являются достаточно регулярным явлением?

В работе [7] была предложена модель так называемого *токена Шредингера*[3]. В этой модели, так же, как в мысленном эксперименте Шредингера, при анализе выражения >> на уровне лексического анализатора не представляется возможным, не используя синтаксический контекст, решить, какой токен необходимо возвратить при чтении строки >>. Поэтому было предложено возвратить токен, который бы представлял собой суперпозицию токенов (>>, пр. сдвиг) и (>, шаблон), (>, шаблон). В этом случае синтаксический анализатор оперировал бы токеном, содержащим в себе композицию трех токенов, и, в соответствии с текущим синтаксическим контекстом, в необходимый момент выбирал бы необходимый.

Предложенная модель достаточно оригинальна, но доставляет некоторые неудобства. Дело в том, что позиции токенов, входящих в токен Шредингера, могут не совпадать. В данном примере получилось так, что токен (>>, пр. сдвиг) и пара токенов (>, шаблон), (>, шаблон) представляют одну и ту же строку >>. Но в других языках такого «равномерного нарезания» входного текста на строки и содержащиеся в них токены может и не быть. В этом случае модель токена Шредингера не подходит. В качестве альтернативы, обобщающей модель токена Шредингера, автор предлагает метод, описываемый далее.

Заметим, что при возникновении неоднозначности, которая не может быть разрешена на лексическом уровне, разрешение ее делегируется на синтаксический уровень. Предполагается, что эта неоднозначность будет разрешена синтаксическим анализатором сразу же, чтобы токен, следующий за токеном Шредингера, попал в корректный синтаксический контекст. Но такое разрешение «на месте» не всегда возможно. Грамматика языка может быть составлена таким образом, что разрешение омонимии может быть сделано только после чтения еще нескольких токенов. Все это время синтаксический анализатор вынужден работать с суперпозицией альтернативных, а также идущих за этими альтернативами токенов. Таким образом, «разрешение суперпозиции состояний» откладывается на некоторый срок и синтаксический анализатор продолжает работу одновременно в нескольких состояниях. Это требует модификации работы грамматического анализатора как на лексическом, так и на синтаксическом уровнях.

Рассмотрим, что необходимо модифицировать на лексическом уровне. Прежде всего, необходимо обеспечить последовательность токенов в случае возникновения альтернатив как первого, так и второго рода. Для этого предлагается расширить данные токена еще одним компонентом – позицией окончания строки токена в символьном потоке, обрабатываемом лексическим анализатором. В этом случае токен будет представлять собой тройку (позиция, строка, лексический тип). Во-вторых, необходимо обеспечить возможность чтения данных символьного потока с любой позиции, чтобы была возможность прочитать токен, идущий за данным, даже если его альтернатива выделила строку в символьном потоке где-то впереди. При соблюдении этих двух условий лексический анализатор может предоставить в качестве интерфейса процедуру GetNextToken, которая будет брать в качестве параметра токен и возвращать множество токенов, выделенных с позиции, следующей за той, на которой заканчивается переданный в качестве параметра токен. Конечно, в этом случае основная ответственность по обработке альтернатив ложится на синтаксический анализатор. Необходимо подобрать алгоритм, который бы хорошо обрабатывал лексические неоднозначности и мог также обрабатывать синтаксические неоднозначности, которые неизбежно будут возникать при обработке такого рода языков. В данной работе автор описывает такой алгоритм синтаксического анализа, модифицированный специально, чтобы обрабатывать описанные выше лексические и синтаксические неоднозначности.

## 3. МОДИФИЦИРОВАННЫЙ АЛГОРИТМ ЭРЛИ

В качестве алгоритма синтаксического анализа, призванного эффективно работать с лексическими и синтаксическими неоднозначностями, предлагается использовать специальным образом модифицированный алгоритм Эрли [8]. Опишем этот алгоритм.

### 3.1. Описание алгоритма Эрли

Алгоритм Эрли представляет собой алгоритм синтаксического анализа языков, описанных контекстно-свободной порождающей грамматикой Хомского, который производит синтаксический анализ, комбинируя методы «сверху-вниз» и «снизу-вверх», а так-

[3] Название навеяно известным мысленным экспериментом, предложенным Эрвином Шредингером, австрийским физиком-теоретиком, одним из создателей квантовой механики. Шредингер хотел показать неполноту квантовой механики при переходе от субатомных систем к макроскопическим. Эксперимент описывается следующим образом. В закрытой комнате заперты кот и емкость с ядовитым газом. С вероятностью 0.5 емкость в ближайшую секунду разобьется и кот умрет. В квантовой механике вынуждены оперировать суперпозицией двух состояний, т.е. рассматривать кота одновременно «живым и мертвым». Вопрос Шредингера заключался в следующем: когда перестает существовать смешение двух состояний и выбирается только одно? Целью эксперимента являлась иллюстрация того, что квантовая механика неполна без правил, показывающих, при каких условиях происходит переход от суперпозиции состояний к одному состоянию.

