

Введение в робототехнику

Лекция 11 Моделирование понимания ЕЯ

НГТУ, кафедра ПТМ
А.В.Гаврилов

«Понимающий» компьютер должен осуществлять обработку всех уровней языка

- Фонетика (при голосовом общении)
- Морфология
- Синтаксис
- Семантика
- Прагматика
- Дискурс

Решается только одна проблема:
НЕОДНОЗНАЧНОСТЬ

Два подхода к моделированию понимания смысла ЕЯ

- *Синтаксически-ориентированный подход* основан на детальном синтаксическом разборе предложения. Средствами синтаксического анализа вычленяются связанные понятия, которые объединяются в так называемые атомы смысла (АС). Создание АС идет только на основе данного предложения, определение связанных понятий идет только на основе синтаксических правил.
- *Семантически-ориентированный подход* на основе распознавания семантики. Разбор предложения идет путем вычленения связанных понятий с помощью базы знаний. База знаний хранит АС и определенным образом представленные связи между этими АС. На втором плане стоит синтаксический анализ с помощью которого вычленяются дополнительные АС, те которые не были сгенерированы с помощью БЗ.

Технологии анализа ЕЯ. Синтаксический анализ.

- **Парсинг** – процесс структурирования линейной репрезентации в соответствии с заданной грамматикой
- **Линейной репрезентацией** предложения естественного языка называется цепочка элементов, где каждый элемент является минимальной синтаксической единицей

5 языковых средств синтаксического анализа

1. Словоизменительные морфологические средства

- *w1 зависит от w2 по C, если граммема g категории C, характеризующая w1 выбирается в зависимости от слова w2*
- *В русском языке к словоизменительным категориям относятся категории падежа и числа существительного; категория падежа числительного; категории рода, числа, падежа и степени сравнения прилагательного; категории лица, числа, времени, наклонения и рода (в прош. вр. и сослагат. накл.) глагола; категория степени сравнения наречия*
- *пример: **новые книги***

5 языковых средств синтаксического анализа (2)

2. Селективные признаки

- Частеречные признаки (например, *er* в конце слова в англ. языке)
- Одушевленность

3. Служебные слова (*в, к, при, и, а, или, бы, же, уж, в течение; несмотря на то что; пускай, давай*)

4. Знаки препинания

5. Порядок слов

Формализмы

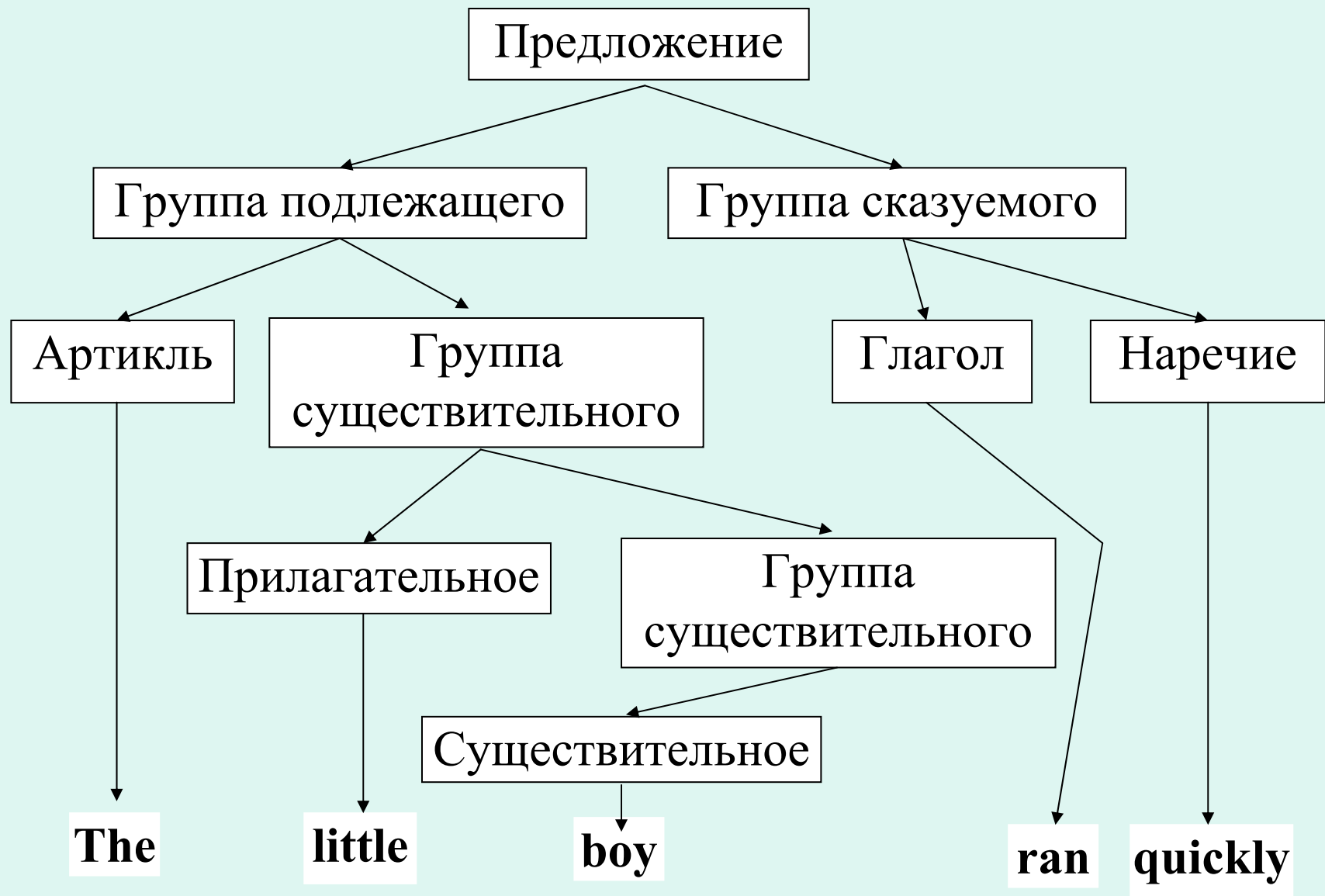
- Контекстно-свободные грамматики (грамматика составляющих)
- Head-driven phrase structure grammar (HPSG)
- Грамматика зависимостей
- Грамматика связей (Link Grammar)
- LR-грамматтики

