

## 1. Giải thích sơ lược về quá trình cài đặt thuật toán và đánh giá về mức độ hoàn thành:

Quá trình cài đặt hàm **PL\_RESOLUTION** đều dựa trên gợi ý tham khảo từ sách Artificial Intelligence: A Modern Approach, Third Edition, chương 7, hình 7.12 được giáo viên hướng dẫn. Hàm **PL\_RESOLVE** dùng để hợp giải 2 mệnh đề ban đầu và cho ra 1 một mệnh đề mới.

Trong quá trình cài đặt, có thêm một số hàm hỗ trợ cho việc đọc ghi file, có các hàm xóa đi những mệnh đề mới / symbol bị trùng lặp. Tất cả các hàm và khai báo đều nằm trong file *ps4.py*.

### Yêu cầu đã thực hiện:

- + Đọc dữ liệu đầu vào và chuyển sang logical clauses.
- + Cài đặt thuật toán cho việc hợp giải trên logic mệnh đề.
- + Các step suy diễn phát sinh đủ mệnh đề và trả ra kết luận đúng.
- + Theo đúng form định dạng dữ liệu mô tả trong file input và output, các file input và output đều nằm đúng trong folder INPUT/OUTPUT.
- + Chạy đủ tối thiểu 5 test case khác nhau, 3 trường hợp trả kết quả YES, 2 trường hợp trả kết quả NO ứng với các file từ file output\_1.txt  
→ output\_5.txt.



### Yêu cầu chưa thực hiện:

- Các literal trong cùng mệnh đề chưa được xếp theo thứ tự chữ cái.
- Chưa cài đặt cho file run.sh để thực thi chương trình nên việc thực thi sẽ thay thế bằng cách run trên command line: ***python ps4.py***

## 2. Source code:

- + Folder Input: chứa 5 file Input\_i.txt để đọc dữ liệu mẫu.
- + Folder Output: chứa 5 file Output\_i.txt tương ứng với các kết quả trả ra ứng với từng file Input.txt.
- + File ps4.py là file thực thi chương trình.

### 3. Các Test case:

#### Test Case 1

Input: input\_1.txt

Output: output\_1.txt | 'YES'

```
*****
Base knowledge: A OR B OR C ; -B OR D ; -A OR D ; B ; -D
Alpha: -A

Available Clauses: A OR B OR C ; -B OR D ; -A OR D ; B ; -D ; --A
(A OR B OR C) and (-B OR D ) === Resolve ==> (A OR C OR D )
(A OR B OR C) and (-A OR D ) === Resolve ==> (B OR C OR D )
(-B OR D ) and (B ) === Resolve ==> (D )
(-B OR D ) and (-D ) === Resolve ==> (-B )
(-A OR D ) and (-D ) === Resolve ==> (-A )
(-A OR D ) and (--A ) === Resolve ==> (D )
n Added clauses = 5

Available Clauses: A OR B OR C ; -B OR D ; -A OR D ; B ; -D ; --A ; A OR C OR D ; B OR C OR D ; D ; -B ; -A
(A OR B OR C) and (-B ) === Resolve ==> (A OR C )
(A OR B OR C) and (-A ) === Resolve ==> (B OR C )
(-B OR D ) and (B OR C OR D) === Resolve ==> (C OR D )
(-A OR D ) and (A OR C OR D) === Resolve ==> (C OR D )
(B ) and (-B ) === Resolve ==> ({} )
(-D ) and (A OR C OR D) === Resolve ==> (A OR C )
(-D ) and (B OR C OR D) === Resolve ==> (B OR C )
(-D ) and (D ) === Resolve ==> ({} )
(--A ) and (-A ) === Resolve ==> ({} )
(A OR C OR D) and (-A ) === Resolve ==> (C OR D )
(B OR C OR D) and (-B ) === Resolve ==> (C OR D )
n Added clauses = 4

KB entails alpha: True
*****
```



## Test Case 2

Input: input\_2.txt

Output: output\_2.txt | 'YES'