же используя принцип динамического программирования, при котором синтаксический анализ текущей позиции основан на результатах построений алгоритма на предыдущих позициях.

Алгоритм Эрли оперирует так называемыми *ситуациями Эрли*. Ситуация Эрли (далее просто «ситуация») представляет собой пару [помеченное правило; номер позиции токена]. Помеченное правило – это правило контекстно-свободной грамматики Хомского с точкой в правой части. Например, для правила $S \rightarrow Abc$ возможны следующие помеченные правила[4]: $S \rightarrow \bullet Abc$, $S \rightarrow A \bullet bc$, $S \rightarrow Ab \bullet c$ и $S \rightarrow Abc \bullet$.

Для каждой позиции в потоке токенов, выдаваемых лексическим анализатором[5], включая нулевую позицию, когда ни одного токена еще не выдано, алгоритм Эрли строит *состояния Эрли*. Состояние Эрли – это просто множество ситуаций Эрли, т.е. список неповторяющихся ситуаций. Состояния Эрли строятся с помощью трех операций, которые применяются ко всем ситуациям в состоянии: Predictor, Completer и Scanner. Опишем эти операции.

Операция Predictor применяется к ситуациям Эрли вида $[A \rightarrow \ldots \bullet X \ldots; i]$, где $X$ – нетерминальный символ грамматики, i – номер состояния (позиция токена, для которого построено это состояние). Для каждого такого нетерминала $X$ Predictor ищет в грамматике правила вида $X \rightarrow \ldots$ с этим нетерминалом в левой части и добавляет в состояние ситуацию вида $[X \rightarrow \bullet \ldots; j]$ с точкой в начале правой части и текущим номером состояния j. Смысл этой операции состоит в «предсказании» того, какие правила могут участвовать в синтаксическом анализе далее.

Операция Completer применяется к ситуациям вида $[A \rightarrow \ldots \bullet; i]$, у которых точка стоит в крайней правой позиции правой части правила. Задача операции состоит в том, чтобы просканировать состояние под номером i, найти в нем ситуации вида $[B \rightarrow \ldots \bullet A \ldots; j]$ с точкой перед нетерминалом $A$ и добавить в текущее состояние ситуацию $[B \rightarrow \ldots A \bullet \ldots; j]$, передвинув точку через символ $A$.

Наконец, операция Scanner позволяет добавить в новое состояние те ситуации, которые подходят для текущего токена в символьном потоке. Лексический тип токена в контекстно-свободной грамматике представляется терминальным символом. Поэтому операцию Scanner можно описать следующим образом. Для терминального символа $a$, для которого строится состояние номер i+1, операция сканирует состояние номер i и для каждой ситуации вида $[A \rightarrow \ldots \bullet a \ldots; j]$ с точкой перед символом $a$ добавляет в состояние номер i+1 ситуацию $[A \rightarrow \ldots a \bullet \ldots; j]$, передвигая, таким образом, точку через символ $a$.

Алгоритм Эрли работает следующим образом:

1. Создаем состояние номер 0 и для каждого правила вида $S \rightarrow \ldots$ с начальным нетерминалом $S$ в левой части добавляем в это состояние ситуации вида $[S \rightarrow \bullet \ldots, 0]$, после чего применяем ко всем ситуациям состояния операцию Predictor до тех пор, пока в это состояние что-нибудь добавляется.

2. Для каждого токена, выдаваемого лексическим анализатором, создаем новое состояние с номером позиции этого токена в потоке. Запускаем операцию Scanner на предыдущем состоянии, после чего запускаем поочередно операции Predictor и Completer на всех ситуациях состояния до тех пор, пока в состояние добавляются новые ситуации.

Последовательность токенов считается распознанной, если в состоянии, соответствующем последнему токену, имеется ситуация вида $[S \rightarrow \ldots \bullet; n]$ с точкой в конце правой части. Это означает, что последовательность токенов, выдаваемая лексическим анализатором и рассматриваемая на уровне синтаксического анализатора как последовательность лексических типов, интерпретируемых как терминальные символы грамматики, была распознана правилом с начальным нетерминалом в левой части. Это эквивалентно тому факту, что для данной строки терминальных символов построено порождение из начального нетерминала, т.е. принадлежности этой строки языку данной грамматики.

Приведем иллюстративный пример. Пусть имеется грамматика для языка $L$ арифметических выражений с операциями умножения и сложения. Для всех чисел используется лексический тип $n$, для операции умножения – лексический тип *, для операции сложения – лексический тип +. Таким образом, символьный поток «12+45*34» будет преобразован в поток лексических типов «$n+n*n$».

