Khoa Công nghệ thông tin

Lớp: TH05

Môn: Trí tuệ nhân tạo

BÁO CÁO ĐỒ ÁN 2

SUY DIỄN TRÊN LOGIC BẬC NHẤT

Sinh viên thực hiện:

* MSSV: 0512175, Họ tên: Nguyễn Đăng Khoa
* MSSV: 0512206, Họ tên: Nguyễn Thành Luân

# Nội dung:

Cài đặt động cơ suy diễn lùi trên logic bậc nhất.

# Mục tiêu:

Áp dụng thuật giải đồng nhất vào cài đặt động cơ suy diễn lùi trên logic vị từ bậc nhất.

# Cài đặt:

* Cấu trúc của chương trình được thiết kế theo hướng đối tượng, gồm có các lớp chính sau:



* Danh sách các thuộc tính, phương thức của từng lớp:
* \_Term: lớp trừu tượng, cung cấp các phương thức thuần ảo để các lớp con định nghĩa lại. Là lớp thể hiện biểu thức.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
| 1 | mName | String | Tên của biểu thức |
| 2 | arrPatternObject | \_Term[] | Được khai báo static. Dùng cho việc tạo đối tượng mẫu. Gồm 3 phần tử, lần lượt là: \_Variable, \_Constant, \_Composite. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
| 1 | ToString | String | Trả về thông tin của đối tượng |
| 2 | NewObject | \_Term | Hàm thuần ảo, dùng trong việc tạo mẫu đối tượng. Các lớp con sẽ định nghĩa hàm này để tạo ra đối tượng tương ứng với lớp đó. |
| 3 | GetType | \_TypeOfTerm : kiểu enum gồm 3 phần tử {VARIABLE, CONSTANT, COMPOSITE} | Hàm thuần ảo. Trả về loại của đối tượng tương ứng với từng lớp con. |
| 4 | CompareTo | Boolean | Hàm thuần ảo. Kiểm tra 2 biểu thức có bằng nhau không. (True: bằng nhau; False: khác nhau) |
| 5 | IsAppear | Boolean | Hàm thuần ảo. Kiểm tra một biểu thức có nằm trong biểu thức kia hay không. |
| 6 | Replace | \_Term | Hàm thuần ảo. Thay thế các biến của biểu thức bằng các phép thế tương đương (phép thế là đầu vào của hàm). Kết quả là trả ra một biểu thức mới sau khi đã thay thế các biến. |
| 7 | GetListOfVariable | ArrayList | Hàm thuần ảo. Trả về tất cả các biến có trong biểu thức, kể cả những biến trùng tên. |
| 8 | GetListOfVariable2 | ArrayList | Hàm thuần ảo. Trả về tất cả các biến có trong biểu thức, nếu có nhiều biến trùng tên thì chỉ nhận một biến. |

* \_Constant: kế thừa từ lớp \_Term. Là lớp thể hiện hằng trong một biểu thức.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
| Không có thuộc tính mới | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
| Không có phương thức mới | | | |

* \_Variable: kế thừa từ lớp \_Term. Là lớp thể hiện biến trong một biểu thức.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
| Không có thuộc tính mới | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
| Không có phương thức mới | | | |

* \_Composite: vừa sử dụng, vừa kế thừa lớp \_Term. Áp dụng thiết kế mẫu Composite để thiết kế lớp này. Là lớp thể hiện hàm hoặc vị từ trong biểu thức.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
| 1 | mComponents | ArrayList | Chứa danh sách các biểu thức (hằng, biến, hàm hoặc vị từ). Mỗi phần tử trong mảng trả về là một \_Term. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
| 1 | GetArguments | ArrayList | Lấy toàn bộ các tham số của vị từ (hàm). Mỗi phần tử trong mảng trả về là một \_Term |
| 2 | ExtraName | \_Composite | Trả về vị từ (hàm) mới mà các biến đã được đổi tên. Đổi tên bằng cách thêm một chuỗi extra (tham số đầu vào) vào đầu tên biến. |

