Introduction To AI
(SENG 463 - Game Programming)

Dr.Çağatay ÜNDEĞER

Research and Innovation Director

SimBT Inc.

e-mail :

cagatay.undeger@simbt.com.tr

cagatay@undeger.com

Outline

- Description of AI
- Behavior and Agent Types
- Properties of Environments
- Problem Solving
- Well-Known Search Strategies

- Introduction to AI Architectures
- Entities, Attributes and Relations
- Tasks, Actions, States and Events

Artificial Intelligence in Games

- Artificial intelligence (AI) in games deals with creating synthetic characters (agents, animats, bots) with realistic behaviors.
- These are autonomous creatures with an artificial body situated in a virtual world.
- Job of AI developers
  - Give them unique skills and capabilities so that they can interact with their environment intelligently and realistically.

* Animats stand for artificial animals, robots or virtual simulations.

* Bots stand for a computer program that runs automatically or robots.

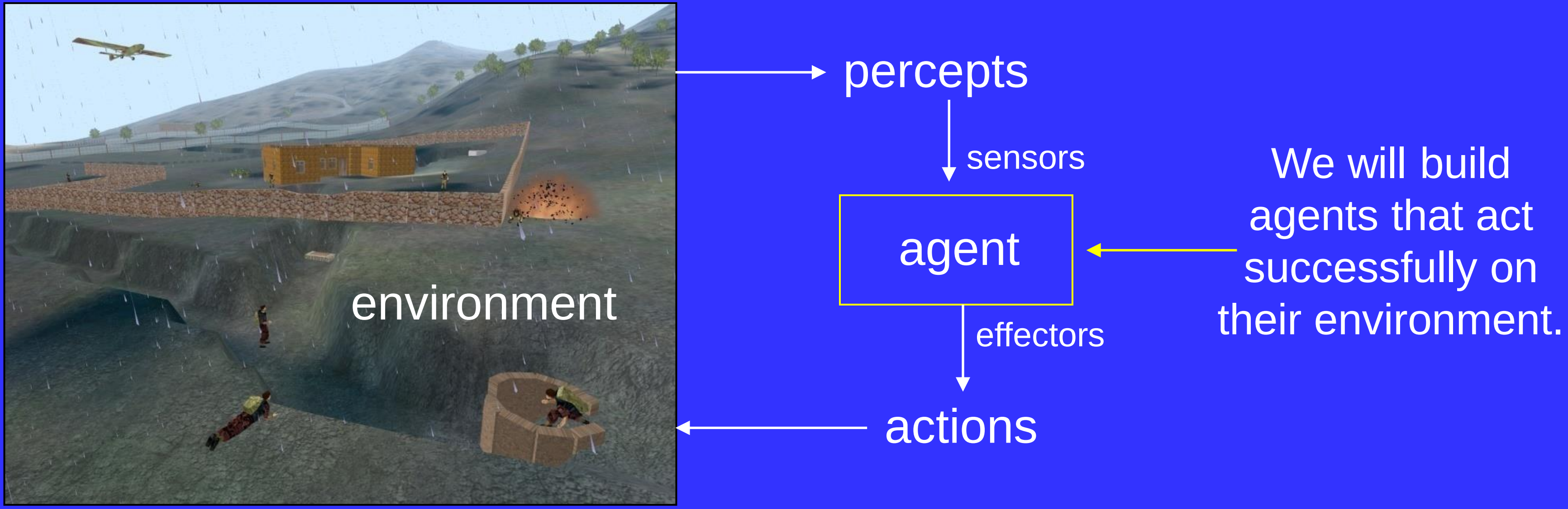
Description of AI

- Definitions are divided into 2 broad categories:
  - Success in terms of human performance,
  - Ideal concept of intelligence, which is called Rationality.

- A system is said to be rational:
  - If it does the right thing,
  - If it is capable of planning and executing the right task at the right time.

Intelligent Agents

- Anything that can be viewed as:
  - Perceiving its environment through sensors and acting on that environment through effectors.



environment

percepts

sensors

agent

effectors

actions

We will build agents that act successfully on their environment.

