

# Problem Solving using Search (Single Agent)

## Pemecahan Masalah dengan Pencarian (Agen Tunggal)

---

### 2.1 Tujuan Instruksional

1. Mahasiswa dapat mendeskripsikan sebuah permasalahan secara formal
2. Mahasiswa dapat merancang ruang pencarian dari sebuah permasalahan
3. Mahasiswa dapat merancang dan mengkodekan algoritma pencarian:
  - a. Greedy search
  - b. BFS
  - c. DFS
  - d. Uniform search
  - e. Iterative deepening search
  - f. Bidirectional search
4. Mahasiswa dapat menganalisis sebuah algoritma pencarian dari sisi:
  - a. Kompleksitas waktu
  - b. Kompleksitas ruang
  - c. Terminasi algoritma
  - d. Optimasi

## 2.2 Ruang pencarian

- Memformulasikan ruang pencarian terhadap definisi permasalahan
- Sebuah state / keadaan didefinisikan dengan nilai dari semua atribut yang spesifik dari sebuah lingkungan
- Sebuah operator berubah dalam setiap keadaan; yang dimulai dari sebuah prakondisi, operator, akibat, dan keadaan yang dipengaruhi
- Initial state / keadaan awal adalah keadaan dimulainya sebuah pencarian
- Goal / tujuan adalah sebuah solusi dari pencarian yang dilakukan

### 2.2.1 Agen yang Memiliki Tujuan

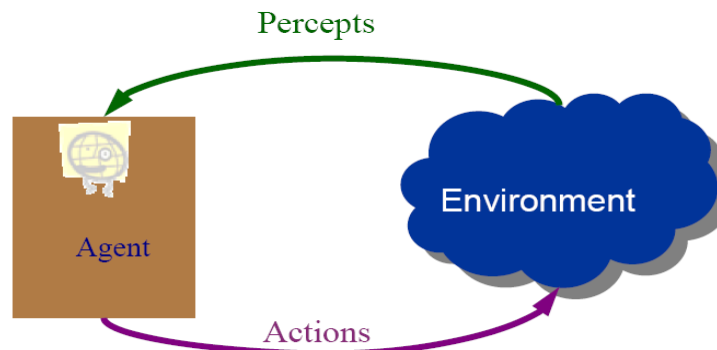


Figure 1

Pada pelajaran sebelumnya, telah disampaikan bagaimana agen cerdas dapat bereaksi dengan lingkungannya, salah satunya adalah agen yang memiliki tujuan (goal directed agent).

Sebuah agen yang memiliki tujuan, memilih aksinya untuk mencapai performa yang diharapkan darinya, yaitu mencapai tujuan yang spesifik. Banyak permasalahan yang dapat direpresentasikan sebagai sebuah kumpulan keadaan dan kumpulan aturan (rules) yang akan menjelaskan bagaimana perubahan dari sebuah keadaan ke keadaan lainnya. Setiap keadaan adalah gambaran abstrak dari lingkungan agen. Abstraksi tersebut merupakan sebuah konfigurasi dari agen. Keadaan awal memberikan deskripsi keadaan agen saat dimulainya pencarian. Sebuah aksi / operator akan membawa agen ke keadaan lainnya. Sebuah keadaan dapat memiliki runtutan keadaan lainnya. Untuk itulah diperlukan perencanaan aksi (planning).

Sebuah tujuan adalah deskripsi dari keadaan dalam lingkungan yang diinginkan untuk dicapai. Keadaan akhir ini seringkali dispesifikan oleh sebuah pengujian tujuan dengan pencapaian keadaan tertentu.

Beberapa contoh tujuan dari sebuah agen yang memiliki tujuan:

1. 8-puzzle: tujuan dari agen pada 8-puzzle adalah untuk mencapai konfigurasi bahwa baris teratas dari puzzle terisi dengan ubin 1, 2, dan 3.
2. Tujuan dari agen yang bernavigasi dalam sebuah maze adalah untuk mencapai posisi HOME.

Agen harus memilih urutan aksi yang tepat untuk mencapai tujuan yang ditugaskan kepadanya.

### 2.2.2 Notasi Ruang Pencarian

Keadaan awal (initial state) adalah konfigurasi keadaan yang dimiliki agen dalam ruang pencarian.

Aksi / Operator adalah sesuatu yang akan membawa agen dari sebuah keadaan ke keadaan lain, yang disebut dengan keadaan berikutnya (successor state).

