Decision Making

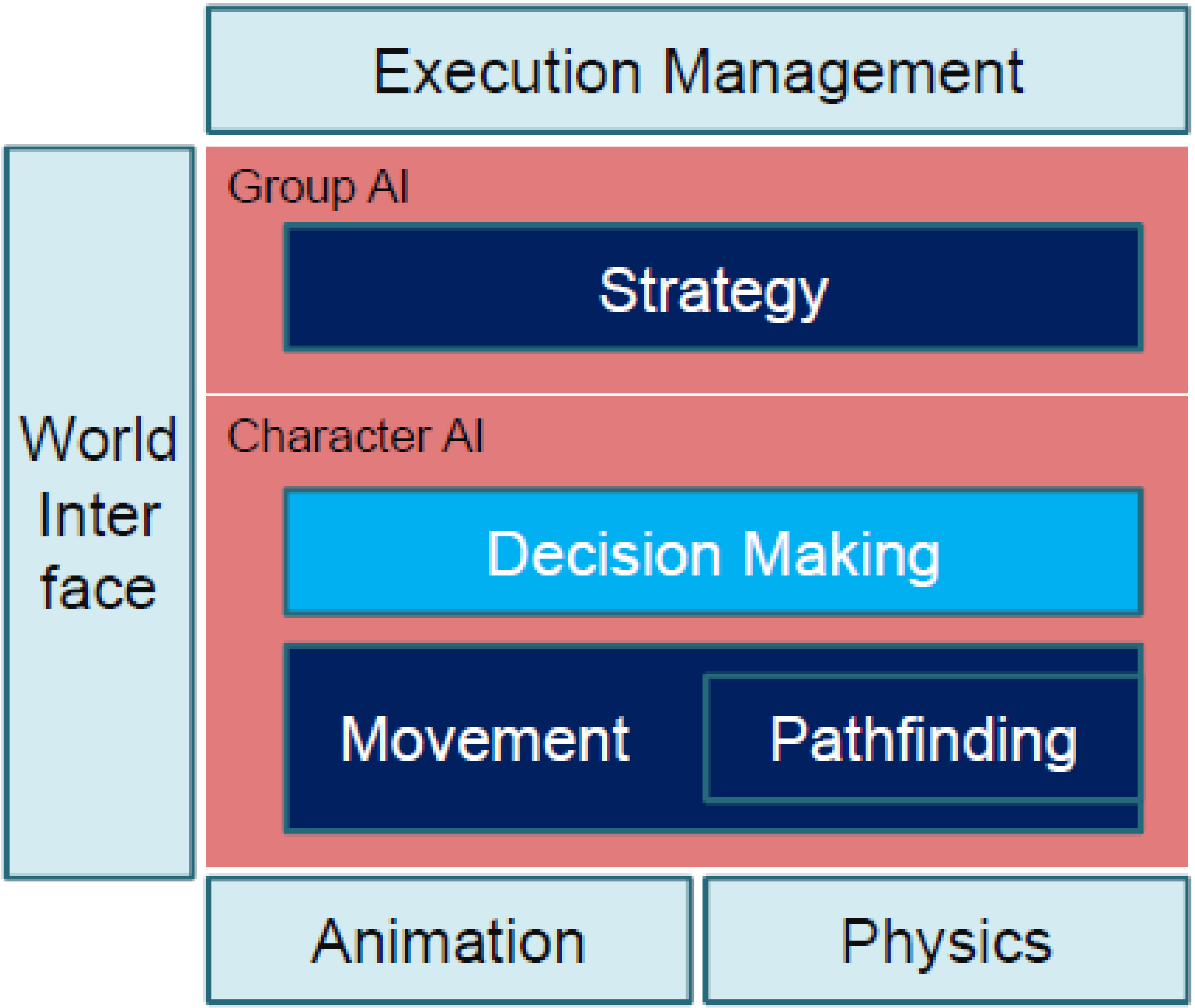
Basics

선택되는 Action 결정.

- Avatar has a set of information
- Avatar has a goal
- Needs to generate a sequence of actions in order to reach the goal

AI문제 경로를 찾는 문제 또는 탐색과 같다.

