

О генерации языковых процессоров на основе формальной спецификации лексики обрабатываемых текстов

Н.М. Мищенко

Институт кибернетики имени В.М. Глушкова НАН Украины

nady@dolphin.icyb.kiev.ua

Введение

В докладе рассматриваются средства автоматической генерации языковых процессоров для естественных языков. Эти средства реализованы в виде генератора языковых процессоров (ЯП) DUAL, который обеспечивает повторное использование в генерируемых ЯП ранее созданных программных компонентов по обработке текстов. ЯП, создаваемые с помощью генератора DUAL, предназначены для пословной обработки научно-технических и отраслевых текстов на флективных языках, в первую очередь, на русском и украинском, и ориентированы на профессионалов-нелингвистов. Для таких ЯП характерны небольшой объем словаря и сравнительно небольшое число случаев омонимии слов.

Главное условие применения генератора DUAL — наличие формальной спецификации процессов пословной обработки в ЯП, который генерируется. В качестве языка для разработки спецификаций предлагается декларативный язык Dual. Его назначение — описание лексики, содержащее необходимую для ее обработки информацию. Генераторы обеспечивают отладку генерируемых ЯП на уровне спецификаций.

Генератор DUAL включает систему универсальных (для класса флективных языков) программных компонентов пословной обработки текстов, словарь общеупотребительной лексики, а также программу, которая осуществляет специализацию универсальных компонентов путем пополнения словаря общеупотребительной лексики за счет словаря и сопутствующих структур данных, сгенерированных по формальной спецификации узкоспециальной лексики. Универсальные программные компоненты и расширенный словарь образуют ЯП для пользователей из конкретной области знаний.

При разработке генератора DUAL были применены средства генерации компиляторов [1] для формальных языков, в частности, языков программирования.

Таким образом, цель разработки генератора DUAL состояла в генерации работоспособных ЯП для естественных языков, а также в проверке на практике эффективности применения формальных средств для описания процессов пословной обработки научно-технических текстов на флективных языках.

Архитектура генератора DUAL

Основная идея применения генераторов вообще состоит в четком разделении средств для постановки задачи, то есть формулировки "что нужно сделать?" и средств, отвечающих на вопрос "как это сделать?". На первый вопрос отвечает декларативная спецификация, составляемая пользователем или при его участии. На второй отвечают процедуры обработки текстов, реализуемые разработчиками генераторов в невидимых пользователю универсальных программных компонентах. Созданные вручную при разработке первого ЯП, они затем включаются в генерируемые ЯП в требуемой конфигурации. Многократное использование универсальных программных компонентов повышает их надежность. Идею применения генератора DUAL как инструмента генерации ЯП для естественных языков представим описанием его архитектуры в виде пятерки составляющих разного назначения:

{ T, Flex, D, Dual, Gen }.

Опишем кратко каждую составляющую.

T — универсальные программные компоненты, выполняющие: морфологический анализ (МА), проверку согласования соседних словоформ, синтез выходного текста при переводе с одного флективного языка на другой, генерацию словоформ на основе словарного представления лексем. Составляющая открыта для пополнения новыми компонентами. Программа МА является основной в составляющей T, поскольку включается во все генерируемые ЯП. Она зависит от структуры словаря и сопутствующих таблиц, а не от конкретного наполнения этой структуры.

Flex — машинное представление окончаний флективных языков (русского и украинского), которое строится автоматически по текстовому представлению.

D — словари (машинные представления лексики) и сопутствующие структуры данных для русского и украинского языков. Они содержат общую лексику (общеупотребительную и устойчивые словосочетания). При генерации ЯП для специального языка словарь общей лексики расширяется специальной.

Dual — непроцедурный (декларативный) язык описания лексики для ее анализа и перевода. Синтаксис языка описывается контекстно-свободной грамматикой. Описание лексики некоторого языка на языке Dual — это текст, разделенный на так называемые словарные модули в соответствии с делением лексики на общую, отраслевую и узкоспециальную. Описание каждого модуля может быть расположено в нескольких текстовых файлах. Словарные модули отраслевой и узкоспециальной лексики создаются для каждого нового ЯП, поэтому построенные на их основе словари не входят в составляющую D.

Gen — собственно генератор, состоящий из программы MORF, генерирующей машинное представление системы окончаний (составляющая Flex) и программы CON — генератора словарей и сопутствующих структур данных по текстовой спецификации лексики.

