ДОПОВІДІ
НАЦІОНАЛЬНОЇ
АКАДЕМІЇ НАУК
УКРАЇНИ

10 • 2016

ІНФОРМАТИКА

http://dx.doi.org/10.15407/dopovidi2016.10.028

УДК 004.8

А.Ф. Кургаев, С.Н. Григорьев

Институт кибернетики им. В.М. Глушкова НАН Украины, Киев
E-mail: afkurgae@ukr.net, Sergey@Grigoriev.kiev.ua

Интерпретатор универсальной машины Тьюринга

(Представлено академиком НАН Украины А.В. Палагиным)

На примере интерпретатора универсальной машины Тьюринга доказана достаточность выразительных возможностей метаязыка нормальных форм знаний (НФЗ) для постановки и решения произвольной задачи, для которой есть решение, и, тем самым, универсальность машины, реализующей этот язык. В процессе обоснования универсальности метаязыка НФЗ даны формальные текстовое и графическое описания интерпретатора универсальной машины Тьюринга.

Ключевые слова: *формальный язык, метаязык нормальных форм знаний, интерпретатор универсальной машины Тьюринга, база знаний, интерпретирующе-транслирующий процесс постановки и решения задач.*

Формальным языком, заданным на базовом множестве U, называют любое подмножество свободной полугруппы U*, т.е. множество цепочек из элементов U (подмножество множества U*), вследствие чего на (формальные) языки распространяются теоретико-множественные (объединения, умножения, дополнения, пересечения, возведения в степень и составные) операции [1—4].

Для определения языков используют формальные грамматики, которые задают четверкой

$$G = \langle \mathbf{V_T}, \mathbf{V_N}, S, \mathbf{P} \rangle,$$

где $\mathbf{V_T}$ и $\mathbf{V_N}$ — непересекающиеся терминальный и нетерминальный алфавиты; S — аксиома, начальный символ; $\mathbf{P}$ — конечная система правил подстановки. Каждой грамматике G однозначно соответствует анализируемый (или порождаемый) язык $L(G)$.

В зависимости от выразительных возможностей различают четыре уровня грамматик (и языков) с номерами от 3 до 0 в порядке возрастания их общности: автоматные (или регулярные), контекстно-свободные (КС), далее контекстные (или НС-грамматики) и на вершине иерархии — уровня 0 и соответствующие им машины Тьюринга.

Множество $\mathbf{G}$ грамматик $G \in \mathbf{G}$ с единообразным набором структур правил множества $\mathbf{P}$ оформляют в некоторый метаязык для определения синтаксиса объектных (описываемых) языков путём использования некоторого количества правил. Среди известных метаязыков наиболее популярны BNF (Backus-Naur Form) и Extended (расширенный) BNF [5]. В [6] предложен и в [7] описан метаязык нормальных форм знаний (НФЗ), развивающий набор выразительных средств EBNF.

Покажем, что выразительных средств метаязыка НФЗ достаточно для описания интерпретатора универсальной машины Тьюринга.

1. База знаний в метаязыке НФЗ. Всякое описание предметной области в метаязыке НФЗ составляется из двух частей. Первая является информационной структурой множества определений понятий (нетерминалов), связанных базовыми отношениями метаязыка НФЗ. Вторая часть состоит из двух подмножеств терминалов — множества элементарных процедур и множества элементарных структур данных, которые реализуют на одном из традиционных языков программирования в форме единой библиотечной системы. Каждый из терминалов имеет собственное имя и в описании структуры используется наравне с нетерминальными понятиями.

Определение 1. База знаний — это всякое замкнутое на предметы предметной области D определение понятия в форме информационной структуры, элементами которой являются понятия, связанные между собой системой отношений метаязыка НФЗ, выполнимых на предметной области D.

Определение 2. Данные — это предметные значения понятий.