본 강의에서, **Avatar**는 **‘Agent’**, **‘AI를 가진 Character’**와 같은 용어로 사용



Execution Management

World Interface

Group AI

Strategy

Character AI

Decision Making

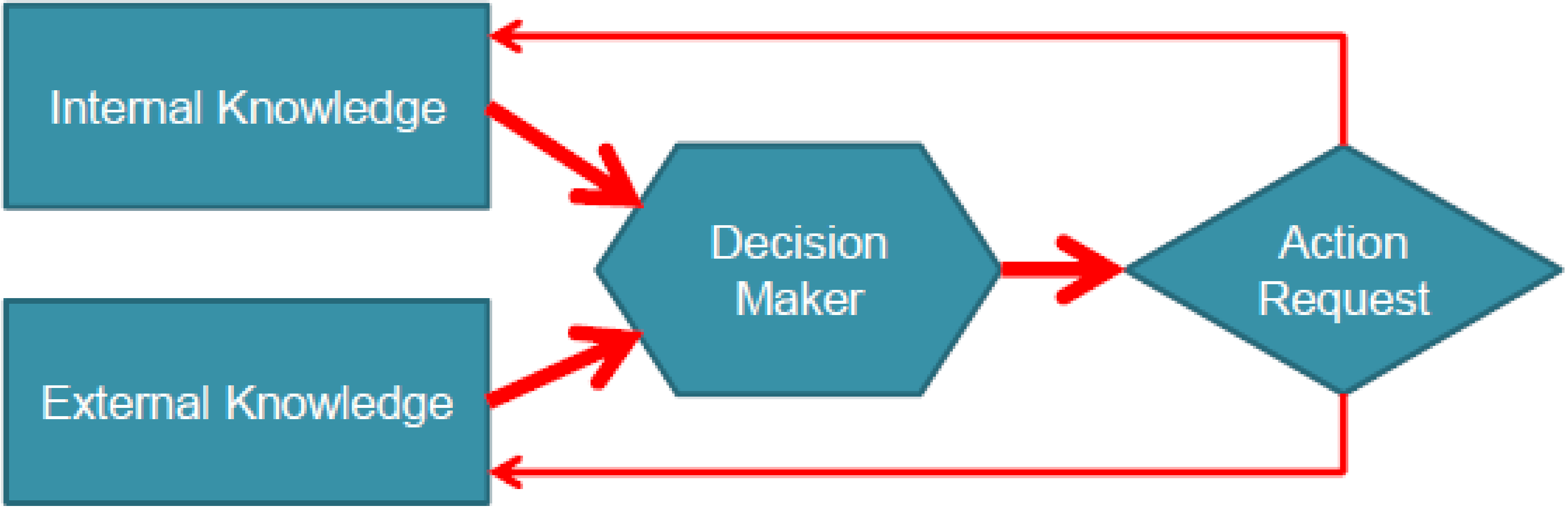
Movement

Pathfinding

Animation

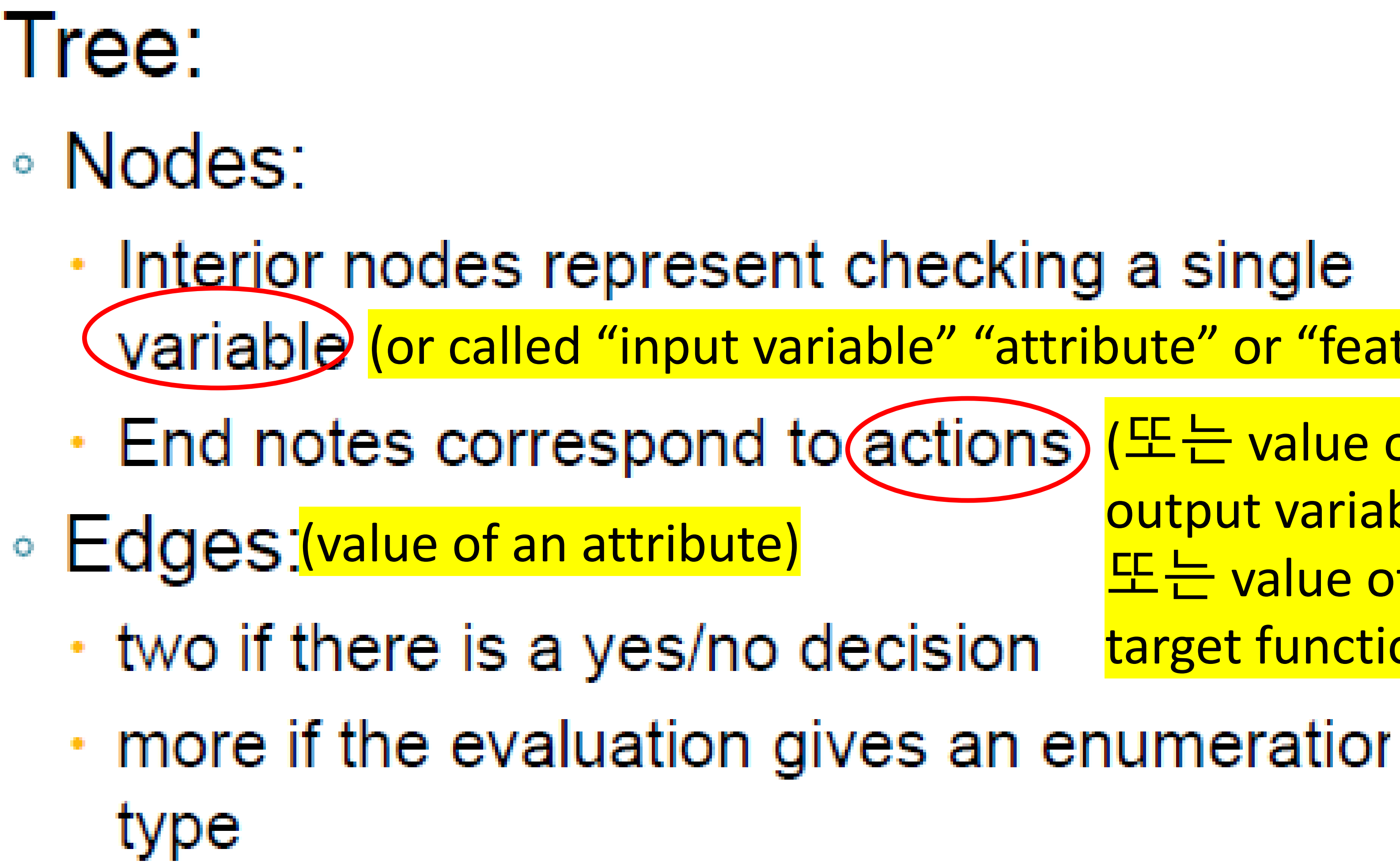
Physics

Decision Making



External Knowledge from Environment

DECISION TREE



Tree:

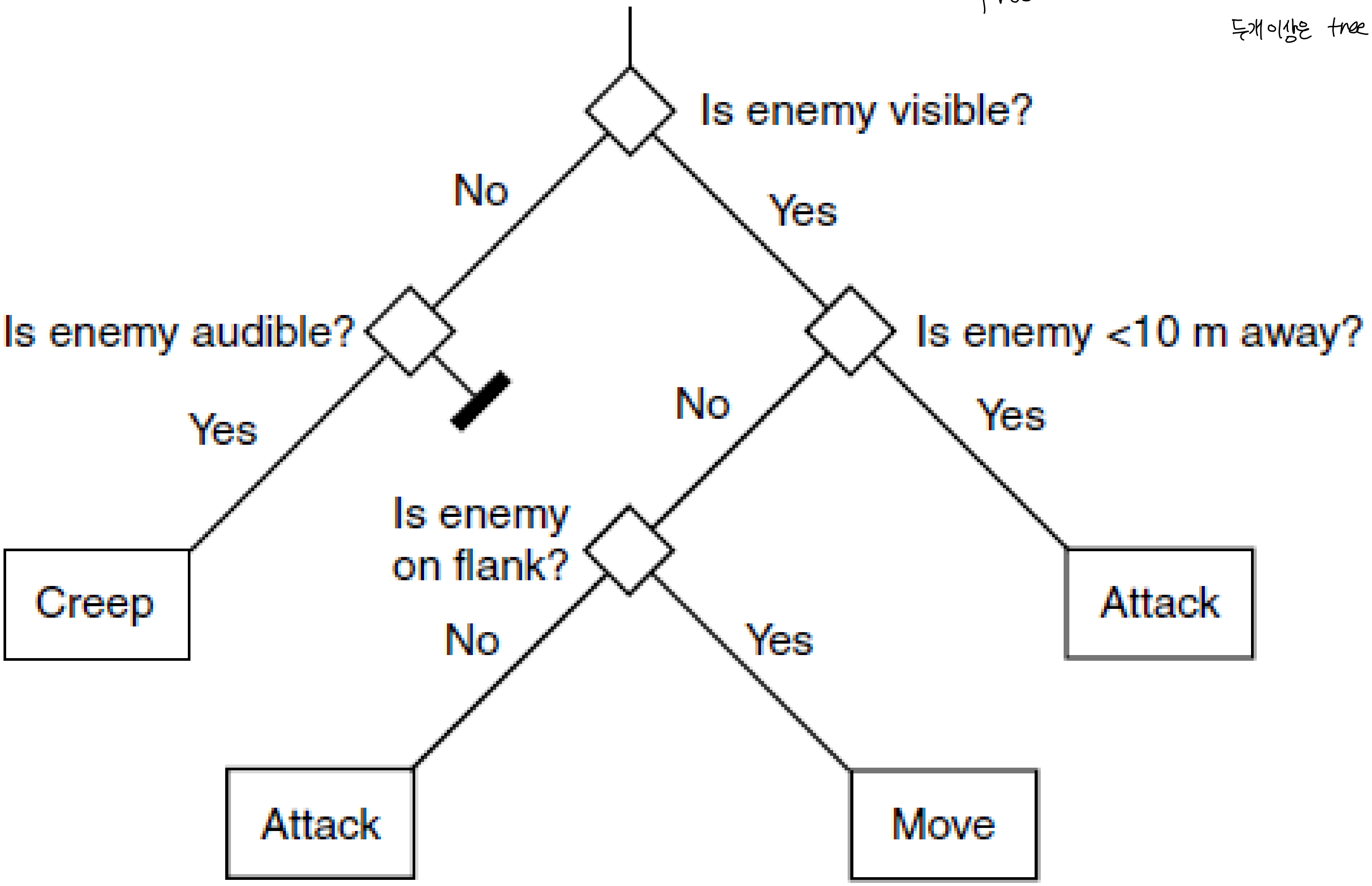
- Nodes:
  - Interior nodes represent checking a single variable (or called “input variable” “attribute” or “feature”)
  - End notes correspond to actions (또는 value of an output variable 또는 value of a target function)
- Edges:(value of an attribute)
  - two if there is a yes/no decision
  - more if the evaluation gives an enumeratior type

