**BÀI TẬP NHÓM 8:**

**Tìm hiểu phương pháp**

**phân tích cú pháp Earley**

**(Có mô phỏng các bước**

**tính toán và dẫn xuất)**

Danh sách nhóm 8 :

1. Nguyễn Thanh Phong
2. Hoàng Minh Hưng
3. Nguyễn Hồng Khánh

# MỤC LỤC

[MỤC LỤC 1](#_Toc367896370)

[**1.** **Mở đầu :** 2](#_Toc367896371)

[**2.** **Giải thuật Earley :** 3](#_Toc367896372)

[*a.* *Giải thuật :* 3](#_Toc367896373)

[*b.* *Thuật toán :* 3](#_Toc367896374)

[*c.* *Ví dụ minh họa :* 3](#_Toc367896375)

[**3.** **Chương trình mô phỏng :** 5](#_Toc367896376)

1. **Mở đầu :**

Giải thuật Earley là một trong những giải thuật được sử dụng trong việc phân tích cú pháp trong xử lý ngôn ngữ tự nhiên. Nó là một giải thuật tổng quát, có thể phân tích bất kỳ văn phạm phi ngữ cảnh nào. Giải thuật này sử dụng phương pháp phân tích kiểu trên xuống (top – down), bắt đầu với một ký hiệu kkhông kết thúc đại diện cho câu và sử dụng các luật khai triển cho đến khi thu được câu vào. Hạn chế của cách tiếp cận này là không chú trọng nhiều đến các từ đầu vào. Vì vậy trong quá trình phân tích, giải thuật Earley phát sinh ra nhiều luật dư thừa.

1. **Giải thuật Earley :**
   1. *Giải thuật :*

*\* Input :* Văn phạm G = (N, T, S, P) trong đó :

* N : tập ký hiệu không kết thúc.
* T : tập ký hiệu kết thúc.
* S : ký hiệu không kết thúc bắt đầu.
* P : tập luật cú pháp.

Xâu đâu vào w = a1a2...an W\*

*\* Output :* Phân tích đối với w ( là một luật có chấm khi ) hoặc “sai”

\* Giải thuật Earley được biểu diễn thông qua việc xây dưng bảng chứa tậo các luật có chấm. Người ta xây dựng bảng Earley với các cột Ii (i = 0,1,...,n), cột I0 nhận giá trị khởi tạo, n là độ dài của chuỗi từ loại nhập. Mỗi ô sẽ có các giá trị: ***giá trị gốc*** để biết luật đó phát sinh từ cột nào, và ***luật có chấm***.

Ví dụ :

|  |  |
| --- | --- |
| Giá trị gốc | Luật có chấm |
| 0 | A 🡪 •B C |
| 1 | C 🡪 D• E |

* 1. *Thuật toán :*

Giải thuật bao gồm 3 bước (tại cột Ii) :

1. Predict (Dự đoán) :

Đối với các luật có ký tự không kết thúc ở bên phải dấu chấm, ta thêm các luật mới mà ký tự không kết thúc đó là vế trái của các luật với giá trị gốc là i. Điều này có nghĩa là, với mỗi trong Ii ta thêm vào Ii nếu.

1. Scanner (Duyệt) :

Đối với các luật mà ký tự kết thúc ở bên phải dấu chấm, luật này sẽ được chuyển sang cột Ii+1 với dấu chấm được dịch ra sau ký tự kết thúc. Điều này có nghĩa là, với sẽ được đổi thành trong cột Ii+1.

1. Complete (Hoàn thiện) :

Khi có luật thì sao chép và đổi trong cột Ij thành trong cột Ii.

* 1. *Ví dụ minh họa :*

Cho văn phạm G với tập luật sinh như sau :

|  |  |  |  |
| --- | --- | --- | --- |
| S 🡪 N VP | S 🡪 VP AP | NP 🡪 Ad N | VP 🡪 V N |
| S 🡪 N AP | S 🡪 NP VP | NP 🡪 N N | VP 🡪 V NP |
| S 🡪 P VP | S 🡪 NP AP | NP 🡪 N A | AP 🡪 R A |
| N 🡪 tôi | môn | CTD | V 🡪 học | Ad 🡪 Hôm nay |  |

Chú thích :

|  |  |  |  |
| --- | --- | --- | --- |
| S : câu | P : đại từ | Ad: trạng từ | VP: cụm động từ |
| N : danh từ | A : tính từ | AP : cụm động từ | NP: cụm danh từ |
| V : động từ | R : phụ từ |  |  |