Контекстно-свободная грамматика для этого языка будет иметь следующие правила: $A \rightarrow M+A$, $A \rightarrow M$, $M \rightarrow n*M$, $M \rightarrow n$. Здесь в качестве начального нетерминала выступает символ $A$. Рассмотрим строку лексических типов «$n+n$», принадлежащую данному языку, и проиллюстрируем на этой строке работу алгоритма Эрли.

Работа алгоритма начинается с создания нулевого состояния и добавления в него всех ситуаций с начальным нетерминалом в левой части:

$[A \rightarrow \bullet M+A; 0]$

$[A \rightarrow \bullet M; 0]$

Запускаем теперь операцию Predictor на ситуациях этого состояния. Обе ситуации содержат символ $M$ после точки, поэтому добавляем в состояние ситуации, полученные из правил, содержащих символ $M$ в левой части. Имеем:

$[A \rightarrow \bullet M+A; 0]$

$[A \rightarrow \bullet M; 0]$

$[M \rightarrow \bullet n*M; 0]$

$[M \rightarrow \bullet n; 0]$

Больше добавлять нечего, поэтому читаем первый символ из потока токенов – токен с лексическим типом $n$. Осуществляем операцию Scanner для состояния номер 0. С точкой перед символом $n$ имеется только одна ситуация – $[M \rightarrow \bullet n; 0]$. Добавляем ее в состояние 1, предварительно передвинув точку на символ правее:

$[M \rightarrow n \bullet; 0]$

Теперь необходимо применять последовательно операции Predictor и Completer до тех пор, пока в состояние будут добавляться новые ситуации. Сейчас

[4] Точка в правой части правила обозначается символом «•».

[5] Оригинальный алгоритм Эрли не работает с лексическими неоднозначностями, поэтому токены, выдаваемые лексическим анализатором, представляют собой линейно упорядоченную последовательность.

можно применить операцию Completer. Сканируем ситуации состояния 0 и ищем среди них те, у которых точка находится перед символом *M*. Добавляем такие ситуации в состояние 1, передвинув точку через символ. Получаем:

[*M*→*n*•; 0]
[*A*→*M*•+*A*; 0]
[*A*→*M*•; 0]

Ситуацию [*A*→*M*•; 0] обрабатываем операцией Completer. Для этого ищем в состоянии 0 все ситуации с точкой перед символом *A* и добавляем их в состояние 1. Но таких ситуаций нет, поэтому обработка данной ситуации закончена.

Берем следующий символ из потока токенов. Это символ +. Запускаем операцию Scanner, она выбирает ситуацию [*A*→*M*•+*A*; 0] и добавляет ее в состояние 2:

[*A*→*M*+•*A*; 0]

Запускаем теперь поочередно операции Predictor и Completer. Predictor обрабатывает состояние [*A*→*M*+•*A*; 0] и добавляет ситуации с правилами с символом *A* в левой части:

[*A*→*M*+•*A*; 0]
[*A*→•*M*+*A*; 2]
[*A*→•*M*; 2]

Далее Predictor делает это же для символа *M*:

[*A*→*M*+•*A*; 0]
[*A*→•*M*+*A*; 2]
[*A*→•*M*; 2]
[*M*→•*n**M*; 2]
[*M*→•*n*; 2]

Больше обрабатывать нечего, берем очередной символ из входного потока. Это символ *n*. Обрабатываем его и получаем в конце концов состояние 3:

[*M*→*n*•**M*; 2]
[*M*→*n*•; 2]
[*A*→*M*•+*A*; 2]
[*A*→*M*•; 2]
[*A*→*M*+*A*•; 0]

Это состояние содержит ситуацию [*A*→*M*+*A*•; 0], правило которой содержит в левой части начальный нетерминальный символ, а точка стоит в конце правой части. Кроме того, индекс ситуации равен нулю. Все это означает, что строка «*n*+*n*» была распознана алгоритмом Эрли как принадлежащая языку *L*, что и должно было произойти.

### 3.2. Алгоритм Эрли, модифицированный для анализа синтаксических неоднозначностей

Алгоритм Эрли работет со всеми типами контекстно-свободных грамматик, включая неоднозначные различной степени сложности. Но алгоритм только отвечает на вопрос о том, принадлежит ли переданная на вход цепочка терминалов (лексических типов) данному языку, но не строит порождения этой цепочки в данной грамматике. Традиционно алгоритмы синтаксического анализа строят *деревья порождений* входных терминальных строк. На вершине такого дерева всегда стоит начальный нетерминальный символ грамматики, а листы дерева (узлы, не имеющие дочерних) помечены терминальными символами. Список таких терминальных узлов образует входную цепочку, если обойти дерево порождения в порядке слева направо и снизу вверх.