Грамматический разбор

При разборе мы имеем дело с *грамматическими категориями*:

‘предложение’, ‘группа существительного’, ‘группа сказуемого’, ‘существительное’, ‘глагол’, ‘наречие’ и т. д. и пользуемся собственно *словами*, составляющими разбираемое предложение.

Например, структуру английского предложения: “The little boy ran quickly” можно изобразить в виде диаграммы.



Синтаксическая структура предложения

Правила грамматики

Грамматический разбор предложений подразумевает использование *правил* некоторой грамматики. Мы их будем представлять в следующей форме (приведены не все правила грамматики):

< предложение > → < группа подлежащего >

< группа сказуемого >

< группа подлежащего > → < артикль >

< группа

существительного >

< группа существительного > → < существительное >

< группа сказуемого > → < глагол > < наречие >

< артикль > → The

< прилагательное > → little

< существительное > → boy

< глагол > → ran

< наречие > → quickly

Механизм порождения

Здесь стрелочка \rightarrow отделяет *левую часть правила* от *правой*, а грамматические термины заключены в *металингвистические скобки* $<$ и $>$ для того, чтобы отличать их от слов, составляющих разбираемое предложение.

По этим правилам можно не только *проверять грамматическую правильность предложений*, но также *порождать грамматически правильные предложения*.

Механизм порождения

Механизм порождения

Начиная с цепочки, включающей только грамматический термин, являющимся *главным* (< предложение >), каждый грамматический термин, входящий в текущую цепочку, замещается правой частью того правила, которое содержит его в левой части. Когда в результате таких замен в текущей цепочке не останется ни одного термина грамматики, а только слова языка, мы получаем *грамматически правильное предложение* языка.

Грамматика.

Язык, порождаемый грамматикой

Ранее речь шла о конкретной грамматике. В ней имеются два словаря:

1) *нетерминалы* — грамматические термины

<предложение>, <группа подлежащего >, ...;

2) *терминалы* — слова, составляющие предложения языка

The, little, boy, ran, quickly;

Грамматики

3) *правила, левые и правые* части которых состоят из нетерминалов и терминалов;

<предложение> → < группа подлежащего >

< группа сказуемого >

< артикль > → The, ...