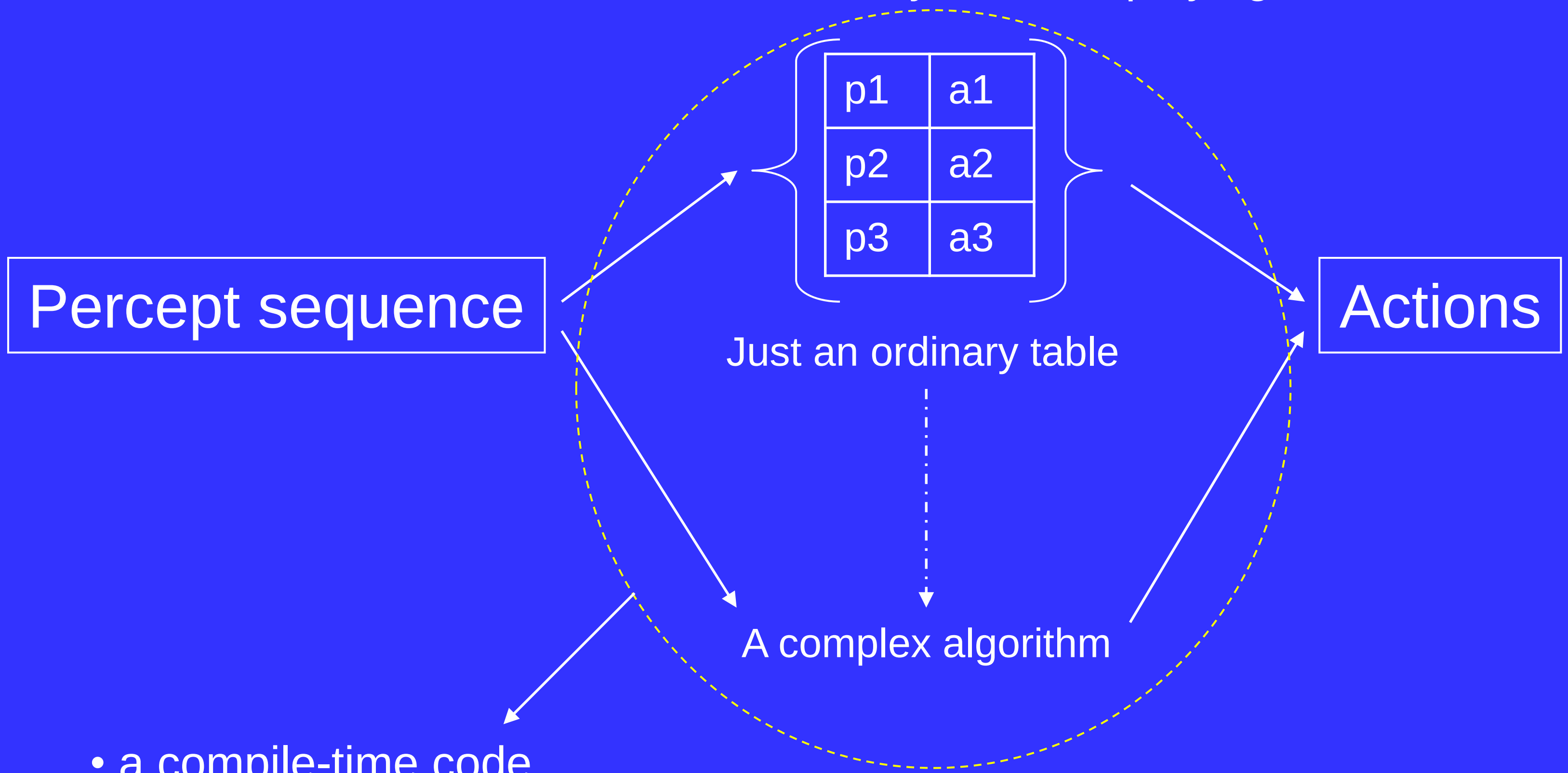
Agent Program

- In each step (cycle / iteration / frame):
  - Gets the percepts from the environment,
  - [Builds a percept sequence in memory],
  - Maps this percept sequence to actions.

- How we do the mapping from percepts to actions determines the success of our agent program.

Percepts to Actions

For instance, a 35^{100} entry table for playing chess



Percept sequence

p1	a1
p2	a2
p3	a3

Just an ordinary table

A complex algorithm

Actions

- a compile-time code
- a run-time script with fixed commands
- a run-time script with extendable commands that call compile-time code

Behavior Types

- Reactive behaviors:
  - A simple decision is taken immediately depending on the developing situation (lying down when someone is shooting at you,...).
- Deliberative behaviors:
  - A complex decision is taken in a longer time by reasoning deeply (evaluation, planning, conflict resolution,...).
- Learning behaviors:
  - Learning how to act by observation, try and error.

Agent Types

- Reactive agents:
  - Simple reflex agents
  - Reflex agents with internal state

- Deliberative agents:
  - Goal–based agents
  - Utility-based agents

Simple Reflect Agents

- In each step:
  - Gets a set of percepts (current state)

    Learn “what the world I sense now is like”
  - Performs rule matching on current state
  - Takes decision without memorizing the past

    Answer “what action I should do now” by using condition-action rules
  - Performs the selected action/s

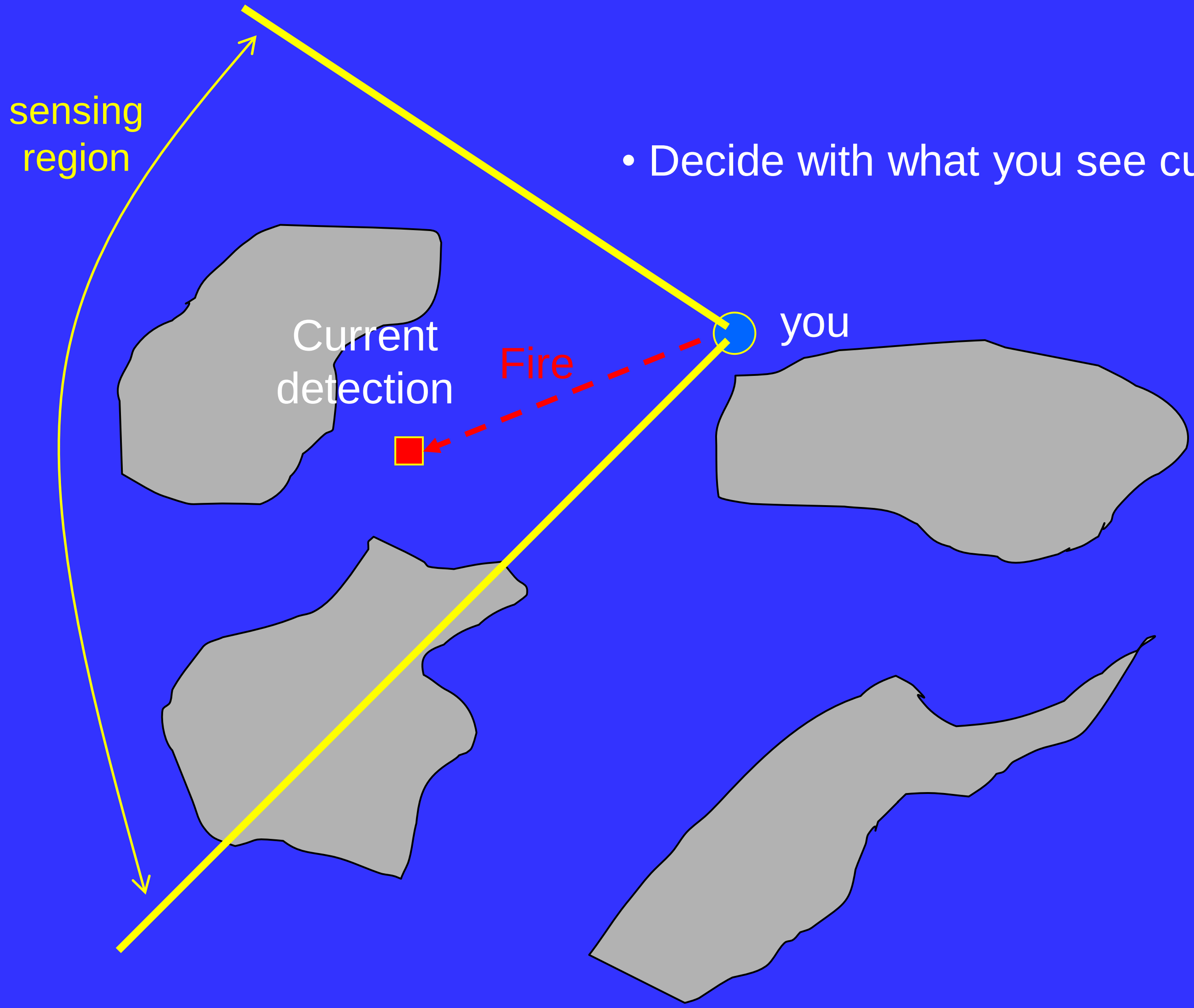
    Change the environment

Simple Reflect Agents

- Rules of a traffic simulation may look like:
  - If distance between me and car in front of me is large then
    - Accelerate
  - If distance between me and car in front of me is normal then
    - Keep state
  - If distance between me and car in front of me is small then
    - Decelerate
  - If distance between me and car in front of me is large and car in front of me breaks then
    - Keep state
  - If distance between me and car in front of me is normal and car in front of me breaks then
    - Break
  - .....

