Язык описания алгоритма перевода

Д. Анисимов

24 июля 2004 г.

Содержание

1 Введение

Ниже будет дано формальное описание языка описания алгоритма перевода. Имейте в виду, что это именно формальное описание синтаксиса. Смысл всех используемых понятий обьяснен в idea.ps (idea.pdf).

Формальное описание состоит из серии статей. Каждая статья описывает одну синтаксическую конструкцию языка описания алгоритма перевода. Каждая статья содержит формальное определение синтаксической конструкции, неформальные очень короткие пояснения к ней и пример(ы) такой конструкции.

Каждое формальное определение состоит из трех частей:

1. определяемая конструкция (ее название)
2. знак равно
3. составные части этой конструкции

При описании формальных определений приняты следующие обозначения.

1. Жирный шрифт выделяет ключевые слова языка описания алгоритма перевода.
2. Прямой шрифт - элементы (конструкции) языка, определяемая конструкция и ее составляющие.
3. Курсив - элементы описываемые пользователем (имена частей речи, параметров, значения параметров и пр). Эти элементы всегда представляют из себя строку.
4. Знак вопроса перед элементом конструкции означает необязательный элемент конструкции.
5. Многоточие "..."после элемента конструкции означает, что таких элементов может быть описано много.
6. Запятая и многоточие ",.."означает, что элементов может быть много, и между ними должны стоять запятые.
7. Если после знака равно встречается вертикальная черта '|' это означает, что конструкция может быть представлена двумя (или более) вариантами. Вертикальная черта отделяет варианты друг от друга.

2 Коментарии и прочие «мелочи».

Файл описания грамматики может содержать комментарии, как в С++, одно- и многострочные. Например вот такие:

```
/* Это многострочный
   комментарий
*/
// Это однострочный комментарий
```

Служебные слова начинаются с `@` (эт коммерческое). Например, `@Атомы_источника`.

Одно замечание, которое нужно сделать именно сейчас, иначе потом не будет понятно. В языке описания грамматики в разных местах часто встречается символ @0. Он везде означает "нечто не заданное". Незаданную часть речи, незаданный или непроверяемый параметр, и т.д. Нуль это, короче.

3 Файл описания грамматики

Синтаксис:

файл_описания_грамматики =
    атомы_источника
    атомы_приемника
    словари_переводов
    правила_перевода

Пояснения:

Собственно, не знаю, что тут можно пояснить. Файл описания грамматики содержит четыре секции, которые следуют друг за другом. Пример упрощенного, но тем не менее полного описания языка смотрите в главе «Пример файла описания грамматики».

4 Атомы

Синтаксис:

атомы_источника =
@Атомы_источника описание_языка

атомы_приемника =
@Атомы_приемника описание_языка

описание_языка =
{
параметр_языка...
часть_речи...
составная_часть_речи...
формат_грамматического_словаря...
файлы_грамматических_словарей...
}

Пояснения:

Собственно, я опять не знаю, что тут можно пояснить. Секция описания атомов языка содержит записи пяти типов, которые следуют друг за другом.

5 Параметры

Синтаксис:

параметр_языка =
@Параметр *тип_параметра* **{** *значение_параметра,..* **} ;**

Примеры:

```
@Параметр лицо  {  1е,2е,3е  };
@Параметр время {  прошлое,настоящее,будущее  };
```

Эта запись означает, что в описываемом (русском) языке есть параметр лицо, который имеет три значения - 1е,2е,3е и есть параметр время, который имеет три значения - прошлое,настоящее,будущее.

6 Части речи

Синтаксис:

часть_речи =
@Часть_речи *имя_части_речи* **(** *параметр_конструкции,..* **) ;**

параметр_конструкции =
    тип_параметра ?**&** *имя_параметра*

тип_параметра - имя одного из параметров описанных для этого языка. Это своего рода ‘тип переменной’. Часть речи может иметь два и более параметров одного типа. Такая ситуация например существует для русских притяжательных местоимений.

& - признак того, что данный параметр у этой части речи постоянный. Например как род у русского существительного.

имя_параметра - своего рода ‘имя переменной,

Пример:

```
@Часть_речи существительное ( род &Род, падеж Падеж, число Число );
```

В русском языке у существительных грамматический род является постоянным параметром. Этот факт отмечен амперсандом ’&’. Следует ли говорить о том, что род, падеж, число и т.п. должны быть описаны в секции параметров этого языка?

