Bài tập 1: So sánh **offline** và **online search** về những điểm giống và khác nhau, nêu ví dụ về ứng dụng, thuật toán sử dụng cho mỗi loại

Giống nhau:

Khác nhau:

|  |  |  |
| --- | --- | --- |
|  | Offline search | Online search |
| Hình thức | Chạy trên máy tính trước để tìm kết quả rồi gửi cho agent thực hiện | Đặt agent vào môi trường để tự nó khám khá xung quanh và rút ra các bài học. (có nghĩa là các bước giải) |
| Cấu trúc | Gồm 5 component: Initial state, actions, transition model, goal test, cost. | Không có transition model vì nó tự khám phá môi trường. |
| Ví dụ | Khi chúng ta lấy 1 thứ gì đó ở trong nhà thì ta đủ hiểu biết để biết chỗ lấy, công việc của ta chỉ là tìm đường nào thuận tiện nhất, dựa vào kiến thức có sẵn | Khi chúng ta lấy 1 thứ gì đó ở nơi xa lạ mà còn thiếu ánh sáng. Thì ta phải khám phá nơi đó và cập nhật những chỗ đã đi qua. Mỗi hành động sẽ dựa vào nhận thức nhiều hơn. |
| Ứng dụng | Tìm đường đi ngắn từ bản đồ đã được cung cấp, sắp lịch học hoặc là lịch làm việc, giải trờ chơi 8-puzzble, …. | Khám phá những nơi con người chưa có nhiều thông tin về nó: đáy biển, sao hỏa, lòng đất (dùng robot),… |
| Thuật toán | BFS, DFS, A star, Annealing Simulated, And-or search, Hill-climbing | Online DFS, Online A\* search, AC-3 algorithm |
| Môi trường | Fully observable, partly observable, nondeterministic | Nondeterministic |

Transition Model: agent sẻ biết được mỗi hoạt động của môi trường sẻ đáp ứng như thế nào ở mỗi trạng thái. Mô phỏng chúng, nếu môi trường quá lớn -> mô phỏng không được vì quá phức tạp hoặc không đủ thông tin.

Bài tập 2: Mô tả components của bài toán 8-queens dưới dạng constraint satisfaction problem (CSP).

Variable = {các vị trí cần điền trên bàn cờ 8 queens -> 8 biến}

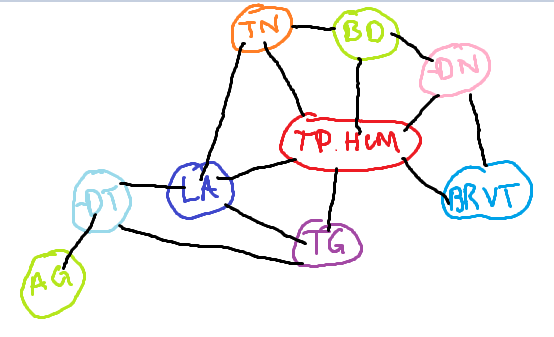
Domain = {số nguyên dương} Di = {1 -> 8: tương ứng với số quân hậu}

Constraint = {

1. Trên mỗi hàng hoặc cột chỉ có 1 quân hậu
2. Vị trí đứng của 2 quân hậu kề bên quân hậu chính giữa không được liền kề. Ví dụ: 5 6 7 -> 2 6 9, 4 6 8
3. Không có bất kì quân hậu nào tấn công được quân hậu khác (quân hậu đi được 8 hướng)

}

Bài tập 3: Vẽ constraint graph và mô tả 3 components mô tả bài toán map coloring cho các tỉnh có tên trong hình sau.



Variables = {TP.HCM, Đồng Nai, Bà Rịa Vũng Tàu, Bình Dương, Tây Ninh, Tiền Giang, Long An, Đồng Tháp, An Giang} -> 9 variables

Domain = {kí tự, chuỗi dãy màu}, Di = {Đỏ, Hồng, Xanh lam, Cam, Xanh nước biển, Tím, Xanh xẫm, Xanh lá}

Constraint = {

Hai state liền kề nhau phải khác màu nhau

Màu của state TP.HCM ≠ {Đồng Nai, Bà Rịa VT, Bình Dương, Tây Ninh, Tiền Giang, Long An}

Màu của state Đồng Nai ≠ {Bà Rịa VT, Bình Dương, TP.HCM}

Màu của state Bà Rịa VT ≠ {Đồng Nai, TP.HCM}

Màu của state Bình Dương ≠ {TP.HCM, Đồng Nai, Tây Ninh}

Màu của state Tây Ninh ≠ {Bình Dương, Long An, TP.HCM}

Màu của state Tiền Giang ≠ {Đồng Tháp, Long An, TP.HCM}

Màu của state Long An ≠ {Tây Ninh, TP.HCM, Tiền Giang, Đồng Tháp}

Màu của state Đồng Tháp ≠ {Long An, Tiền Giang, An Giang}

Màu của state An Giang ≠ {Đồng Tháp}

}

Bài tập 4: Trò chơi Sodoku cũng là một constraint satisfaction problem (CSP). Hãy mô tả 3 components của bài toán này.

