МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет аэрокосмического приборостроения»

ФУНКЦИОНАЛЬНОЕ И ЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

ОБРАБОТКА ЛИНГВИСТИЧЕСКИХ ДАННЫХ СРЕДСТВАМИ PROLOG

Методические указания к выполнению лабораторной работы

Санкт-Петербург

2016

Составитель: Т.М.Максимова.

Методические указания содержат краткое описание специализированных средств Prolog для обработки лингвистических данных и задания на лабораторную работу, предназначены для студентов специальностей 09.03.04- Программная инженерия (бакалавр), 02.03.03 - Математическое обеспечение и администрирование информационных систем (бакалавр), изучающих дисциплину «Функциональное и логическое программирование».

1. Задание на лабораторную работу

Средствами DCG в среде SWI-Prolog разработайте программу, выполняющую синтаксический разбор предложений русского языка. Результат работы программы должен быть представлен перечнем членов предложения с указанием для каждого слова предложения, какой частью речи оно является. Тип предложений, которые предстоит анализировать программе, выберите по номеру варианта задания.

№ варианта	Тип предложений
1	Простые двусоставные распространённые предложения, допускающие простые и составные глагольные сказуемые
2	Простые двусоставные распространённые предложения, допускающие простые и составные именные сказуемые
3	Простые двусоставные распространённые предложения, допускающие однородные подлежащие или однородные сказуемые
4	Сложносочинённые предложения
5	Сложноподчинённые предложения с подчинительными союзами из фиксированного перечня не менее двух
6	Простые двусоставные распространённые предложения, допускающие однородные обстоятельства из фиксированного перечня типов не менее двух
7	Простые односоставные распространённые предложения
8	Простые двусоставные распространённые предложения с несколькими (не менее трёх) способами выражения подлежащего
9	Сложноподчинённые предложения с союзными словами из фиксированного перечня не менее двух
10	Простые распространённые и сложносочинённые предложения с одинаковыми союзами

2. Пояснения к терминологии, используемой в описании синтаксиса русского языка

Простое предложение содержит одну грамматическую основу, основную часть предложения, формируемую из подлежащего и сказуемого.

Односоставное предложение содержит только один главный член предложения: подлежащее или сказуемое.

Двусоставное предложение содержит два главных члена предложения: подлежащее и сказуемое.

Распространённое предложение, помимо главных членов предложения, содержит некоторое количество (не менее одного) второстепенных членов предложения (определения, дополнения, обстоятельства).

Составное глагольное сказуемое состоит из вспомогательной части (глагол в спрягаемой форме) и основной части (инфинитив).

Составное именное сказуемое состоит из вспомогательной части (глагол в спрягаемой форме) и основной части (существительное, прилагательное, местоимение, числительное или наречие).

Обстоятельство – второстепенный член предложения, который обозначает место, время или способ действия предмета, что и определяет тип обстоятельства.

Некоторые **способы выражения подлежащего**: существительное, местоимение, прилагательное в роли существительного, причастие в роли существительного, числительное, наречие, инфинитив.

Союзные слова служат средствами связи придаточного предложения с основным в сложноподчинённых предложениях и одновременно (в отличие от союзов) являются членами предложений. В роли союзных слов могут выступать местоимения (какой, который, кто, что, кем, чем, кого, сколько) и местоименные наречия (где, куда, откуда, как, когда, зачем, почему, отчего).

Более подробную информацию на эту тему можно почерпнуть в [2], а почти исчерпывающую – в [3] и [4].

3. Дополнительные средства Prolog для описания грамматик

В некоторых реализациях Prolog имеется возможность выполнять описание синтаксиса какого-либо языка в системе обозначений, близкой к форме Бэкуса-Наура. Такое расширение Prolog известно под названием DCG (Definite Clause Grammar) [1]. Особенности обозначений, принятых в DCG, заключаются в следующем:

1) вместо символа «::=» для разделения левой и правой частей правила грамматики используется «-->» ;
2) символы правой части правила отделяются друг от друга запятыми, а завершается правая часть правила точкой;
3) терминальные символы грамматики помещаются в квадратные скобки.

Так, например, язык строк круглых скобок, расставленных в соответствии с правилами арифметических выражений, средствами DCG можно описать так (грамматика приведена к виду, не содержащему леворекурсивных правил):

```
a-->t,z.
z-->t,z.
z-->[].
t-->['('],[')'].
t-->['('],a,[')'].
```

Такой текст рассматривается системой Prolog как последовательность утверждений (clauses), которая может использоваться для формирования ответов на запросы (goal). Это формирование, естественно, является не чем иным, как синтаксическим анализом строки терминальных символов, предложенной в запросе. Запрос на синтаксический анализ строки записывается в виде структуры с нетерминальным символом в качестве главного функтора и с двумя компонентами-списками. В первом списке размещается предназначенная для анализа строка, второй список можно оставлять пустым. При этом надо иметь в виду, что символы анализируемой строки рассматриваются Prolog-анализатором как элементы списка, и поэтому их следует разделять запятыми, что, к сожалению, несколько портит внешний вид строки. Так, например, запрос на синтаксический анализ строки «(()())», с учётом приведённых выше обозначений, может быть следующим:

```
a(['(','(',')','(',')',')'],[]).
```

Ответ SWI-Prolog на этот запрос изображается словом «true». Соответственно, запрос, содержащий строку-список, не соответствующую правилам грамматики, приводит к ответной реакции в виде слова «fail». Заметим коротко, что необходимость указания двух списков в запросе связана с внутренним представлением анализируемых строк в виде т.н. разностных списков [1]. Список, подвергаемый анализу, – это разность первого и второго списков, указанных в запросе. Поэтому возможны варианты запроса вроде такого: a(['(','(',')','(',')',')',x,y,z],[x,y,z]).