В работе [9] было предложено модифицировать алгоритм Эрли таким образом, чтобы строить по ходу синтаксического анализа все деревья порождений, которые можно построить для данной входной строки терминальных символов грамматики. Для этого в каждую ситуацию добавляются еще два компонента:

1. Компонент lp, содержащий идентификатор ситуации с теми же компонентами, у которой точка стоит на символ левее. Такие компоненты появляются при добавлении ситуаций в состояние операциями Scanner и Completer.

2. Компонент tp, содержащий список идентификаторов ситуаций с точкой в конце правой части, нетерминал в левой части правила которой стоит перед точкой в данной ситуации. Идентификатор, очевидно, добавляется в список только в результате сдвига точки через нетерминальный символ в результате применения операции Completer.

Таким образом, ситуация Эрли в модифицированном алгоритме будет иметь следующий вид: [*A*→*…*•*…;i;lp;(tp*1,*tp*2,*…,tpn)*]. В списке tp будет более одного компонента только при появлении синтаксических неоднозначностей для данной входной строки.

В результате работы модифицированного алгоритма Эрли для каждой позиции терминального символа входной строки, включая нулевую позицию, строятся состояния Эрли, содержащие модифицированные ситуации. И по-прежнему входная строка считается принадлежащей языку, если состояние последнего символа входной цепочки содержит ситуацию вида [*S*→*…*•;0*;lp;(tp*1,*tp*2,*…,tpn)*], где *S* является начальным нетерминалом грамматики.

Деревья порождений упакованы в модифицированных ситуациях Эрли, и для извлечения всевозможных деревьев из состояний необходимо после завершения алгоритма распознавания пройти по всем ситуациям в состояниях, начиная с ситуаций вида [*S*→*…*•;0*;lp;(tp*1,*tp*2,*…,tpn)*], находящихся в состоянии, соответствующем последнему символу входной строки. Каждая такая ситуация соответствует вершине дерева порождения. Для перехода к дочерним узлам дерева следует пройти по состояниям, соответствующим идентификаторам, хранящимся в tp. Такие переходы будут равнозначны проходу по дочернему уровню дерева порождения. Для перехода к уровням ниже необходимо использовать состояния, идентификаторы которых хранятся в списке tp. Если в этом списке более одного идентификатора, то это означает, что в процессе анализа входной строки возникла синтаксическая неоднозначность. В работе [9] эта неоднозначность разрешалась копированием уже построенной части дерева порождения. Таким образом, в результате работы алгоритма построения дерева на основе списка состояний Эрли, построенных модифицированным алгоритмом Эрли, конструировалось столько деревьев, сколько было необходимо для разрешения возникших синтаксических неоднозначностей.

Описанный алгоритм работал с цепочкой терминальных символов, следовательно, не мог корректно обрабатывать лексические неоднозначности. В следующем разделе описывается алгоритм, основанный на похожих принципах, но дающий возможность производить синтаксический анализ с неоднозначностями как на лексическом, так и на синтаксическом уровнях.

## 4. АЛГОРИТМ ЭРЛИ, МОДИФИЦИРОВАННЫЙ ДЛЯ АНАЛИЗА ЛЕКСИЧЕСКИХ И СИНТАКСИЧЕСКИХ НЕОДНОЗНАЧНОСТЕЙ

В данном разделе описывается алгоритм Эрли, модифицированный для обработки неоднозначностей, возникающих на лексическом и синтаксическом уровнях грамматического анализа. Как уже говорилось выше, классический алгоритм Эрли, предложенный в работе [8], оперирует в качестве входа со строкой терминальных символов, т.е. никакой неоднозначности при лексическом анализе не предполагается. Для описания работы алгоритма при наличии лексических неоднозначностей необходимо определить интерфейс взаимодействия между модулями лексического анализа и синтаксического анализа таким образом, чтобы корректно обрабатывать лексические неоднозначности.

Для этого определим основную процедуру взаимодействия следующим образом:

TokenList GetNextTokens(Token).

Эта процедура предоставляется лексическим анализатором и вызывается модулем синтаксического анализа всякий раз, когда необходимо получить токен (или несколько токенов, в случае лексической неоднозначности), следующий за токеном, переданным в качестве входного параметра процедуры.

Как уже говорилось выше, токен представляет собой тройку вида (выделенная строка, позиция конца строки, лексический тип). При передаче в процедуру GetNextTokens токена Token лексический анализатор выделит множество токенов с позиции, соответствующей концу строки, хранящейся в переданном токене. Каждый из этих токенов будет обработан в модуле синтаксического анализатора, построенном на основе описываемого модифицированного алгоритма Эрли. Для того чтобы это сделать корректно, необходимо также модифицировать алгоритм синтаксического анализа так, чтобы он был в состоянии правильно обрабатывать множество лексических типов, возвращаемых процедурой GetNextTokens.

