# 人工知能入門

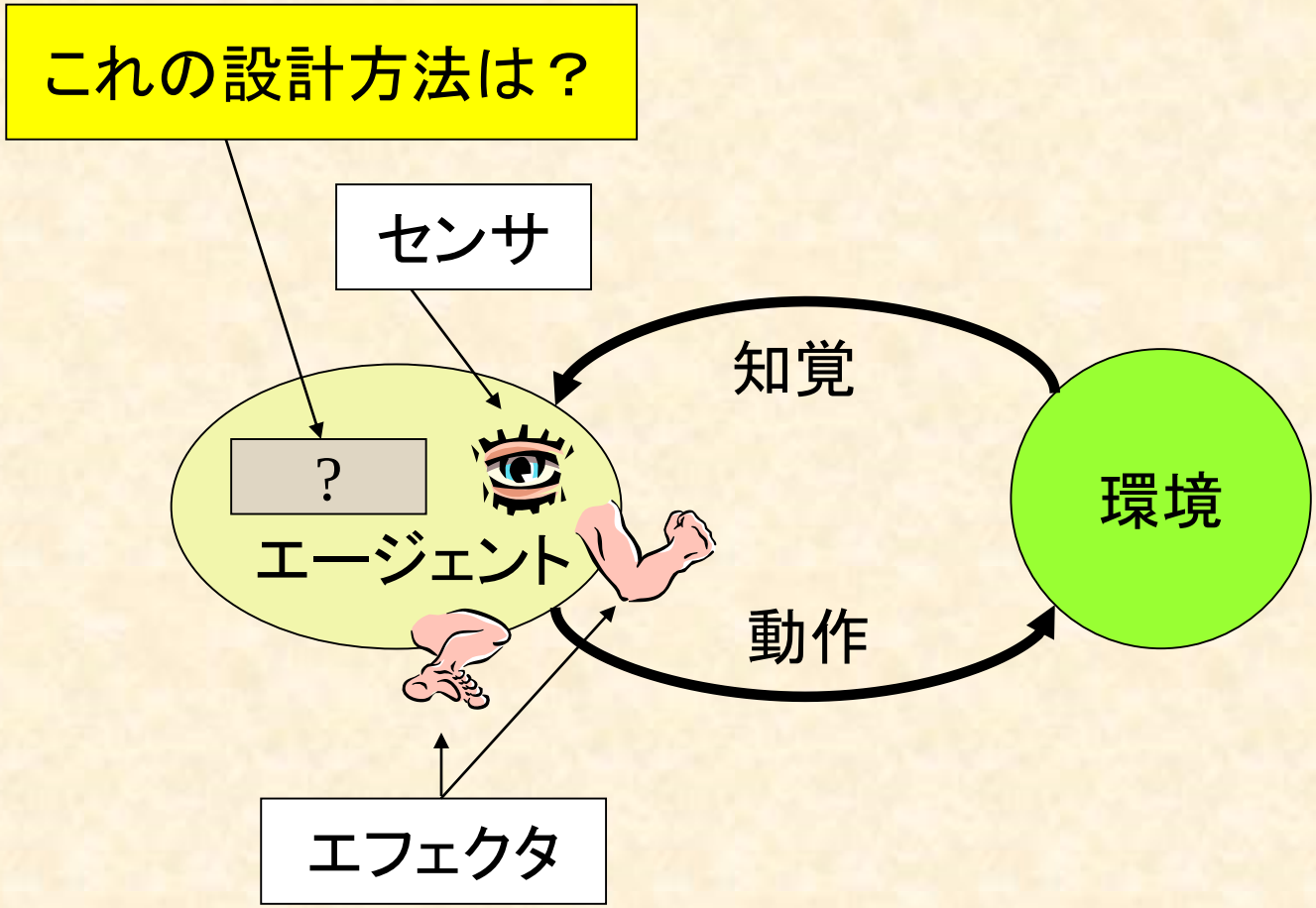
## -探索による人工知能-

Lecture 3

## 問題定式化：最適解と応用例

http://www2.teu.ac.jp/gamelab/LECTURES/ArtificialIntelligence/index.html

# 合理的エージェント



# 問題を定式化すること
## 明確に定義された問題と解

- 問題
  - 何を行うかを決めるためにエージェントが使う情報の集まり
- 問題の定式化
  - エージェントは自分自身がそこにいることを知っている初期状態
  - オペレータ：ある特定の状態において行為を実行することによってどの状態に到達するか
    - 問題の状態空間：初期状態から行為の任意の系列によって到達可能なすべての状態の集合
    - 状態空間の経路：単に一つの状態を他の状態に導く行為列
  - ゴール検査：単一状態に対してそれがゴール状態であるかを決定できるもの