* \_Unify : lớp dùng để đồng nhất hai biểu thức.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
| 1 | mArg1 | \_Term | Là biểu thức cần đồng nhất. |
| 2 | mArg2 | \_Term | Là biểu thức cần đồng nhất. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
| 1 | Run | Boolean | Chạy thuật toán đồng nhất trên 2 biểu thức mArg1, mArg2. (True: đồng nhất thành công; False: không đồng nhất được). Nếu đồng nhất thành công thì trả về một tập thế (tham số của hàm). |

* \_Element: sử dụng lớp \_Term, \_Variable. Là lớp thể hiện thông tin của một phần tử trong phép thế.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
| 1 | mArg1 | \_Variable | Biến sẽ được thay thế bằng biểu thức thế mArg2 trong một phần tử của phép thế đồng nhất. |
| 2 | mArg2 | \_Term | Biểu thức thế trong một phần tử của phép thế đồng nhất. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
| 1 | IsBelong | Boolean | Kiểm tra một biến, hoặc một biểu thức thế có nằm trong đối tượng \_Element hay không. (True: nằm trong; False: không nằm trong). |
| 2 | CompareTo | Boolean | Kiểm tra 2 đối tượng \_Element có giống nhau hay không. (True: giống nhau; False: khác nhau). |
| 3 | ToString | String | Trả về chuỗi thể hiện thông tin của đối tượng \_Element. |

* \_MGU: sử dụng lớp \_Element. Là lớp thể hiện phép thế.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
| 1 | mContent | ArrayList | Chứa danh sách các \_Element, thể hiện cho phép thế. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
| 1 | Add | void | Thêm một đối tượng \_Element vào phép thế. |
| 2 | Find | \_Element | Tìm xem trong phép thế có x (là biến hoặc biểu thức) hay không. Nếu có thì trả về vị trí xuất hiện của x (ở đây dùng \_Element để thể hiện vị trí). |
| 3 | CompareTo | Boolean | Kiểm tra 2 phép thế có giống nhau hay không. |
| 4 | ToString | String | Trả về chuỗi thể hiện thông tin của phép thế. |
| 5 | operator+ | \_MGU | Là một toán tử. Trả về một phép thế mới là kết quả của việc hợp 2 phép thế lại với nhau. |

* \_Horn: sử dụng lớp \_Term. Là lớp thể hiện biểu thức dạng Horn.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
| 1 | mLeft | ArrayList | Chứa các biểu thức nằm bên trái của biểu thức dạng Horn. Mỗi phần tử của mảng là một \_Term |
| 2 | mRight | \_Term | Biểu thức bên phải của biểu thức dạng Horn. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
| 1 | ToString | String | Trả về chuỗi thể hiện thông tin của biểu thức dạng Horn. |

* \_KB: sử dụng lớp \_Horn. Là lớp thể hiện thông tin của tập cơ sở tri thức.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
| 1 | mContent | ArrayList | Chứa danh sách các biểu thức Horn. Biểu diễn tập cơ sở tri thức. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
| 1 | Add | void | Thêm một biểu thức dạng Horn vào tập KB. |
| 2 | ToString | String | Trả về chuỗi thể hiện thông tin của tập cơ sở tri thức. |

* \_BackwardChaining: sử dụng lớp \_Term, \_KB. Lớp dùng để thực hiện thuật giải suy diễn lùi.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
| 1 | mKB | \_KB | Tập cơ sở tri thức, dùng khi thực hiện suy diễn lùi. Tập này đã được thay đổi tất cả các tên biến sau khi đưa vào để phù hợp với cách cài đặt thuật toán. |
| 2 | mResult | ArrayList | Chứa các phép thế (nếu có nhiều cách thế) sau khi thực hiện hàm Run (hàm chạy thuật giải suy diễn lùi). |
| 3 | mG | \_Term | Là biểu thức dùng để hỏi. Biểu thức này đã được biến đổi cho phù hợp với cách cài đặt thuật toán. |
| 4 | mVarible | ArrayList | Chỉ được dùng trong thuật giải suy diễn lùi. Không dùng để lưu trữ các thông tin đầu vào. Lưu tất cả các biến đã xuất hiện trong khi thực hiện thuật giải. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
| 1 | Run | void | Hàm thực hiện thuật giải suy diễn lùi. Hàm nhận một tham số đầu vào, là một câu truy vẫn, dựa vào tập KB đã được khởi tạo ban đầu để suy diễn. Dùng hàm này có thể hỏi nhiều câu của cũng một KB mà không cần phải khởi tạo lại KB, chỉ đưa câu hỏi khác vào là được. |
| 2 | GetResult | String | Trả về một chuỗi thể hiện kết quả của câu truy vẫn sau khi chạy hàm Run. |

