Изображение выглядит как текст

Автоматически созданное описание

Содержание

[Введение 3](#_Toc56967654)

[Постановка задачи 4](#_Toc56967655)

[Ограничения 6](#_Toc56967656)

[Физическая интерпретация задачи 9](#_Toc56967657)

[Заключение 11](#_Toc56967658)

[Список литературы 12](#_Toc56967659)

# Введение

В ходе выполнения курсовой работы мной была решена головоломка «Задача Эйнштейна». Данная «задача» является примером широкого класса задач программирования в ограничениях, в них объекты должны удовлетворять некоторому набору ограничений. Для решения такого класса задач можно использовать BDD. BDD – ациклический орграф, в котором отсутствуют повторения в структуре, с одной корневой вершиной, двумя листьями, помеченными 0 и 1, и промежуточными вершинами. Для представления BDD и выполнения операций над ними существует много различных библиотек. Мы рассмотрим свободно распространяемую библиотеку BuDDy для языка C++.

# Постановка задачи

Вариант 1:

Имеется *N = 9* объектов, расположенных следующим образом:

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |

«Соседские» отношения между объектами определены относительно центрального объекта следующим образом:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | 0 | \* |
|  | \* |  |

Понятие «объекты, расположенные рядом» — это соседи (отмечены \* для объекта с пометкой 0), а понятия «объекты, расположенные слева», «объекты, расположенные справа» — это соседи, расположенные слева, справа соответственно, если таковые есть.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  | 0 |

|  |  |  |
| --- | --- | --- |
| 0 | \* |  |
| \* |  |  |
|  |  |  |

Для всех остальных объектов отношения строятся сдвигом. Например, для объекта в правом нижнем углу этих соседский отношений соседями не будут:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
| 0 | \* |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  | 0 | \* |

Для решения задачи необходимо:

* Выбрать *M = 4* свойств, принимающих *N* различных значений.
* Задать ограничений типа 1, ограничений типа 2.
* Для ограничений типа 3 и типа 4 использовать «соседские» отношения. Придумать, как ограничения, подобные типу 3, 4, выражаются в соседских отношениях, задать ограничений подобных отношениям типа 3 и ограничений подобных отношениям типа 4.
* Описать дополнительный тип ограничения : «сумма свойств объектов-соседей не должна быть больше K», где K – некоторое число от 0 до N\*M. K выбирается самостоятельно.
* Найти все возможные решения.
* Придумать физическую интерпретацию.
* Если задача имеет не одно решение, следует добавить и/или изменить некоторые ограничения так, чтобы задача имела только одно единственное решение.
* Если задача не имеет решений, следует удалить и/или изменить некоторые ограничения так, чтобы задача имела только одно единственное решение. Ограничение типа n7 удалять нельзя.
* Для выполнения задания необходимо написать программу на языке C++ с использованием библиотеки BuDDy.

# Ограничения

1. Ограничение, устанавливающее свойству k1 конкретного объекта i1 значение j1

F := F & p(k1, i1, j1).

1. Ограничение, устанавливающее соответствие между двумя свойствами какого-либо объекта:  
   F := F & ( p(k1, i, j1)⬄p(k2, i, j2) ) для всех i от 0 до N-1.
2. Ограничение расположения объектов – расположение «слева» («справа»). Это ограничения типа объект, у которого свойство k1 имеет значение j1, расположен слева от объекта, у которого свойство k2 имеет значение j2

F := F & ( p(k1, i, j1)⬄p(k2, i + 1, j2) ) для всех i от 0 до N – 2. Очевидно, нужно добавить ограничение: объект, у которого свойство k2 имеет значение j2, не будет первым, а объект, у которого свойство k1 имеет значение j1, не будет последним.

1. Ограничение расположения объектов – расположение «рядом» – дизъюнкция ограничений «слева» и «справа».  
   F := F & [( p(k1, i, j1)⬄p(k2, i + 1, j2) ) ∨ ( p(k2, i, j2)⬄p(k1, i + 1, j1))]

*По умолчанию существуют еще следующие ограничения:*

1. У двух различных объектов значения любого параметра (свойства) не совпадают.

F := F & (p(k, i1, j) ⟹ !p(k, i2, j)) при i1≠ i2.

1. Параметры принимают значения только из заданных множеств (значение свойств должно быть меньше N).
2. Сумма свойств объектов-соседей не должна быть больше K, где K – некоторое число от 0 до N\*M.

Решение задачи

Сначала зададим необходимые ограничения:

N1 = 5, N2 = 6, N3 = 3, N4 =1;

Тип 1:

1. Свойство 0 у объекта 0 равно 0
2. Свойство 0 у объекта 1 равно 1.
3. Свойство 3 у объекта 2 равно 1.
4. Свойство 2 у объекта 6 равно 7.
5. Свойство 2 у объекта 4 равно 5.