Для корректной обработки лексических неоднозначностей введем в алгоритм Эрли модуль под названием «диспетчер состояний». Функции этого модуля будут заключаться в хранении состояний в виде словаря, в качестве ключа которого будет выступать уникальный идентификатор состояния, генерируемый диспетчером состояний при добавлении в него нового состояния. Также диспетчер состояний будет предоставлять сервис быстрого поиска состояния по переданному идентификатору. Предположим также, что каждая ситуация имеет в состоянии, которому она принадлежит, свой уникальный номер, начиная с единицы. Тогда каждая ситуация однозначно идентифицируется парой (идентификатор состояния, идентификатор ситуации).

Таким образом, состояния будут представлять собой пары (токен, список ситуаций), а ситуации в состояниях будут идентифицироваться парами (идентификатор состояния, идентификатор ситуации), т.е. иметь вид $[A\rightarrow\ldots\bullet\ldots;i;lp;(tp1,tp2,\ldots,tpn)]$, где lp и все tp являются такими идентифицирующими парами.

Другая особенность модифицированного алгоритма состоит в том, что процедура GetNextTokens может возвращать более одного токена. В этом случае необходимо запустить операцию Scanner для каждого токена в возвращенном списке и, если это необходимо, создать соответствующие состояния. Все такие новые состояния должны быть обработаны поочередными вызовами операций Predictor и Completer, после чего состояния должны быть добавлены в список для получения токенов, следующих за токенами состояний. Таким образом, алгоритм оперирует не одним текущим состоянием, над которым проводятся все операции, но множеством текущих состояний, которое в дальнейшем будем называть поколением обработки. В результате обработки текущего поколения создается следующее поколение. Этот процесс продолжается до тех пор, пока в следующем поколении появляется по крайней мере одно состояние.

Приведем определения процедур Predictor, Scanner и Completer.

Процедура **Predictor** берет на вход некоторую ситуацию с точкой перед нетерминальным символом и состояние S[i], в которое добавляет новые ситуации. Для переданной на вход ситуации вида $[A\rightarrow\ldots\bullet B\ldots;j;lp;(tp1,tp2,\ldots,tpn)]$, в которой точка стоит перед нетерминальным символом $B$, процедура добавляет в состояние S[i] ситуации вида $[B\rightarrow\bullet\ldots;i;(0,0);()]$. Иначе говоря, ситуации добавляются в текущее состояние с идентификатором i, в качестве lp выступает пара (0,0), обозначающая пустую ссылку на состояние, список tp пуст.

Процедура **Scanner** берет на вход токен Token=(выделенная строка, позиция n, лексический тип t), состояние S[i] для обработки и состояние S[n] для добавления. Процедура сканирует ситуации состояния S[i] и для каждой ситуации вида $[A\rightarrow\ldots\bullet t\ldots;j;lp;(tp1,tp2,\ldots,tpn)]$ под номером k добавляет в состояние S[n] ситуацию вида $[A\rightarrow\ldots t\bullet\ldots;j;(i,k);()]$. Компонент lp содержит пару (i,k), т.е. ссылку на ситуацию $[A\rightarrow\ldots\bullet t\ldots;j;lp;(tp1,tp2,\ldots,tpn)]$, переданную для обработки, а список tp – пустой.

Процедура **Completer** берет на вход ситуацию вида $[A\rightarrow\ldots\bullet;j;lp;(tp1,tp2,\ldots,tpn)]$ с номером k в состоянии S[i] и сканирует состояние S[j] с идеентификатором j. Для каждой ситуации вида $[B\rightarrow\ldots\bullet A\ldots;m;lp;(tp1,tp2,\ldots,tpn)]$ с номером p процедура добавляет в состояние S[i] ситуацию вида $[B\rightarrow\ldots A\bullet\ldots;m;(j,p);((i,k))]$. Если ситуация с компонентами $[B\rightarrow\ldots A\bullet\ldots;m;(j,p);(\ldots)]$ уже имеется в состоянии S[i], то ссылка (i,k) просто добавляется в список tp.

