

Lecture 4: Problem-solving as search

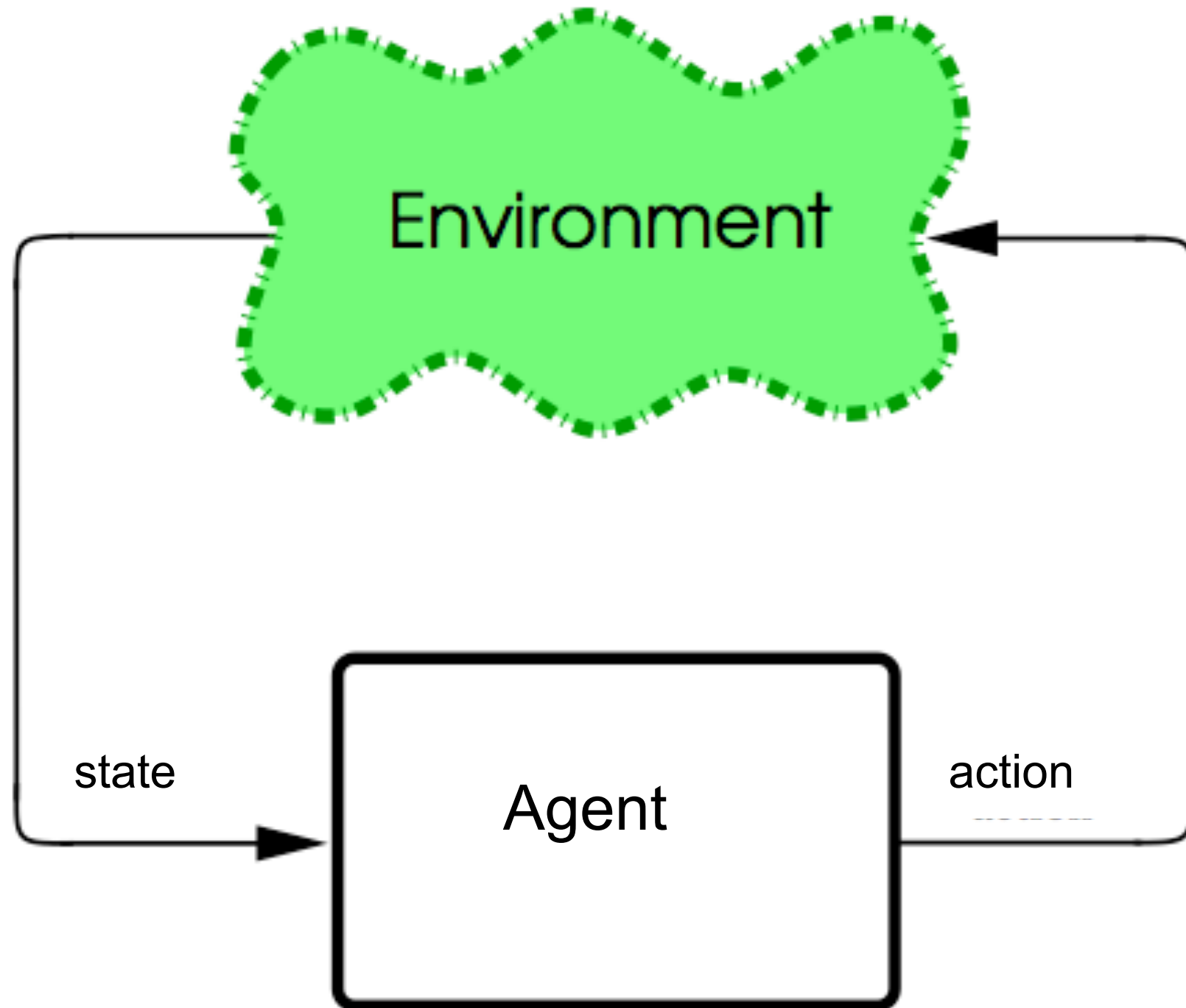
Artificial Intelligence

CS-GY-6613-I

Julian Togelius

julian.togelius@nyu.edu

An agent



Problem-solving as search