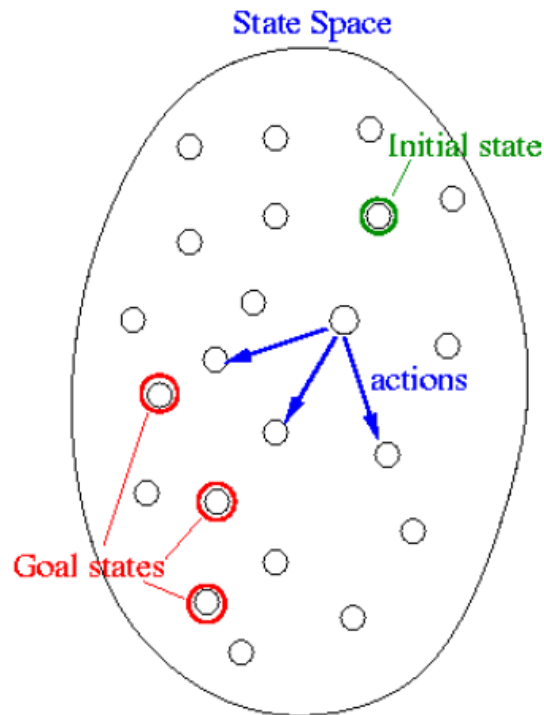
Rencana (plan) adalah sebuah urutan aksi. Biaya dalam sebuah perencanaan disebut dengan biaya jalur (path cost). Biaya selalu positif, dan sebuah total biaya adalah penjumlahan terhadap biaya yang diperlukan dalam sebuah jalur pencarian.

Formulasi permasalahan berarti memilih himpunan keadaan yang dapat dipertimbangkan untuk dilalui, dan himpunan aksi untuk bergerak dari suatu keadaan ke keadaan lainnya. Pencarian merupakan sebuah proses untuk mempertimbangkan berbagai urutan yang dimungkinkan dalam himpunan aksi yang ditentukan oleh keadaan awal, sehingga pada akhirnya mencapai keadaan tujuan.

### 2.2.3 Formulasi Permasalahan

Dari uraian pada paragraf sebelumnya, dapat disimpulkan bahwa sebuah permasalahan adalah abstraksi dari:

- S: himpunan keadaan
- S0: keadaan awal
- A:  $S \rightarrow S$  adalah himpunan aksi
- G: keadaan akhir, perhatikan bahwa G adalah himpunan bagian dari S



**Figure 2**

Dengan demikian yang harus ditemukan dalam sebuah pencarian adalah urutan aksi yang akan membawa agen dari keadaan awal untuk mencapai tujuan. Urutan ini disebut dengan perencanaan (plan).  $P = \{a_0, a_1, \dots, a_N\}$  akan menjalani keadaan-keadaan dalam sebuah permasalahan  $\{s_0, s_1, \dots, s_{N+1} \in G\}$ . Sebuah urutan keadaan disebut sebagai jalur solusi. Biaya yang terjadi dalam sebuah jalur selalu positif, dan dalam banyak kasus dihitung dari jumlah total biaya yang terjadi dalam setiap aksi.

#### Representasi Formulasi Permasalahan

Setiap permasalahan akan dapat direpresentasikan dengan graf terarah (directed graph).

- Keadaan akan digambarkan dalam sebuah simpul (node)
- Aksi yang diijinkan digambarkan sebagai sebuah arah (arc)

#### Proses Pencarian

Algoritma umum dalam melakukan pencarian dapat dijelaskan sebagai berikut:

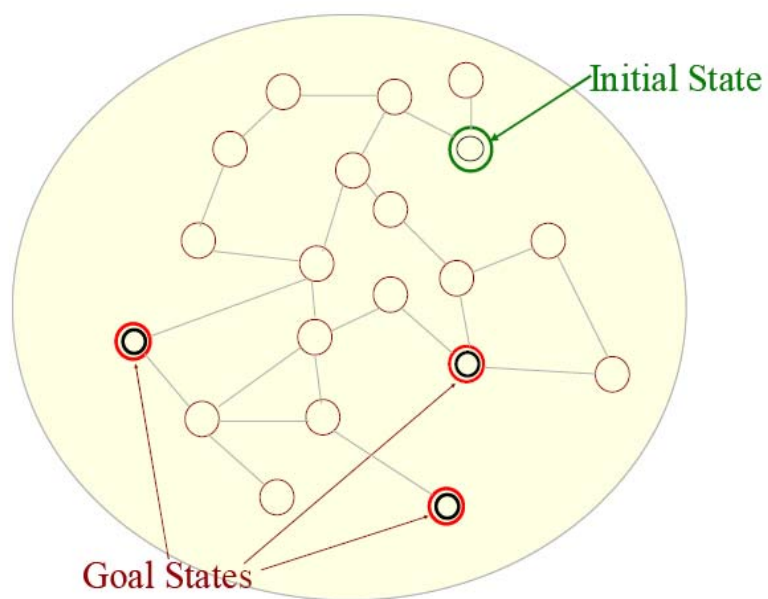
Lakukan sampai sebuah solusi ditemukan atau ruang pencarian sudah dilalui semua