Тип 2:

1. Если у объекта свойство 2 имеет значение 6, то у него свойство 1 имеет значение 5 и обратно.
2. Если у объекта свойство 0 имеет значение 3, то у него свойство 2 имеет значение 6 и обратно.
3. Если у объекта свойство 1 имеет значение 8, то у него свойство 2 имеет значение 4 и обратно.
4. Если у объекта свойство 2 имеет значение 7, то у него свойство 0 имеет значение 7 и обратно.
5. Если у объекта свойство 2 имеет значение 8, то у него свойство 0 имеет значение 2 и обратно.
6. Если у объекта свойство 2 имеет значение 1, то у него свойство 3 имеет значение 6 и обратно.

Тип 3:

1. Если у объекта свойство 0 имеет значение 5, то сверху от него стоит объект, у которого свойство 2 имеет значение 1. И наоборот.
2. Если у объекта свойство 0 имеет значение 0, то сверху от него стоит объект, у которого свойство 3 имеет значение 2. И наоборот.
3. Если у объекта свойство 1 имеет значение 4, то над ним стоит объект, у которого свойство 1 имеет значение 8. И наоборот.

Тип 4:

1. Объект, у которого свойство 1 имеет значение 5, стоит рядом с объектом, у которого свойство 3 имеет значение 7.

По умолчанию существуют еще следующие ограничения:

Тип 5:

У двух различных объектов значения любого параметра (свойства) не совпадают.

Тип 6:

Параметры принимают значения только из заданных множеств

Тип 7:

Сумма значений свойств объектов-соседей должна быть меньше, чем .

Для того, чтобы задача имела только 1 решение, потребовалось ввести дополнительные ограничения:

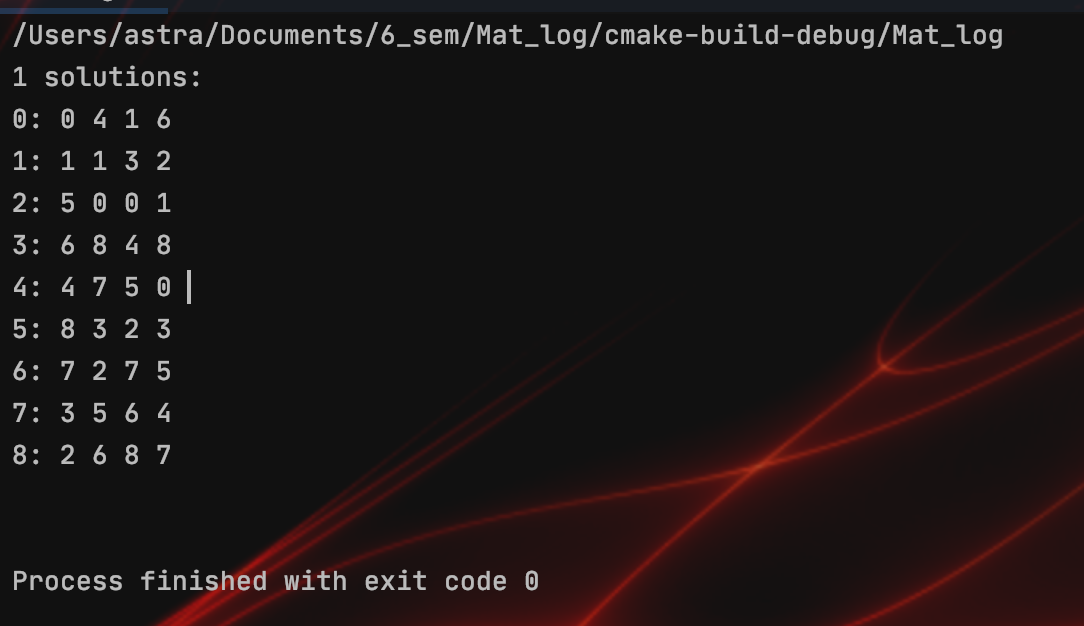
Тип 1:

* Свойство 1 у объекта 8 равно 6.
* Свойство 3 у объекта 5 равно 3.
* Свойство 0 у объекта 5 равно 8.
* Свойство 1 у объекта 0 равно 4.
* Свойство 0 у объекта 3 равно 6.
* Свойство 1 у объекта 6 равно 2.
* Свойство 2 у объекта 2 равно 0.
* Свойство 2 у объекта 5 равно 2.
* Свойство 3 у объекта 6 равно 5.
* Свойство 1 у объекта 1 равно 1.
* Свойство 3 у объекта 7 равно 4.
* Свойство 1 у объекта 2 равно 0.

Тип 2:

* + Если у объекта свойство 0 имеет значение 6, то у него свойство 3 имеет значение 8 и обратно.
  + Если у объекта свойство 0 имеет значение 4, то у него свойство 1 имеет значение 7 и обратно.