4) *начальный нетерминал* — главный грамматический термин; из него выводятся те цепочки терминалов, которые считаются предложениями языка

<предложение>

Основные виды грамматик

- Контекстно-свободная грамматика – у которой в левой части правил содержится только один нетерминал
 $A \rightarrow a, b, c.$
- Контекстно-зависимая грамматика – у которой в левой части правил может содержаться помимо нетерминала и терминалы
 $Ad \rightarrow a, b, c$
- Регулярная грамматика – у которой правая часть каждого правила начинается с терминала
 $A \rightarrow aB$

Виды синтаксического анализа (грамматического разбора)

- **Сверху вниз**
 - Программа пытается породить, начиная с главного правила (описывающего структуру предложения) разбираемое предложение (последовательность терминалов)
- **Снизу вверх**
 - Программа пытается на основе текущего слова в предложении (и может быть следующих слов) распознать нетерминалы и в конце концов предложение в целом

Технологии анализа ЕЯ. Синтаксический анализ. Свободно-контекстные грамматики.

Недостатки:

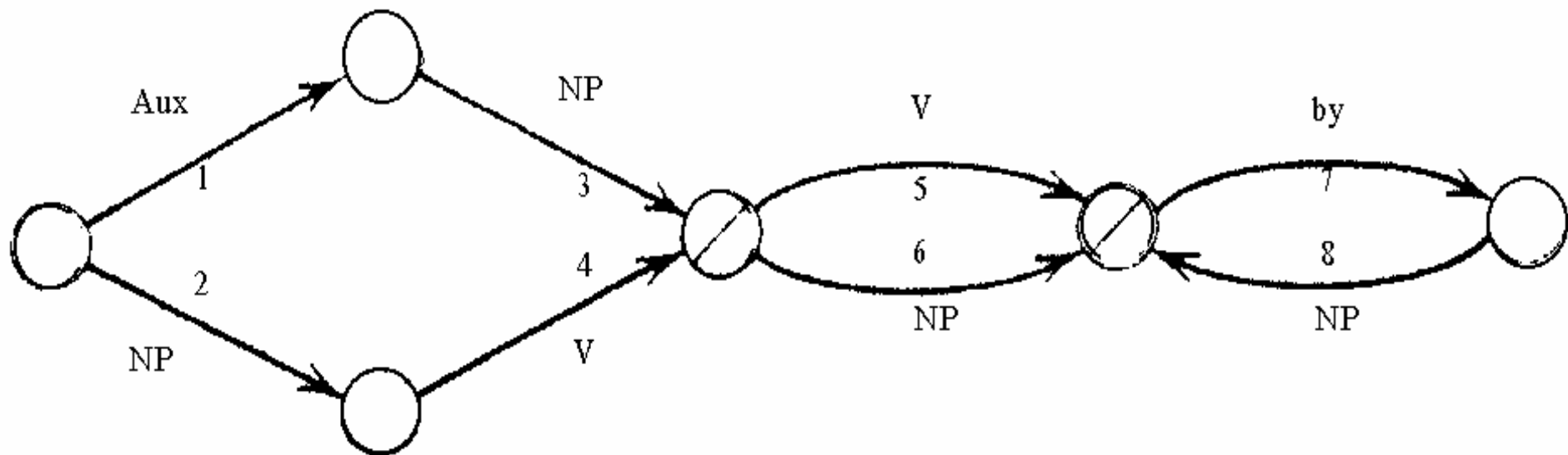
- отсутствие запрета на грамматически неправильные фразы, где, например, подлежащее не согласовано со сказуемым в числе,
- Разрастание грамматики для всех вариантов разбора, в том числе, грамматически неправильных фраз,
- Следствие – непригодны для анализа ЕЯ.

Технологии анализа ЕЯ. Синтаксический анализ. Трансформационные (генеративные) грамматики.

- Автор – Хомский,
- Для порождения грамматически правильных предложений,
- Центральная идея трансформационной теории состоит в том, что поверхностные формы любого языка - его предложения - являются результатом взаимодействия между несколькими модульными подсистемами
- Трансформационные правила предназначены для описания систематических отношений в предложении, как то:
 - отличия между активным и пассивным предложением
 - глобальные отношения в предложении (например, связь между what и eat в предложении "What will John eat")
 - неоднозначности, причиной которых является одна и та же форма предложения, выведенная из двух различных базовых предложений (например, в предложении "They are flying planes" flying можно рассматривать и как прилагательное и как основной глагол)
- Непригодны для анализа ЕЯ.

Технологии анализа ЕЯ. Синтаксический анализ. Расширенная сеть переходов.

- Авторы – Бобров, Фрейзер и Вудс,
- Продолжение свободно-контекстных грамматик.
- *The rabbit nibbles the carrot* (Кролик грызет морковь).



