

A model of planning in human complex problem solving

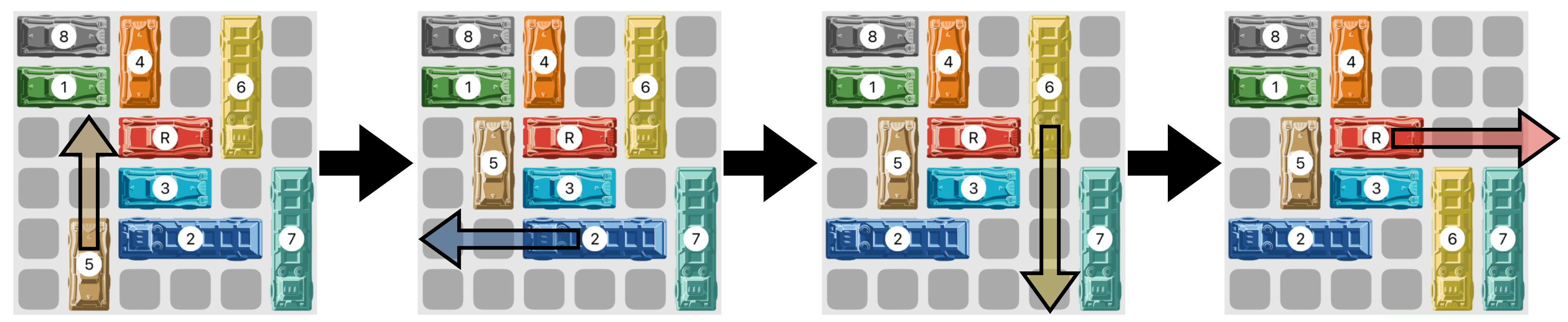
Jeroen Olieslagers¹, Zahy Bnaya¹, Wei Ji Ma^{1,2}

Center for Neural Science¹ and Department of Psychology², New York University



The task

- Requirements:**
- Complex yet tractable
 - Minimal perceptual effects
 - Minimal social component
 - Fun
- Rush Hour**
- Need to plan multiple steps ahead
 - Tractable state space
 - Optimal solutions available
 - Minimal pattern detection
 - Single player
 - No language
 - Popular game