1. Lihat keadaan saat ini
2. Eksekusikan aksi yang diijinkan untuk mencapai keadaan berikutnya
3. Pilih salah satu keadaan
4. Lihat apakah keadaan baru merupakan tujuan,  
Jika tidak, keadaan baru merupakan keadaan saat ini, dan ulangi proses

## 2.3 Contoh Permasalahan

### 2.3.1 Ilustrasi sebuah Proses Pencarian

Lihat pada ilustrasi berikut:

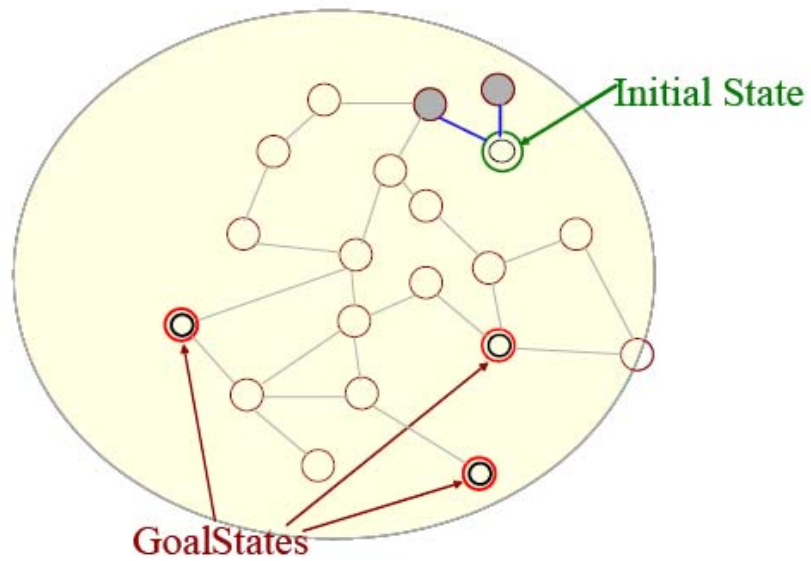


**Figure 3**

S0 adalah keadaan awal.

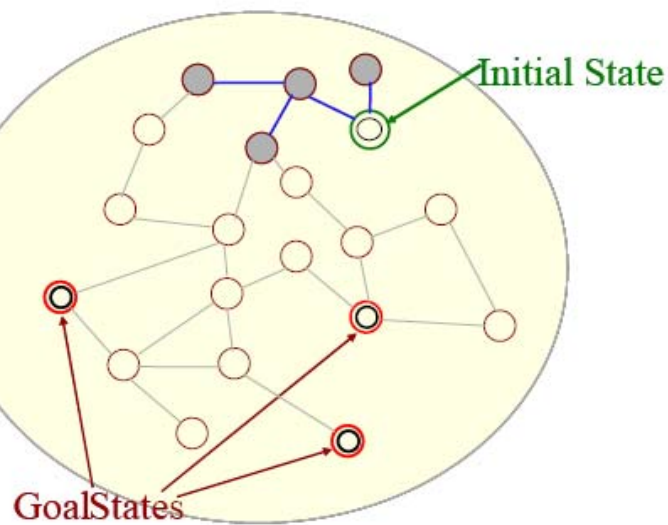
Suksesori dari sebuah keadaan adalah keadaan-keadaan disekitarnya.

Terdapat tiga jenis solusi dalam himpunan keadaan.



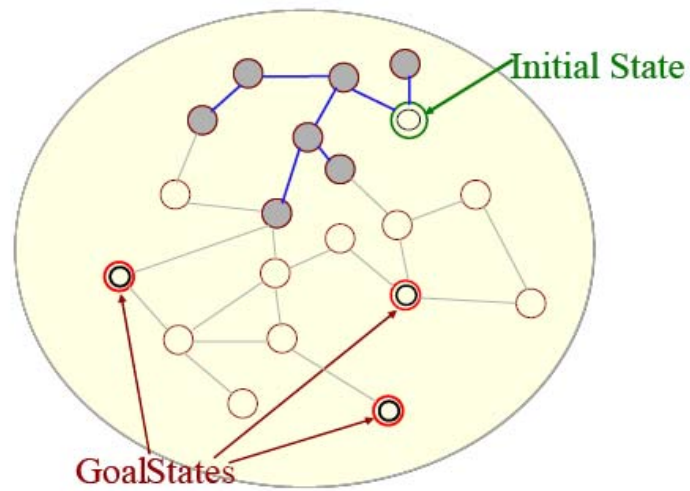
**Figure 4**

Dari gambar tampak bahwa dua keadaan dapat dipilih dari keadaan awal.