7 Грамматические словари

Синтаксис:

формат_грамматического_словаря =
    @Формат *имя_части_речи имя_формата*
    { запись_словаря...
    }

запись_словаря =
    псевдоним_словаря |
    присвоение_параметра |
    имя_параметра |
    @Слово |
    блок_словаря

псевдоним_словаря =
    @Псевдоним *псевдоним значение_параметра*

присвоение_параметра =
    имя_параметра = *значение_параметра*

блок_словаря =
    { запись_словаря...
    }

файлы_грамматических_словарей =
    @Файлы *имя_части_речи имя_формата* **{** *имя_файла...* **};**

имя_части_речи должно соответствовать одной из частей речи описываемого языка.

имя_формата - произвольная строка

псевдоним - короткая строка, которая обозначает значение_параметра (для сокращения объема словаря) присвоение_параметра - конструкция, позволяющая указать, что последующие в строке слова будут иметь заданное значение указанного парамера. Подряд может следовать несколько таких записей, устанавливающих значения различных параметров.

параметр - запись, указывающая. что очередное слово в словаре является не формой слова, а значением постоянного параметра. имя_параметра должно соответствовать имени одного из параметров этой части речи.

@Слово - запись, означающая, что далее в словаре будет присутствовать слово в форме, заданной значениями параметров.

блок - это нечто в фигурных скобках. Параметры, заданные внутри блока при выходе из блока возвращаются к значениям, которые были до начала блока.

Пример (формат словаря русских местоимений):

```
@Формат местоимение формат4
{
    Падеж=И @Слово
    Падеж=Р @Слово
    Падеж=Д @Слово
    Падеж=В @Слово
    Падеж=Т @Слово
    Падеж=П @Слово

    Лицо
    Число
    Род
}
```

8 Словари переводов

Синтаксис:

словари_перевода =

@Словари_перевода

{

формат_перевода...

файлы_перевода...

}

формат_перевода =

@Формат_перевод *имя_формата*

{ @Части_речи_источника

{

запись_о_части_речи...

}

@Части_речи_приемника

{

запись_о_части_речи...

}

}

запись_о_части_речи =
псевдоним имя_части_речи **;**

файлы_переводов =
@Файлы_переводов *имя_формата* **{** *имя_файла…* **};**

Пример:

```
@Словари_перевода
{
   @Формат_перевод формат1
   {  @Части_речи_источника
      {
         V глагол ;
         S сущ    ;
         N сущ    ;
         A прил   ;
         D наречие ;
         I числительное ;
         P предлог ;
         C союз ;
         Q вопрос ;
         O местоимение ;
         R прит_местоимение ;
      }
      @Части_речи_приемника
      {
         V глагол ;
         S сущ    ;
         N сущ    ;
         A прил   ;
         D наречие ;
         I числительное ;
         P предлог ;
         C союз ;
         Q вопрос ;
         O местоимение ;
         R прит_местоимение ;
      }
   }
   @Файлы_переводов формат1 { base_enru };
}
```

Формат перевода задает каждой части речи короткий псевдоним. Каждая запись о части речи содержит псевдоним и имя части речи. Естественно, имена частей речи должны соответствовать частям речи источника или приемника.

Описание файлов перевода строится аналогично описанию файлов грамматических словарей. Точно также есть предваряющее ключевое слово, имя формата, и далее в фигурных скобках идет список файлов, содержащих словари перевода.

9 Правила перевода

Алгоритм перевода описывается как набор правил, каждое из которых ставит в соответствие конструкцию языка-источника и конструкцию языка-приемника. Желательно этим двум конструкциям задавать одинаковые имена. Компьютеру все равно, а человеку желательно не запутаться. Каждое правило описывает внутреннее устройство конструкции языка-источника внутреннее устройство конструкции языка- приемника, и правила вычисления параметров приемника по параметрам источника.

Синтаксис:

правила_перевода =
@Переводы
{
перевод_конструкции...
}

10 Перевод_конструкции

Синтаксис:

перевод_структуры =
@Перевод
{
тип_конструкции
конструкция
тип_конструкции
конструкция
?таблица...
}

Перевод задает одно правило трансляции. Первая конструкция - это конструкция языка оригинала, вторая - конструкция языка перевода). Таблицы задают правила вычисления параметров приемника по параметрам источника. Тип_конструкции оригинала и перевода должен совпадать.

Синтаксис:

тип_конструкции =
@Структура |
@Структура1 |
@Структура2 |
@Выбор |
@Выбор1|
@Выбор2
@Беспорядок
@Беспорядок1
@Беспорядок2
@Множество

@Структура - это «обычная» структура
@Структура1 - это «структура из словаря»
@Структура2 - это «произвольная структура из словаря»
@Выбор - это выбор
@Выбор1 - это «выбор определяемый словарем»
@Выбор2 - это выбор, в котором может быть огромное число вариантов [1]
@Беспорядок - это «структура с произвольным порядком членов»
@Беспорядок1 - это «беспорядок из словаря»
@Беспорядок2 - это «произвольный беспорядок из словаря»
@Множество - это выбор из множества

Синтаксис:

```
конструкция =
    заголовок_конструкции =
    элемент_конструкции...
    ;
```

Конструкция - это некая структура естественного языка. Заголовок конструкции - это ее название и ее параметры. Элемент конструкции - составляющие конструкции и их параметры. В конструкцию могут входить слова и другие конструкции, в том числе и конструкции того же самого типа.