*Gợi ý của Thầy:* Số biến là số ô cần điền giá trị. Sử dụng Alldiff constraint để mô tả constraint.

Variables = {Số ô hoặc số vị trí cần điền giá trị} ví dụ: ta có bàn cờ 12\*12 và 50 ô có sẵn -> số biến = 12 \* 12 – 50 = 94 biến.

Domain = {số nguyên dương} theo ví dụ trên bàn cờ 12\*12 -> Di = {1 -> 12}

Constraint = {

Giá trị các biến trên 1 hàng hoặc cột phải khác nhau.

Đối với trò chơi Sodoku 12\*12 ta sẽ chia thành 12 areas variable, mỗi area sẽ chứa 12 biến với công thức là số area = N và variable = N (N = số cột hoặc hàng).

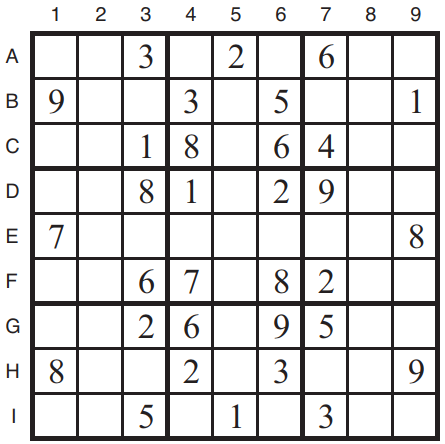
Vì thế ở mỗi area sẻ không có biến nào trùng nhau. Nghĩa là nếu 1 trong 12 biến đó chứa giá trị là 2 thì 11 biến còn lại phải khác 2.

Các biến có giá trị giống nhau trên toàn các hang hoặc cột khác nhau có tổng số biến phải bằng N cột hoặc hạng.

Bỏ qua vị trí nếu tất cả các biến đã điền ở hàng hoặc cột < option, ví dụ ở vị trí (1, 2) hàng 1 có 2 biến được điền và cột 2 có 0 biến. Variables = 2 hoăc 0 < option = 4  
}

Bài tập 5: Giải sodoku sau đây sử dụng **constraint propagation.**

*Gợi ý của thầy:* Sử dụng constrait propagation để thu hẹp domains của các ô (mỗi ô là 1 variable). Bạn có thể bắt đầu bằng cách tìm ô có domain chỉ chứa 1 giá trị sau khi duyệt quan các constraits chứa ô đó.



Ta sử dụng constraint ở bài 4 và constraint propagation để giải quyết bài 5

Constraint propagation: ta giảm domain của các biến còn lại sau mỗi lần assignment giá trị cho các biến. Put CP = Constraint propagation.

Xét vị trí (x, y) số biến của hàng x hoặc cột y < option = 5. Xét biến trên các cột, hàng, sau đó ta chia thành 9 areas variable từ đó ta ràng buộc để tạo local consistency (1 nhóm các biến nằm gần nhau để thóa điều kiện).

Xét vị trí (1, 1) => R1 = {3, 2, 6} C0 = {9, 7, 8} -> P11 = {3, 2, 6, 9, 7, 8}. Vì ô (1, 1) A1 = {1, 3, 9} -> CP11 = {4, 5}

Xét vị trí (3, 1) => R3 = {1, 8, 6, 4} C0 = {9, 7, 8} -> P31 = {1, 8, 6, 4, 9, 7}. Vì ô (3, 1) A1 = {1, 3, 9} -> CP13 = {2, 5}

Xét vị trí (4, 1) => R4 = {8, 1, 2, 9} C0 = {9, 7, 8} -> P41 = {8, 1, 2, 9, 7}. Vì ô (4, 1) A4 = {6, 7, 8} -> CP11 = {3, 4, 5}

Xét vị trí (6, 1) => R6 = {6, 7, 8, 2} C0 = {9, 7, 8} -> P61 = {6, 7, 8, 2, 9}. Vì ô (6, 1) A4 = {6, 7, 8} -> CP11 = {1, 3, 4, 5}

Xét vị trí (7, 1) => R7 = {2, 6, 9, 5} C0 = {9, 7, 8} -> P61 = {2, 6, 9, 5, 7, 8}. Vì ô (7, 1) A7 = {2, 5, 8} -> CP11 = {1, 3, 4}

Xét vị trí (9, 1) => R9 = {5, 1, 3} C0 = {9, 7, 8} -> P61 = {5, 1, 3, 9, 7, 8}. Vì ô (7, 1) A4 = {6, 7, 8} -> CP11 = {2, 4, 6}

|  |  |
| --- | --- |
|  |  |