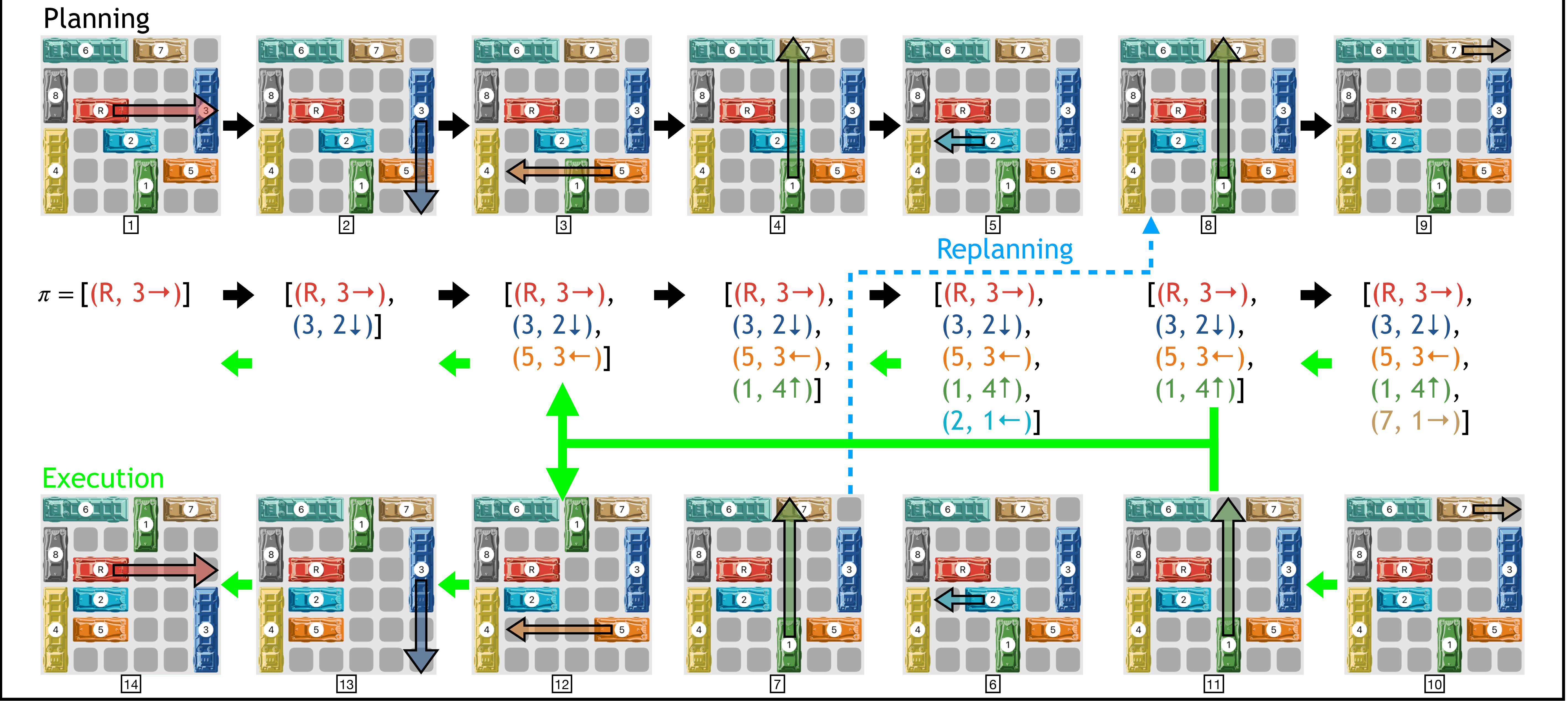
The implementation

```

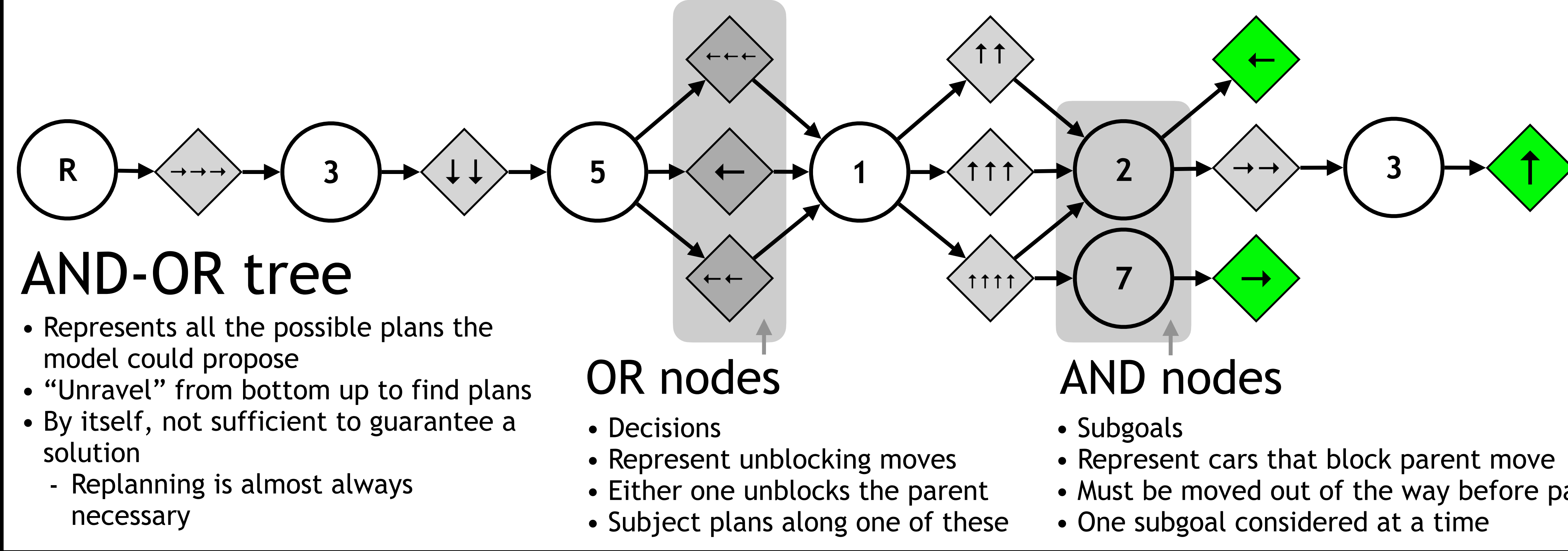
Algorithm solve_puzzle(s)
  a_init ← find_solving_move(s)      ▷ Ultimate goal of the game
  π ← [a_init]
  while π ≠ ∅ do
    a ← π.pop()                      ▷ Choose move at top of plan
    if is_valid_move(s, a) then      ▷ Execution phase
      s ← make_move(s, a)
    else
      π ← extend_plan(s, π)          ▷ Replanning
    end if
  end while

Algorithm extend_plan(s, π)
  a ← π[end]                        ▷ Select most recently planned move
  C ← find_blocking_cars(s, a)      ▷ AND node creation
  if C = ∅ then                      ▷ If move is valid, end planning phase
    return π
  end if
  c ← choose_car(s, C)              ▷ AND node selection
  A ← find_unblocking_moves(s, a, c) ▷ OR node creation
  ā ← choose_move(s, A)             ▷ OR node selection
  π.push(ā)                         ▷ Plan extension
  return extend_plan(s, π)          ▷ Recurse
  
```

The model



The representation



- Model fitting**
- Past moves inform which chain a subject is on
 - Each chain has a different probability according to model parameters:
 - γ : stopping probability
 - λ : lapse rate
 - h : heuristics to decide which AND/OR node to expand
 - Based on last move in chain, probabilities over chains turn into probabilities over moves