

Đại học Quốc gia Thành phố Hồ Chí Minh
Trường Đại học Công nghệ Thông tin

BÁO CÁO 1: SOKOBAN, DFS, BFS VÀ UCS



UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Lớp: CS106.M11.KHTN
Thực hiện: Nguyễn Minh Phú - 19520218
GVHD: Lương Ngọc Hoàng

I. Mô hình hóa trong Sokoban

Xem mỗi màn chơi của Sokoban như 1 bài toán tìm kiếm đường đi cho nhân vật (player) để hoàn thành mục tiêu của game. Mục tiêu là tìm ra đường đi và đẩy khối hộp về vị trí đúng tương ứng với đích của bài toán cho sẵn.

1. Trạng thái của game

Chúng ta tạm gọi tắt là trạng thái. Trạng thái của game là những thông tin về vị trí của nhân vật được người chơi điều khiển, vị trí của khối hộp, vị trí của mục tiêu được thiết kế sẵn. Và sau mỗi lần điều khiển hướng đi của người chơi, thì ta sẽ được 1 trạng thái mới. Những bước đi của nhân vật có ảnh hưởng tới khối hộp hay là không, khối hộp di chuyển bởi những bước đi của nhân vật, v.v. sẽ là 1 trạng thái của game.

2. Trạng thái khởi đầu

Trạng thái khởi đầu là trạng thái khởi tạo của cấp độ tương ứng trong trò chơi, dựa vào thiết kế được thiết lập sẵn. Chứa các thông tin như vị trí của nhân vật, vị trí của các khối hộp, vị trí của các mục tiêu

3. Trạng thái kết thúc

Trạng thái kết thúc là trạng thái khi mà mục tiêu của trò chơi được đáp ứng. Vị trí các khối hộp trùng với vị trí các mục tiêu sẽ là trạng thái kết thúc cho màn chơi tương ứng.

4. Không gian trạng thái

Không gian trạng thái là tập bao gồm những trạng thái có thể xảy ra của game. Bao gồm cả trạng thái khởi đầu và kết thúc.

5. Hàm tiến triển

Là những thuật toán tìm kiếm (DFS,BFS...) những bước đi mới, giúp chúng ta đạt được trạng thái kết thúc của màn chơi. Hàm tiến triển sẽ nhận vào trạng thái

hiện tại của game và những bước đi có thể (lên, xuống, trái, phải) được tạo ra ở hàm `LegalAction()`, kết thúc hàm tiến triển sẽ là trạng thái mới của game sau khi thực hiện 1 trong 4 hành động có thể kể trên.

II. Nhận xét

1. Depth First Search:

Là 1 thuật toán tìm kiếm duyệt theo chiều sâu. Khi áp dụng vào trò chơi, DFS sẽ tìm kiếm những trạng thái ngoài cùng (cuối cùng), và xét những hành động có thể, cập nhật trạng thái mới và cứ tiếp tục duyệt theo chiều sâu để tìm kiếm trạng thái kết thúc. Hoạt động theo cơ chế LIFO. Do đó điểm yếu lớn nhất của DFS là tốn rất nhiều bước đi của nhân vật mới giải quyết được bài toán.

2. Breadth-First Search:

Là 1 thuật toán tìm kiếm duyệt theo chiều rộng. Khi áp dụng vào trò chơi, BFS sẽ tìm kiếm những trạng thái đầu, những trạng thái kề trạng thái đang duyệt, tiếp tục cho đến khi gặp được trạng thái kết thúc. Hoạt động theo cơ chế FIFO. BFS cho số bước để đạt đến trạng thái kết thúc giảm hẳn (ngắn hơn) so với DFS với việc duyệt theo chiều rộng.

3. Uniform Cost Search

Là 1 thuật toán tìm kiếm dựa vào hàng đợi ưu tiên. UCS sẽ duyệt lần lượt các trạng thái có chi phí thấp nhất, dừng lại khi gặp trạng thái kết thúc. Do sử dụng các trạng thái có chi phí thấp, UCS cũng cho ra những bước đi ít hơn nhiều so với DFS.

III. So sánh

- DFS luôn cho số bước tới trạng thái kết thúc là nhiều nhất (đường đi dài nhất)
- BFS và UCS cho số bước tới trạng thái kết thúc là như nhau khi BFS chính là UCS với chi phí là 1
- Khi gặp những màn chơi mở, không có nhiều chướng ngại vật, BFS và UCS sẽ mất nhiều thời gian tính toán để tìm ra trạng thái kết thúc. (Ví dụ màn chơi lv5)

- Khi càng chơi tới level khó hơn, cả 3 thuật toán đều gặp vấn đề như nhau, tốn thời gian gần giống nhau để giải quyết màn chơi.
- DFS sẽ là thuật toán tệ nhất khi luôn cho đường đi dài. BFS và UCS tốt, đều cho ra đường đi tối ưu, nhưng UCS hơi chậm hơn 1 tí do dùng PriorityQueue có độ phức tạp $O(n \log n)$ so với DeQue $O(n)$