Пример:

```
// ------------- подлежащее ---------------------------
@Перевод
{
  // ------ описание конструкции оригинала -----------
          // заголовок
   @Выбор  подлежащее( лицо &Лицо, число &Число ) =
          // вариант, когда подлежащее представлено местоимением
          личное_местоимение( &Лицо, &Число )
          // вариант, когда подлежащее представлено существительным
          гр_сущ( &Число ) ;

   // ------ описание конструкции перевода -----------
          // заголовок
   @Выбор  подлежащее( лицо &Лицо, число &Число, род &Род ) =
          // вариант с местоимением
          личное_местоимение( &Лицо, &Число, &Род )
          // вариант с существительным
          гр_сущ( &Число, &Род ) ;

   // ------ описание зависимости параметров ---------
   @Таблица( Число = Число )
   {  Ед = Ед ;
      Мн = Мн   ;
```

[1] С точки зрения результатов работы программы Выбор2 от Выбор ничем не отличается. Но это “подсказка” программе, о том, что в этом месте может быть огромное количество вариантов. В этом случае программа начинает бороться с ростом используемой памяти специфичными приемами.

```
   }
   @Таблица( @Выбор Лицо = Лицо )
   {  1 1е = 1е ;
      1 2е = 2е ;
      1 3е = 3е ;
      2 @0 = 3е ;
   }
}
```

11 Заголовок_конструкции

Синтаксис:

заголовок_конструкции =
имя_конструкции
?параметры_конструкции

параметры_конструкции =
(парамер_конструкции...)

заголовок_конструкции - это имя структуры и ее параметры. Параметры в заголовке имеют тип. Заголовок конструкции похож на описание части речи. Это сходство не случайное, часть речи - это тоже элементарная (атомарная) конструкция, которая не имеет внутренних частей.

12 Конструкция

Синтаксис:

элемент_конструкции =
имя_конструкции
?[*константное_слово*]
?<*индекс*>
?параметры_конструкции

Конструкция описывает части структуры или варианты выбора.
имя конструкции - имя конструкции или части речи этого языка. Существуют три особых имени конструкции:

@0 - жесткая константа, не являющаяся частью речи. У нее не бывает параметров или индексов.

@00 - пустой выбор. Такой элемент бывает только у конструкций типа «Выбор», И означает он, что данная конструкция может быть пустой. Например нулевой артикль. У пустого выбора не бывает параметров, индексов и смысловых констант.

все_все - самая крупная единица перевода (фраза). Все остальные конструкции это варианты реализации фразы, или ее составные части. Все_все - это аналог функции main в языке C++. Все_все не является частью какой-то еще более общей конструкции.

индекс - некое число, которое помогает понять соответствие между составляющими источника и приемника. Это бывает нужно, когда в конструкции присутствуют элементы одного типа (два существительных). Индексы применяются только в конструкциях типа «структура».

параметры_конструкции в целом аналогичны параметрам заголовка конструкции. однако в данном случае не надо указывать типы параметров, поскольку они уже указаны, когда описывалась эта конструкция или часть речи.

Пример:

```
@Перевод
{  @Структура ядро_мочь_будущее( смысл_глагола Смысл ) =
             гр_сущ( &Число, &Лицо )
             @0[will] Not @0[be] @0[able] @0[to]
             глагол<1>( Inf,&Смысл ) ;
   @Структура ядро_мочь_будущее =
             гр_сущ( &Число, &Лицо, И, &Род )
             Not мод_глагол[мочь]( Сов,глагол,Лицо,Число,будущее,@0,Актив )
             глагол<1>( Сов,Inf,@0,@0,@0,@0,Актив,@0 ) ;
   @Таблица( Число = Число )
   {  Ед = Ед ;
      Мн = Мн   ;
   }
}
```

13 Параметры_конструкции

Синтаксис:

параметры_конструкции =
(параметр...)

параметр =
?& *имя_параметра* |
значение_параметра |
@0

& - признак параметра источника.
имя_параметра - имя параметра. Не путать с типом параметра!
значение_параметра - одно из значений параметра этого типа. Имеет разный смысл в источнике и приемнике. В источнике это "требуемое значение"параметра. То есть данная составляющая должна иметь именно такое значение параметра, чтобы описываемая конструкция соответствовала шаблону. В приемнике это "задаваемое значение"параметра, то есть это значение, которое передается вниз по дереву иерархии и в конечном счете участвует в склонении и спряжении слов.