Приведем теперь описание основного алгоритма в псевдокоде:

```
L - лексический анализатор.
D - диспетчер состояний.
Q - множество идентификаторов состояний.

S[0] = D.AddState(0);

Заполняем состояние S[0] ситуациями с правилами, где слева
стоит начальный нетерминал грамматики.

while (S[0].BeenChanged()) {
  Predictor(S[0]);
}

Q.AddId(0);
while (not Q.empty()) {
  P - множество идентификаторов состояний.
  foreach i in Q {
    S[i] = D.GetState(i);
    TokenList tl =
L.GetNextTokens(S[i].Token);
    foreach token in tl {
      S[j] = D.AddState(token);
      Scanner(token, S[i], S[j]);
      while (S[j].BeenChanged()) {
        Predictor(S[j]);
        Completer(S[j]);
      }
      if (not S[j].Empty()) {
        P.AddId(j);
      }
    }
  }
  Q=P;
}
```

Как уже указывалось выше, поток токенов, выдаваемых лексическим анализатором, образует дерево, вершиной которого является специальный токен, находящийся в состоянии S[0], у которого позиция конца равна нулю. Листами дерева являются те токены, для которых процедура GetNextTokens возвратила пустой список токенов. Состояния таких листовых токенов необходимо запомнить, чтобы после окончания работы основного алгоритма просканировать на предмет нахождения в них ситуаций вида [*S*→…•;0;…;(…)]. Каждая такая ситуация представляет вершину дерева порождения цепочки лексических типов, составленных из кроны этого дерева.

Легко видеть, что алгоритм для каждой цепочки лексических типов от вершины дерева до любого его листа работает совершенно аналогично модифицированному алгоритму Эрли, рассмотренному в предыдущем разделе. Поэтому корректность работы рассматриваемого алгоритма основывается на корректности работы алгоритма Эрли, модифицированного для синтаксических неоднозначностей. Корректность последнего алгоритма была доказана в работе [9]. Следовательно, корректность работы данного алгоритма также не должна вызывать сомнений.

Рассмотрим пример работы алгоритма для языка с лексической и синтаксической неоднозначностями. Пусть язык *L* содержит грамматические конструкции, содержащие арифметические выражения с операцией сложения (+) вещественных чисел в записи с десятичной точкой. В дополнение к этому, пусть язык *L* содержит конструкцию комментариев, начинающихся и заканчивающихся символами «+».

Контекстно-свободная грамматика для этого языка будет содержать следующие лексические типы:

- Операция сложения с единственной цепочкой – строкой «+». Лексический тип обозначим через «+».
- Язык вещественных чисел в записи с десятичной точкой. Лексический тип обозначим через «n».
- Комментарий – строка между парными символами «+». Лексический тип обозначим через «c».

Грамматика языка *L* содержит следующие правила:

*A*→*n*+*A*
*A*→*n*
*A*→*cn*+*A*
*A*→*nc*+*A*
*A*→*n*+*cA*
*A*→*n*+*Ac*
*A*→*cn*
*A*→*nc*

Начальный и единственный нетерминал грамматики – это *A*. Правила грамматики перечисляют все возможные случаи вхождения комментариев в выражения (для упрощения изложения будем считать, что входная строка не может иметь более одного комментария). Пусть теперь на вход подается символьный поток «12.3+3.5+.98». Продемонстрируем работу анализатора на этой строке.

Сначала создадим состояние 0:

1. [*A*→•*n*+*A*;0;;()]
2. [*A*→•*n*;0;();]
3. [*A*→•c*n*+*A*;0;;()]
4. [*A*→•*n*c+*A*;0;;()]
5. [*A*→•*n*+c*A*;0;;()]
6. [*A*→•*n*+*A*c;0;;()]
7. [*A*→•c*n*;0;;()]
8. [*A*→•*n*c;0;;()]

Вызываем GetNextTokens(0). Процедура возвращает только один токен: («n»,n, 5). Запускаем на нем операцию Stamper и добавляем в состояние 1 все ситуации с переходом через n.

1. [*A*→*n*•+*A*;0;0.1;()]
2. [*A*→*n*•;0;0.2;()]
3. [*A*→*n*•c+*A*;0;0.4;()]
4. [*A*→*n*•+c*A*;0;0.5;()]
5. [*A*→*n*•+*A*c;0;0.6;()]
6. [*A*→*n*•c;0;0.8;()]

Операции Predictor и Completer, запущенные на состоянии 1, не добавляют ничего в данное состояние, поэтому снова запускаем процедуру GetNextTokens(5). Процедура возвращает два токена: («+»,+, 6) и («+3.5+»,c, 10). Обрабатываем сначала токен («+»,+, 6), создаем для него состояние 2 и запускаем Scanner на состоянии 1. Имеем:

1. [*A*→*n*+•*A*;0;0.1;()]
2. [*A*→*n*+•*A*c;0;0.6;()]

Запуск операции Predictor добавляет еще несколько ситуаций:

1. [*A*→*n*+•*A*;0;1.1;()]
2. [*A*→*n*+•*A*c;0;0.6;()]
3. [*A*→•*n*+*A*;2;;()]
4. [*A*→•*n*;2;;()]
5. [*A*→•c*n*+*A*;2;;()]
6. [*A*→•*n*c+*A*;2;;()]