Simple Reflect Agents

- Decide with what you see currently

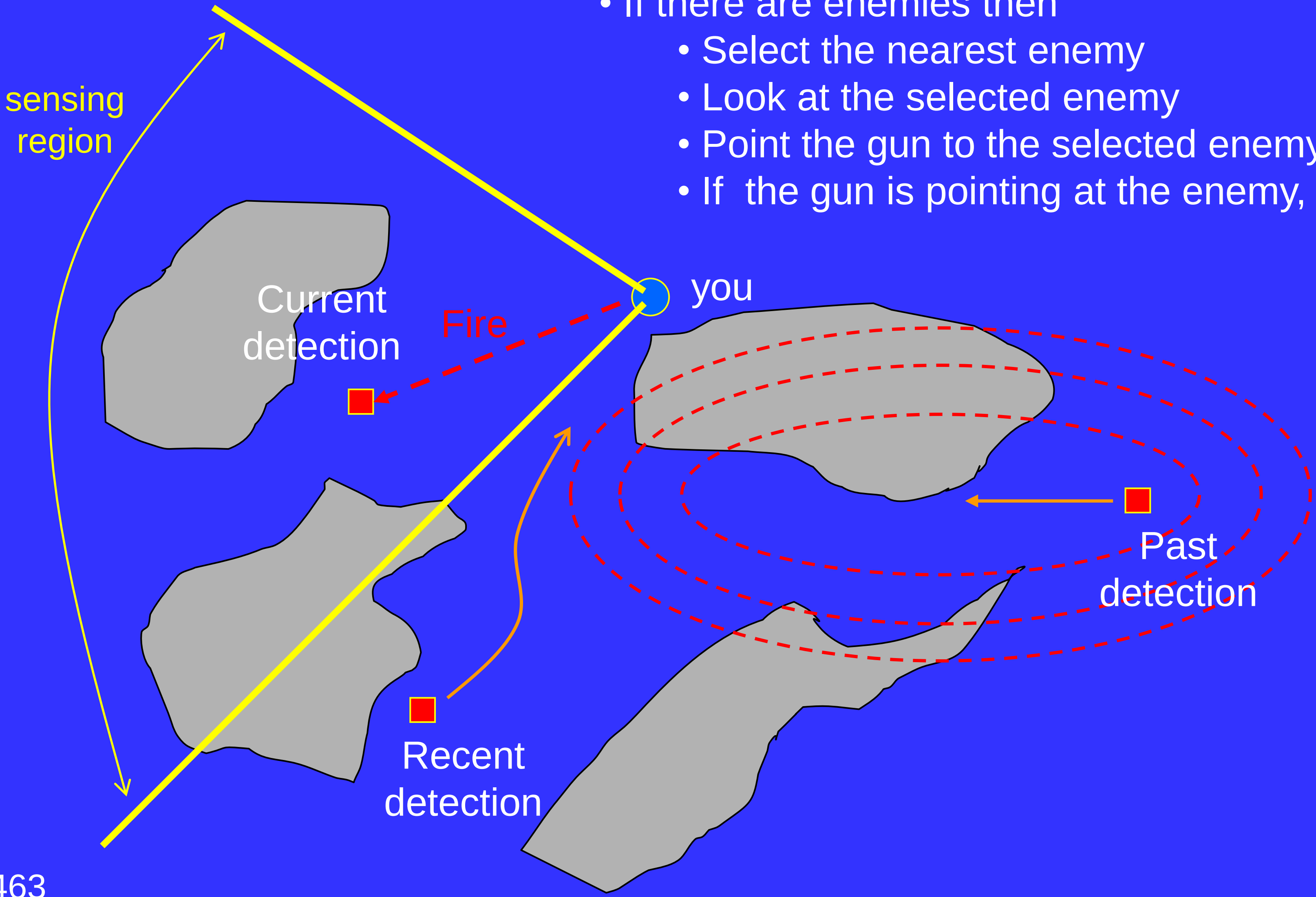


Reflect Agents with Int. State

- In each step:
  - Gets a set of percepts (current state)
    - Learn "what the world I sense now is like"
  - Integrates past state to current state to get an enriched current state (internal state)
    - Keep track of the world and estimate "what the world is like now"
  - Performs rule matching on current state
  - Takes decision
    - Answer "what action I should do now" by using condition-action rules
  - Performs the selected action/s
    - Change the environment

Reflect Agents with Int. State

- If there are enemies then
  - Select the nearest enemy
  - Look at the selected enemy
  - Point the gun to the selected enemy
  - If the gun is pointing at the enemy, fire

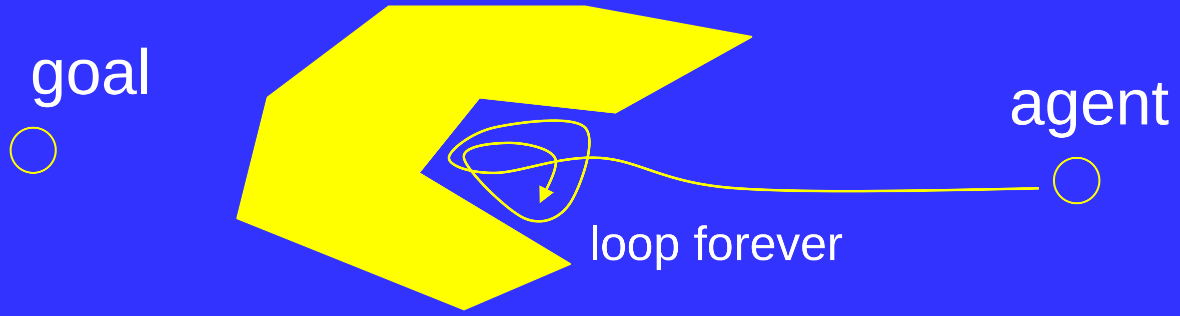


Goal-based Agents

- Knowing the current state of the environment will not always enough to decide correctly.
- You may also be given a goal to achieve.
- It may not be easy or even possible to reach that goal by just reactive decisions.