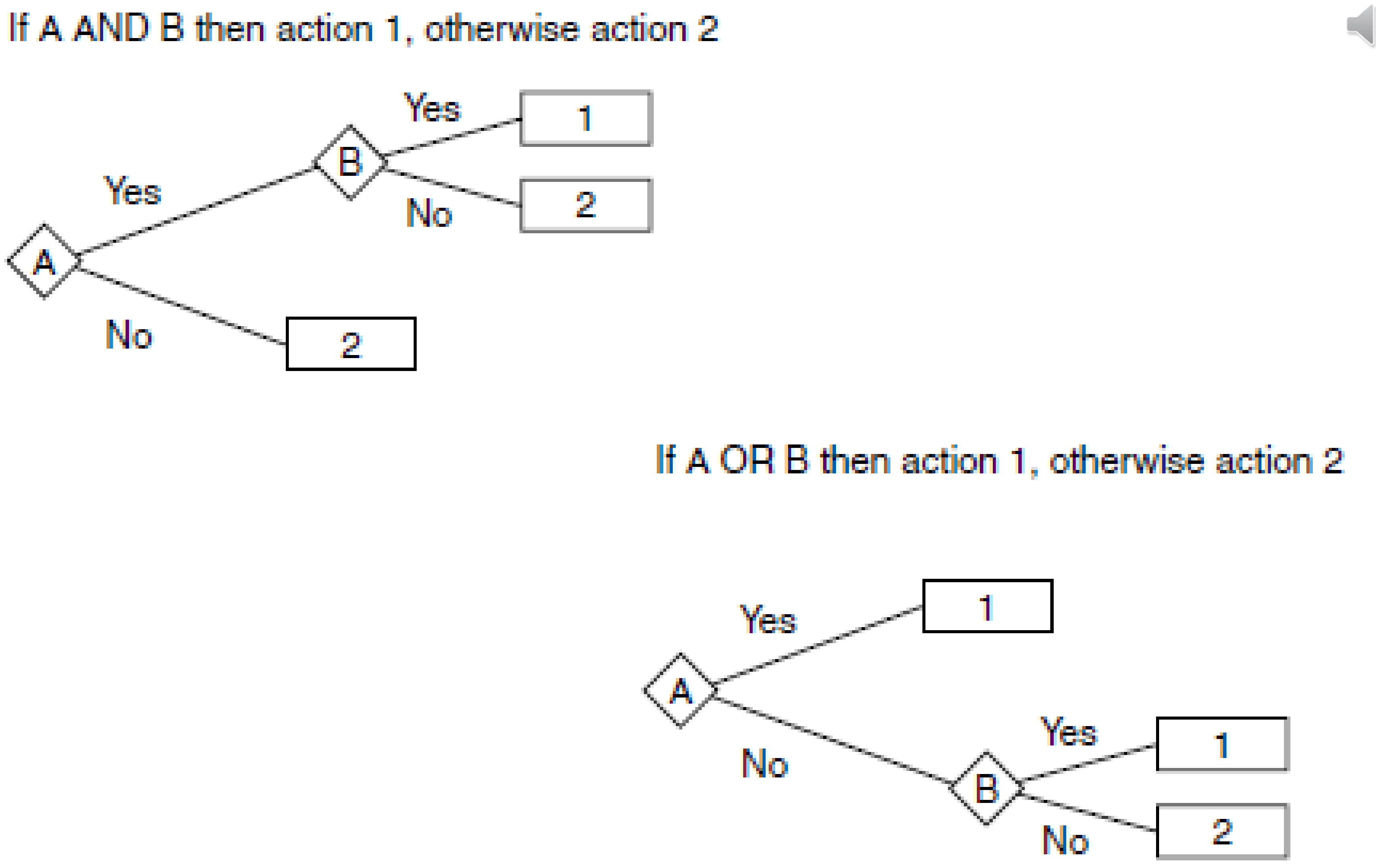
Each path corresponds to a conjunction of attribute tests for a case. Ex) ("Is enemy visible" == No) AND ("is enemy audible == "Yes")

A decision tree represents a disjunction of such paths (conjunction of constraints on the attribute values of cases).