2. Структура области данных. Общую предметную область D представим двумя независимыми массивами, удовлетворяющими аксиомам отношения предшествования, — входным INP и выходным OUT, с которыми свяжем переменные m и n, принимающие значения текущих координат (анализа и порождения) соответствующих массивов. Введем также две библиотеки процедур одноименных предикатов — библиотеку анализа над данными INP и библиотеку порождения над данными OUT.

Каждая процедура библиотеки анализа согласно своему содержанию:

• выполняет некоторое преобразование данных, взятых из массива INP, начиная с координаты, определяемой значением переменной m;

• формирует значение истинности, зависящее от успеха преобразования;

• при успехе (истинность — "истина") модифицирует значение переменной m, переводя указатель на начало следующего фрагмента массива INP.

Каждая процедура библиотеки порождения:

• порождает некоторый фрагмент данных в массив OUT, начиная с координаты, определяемой значением переменной n;

• формирует значение истинности "истина" и модифицирует значение переменной n, переводя указатель на начало следующего свободного фрагмента массива OUT.

Введем также набор системных процедур, управляющих этими элементами пространства данных, в том числе:

RB — всегда истинная процедура переключения библиотек. После ее выполнения вызов процедуры из библиотеки порождения заменяется вызовом процедуры из библиотеки анализа и наоборот;

RIO — всегда истинная процедура переключения массивов данных. После ее выполнения процедуры из библиотеки анализа обрабатывают данные из OUT, а процедуры из библиотеки порождения — данные из INP;

UIO — всегда истинная процедура объединения / разделения массивов данных. После однократного ее выполнения INP обрабатывается как два массива (INP и OUT) с всегда равными координатами анализа и порождения. Повторный ее вызов разделяет массивы.

Процедура RIO переключения массивов объединяет в единый процесс преобразование данных одного из массивов с преобразованием данных другого массива. То, что порождено в OUT, становится доступным для анализа и наоборот, то, что проанализировано из INP, может быть заменено порожденным.

3. Определение 0-языков. Язык принадлежит уровню 0 (является 0-языком), если он задан 0-грамматикой с использованием правил "$\varphi \to \psi$", ограниченных лишь тем, что φ и ψ (цепочки из $\mathbf{V_N} \cup \mathbf{V_T}$) не могут быть пустыми.

Общие системы подстановок универсальны: если существует формальная процедура задания языка, то его можно определить 0-грамматикой [1]. Множество правил 0-грамматики включает НС-грамматику, с каждой интерпретацией которой можно связать С-маркер (структуру составляющих [3] или, в другой терминологии, дерево составляющих [1]), множество трансформаций, каждая из которых *преобразует* набор (k, $k \geq 1$) С-маркеров в новый С-маркер, и множество правил, определяющих порядок применения трансформаций. Среди разновидностей трансформаций различают присоединение, перестановки, устранение (или добавление) элементов собственного анализа, всякую из которых можно применять только к полностью проанализированной цепочке [3].

4. Универсальная машина Тьюринга. Универсальная машина Тьюринга — это U-машина с фиксированной структурой, имитирующая *поведение* любой машины Тьюринга (Т-машины).

Для Т-машины исходный результат состоит в том, что если язык L определен 0-грамматикой, то по ней можно построить Т-машину, которая *допускает* язык L, и наоборот [3, 8]. Т.е. утверждается однозначное соответствие между 0-языком и Т-машиной. Следовательно, для множества 0-языков *можно* построить множество Т-машин.

U-машина отличается тем, что на ней одной *можно* моделировать поведение всего множества Т-машин, каждая из которых *допускает* некоторый 0-язык, и тем самым *решить произвольную задачу* уровня 0, для которой есть решение.