7. [*A*→•*n*+c*A*;2;;()]
8. [*A*→•*n*+*A*c;2;;()]
9. [*A*→•c*n*;2;;()]
10. [*A*→•*n*c;2;;()]

Больше добавлять нечего, поэтому переходим к обработке второго токена («+3.5+»,c, 10). Добавляем состояние 3 и проходим по состоянию 1 операцией Scanner. Имеем:

1. [*A*→*n*c•+*A*;0;1.3;()]
2. [*A*→*n*c•;0;1.6;()]

Запуск операции Completer на ситуации 2 не добавляет новых ситуаций в состояние. Поэтому запускаем новую итерацию на состояниях 2 и 3.

Вызываем процедуру GetNextTokens(6), т.е. для токена («+»,+, 6), и добавляем состояние 4. Лексический анализатор возвращает один токен («3.5»,n, 9). Проходим процедурой Scanner по состоянию 2, имеем:

1. [*A*→*n*•+*A*;2;2.3;()]
2. [*A*→*n*•;2;2.4;()]
3. [*A*→*n*•c+*A*;2;2.6;()]
4. [*A*→*n*•+c*A*;2;2.7;()]
5. [*A*→*n*•+*A*c;2;2.8;()]
6. [*A*→*n*•c;2;2.10;()]

Запуск процедуры Completer добавляет еще две ситуации. В результате имеем:

1. [*A*→*n*•+*A*;2;2.3;()]
2. [*A*→*n*•;2;2.4;()]
3. [*A*→*n*•c+*A*;2;2.6;()]
4. [*A*→*n*•+c*A*;2;2.7;()]
5. [*A*→*n*•+*A*c;2;2.8;()]
6. [*A*→*n*•c;2;2.10;()]
7. [*A*→*n*+*A*•;0;2.1;()]
8. [*A*→*n*+*A*•c;0;2.2;(4.2)]

Вызываем процедуру GetNextTokens(10), т.е. для токена («+3.5+»,c, 10), и добавляем состояние 5. Лексический анализатор возвращает один токен («.98»,n, 14). Запускаем процедуру Scanner на состоянии 3, в результате не добавляется ни одной ситуации, т.е. эта ветка лексического анализа кончилась. Весь символьный поток не разобран, поэтому ветка считается неразобранной.

Вызываем теперь процедуру GetNextTokens(9), т.е. для токена («3.5»,n, 9). Лексический анализатор возвращает один токен («+»,+, 10), добавляем для него состояние 5 (этот номер теперь свободен) и запускаем процедуру Scanner на состоянии 4, имеем:

1. [*A*→*n*+•*A*;2;4.1;()]
2. [*A*→*n*+•c*A*;2;4.4;()]
3. [*A*→*n*+•*A*c;2;4.5;()]

Запуск процедуры Predictor добавляет в состояние 5 еще несколько ситуаций:

1. [*A*→*n*+•*A*;2;4.1;()]
2. [*A*→*n*+•c*A*;2;4.4;()]
3. [*A*→*n*+•*A*c;2;4.5;()]
4. [*A*→•*n*+*A*;5;;()]
5. [*A*→•*n*;5;;()]
6. [*A*→•c*n*+*A*;5;;()]
7. [*A*→•*n*c+*A*;5;;()]
8. [*A*→•*n*+c*A*;5;;()]
9. [*A*→•*n*+*A*c;5;;()]
10. [*A*→•c*n*;5;;()]
11. [*A*→•*n*c;5;;()]

Вызываем теперь GetNextTokens(10), возвращается токен («.98»,n, 13). Добавляем для этого токена состояние 6 и запускаем Scanner на состоянии 5, имеем:

1. [*A*→*n*•+*A*;5;5.4;()]
2. [*A*→*n*•;5;5.5;()]
3. [*A*→*n*•c+*A*;5;5.7;()]
4. [*A*→*n*•+c*A*;5;5.8;()]
5. [*A*→*n*•+*A*c;5;5.9;()]
6. [*A*→*n*•c;5;5.11;()]

Запускаем Completer на ситуации 6.2, имеем:

1. [*A*→*n*•+*A*;5;5.4;()]
2. [*A*→*n*•;5;5.5;()]
3. [*A*→*n*•c+*A*;5;5.7;()]
4. [*A*→*n*•+c*A*;5;5.8;()]
5. [*A*→*n*•+*A*c;5;5.9;()]
6. [*A*→*n*•c;5;5.11;()]
7. [*A*→*n*+*A*•;2;5.1;(6.2)]
8. [*A*→*n*+*A*•c;2;5.3;(6.2)]

Запускаем Completer на ситуации 6.6, имеем:

1. [*A*→*n*•+*A*;5;5.4;()]
2. [*A*→*n*•;5;5.5;()]
3. [*A*→*n*•c+*A*;5;5.7;()]
4. [*A*→*n*•+c*A*;5;5.8;()]
5. [*A*→*n*•+*A*c;5;5.9;()]
6. [*A*→*n*•c;5;5.11;()]
7. [*A*→*n*+*A*•;2;5.1;(6.2)]
8. [*A*→*n*+*A*•c;2;5.3;(6.2)]
9. [*A*→*n*+*A*•;0;2.1;(6.7)]
10. [*A*→*n*+*A*•c;0;2.2;(6.7)]

На этом анализ заканчивается, так как лексический анализатор возвращает признак конца потока. Проходим по состоянию 6 и находим ситуацию [*A*→*n*+*A*•;0;2.1;(6.7)]. Значит, символьный поток был проанализирован корректно. Читатель может самостоятельно построить дерево порождения из этой ситуации, проходя по ссылкам влево и вниз, в соответствующих позициях lp и списка tp.

## 5. ЗАКЛЮЧЕНИЕ

В данной статье рассматривается проблематика, связанная с грамматическим анализом языков, обладающих лексической и синтаксической неоднозначностями. Для анализа таких языков предлагается использовать специальным образом модифицированный алгоритм Эрли [8]. Предлагаемая модификация алгоритма основана на идеях, изложенных в диссертационной работе автора [9].

В модифицированном алгоритме Эрли используются идеи, основанные на формальной модели «токена Шредингера» [7]. Автор развивает данный подход и делает его в некотором смысле «универсальным» для разрешения лексических неоднозначностей не в момент их возникновения, но позднее – при анализе символьного потока, в подходящем синтаксическом контексте.

Предлагаемый алгоритм может быть полезен для синтаксического анализа флективных языков, обладающих лексической омонимией, например, русского языка. Основным его отличием от других алгоритмов грамматического анализа, использующихся для таких языков, является достаточно прозрачная для разработчика и довольно производительная обработка не-

однозначностей, как лексических, так и синтаксических. Алгоритм можно использовать для анализа так называемых «открытых языков», для которых грамматика и набор лексических типов постоянно меняются, что приводит к большому количеству лексических и синтаксических неоднозначностей.

## СПИСОК ЛИТЕРАТУРЫ


1. Лапшин В. А. Лекции по математической лингвистике. – М. : Научный мир, 2010.
2. Бениаминов Е. М., Лапшин В. А., Перов Д. В. EZOP – Web-сервис коллективного построения онтологий с открытым языком представлений и запросов // Труды семинара «Знания и Онтологии *ELSEWHERE* 2009», ассоциированного с 17-й Международной конференцией по понятийным структурам. – М. : Гос. ун-т – Высшая школа экономики, 2009. – С. 47 – 56.
3. Бениаминов Е. М., Лапшин В. А., Перов Д. В. Проект Веб-сервера онтологий в стиле Веб 2.0 // Труды 12-й Национальной конференции по искусственному интеллекту с международным участием (КИИ-2010). – Тверь : ТГТУ, 2010.
4. Зеленков Ю. Г., Сегалович Ю. А., Титов В. А. Вероятностная модель снятия морфологической омонимии на основе нормализующих подстановок и позиций соседних слов // Компьютерная лингвистика и интеллектуальные технологии : труды международного семинара «Диалог 2005». – М. : Наука, 2005.
5. Литвинов М. И. Комплексный метод снятия частеречной омонимии с использованием статистики совместного употребления слов в тексте на русском языке // Труды 12-й Национальной конференции по искусственному интеллекту с международным участием (КИИ-2010). – Тверь : ТГТУ, 2010.
6. Сокирко А. В., Толдова С. Ю. Сравнение эффективности двух методик снятия лексической и морфологической неоднозначности для русского языка (скрытая модель Маркова и синтаксический анализатор именных групп) // Международная конференция «Корпусная лингвистика 2004». – СПб., 2004.
7. Aycock J., Horspool R. N. Schrodinger's Token // Software, Practice and Experience. –2001. – Vol. 13, issure 8. – P. 803 – 814.
8. Earley J. An efficient context-free parsing algorithm // Communications of the Association for Computing Machinery. – 1970. – Vol. 13, № 2. – P. 94 – 102.
9. Лапшин В. А. Особенности синтаксического анализа открытых интерфейсных контекстно-свободных языков : дис. … канд. физ.-мат. наук. – М., 2005.



*Материал поступил в редакцию 09.08.11.*


**Сведения об авторе**


**ЛАПШИН Владимир Анатольевич** – кандидат физико-математических наук, руководитель отдела обработки запросов на естественном языке компании ABBYY, Москва
E-mail: mefrill@yandex.ru