Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

**Отчет по лабораторной работе №3**

**по курсу «ЛОИС»**

**на тему: «Решение логических задач на языке Prolog»**

Выполнил студент группы 721701: Тесловский А.П.

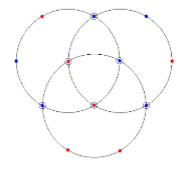
Проверил: Бобков А.В

Минск

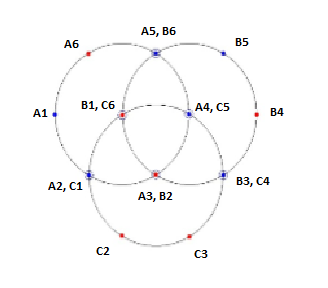
2020

Цель: В соответствии с вариантом необходимо реализовать программу на языке Prolog, решающую поставленную задачу.

Вариант: 11) Задача с тремя окружностями. Вращением окружностей добиться того, чтобы все синее точки оказались в синих кружочках. Синие кружочки остаются неподвижными. Окружности за один шаг поворачиваются на одну шестую круга, т.е. сменяются точки, центральная точка круга не смещается при повороте. Начальное состояние представлено на рисунке:

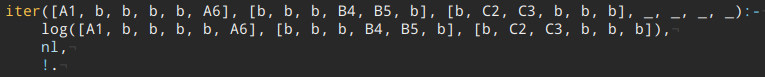


Для решения поставленной задачи было необходимо описать структуру окружностей и точек, для этого были использованы три списка атомов, каждый из которых представляет собой одну окружность с точками, где b - это синяя точка, а r - красная точка. Например для начального состояния из условия задачи можно использовать следующие списки: [b, b, r, b, b, r], [r, r, b, r, b, b], [b, r, r, b, b, r]. Для наглядности каждой точке поставим в соответствие переменную: [A1, A2, A3, A4, A5, A6], [B1, B2, B3, B4, B5, B6], [C1, C2, C3, C4, C5, C6]. Обозначим позиции точек на рисунке:

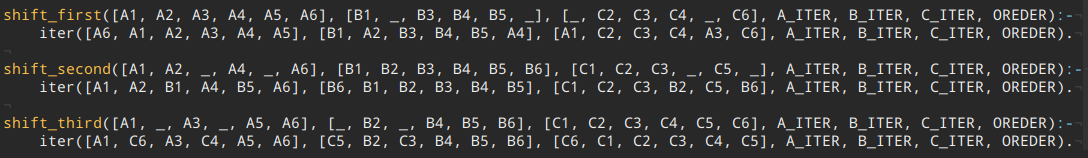


Важно чтобы нумерация точек каждой окружности начиналась с одинаковой позиции. Например в данном случае все переменные с индексом 1 - крайние левые точки соответствующих окружностей.

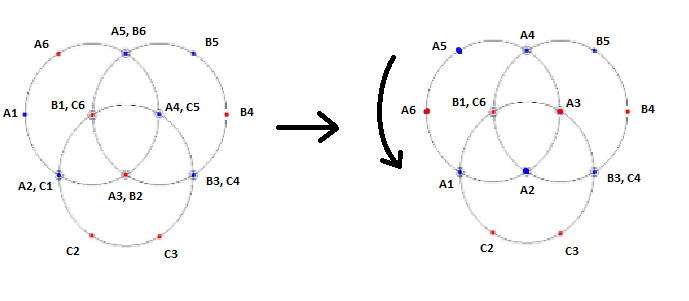
Далее было описано конечное состояние, когда все синие точки находятся в синих кружочках, это условие завершения программы:



Также было необходимо описать правила поворота каждой окружности. При этом было необходимо учесть что окружности имеют общие точки и поворот одной окружности влияет на другие окружности. Повороты были реализованы с помощью следующих правил:

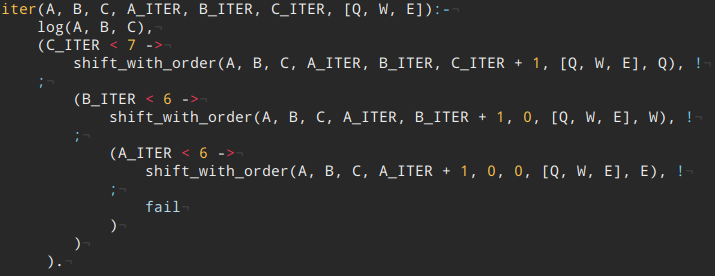


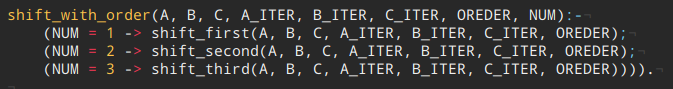
На примере shift\_first рассмотрим поворот первой окружности (А). Правило смещает точки первой окружности против часовой стрелки, что в результате даёт следующий список: [A6, A1, A2, A3, A4, A5]. В остальных списках заменяются те точки, которые пересекаются с первой окружностью. Это точки B2, B6, C1, C5, на их место станут точки списка А, в результате получаем следующие списки для окружностей B и С: [B1, A2, B3, B4, B5, A4], [A1, C2, C3, C4, A3, C6]. Для наглядности поворот отображён на картинке:



Правила поворотов двух других окружностей реализованы аналогично.

Далее было реализовано правило обхода состояний которое имеет следующий вид:

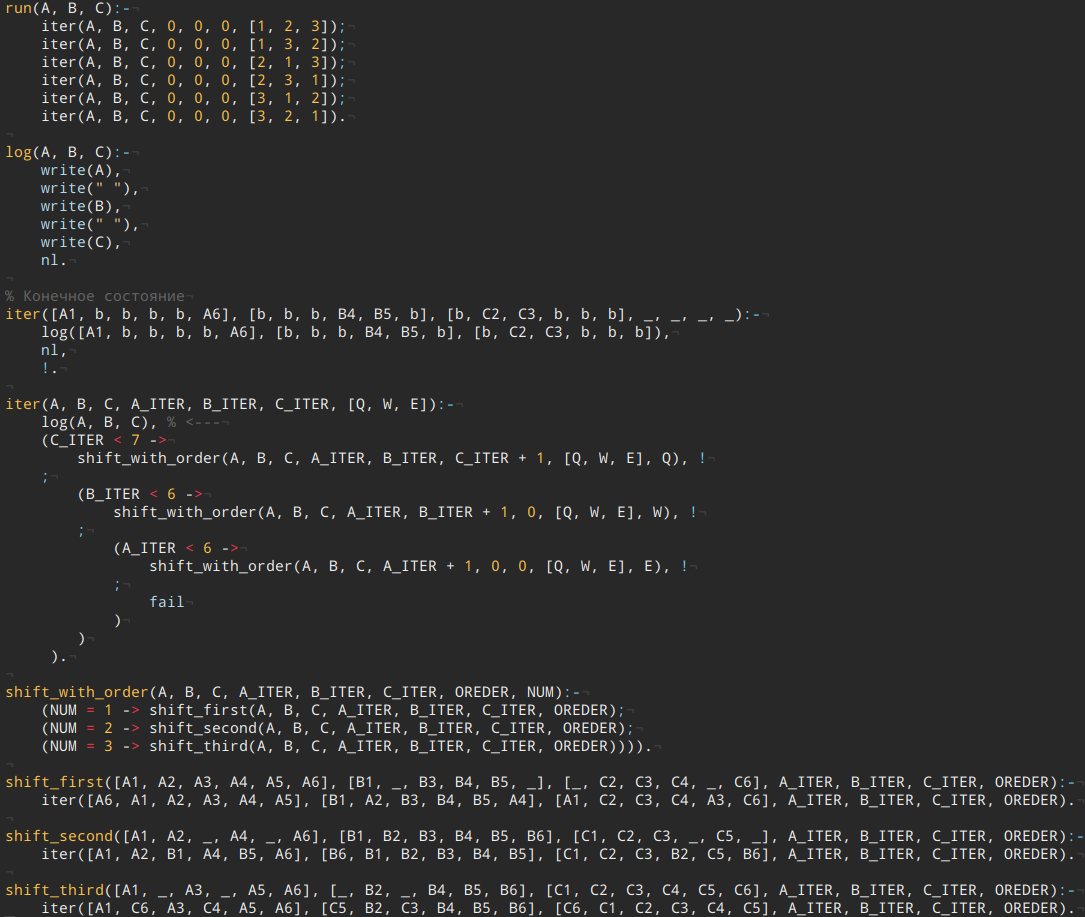


Здесь также используются переменные A\_ITER, B\_ITER и C\_ITER, которые нужны для учёта итераций вращения каждой окружности. После полного оборота окружность возвращается в исходное состояние, затем сдвигается следующая окружность и повторяется полный оборот предыдущей. После полного оборота второй окружности сдвигается последняя и повторяются обороты двух предыдущих. Для обхода всех состояний окружностей необходимо также учесть порядок вращения, для этого используются переменные Q, W, E и правило shift\_with\_order, определяющее какую окружность сдвигать на данной итерации. Правило имеет следующий вид:

В качестве входной точки программы было определено правило run, запускающее обход окружностей с разным порядком вращения. Примеры запросов для запуска программы:

1. run([b, b, r, b, b, r], [r, r, b, r, b, b], [b, r, r, b, b, r]). - начальное состояние из условия задачи
2. run([b, r, b, r, r, r], [r, b, b, r, b, r], [r, b, b, b, r, r]).
3. run([b, b, b, b, b, b], [r, b, r, r, r, b], [b, r, r, r, b, r]). - все точки первой окружности синие, остальные красные.
4. run([b, b, b, b, b, b], [b, b, b, b, b, b], [b, b, b, b, b, b]). - все точки синие.
5. run([r, r, r, r, r, r], [r, r, r, r, r, r], [r, r, r, r, r, r]). - все точки красные, достигнуть конечного состояния невозможно и программа вернёт false.
6. run([r, b, r, b, b, r], [r, r, b, r, b, b], [b, r, r, b, b, r]). - начальное состояние из условия задачи, но точка А1 красная, тогда всего синих точек 5, достигнуть конечного состояния невозможно и программа вернёт false.

Полный код программы имеет вид:



Пример вывода программы (часть шагов сокращена):

[b,b,r,b,b,r] [r,r,b,r,b,b] [b,r,r,b,b,r]

[r,b,b,r,b,b] [r,b,b,r,b,b] [b,r,r,b,r,r]

[b,r,b,b,r,b] [r,b,b,r,b,r] [r,r,r,b,b,r]

[b,b,r,b,b,r] [r,r,b,r,b,b] [b,r,r,b,b,r]

[r,b,b,r,b,b] [r,b,b,r,b,b] [b,r,r,b,r,r]

[b,r,b,b,r,b] [r,b,b,r,b,r] [r,r,r,b,b,r]

[b,b,r,b,b,r] [r,r,b,r,b,b] [b,r,r,b,b,r]

[r,b,b,r,b,b] [r,b,b,r,b,b] [b,r,r,b,r,r]

[r,b,r,r,b,b] [b,r,b,b,r,b] [b,r,r,b,r,b]

[b,r,b,r,r,b] [b,b,b,b,r,r] [r,r,r,b,r,b]

...

[b,b,r,r,b,r] [b,r,r,b,b,b] [b,r,r,r,r,b]

[b,b,b,r,b,r] [b,b,r,r,b,b] [b,r,r,r,r,b]

[b,b,b,r,b,r] [b,b,b,r,r,b] [b,r,r,b,r,b]

[b,b,b,r,r,r] [b,b,b,b,r,r] [b,r,r,b,r,b]

[r,b,b,b,r,r] [b,b,b,b,r,r] [b,r,r,b,b,b]

[r,b,b,b,r,r] [r,b,b,b,b,r] [b,r,r,b,b,r]

[r,b,r,b,b,r] [r,r,b,b,b,b] [b,r,r,b,b,r]

[r,b,r,b,b,r] [b,r,r,b,b,b] [b,r,r,r,b,b]

[r,b,b,b,b,r] [b,b,r,r,b,b] [b,r,r,r,b,b]

[r,b,b,b,b,r] [b,b,b,r,r,b] [b,r,r,b,b,b]

При начальном состоянии из условия задачи конечное состояние достигается за 870 шагов.

**Вывод:** В рамках лабораторной работы была реализована программа решения логической задачи на языке Prolog. Также был описан алгоритм решения задачи.