МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

# Кафедра Інформатики

### ЗВІТ

### з лабораторної роботи № 2

з дисципліни: «[Машинне навчання](https://dl.nure.ua/course/view.php?id=4882)»

Виконав: Перевірила:

ст. гр. ІТІНФ-20-1 Шафроненко А.Ю.

Самченко С. О.

Харків 2022

**Мета заняття:**

Закріпити основні поняття навчання концептам. Застосувати алгоритми Find-S і Candidate-Elimination для задач навчання концептам.

**Завдання:**

1. Обчислити кількість можливих прикладів та гіпотез.
2. Застосувати Find-S алгоритм.
3. Застосувати Candidate-Elimination алгоритм та показати послідовність S та G множин.
4. Реалізувати процедуру відновлення простору версій з S та G множин.

**Хід Роботи:**

Изображение выглядит как текст, табло, черный, снимок экрана

Автоматически созданное описание

Рис.1 - Набір навчальних даних для алгоритмів

Изображение выглядит как текст

Автоматически созданное описание

Рис. 2 - Результати роботи алгоритмів

Код програми:

using System;

using System.IO;

using System.Collections.Generic;

namespace MNtAD1

{

    class Program

    {

        public class Purchase

        {

            private string citations;

            private string size;

            private string inLibrary;

            private string price;

            private string editions;

            private string buy;

            public string Citations

            {

                get

                {

                    return citations;

                }

                set

                {

                    citations = value;

                }

            }

            public string Size

            {

                get

                {

                    return size;

                }

                set

                {

                    size = value;

                }

            }

            public string InLibrary

            {

                get

                {

                    return inLibrary;

                }

                set

                {

                    inLibrary = value;

                }

            }

            public string Price

            {

                get

                {

                    return price;

                }

                set

                {

                    price = value;

                }

            }

            public string Editions

            {

                get

                {

                    return editions;

                }

                set

                {

                    editions = value;

                }

            }

            public string Buy

            {

                get

                {

                    return buy;

                }

                set

                {

                    buy = value;

                }

            }

            public Purchase() { }

            public Purchase(string citations, string size, string inLibrary, string price, string editions,

                string buy)

            {

                this.citations = citations;

                this.size = size;

                this.inLibrary = inLibrary;

                this.price = price;

                this.editions = editions;

                this.buy = buy;

            }

            public Purchase(Purchase p)

            {

                this.citations = p.citations;

                this.size = p.size;

                this.inLibrary = p.inLibrary;

                this.price = p.price;

                this.editions = p.editions;

                this.buy = p.buy;

            }

            public string PrintPurchase()

            {

                string result = $"Citations = {citations} size = {size} inLibrary = {inLibrary} price = {price} " +

                    $"editions = {editions}";

                return result;

            }

        }

        public class Algorithms

        {

            public Purchase FindSAlgorithm(List<Purchase> purchases)

            {

                Purchase hypothes = new Purchase();

                int n = 0;

                for (int i = 0; i < purchases.Count; i++)

                {

                    if (purchases[i].Buy.Equals("yes"))

                    {

                        hypothes = new Purchase(purchases[i]);

                        n++;

                        break;

                    }

                }

                for (int i = n; i < purchases.Count; i++)

                {

                    if (purchases[i].Buy.Equals("yes"))

                    {

                       if (!purchases[i].Citations.Equals(hypothes.Citations))

                        {

                            hypothes.Citations = "?";

                        }

                        if (!purchases[i].Size.Equals(hypothes.Size))

                        {

                            hypothes.Size = "?";

                        }

                        if (!purchases[i].InLibrary.Equals(hypothes.InLibrary))

                        {

                            hypothes.InLibrary = "?";

                        }

                        if (!purchases[i].Price.Equals(hypothes.Price))

                        {

                            hypothes.Price = "?";

                        }

                        if (!purchases[i].Editions.Equals(hypothes.Editions))

                        {

                            hypothes.Editions = "?";

                        }

                    }

                }

                return hypothes;

            }

            public List<Purchase> CandidateEliminationAlgorithm(List<Purchase> purchases)