Goal-based Agents

- For instance, you will not be able to go to a specified adress in a city by just doing simple reactive bahaviors.
- Since you may enter a dead-lock situation.



- Planning towards a desired state (goal) may be inevitable.

Goal-based Agents

- In each step:
  - Gets a set of percepts (current state)

    Learn “what the world I sense now is like”
  - Integrates past state to current state to get an enriched current state (internal state)

    Keep track of the world and estimate “what the world is like now”
  - Examines actions and their possible outcomes

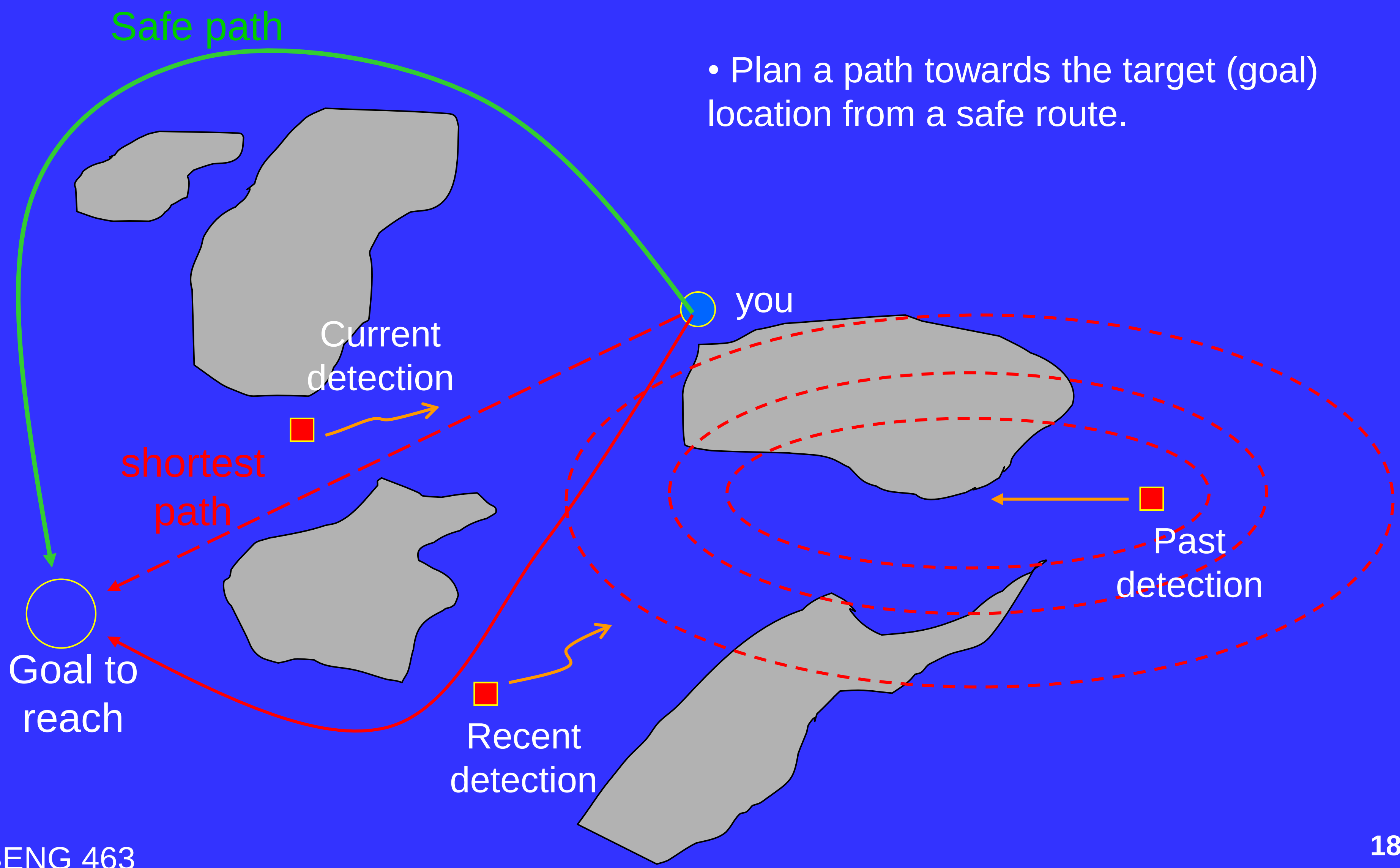
    Answer of “what the world will be like in the future if I do A,B,C,...”
  - Selects the action which may achieve the goal

    Answer of “which action->state will make me closer to my goal”
  - Performs the selected action/s

    Change the environment

Goal-based Agents

- Plan a path towards the target (goal) location from a safe route.



Utility-based Agents

- Goals are not enough to generate a high-quality behavior.
- There are many action sequences that will take you to the goal state.
- But some will give better utility than others:
  - A more safer, more shorter, more reliable, more cheaper....
- A general performance measure for computing utility is required to decide among alternatives.

Utility-based Agents

- In each step:
  - Gets a set of percepts (current state)

    Learn “what the world I sense now is like”
  - Integrates past state to current state to get an enriched current state (internal state)

    Keep track of the world and estimate “what the world is like now”
  - Examines actions and their possible outcomes