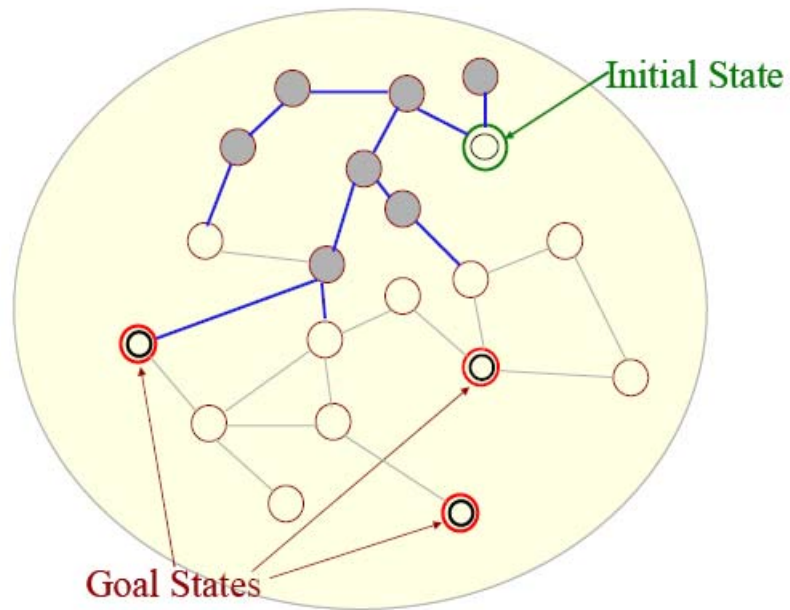
**Figure 5**

Sebuah keadaan dipilih dan keadaan berikutnya dapat dipilih kembali.



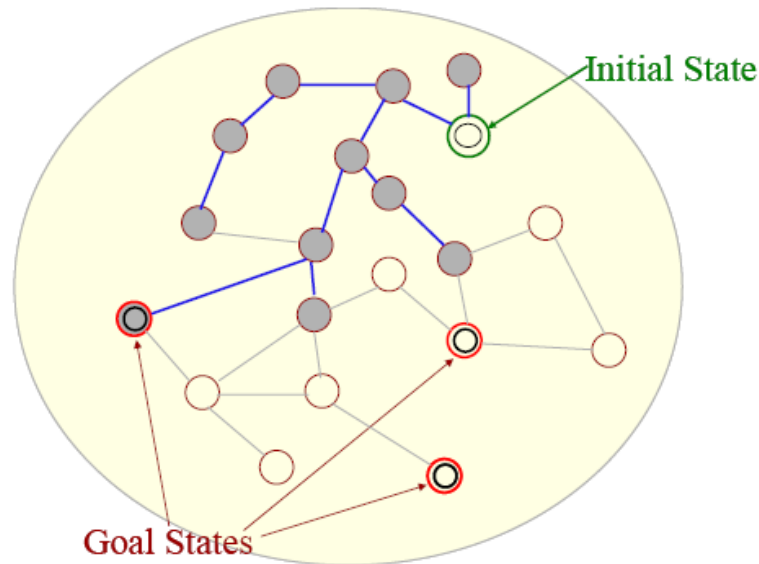
**Figure 6**

Langkah berikutnya dari keadaan yang dipilih kembali dibuka, sebagai kemungkinan pilihan keadaan yang diijinkan.



**Figure 7**

Pilihan keadaan berikutnya kembali dibuka.



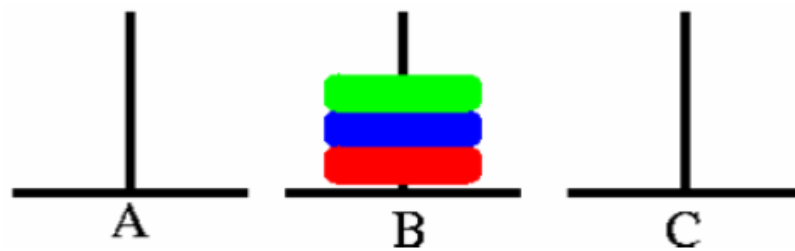
**Figure 8**

Sampai pada akhirnya sebuah tujuan ditemukan.

Contoh di atas, mengilustrasikan bagaimana dari sebuah keadaan awal, melalui pemilihan keadaan pada akhirnya dapat membawa ke keadaan akhir (tujuan). Simpul yang berwarna abu-abu adalah pohon pencarian yang terbentuk. Biasanya dalam sebuah pohon pencarian, sebuah simpul dibuka sekali dalam setiap waktu. Urutan yang menggambarkan urutan pembukaan simpul dalam pohon pencarian disebut sebagai strategi pencarian.