В теории алгоритмов интерпретатор U-машины исследовался столь детально потому, что составляет отдельную задачу, сложность которой (по определению) не ниже уровня 0, т.к. в конечном итоге, именно интерпретатор решает произвольную задачу уровня 0 (интерпретируя модель любой Т-машины, решающей задачу уровня 0). В силу этого реализация в некотором языке интерпретатора U-машины гарантирует *достаточность* его выразительных возможностей для постановки и решения произвольной задачи (для которой есть решение) и, тем самым, *универсальность* машины, реализующей этот язык. А. Тьюринг предложил и показал способ построения U-машины, М. Минский [9] разработал форму представления любой Т-машины на ленте U-машины и описал интерпретатор U-машины в форме диаграммы состояний.

Используя эти результаты, в представляемой формализации примем два допущения.

1. Пусть полубесконечная вправо лента U-машины отождествлена с совмещенными (после выполнения процедуры UIO) массивами INP и OUT пространства данных метаязыка НФЗ и разделена на зоны, расположенные слева—направо в следующем порядке:

- начальное состояние q_t некоторой произвольной Т-машины;
- первый, обозреваемый читающе-записывающей головкой Т-машины, символ a_t;
- конечное описание Т-машины (последовательность взаимосвязанных продукций), представленное пятерками q_i, a_i, q_j, a_j, d_i — текущее состояние Т-машины, считываемый символ, новое состояние Т-машины, записываемый символ и значение (0 — вправо, 1 — влево) сдвига (читающе-записывающей головки Т-машины) соответственно. Каждая из продукций имеет смысл выражения: "*Если* Т-машина находится в состоянии q_i и на ее ленте обозревается символ a_i, то заменить его на символ a_j и перейти в состояние q_j, в котором выполнить преобразование соседнего (слева или справа от a_i) символа ленты Т-машины". Каждое состояние q_i, q_j, q_t из множества Q пусть именуется одним символом латиницы или цифры. Входные символы a_i, a_j, a_t пусть принадлежат алфавиту Σ, составленному из букв латиницы и цифр, а также включают нулевой код;
- разделяющий выделенный символ Код0;
- полубесконечная вправо псевдолента Т-машины содержит конечную цепочку символов (за ней — нулевые коды), одно из ее знакомест занято маркером (тот же выделенный символ Код0), указывающим положение читающе-записывающей головки Т-машины.

2. Положение читающе-записывающей головки U-машины однозначно определяется значением переменной m (координаты INP).

В начальный момент: читающе-записывающая головка U-машины расположена в крайнем левом положении и обозревает символ, именующий начальное состояние q_t Т-машины; читаю-

ще-записывающая головка Т-машины установлена также в крайнем левом положении ее ленты (маркер Код0 записан в первой клетке псевдоленты Т-машины).

Семантику интерпретатора U-машины *формально определим* базой знаний:

```
интерпретация = состояние символ# (^Код0 ^правило ^преобразовать);
^правило = та_пятерка? состояние символ сдвиг# / пятерка ^Код0 ^правило;
та_пятерка = RB состояние символ RB! / ^RB;
пятерка = состояние символ состояние символ сдвиг;
^преобразовать = (^Код0 пятерка) Код0 (символ ^Код0) ^зап_сдвиг заменить ^чтение? Код0!;
^зап_сдвиг = символ? сдвиг! / сдвиг!;
сдвиг = '0' / '1';
заменить = символ ^если_0? символ! /^если_0? символ! /^записать_символ;
^если_0 = '0';
^записать_символ = символ? символ! / символ!;
^чтение = символ#.
```

База знаний состоит из 14 понятий, в т.ч. 11 определяют состояния U-машины (это число можно уменьшить до 10, заменив понятие “пятерка” его определяющей частью), и трех процедур (“символ”, “состояние” — значения из множеств Σ и Q соответственно; “Код0” — разделяющий символ или маркер), а также двух констант ('0', '1').

Графическое описание интерпретатора U-машины, выполненное графическими средствами метаязыка НФЗ, представлено на рис. 1.

Очевидно, что оно эквивалентно приведенному текстовому описанию.

Представленное определение интерпретатора U-машины может быть применено для решения задачи моделирования поведения всякой Т-машины (решающей задачу преобразования данных на ее ленте памяти), которая формулируется обычным образом — как доказательство общего суждения:

“Всякое значение данных из INP (начало которых определено значением переменной m) суть интерпретация”.