  - 経路コスト関数：経路にコストを割り当てる関数（各々の行為のコストの総和）

# 問題を定式化すること

## 明確に定義された問題と解

**datatype** Problem
**components** Initial-State, Operators, Goal-Test, Path-Cost-Function

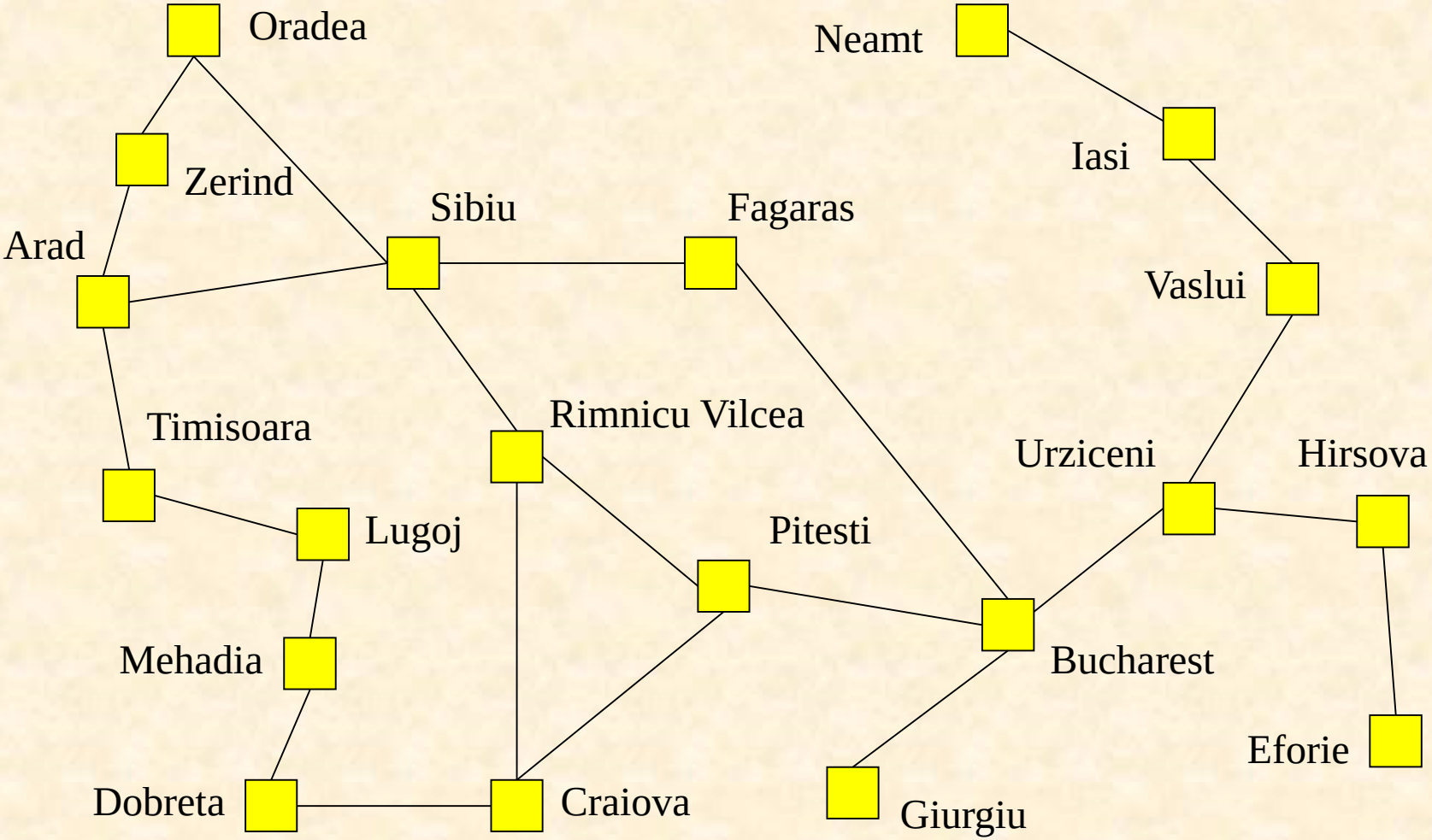
- 探索アルゴリズム
  - 入力はこのデータタイプのインスタンス
  - 出力は解：初期状態からゴール検査を満足する状態への経路
- 多重状態問題
  - 問題は初期状態集合を持つ
  - 各行為に対して、与えられた状態から到達する状態の集合を規定するオペレータの集合
  - ゴール検査と経路コスト関数は以前と同じ
  - 経路は状態の集合を結びつける
  - 解はすべての状態がゴール状態であるような状態集合に至る経路
  - 状態空間は状態集合空間に替わる

