1. Xét giải thuật Best First Search với hàm lượng giá: f(n) = g(n) + h(n).

- g(n) là chi phí tìm kiếm từ trạng thái khởi đầu đến nút n

- h(n) là chi phí từ nút n đến trạng thái mục tiêu.

Nếu h(.) được sử dụng làm tiêu chí chọn trạng thái hiện thời tốt nhất trong giải thuật Best Firsr Search 🡪 Đó gọi là sự tìm kiếm tham lam (Greedy).

* Vì h(n) chỉ quan tâm đến làm thế nào để từ trạng thái hiện tại đến trạng thái mục tiêu với chi phí nhỏ nhất mà không tính đến chi phí đã mất để đến được trạng thái hiện tại.
* Nếu định nghĩa h(n) là số tầng tách biệt giữa tầng chứa nút trạng thái n, đến tầng chứa trạng thái mục tiêu, thì giải thuật **tìm kiếm tham lam** sẽ trở thành giải thuật **Depth First Search**. Vì càng sâu thì càng gần tầng chứa trạng thái mục tiêu.

Nếu g(.) được sử dụng thì Đó là sự tìm kiếm đẳng phí (Uniform-cost).

* Vì các trạng thái ở biên của cây trạng thái trong quá trình mở rộng, sẽ có lượng giá gần bằng nhau, vì trạng thái nào có lượng giá nhỏ sẽ được lấy ra thay bằng các trạng thái kế tiếp có lượng giá lớn hơn.
* Nếu định nghĩa g(n) là số tầng của nút n, tức chiều cao của cây trạng thái tính từ nút gốc đến nút n, thì giải thuật **tìm kiếm đẳng phí** sẽ trở thành giải thuật **Breadth First Seart.** Vì các trạng thái ở cùng 1 tầng sẽ có lượng giá bằng nhau và nhỏ hơn lượng giá của các trạng thái ở tầng sâu hơn

2.

**<=50**

min

**<=28**

**>=48**

**>=28**

**>=28**

**>=-38**

**<=-38**

**>=7**

**<=50**

**>=50**

**>=100**

max

**>=18**

**>=50**

**<=18**

**>=74**

min

max

-17

28

-22

48

-92

-38

7

-91

100

10

50

42

14

18

74

22

3. Planing

Condition & result:

ON(A, B)

ONT(A)

**CLEAR(A)**

**HOLDING(A)**

**ARME**

**Action:**

|  |  |  |
| --- | --- | --- |
| STACK(x, y) |  |  |
| **PreCondition** | **Delete** | **Add** |
| CLEAR(y) | CLEAR(y) | ARME |
| HOLDING(x) | HOLDING(x) | ON(x, y) |
|  |  | CLEAR(x) |

|  |  |  |
| --- | --- | --- |
| UNSTACK(x, y) | | |
| **PreCondition** | **Delete** | **Add** |
| ON(x, y) | ON(x, y) | HOLDING(x) |
| CLEAR(x) | ARME | CLEAR(y) |
| ARME |  |  |

|  |  |  |
| --- | --- | --- |
| PICKUP(x) |  |  |
| **PreCondition** | **Delete** | **Add** |
| CLEAR(x) | ONT(x) | HOLDING(x) |
| ONT(x) | ARME |  |
| ARME |  |  |

|  |  |  |
| --- | --- | --- |
| PUTDOWN(x) |  |  |
| **PreCondition** | **Delete** | **Add** |
| HOLDING(x) | HOLDING(x) | ONT(x) |
|  |  | ARME |
|  |  | CLEAR(x) |

Push the Original Goal to the stack.

* If the top is compound goal, **then**:
  + Pop it
* If the top is a single unsatisfied goal **then:**
  + Replace it with an action & Push actions precondition.
* If top is action, **then**:
  + Pop it and perform the action.
* If top is empty and satisfied goal, **then**:
  + Success/ Complete

|  |
| --- |
| Start state / Database |
| ON(B, A) |
| ONT A-C-D |
| ARMEMPTY |
| CL B-C-D |

A diagram of a stack planning

Description automatically generated

|  |
| --- |
| Start state / Database (**DB**) |
| ON(B, A) |
| ONT A-C-D |
| ARMEMPTY |
| CL B-C-D |

|  |
| --- |
| Start state / DB (0) |
| ON(B, A) |
| ONT A-C-D |
| ARMEMPTY |
| CL B-C-D |

Trước tiên xét từ bảng Goal , thực hiện push các yếu tố từ bảng Goal vào 1 Goal\_stack

|  |
| --- |
| ON(C, A) |
| ON(B, D) |
| Goal\_stack |

Vì ON(C,A) là 1 câu **single unsatisfied goal** nên Pop nó ra và thay nó bằng 1 action, đồng thời thêm PreCondition của action đó

|  |  |
| --- | --- |
| **1** | **~~UNSTACK(B,A)~~** |
|  | **STACK(C, A)** |
|  | ON(B, D) |
|  | ONT A-D |
|  | ARMEMPTY |
|  | Goal\_stack |

|  |
| --- |
| Start state / DB (1) |
| ONT A-C-D |
| Holding(B) |
| CL A-C-D |

|  |  |
| --- | --- |
| **2** | **~~STACK(B, D)~~** |
|  | **PickUp(C)** |
|  | **STACK(C, A)** |
|  | ON(B, D) |
|  | ONT A-D |
|  | ARMEMPTY |
|  | Goal\_stack |

|  |
| --- |
| Start state / DB (2) |
| ONT A-C-D |
| ARMEMPTY |
| CL A-B-C |
| ON(B, D) |

|  |  |
| --- | --- |
| **3** | **~~PickUp(C)~~** |
|  | **STACK(C, A)** |
|  | ON(B, D) |
|  | ONT A-D |
|  | ARMEMPTY |
|  | Goal\_stack |

