Министерство науки и образования РФ

Федеральное государственное бюджетное учреждение

высшего образования

**«Тверской государственный технический университет»**

(ТвГТУ)

Кафедра программного обеспечения

**Отчет по лабораторной работе №4**

по дисциплине: ***«Интеллектуальные информационные системы»***

Тема: ***«Реализация поиска в пространстве состояний»***

Выполнил:

Студент группы

Б.ПИН.РИС – 18.06.

Кондратьев М.А.

Проверил:

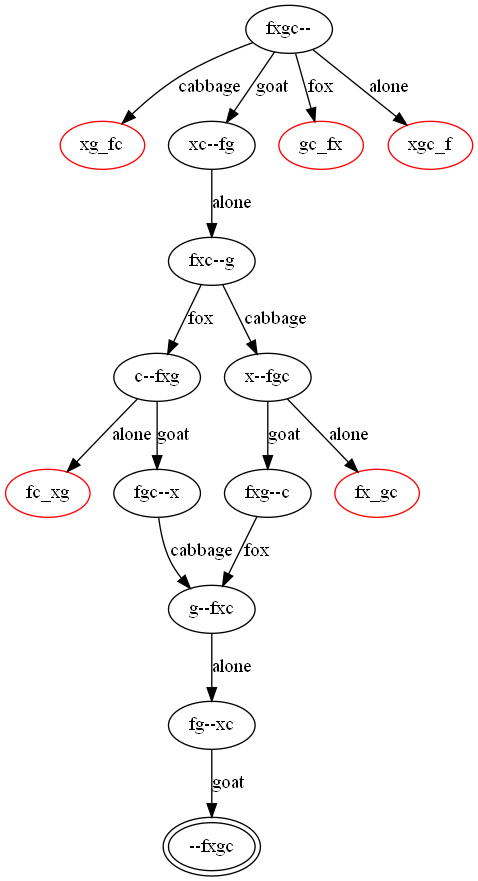
Мальков А.А.

# Цель

Реализация в среде CLIPS задачи про фермера, лису, козу и капусту. Поиска в пространстве состояний и анализ ее решения

# Дерево поиска задачи

Обозначения f - Фермер x - Лиса g - Коза с ­– Капуста



# Структура информационной системы

Структура программы состоит из трёх модулей

* Основного (MAIN)
* Ограничений (CONSTRAINS)
* Вывода (SOLUTION)

Основной модуль

Модуль отвечает за определения основных правил и понятий задачи

Содержит:

* Структуру шаблона состояния (status)

(deftemplate MAIN::status

(slot farmer-location (type SYMBOL) (allowed-symbols shore-1 shore-2))

(slot fox-location (type SYMBOL) (allowed-symbols shore-1 shore-2))

(slot goat-location (type SYMBOL) (allowed-symbols shore-1 shore-2))

(slot cabbage-location (type SYMBOL) (allowed-symbols shore-1 shore-2))

(slot parent (type FACT-ADDRESS SYMBOL) (allowed-symbols no-parent))

(slot search-depth (type INTEGER) (range 1 ?VARIABLE))

(slot last-move (type SYMBOL) (allowed-symbols no-move alone fox goat cabbage))

)

* Определения исходных фактов

(deffacts MAIN::initial-positions

(status

(search-depth 1)

(parent no-parent)

(farmer-location shore-1)

(fox-location shore-1)

(goat-location shore-1)

(cabbage-location shore-1)

(last-move no-move)

)

)

* Определение отношения между берегами

(deffacts MAIN::opposites

(opposite-of shore-1 shore-2)

(opposite-of shore-2 shore-1)

)

* Правила генерации пути

;Правило взять с собой лису

(defrule MAIN::move-with-fox

?node <- (status (search-depth ?num)

(farmer-location ?side)

(fox-location ?side)

)

(opposite-of ?side ?nside)

=>

(duplicate ?node

(search-depth =(+ 1 ?num))

(parent ?node)

(farmer-location ?nside)

(fox-location ?nside)

(last-move fox)

)

)

;Правило взять с собой капусту

(defrule MAIN::move-with-cabbage

?node <- (status (search-depth ?num)

(farmer-location ?side)

(cabbage-location ?side)

)

(opposite-of ?side ?nside)

=>

(duplicate ?node

(search-depth =(+ 1 ?num))

(parent ?node)

(farmer-location ?nside)

(cabbage-location ?nside)

(last-move cabbage)

)

)

;Правило взять с собой козу

(defrule MAIN::move-with-goat

?node <- (status (search-depth ?num)

(farmer-location ?side)

(goat-location ?side)

)

(opposite-of ?side ?nside)

=>

(duplicate ?node

(search-depth =(+ 1 ?num))

(parent ?node)