tree는 사이클 X

부모노드는 1개
두개이상은 tree x

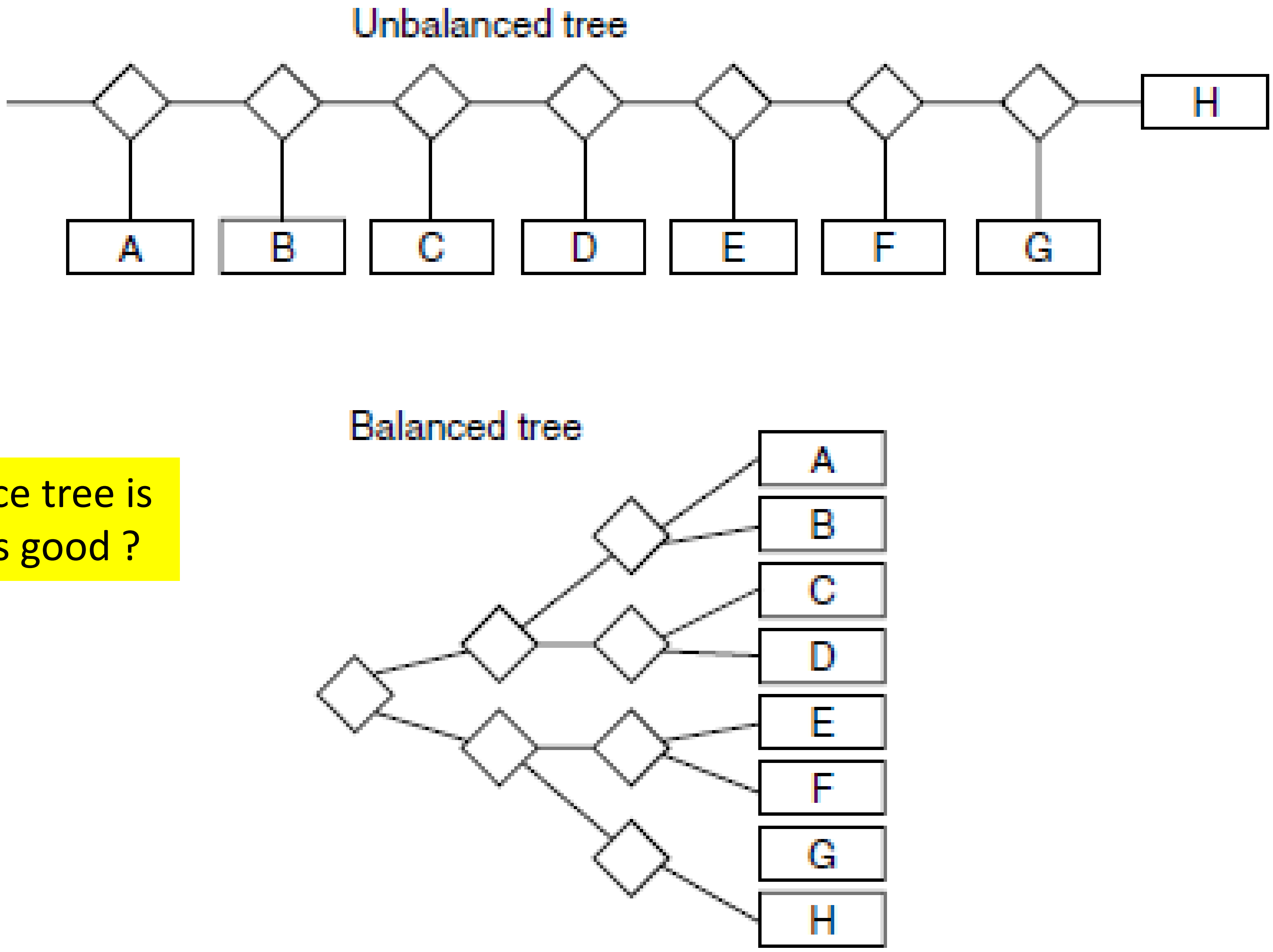




If A AND B then action 1, otherwise action 2

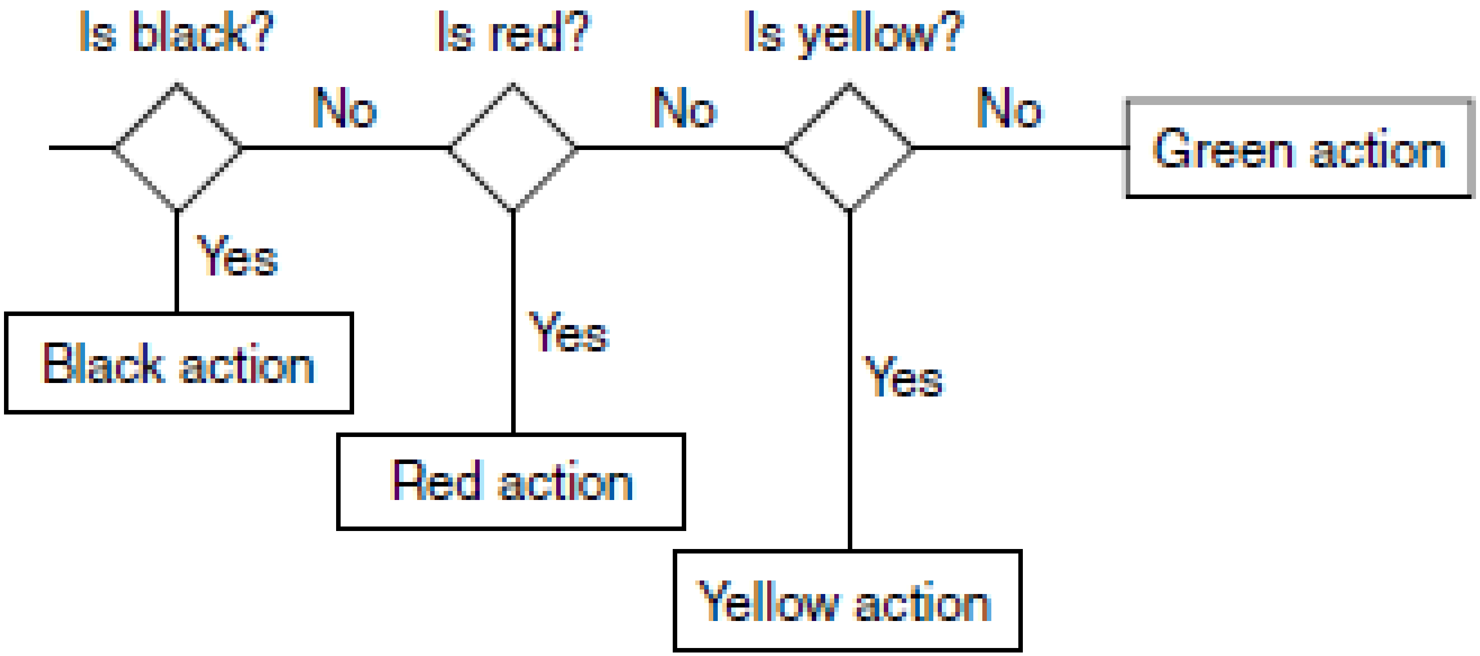
If A OR B then action 1, otherwise action 2

5 Trees representing AND and OR



Balanced and unbalanced trees

Balance tree is always good ?



Is black?

Is red?

Is yellow?

No

No

No

Yes

Yes

Yes

Black action

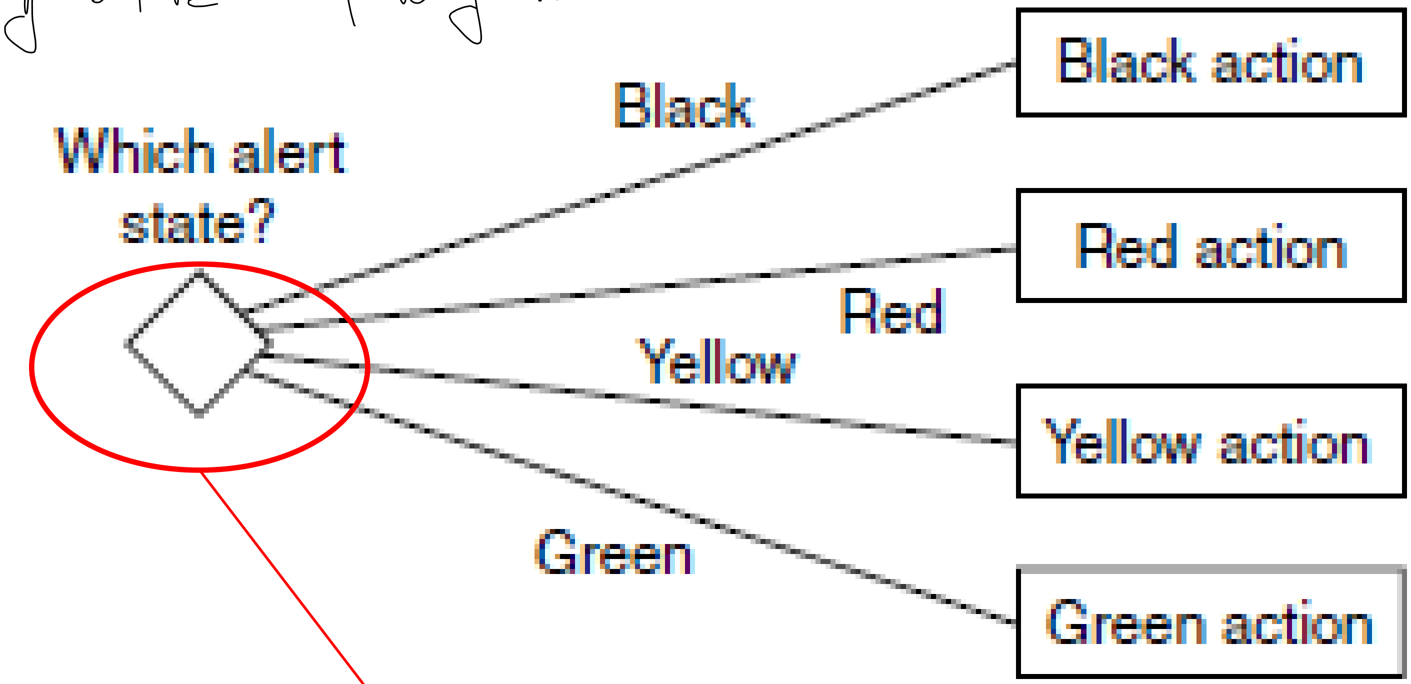
Red action

Yellow action

Green action

Deep binary decision tree

binary check이 아닌 4-way check



Flat decision tree with four branches

Enumeration type (열거형) output을 출력하는 checking node

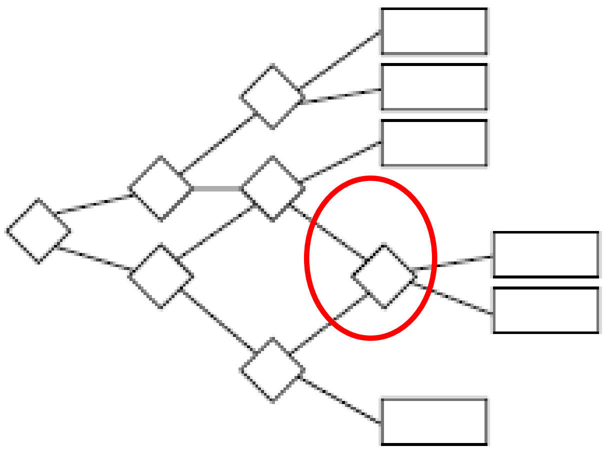
Decision Tree

Transform a decision tree in order to achieve better performance

- Use dynamic programming for an optimal solution if statistics are known

공유되는 subtree들이 존재하도록, 필요하다면 decision tree들의 모양을 변형시킴.