Технологии анализа ЕЯ. Синтаксический анализ. Расширенная сеть переходов (2)

Недостатки:

- немодульность;
- сложность при модификации, вызывающая непредвиденные побочные эффекты;
- хрупкость (когда единственная неграмматичность в предложении делает невозможным дальнейший правильный анализ);
- неэффективность при переборе с возвратами, т.к. ошибки на промежуточных стадиях анализа не сохраняются;
- неэффективность с точки зрения смысла, когда с помощью полученного синтаксического представления оказывается невозможным создать правильное семантическое представление.

Технологии анализа ЕЯ. Семантические грамматики.

- Вместо синтаксических категорий в грамматике используются семантические,
- Используются в узких предметных областях,
- Недостатки:
 - для каждой предметной области нужна своя грамматика,
 - Имеют тенденцию быстро увеличиваться в размерах.

S → <present> the <attribute> of <ship>

<present> → what is | [can you] tell me

<ship> → the <shipname> | <classname> class ship

Can you tell me the class of the Enterprise? (Enterprise

- название корабля).

НГТУ, кафедра ПТМ
А.В.Гаврилов

Технологии анализа ЕЯ. Подбор шаблона.

Диалог программы Eliza, имитирующей психиатра:

X всегда Y: Вы можете привести какой-нибудь пример?
Когда?
Действительно, всегда?

X Вы Y меня: Почему Вы думаете, что я Y Вас?
Вам нравится думать, что я Y Вас, не правда ли?
Представьте, что я Y Вас. Что Вы об этом думаете?

В результате *Элиза* способна вести такой диалог:

Пац.: Люди всегда пристально смотрят на меня.	подбор по 1-му шаблону с учетом <i>всегда</i>
Эл.: Вы можете привести какой-нибудь пример?	первый ответ по 1-му шаблону
Пац.: Например, сейчас ты смотришь на меня.	подбор по 2-му шаблону, Y= <i>смотреть</i>
Эл.: Почему Вы думаете, что я <u>смотрю</u> на Вас?	первый ответ по 2-му шаблону
Пац.: Потому что ты ненавидишь меня.	подбор по 2-му шаблону, Y= <i>ненавидеть</i>
Эл.: Вам нравится думать, что я <u>ненавижу</u> Вас, не правда ли?	второй ответ по 2-му шаблону

Язык шаблонов AIML – язык для Alice-подобных диалоговых систем

- AIML – язык разметки
- Набор категорий <Category>
- Категория состоит из
 - Паттернов <pattern>
 - Шаблонов <template>
- При анализе предложения ЕЯ интерпретатор AIML ищет подходящий паттерн, который сопоставим с предложением, а в ответ выдает один из соответствующих шаблонов
- Имеется возможность запоминать слова из предложения и использовать их в виде переменных в шаблонах и при проверке условий

Пример

```
<category>
```

```
<pattern>ПОДНИМИ * СО СТОЛА</pattern>
```

```
<template>Поднимаю, <star/>.
```

```
<think>
```

```
<set name="object"><star/></set>
```

```
</think>
```

```
</template>
```

```
</category>
```

- Тег <star/> заменяется на слово, соответствующее символу * в образце <pattern>, а object – имя переменной для запоминания этого слова с целью использования в дальнейшем. Для сохранения переменной используется тег <set name="object">.

Пример (2)

```
<category>
<pattern>ЕСТЬ ЛИ ДЕТАЛИ В ЗАХВАТЕ</pattern>
<template>
<condition>
<li name="object" value="">Нет. Я пуст</li>
<li>Я поднял <get name="object"/></li>
</condition>
</template>
</category>
```

- первая строка после `<condition>` кодирует условие «переменная *object* имеет значение *пусто*», вторая строка выполняется если переменная *object* имеет какое-то значение. В ответ на вопрос пользователя робот сообщит о наличии в его захвате детали или об отсутствии детали в захвате.

Технологии анализа ЕЯ. Падежные фреймы.

- Падежный фрейм состоит из заголовка и набора ролей (падежей), связанных определенным образом с заголовком

[Заголовочный глагол

[падежный фрейм

агент: <активный агент, совершающий действие>

объект: <объект, над которым совершается действие>

инструмент: <инструмент, используемый при совершении действия>

реципиент: <получатель действия - часто косвенное дополнение>

направление: <цель (обычно физического) действия>

место: <место, где совершается действие>

бенефициант: <сущность, в интересах которой совершается действие>

коагент: <второй агент, помогающий совершать действие>

]]

Технологии анализа ЕЯ. Падежные фреймы.