Построение словаря для нового языка начинается обработкой программой Gen словарного модуля с описанием общей лексики этого языка. Результат включается в составляющую D. Дальнейшее наращивание словаря может происходить в нескольких направлениях за счет специальной лексики из различных областей знаний, представленной спецификациями. Так, если словарный модуль s — спецификация перевода специальной лексики с русского языка на украинский, Flex — таблицы окончаний для этих языков, а d — двуязычный словарь для

перевода общей лексики, то ЯП с именем DUET, осуществляющий русско-украинский перевод специальных текстов, можно представить тройкой:

{ Ts, Flex, ds },

где Ts — конфигурация программных компонентов из T, осуществляющих перевод по спецификации s, а ds — словарь d, расширенный за счет обработки спецификации s.

Отметим, что один и тот же ЯП может использовать различные профессиональные словари, поскольку они связываются с процессором через специальный параметр. В частности, DUET выполняет как русско-украинский так и обратный переводы в зависимости от вида подаваемых на его вход двуязычных словарей. В этом состоит мобильность генерируемых ЯП.

Спецификация системы окончаний

Прежде чем начать составлять спецификацию лексики для того или иного способа обработки текстов, необходимо определить распознаваемую программой МА структуру словоформ обрабатываемых текстов. Будем предполагать, что словоформы — это буквенные цепочки, разделенные в тексте пробелами, разделителями, формулами, числами, скобками, то есть, всем тем, что не подвергается морфологическому анализу.

С точки зрения программы МА словоформа, принадлежащая входному или выходному (при переводе) языкам состоит максимум из трех частей: основы — непустой общей начальной части словоформ, суффикса и окончания. Окончания в основном совпадают с каноническими. Они задаются списком. Если словоформа не имеет окончания, говорят, что она имеет нулевое окончание.

Суффикс — это часть словоформы, которая находится между основой и окончанием. Если суффикс является общим для всех словоформ одной и той же лексемы, его можно включить в основу. Суффикс может состоять из нескольких канонических суффиксов или не совпадать ни с одним из них. Список суффиксов и список основ, в отличие от окончаний, не составляются заранее, а формируются генератором в процессе обработки спецификаций, в которых они используются в явном виде.

Переходим к описанию понятий языка спецификаций Dual. Ориентация на пословную обработку текстов позволяет использовать в языке Dual только такие грамматические сущности флективных языков, которые изложены в учебниках и справочниках для средней школы, и, следовательно, должны быть известны пользователям-нелингвистам.

Конечную последовательность окончаний языка, принимаемых некоторым классом словоформ, будем называть кортежем окончаний, определяющим этот класс. Каждый класс словоформ получает уникальное мнемоническое имя — шифр класса, который одновременно является и шифром соответствующего кортежа. Он состоит из символов, с помощью которых задаются грамматические категории, присущие всем представителям класса и используемые в программах обработки текстов.

Для изменяемых слов первая буква шифра определяет часть речи: i — имя существительное, p — имя прилагательное, d — глагол и т. д. Следующие символы

определяют дальнейшую классификацию словоформ в пределах класса, заданного первым символом. Для имен существительных и прилагательных — это род: ж — женский, ч — мужской, с — средний. Если требуемых признаков недостаточно для однозначной идентификации кортежа, то шифр дополняется порядковым номером или окончанием основной словоформы. Например, пчдщий — шифр обозначает кортеж окончаний для имен прилагательных русского языка мужского рода, в роли которых выступают причастия на -щий (*несущий, идущий*). Шифры классов неизменяемых слов начинаются буквой н, за которой следует буква, обозначающая часть речи (класс слов): й — предлог, с — союз и т. д. Классы неизменяемых слов имеют пустые кортежи окончаний.

Нулевое окончание в кортеже представляется символом 0. Если слово не может иметь окончания в некоторой позиции кортежа, то в ней ставится точка. Например, имя существительное *программа* принадлежит классу имен существительных с таким кортежем (шифр іжа):

іжа: {-а, -ы, -е, -у, ., -ой, -е, -ы, 0, -ам, -ы, ., -ами, -ах};

Число позиций во всех непустых кортежах одинаково и равно 14. Для указания позиций используются специальные обозначения. В частности, позиции окончаний именных частей речи имеют такие обозначения:

{ он, ор, од, озн, озі, оо, ом, мн, мр, мд, мзн, мзі, мо, мм }.