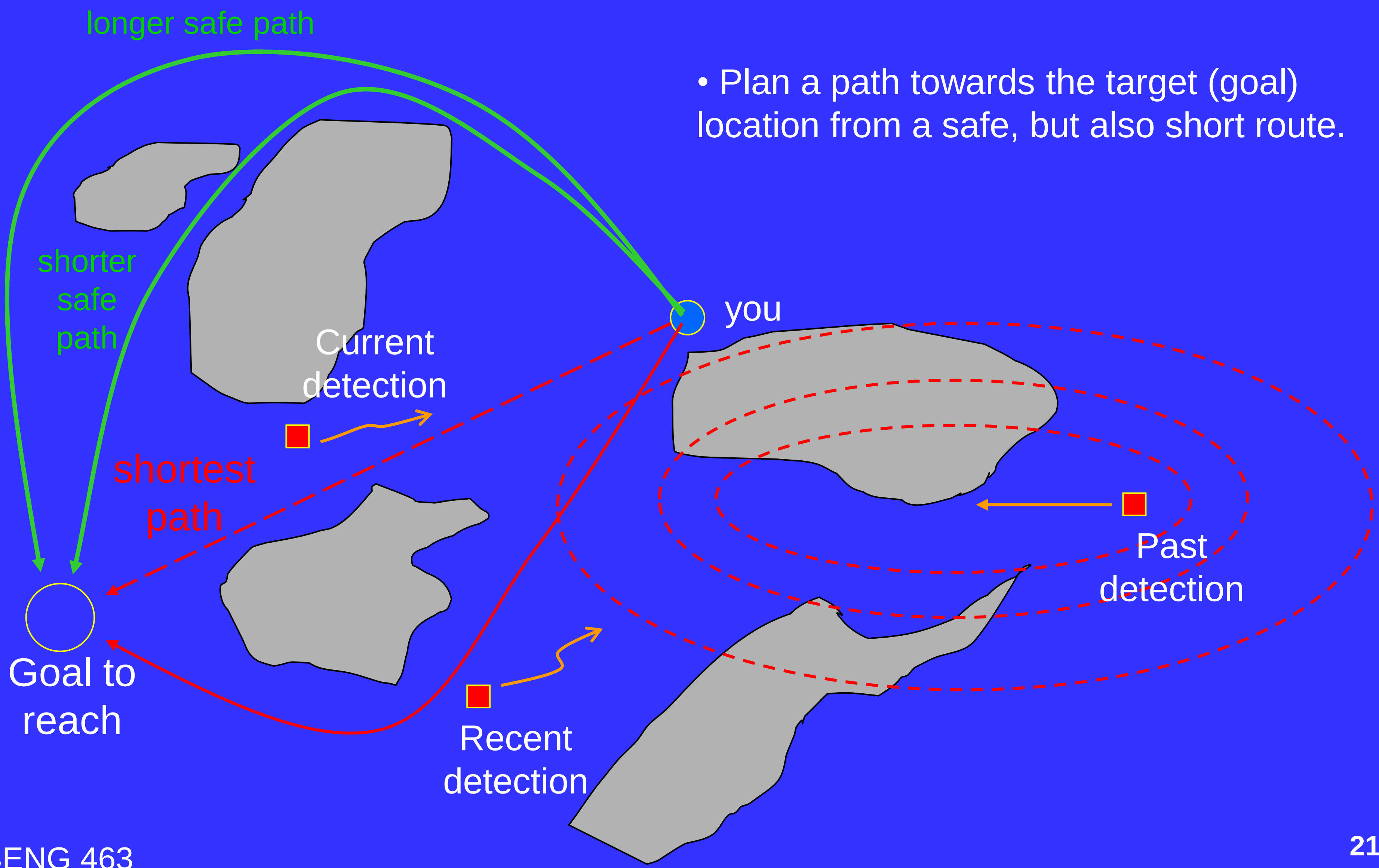
    Answer of “what the world will be like in the future if I do A,B,C,...”
  - Selects the action which may achieve the goal and maximize the utility

    Answer of “which action->state will lead me to my goal with higher utility”
  - Performs the selected action/s

    Change the environment

Utility-based Agents

- Plan a path towards the target (goal) location from a safe, but also short route.



Properties of Environments

- Accessible:
  - Sensors gives complete state of the environment (e.g. chess).
- Inaccessible:
  - Sensors gives a partial state of the environment.

Properties of Environments

- Deterministic:
  - If the next state is determined by just current state and actions of agents.
- Non-deterministic:
  - If there is uncertainty, and
  - Next state is not determined by only current state and actions of agents.

Properties of Environments

- Episodic:
  - The environment is divided into episodes.
  - Next episode is started when current one is completed.
- Non-episodic:
  - The environment is defined as a large single set.

Properties of Environments

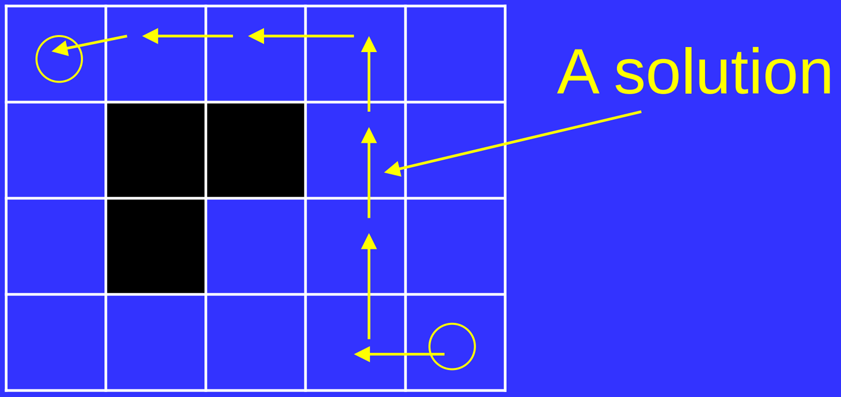
- Static:
  - If environment can not change while agent is deliberating and acting.
- Dynamic:
  - If environment can change while agent is deliberating and acting.

Properties of Environments

- Discrete:
  - If there are a limited number of distinct, clearly defined percepts and actions (chess).
- Continuous:
  - If there are infinite number of percepts and/or actions (velocity, position, ...).