С заданными ограничениями было получено единственное решение:



# Физическая интерпретация задачи

В отель пришли 9 человек. Для них выделили места с номерами от 0 до 8, которые расположены следующим образом:

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |

Соседями дипломата в центре (помечен 4) будут дипломаты, сидящие выше, ниже, слева и справа (1, 7, 3 и 5). На каждом месте может сидеть только один дипломат. Все места имеют разные цвета. Имена и национальность дипломата не совпадают. Каждому дипломату подали его любимый напиток. Напитки также не совпадают.

Известно, что:

* На месте 0 сидит человек, у которого любимая игра CS: GO
* На месте 1 сидит человек, у которого любимая игра HS
* На втором сидит человек, который любит шампанское
* На шестом месте сидит человек, который любит суп
* На четвертом месте сидит человек, который любит суши
* Если человек любит хлеб, то у него национальность – Англичанин
* Если человек любит играть в Call of duty, то он любит хлеб
* Если человек Канадец, то у него любимая еда – Орехи
* Если человек любит суп, то он любит играть в Warface
* Если человек любит Бургер, то он играет в Dota
* Если человек любит Мясо, то он любит кока-колу
* Если человек играет в Mortal combat, то сверху от него человек, который любит фрукты.
* Если человек играет в CS:GO, то сверху от него человек, который любит водку.
* Если человек – Итальянец, то снизу от него канадец.

Дополнительно известно, что:

* На восьмом месте сидит Бельгиец
* На пятом месте сидит человек, который любит рыбу
* На пятом месте человек играет в tanki
* На нулевом месте сидит Итальянец
* На третьем месте человек играет в покер
* На шестом месте Американец
* На втором месте любит мясо
* На пятом месте любит овощи
* На шестом месте любит суп
* На первом месте Грузин
* На седьмом месте любит виски
* На втором месте Мадагаскарец
* Если человек играет в Poker, то он любит кофе.
* Если человек играет в ORI, то он Японец.

Закодируем домены свойств следующим образом:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Место | Свойство 0  Любимая Игра | Свойство 1  Национальность | Свойство 2 Любимая еда | Свойство 3 Любимый напиток |
| 0 | CS:GO | Мадагаскарец | Мясо | Шампанское |
| 1 | HS | Грузин | Фрукты | Вода |
| 2 | Dota | Американец | Овощи | Водка |
| 3 | Call of duty | Русский | Рыба | Ром |
| 4 | ORI | Итальянец | Орехи | Виски |
| 5 | Mortal combat | Англичанин | Суши | Суп |
| 6 | Poker | Бельгиец | Хлеб | Кока-кола |
| 7 | WarFace | Японец | Суп | Пиво |
| 8 | Tanki | Канадец | Бургер | Кофе |

Интерпретация решения:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Место | Компьютерная Игра | Национальность | Любимая еда | Любимый напиток |
| 0 | CS:GO | Итальянец | Фрукты | Кока-кола |
| 1 | HS | Грузин | Рыба | Водка |
| 2 | Mortal combat | Мадагаскарец | Мясо | Вода |
| 3 | Poker | Канадец | Орехи | Кофе |
| 4 | ORI | Японец | Суши | Шампанское |
| 5 | Tanki | Русский | Овощи | Ром |
| 6 | WarFace | Американец | Суп | Суп |
| 7 | Call of duty | Англичанин | Хлеб | Виски |
| 8 | Dota | Бельгиец | Бургер | Пиво |

# 

# Заключение

Благодаря известной библиотеке BuDDy, я смог решить задачу в парадигме программирования в ограничениях. При заданных значениях количества объектов и свойств было получено единственное решение. Решение задачи было сведено к операциям над булевыми значениями с помощью введения бинарного кодирования множеств значений свойств. Также на конкретном примере “задачи Эйнштейна” были изучены методы библиотеки BDD BuDDy, которая позволила быстро и эффективно решить задачу для конечного количества переменных.

Данная курсовая работа была выполнена на языке С++ в среде разработки – CLion.

Код проекта представлен на сервисе GitHub по ссылке: **https://github.com/VortexAstra/Mathematical-Logic**

# Список литературы

1. Ю.Г. Карпов. Теория автоматов. Семнадцать практических занятий. – СПб.: Питер, 2003. – 208с.: ил.
2. 2. Ю.Г. Карпов. Задачи по курсу «Математическая логика и теория алгоритмов». – М.: Издательство СПбГПУ, 2014. – 105с.
3. Программирование в ограничениях [Электронный ресурс]: Википедия. Свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/Программирование\_в\_ограничениях
4. BuDDy: A BDD package [Электронный ресурс]. URL: http://buddy.sourceforge.net/manual/main.html
5. BuDDy Binary Decision Diagram (BDD) library [Электронный ресурс]. URL: https://github.com/jgcoded/BuDDy