

УДК 519.876.2

С.С. МагазовМосква, Россия, magazov@mail.ru

Челночный алгоритм обхода 2D области*

Работы по созданию когнитивных роботов и мультиагентных систем интенсивно ведутся в США, Германии, Испании, Японии. Большое внимание уделяется проблеме навигации робота. Это направление исследований получило название Spatial cognition. В работе рассматривается задача обхода роботом прямоугольной области, на которой в произвольном порядке расположены выпуклые фигуры. Эта задача имеет математическое решение. Что позволяет «малоинтеллектуальному» роботу решать эту задачу.

1 Основные определения

Онтологией назовем среду, в которой действует робот. Для нашей задачи онтология – это прямоугольная область, поделенная на клетки (рис 1.). Будем считать, что сторона клетки равна 1. На прямоугольной области в произвольном порядке расположены выпуклые фигуры. Фигуры не пересекаются. Будем рассматривать такие онтологии, когда между любыми двумя фигурами минимальное расстояние строго более 1. Для описания поведения робота будут использоваться две системы координат:

Стационарная – декартова система? как на рис.1. Координаты будем обозначать большими буквами X,Y;

Мобильная – декартова система координат жестко связана с роботом: ось x направлена параллельно фронту (слева направо с точки зрения робота), ось y перпендикулярна фронту по направлению движения.

Если робот находится не у края прямоугольной области, то в этой системе координат он всегда имеет координаты (1,0). Если находится у края? то его координаты (0,0). Мобильные координаты будем обозначать маленькими буквами x,y.

Определение 1.1

1. Точки фигуры, максимально или минимально удаленные от оси Y, назовем *особыми точками*.

2. Будем говорить, что максимальная точка [X,Y] фигуры Ф1 доступна для фигуры Ф2, если фигура Ф2 имеет пересечение со строкой X и участок строки от этого пересечения до Y свободен.

* Работа выполнена при поддержке гранта РФФИ N 03-06 80192

2 Свойства онтологии

Нам потребуются следующие свойства онтологии.

Лемма 2.1

У любой выпуклой фигуры может быть либо две особые точки, либо бесконечное множество особых точек. Если особых точек бесконечное множество, то они лежат на одной или двух отрезках прямых, параллельных оси Y .

3 Робот

Робот занимает *одну клетку* и имеет фронт, спину. Он может двигаться параллельно сторонам области и устанавливать маркированные флажки. Робот движется всегда фронтом вперед. Робот получает информацию о состоянии онтологии через функцию наблюдения и имеет регистровую память. Робот не имеет карты области и не строит ее в процессе движения. В нашей онтологии робот может проходить между любыми фигурами.

Определение 2.2 Функция наблюдения – это функция, которая определяет расстояние от фронта робота до границы фигуры в пределах одной клетки. Точнее, в мобильных координатах функция наблюдения определена на $[0,3]$ и имеет значения в интервале $[0,1]$.

Свободный путь

1. Нигде не определенная на $[0,3]$ функция наблюдения (отсутствие препятствия).

Помеха

1. Функция кусочно-непрерывная, состоящая из одного куска, определенного в интервале $[0,1)$ (помеха слева).
2. Функция кусочно-непрерывная, состоящая из одного куска, определенного в интервале $(2,3]$. Функция кусочно-непрерывная, состоящая из двух кусков, определенных в интервалах $[0,1)$ $(2,3]$ (помеха справа и слева).

Препятствия

1. Функция кусочно-непрерывная, состоящая из одного куска частично определенного в интервале $[0,1]$ (в мобильных координатах).
2. Функция кусочно-непрерывная, состоящая из одного куска частично определенного в интервале $[2,3]$ (в мобильных координатах) (препятствие справа).
3. Функция кусочно-непрерывная, состоящая из одного куска, частично определенного в интервале $[1,2]$ (в мобильных координатах), (препятствие по центру).

Регистровая память

В эту память робот заносит направления своего движения.

Регистровая память (Rm)

Направление по горизонтали	Направление по вертикали
“→” направо; “←” -налево	↓ вниз; ↑ вверх

Воспринимаемая внешняя информация

На каждом шаге роботу доступна информация о состоянии пространства в трех или двух фронтальных клетках. Количество наблюдаемых клеток зависит от положения робота относительно границы области. Информация дается как функция наблюдения. Робот может:

1. Распознавать все типы функций наблюдения.
2. Распознавать особые точки. Если он находится, в обозреваемой области.
3. Определять флаг и читать надписи на нем, то есть переносить их в регистр памяти.