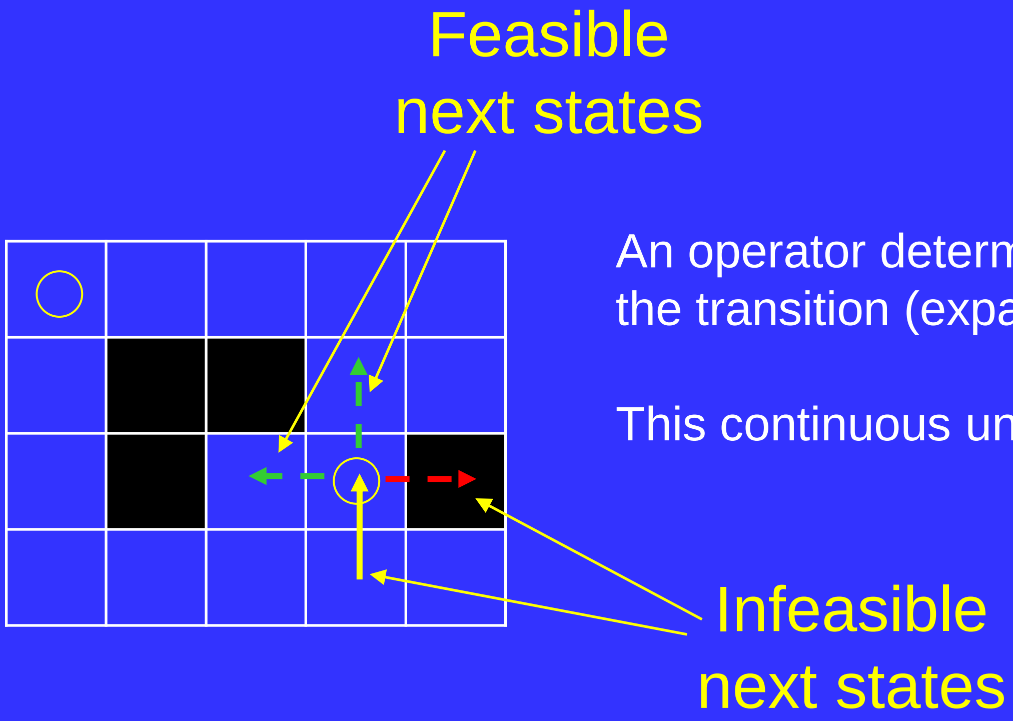
Problem Solving

- AI algorithms deal with "states" and sets of states called "state spaces".
- A state is one of the feasible steps towards a solution for a problem and its problem solver.
- So solution to a problem is a sequential and/or parallel collection of the problem states.



Problem Solving Strategies

- The problem solver applies an operator to a state to get the next state for performing a search.
- Called "state space approach"



An operator determines the next state and performs the transition (expands the next state),

This continuous until the goal state is expanded

4-Puzzle Problem

- a 4 cell board
- 3 cells filled
- 1 cell blank
- Initial: a random distribution of cells and blank
- Final: a fixed location of cells and blank

1	3
2	

initial state

	3
1	2

final (goal) state

4-Puzzle Problem

- 4 operators (actions):
  - Blank-up
  - Blank-down
  - Blank-left
  - Blank-right

1	3
2	

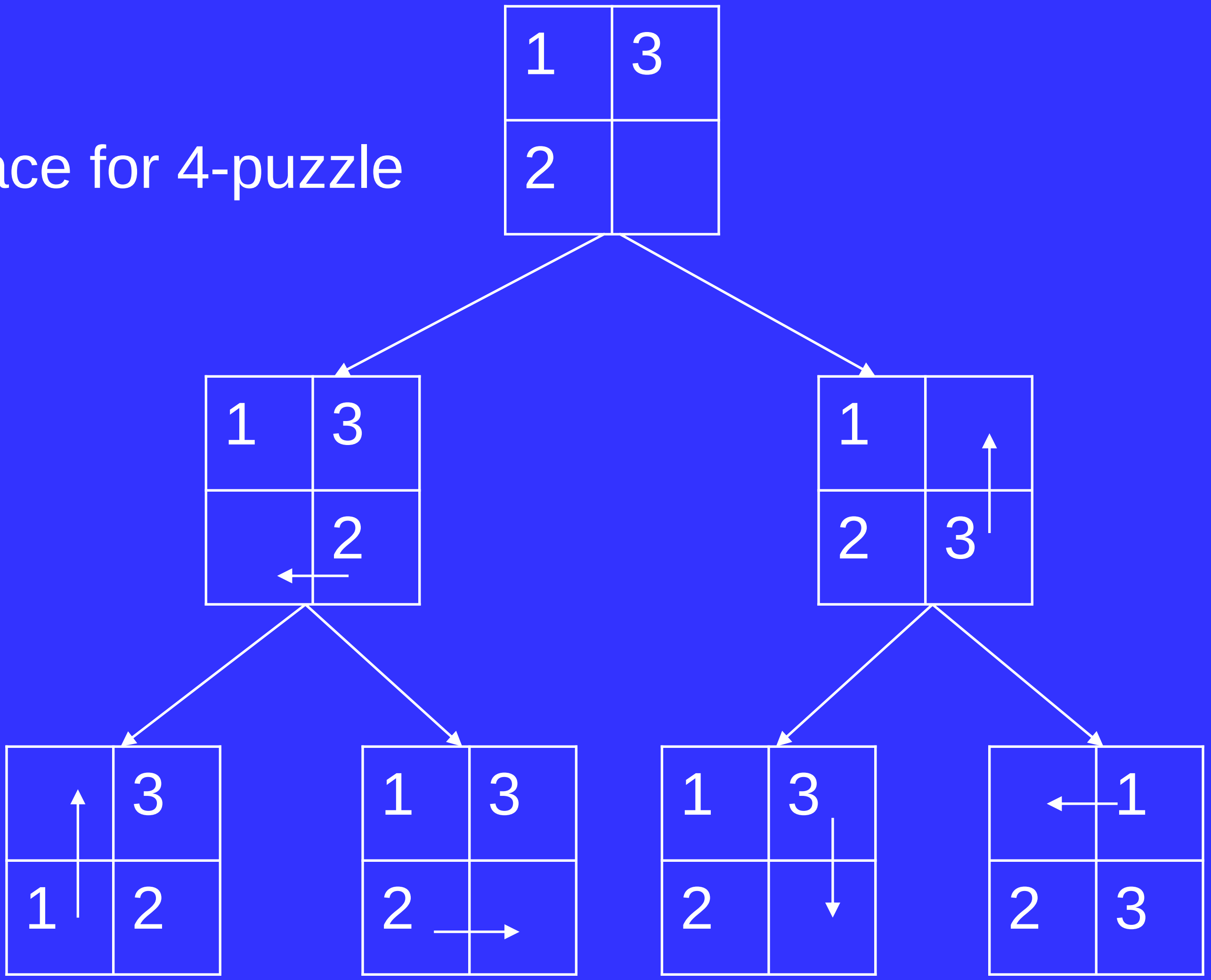
initial state

	3
1	2

final (goal) state

4-Puzzle Problem

State space for 4-puzzle

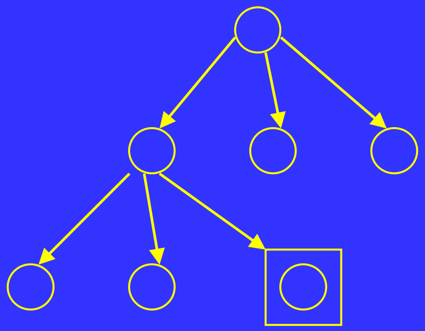


Common Search Strategies

- Generate and Test
- Heuristic Search
- Hill-Climbing

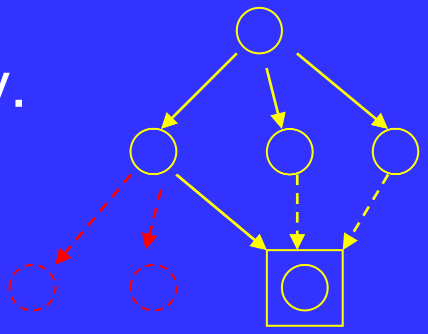
Generate and Test

- From a known starting state (root)
  - Generation of state space
- Continues expanding the reasoning space
  - Until goal node (terminal state) is reached
- In case there are muliple paths leading to goal state,
  - The path having the smallest cost from the root is preferred