@0 обозначает не заданное значение параметра. Для слова это означает, что при извлечении слова из грамматического словаря этот параметр не проверяется.

В этом месте мой язык имеет крупный недостаток. В процессе чтения не существует возможности различить значение параметра написанное с ошибкой и имя параметра. Будьте внимательны при написании файла.

14 Таблицы

Таблицы служат для вычисления параметров источника по параметрам приемника. Один перевод может содержать несколько (в том числе ни одной) таблиц.

Синтаксис:

таблица =
    @Таблица
    заголовок_таблицы
    { строка_таблицы...
    }

заголовок_таблицы =
    (*параметр_таблицы... = параметр_таблицы...*)

параметр_таблицы =
    имя_параметра |
    @Выбор

строка_таблицы =
    значение_параметра... = значение_параметра...

Каждая строка является комбинацией входных и выходных значений параметров. Количество значений параметров до и после знака равно должно соответствовать количеству имен параметров в заголовке таблицы. Значения параметров должны соответствовать типу соответствующего параметра. При вычислении параметров перевода осуществляется последовательный просмотр строк таблицы. Если в конструкции оригинала все указанные в заголовке (до знака равно) параметры равны указанным в строке (до знака равно) значениям - параметрам конструкции перевода присваиваются значения из этой строки после знака равно.

Имя параметра @Выбор применяется только в конструкциях перевод_выбора. Это означает, что строка может быть использована только в том случае, если при переводе используется соответствующий вариант выбора. Варианты выбора считаются с 1.

Пример:

```
@Таблица( форма Форма, лицо Лицо = время Время, лицо Лицо )
{  Inf   1e = настоящее 1e ;
   Inf   2e = настоящее 2e ;
   Inf   3e = настоящее 3e ;
   Form2 @0 = прошлое   3e ;
}
```

15 Полный пример файла описания грамматики.

Ниже приводится пример пример файла описания грамматики. Описывается перевод с "идеального аналитического"языка в "идеальный синтетический". Пример учебный и очень упрощенный. В реальных языках гораздо больше частей речи и параметров. И самое главное - гораздо больше исключений из правил. Не надо думать, что эти два языка — упрощенный английский, и упрощенный русский.

Тем не менее, на обоих этих языках можно говорить, и даже выразить любую мысль. Это к вопросу об оптимальности естественных языков.

Ну и для тех, кто не знает, что такое аналитические языки скажу, что в аналитических языках роли слов в предложении определяются порядком и служебными словами. В синтетических - грамматическими формами слов. Все реальные языки не бывают чисто аналитическими или чисто синтетическими. Почти аналитическим является английский. Почти синтетическим являетются русский, латынь и вся группа романских языков. Языки германской группы примерно посередине.

Да простят меня лингвисты, но я не считаю английский язык представителем германской группы. У него гораздо больше сходства с китайским нежели с немецким языком. То, что несколько сотен слов у немцев и англичан звучит похоже, так по татарски интерфейс тоже интерфейсом называется, а шоссе — шоссе.

Я надеюсь, что даже и эту распечатку Вы прочтете с таким же интересом, как и все остальное. Какой бы язык Вы не описывали, его нужно будет привести к этим нескольком конструкциям:

сложноподчиненная фраза
простая фраза
подлежащее
сказуемое
дополнение
обстоятельство.