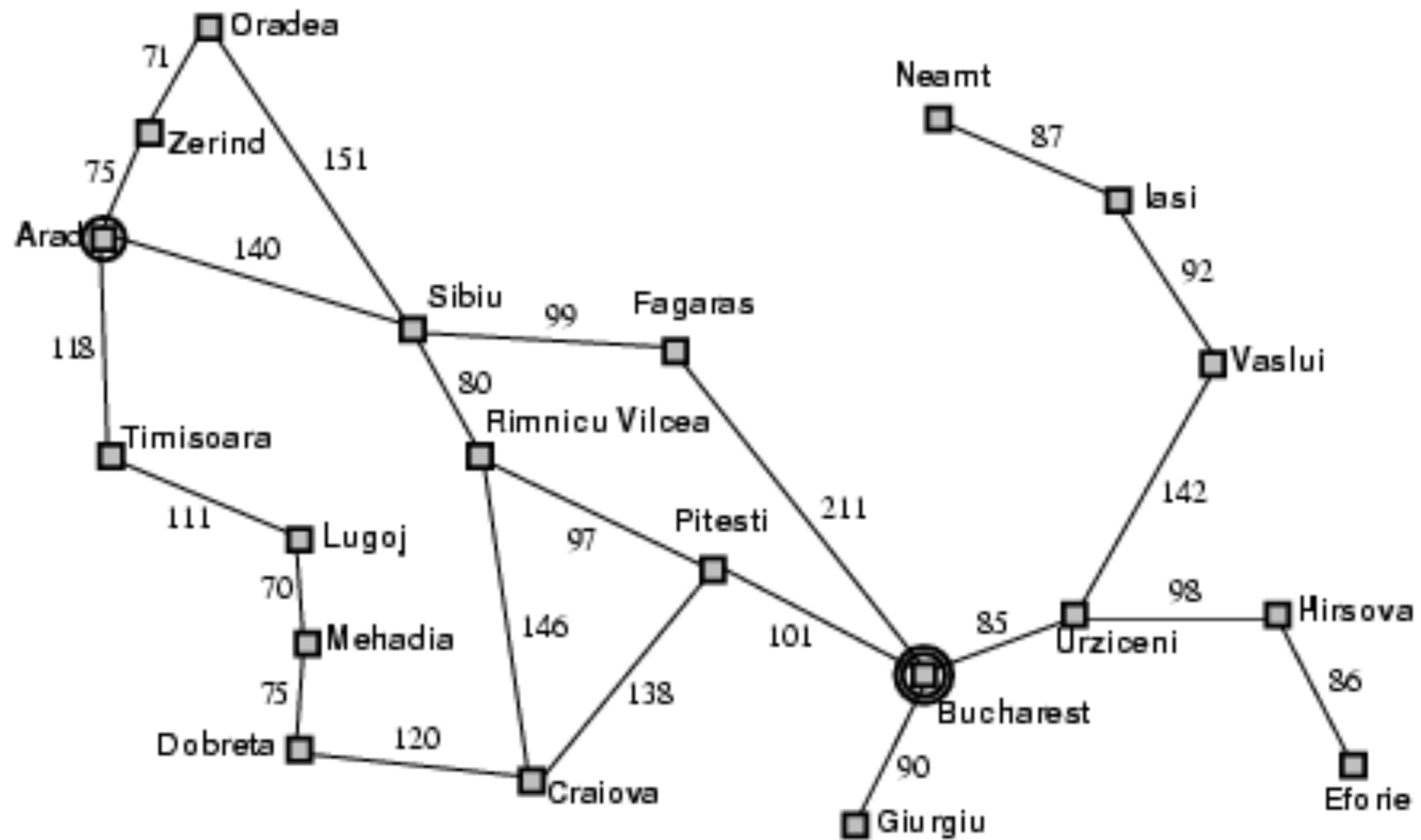
- An agent is in a *state*
- It wants to get to a *goal state*
- It needs to find the sequence of actions that gets it there
 - (cheapest, fastest, safest...)