Акции с флажками

Робот имеет пронумерованные пары флажков – **start n** и **finish n**, где **n** номер пары. Робот может:

- 1) устанавливать пару флажков. На каждом флаге робот пишет следующую информацию;
 - а) направление движения робота на момент установления флага, т.е. состояние регистровой памяти;
 - б) на особой точке или не особой точке устанавливать флажки;
 - с) дописывать последовательность номеров. Порядок формирования последовательности будет описан в алгоритме;
- 2) передвигать флаг на одну клетку по границе фигуры, менять запись на флажке;
- 3) уничтожать флаг.

Движения

Простые маневры

1. Двигаться по шагам параллельно любой из сторон прямоугольника. Шаг возможен только в незанятую клетку. Шаг – это продвижение на одну клетку *вверх, вниз, направо, налево*.
2. Поворот по часовой стрелке или против часовой стрелки на девяносто градусов. Поворот производится внутри клетки.

Сложные маневры

1. *Маневр спуска (подъема)* на строку ниже (выше) вдоль препятствия. Если робот на своем пути встретил препятствие, то он делает маневр перехода на строку ниже (выше). Маневр заключается в следующем:

а) возврат назад по строке, по которой робот пришел к препятствию, и так до тех пор, пока переход на смежную строку не станет возможен (самая первая возможность);

б) переход на смежную строку ($\downarrow\uparrow$). Робот делает поворот в сторону фигуры – (фронт к фигуре) и движется, пока не возникнет одно из следующих событий:

- встретит препятствие в виде границы фигуры;
 - встретит особую точку;
- с) останавливается.

2. *Маневр возврат к флажку N.* Робот может вернуться к флажку с номером 1 с пометкой «финиш», но на строку ниже или выше в зависимости от направления движения.

3. *Маневр движения вдоль фигуры.* Робот может двигаться вдоль границы фигуры.

4 Алгоритм обхода

Основная идея алгоритма обхода следующая: робот пытается обойти фигуру, совершая горизонтальные челночные движения от границы до границы. Робот, встретив препятствие, меняет строку либо переходит вверх или вниз в зависимости от состояния регистровой памяти. В процессе такого обхода фигура может быть обойдена не полностью ввиду того, что на пути робота встретилась другая фигура. И эта фигура будет препятствовать горизонтально-челночному движению. Робот «переключит свое внимание» на препятствие, уже пытаясь его обойти, придерживаясь той же стратегии обхода. При помощи установки флагов робот контролирует, была ли обойдена фигура полностью или нет. Также робот использует флаг при возврате к обойденной не полностью фигуре. Перейдем к точной спецификации алгоритма.

Начальное положение робота в – верхней точке оси, как показано на рис.1.

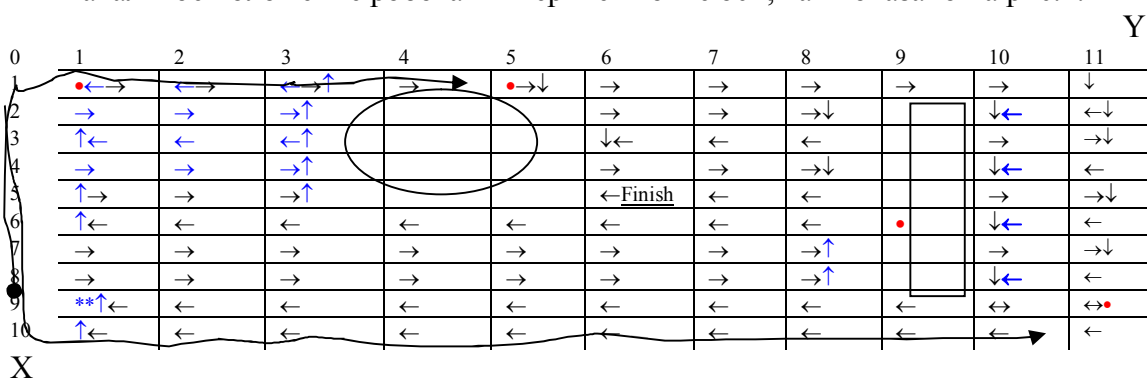


Рисунок 1 – Пример обхода модели онтологии

В алгоритме используется мобильная система координат. Робот должен поставить флажки **start(1)**, **finish(1)**, на которых указано направление движения.

Алгоритм

М: Пока не встретит препятствие, робот горизонтально движется в направлении, указанном в регистровой памяти.