Chuỗi nhập vào : “Hôm nay, tôi học môn CTD”

Với thuật toán Earley ta sẽ có bảng sau :

|  |  |  |  |
| --- | --- | --- | --- |
| Cột 0 | Cột 1 | Cột 2 | Cột 3 |
| ROOT • S, 0 | Ad Hôm nay •, 0 | V học •, 1 | N môn •, 2 |
| S • N VP, 0 | N • tôi, 0 | VP V • N, 1 | N CTD •, 2 |
| S • N AP, 0 | NP Ad N •, 0 | VP V • NP, 1 | NP N N •, 2 |
| S • P VP, 0 | S NP • VP, 0 | NP • Ad N, 2 | VP V NP •, 1 |
| S • VP AP, 0 | S NP • AP, 0 | NP • N N, 2 | NP Ad N •, 0 |
| S • NP VP, 0 | NP • Ad N, 1 | NP • N A, 2 | S NP VP •, 0 |
| S • NP AP, 0 | NP • N N, 1 | N • tôi, 2 | ROOT S •, 0 |
| NP • Ad N, 0 | NP • N A, 1 | N • môn, 2 |  |
| NP • N N, 0 | VP • V N, 1 | N • CTD, 2 |  |
| NP • N A, 0 | VP • V NP, 1 |  |  |
| VP • V N, 0 | N • tôi, 1 |  |  |
| VP • V NP, 0 | N • môn, 1 |  |  |
| N • tôi, 0 | N • CTD, 1 |  |  |
| N • môn, 0 | Ad • Hôm nay, 1 |  |  |
| N • CTD, 0 | V • học, 1 |  |  |
| Ad • Hôm nay, 0 |  |  |  |
| V • học, 0 |  |  |  |

1. Chương trình mô phỏng :

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace Earley\_Parser

{

#region Khoi tao ctdl can thiet

// ctdl cho du lieu token

public class db\_token :IEquatable<db\_token>

{

public string name; // gia tri token

public string type; // loai token

public db\_token(string val, string typ)

{

name = val;

type = typ;

}

public bool Equals(db\_token item)

{

if (name == item.name && type == item.type)

return true;

else

return false;

}

public override bool Equals(object obj)

{

return base.Equals(obj);

}

public string Show()

{

return type + " -> " + name;

}

public override int GetHashCode()

{

return base.GetHashCode();

}

}

// ctdl cho tap luat sinh

class rule

{

// res --> opt1 opt2

public string opt1;

public string opt2;

public string res;

public string Show()

{

return res + " -> " + opt1 + " " + opt2;

}

}

// ctdl cau

class sentence

{

public List<string> list\_string;

public sentence()

{

list\_string = new List<string>();

}

public sentence(string input)

{

list\_string = new List<string>();

string tmp = "";

for (int i = 0; i < input.Length; i++)

{

if (input[i] != ' ' && input[i] != '$')

tmp += input[i].ToString();

else if (input[i] == ' ' || input[i] == '$')

{

list\_string.Add(tmp);

tmp = "";

}

}

}

}

// ctdl token

class token

{

public string value; // gia tri goc, phan tach tu cau nhap vao

public string token\_name; // gia tri token, truy van tu du lieu db\_token

public token()

{

value = token\_name = "";

}

public token(string val, string name)

{

value = val;

token\_name = name;

}

// so sanh token

public bool Compare(token t)

{

if (value == t.value && token\_name == t.token\_name)

return true;

return false;

}

}

// ctdl bang ketqua

class node

{

public List<token> tokens;

public node()

{

tokens = new List<token>();

}

// Kiem tra token co value = S|s co ton tai trong node hay khong

public bool checkS()

{

foreach (token t in tokens)

{

if (t.value == "S" || t.value == "s")

return true;

}

return false;

}

// Kiem tra token co value = val co ton tai trong node hay khong

public bool check(string val)

{

foreach (token t in tokens)

{

if (t.value == val)

return true;

}

return false;

}

// xuat ra so phan tu hien co trong danh sach token

public int sopt()

{

return tokens.Count;

}

// kiem tra token ton tai trong danh sach token, tra ve vi tri cua token

public int check(token t\_check)

{

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

if (tokens[i].value == t\_check.value && tokens[i].token\_name == t\_check.token\_name)

return i;

return -1;