Problem-solving agent

```
function SIMPLE-PROBLEM-SOLVING-AGENT(percept) returns an action
  static: seq, an action sequence, initially empty
           state, some description of the current world state
           goal, a goal, initially null
           problem, a problem formulation

  state ← UPDATE-STATE(state, percept)
  if seq is empty then do
    goal ← FORMULATE-GOAL(state)
    problem ← FORMULATE-PROBLEM(state, goal)
    seq ← SEARCH(problem)
  action ← FIRST(seq)
  seq ← REST(seq)
  return action
```

From Arad to Bucharest



Problem types

- Deterministic, fully observable: single-state problem
 - Agent knows exactly which state it will be in; solution is a sequence
- Non-observable: sensorless problem
 - Agent may have no idea where it is; solution is a sequence
- Nondeterministic and/or partially observable: contingency problem
 - percepts provide new information about current state
- Unknown state space: exploration problem

What problem type is...

- Driving to Bucharest?
- Flying to Bucharest?
- Playing Pac-Man?
- Playing StarCraft?
- Playing Poker?
- Living a good life?

Problem components

- Initial state
- Actions and successor function
 $S(\text{state}, \text{action}) \rightarrow \text{state}'$
- Goal test (are we there yet?)
- Path cost
- A solution is a sequence of actions leading from the initial state to a goal state

Problem components?

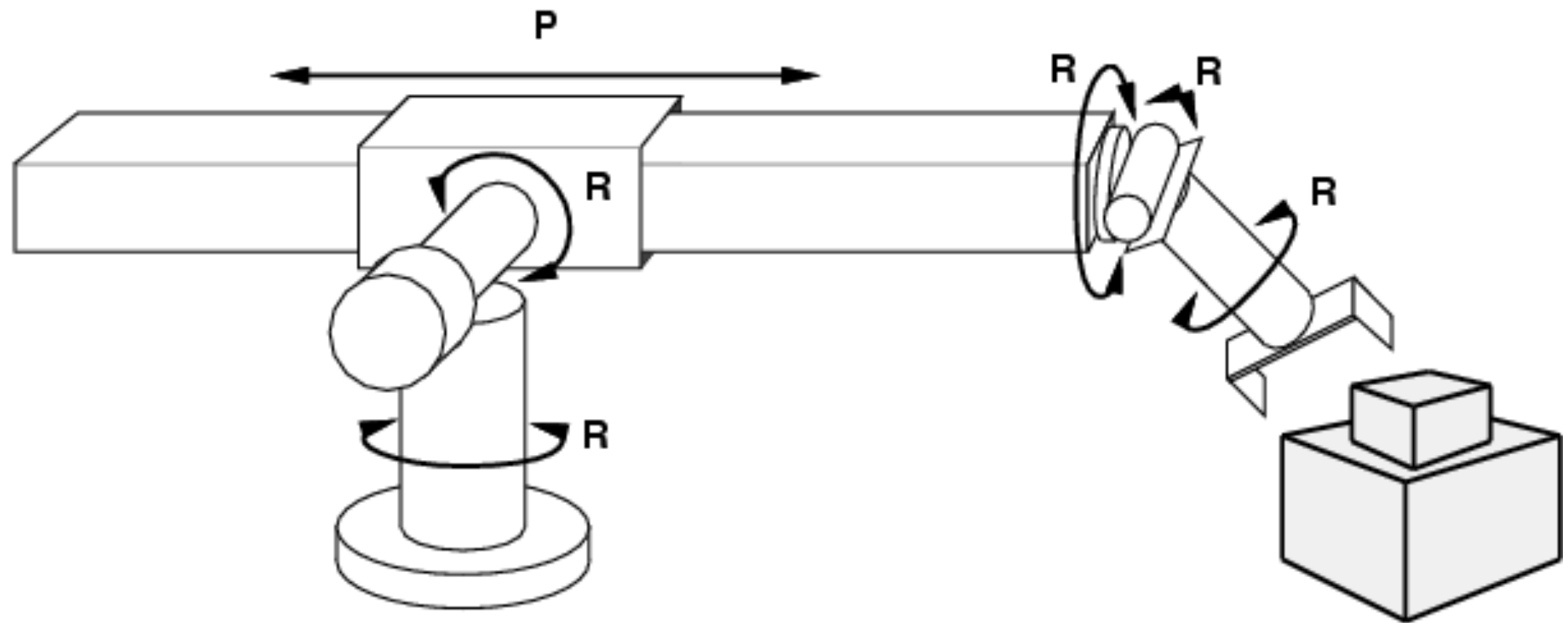
7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