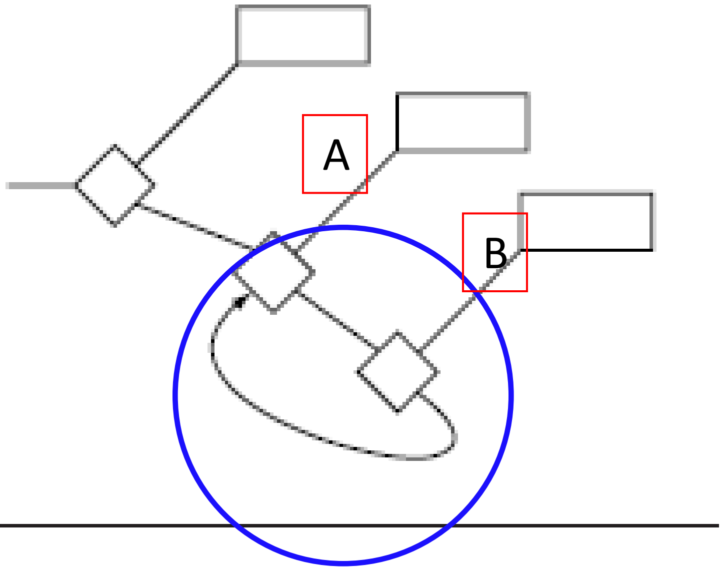
Beyond the Tree



실행시간의 단축은 기대할 수 없다.

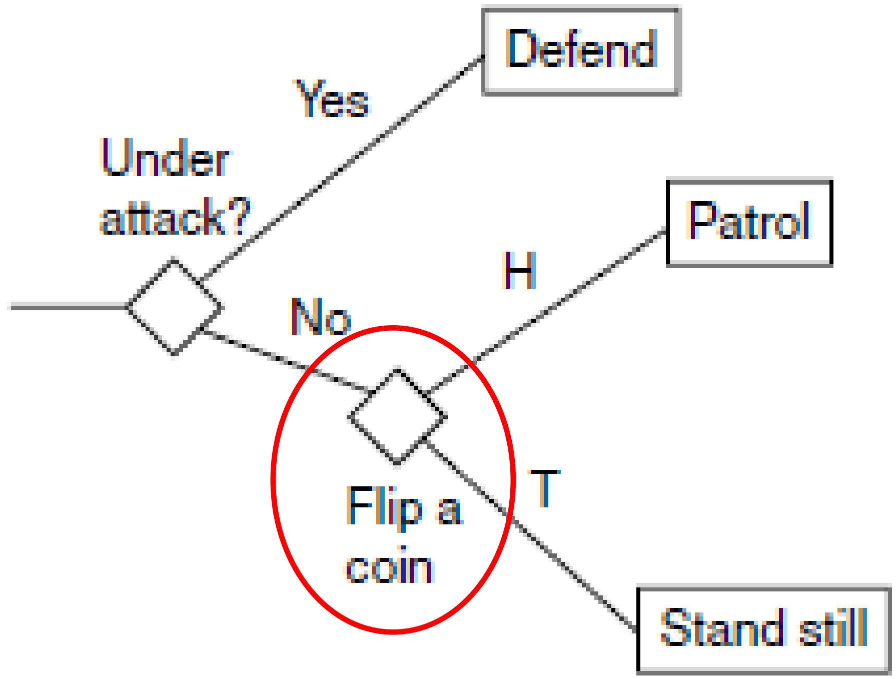
decision graph 정도로 볼수 있다.

Merging branches



Pathological tree

Random Decision Trees



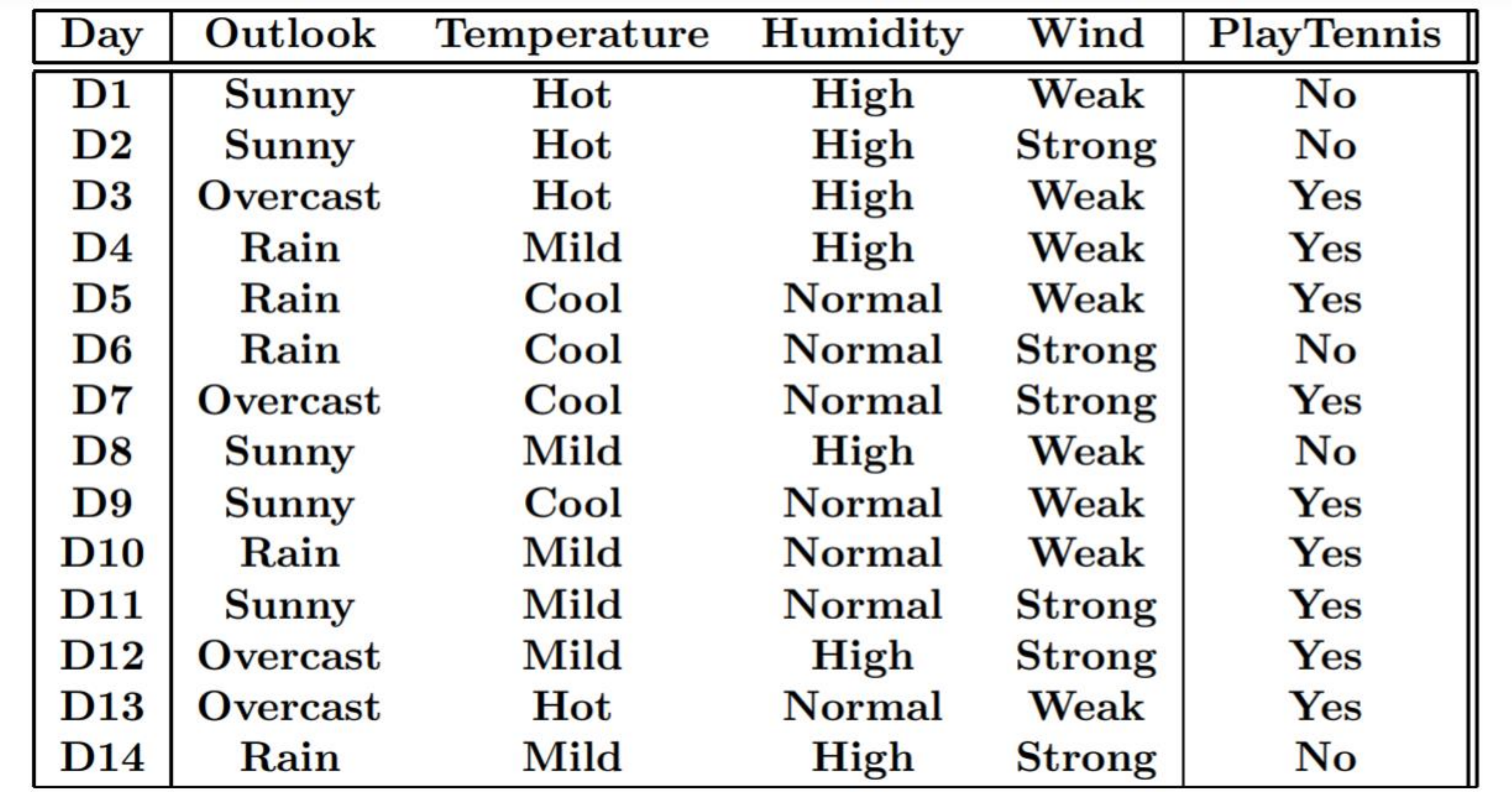
Random tree

Use a timer for discarding the random decision currently being used, and try to flip a coin again.