Естественные языки имеют несколько больше вариантов реализации этих конструкций, и никаких других, котрые не входили бы как варианты в эти шесть.

```
/*************************************************************************/
//             Пример описания перевода с идеального аналитического
//                      языка на идеальный синтетический
//
/*************************************************************************/

/*************************************************************************/
//                              язык оригинала
/*************************************************************************/
@Атомы_источника
{
   // ------- параметры ---------------------
   @Параметр число {  Ед,Мн  };
   @Параметр лицо  {  1е,2е,3е  };

   // ------- части речи ---------------------
   @Часть_речи глагол ;
   @Часть_речи модальный_глагол ;
   @Часть_речи существительное ;
   @Часть_речи прилагательное ;
   @Часть_речи наречие ;
   @Часть_речи местоимение( число &Число, лицо &Лицо ) ;
   @Часть_речи артикль

   // ------- структуры из словаря -----------
   @Структура1 существительное1 = прилагательное существительное ;
   @Структура1 существительное2 = существительное @1 ;

   // ------- форматы и файлы ----------------
   @Формат местоимение формат1
   {
      @Слово
      Лицо
```

```
      Число
   }

   @Файлы местоимение формат1 { s_pronoun };
}
/****************************************************************************/
//                           язык перевода
/****************************************************************************/
@Атомы_приемника
{
   // ------- параметры ----------------------
   @Параметр число   {  Ед,Мн  };
   @Параметр лицо    {  1е,2е,3е  };
   @Параметр падеж   {  И,Р,Д,В,Т,П  };
   @Параметр род     {  М,Ж,С  };
   @Параметр время   {  прошлое,настоящее,будущее  };
   @Параметр залог   {  Неопредел,Актив,Повелит  };

   // ------- части речи ---------------------
   @Часть_речи глагол          ( число Число, лицо Лицо, время Время, залог Залог );
   @Часть_речи модальный_глагол( число Число, лицо Лицо, время Время, залог Залог );
   @Часть_речи существительное ( число Число, падеж Падеж, род &Род );
   @Часть_речи прилагательное  ( число Число, падеж Падеж, род  Род );
   @Часть_речи наречие         ( степень С );
   @Часть_речи местоимение     ( число Число, лицо Лицо, падеж Падеж, род &Род );

   // ------- структуры из словаря -----------
   @Структура1 существительное1( число &Число, падеж &Падеж, род  Род ) =
            прилагательное  ( Число, Падеж,  Род )
            существительное ( Число, Падеж, &Род );

   @Структура1 существительное2( число &Число, падеж &Падеж, род  Род ) =
            существительное ( Число, Падеж, &Род )
            @1 ;
   @Структура1 существительное3( число &Число, падеж &Падеж, род  Род ) =
            прилагательное  ( Число, Падеж,  Род )
            существительное( Число, Падеж, &Род )
            @1 ;

   // ------- форматы и файлы ----------------
   @Формат глагол формат1
   {
      залог=Неопредел @Слово
      залог=Актив
      Время=настоящее
      Число=Ед   Лицо=1е @Слово  Лицо=2е @Слово  Лицо=3е @Слово
      Число=Мн   Лицо=1е @Слово  Лицо=2е @Слово  Лицо=3е @Слово

      Время=прошлое
      Число=Ед   Лицо=1е @Слово  Лицо=2е @Слово  Лицо=3е @Слово
      Число=Мн   Лицо=1е @Слово  Лицо=2е @Слово  Лицо=3е @Слово
```

```
   Время=будущее
   Число=Ед   Лицо=1е @Слово  Лицо=2е @Слово  Лицо=3е @Слово
   Число=Мн   Лицо=1е @Слово  Лицо=2е @Слово  Лицо=3е @Слово

   залог=Повелит @Слово
}
@Формат существительное формат2
{  @Псевдоним  m М
   @Псевдоним  f Ж
   @Псевдоним  n С

   Число=Ед
   Падеж=И @Слово  Падеж=Р @Слово
   Падеж=Д @Слово  Падеж=В @Слово
   Падеж=Т @Слово  Падеж=П @Слово

   Число=Мн
   Падеж=И @Слово  Падеж=Р @Слово
   Падеж=Д @Слово  Падеж=В @Слово
   Падеж=Т @Слово  Падеж=П @Слово

   Род
}
// ---- пример использования препроцесора С++
// ---- повторяющуюся последовательность
// ---- не обязятельно писать несколько раз
# define pril_chislo_rod \
      Число=Ед        \
      Род=М @Слово    \
      Род=Ж @Слово    \
      Род=С @Слово    \
      Число=Мн @Слово \

@Формат прилагательное формат3
{
   Падеж=И pril_chislo_rod
   Падеж=Р pril_chislo_rod
   Падеж=Д pril_chislo_rod
   Падеж=В pril_chislo_rod
   Падеж=Т pril_chislo_rod
   Падеж=П pril_chislo_rod
}
@Формат местоимение формат4
{
   Падеж=И @Слово
   Падеж=Р @Слово
   Падеж=Д @Слово
   Падеж=В @Слово
   Падеж=Т @Слово
   Падеж=П @Слово
```

```
      Лицо
      Число
      Род
   }

   @Файлы глагол          формат1 { d_verb } ;
   @Файлы модальный_глагол формат1 { d_mverb } ;
   @Файлы существительное  формат2 { d_substantive } ;