Вывод этого суждения идет в следующей последовательности. При анализе со следом с понятиями “состояние” и “символ” связываются: начальное состояние Т-машины и первый символ на ее ленте. Затем в итерационном цикле интерпретируется последовательность понятий “^поиск_правила” и “^преобразовать”, первое из них определяет ассоциативный поиск пятерки, описывающей очередную смену состояния моделируемой Т-машины, второе — замену на ленте Т-машины ранее взятого символа новым (из соответствующей пятерки), читает очередной символ (согласно указателю о сдвиге головки Т-машины) и записывает на его место маркер Т-машины. Поиск подходящего правила Т-машины выполняется просмотром пятерок описания Т-машины отождествлением текущего и связанного значений. Если проанализированные со следом состояние или символ не совпадают со значениями элементов очередной пятерки, то пятерка пропускается, иначе — анализируются со следом новое состояние, заменяющий символ и значение сдвига головки Т-машины.

Если все описание Т-машины уже просмотрено безуспешно и на очереди разделяющий Код0, то формируется значение истинности “ложь”, рекурсивный и затем итерационный цикл прерываются и интерпретатор заканчивает свою работу с заключительным значением истинности “истина”. Иначе поиск пятерки рекурсивно повторяется до нужной пятерки или пока не выявится, что запомненное состояние — заключительное. Если пятерка найдена, — после последовательной интерпретации четного числа отрицаний головка U-машины возвращается к началу описания Т-машины. Интерпретация понятия “^преобразовать” начинается с пропуска описания Т-машины, затем пропуск разделяющего символа и далее поиск маркера. После того, как положение головки Т-машины найдено, вместо маркера записывается значение сдвига, затем