Например,
для фразы *Иван дал мяч Кате* падежный фрейм
выглядит так:

[Давать

[падежный фрейм

агент: Иван

объект: мяч

реципиент: Катя]

[грам

время: прош

залог: акт]

]

Технологии анализа ЕЯ. Падежные фреймы.

Анализ текста с помощью падежных фреймов состоит из следующих шагов:

- Используя существующие фреймы, подобрать подходящий для заголовка. Если такого нет, текст не может быть проанализирован.
- Вернуть в систему подходящий фрейм с соответствующим заголовком-глаголом.
- Попытаться провести анализ по всем обязательным падежам. Если один или более обязательных заполнителей падежей не найдены, вернуть в систему код ошибки. Такой случай может означать наличие эллипсиса, неверный выбор фрейма, неверно введенный текст или недостаток грамматики. Следующие шаги используются уже для анализа и исправления таких ситуаций.
- Провести анализ по всем необязательным падежам.
- Если после этого во введенном тексте остались непроанализированные элементы, выдать сообщение об ошибке, связанной с неправильным вводом, недостаточностью данного анализа или необходимостью провести другой, более гибкий анализ.

Технологии анализа ЕЯ. Падежные фреймы.

Преимущества использования падежных фреймов таковы:

- совмещение двух стратегий анализа (сверху вниз и снизу вверх);
- комбинирование синтаксиса и семантики;
- легкая встраиваемость в интеллектуальные системы на основе фреймов;
- удобство при использовании модульных программ.

Растущие семантические сети.

Вариант из работ:

1. Гаврилов А.В. Диалоговая система подготовки программ для роботов. - В сб. "Диалоговые системы в задачах управления", Новосибирск, НЭТИ, 1987.
2. Гаврилов А.В. Диалоговая система подготовки программ для роботов. - В сб. Automatyka, v.99, Gliwice, 1988, p.173-180.
3. Гаврилов А.В. Архитектура программного обеспечения для поиска документов по запросу на естественном языке. - Труды Межд. конф. KDS-2001 "Знание-Диалог-Решение". С.-Петербург, 2001. - Т.1, с. 124-130.
4. Gavrilov A.V. A combination of Neural and Semantic Networks in Natural Language Processing. - // Proc. of the 7th Korea-Russia Int. Symp. KORUS-2003, Ulsan, 2003. – Vol. 2, Pp. 143-147.

- автор исходит из того, что при обучении языковым понятиям осуществляется запоминание двух основных типов ассоциаций:
 - неявно задаваемые ассоциации между словами, встречающимися рядом в тексте, формирующие понятия, связанные со словосочетаниями,
 - явно задаваемые ассоциации между группами слов, задающие смысл новых слов и словосочетаний в контексте того, что уже известно интеллектуальной системе.
- Далее ассоциативными (горизонтальными) связями будем называть только ассоциации последнего типа, ассоциации первого типа будем использовать для создания структур данных, описывающих понятия.

Принципы, положенные в основу моделирования понимания ЕЯ

- ориентация на распознавание семантики с минимальным использованием знаний о синтаксисе языка,
- формирование иерархических структур из понятий, как результат обработки предложений ЕЯ, с горизонтальными (ассоциативными) связями между узлами этих иерархических структур,
- распознавание смысла (интерпретация) предложения на ЕЯ как процесс, состоящий из трех этапов
 - движения снизу вверх по иерархии понятий (распознавание понятий),
 - движение по горизонтальным связям (поиск ассоциаций),
 - движение сверху вниз по иерархии понятий (генерация отклика-выхода системы),
- распознавание слов, словосочетаний и понятий по максимальной похожести с использованием нейроподобных алгоритмов.