\*\*\*\*\*

Base knowledge: X OR Y OR Z ; -Y OR X ; -Z OR X

Alpha: X

Available Clauses: X OR Y OR Z ; -Y OR X ; -Z OR X ; -X

(X OR Y OR Z) and (-Y OR X ) === Resolve ==> (X OR Z )

(X OR Y OR Z) and (-Z OR X ) === Resolve ==> (X OR Y )

(X OR Y OR Z) and (-X ) === Resolve ==> (Y OR Z )

(-Y OR X ) and (-X ) === Resolve ==> (-Y )

(-Z OR X ) and (-X ) === Resolve ==> (-Z )

n Added clauses = 5

Available Clauses: X OR Y OR Z ; -Y OR X ; -Z OR X ; -X ; X OR Z ; X OR Y ; Y OR Z ; -Y ; -Z

(-Y OR X ) and (X OR Y ) === Resolve ==> (X )

(-Z OR X ) and (X OR Z ) === Resolve ==> (X )

(-X ) and (X OR Z ) === Resolve ==> (Z )

(-X ) and (X OR Y ) === Resolve ==> (Y )

(X OR Z ) and (-Z ) === Resolve ==> (X )

(X OR Y ) and (-Y ) === Resolve ==> (X )

(Y OR Z ) and (-Y ) === Resolve ==> (Z )

(Y OR Z ) and (-Z ) === Resolve ==> (Y )

n Added clauses = 3

Available Clauses: X OR Y OR Z ; -Y OR X ; -Z OR X ; -X ; X OR Z ; X OR Y ; Y OR Z ; -Y ; -Z ; X ; Z ; Y

(-X ) and (X ) === Resolve ==> ({} )

(-Y ) and (Y ) === Resolve ==> ({} )

(-Z ) and (Z ) === Resolve ==> ({} )

n Added clauses = 1

KB entails alpha: True

\*\*\*\*\*



## Test Case 3

Input: input\_3.txt

Output: output\_3.txt | 'YES'

\*\*\*\*\*

Base knowledge: -B OR A ; -C OR B ; -D OR C ; D

Alpha: A

Available Clauses: -B OR A ; -C OR B ; -D OR C ; D ; -A

(-B OR A ) and (-C OR B ) === Resolve ==> (-C OR A )

(-B OR A ) and (-A ) === Resolve ==> (-B )

(-C OR B ) and (-D OR C ) === Resolve ==> (-D OR B )

(-D OR C ) and (D ) === Resolve ==> (C )

n Added clauses = 4

Available Clauses: -B OR A ; -C OR B ; -D OR C ; D ; -A ; -C OR A ; -B ; -D OR B ; C

(-B OR A ) and (-D OR B ) === Resolve ==> (-D OR A )

(-C OR B ) and (-B ) === Resolve ==> (-C )

(-C OR B ) and (C ) === Resolve ==> (B )

(-D OR C ) and (-C OR A ) === Resolve ==> (-D OR A )

(D ) and (-D OR B ) === Resolve ==> (B )

(-A ) and (-C OR A ) === Resolve ==> (-C )

(-C OR A ) and (C ) === Resolve ==> (A )

(-B ) and (-D OR B ) === Resolve ==> (-D )

n Added clauses = 5

Available Clauses: -B OR A ; -C OR B ; -D OR C ; D ; -A ; -C OR A ; -B ; -D OR B ; C ; -D OR A ; -C ; B ; A ; -D

(D ) and (-D ) === Resolve ==> ({} )

(-A ) and (A ) === Resolve ==> ({} )

(-B ) and (B ) === Resolve ==> ({} )

(C ) and (-C ) === Resolve ==> ({} )