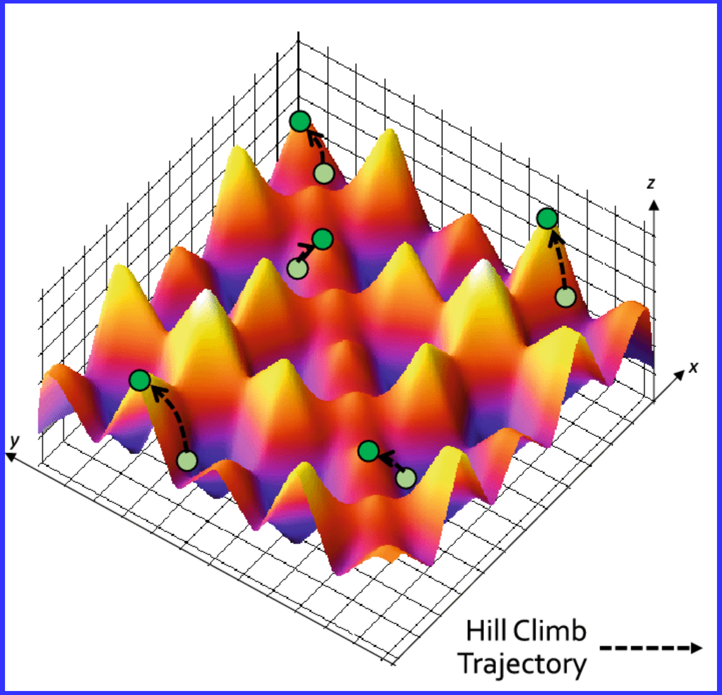
Heuristic Search

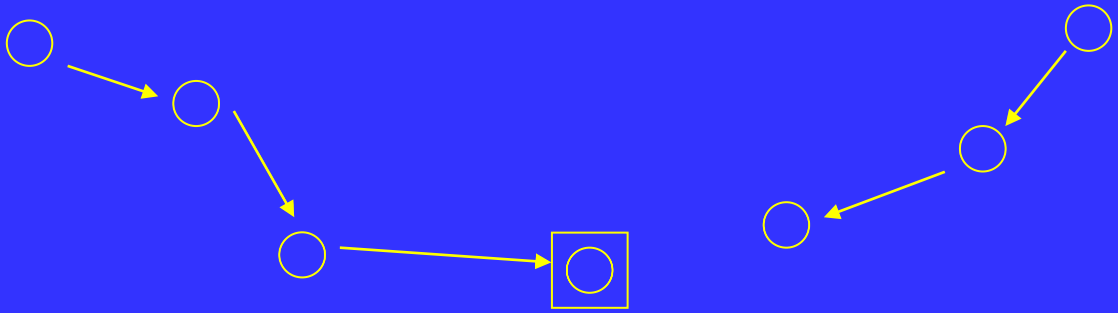
- From a known starting state (root)
  - Generation of state space
- Continues expanding the reasoning space
  - Until goal node (terminal state) is reached
- One or more heuristic functions are used to determine the better candidate states among legal states.
- Heuristics are used in order to see the measure of fitness
- Difficult task:
  - Heuristics are selected intuitively.



Hill-Climbing

- From a known or random starting state
  - Measure / estimate the total cost from current to goal state
- If there exists a neigbor state having lower cost
  - Move to that state
    - Until goal node (terminal state) is reached
- If trapped at a hilllock or local extrema
  - Select a random starting state and restart