   @Файлы прилагательное   формат3 { d_adjective };
   @Файлы местоимение      формат4 { d_pronoun } ;
}
/***************************************************************************/
//                         словарь перевода
/***************************************************************************/
@Словари_перевода
{
   @Формат_перевод формат1
   {  @Части_речи_источника
      {
         V  глагол ;
         M  модальный_глагол ;
         S  существительное ;
         S1 существительное1 ;  // структура из словаря
         S2 существительное2 ;  // структура из словаря
         SS гр_сущ1 ;           // произвольная структура из словаря
         A  прилагательное ;
         D  наречие ;
         O  местоимение ;
         _  артикль
      }
      @Части_речи_приемника
      {
         V  глагол ;
         M  модальный_глагол ;
         S  существительное ;
         S1 существительное1 ;  // структура из словаря
         S2 существительное2 ;  // структура из словаря
         S3 существительное3 ;  // структура из словаря
         SS гр_сущ1 ;           // произвольная структура из словаря
         A  прилагательное ;
         D  наречие ;
         O  местоимение ;
      }
   }
   @Файлы_переводов формат1 { base_dict };
}
/***************************************************************************/
//                         правила переводов
/***************************************************************************/
@Переводы
```

```
{
// ----------------------------------------------------------------
//                      группа существительного
// ----------------------------------------------------------------
@Перевод
{
   // структура ссылается сама на себя, это чтобы распознавать
   // однородные члены типа "большой тупой и жизнерадостный программист "
   @Структура определение =
              определение
              прилагательное ;
   @Структура определение( число &Число, падеж &Падеж, род  &Род ) =
              определение( Число, Падеж, Род )
              прилагательное( Число, Падеж, Род ) ;
}
@Перевод
{
   @Структура определитель1 = @0[of] группа_существительного ;
   @Структура определитель1 = группа_существительного( Р );
}
@Перевод
{
   @Структура уточняющий_оборот =
              @0[wich]        // "жесткая" константа
              неполная_фраза ;
   @Структура уточняющий_оборот( число &Число, падеж &Падеж ) =
              существительное[который]( Число, Падеж ) // мягкая константа
              неполная_фраза ;
}
@Перевод
{
   @Выбор     определитель =
              определитель1
              уточняющий_оборот
              @00 ;          // пустой выбор
   @Выбор     определитель( число &Число, падеж &Падеж ) =
              определитель1
              уточняющий_оборот( Число, Падеж )
              @00 ;
}
@Перевод
{
   @Структура гр_сущ1( число Число ) =
              артикль( число &Число )
              определение
              существительное
              определитель ;
   @Структура гр_сущ1( число &Число, падеж &Падеж ) =
              определение( Число, Падеж, Род )
              существительное( Число, Падеж, &Род )
              определитель( Число, Падеж ) ;
```

```
}
@Перевод
{  // ------- вот он красавец - выбор из словаря ---------
   @@Выбор1   составное_существительное1 =
              существительное1
              существительное2 ;
   @@Выбор1   составное_существительное1( число &Число, падеж &Падеж ) =
              существительное1( Число, Падеж, &Род )
              существительное2( Число, Падеж, &Род )
              существительное3( Число, Падеж, &Род );
}
@Перевод
{
   @Структура составное_существительное( число Число ) =
              артикль( &Число )
              составное_существительное1 ;
   @Структура составное_существительное( число &Число, падеж &Падеж ) =
              составное_существительное1( Число, Падеж ) ;
}
@Перевод
{
   @Выбор     гр_сущ( число Число, лицо Лицо ) =
              гр_сущ1( &Число )
              местоимение( &Число, &Лицо )
              составное_существительное( &Число ) ;
   @Выбор     гр_сущ( число &Число, лицо &Лицо, падеж &Падеж ) =
              гр_сущ1( Число, Падеж )
              местоимение( Число, Лицо, Падеж, &Род )
              составное_существительное( Число, Падеж );

   @Таблица( число Число = число Число )
   {  Ед  = Ед ;
      Мн = Мн ;
   }
}
// ------- а вот так делаются однородные члены с союзом "и"
// ------- аж три конструкции понадобились
@Перевод
{
   @Структура и__гр_сущ1 =
              @0[and]
              гр_сущ( &Число, &Лицо ) ;
   @Структура и__гр_сущ1( число &Число, лицо &Лицо, падеж &Падеж ) =
              @0[и]
              гр_сущ( Число, Лицо, Падеж ) ;
}