n Added clauses = 1

KB entails alpha: True

\*\*\*\*\*





## Test Case 4

Input: input\_4.txt

Output: output\_4.txt | 'NO'

```
*****
Base knowledge: -N OR L ; -N OR -T OR M ; -L OR R ; L ; N OR T ; -T OR R
Alpha: M

Available Clauses: -N OR L ; -N OR -T OR M ; -L OR R ; L ; N OR T ; -T OR R ; -M
(-N OR L ) and (-L OR R ) === Resolve ==> (-N OR R )
(-N OR L ) and (N OR T ) === Resolve ==> (L OR T )
(-N OR -T OR M) and (-M ) === Resolve ==> (-N OR -T )
(-L OR R ) and (L ) === Resolve ==> (R )
(N OR T ) and (-T OR R ) === Resolve ==> (N OR R )
n Added clauses = 5

Available Clauses: -N OR L ; -N OR -T OR M ; -L OR R ; L ; N OR T ; -T OR R ; -M ; -N OR R ; L OR T ; -N OR -T ; R ; N OR R
(-N OR L ) and (N OR R ) === Resolve ==> (L OR R )
(-N OR -T OR M) and (L OR T ) === Resolve ==> (-N OR L OR M )
(-N OR -T OR M) and (N OR R ) === Resolve ==> (-T OR M OR R )
(-L OR R ) and (L OR T ) === Resolve ==> (R OR T )
(N OR T ) and (-N OR R ) === Resolve ==> (R OR T )
(-T OR R ) and (L OR T ) === Resolve ==> (L OR R )
n Added clauses = 4

Available Clauses: -N OR L ; -N OR -T OR M ; -L OR R ; L ; N OR T ; -T OR R ; -M ; -N OR R ; L OR T ; -N OR -T ; R ; N OR R ; L OR R ;
-N OR L OR M ; -T OR M OR R ; R OR T
(-N OR -T OR M) and (R OR T ) === Resolve ==> (-N OR M OR R )
(-L OR R ) and (-N OR L OR M) === Resolve ==> (-N OR M OR R )
(N OR T ) and (-N OR L OR M) === Resolve ==> (L OR M OR T )
(N OR T ) and (-T OR M OR R) === Resolve ==> (M OR N OR R )
(L OR T ) and (-T OR M OR R) === Resolve ==> (L OR M OR R )
(N OR R ) and (-N OR L OR M) === Resolve ==> (L OR M OR R )
(-T OR M OR R) and (R OR T ) === Resolve ==> (M OR R )
n Added clauses = 5

Available Clauses: -N OR L ; -N OR -T OR M ; -L OR R ; L ; N OR T ; -T OR R ; -M ; -N OR R ; L OR T ; -N OR -T ; R ; N OR R ; L OR R ;
-N OR L OR M ; -T OR M OR R ; R OR T ; -N OR M OR R ; L OR M OR T ; M OR N OR R ; L OR M OR R ; M OR R
(-L OR R ) and (L OR M OR T) === Resolve ==> (M OR R OR T )
(N OR T ) and (-N OR M OR R) === Resolve ==> (M OR R OR T )
(-T OR R ) and (L OR M OR T) === Resolve ==> (L OR M OR R )
(-T OR M OR R) and (L OR M OR T) === Resolve ==> (L OR M OR R )
n Added clauses = 2

Available Clauses: -N OR L ; -N OR -T OR M ; -L OR R ; L ; N OR T ; -T OR R ; -M ; -N OR R ; L OR T ; -N OR -T ; R ; N OR R ; L OR R ;
-N OR L OR M ; -T OR M OR R ; R OR T ; -N OR M OR R ; L OR M OR T ; M OR N OR R ; L OR M OR R ; M OR R ; M OR R OR T ; L OR M OR R

KB entails alpha: False
*****
```



## Test Case 5

Input: input\_5.txt

Output: output\_5.txt | 'NO'

```
*****
```

```
Base knowledge: -R OR Q ; R OR T ; Q OR R OR T ; -Q OR R
```

```
Alpha: P
```

```
Available Clauses: -R OR Q ; R OR T ; Q OR R OR T ; -Q OR R ; -P
```

```
(-R OR Q ) and (R OR T ) == Resolve ==> (Q OR T )
```

```
(-R OR Q ) and (Q OR R OR T) == Resolve ==> (Q OR T )
```

```
(Q OR R OR T) and (-Q OR R ) == Resolve ==> (R OR T )
```

```
n Added clauses = 2
```

```
Available Clauses: -R OR Q ; R OR T ; Q OR R OR T ; -Q OR R ; -P ; Q OR T ; R OR T
```

```
KB entails alpha: False
```

```
*****
```