204764 – Problem Sets for Search

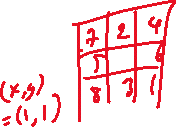
1. **8-Puzzle**

Slide tiles into empty tile from Start to Goal

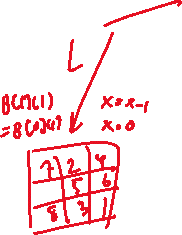


Diagram

Description automatically generated



* **States**: 3x3 matrix B[0][0] … B[2][2]
* **Actions**: Let (x, y) be location of empty tile
  + **Left:** move left adjacent tile into empty tile. **Precondition**: x > 0



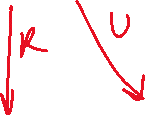
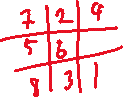
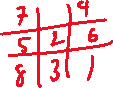
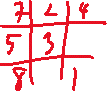
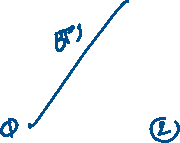
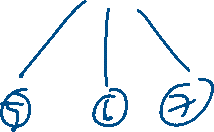
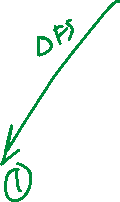
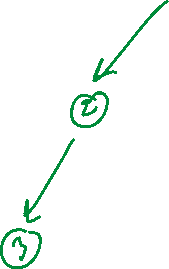
* + **Right:** move right adjacent tile into empty tile. **Precondition**: x < 2
  + **Up:** move top adjacent tile into empty tile. **Precondition**: y > 0



* + **Down:** move lower adjacent tile into empty tile. **Precondition**: y < 2
* Transition Model:
  + **Left:** left tile move to (x, y):
    - **left tile move to (x,y):** B[x][y] = B[x-1][y],
    - **Empty tile move left:** B[x-1][y] = \_, x = x-1
  + **Right:** right tile move to (x, y):
    - B[x][y] = B[x+1][y], B[x+1][y] = \_, x = x+1
  + **Up:** upper tile move to (x, y), y = y – 1
    - B[x][y] = B[x][y-1], B[x][y-1] = \_, y = y-1
  + **Down:** lower tile move to (x, y), y = y + 1
    - B[x][y] = B[x][y+1], B[x][y+1] = \_, y = y+1
* Goal Test: As pictured above
* Step Cost: 1 per action
* Path Cost: Sum of step costs in path

Diagram

Description automatically generated



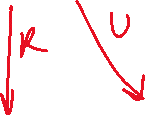
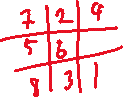
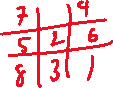
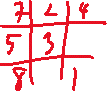
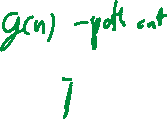
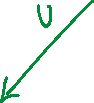
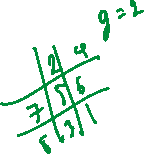
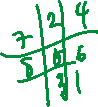
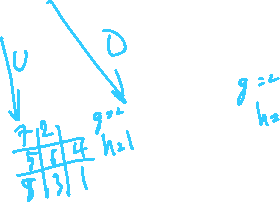
Diagram

Description automatically generated



Diagram

Description automatically generated



1. **3 Jugs**



“You have three jugs, measuring 12 gallons, 8 gallons, and 3 gallons, and a water faucet. You can fill the jugs up or empty them out onto the ground, or pour from one jug (A) to another (B) until (1) the target jug (B) is full, or (2) the source jug is empty (A). You need to measure out exactly one gallon.”



* States: Water level in 3 jugs. Let jugs be *J*12, *J*8, and *J*3. Variables are *W*12, *W*8, and *W*3, water level of each jugs. State is {*W*12, *W*8, *W*3}
* Actions:
  + Fill(Jx): fill jug x from the faucet. **Precondition:** Jx is not full
  + Empty(Jx): empty jug x. **Precondition:** x Jx is not empty
  + Pour(Jx, Jy): pour water from jug Jx, to jug Jy. **Precondition:** Jx is not empty AND Jy is not full
* Transition Model:
  + Fill(Jx): Wx = x, or Jx becomes full
  + Empty(x): Wx = 0, or Jx becomes empty
  + Pour(x, y):
    - If y - Wy > Wx : #Pour until Jx is empty
      * Wy = Wx + Wy
      * Wx = 0
    - Else: # Pour until Jy is full
      * Wy = y
      * Wx = Wx – (y - Wy)
* Goal Test: One of the followings
  + SUM(Wi) == 1
  + *W*12 == 1 OR *W*8 == 1 OR *W*3 == 1
* Step Cost:
  + 1 per action
* Path Cost:
  + Sum of step costs in the path