Goal State

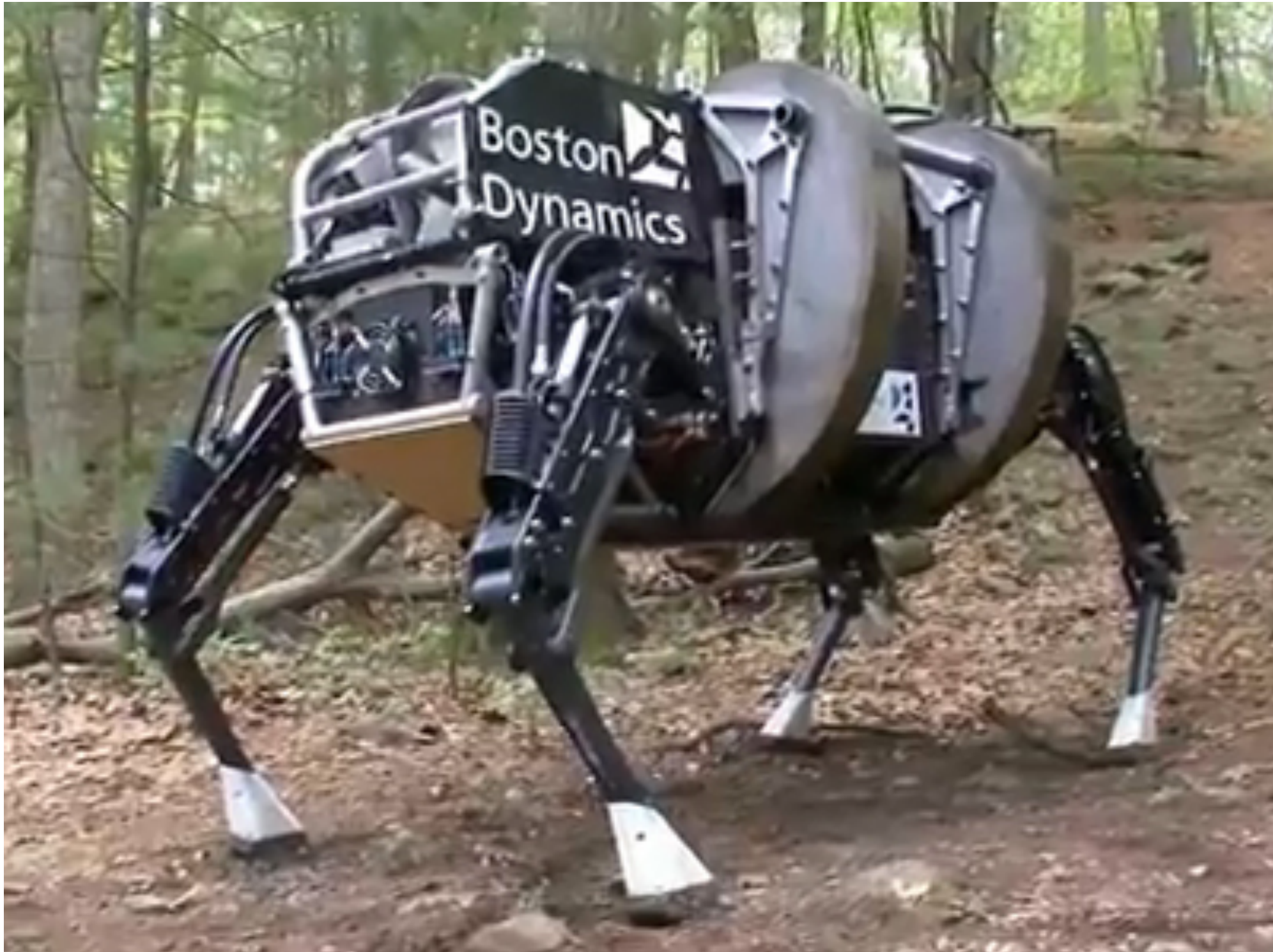
Problem components?



Problem components?



Problem components?



Problem components?



Problem components?