}

// xuat ra man hinh toan bo token trong danh sach

public void Show()

{

if (tokens.Count == 0)

Console.WriteLine("Khong co phan tu!!!");

else

foreach (token t in tokens)

Console.WriteLine("value: " + t.value + " name: " + t.token\_name);

}

// xuat ra man hinh toan bo gia tri token trong danh sach

public void Show2()

{

if (tokens.Count == 0)

Console.WriteLine("Khong co phan tu!!!");

else

foreach (token t in tokens)

Console.Write(t.value + "\t");

}

// them token moi vao danh sach token

public void Add(token t)

{

tokens.Add(t);

}

// xoa token trong danh sach

public bool Del(token t)

{

int index = check(t);

if (index == -1)

return false;

else

tokens.RemoveAt(index);

return true;

}

// xoa toan bo token

public void Del()

{

tokens.Clear();

}

// so sanh node

public bool Compare(node n)

{

if (tokens.Count != n.tokens.Count)

return false;

else

{

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

if (!tokens[i].Compare(n.tokens[i]))

return false;

return true;

}

}

}

class tab

{

public List<node> data;

public tab()

{

data = new List<node>();

}

// them phan tu vao tab

public void Add(node n)

{

data.Add(n);

}

// xoa toan bo phan tu

public void Del()

{

data.Clear();

}

// xoa phan tu trong tab theo index

public bool Del(int index)

{

if (index > data.Count)

return false;

data.RemoveAt(index);

return true;

}

// xoa phan tu trong tab

public bool Del(node n)

{

int index = check(n);

if (index == -1)

return false;

else

return Del(index);

}

// tim kiem node trong tab tra ve index neu ton tai

public int check(node n)

{

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

if (data[i].Compare(n))

return i;

return -1;

}

// xuat bang ket qua

public void Show()

{

foreach (node n in data)

{

n.Show2();

Console.WriteLine();

}

}

}

#endregion

class Program

{

#region Ham xu ly

// Doc file luu du lieu phan tach tokens

static List<db\_token> ReadDB(String src, string type)

{

string[] lines = File.ReadAllLines(src);

List<db\_token> results = new List<db\_token>();

int i = 0;

foreach (string line in lines)

{

results.Insert(i, new db\_token(line, type));

i++;

}

return results;

}

// Doc file luu du lieu tap luat sinh

static List<rule> ReadRules(String src, string type)

{

string[] lines = File.ReadAllLines(src);

List<rule> results = new List<rule>();

rule tmp2 = new rule();

int i,j, k;

i = j = k = 0;

string tmp = "";

foreach (string line in lines)

{

foreach (char c in line)

{

if (c == ' ' || c == '$')

{

switch (j)

{

case 0:

tmp2.res = tmp;

break;

case 1:

tmp2.opt1 = tmp;

break;

case 2:

tmp2.opt2 = tmp;

results.Insert(i, tmp2);

break;

}

j++;

tmp = "";

k = 0;

}

else if (c != '$')

{

tmp = tmp.Insert(k, c.ToString());

k++;

}

}

j = k = 0;

tmp2 = new rule();

i++;

}

return results;

}

static string NhapCau()

{

// Nhap cau tu ban phim

Console.WriteLine("Nhap cau(Ket thuc bang ky tu $ hoac ky tu khoang trang) : ");

return Console.ReadLine();

}

// kiem tra gia tri token co nam trong gia tri res cua tap luat sinh (res -> opt1 opt2)

static bool checkResRule(string res, List<rule> rules)

{

foreach (rule r in rules)

if (r.res == res)

return true;

return false;

}

// kiem tra gia tri token S co nam trong gia tri res cua tap luat sinh (res -> opt1 opt2)

static bool checkResRule(List<rule> rules)

{

foreach (rule r in rules)

if (r.res == "S")

return true;

return false;

}

#endregion

static void Main()

{

#region khai bao, load du lieu ban dau

// dia chi cac file du lieu token

string src\_noun = @"E:\CaoHoc\Chuong trinh dich\Bai tap\PP Phan tich cu phap Earley\Earley\_Parser\nouns.txt";

string src\_verb = @"E:\CaoHoc\Chuong trinh dich\Bai tap\PP Phan tich cu phap Earley\Earley\_Parser\verbs.txt";

string src\_adj = @"E:\CaoHoc\Chuong trinh dich\Bai tap\PP Phan tich cu phap Earley\Earley\_Parser\adjectives.txt";