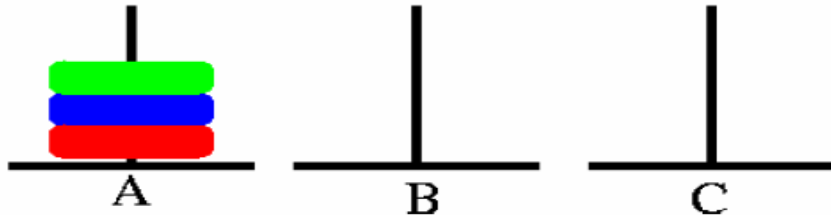
### 2.3.2 Contoh Permasalahan: Tower of Hanoi

Diberikan 3 buah cakram dan 3 tiang. Aksi yang diijinkan adalah memindahkan cakram paling atas ke tiang lainnya dimanapun. Tujuan yang diinginkan adalah bahwa semua cakram berada di tiang B.



**Figure 9**

Keadaan awal adalah sebagai berikut:



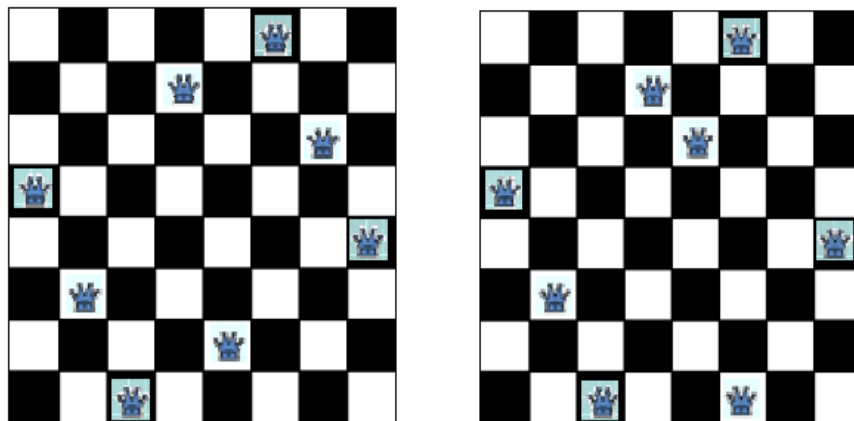
**Figure 10**

Urutan aksi yang dapat dilakukan untuk mencapai tujuan adalah:

1. Pindahkan A  $\rightarrow$  C
2. Pindahkan A  $\rightarrow$  B
3. Pindahkan A  $\rightarrow$  C
4. Pindahkan B  $\rightarrow$  A
5. Pindahkan C  $\rightarrow$  B
6. Pindahkan A  $\rightarrow$  B
7. Pindahkan C  $\rightarrow$  B

### 2.3.3 Contoh problem: 8-queens problem

Persoalannya adalah bagaimana menempatkan 8 ratu dalam papan catur, sehingga tidak ada satu ratupun berada dalam satu baris lurus atau diagonal.



**Figure 18**

Gambar di sebelah kiri merupakan contoh sebuah solusi, sedangkan gambar di sebelah kanan bukan solusi karena masih ada ratu yang dapat saling menyerang.

Bagaimana cara yang tepat untuk memformulasikan permasalahan ini sebagai sebuah ruang pencarian? Sebuah formulasi permasalahan melibatkan keputusan untuk merepresentasikan keadaan, memilih keadaan awal, mendeskripsikan aksi dan urutan langkah yang dapat dipilih.

#### Formulasi Permasalahan N-queens: Alternatif 1

- States: susunan 0 sampai 8 ratu pada papan catur.
- Keadaan awal: 0 ratu di atas papan.
- Suksesor: tambahkan sebuah ratu pada kotak manapun.
- Tes tujuan: ada 8 ratu di atas papan, namun tidak ada yang saling menyerang.

Keadaan awal memiliki 64 kemungkinan suksesor. Setiap keadaan pada level berikutnya memiliki 63 suksesor, dst. Pohon pencarian dapat dibatasi dengan hanya mengambil suksesor yang memungkinkan posisi ratu tanpa dapat saling menyerang. Untuk melakukannya, harus dilihat apakah posisi ratu yang baru bertentangan dengan yang sudah ada di atas papan. Solusi akan diperoleh dalam 8 tingkat kedalaman pencarian.