(farmer-location ?nside)

(goat-location ?nside)

(last-move goat)

)

)

;Переплыть одному

(defrule MAIN::move-alone

?node <- (status (search-depth ?num)

(farmer-location ?side)

)

(opposite-of ?side ?nside)

=>

(duplicate ?node

(search-depth =(+ 1 ?num))

(parent ?node)

(farmer-location ?nside)

(last-move alone)

)

)

Модуль ограничений

Этот модуль отвечает за ограничение ходов сделанных в правилах основного модуля удаляя варианты событий где были нарушены правила

Содержит:

* Правила поедания

;Недопустить лиса ест козу

(defrule CONSTRAINTS::fox-eats-goat

(declare (auto-focus TRUE))

?node <- (status

(farmer-location ?side1)

(fox-location ?side2 & ~?side1)

(goat-location ?side2))

=> (retract ?node)

)

;Недопустить коза ест капусту

(defrule CONSTRAINTS::goat-eats-cabbage

(declare (auto-focus TRUE))

?node <- (status

(farmer-location ?side1)

(goat-location ?side2 & ~?side1)

(cabbage-location ?side2))

=> (retract ?node)

)

* Правило не зацикленности

;Недопустить путь который уже был найден но с более длинным путём

(defrule CONSTRAINTS::circular-path

(declare (auto-focus TRUE))

(status

(search-depth ?sd1)

(farmer-location ?fs)

(fox-location ?xs)

(goat-location ?gs)

(cabbage-location ?cs)

)

?node <- (status

(search-depth ?sd2 & :(< ?sd1 ?sd2))

(farmer-location ?fs)

(fox-location ?xs)

(goat-location ?gs)

(cabbage-location ?cs))

=> (retract ?node)

)

Модуль вывода

Модуль ответственен за вывод результат после нахождения ответа

; Найти правильный ответ

(defrule SOLUTION::goal-test

(declare (auto-focus TRUE))

?node <- (status

(parent ?parent)

(farmer-location shore-2)

(fox-location shore-2)

(goat-location shore-2)

(cabbage-location shore-2)

(last-move ?move))

=>

(retract ?node)

(assert (moves (id ?parent) (moves-list ?move)))

)

; Построить ответ

(defrule SOLUTION::build-solution

?node <- (status

(parent ?parent)

(last-move ?move)

)

?mv <- (moves (id ?node) (moves-list $?rest))

=>

(modify ?mv (id ?parent) (moves-list ?move ?rest))

)

; Вывести ответ

(defrule SOLUTION::print-solution

?mv <- (moves (id no-parent) (moves-list no-move $?m))

=>

(retract ?mv)

(printout t "Solution found: " t)

(bind ?length (length ?m))

(bind ?i 1)

(bind ?shore shore-2)

(while (<= ?i ?length)

(bind ?thing (nth ?i ?m))

(if (eq ?thing alone)

then (printout t " Farmer moves alone to " ?shore "." t)

else (printout t " Farmer moves with " ?thing " to " ?shore "." t)

)

(if (eq ?shore shore-1)

then (bind ?shore shore-2)

else (bind ?shore shore-1)

)

(bind ?i (+ 1 ?i))

)

)

# Анализ процесса решения

Решение можно рассмотреть как действия производимые при нахождении нового варианта событий:



Таким образом система находит все (два) решения

Solution found:

Farmer moves with goat to shore-2.

Farmer moves alone to shore-1.

Farmer moves with fox to shore-2.

Farmer moves with goat to shore-1.

Farmer moves with cabbage to shore-2.

Farmer moves alone to shore-1.

Farmer moves with goat to shore-2.

Solution found:

Farmer moves with goat to shore-2.

Farmer moves alone to shore-1.

Farmer moves with cabbage to shore-2.

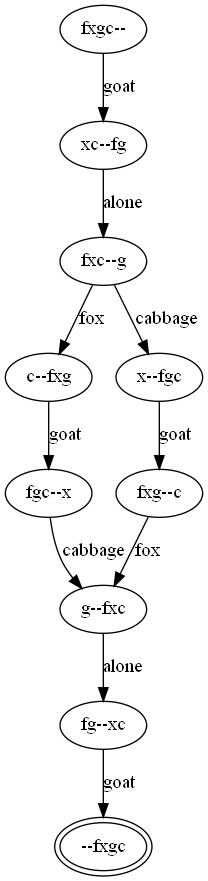
Farmer moves with goat to shore-1.

Farmer moves with fox to shore-2.

Farmer moves alone to shore-1.

Farmer moves with goat to shore-2.

Граф двух найденных решений:



# Заключение

Была реализована задача поиска в пространстве состояний. Проанализирован процесс работы программы. Параллельно изучены азы Модулей в CLIPS.