Принципы контекстно-ориентированного обучения/программирования роботов с использованием предложенной модели понимания ЕЯ

- Операторы очень просты с 1 или 0 параметрами для задания значений контекстных переменных
- Действие может начаться только тогда, когда все необходимые для него контекстные переменные имеют значения
- Эти особенности похожи на особенности действий человека, получающего указания на естественном языке
- Комбинация алгоритмического и событийного программирования
- Оператор, вызывающий действие, не имеет вообще параметров. Все параметры берутся из контекстных переменных

Изложены в

Таблица контекстных переменных (примитивов) для мобильного робота

Имя контекстной переменной	Возможные значения	Как этот параметр используется при выполнении примитива действия (<i>say</i> или <i>act</i>)
<i>Object</i>	Имя объекта	Может быть использован в действии “ <i>say</i> ”
<i>Direction</i>	Left, Right, Forward, Back	Может вызывать соответствующий поворот в зависимости от значения <i>Internal_State</i>
<i>Person</i>	Имя или иной идентификатор одушевленного лица	Может быть использован в действии “ <i>say</i> ”
<i>Obstacle_Distance</i>	Far, Middle, Close	Может быть использован в действии “ <i>act</i> ” для обхода препятствий
<i>Obstacle_Type</i>	Static, Dynamic	Может быть использован в действии “ <i>act</i> ” для обхода препятствий
<i>Speed</i>	Low, Normal, High	Может быть использован в действии “ <i>act</i> ”
<i>Behavior</i>	Название поведения	Может быть использован в действии “ <i>say</i> ”

Таблица контекстных переменных (примитивов) для мобильного робота (2)

Time	Future, Present, Past	Может быть использован в действии “say”
Place	Имя места	Может быть использован в действии “say”
X, Y	Координаты точки в абсолютной системе координат (в сантиметрах или метрах, в зависимости от типа робота и задач)	Может быть использован в действии “say” или в “act” для выполнения перемещения
Left, Right, Front, Back	Имя объекта или места, распознаваемое роботом в соответствующем направлении от него	Может быть использован в действии “say” или в “act” для выполнения поворота
Angle	Значение угла относительно направления движения робота	Может быть использован в действии “say” или в “act” для выполнения поворота
Distance	Значение расстояния до некоторой точки, объекта или места (в сантиметрах или метрах, в зависимости от типа робота и задач)	Может быть использован в действии “say”

Примеры предложений для обучения мобильного робота

1. Идти – *{#act}*.
2. Объект – *{#object}*.
3. Яблоко, стол – объекты.
4. Идти к объекту – *{#object; #act}*.
5. Стол это мебель.
6. Мебель: стол, стул, книжный шкаф, посудный шкаф.
7. Какой объект прямо – *{#direction= Forward; #say}*.
8. Найти объект – *{#action=Find; #act}*.

Этапы обучения робота

- Обучение соответствию между словами и контекстными переменными
 - Объект – `{#object}`.
- Обучение понятиям предметной области (окружения)
 - Мебель: стол, стул, книжный шкаф, посудный шкаф.
- Обучение поведению
 - Найти объект – `{#action=Find; #act}`.

На каком-то этапе обучения поведению не надо будет использовать встроенные примитивы, а только естественный язык

Примеры предложений-команд, вызывающих действия:

- Иди к столу.
- Какой объект находится прямо?
- Найди яблоко

В случае работа с манипулятором в набор примитивов-действий можно ввести примитивы `#take` и `#put`, обеспечивающие два базовых действия манипулятора – взять и положить.

Этот подход описан в работах:

- Гаврилов А.В. Диалоговая система подготовки программ для роботов. - В сб. "Диалоговые системы в задачах управления", Новосибирск, НЭТИ, 1987.
- Гаврилов А.В. Диалоговая система подготовки программ для роботов. - В сб. Automatyka, v.99, Glivice, 1988, p.173-180.
- Andrey V.Gavrilov. Context and Learning based Approach to Programming of Intelligent Equipment. - The 8th Int. Conf. on Intelligent Systems Design and Applications ISDA'08. Kaohsiung City, Taiwan, November 26-28, 2008. – Pp. 578-582.
- Andrey V.Gavrilov. New Paradigm of Context based Programming-Learning of Intelligent Agent. - Proc. of 1st Workshop on Networked embedded and control system technologies. In conjunction with 6th International Conference on Informatics in Control, Automation and Robotics ICINCO-2009, Milan, Italy, 2-5 July, 2009. – Pp. 94-99.
- Andrey V. Gavrilov. Context based Programming-Learning of Robots. - 6th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2009), Gwangju, South Korea, October 28–31, 2009. – Pp. 806-809.
- Гаврилов А.В. Контекстно-ориентированная гибридная архитектура системы х. управления интеллектуального робота. – 6-я Всероссийская научно-тех. конф. С международным участием «Робототехника и искусственный интеллект», Железногорск, 2014 (в печати)

Литература

- <http://www.aot.ru/>
- <http://sz.ru/parser/>
- <http://www.insycom.ru/html/articles.html>