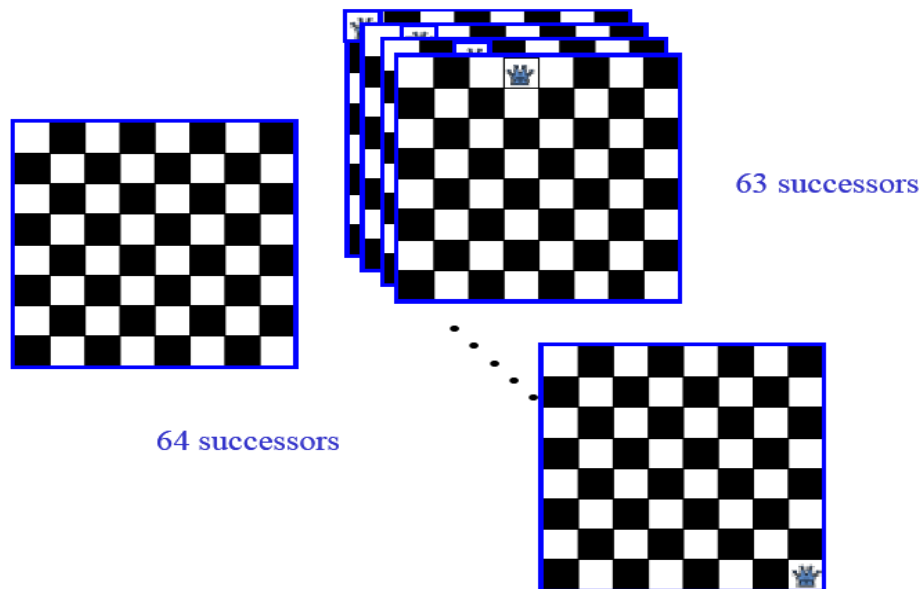


Figure 19

#### Formulasi Permasalahan N-queens: Alternatif 2

- States: Susunan apapun untuk 8 ratu di atas papan catur.
- Keadaan awal: Semua ratu berada pada kolom pertama.
- Suksesor: Ubah posisi salah satu ratu.
- Tes tujuan: ada 8 ratu di atas papan, namun tidak ada yang saling menyerang.

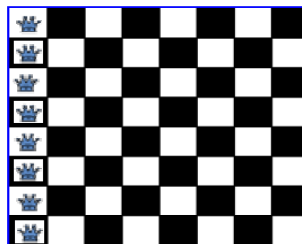


Figure 20

Jika dipindahkan salah satu ratu dari kolom pertama, maka akan ada 7 kemungkinan kolom untuk penempatannya.

Formulasi Permasalahan N-queens: Alternatif 3

- States: Susunan apapun untuk k-queens pada k-baris pertama sehingga tidak ada yang saling menyerang.
- Keadaan Awal: 0 ratu di atas papan catur.
- Suksesor: tambahkan sebuah ratu pada baris (k+1) sehingga tidak ada yang saling menyerang.
- Tes tujuan: ada 8 ratu di atas papan, namun tidak ada yang saling menyerang.