# 問題を定式化すること
## 問題解決の性能の測定

- 探索の有効性を測定する
  - １番目：そもそも解を見つけるか？
  - ２番目：それは良い解（少ない経路コストをもつ解）であるか？
  - ３番目：解を見つけるために要した時間とメモリに関する探索コストはどれほどか？
- ルーマニアの例の場合
  - 経路コストは町と町の間の総キロ
  - 解を見つかるために長く熟慮するのは望ましくない
  - 総コストを計算するためにはキロとミリ秒を合算しなければならない
- 資源の配分
  - どの資源をどの程度探索にに充て、どの程度実行に充てるか
  - トレードオフ：最適解を得るために非常に長い時間探索するか短い間しか探索して、少し多い経路コストの解を得るか

# 問題を定式化すること

## ルーマニアの地図



# 問題を定式化すること
## 状態と行為の選択

- 問題の定式化
  - 適切な状態空間は20状態があり、各状態は場所によって定義され、都市として規定される
  - 初期状態:「Aradにいる」
  - ゴール検査:「ここはBucharestか？」
  - オペレータ:都市間を道に沿ってドライブする
- 最適な解は？
- 問題解決の本当の技芸:何を状態とオペレータの記述に取り入れるか？
- 抽象化
  - 無関係な詳細を取り去る(例:旅行に使われる乗り物)
  - 無関係な行為を取り去る(例:ラジオをつける)
  - 取り去っても解を得ることができるなら抽象化は妥当

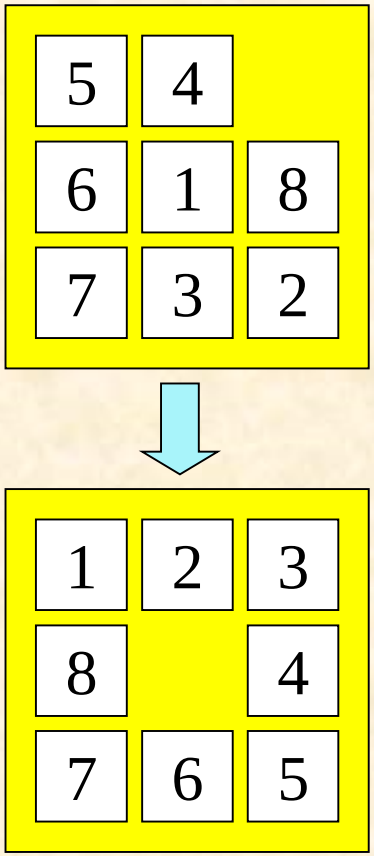
# 例題

- おもちゃの問題
  - 問題解決方法を説明、演習する問題
  - 簡潔で正確な記述を与えることができる
  - アルゴリズムの性能を比較することは可能
  - 例: 8パズル、8クイーン問題、覆面算、掃除機の世界、宣教師と人食い人種
- 現実世界問題
  - おもちゃの問題より難しい
  - 解に興味を持つ人が多い
  - 唯一の合意された記述を持たない
  - 例: ルート発見、旅行と巡回セールスマン問題、VLSIレイアウト、ロボットのナビゲーション、アセンブリの順序付け