            {

                Purchase specificHypothes = new Purchase("?", "?", "?", "?", "?", null);

                List<Purchase> generalHypothesis = new List<Purchase>();

                for (int i = 0; i < 5; i++)

                {

                    generalHypothesis.Add(new Purchase("?", "?", "?", "?", "?", null));

                }

                for (int i = 0; i < purchases.Count; i++)

                {

                    if (purchases[i].Buy.Equals("yes"))

                    {

                        specificHypothes = new Purchase(purchases[i]);

                        break;

                    }

                }

                for (int i = 0; i < purchases.Count; i++)

                {

                    if (purchases[i].Buy.Equals("yes"))

                    {

                        if (!purchases[i].Citations.Equals(specificHypothes.Citations))

                        {

                            specificHypothes.Citations = "?";

                        }

                        if (!purchases[i].Size.Equals(specificHypothes.Size))

                        {

                            specificHypothes.Size = "?";

                        }

                        if (!purchases[i].InLibrary.Equals(specificHypothes.InLibrary))

                        {

                            specificHypothes.InLibrary = "?";

                        }

                        if (!purchases[i].Price.Equals(specificHypothes.Price))

                        {

                            specificHypothes.Price = "?";

                        }

                        if (!purchases[i].Editions.Equals(specificHypothes.Editions))

                        {

                            specificHypothes.Editions = "?";

                        }

                    }

                    else

                    {

                        if (!purchases[i].Citations.Equals(specificHypothes.Citations))

                        {

                            generalHypothesis[0].Citations = specificHypothes.Citations;

                        }

                        if (!purchases[i].Size.Equals(specificHypothes.Size))

                        {

                            generalHypothesis[1].Size = specificHypothes.Size;

                        }

                        if (!purchases[i].InLibrary.Equals(specificHypothes.InLibrary))

                        {

                            generalHypothesis[2].InLibrary = specificHypothes.InLibrary;

                        }

                        if (!purchases[i].Price.Equals(specificHypothes.Price))

                        {

                            generalHypothesis[3].Price = specificHypothes.Price;

                        }

                        if (!purchases[i].Editions.Equals(specificHypothes.Editions))

                        {

                            generalHypothesis[4].Editions = specificHypothes.Editions;

                        }

                    }

                }

                generalHypothesis.Add(specificHypothes);

                return generalHypothesis;

            }

        }

        static void Main(string[] args)

        {

            string path = @"D:\work\data.txt";

            using (StreamReader sr = new StreamReader(path, System.Text.Encoding.Default))

            {

                List<Purchase> purchases = new List<Purchase>();

                string line;

                while ((line = sr.ReadLine()) != null)

                {

                    string[] s = line.Split('\t');

                    Purchase purchase = new Purchase(s[0], s[1], s[2], s[3], s[4], s[5]);

                    purchases.Add(purchase);

                }

                Algorithms algorithms = new Algorithms();

                Console.WriteLine("Data set:");

                for (int i = 0; i < purchases.Count; ++i)

                {

                    Console.WriteLine(purchases[i].PrintPurchase() + " buy = " + purchases[i].Buy);

                }

                Console.WriteLine("Result Find-S Algorithm:");

                Console.WriteLine(algorithms.FindSAlgorithm(purchases).PrintPurchase());

                List<Purchase> generalSolution = algorithms.CandidateEliminationAlgorithm(purchases);

                Console.WriteLine("Result Candidate Elimination Algorithm:");

                Console.WriteLine("general hypothesis:");

                for (int i =0; i < generalSolution.Count - 1; ++i)

                {

                    Console.WriteLine(generalSolution[i].PrintPurchase());

                }

                Console.WriteLine("specific hypothesis:");

                Console.WriteLine(generalSolution[generalSolution.Count - 1].PrintPurchase());

            }

        }

    }

}

**Висновок:** У ході лабораторної роботи я ознайомився з роботою алгоритмів Find-S та Candidate-Elimination, також опанував навчання концептам та знаходження гіпотез які розглядаються як кон`юнктивні концепти.