# Hướng dẫn sử dụng

Chạy bằng dòng lệnh với cấu trúc lệnh như sau:

0512175\_0512206 <input file> <output file>

Input file: tên tập tin input.

Output file: tên tập tin output.

# Thử nghiệm:

* Cho tập tin Input như bảng bên dưới, và có kết quả như sau:

|  |  |
| --- | --- |
| **Input File** | **Output file** |
| KB  American(x), Weapon(y), Sells(x,y,z), Hostile(z) => Criminal(x)  Owns(Nono, M1)  Missile(M1)  Missile(x), Owns(Nono, x) => Sells(West,x,Nono)  Missile(x) => Weapon(x)  Enemy(x, America) => Hostile(x)  American(West)  Enemy(Nono, America)  MotherOf(Jane, Billy)  FatherOf(John, Billy)  FatherOf(Sam, John)  MotherOf(Mary,John)  MotherOf(x,y) => ParentOf(x,y)  FatherOf(x,y) => ParentOf(x,y)  ParentOf(x,z),ParentOf(z,y) => AncestorOf(x,y)  G(x,y), G(y,z) => F(x,z)  ENDKB  G: Criminal(x)  G: Criminal(West)  G: G(Art, z)  G: x  G: MotherOf(x, y)  G: AncestorOf(Sam,Billy)  G: AncestorOf(Mary,Jane)  G: AncestorOf(x,y)  G: AncestorOf(Sam, y)  G: AncestorOf(x, Billy)  G: ParentOf(x, y)  G: MotherOf(Jane,Billy)  G: FatherOf(Jane,Billy)  G: My  G: F(x,y)  G: FatherOf(Sam, John) | Criminal(x)  x/West  Criminal(West)  True  G(Art, z)  False  x  x/Criminal(West)  x/Owns(Nono, M1)  x/Missile(M1)  x/Sells(West, M1, Nono)  x/Weapon(M1)  x/Hostile(Nono)  x/American(West)  x/Enemy(Nono, America)  x/MotherOf(Jane, Billy)  x/FatherOf(John, Billy)  x/FatherOf(Sam, John)  x/MotherOf(Mary, John)  x/ParentOf(Jane, Billy)  x/ParentOf(Mary, John)  x/ParentOf(John, Billy)  x/ParentOf(Sam, John)  x/AncestorOf(Mary, Billy)  x/AncestorOf(Sam, Billy)  MotherOf(x, y)  x/Jane, y/Billy  x/Mary, y/John  AncestorOf(Sam, Billy)  True  AncestorOf(Mary, Jane)  False  AncestorOf(x, y)  x/Mary, y/Billy  x/Sam, y/Billy  AncestorOf(Sam, y)  y/Billy  AncestorOf(x, Billy)  x/Mary  x/Sam  ParentOf(x, y)  x/Jane, y/Billy  x/Mary, y/John  x/John, y/Billy  x/Sam, y/John  MotherOf(Jane, Billy)  True  FatherOf(Jane, Billy)  False  My  False  F(x, y)  False  FatherOf(Sam, John)  True |

* Kết quả cho ra hoàn toàn đúng.

# Kết luận

* Với cách cài đặt thuật toán này, chương trình có thể trả lời được các câu truy vấn dựa vào tập KB ban đầu.
* Trong trường hợp tập KB ban đầu có mạch vòng (A 🡪 B, B 🡪 C, C 🡪 A) thì chương trình không chạy được, vì chưa kiểm tra trường hợp này.
* Chưa làm chức năng nâng cao: kiểm tra lỗi nhập.
* Chưa tối ưu hóa các trường hợp lặp lại của biểu thức, nên kết quả sẽ ra chậm nếu tập KB lớn.