@Перевод
{
   @Выбор     и__гр_сущ =
              и__гр_сущ1
              @00 ;
```

```
   @Выбор     и__гр_сущ( число &Число, лицо &Лицо, падеж &Падеж ) =
              и__гр_сущ1( Число, Лицо, Падеж )
              @00 ;
}
@Перевод
{  // ------- дополнение (возможно с однородными членами ) -----------
   @Структура группа_существительного =
              гр_сущ   ( &Число ,&Лицо )
              и__гр_сущ ;
   @Структура группа_существительного( падеж &Падеж ) =
              гр_сущ   ( &Число1, 3е, Падеж )
              и__гр_сущ( &Число2, 3е, Падеж ) ;
   @Таблица( число Число = число Число )
   {  Ед  = Ед ;
      Мн = Мн ;
   }
}
@Перевод
{  // ------- подлежащее (возможно с однородными членами ) -----------
   @Структура подлежащее1 =
              гр_сущ( &Число, &Лицо )
              и__гр_сущ( &Число1, &Лицо1 ) ;
   @Структура подлежащее1( число &Число, лицо &Лицо )=
              гр_сущ   ( Число, Лицо, И )
              и__гр_сущ( Число, Лицо, И ) ;
}
@Перевод
{
   @Выбор     подлежащее =
              подлежащее1
              местимение( &Число, &Лицо );
   @Выбор     подлежащее( число &Число, лицо &Лицо )=
              подлежащее1( Число, Лицо );
              местимение( Число, Лицо, И, &Род ) ;
   @Таблица( @Выбор лицо Лицо = лицо Лицо )
   {  1 @0 = 3е ;
      2 1е = 1е ;
      2 2е = 2е ;
      2 3е = 3е ;
   }
   @Таблица( число Число = число Число )
   {  Ед  = Ед ;
      Мн = Мн ;
   }
}
// ----------------------------------------------------------------------
//                            сказуемое
// ----------------------------------------------------------------------
@Перевод
{  // ------- пример использования констант ------
   @Выбор     _Not = @0[not] @00 ;
```

```
   @Выбор     _Не =  @0[не]  @00 ;
}
@Перевод
{
   @Структура сказуемое_настоящее =
              глагол _Not ;
   @Структура сказуемое_настоящее( лицо &Лицо, число &Число ) =
              _Не глагол( Лицо, Число, настоящее, Актив ) ;
}
@Перевод
{
   @Структура сказуемое_прошлое =
               have _Not глагол ;
   @Структура сказуемое_прошлое( лицо &Лицо, число &Число ) =
              _Не глагол( Лицо, Число, прошлое, Актив );
}
@Перевод
{
   @Структура сказуемое_будущее =
               will _Not глагол ;
   @Структура сказуемое_будущее( лицо &Лицо, число &Число ) =
              _Не глагол( Лицо, Число, будущее, Актив ) ;
}
@Перевод
{
   @Структура сказуемое_модальное_настоящее =
               модальный_глагол _Not глагол  ;
   @Структура сказуемое_модальное( лицо &Лицо, число &Число ) =
              _Не модальный_глагол( Лицо, Число, настоящее, Актив )
              глагол( @0, @0, @0, Неопредел ) ;
}
@Перевод
{
   @Структура сказуемое_модальное_прошлое =
               модальный_глагол _Not have глагол ;
   @Структура сказуемое_модальное_прошлое( лицо &Лицо, число &Число ) =
              _Не модальный_глагол( Лицо, Число, прошлое, Актив )
              глагол( @0, @0, @0, Неопредел ) ;
}
@Перевод
{
   @Структура сказуемое_модальное_будущее =
               модальный_глагол _Not will глагол ;
   @Структура сказуемое_модальное_будущее( лицо &Лицо, число &Число ) =
              _Не модальный_глагол( Лицо, Число, будущее, Актив )
              глагол( @0, @0, @0, Неопредел ) ;
}
@Перевод
{  // ------- с аналитическими языкам всегда получаются вот такие "кусты"
   @Выбор     сказуемое =
              сказуемое_настоящее
```

```
              сказуемое_прошлое
              сказуемое_будущее
              сказуемое_модальное
              сказуемое_модальное_прошлое
              сказуемое_модальное_будущее ;
   @Выбор     сказуемое( лицо &Лицо, число &Число ) =
              сказуемое_настоящее          ( Лицо, Число )
              сказуемое_прошлое            ( Лицо, Число )
              сказуемое_будущее            ( Лицо, Число )
              сказуемое_модальное_настоящее( Лицо, Число )
              сказуемое_модальное_прошлое  ( Лицо, Число )
              сказуемое_модальное_будущее  ( Лицо, Число );
}
// ---------------------------------------------------------------------------
//         структуры, нужные для второстепенных членов предложения
// ---------------------------------------------------------------------------
@Перевод
{  @Структура предложный_оборот1 = @0[from] группа_существительного ;
   @Структура дополнение1        = группа_существительного( Р );
}
@Перевод
{  @Структура предложный_оборот2 = @[to] группа_существительного ;
   @Структура дополнение2        = группа_существительного( Д );
}
@Перевод
{  @Структура предложный_оборот3 = @[at] группа_существительного ;
   @Структура дополнение3        = группа_существительного( В );
}
@Перевод
{  @Структура предложный_оборот4 = @0[by] группа_существительного ;