1. n-Queen

Place n queens onto n n board without any piece being able to attack another

A picture containing crossword puzzle, checker

Description automatically generated

* **Incremental Formulation** – Place 1 queen at a time, 1 per column.
  + **Action**: placing a queen on the leftmost column not yet occupied
* **Complete-state Formulation** – Rearrange *n* queens already on the board, also one per columns
  + **Action**: changing row of a queen

Constraint: cannotAttack(Xi, Xj) when i and j are column number (1..n) and Xi and Xj are row position of the queen on those columns

cannotAttack(Xi, Xj) = (Xi ≠ Xj) AND (|Xi - Xj| ≠ |i - j|)

1. Sudoku

Fill each square with 1..9 such that no two squares that share (1) column, (2) row, or (3) 3x3 group have the same number.

Shape

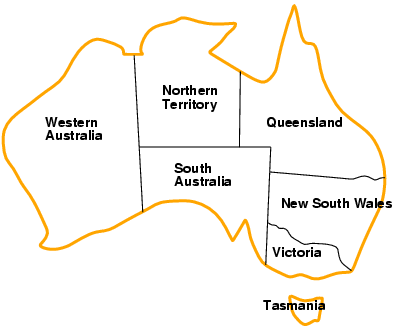
Description automatically generated with medium confidence



* *X* =
* *D* =
* C =

1. **Map-coloring**

Color each state of Australia in the map using {red, green, blue} such that no states that share border have the same color.





* *X* =
* *D* =
* C =

1. **Conference Scheduling**

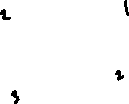
You are scheduling events for a small conference in a day. There are 4 events, P1 - P4. Each event takes either morning session (AM) or afternoon session (PM). There are 2 rooms available that can host any one event at a time. There are some constraint between events, however:

* Event P1 must take place before event P2
* Event P1 and P3 share the same speaker, and cannot take place at the same time

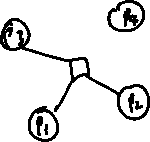
What will be the schedule?

|  |  |  |
| --- | --- | --- |
|  | AM | PM |
| Room 1 |  |  |
| Room 2 |  |  |

* *X* = {P1, P2, P3, P4} – Room & Time for events 1-4 respectively
* *Di* = {R1AM, R1PM, R2AM, R2PM} – Room & Time placement, same for all *X*i
* Functions
  + Time(x) – return time of placement x. For example, Time(R1PM) = PM
  + Room(x) – return room of placement x. For example, Room(R1PM) = R1
  + AM < PM
* C = {



* + Alldiff(P1..4)



* + Time(P1) < Time(P2)

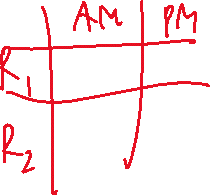
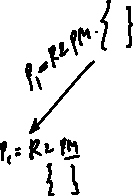
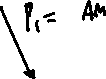
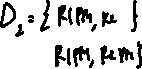
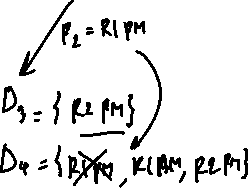
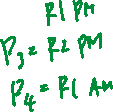


* + Time(P1) ≠ Time(P3)



}

Search



1. **Cryptarithmetic**

A picture containing pool ball

Description automatically generated

Problem is shown on (a). Each letter is a digit. No two letters can have the same digit. Hypergraph (b) shows constraints between variables, where *C*1, *C*2 and *C*3 are carries, and can have value between 0 and 1. Find the correct digit for each letter.

* Variables: *F T U W R O C1 C2 C3*
* Domains:
  + {0,1,2,3,4,5,6,7,8,9} for *F, T, U, W, R,* and *O*
  + {*0, 1*} *for* carry variables *C1, C2,* and *C3*
* Constraints: *Alldiff* (*F,T,U,W,R,O*)
  + O + O = R + 10 · *C1*
  + *C1* + W + W = U + 10 · *C2*
  + *C2* + T + T = O + 10 · *C3*
  + *C3* = F, T ≠ 0, F ≠ 0