Decision Tree Learning (Induction of Decision Tree)

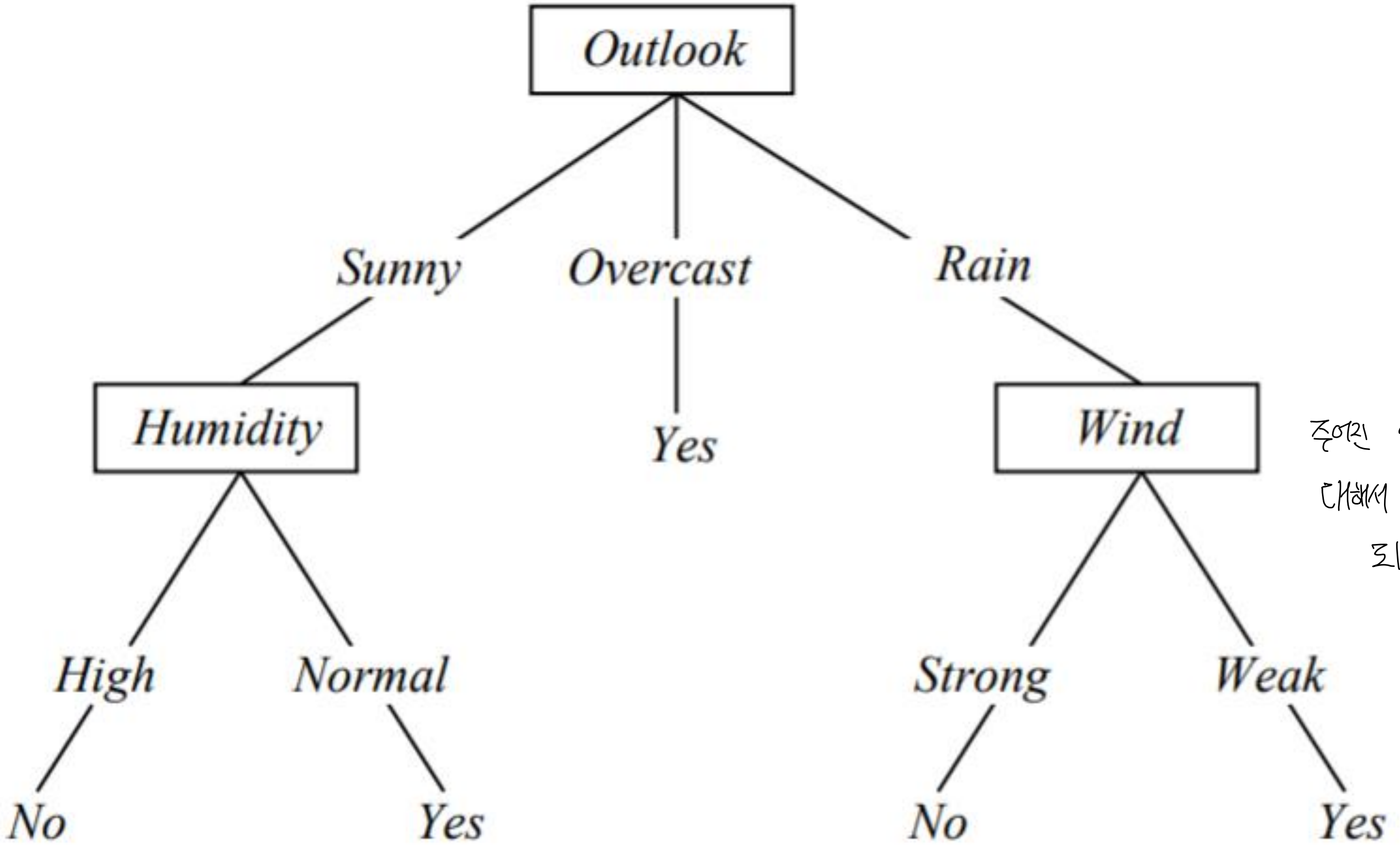
음성없음

Given the following data, predict the value of “PlayTennis” for (Outlook = sunny, Temp = cool, Humidity = high, Wind = strong)



Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Decision Tree for *PlayTennis*



음성없음

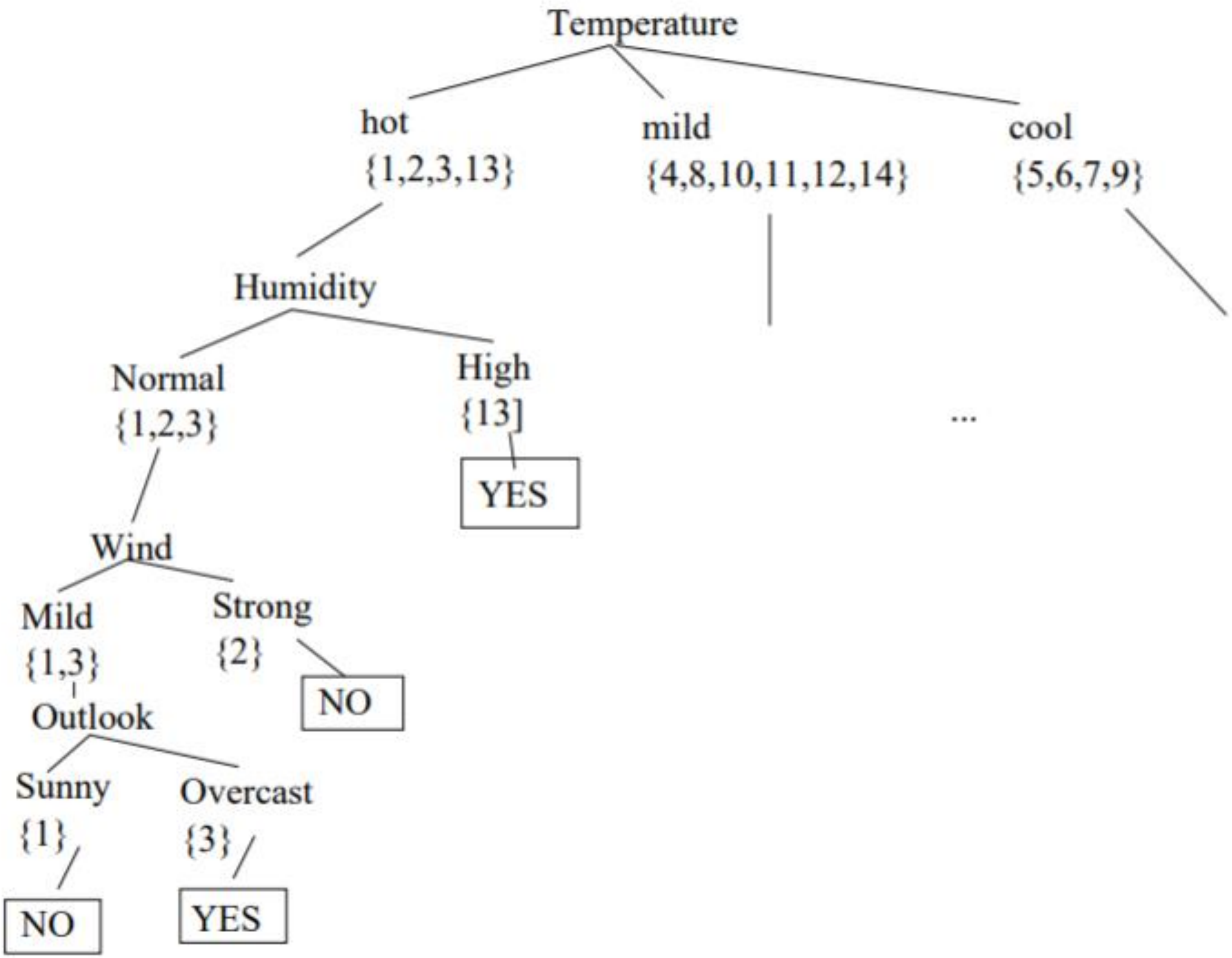
주어진 입력 사례에
대해서 분류가 어떻게
되는지 분류문제처럼

이런 Decision Tree를 Classification Tree 라고 부름.

Regression Tree: predicted output value is a real value (e.g. the price of a house, or a patient’s length of stay in a hospital).

Alternative Decision Tree for *PlayTennis*

음성없음



What is different?

Which attribute is a root node ?