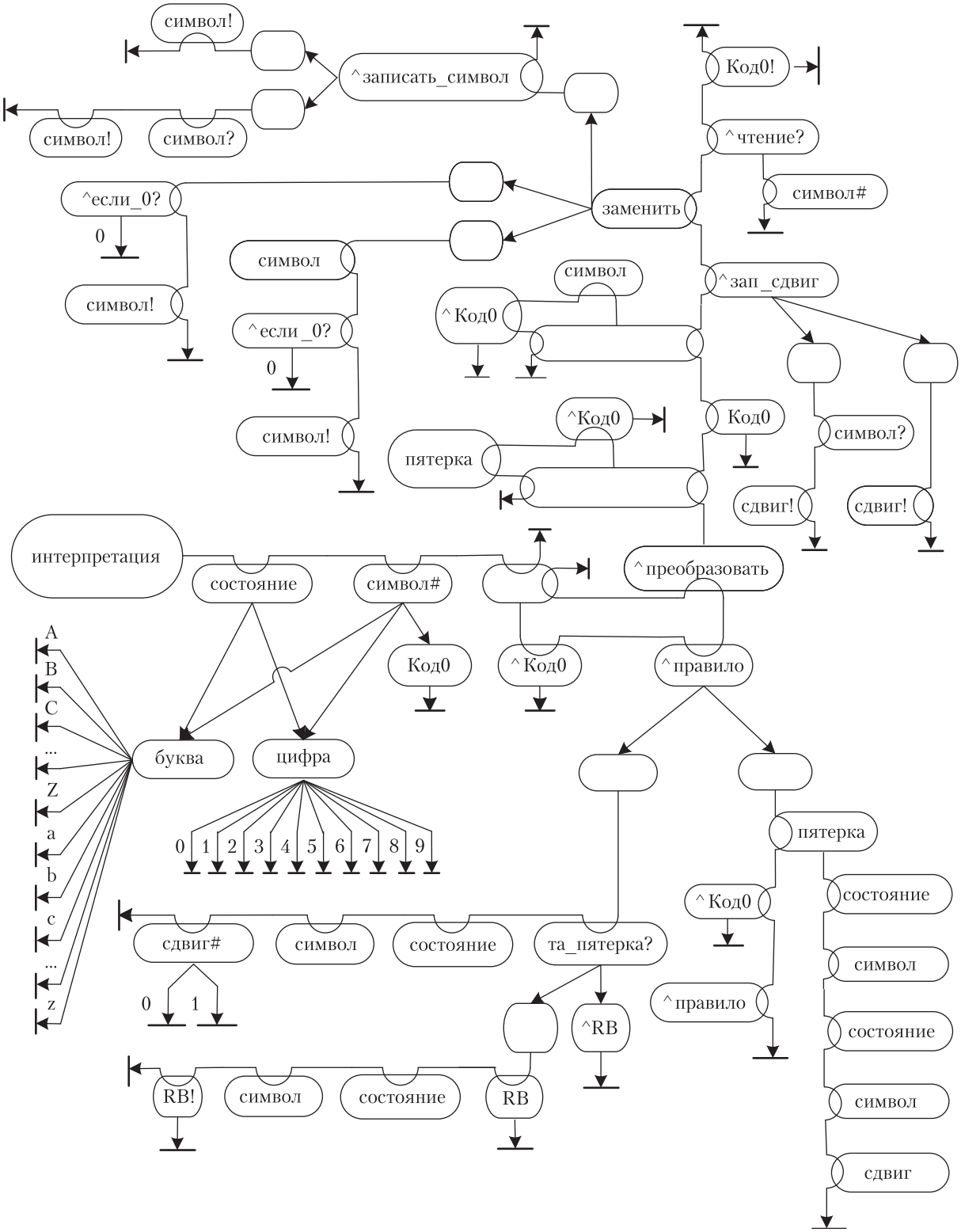

Рис. 1. Граф интерпретатора U-машины

анализ значения сдвига, запись на его место заменяющего символа, чтение очередного символа (слева или справа от маркера) и запись вместо него маркера.

Таким образом, рассмотренная интерпретация представленного в метаязыке определения U-машины подтверждает его эквивалентность диаграмме состояний, составленной М. Минским [9].

В ряду известных определений U-машины, составленных в свое время Икено, Ватанабе и Минским, данное является *простейшим* по критерию К. Шеннона [10] и наиболее *естественным*.

Метаязык НФЗ включает семантические возможности U-машины, следовательно, всякий 0-язык в форме интерпретатора может быть определен в метаязыке НФЗ.

Используя этот результат, *интерпретирующе-транслирующий процесс* постановки и решения произвольной задачи представляется следующей последовательностью преобразований:

• на метаязыке описать интерпретатор объектного языка и ввести в машину в качестве данных — разместить в массиве INP;

• выполнить ввод (трансляцию в машинное представление) определения интерпретатора объектного языка;

• на объектном языке описать задачу и ввести в машину в качестве данных — разместить в массиве INP;

• дать задание машине вывести общее суждение в форме “Всякое значение данных из INP (начало которых определено значением переменной *m*) суть интерпретация объектного языка”.

Результат: значение истинности последнего суждения и преобразованные данные.

ЦИТИРОВАННАЯ ЛИТЕРАТУРА

1. *Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л.* Алгебра. Языки. Программирование. — Киев: Наук. думка, 1978. — 320 с.
2. *Гросс М., Лантен А.* Теория формальных грамматик. — Москва: Мир, 1971. — 294 с.
3. *Chomsky N.* Aspects of the Theory of Syntax. — Cambridge, MA: MIT Press, 1965.
4. *Chomsky N.* Some Concepts and Consequences of the Theory of Government and Binding. — Cambridge, MA: MIT Press, 1982.
5. *International Standard* ISO/IEC 14977 : 1996(E). — Электронный ресурс. — Режим доступа: http://www.cl.cam.ac.uk/~mgk25/iso-14977.pdf
6. *Кургаев О.П., Григор’єв С.М.* Спосіб представлення і використання знань: Патент на корисну модель UA 92484 U, 2014 р., Бюл. №16.
7. *Кургаев А.Ф., Григорьев С.Н.* Нормальные формы знаний // Доп. НАН України. — 2015. — № 11. — С. 36—43.
8. *Fu K.S.* Syntactic Pattern Recognition and Applications. — Englewood Cliffs: Prentice-Hall, 1982. — 596 p.
9. *Minsky M.L.* Computation: Finite and Infinite Machines. — Englewood Cliffs: Prentice-Hall, 1967.
10. *Shannon C.E.* A universal Turing machine with two internal states. — Princeton: Automata Studies, 1956.