Первая буква обозначения указывает категорию числа: о — единственное число, м — множественное число; вторая буква означает падеж: н — именительный, р — родительный и т.д. Для винительного падежа в обоих числах введено по две позиции для окончаний, соответственно, неодушевленных и одушевленных имен существительных. Обозначения этих позиций включают третью букву: н — для неодушевленных и і — для одушевленных имен существительных. В кортежах для глаголов позиции окончаний имеют другие обозначения:

{ он, она, оно, мы, они, , инф, , , , , , }

В кортежах для глаголов отсутствуют позиции для первого лица единственного числа и для второго лица единственного и множественного чисел на том основании, что в научно-технических текстах они встречаются редко. Поскольку реальных позиций окончаний в кортежах для глаголов меньше, чем для именных частей речи, то в "незанятых" позициях кортежа ставятся точки. Например, глагол **мочь** принимает такие окончания в указанных выше позициях:

дт_5: {-ет, -ет, -ет, -ем, -ут, ., -чь, ., ., ., ., ., ., .};

Хотя кортежи окончаний входного и выходного языков (для перевода) различны, их шифры могут совпадать. Принадлежность шифра входному или выходному языку легко распознается по месту их использования в спецификациях. Спецификация всех окончаний — это последовательность кортежей с шифрами, на основании которой программой MORF автоматически строятся такие структуры данных как список окончаний, древовидное представление окончаний (для МА), список значений омонимичных окончаний, представление последовательности кортежей в виде двумерного массива чисел.

Представление кортежей окончаний в виде двумерного массива чисел дает возможность в процессе МА определять окончание словоформы и порядковый номер кортежа, содержащего окончание, что обеспечивает доступ к любому окончанию кортежа, в том числе и к окончанию основной формы, например, при составлении частотного списка лексем текста.

Спецификации на языке Dual

Отбор лексики для спецификации основан на так называемой текстовой идеологии, суть которой состоит в том, что в словарь заносятся лексемы, представляющие словоформы, нераспознанные во время анализа текста. Такие словоформы заносятся в специальный список, где они представляется двумя частями: начальной частью и окончанием, которое может быть и ложным. Используя этот список, программа PHRASE строит частотный список основ, каждая из которых сопровождается списком окончаний всех нераспознанных словоформ с данной основой. На основании частотного списка основ выбираются первоочередные претенденты на введение в словарный модуль. Сгенерированные ЯП могут работать вообще без словаря, например, при обработке первого текста. В этом случае строится частотный список основ всех его словоформ.

Возможности языка Dual лучше всего показать на конкретных примерах спецификации пословного перевода. Спецификация лексем для тех способов обработки, где требуется только анализ словоформ, представляет собой частный случай спецификации пословного перевода.

Итак, спецификация перевода — это последовательность правил перевода, оканчивающихся символом ;. Каждое правило определяет перевод целого слова, нескольких словоформ, представленных в правиле общей основой, или фразы (сложные предлоги, устойчивые словосочетания). Такими цепочками входного языка начинаются правила. Символы входной цепочки, отличные от букв и пробела, берутся в кавычки. После входной цепочки в правиле следуют символы =>, за которыми стоит выходная цепочка-перевод в виде слова, основы нескольких словоформ или взятой в кавычки фразы выходного языка. Если выходное слово графически совпадает со входным, то на месте цепочки-перевода ставится символ *.

Рассмотрим несколько конкретных примеров спецификации перевода с русского языка на украинский.

1. Если переводится фраза или неизменяемое слово, то описание может закончиться указанием выходного слова или фразы. Примеры:

часто => * ;

всегда => *завжди* ;

в одно и то же время => *одночасно* ;

по"-*"видимому* => *очевидно* ;

2. Для успешного перевода чаще всего требуется некоторая дополнительная информация, которая размещается в правиле за выходной цепочкой, отделяясь от неё двоеточием. Эта информация образует в правиле так называемую схему перевода, в которой задействованы шифры кортежей окончаний, имена позиций в кортежах, суффиксы и превдосуффиксы. Как правило, схема перевода состоит из двух частей, разделенных знаком равенства. Информация слева от знака равенства относится ко входной цепочке, справа — к выходной. Если правая часть отсутствует, то знак равенства опускается. Начнем с простых правил:

согласно => *"згідно з"* : нй (од,мд) = нй (оо,мо);

В этом правиле для входного и выходного предлогов в скобках указаны падежи, которые принимают следующие за предлогами словоформы, входящие в группу имени существительного. В данном случае в обоих числах дательный падеж — для русского предлога, творительный — для украинского. Такие элементы схемы при распознавании предлогов образуют локальный контекст, который учитывается программой перевода для согласования имен существительных и прилагательных, а также для устранения омонимии окончаний.