음성없음

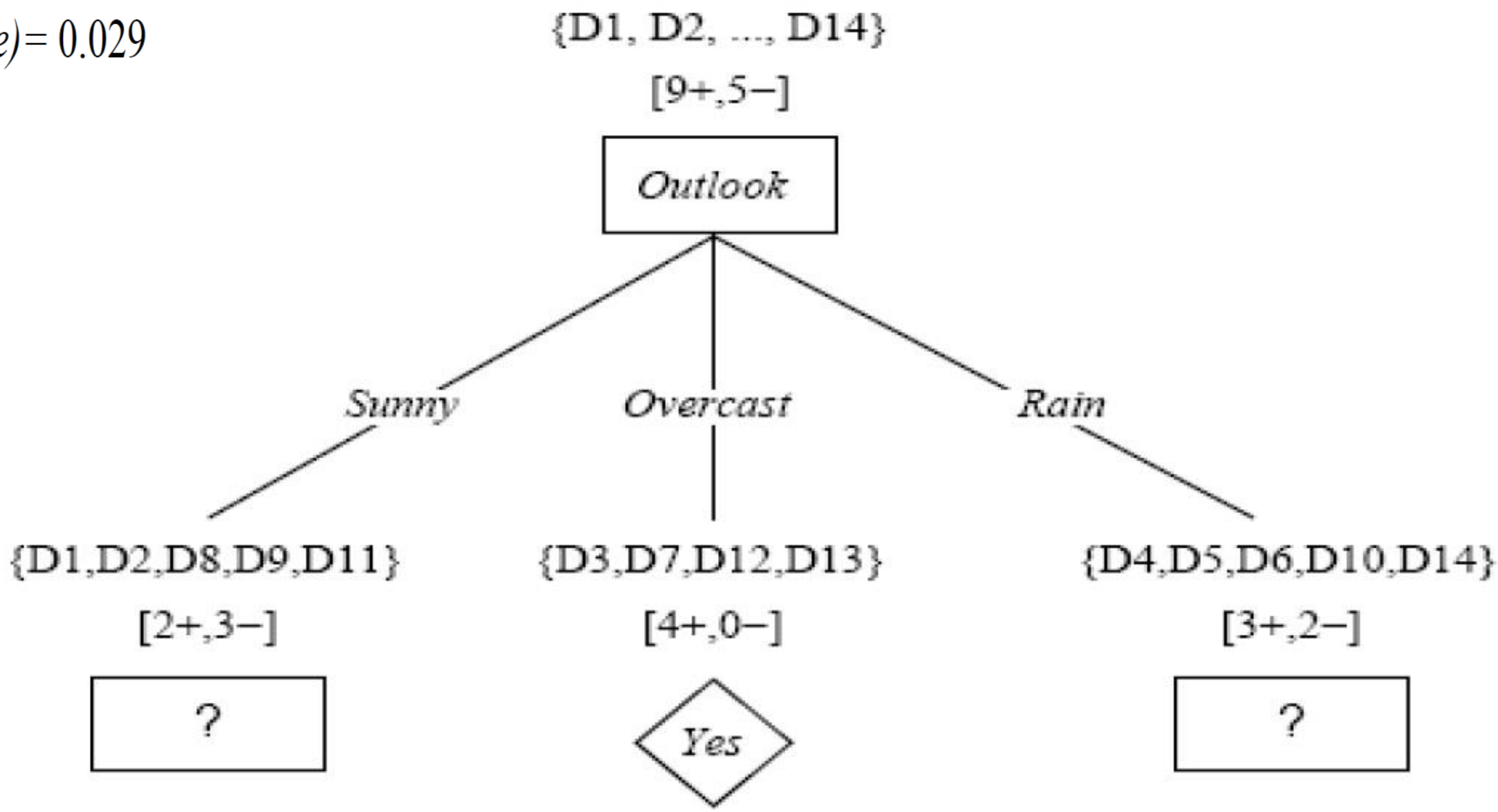
대표적인 알고리즘이 ID3

Gain (S ,A)–The expected reduction in entropy caused by knowing the value of attribute *A. S is a set of cases*,

Entropy: Entropy specifies the minimum # of bits of information needed to encode the classification of an arbitrary member of *S* ➔ *amount of information or level of surprise.*

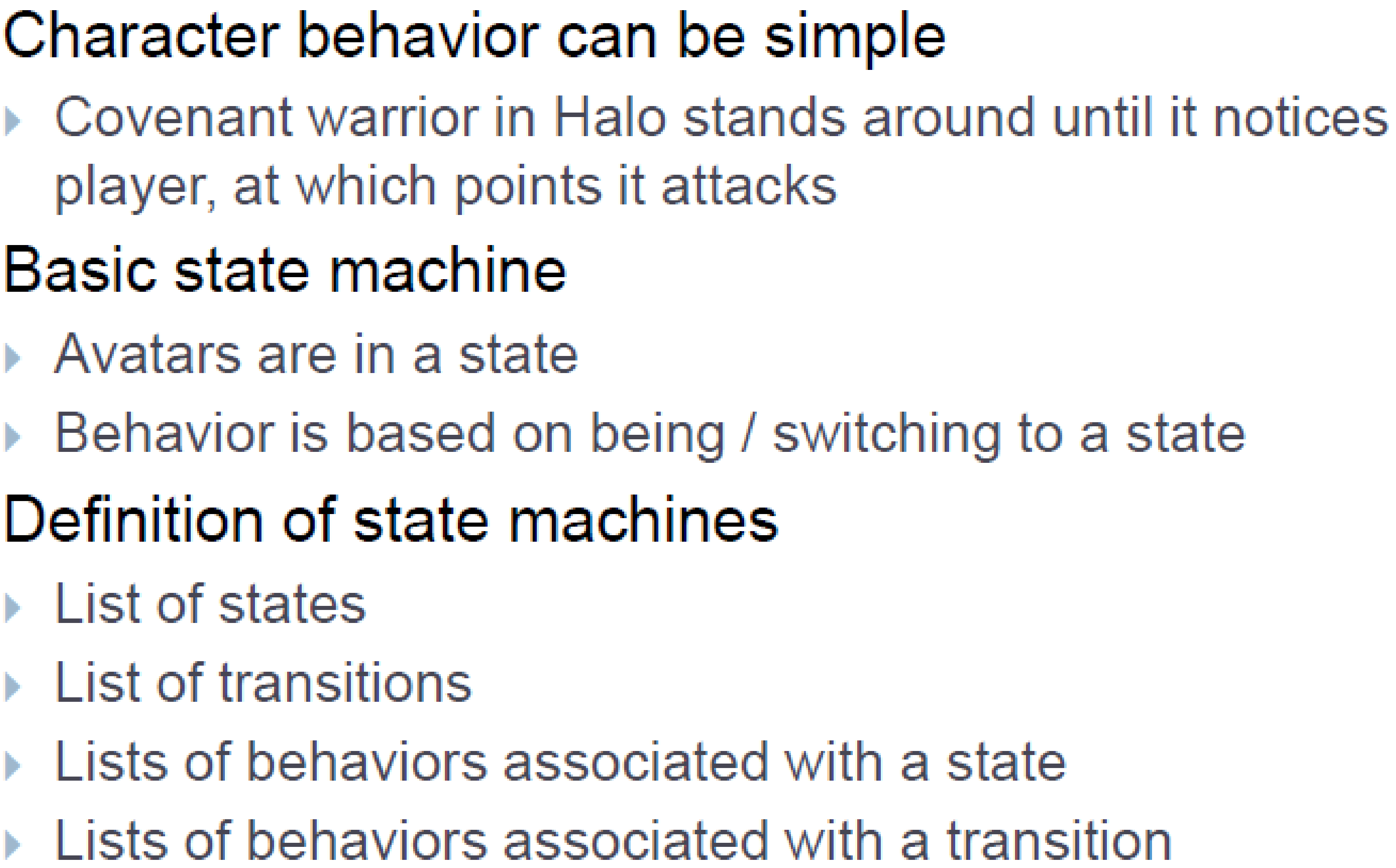
- *Gain(S, Outlook)*= 0.246 🡐 selected as root attribute
- *Gain(S, Humidity)*= 0.151
- *Gain(S, Wind)*= 0.048
- *Gain(S, Temperature)*= 0.029

음성없음



Finite State Machines

State Machines



Character behavior can be simple

- Covenant warrior in Halo stands around until it notices player, at which points it attacks