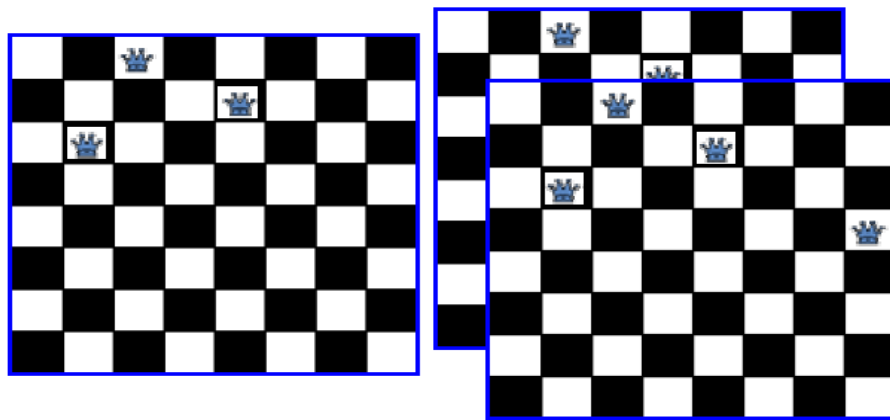


Figure 21

2.3.4 Definisi Permasalahan: 8-puzzle

5	4	
6	1	8
7	3	2

Initial State

1	4	7
2	5	8
3	6	

Goal State

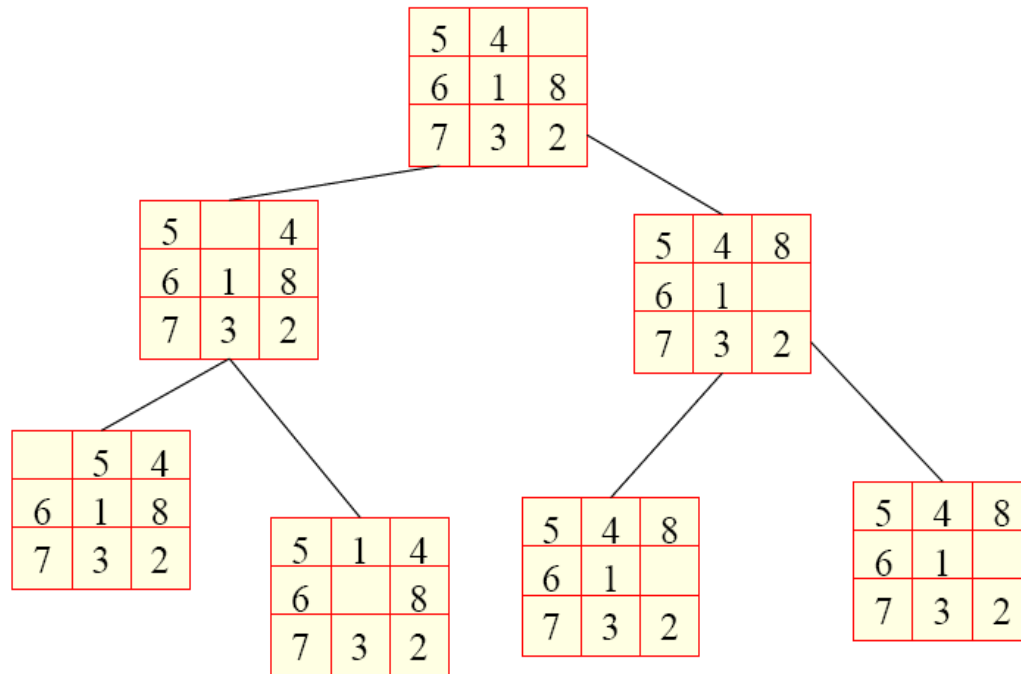
Figure 22

Dalam 8-puzzle, terdapat sebuah papan berukuran 3x3 kotak, dan 8 ubin bernomor. Pada papan terdapat sebuah posisi kosong. Setiap ubin dapat digeser ke tempat kosong. Dari situasi ini, dapat dianggap bahwa pergerakan yang terjadi adalah perpindahan ubin kosong ke atas, bawah, kiri atau kanan. Tujuan dari puzzle ini adalah melakukan pergerakan dari keadaan awal untuk mencapai konfigurasi nomor ubin tertentu. Bentuk lain dari 8-puzzle adalah 15-puzzle, jika terdapat 4x4 kotak engan 15 ubin bernomor.

Representasi ruang pencarian dapat disimpulkan sebagai berikut:

- States: deskripsi dari setiap ubin pada lokasinya.
- Aksi: ubin kosong bergerak ke atas, bawah, kiri atau kanan.
- Tes tujuan: keadaan sekarang sesuai dengan urutan tertentu.
- Biaya: setiap pergerakan bernilai 1.

Bagian dari ruang pencarian dapat dilihat pada gambar berikut, perhatikan bahwa semua state yang dapat dibentuk, hanya perlu dilakukan pada saat proses pencarian telah dimulai.