3. Для изменяемых слов левая часть схемы содержит грамматические признаки словоформ с заданной в правиле основой, необходимые для их морфологического анализа. Правая часть содержит грамматические признаки выходных словоформ, необходимые для их синтеза. В случае альтернативных переводов правая часть отсутствует и выбор перевода осуществляется в процессе постредактирования. Чаще всего для перевода изменяемых слов требуются шифры кортежей окончаний:

множеств => множин : ісо = іжа_1 ;

Схема перевода в этом правиле указывает на то, что все словоформы имени существительного русского языка среднего рода с основой *множеств-*, для которых указан кортеж окончаний с шифром ісо, переводятся на украинский язык словоформами — именами существительными женского рода с основой *множин-* и с окончаниями в кортеже іжа_1.

4. Суффикс, употребляемый во всех словоформах, можно присоединить к основе, в противном случае следует указать явно условия его употребления:

готовност => готовн : іжъ = іжЗь_1 "іст" (он, озн, оо) "ост" ;

Чередование гласных в суффиксах выходных словоформ обусловило указание трёх падежей единственного числа, где употребляется суффикс -іст-. В остальных падежах употребляется суффикс -ост-.

5. Следующее правило определяет перевод словоформ, образованных от двух основных: *имеющийся* (перевод *наявний*) и *имеющий* (перевод *"що має"*). Указанные словоформы имеют общую основу, которая в словарном модуле должна встречаться только один раз. В таком случае задаются альтернативные переводы, каждый со своей схемой. Они разделены символом !. Схема первого перевода содержит три шифра, поскольку род имени прилагательного является изменяемым признаком, а причастие изменяется так как имя прилагательное.

Второй перевод сопровождается схемой без шифра в правой части правила, потому что имеющейся информации достаточно для синтеза выходной цепочки: все формы причастия *имеющий* в единственном числе переводятся *"що має"*, а во множественном — *"що мають"*. Окончания здесь используются как псевдосуффиксы, а цепочка (одн) обозначает все падежи единственного числа. При этом предполагается, что категория числа входной и выходной цепочки совпадает.

имеющ => наявн : пчдщий_1/пждщя_1/псдщее_1 = пчий_1/пжа_1/псе_1

! "що ма" : пчдщий/пждщя/псдщее = "є" (одн) "ють" ;

6. Описание перевода глагола после символа + может содержать указание на управление следующей за ним группой имени существительного:

приобрета => набува : дт = дт + (озн, мзн) = (ор, мр) ;

7. Если при переводе словоформы требуется произвести индивидуальную обработку, то в правиле следует указать имя соответствующего программного компонента:

ли => "@1чи" ;

Компонент с именем @1 выполняет перестановку в выходном тексте частицы *чи* с предыдущей словоформой, если она — глагол: *умолкнет ли — чи замовчить* (но *вряд ли — навряд чи*).

8. При одной и той же входной основе могут быть заданы разные переводы:

цел => *мет* : іжъ (одн) = іж1т_1 (одн)

! *цїл* : іжъ = іж3ь_1,

пчний/пжная/псное = пчий_1/пжа_1/псе_1 ;

Схема первого перевода разрешает переводить словоформы только в единственном числе: *цель, цели, целью* и т.д. Словоформы во множественном числе переводятся другими словоформами с основой *цїл-*. По этому же правилу, но по другой схеме будут переводиться и прилагательные *целый, целого* и т.д., поскольку они имеют общую входную и выходную основы.

Вообще говоря, используя суффиксы и псевдосуффиксы, можно описать одним правилом переводы всего подмножества словоформ той или иной лексемы, если словоформы имеют общую начальную часть. Однако это не всегда удобно. Поэтому рекомендуется совмещать описания переводов в одном правиле только в случае полного совпадения основ входных словоформ, как в случае *цель* и *целый*.

Особенность структуры словаря, построенного по формальной спецификации, состоит в том, что в нем основы разной длины представлены в виде отдельных подсловарей со ссылками на общие сопутствующие структуры данных в пределах одного языка. Сопутствующая информация о связях основ с суффиксами и окончаниями представлена структурами, которые по сути являются управляющими для программных компонентов и прежде всего для программы морфологического анализа, который выполняется путем идентификации частей словоформы поочередно с ее начала и с конца.