В рассмотренном примере внешний вид строки «пострадал» ещё и от необходимости заключать каждый её символ в апострофы. К счастью, присутствие апострофов не является общим требованием для изображения строк всех формальных языков, а связано только с использованием в алфавите языка специальных символов, таких, например, как знаки препинания (в данном случае – скобки).

Обратимся теперь к примеру языка, более близкого к естественному.

```
предложение-->подлежащее,сказуемое.
предложение-->сказуемое,подлежащее.
подлежащее-->местоимение.
местоимение-->[я].
сказуемое-->глагол.
глагол-->[пишу].
```

Такое описание языка не возлагает на Prolog-программу функцию лексического анализа: лексемы (слова) явно присутствуют в правилах грамматики в виде терминальных символов. Оба списка, и [я], и [пишу], являются одноэлементными.

Ответ «true» будет получен в двух вариантах запроса:

1) предложение([я, пишу],[]).

2) предложение([пишу,я],[]).

Для возможности выражения средствами языка того, что писать могу не только «я», но и «мы», «ты», «они» и т.д., грамматику следует дополнить, во-первых, лексикой личных местоимений и форм глагола, а во-вторых – согласованием местоимений с формами глагола. Первое из этих дополнений, расширение словарного запаса языка, может быть выполнено только бесхитростным перечислением всех новых слов. Второе дополнение для данного примитивного примера, разумеется, можно выполнить так же: перечислить явно все предложения языка (в отсутствие рекурсивных правил их количество конечно). Однако в этом случае не придётся претендовать на что-то большее в ответе Prolog-анализатора, чем «true» или «fail», поскольку ответ будет формироваться простым перебором предложений без анализа (разбора) их структуры.

В решении проблемы согласования местоимений с формами глагола могут оказать помощь Prolog-переменные. Областью видимости переменной является только одно утверждение (clause) программы. Это справедливо и для правил DCG-грамматики, в которых нетерминальные символы могут рассматриваться как имена предикатов, и эти предикаты могут иметь аргументы. Именно такая интерпретация нетерминального символа позволяет сформировать запрос на анализ строки: в запросе «предложение([я, пишу],[]).» используется предикат «предложение» с двумя аргументами. Для согласования подлежащего и сказуемого в предложении языка введём переменные «Лицо» и «Число» в соответствующих предикатах. Тогда правила для предложений с согласованными подлежащими и сказуемыми примут вид:

предложение-->подлежащее(Лицо,Число),сказуемое(Лицо,Число).

предложение-->сказуемое(Лицо,Число),подлежащее(Лицо,Число).

А для нетерминалов «местоимение» и «глагол» теперь следует добавить правила в соответствии со спряжением глагола «писать» (ограничимся настоящим временем этого глагола):

местоимение(1,единственное) -->[я].
местоимение(2,единственное) -->[ты].
местоимение(3,единственное) -->[он].
местоимение(1,множественное) -->[мы].
местоимение(2,множественное) -->[вы].
местоимение(3,множественное) -->[они].
глагол(1,единственное) -->[пишу].
глагол(2,единственное) -->[пишешь].
глагол(3,единственное) -->[пишет].
глагол(1,множественное) -->[пишем].
глагол(2,множественное) -->[пишете].
глагол(3,множественное) -->[пишут].

Члены предложения подлежащее и сказуемое получат значения переменных согласования от соответствующих частей речи – местоимения и глагола:

подлежащее(Лицо,Число)-->местоимение(Лицо,Число).

сказуемое(Лицо,Число)-->глагол(Лицо,Число).

Расширенная грамматика позволяет синтаксическому анализатору «узнавать» 12 различных предложений, в которых подлежащие-местоимения согласованы со сказуемыми-глаголами. Но внешний результат «узнавания» или «неузнавания», по-прежнему, двоичен: «true» или «fail». Для того, чтобы сделать видимым распределение ролей членов предложения между словами предложения, следует опять прибегнуть к помощи дополнительных переменных. Добавим такие переменные в правила для предложений:

предложение(Подлежащее, Сказуемое) --> подлежащее(Подлежащее,Лицо,Число), сказуемое(Сказуемое,Лицо,Число).

предложение(Подлежащее,Сказуемое)-->сказуемое(Сказуемое,Лицо,Число), подлежащее(Подлежащее,Лицо,Число).