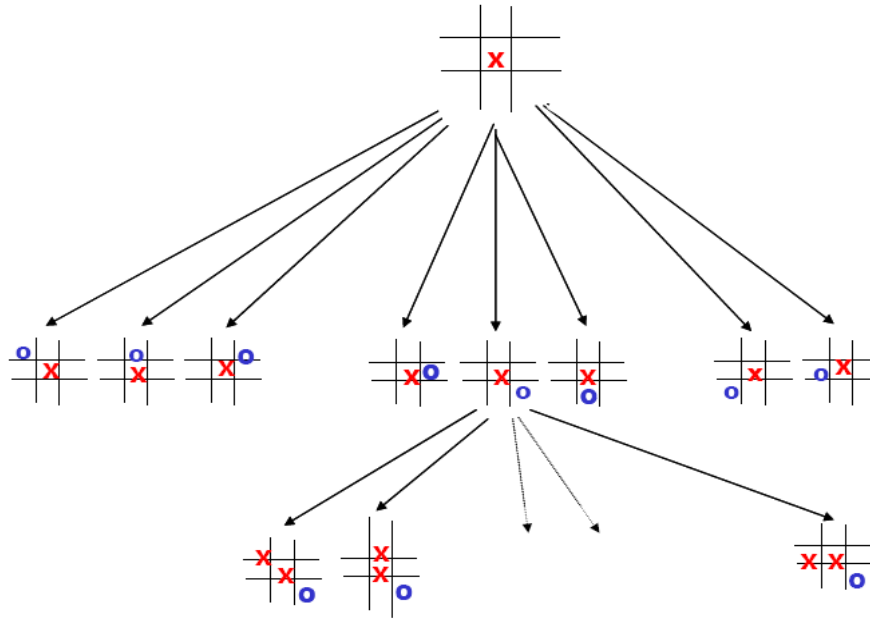


**Figure 23**

### 2.3.5 Definisi Permasalahan: tic-tac-toe

Permainan ini dilakukan oleh dua orang secara bergantian. Seorang pemain menggunakan lambang X, dan lainnya O. Pada awal permainan X menempatkan lambangnya pada papan kosong, O dapat menempatkan lambangnya pada posisi yang tidak ditempati X. Pemenangnya adalah yang dapat menempatkan lambangnya pada 3 posisi berurutan mendatar, menurun ataupun diagonal.

Bagian dari permainan dapat dilihat pada gambar berikut:



**Figure 24**

### 2.3.6 Permasalahan Teko Air

Ada tiga teko dengan ukuran 12, 8, 3 liter, dan sebuah mata air. Salah satu teko tersebut harus tepat terisi 1 liter air. Definisi permasalahan yang dapat diambil adalah:

- Keadaan awal: semua teko kosong.
- Tes tujuan: salah satu teko berisi tepat 1 liter.
- Suksesor:
  - Aksi: memindahkan air ke teko  $i$  dan  $j$ , dengan kapasitas  $C_i$  dan  $C_j$ , dan air yang telah terdapat dalam teko  $L_i$  dan  $L_j$ . Teko  $i$  terisi dengan  $\max(0, L_i - (C_j - G_j))$  liter air, dan teko  $j$  dengan  $\min(C_j, L_i + L_j)$
  - Aplikasi aksi dengan mengisi ke teko  $i$  sejumlah  $C_i$  liter air.
- Biaya: berikan biaya pada setiap kali seliter air dipindahkan dari satu teko ke teko lain.

#### Ruang Pencarian Eksplisit dan Implisit

Ruang pencarian dapat secara eksplisit direpresentasikan dalam sebuah graf. Namun seringkali ruang pencarian direpresentasikan secara implisit, dan dibentuk pada saat diperlukan saja. Untuk membentuk ruang pencarian secara implisit, agen harus mengetahui:

- Keadaan awal
- Operator dan akibat dari setiap langkah dalam aksi

Operator adalah sebuah fungsi yang digunakan untuk membuka sebuah simpul dan menghitung suksesornya. Dari alternatif definisi permasalahan pada N-queens, dapat dilihat bahwa operator ini dapat memiliki akibat yang luas. Sebenarnya tidak perlu semua keadaan yang mungkin dibuka untuk

diperlihatkan, karena akan memperluas ruang pencarian, yang jika dikaitkan dengan memori komputer akan membutuhkan ruang yang besar pula.

Pertanyaan:

1. Berikan keadaan awal, tes tujuan, suksesor, dan biaya untuk setiap problem di bawah ini. Berikan formulasi problem yang menurut Anda paling tepat untuk diimplementasikan:

- a. Anda diminta untuk memberi warna pada peta dengan maksimum 4 warna, sedemikian rupa sehingga tidak ada dua wilayah bertetangga yang memiliki warna sama.
- b. Pada TSP (travelling salesman problem), terdapat  $N$  kota yang dihubungkan dengan jalan. Yang diminta adalah untuk menemukan jalur terpendek antara  $N$ -kota yang dilalui dari sebuah kota asal, tanpa harus melalui kota yang sama dua kali, untuk dapat kembali ke kota asal.
- c. Missionaries & Cannibals problem: ada 3 misionaris dan 3 kanibal pada sebuah sisi sungai. Terdapat sebuah perahu yang dapat membawa 2 penumpang bersamaan. Seorang misionaris tidak boleh berjumlah lebih sedikit dari para kanibal dalam sebuah keadaan. Berikan perencanaan untuk menyeberangi sungai tersebut.

2. Diberikan dua teko berkapasitas 5 dan 2 liter. Tujuannya adalah untuk mengisi teko yang berkapasitas 2 liter, tepat dengan 1 liter air. Anda dapat menggunakan formulasi permasalahan berikut ini:

State:  $(x,y)$ , dimana  $x$  adalah jumlah literan air pada teko 5 lt, dan  $y$  pada teko 2 lt.

Keadaan awal:  $(5,0)$

Tujuan:  $(*, 1)$ , dimana  $*$  berarti jumlah sembarang.

Kembangkan sebuah pohon pencarian, sehingga tujuan tercapai.