   @Структура дополнение4        = группа_существительного( Т );
}
@Перевод
{  // ------- вот еще один "куст" ---------------------
   @Выбор     предложный_оборот =
              предложный_оборот1
              предложный_оборот2
              предложный_оборот3
              предложный_оборот4 ;
   @Выбор     предложный_оборот =
              дополнение1
              дополнение2
              дополнение3
              дополнение4 ;
}
// ----------------------------------------------------------
@Перевод
{
   @Структура неполная_фраза =
              сказуемое
              предложный_оборот<1>   // а вот появились смысловые индексы
```

```
             предложный_оборот<2> ;
   @Структура неполная_фраза =
             сказуемое( 3е,Ед )
             предложный_оборот<2>
             предложный_оборот<1> ;
}
@Перевод
{
   @Структура фраза_повествовательная =
             подлежащее
             сказуемое
             предложный_оборот<1>
             предложный_оборот<2> ;
   @Структура фраза_повествовательная =
             подлежащее( &Лицо, &Число )
             сказуемое( Лицо, Число )
             предложный_оборот<2>
             предложный_оборот<1> ;
}
@Перевод
{  // ------- общий вопрос ------------------
   @Структура фраза_вопросительная =
             сказуемое
             подлежащее
             предложный_оборот<1>
             предложный_оборот<2> ;
   @Структура фраза_вопросительная =
             подлежащее( &Лицо, &Число )
             сказуемое( Лицо, Число )
             @0[ли]
             предложный_оборот<2>
             предложный_оборот<1> ;
}
@Перевод
{  // ------- вопрос к подлежащему ---------
   @Структура фраза_вопросительная1 =
             вопрос_слово
             сказуемое
             предложный_оборот<1>
             предложный_оборот<2> ;
   @Структура фраза_вопросительная1 =
             вопрос_слово
             сказуемое( 3е,Ед )
             предложный_оборот<2>
             предложный_оборот<1> ;
}
@Перевод
{  // ------- вопрос к второстепенному члену предложения
   @Структура фраза_вопросительная2 =
             вопрос_слово
             сказуемое
```

```
              подлежащее
              предложный_оборот ;
   @Структура фраза_вопросительная2 =
              вопрос_слово
              подлежащее( &Лицо, &Число )
              сказуемое( Лицо, Число )
              предложный_оборот ;
}
// нужно определить повелительное сказуемое
@Перевод
{
   @Структура фраза_повелительная
              сказуемое
              предложный_оборот<1>
              предложный_оборот<2> ;
   @Структура фраза_повелительная
              сказуемое
              предложный_оборот<1>
              предложный_оборот<2> ;
}
@Перевод
{
   @Выбор     фраза
              фраза_повествовательная
              фраза_вопросительная
              фраза_вопросительная1
              фраза_вопросительная2
              фраза_повелительная ;
   @Выбор     фраза
              фраза_повествовательная
              фраза_вопросительная
              фраза_вопросительная1
              фраза_вопросительная2
              фраза_повелительная ;
}
// ---------------------------------------------------------------------
//            сложноподчиненные фразы
// ---------------------------------------------------------------------
@Перевод
{
   @Структура сложная_фраза_времени =
              @0[When] фраза_повествовательная @0[then] фраза ;
   @Структура сложная_фраза_времени =
              фраза @0[когда] фраза_повествовательная ;
}
@Перевод
{
   @Структура сложная_фраза_условия =
              @0[If] фраза_повествовательная @0[then] фраза ;
   @Структура сложная_фраза_условия =
              фраза @0[если] фраза_повествовательная ;
```

```
}
@Перевод
{
   @Структура сложная_фраза_цели =
              фраза @0[to] фраза_повествовательная ;
   @Структура сложная_фраза_цели =
              фраза @0[чтобы] фраза_повествовательная ;
}
@Перевод
{
   @Структура сложная_фраза_причины =
              фраза @0[because] фраза_повествовательная  ;
   @Структура сложная_фраза_причины =
              фраза @0[потому_что] фраза_повествовательная ;
}
// ---- все_все - это обязательно присутствующая конструкция
// ---- также, как функция main в языке C++
@Перевод
{
   @Выбор все_все =
          фраза
          сложная_фраза_времени
          сложная_фраза_условия
          сложная_фраза_цели
          сложная_фраза_причины ;
   @Выбор все_все =
          фраза
          сложная_фраза_времени
          сложная_фраза_условия
          сложная_фраза_цели
          сложная_фраза_причины ;
}
}
```

16 Заключение

Прочитал я этот файл... Если бы заранее всего не знал, ничего бы не понял...