Добавим их и в правила для подлежащего и сказуемого, не заботясь о придании смысла именам переменных, поскольку они приобретают значения промежуточных результатов разбора, и это можно считать внутренним делом программы:

подлежащее(А,Лицо,Число)-->местоимение(А,Лицо,Число).
сказуемое(А,Лицо,Число)-->глагол(А,Лицо,Число).

Конкретизироваться значениями слов эти переменные будут при работе синтаксического анализатора с правилами для частей речи – местоимения и глагола:

местоимение(я,1,единственное) -->[я].
глагол(пишу,1,единственное) -->[пишу].
и т.д.

На запрос

предложение(Подлежащее,Сказуемое,[я,пишу],[]).

ответ новой программы выглядит так:

Подлежащее = я,
Сказуемое = пишу.

Сделать форму ответа ещё красивее, выполнив одновременно требование задания на лабораторную работу об указании для каждого слова, какой частью речи оно является, можно использованием в качестве значений переменных структур с именами (главными функторами), совпадающими с названиями частей речи:

местоимение(местоимение(я),1,единственное) -->[я].
глагол(глагол(пишу),1,единственное) -->[пишу].
и т.д.

С учётом этих исправлений в программе её ответ приобретёт вид:

Подлежащее = местоимение(я),
Сказуемое = глагол(пишу).

Если какой-либо член предложения выражается несколькими словами, значения соответствующих переменных можно определить списками структур. Так, если рассматриваемый пример расширить введением такого второстепенного члена предложения, как дополнение, образуемое предлогом и существительным, грамматику можно дополнить правилами (ограничимся, например, двумя падежами существительного):

```
дополнение([Предлог,Существительное])-->предлог(Падеж,Предлог),
существительное(Падеж, Предлог).
дополнение(Существительное)-->
существительное(винительный,Существительное).
предлог(предложный,предлог(о)) -->[о].
существительное(винительный,существительное(программу)) -->[программу].
существительное(предложный,существительное(программе)) -->[программе].
```

Соответственно должно измениться и правило для предложения, ставшего распространённым:

```
предложение(Подлежащее,Сказуемое,Дополнение)-->
подлежащее(Подлежащее,Лицо,Число),
сказуемое(Сказуемое,Лицо,Число),дополнение(Дополнение).
```

На запрос с распространённым предложением:

```
предложение(Подлежащее,Сказуемое,Дополнение,[я,пишу,программу],[]).
```

реакция программы выглядит так:

```
Подлежащее = местоимение(я),
Сказуемое = глагол(пишу),
Дополнение = существительное(программу).
```

Для предложения, в котором дополнение представлено двумя словами:

```
предложение(Подлежащее,Сказуемое,Дополнение,[я,пишу,о,программе],[]).
```

значение переменной Дополнение представляется списком:

```
Подлежащее = местоимение(я),
Сказуемое = глагол(пишу),
Дополнение = [предлог(о),существительное(программе)]
```

Заметим, что приведённое в примере правило использования существительного в винительном падеже без предлога, а в предложном падеже – только с предлогом «о», разумеется, не исчерпывает возможностей русского языка, касающихся всех предлогов и их сочетаний с существительными в этих падежах, а продиктовано лишь соображением компактности примера.

Из недостатков интерфейса DCG, особенно – в применении к естественным языкам, следует отметить то, что наглядность представления анализируемого текста существенно страдает от необходимости записи этого текста в виде списка слов, разделённых запятыми. Этот недостаток не очень сложно исправить, дополнив программу предикатами, позволяющими для строки текста, написанного в соответствии с правилами орфографии и пунктуации русского языка, построить список слов и знаков препинания, пригодный для использования его в Prolog-запросе к грамматике. Оставим эту задачу для самостоятельного её решения студентами, желающими получить **повышенную оценку при защите лабораторной работы.** Возможно, нелишней для таких студентов окажется следующая подсказка. Для преобразования списка символов в атом и наоборот – атома в список символов – в SWI-Prolog имеется встроенный предикат atom_chars(Atom,List). Примеры его использования «в ту и другую стороны»:

1) запрос

atom_chars(A,[a,b,c]).

ответ

A = abc.

2) запрос

atom_chars(abc,A).

ответ

A = [a, b, c].

В заключение краткого изложения возможностей средств DCG следует сказать, что они «в содружестве» с принципами логического Prolog-программирования представляют удобный инструмент для программной реализации синтаксически управляемых процессов обработки текстов. Однако при работе с этим инструментом надо помнить, что в основу алгоритма синтаксического анализа здесь положен backtracking со всеми вытекающими из этого факта последствиями. В частности, алгоритм существенно уступает по времени детерминированным автоматным моделям. Кроме того, поскольку backtracking – это реализация нисходящего разбора, предлагаемые ему грамматики не должны содержать леворекурсивных правил.

4. Библиографический список

1. И.Братко. Алгоритмы искусственного интеллекта на языке PROLOG. – М.:Издательский дом «Вильямс», 2004. – 640с.
2. О.Д.Ушакова. Синтаксический разбор предложения. – СПб.: Издательский дом «Литера», 2014. – 96с. – (Серия «Словарик школьника»).

3. http://gramma.ru/
4. http://www.gramota.ru/

Оглавление