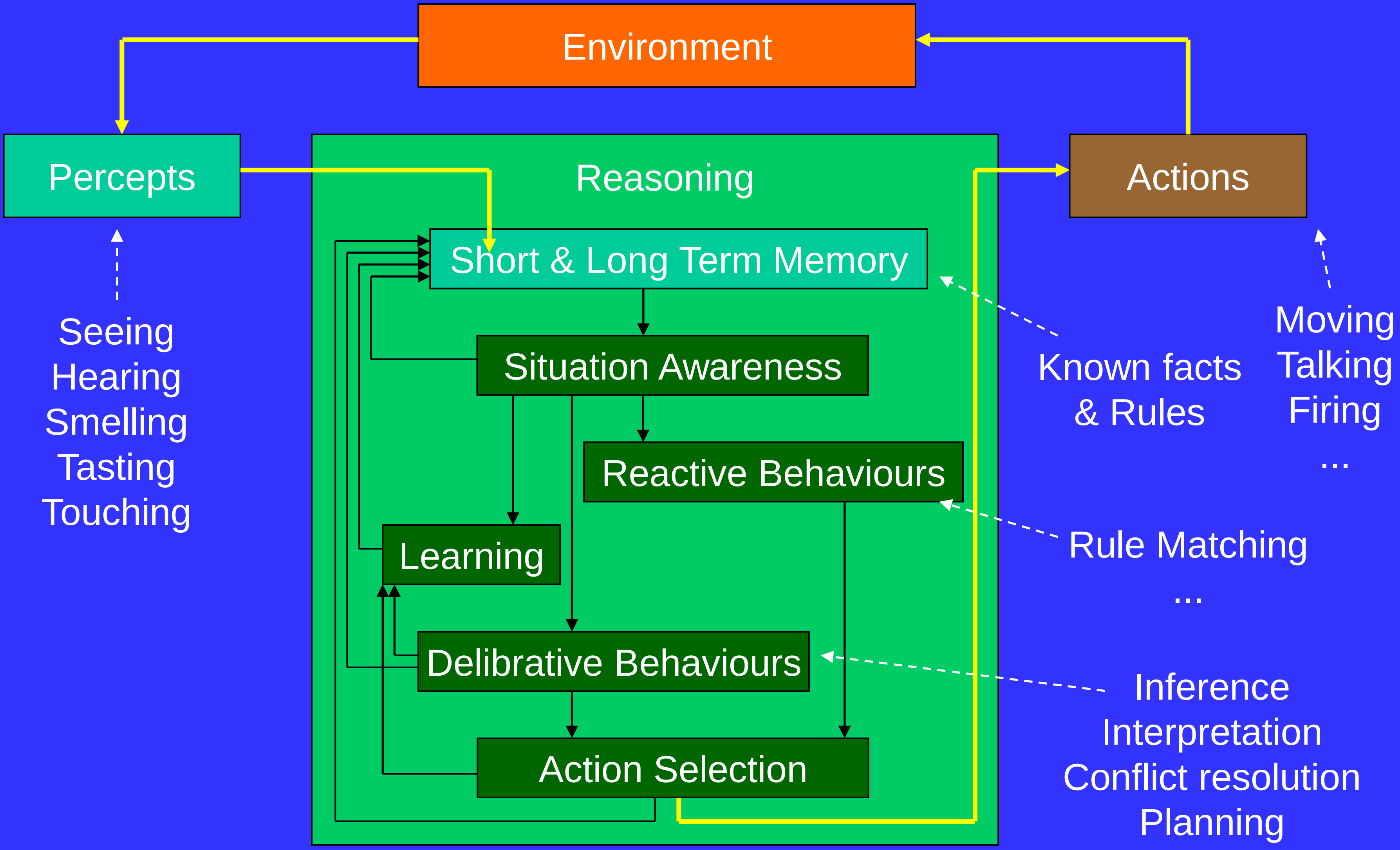




An AI Architecture

- A complete system architecture (AI engine)
  - That defines mechanizm of a percept-reasoning-action cycle
    - For modeling autonomous entity behaviors

An Example AI Architecture



Seeing
Hearing
Smelling
Tasting
Touching

Known facts
& Rules

Moving
Talking
Firing
…

Rule Matching
…

Inference
Interpretation
Conflict resolution
Planning
…

Entities

- Any kind of objects within an environment
  - Trees
  - Bushes
  - Rocks
  - Cars
  - Tanks
  - Bridges
  - Human beings
  - Animals
  - Houses
  - ....

Attributes

- Any kind of parameters describing properties of an entity
  - Tank
    - Type
    - Coordinate
    - Orientation
    - Velocity
    - Damage
    - Gun 1
      - Loaded
      - Number of munitions left
    - Gun 2
      - Loaded
      - Number of munitions left

Relations

- Any link between two entities, groups of entities or types of entities defining a fact
  - Ahmet is father of veli
  - Suzan is mother of veli
  - Mustafa, Kenan and Oya are friends
  - Iraq forces is an opponent of US forces
  - ....
  - X may have a relation with Y

  - Strike is an eagle
  - Eagle is a bird
  - Bird is an animal

Tasks

- Any piece of work,
  - Which has been undertaken or attempted
  - By someone
  - To reach a desired goal

  - Playing backgammon
  - Doing a project
  - Going to school
  - Performing surveillance
  - Managing an accident
  - Defending a town
  - Attacking a town

Sub-Tasks

- Smaller pieces of work to be done in order to perform and complete a task

- Sub Tasks of Going To School Task
  - Remembering the location of school
  - Planing an initial path to school
  - Going to school through the path
    - Replanning a path to school if required
  - Entering the school
  - Planing an initial path to the classroom
  - Going to the classroom through the path
    - Replanning a path to classroom if required
  - Entering the classroom

Actions

- A primitive (may be un-interruptable) work to be done in order to perform a task
  - Do a chess move
  - Do a forward step
  - Change direction
  - Hang up a phone
  - Send a fax
  - Say a statement to someone
  - Talk to someone
  - Engage a target
  - Fire a target
  - Throw a bomb
  - Give an order

States

- A position in time.
- A specific situation/case/condition
  - Among a set of all possible situations a system may be in.
- Standing, Walking, Defending, Attacking
- Usually actions cause state transitions.

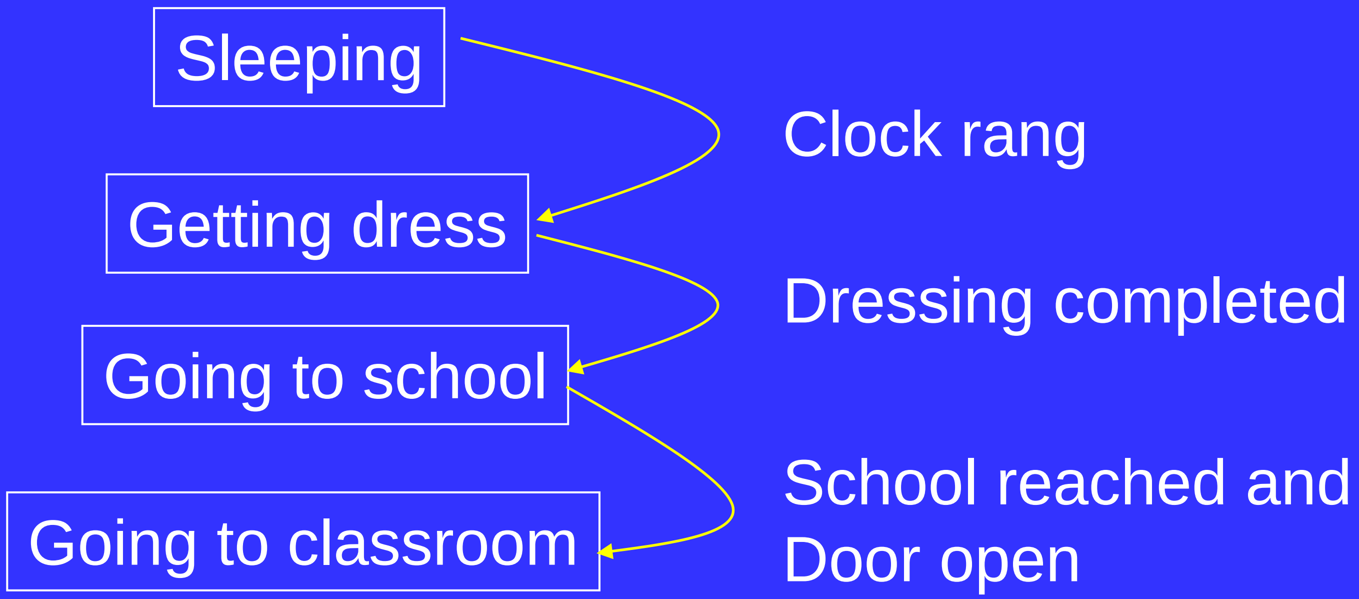
Events

- Something that happens in time.
- An action, occurrence or a condition that causes a state transition
- Thus in addition to our actions,
  - An external or internal event may cause a state transition.

Events

States

Events



Sleeping

Getting dress

Going to school

Going to classroom

Clock rang

Dressing completed

School reached and
Door open