If встретилось препятствие

StateMoveFlag()

If StateMoveFlag() = *‘фигура обойдена’*

then

if (все фигуры обойдены)

Then

Return “Обход завершен”

else

GoBack()

GO TO M

*/*Маневр возврат к поставленному флажку с наибольшим номером. */*

end if

else

case RM= ↓

GoDown()

case RM= ↑

GoUp()

End if

GO TO M

END алгоритм обхода

Procedure StateMoveFlag

*/*Расставляет флаги и изменяет их нумеровку */*

if робот не наблюдает флага на границе фигуры (области)

then

ставится флаг на границе напротив робота;

записывает состояния регистра и тип точки (особая или нет);

return(*‘ Установлен флаг’*)

if else на наблюдаемом участке границы один флага любого типа

флаг передвигается напротив робота;

записывает состояния регистра и тип точки (особая или нет);

if else вершина прямоугольника флага *finish*

флаг передвигается на противоположную сторону;

if else на наблюдаемом участке границы два флага (справа и слева)

/ N1,N2- последовательности номеров */*

case *start {N1} & finish {N2} & N1=N2*

return(*‘фигура обойдена’*)

case *start {N1} & finish {N2} & N1!=N2*

флаги *start {N1} & finish {N2}* убираются;

флаг *finish {N2}* заменяется *finish {N1,N2}*

флаг *start {N1}* заменяется *start {N1,N2}*

Робот переходит по направлению движения вдоль границы фигуры в точку start N1,N2 или finish N1,N2 (в зависимости от направления движения). /*Остаются флаги start N1, N2 и finish N1,N2 направление движения сохраняется */
 записывает состояния регистра и тип точки (особая или нет);
case finish { N1} & finish{N2} then
 флаги finish {N1} и finish {N2} убираются;
 флаг start {N1} заменяется start {N1,N2}
 флаг finish {N2} заменяется finish {N1,N2}
 переходит по направлению движения вдоль границы фигуры в точку start N1,N2 или finish N1,N2 (в зависимости от направления движения).
 /*Остаются флаги start N1, N2 и finish N1,N2 направление движения сохраняется */
 /*Остаются флаги start N1, N2 и finish N1,N2 направление */
 записывает состояния регистра и тип точки (особая или нет);
case один флаг start N1 другой start N2 then
 флаги start N1 и start N2 убираются;
 флаг finish N1 заменяется start {N1,N2}
 флаг finish N2 заменяется finish {N1,N2}
 переходит по направлению движения в точку start {N1,N2} или finish {N1,N2}. /*остаются флаги start {N1,N2} и finish {N1,N2} направление движения сохраняется */
 записывает состояния регистра и тип точки (особая или нет);
return('Флаг передвинут')

END StateMoveFlag

Procedure GoDown

Маневр спуска на строку ниже

if на строке по центру (относительно робота) особая точка

Then

StateMoveFlag();

Разворот на 180°

Двигается до препятствия (граница фигуры или области);

StateMoveFlag();

Двигается в противоположном направлении до препятствия в виде границы, какой-либо фигуры или области (строка не меняется);

Изменить направление вертикального движения в регистровой памяти на противоположное

Разворот на 180°

Else

StateMoveFlag()

Разворот на 180°

End GoDown

Procedure GoUp

Маневр спуска на строку ниже выше

if на строке по центру особая точка

Then

StateMoveFlag();

Разворот на 180°

Двигается до препятствия (граница фигуры или области);

StateMoveFlag();

Двигается в противоположном направлении до препятствия в виде границы, какой-либо фигуры или области (строка не меняется);

Изменить направление вертикального движения в регистровой памяти на противоположное

Разворот на 180°

Else

Разворот на 180°

StateMoveFlag()

End GoUp

Procedure GoBack()

/ Возврат на предыдущую фигуру.*/*

Двигаться вдоль границы фигуры до точки, где был поставлен последний флаг

Разворот на 180°

End GoBack

Основная теорема.

Теорема 4.1 Если клетка [X,Y] свободная, то, двигаясь согласно алгоритму работ на некотором шаге n будет находиться в [X,Y].

Литература

1. Килибарда Г., Кудрявцев В.Б., Ушчюмлич Ш. Независимые системы автоматов в лабиринте // Дискретная математика. – Т. 15. – В. 3. – 2003.
2. Килибарда Г., Кудрявцев В.Б., Ушчюмлич Ш. Коллективы автоматов в лабиринте // Дискретная математика. – Т. 15. – В. 2. – 2003.

С.С. Магазов

Човниковий алгоритм обходу 2D області

Роботи зі створення когнітивних роботів і мультиагентних систем інтенсивно ведуться в США, Німеччині, Іспанії, Японії. Велику увагу приділяють проблемі навігації робота. Цей напрямок досліджень отримав назву Spatial cognition. У роботі розглядається задача обходу роботом прямокутної області, на якій у довільному порядку розташовані опуклі фігури. Ця задача має математичне розв'язання, це дозволяє «малоінтелектуальному» роботу розв'язувати цю задачу.

Статья поступила в редакцию 07.07.2004.