Первым шагом МА есть поиск текущей входной словоформы в словаре $ds(k)$, где k — длина словоформы. Если словоформа — неизменяемое слово или изменяемое с нулевым окончанием, то она должна быть найдена. Если распознанная словоформа имеет признак начала некоторой фразы, то следующие словоформы текста будут сравниваться со словарными, образующими фразу.

Если словоформа целиком не найдена в словаре, то в предположении, что последняя буква словоформы — окончание, будет сделана попытка отыскать однобуквенное окончание в таблице окончаний. В случае его нахождения выполнится поиск основы длины $k-1$ в словаре $ds(k-1)$.

Если основа длины $k-1$ не найдена, то делается попытка найти двухбуквенное окончание или суффикс. При этом всякий раз, когда остаток слова становится короче, он ищется в словаре основ меньшей длины. В итоге словоформа будет или распознана целиком, или поделена на части, или помечена как нераспознанная.

Для распознанной словоформы формируется ее информационная модель в соответствии с требованиями алгоритма дальнейшей обработки. После анализа и обработки всех

словоформ предложения осуществляются действия по формированию конечного результата на основе информационных моделей в зависимости от цели обработки, после чего начинается анализ словоформ следующего предложения.

Проблемы анализа и перевода

С помощью генератора DUAL был реализован ЯП DUET для перевода с русского языка на украинский и с украинского на русский [2]. В состав ЯП вошли программы МА, согласования словоформ, синтеза выходных словоформ, формирования списка отсутствующих в словаре слов. Результаты переводов признаны удовлетворительными, так как свыше 85% текста на выходном языке не требует постредктирования.

Такой результат автоматического перевода является показателем доли рутинного труда при переводе грамматически близких языков, передача которого программе была признана актуальной. DUET решает также проблему словаря, которая на практике оказалась сложнее ожидаемой. Это было обнаружено сравнением переводов, выполненными разными программами. В нашей системе работа по спецификации слов возведена в ранг творческой работы, результат которой сохраняется в виде текстового документа.

Остальная часть входного текста (менее 15%) не всегда может быть переведена. Это связано с локальностью области согласования словоформ, с омонимией словоформ и окончаний. В первом случае пользователю предлагается выбор переводов, а во втором — выбор окончаний. Не учтены в ЯП все случаи изменения структуры выходного текста по сравнению со структурой входного. Здесь мы придерживаемся принципа: лучше недоделать, чем сделать неправильно.

Используя генератор, удалось расширить круг генерируемых ЯП. В частности, были сгенерированы процессоры: FEST — для определения ключевых слов специализированных текстов [3], TEST — для генерации всех словоформ по внутреннему представлению словаря лексем. Для получения этих ЯП доля ручного труда была минимальной, поскольку в них использовались компоненты генератора, разработанные для ЯП DUET.

В заключение отметим следующее.

Использование сгенерированных ЯП свидетельствует о целесообразности применения формальных средств спецификации процессов пословной обработки специальных текстов на флективных языках.

Составление спецификаций — процесс, требующий больших затрат времени. Разработанные средства автоматизации составления спецификаций и их верификации — предмет отдельного сообщения.

Следует отметить также роль генератора компиляторов для формальных языков с именем РСП Терем [1] в разработке языковых процессоров для естественных языков. Прежде всего это общность архитектуры генераторов РСП Терем и DUAL. Отличия — в наполнении составляющих архитектур. С помощью РСП Терем был построен генератор Gen (обе его программы MORF и CON), поскольку формальный язык Dual принадлежит классу допустимых в РСП Терем контекстно-свободных языков.

Литература

Мищенко Н.М. Про повторне використання засобів генерації мовних процесорів // Праці І-ї Міжн. н/п конф. УкрПРОГ'98. (Київ, 2-4 вересня 1998 р.) — Київ, 1998. — С. 462-467.

Мищенко Н.М. Мобильная система пословного перевода научных текстов на флективных языках на основе формальных спецификаций. // Кибернетика и системный анализ. — 1999. — N 1. — С.33-42.

Мищенко Н.М., Федюрко В.В., Щоголева Н.М. Мобильная система статистической обработки профессиональных текстов в целях определения их тематики // Проблемы программирования. — 1999. — N2. С. 11-18.