# おもちゃの問題

## 8パズル

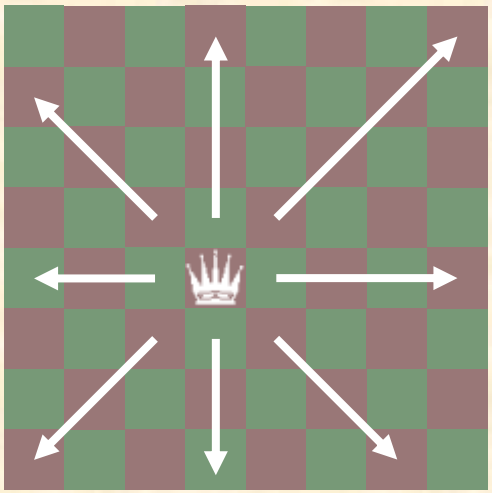
- 八つの番号付けられたタイルと空白のスペースを持つ3×3のボードから構成
- 問題の定式化
  - 状態：八つのタイル（と空白スペース）の各々が9個の区画のどの位置にあるかを規定する
  - オペレータ: 空白を左に動かす、右に動かす、上に動かす、下に動かす
  - ゴール検査: 状態は右下に示されたゴール配置に一致しているか
  - 経路コスト：各々のステップはコスト1を要し、経路コストは経路の長さになる
- 8パズルと15パズルはAIにおける探索アルゴリズムの標準的なテスト問題

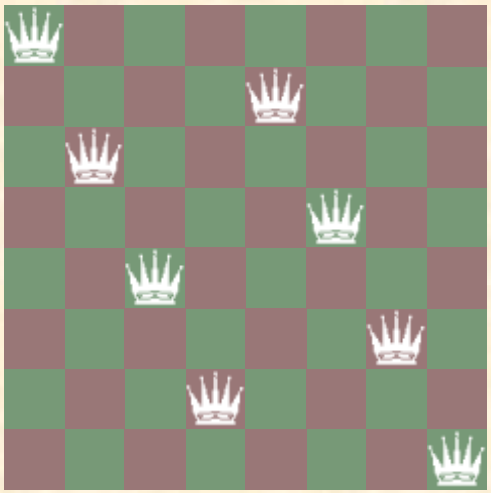


# おもちゃの問題

## 8クイーン問題

- 8クイーン問題の目標：どのクイーンも他のクイーンをアタックしないように、八つのクイーンをチェス盤に置くことである
  - 右下は失敗例





- 演習問題3：試してみて下さい（方法も明確すること！）

# おもちゃの問題

## 8クイーン問題

- 問題の定式化
  - 漸増的定式化：一つずつクイーンを置いていく
  - 完全状態定式化：盤上の八つのすべてのクイーンから出発し、それらを動かす
  - ゴール検査：盤上の八つのクイーンが置かれ、どれもアタックしないか
  - 経路コスト：0
- 一番単純な状態とオペレータ
  - 状態: 0から8個のクイーンの盤上への任意の配置
  - オペレータ: マスにクイーンを加える
  - 問題: $64^8$個の調べなければならない可能な系列がある

# おもちゃの問題

## 8クイーン問題

- より賢明な状態とオペレータ
  - 状態: 0から8個のどれもアタックされていないクイーンの配置
  - オペレータ: 左端の空いている列で他のどのクイーンからもアタックされてないところにクイーンを置く
  - 正しい定式化によって、探索空間のサイズに大きな相違が生ずる
    - 2057通りだけの可能な調べる系列がある
- 完全状態定式化
  - 状態: 八つのクイーンの配置で、各列に1個置く
  - オペレータ: アタックされたクイーンを同じ列の別のマスに動かす
  - 改善方法:可能ならば移動はアタックされていないマスへ

# まとめ

- 問題定式化
  - 初期状態
  - オペレータ
  - ゴール検査
  - 経路コスト関数
- 探索方法の評価
  - 解が見つけるか、良い解が見つけるか、探索コストはどうなっているか
- おもちゃ問題
  - 8パズル
  - 8クイーン問題