REFERENCES

1. *Glushkov V., Zeitlin G., Yushchenko E.* Algebra. Languages. Programming. Kiev, Nauk. Dumka, 1978 (in Russian).
2. *Gross M., Lunten A.* Theory of formal grammars. Moscow: Mir, 1971 (in Russian).
3. *Chomsky N.* Aspects of the Theory of Syntax, Cambridge, MA: MIT Press, 1965.
4. *Chomsky N.* Some Concepts and Consequences of the Theory of Government and Binding, Cambridge, MA: MIT Press, 1982.
5. *International Standard* ISO/IEC 14977: 1996(E). Electronic resource. Access: http://www.cl.cam.ac.uk/~mgk25/iso-14977.pdf
6. *Kurgaev A., Grygoryev S.* Utility model patent UA 92484 U, 2014, Bulletin № 16 (in Ukrainian).
7. *Kurgaev A., Grygoryev S.* The normal forms of knowledge. Dopov. Nac. akad. nauk Ukr., 2015, No 11: 36-43 (in Russian).
8. *Fu K.S.* Syntactic Pattern Recognition and Applications. Englewood Cliffs: Prentice-Hall, 1982.
9. *Minsky M.L.* Computation: Finite and Infinite Machines. Englewood Cliffs: Prentice-Hall, 1967.
10. *Shannon C.E.* A universal Turing machine with two internal states. Princeton: Automata Studies, 1956.

Поступило в редакцию 23.02.2016

О. П. Кургаєв, С.М. Григор'єв

Інститут кібернетики
ім. В.М. Глушкова НАН України, Київ
E-mail: afkurgae@ukr.net, Sergey@Grigoriev.kiev.ua

ІНТЕРПРЕТАТОР УНІВЕРСАЛЬНОЇ МАШИНИ ТЬЮРИНГА

На прикладі інтерпретатора універсальної машини Тьюринга доведена достатність виразних можливостей метамови нормальних форм знань (НФЗ) для постановки та розв'язку довільного завдання, для якого є розв'язок, і отже, універсальність машини, що реалізує цю мову. У процесі обґрунтування універсальності метамови НФЗ дано формальні текстовий і графічний описи інтерпретатора універсальної машини Тьюринга.

Ключові слова: *формальна мова, метамова нормальних форм знань, інтерпретатор універсальної машини Тьюринга, база знань, інтерпретуюче-транслюючий процес постановки та розв'язку задач.*

A.F. Kurgaev, S.N. Grygoryev

V.M. Glushkov Institute
of Cybernetics of the NAS of Ukraine, Kyiv
E-mail: afkurgae@ukr.net, Sergey@Grigoriev.kiev.ua

THE UNIVERSAL TURING MACHINE INTERPRETER

Using an interpreter of the universal Turing machine as the example, it is shown that the NFK (normal forms of knowledge) meta-language expressiveness is sufficient for defining and solving any solvable problem, which proves the versatility of a computer, which realizes this language. In the process of substantiation of the versatility of the NFK meta-language, the formal text and graphical descriptions of an interpreter of the universal Turing machine are given.

Keywords: *formal language, meta-language of normal forms of knowledge, interpreter of the universal Turing machine, knowledge base, interpretative and translating process of statement and solution of problems.*