|  |
| --- |
| Start state / DB (2) |
| ONT A-C-D |
| ARMEMPTY |
| CL A-B-C |
| ON(B, D) |

|  |
| --- |
| Start state / DB (3) |
| ONT A-D |
| Holding(C) |
| CL A-B |
| ON(B, D) |

|  |  |
| --- | --- |
| **4** | **STACK(C, A)** |
|  | ON(B, D) |
|  | ONT A-D |
|  | ARMEMPTY |
|  | Goal\_stack |

|  |
| --- |
| Start state / DB (4) |
| ONT A-B-D |
| Holding(C) |
| CL A-C-D |

|  |  |  |  |
| --- | --- | --- | --- |
|  | ~~Clear(D)~~ | 14 | B14. Xét các yếu tố Top của stack:  - Đã có trong **DB (2)** 🡪Pop **(Compound Goal)** |
|  | ~~Holding(B)~~ | 13 | B13. Xét các yếu tố Top của stack:  - Đã có trong **DB (2)** 🡪Pop **(Compound Goal)** |
| **2** | **~~STACK(B, D)~~**   |  |  | | --- | --- | | **Delete** | **Add** | | CLEAR(D) | ARME | | HOLDING(B) | ON(B, D) | |  | CL(B) | | 12  15 | B12. Xử lý holding(B) từ **DB (2)** để có được ARMEMPTY. Ta dùng PutDown(B) và thêm các PreCondition của action này.  B15. Xét các yếu tố Top của stack:  - Vì nó là 1 action nên pop và thực thi,  - Và điều này làm ảnh hưởng tới database (ADD, DEL) nên cập nhật database khi dừng ở bước này  **- Thêm action này vào queue (để có solution cuối bài)** |
|  | ~~Clear(C)~~ | 9 | B9. Xét các yếu tố Top của stack:  - Đã có trong **DB (1)** 🡪Pop **(Compound Goal)** |
|  | ~~ONT(C)~~ | 10 | B10. Xét các yếu tố Top của stack:  - Đã có trong **DB (1)** 🡪Pop **(Compound Goal)** |
|  | ~~ARMEMPTY~~ | 11 | B11. Xét các yếu tố Top của stack:  - Vì *ARMEMPTY* **KHÔNG CÓ TRONG DB** nên Pop nó ra và thay nó bằng 1 action. |
| **3** | **~~PickUp(C)~~**   |  |  | | --- | --- | | **Delete** | **Add** | | ONT(C) | HOLDING(C) | | ARME |  | | 8  16 | B8. Xử lý ONT(C) từ **DB (1)** để có được Holding(C) Ta dùng Pickup (C) và thêm các PreCondition của action này.  B16. Xét các yếu tố Top của stack:  - Vì nó là 1 action nên pop và thực thi,  - Và điều này làm ảnh hưởng tới database (ADD, DEL) nên cập nhật database khi dừng ở bước này  **- Thêm action này vào queue (để có solution cuối bài)** |
|  | ~~ARMEMPTY~~ | 3 | B3. Xét các yếu tố Top của stack:  - Đã có trong **DB (0)** 🡪Pop **(Compound Goal)** |
|  | ~~Clear(B)~~ | 4 | B4. Xét các yếu tố Top của stack:  - Đã có trong **DB (0)** 🡪Pop |
|  | ~~ON(B,A)~~ | 5 | B5. Xét các yếu tố Top của stack:  - Đã có trong **DB (0)** 🡪Pop |
| **1** | **~~UNSTACK(B,A)~~**   |  |  | | --- | --- | | **Delete** | **Add** | | ON(B, A) | HOLDING(B) | | ARME | CLEAR(A) | | CL(B) |  | | 2  6 | B2. Xử lý ON(B,A) từ **DB (0)** để có được Clear(A). Ta dùng UNSTACK (B,A) và thêm các PreCondition của action này.  B6. Xét các yếu tố Top của stack:  - Vì nó là 1 action nên pop và thực thi,  - Và điều này làm ảnh hưởng tới database (ADD, DEL) nên cập nhật database khi dừng ở bước này  **- Thêm action này vào queue (để có solution cuối bài)** |
|  | ~~Clear(A)~~ | 1 | B1. Xét các yếu tố Top của stack:  - Vì clear(A) **KHÔNG CÓ TRONG DB** nên Pop nó ra và thay nó bằng 1 action. |
|  | ~~Holding(C)~~ | 7 | B7. Xét các yếu tố Top của stack:  - Vì *Holding(C)* là một **single unsatisfied goal (KHÔNG CÓ TRONG DB**) nên Pop nó ra và thay nó bằng 1 action. |
| **4** | **STACK(C, A)**   |  |  | | --- | --- | | **Delete** | **Add** | | CLEAR(A) | ARME | | HOLDING(C) | ON(C, A) | |  | CLEAR(C) | | 0  17 | B0. Xử lý ONT(A) và Clear(C) từ **DB (0)** để có được Clear(A). Ta dùng UNSTACK (B,A) và thêm các PreCondition của action này.  B17. Xét các yếu tố Top của stack:  - Vì nó là 1 action nên pop và thực thi,  - Và điều này làm ảnh hưởng tới database (ADD, DEL) nên cập nhật database khi dừng ở bước này  **- Thêm action này vào queue (để có solution cuối bài)** |
|  | ~~ON(B, D)~~ | 18 | B18. Xét các yếu tố Top của stack:  - Đã có trong **DB (4)** 🡪Pop |
|  | Goal\_stack | 19 | B19. Goal Stack Empty 🡪 Success |

**UNSTACK(B,A) 🡪 STACK(B, D) 🡪 PICKUP(C) 🡪 STACK(A,C)**