// dia chi cac file du lieu tap luat sinh

string src\_rule = @"E:\CaoHoc\Chuong trinh dich\Bai tap\PP Phan tich cu phap Earley\Earley\_Parser\rules.txt";

// Load tap luat sinh, du lieu token

// tap danh tu

List<db\_token> nouns = ReadDB(src\_noun, "noun");

// tap dong tu

List<db\_token> verbs = ReadDB(src\_verb, "verb");

// tap trang tu

List<db\_token> adjs = ReadDB(src\_adj, "adjective");

// tap luat sinh

List<rule> rules = ReadRules(src\_rule, "rules");

#endregion

#region Nhap cau, tien xu ly cau

// cau nguoi su dung nhap vao

string input;

// Cau sau khi da phan tach

sentence sen;

// Cau sau khi da nhan dien tokens

node tokens = new node();

int flag;

flag = 0;

string tmp;

tmp = "";

// Nhap cau can xu ly

input = NhapCau();

Console.WriteLine("Input: "+ input);

// Phan tach cau

sen = new sentence(input);

// Chuyen doi cau thanh dang token

#region Chuyen doi cau thanh dang tokens

foreach (string t in sen.list\_string)

{

if (flag <= 1)

{

if (nouns.Contains(new db\_token(t, "noun")))

tokens.Add(new token(t, "N"));

else if (verbs.Contains(new db\_token(t, "verb")))

tokens.Add(new token(t, "V"));

else if (adjs.Contains(new db\_token(t, "adjective")))

tokens.Add(new token(t, "Ad"));

else

{

if (tmp == "")

tmp = t;

else

tmp = tmp + " " + t;

flag++;

}

}

else

{

if (tmp != "")

{

if (nouns.Contains(new db\_token(tmp, "noun")))

{

tokens.Add(new token(tmp, "N"));

flag = 0;

}

else if (verbs.Contains(new db\_token(tmp, "verb")))

{

tokens.Add(new token(tmp, "V"));

flag = 0;

}

else if (adjs.Contains(new db\_token(tmp, "adjective")))

{

tokens.Add(new token(tmp, "Ad"));

flag = 0;

}

else

{

flag++;

break;

}

tmp = "";

}

if (nouns.Contains(new db\_token(t, "noun")))

tokens.Add(new token(t, "N"));

else if (verbs.Contains(new db\_token(t, "verb")))

tokens.Add(new token(t, "V"));

else if (adjs.Contains(new db\_token(t, "adjective")))

tokens.Add(new token(t, "Ad"));

else

{

if (tmp == "")

tmp = t;

else

tmp = tmp + " " + t;

}

}

}

if (tmp != "")

{

if (nouns.Contains(new db\_token(tmp, "noun")))

tokens.Add(new token(tmp, "N"));

else if (verbs.Contains(new db\_token(tmp, "verb")))

tokens.Add(new token(tmp, "V"));

else if (adjs.Contains(new db\_token(tmp, "adjective")))

tokens.Add(new token(tmp, "Ad"));

tmp = "";

flag = 0;

}

if (flag == 3)

Console.WriteLine("Khong the phan tach cau!!!");

#endregion

#endregion

#region Bang ket qua

tab result = new tab();

bool f;

f = false; // false : 1; true : 2

bool f1; // co danh dau co ket qua hay khong the phan tach cau dua theo bo luat sinh dau vao.

f1 = false; // mac dinh la khong the phan tach cau

bool f2;

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

{

if (f)

{ }

else

{

if (tokens.tokens[i].value == "S")

{

// Truong hop hoan thanh banh ket qua.

if (tokens.tokens.Count == 1)

f1 = true;

// Truong hop xet cau phan tich xuat hien S nhung cac phan tu cua cau chua xet het (buoc phan tach hien tai va truoc do phan tach sai)

if (i != 1)

{ }

else if (i == 1)

// Truong hop cau phan tich xuat hien S nam trong vong 2 buoc

break;

}

else

{

if (checkResRule(tokens.tokens[i].value, rules))

{

}

}

}

}

#endregion

#region Tao Bang Earley Parser

#endregion

#region Xuat bang Ket qua trich xuat tu Earley Parser

#endregion

#region Xuat ket qua

#endregion

Console.ReadKey(); // tam dung xem ket qua

}

}

}