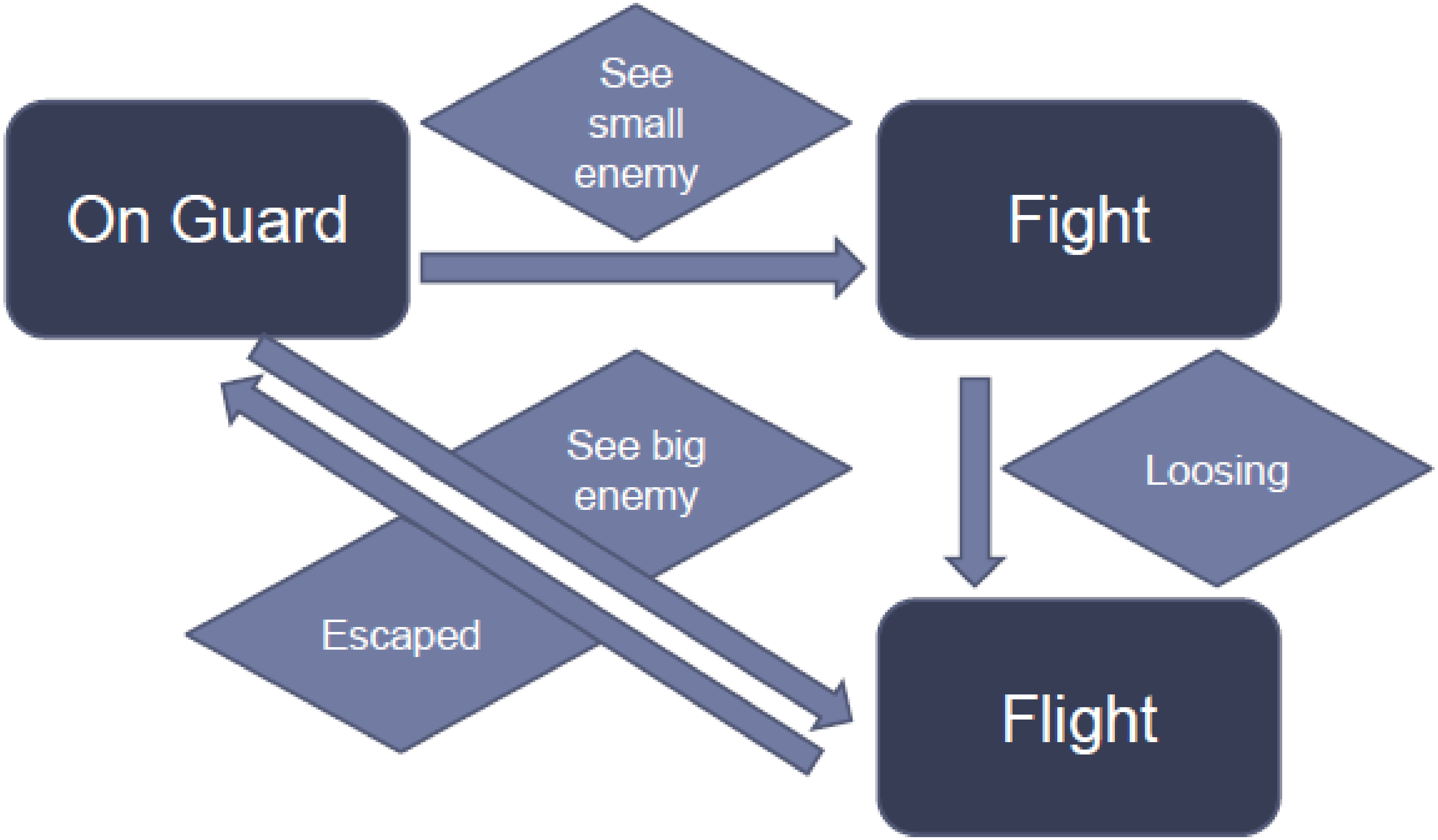
Basic state machine

- Avatars are in a state
- Behavior is based on being / switching to a state

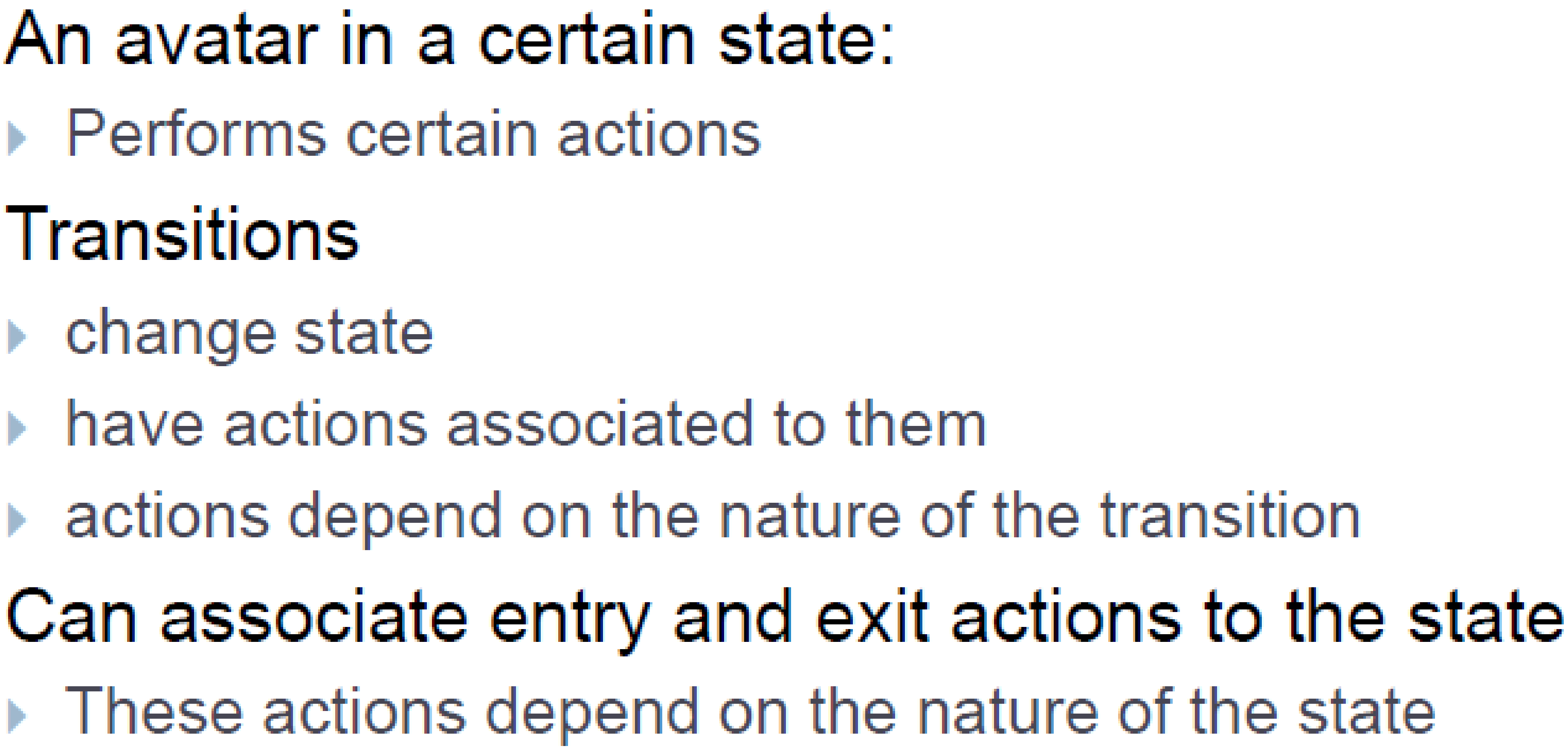
Definition of state machines

- List of states
- List of transitions
- Lists of behaviors associated with a state
- Lists of behaviors associated with a transition

Simple Example



State Machines



An avatar in a certain state:

- Performs certain actions

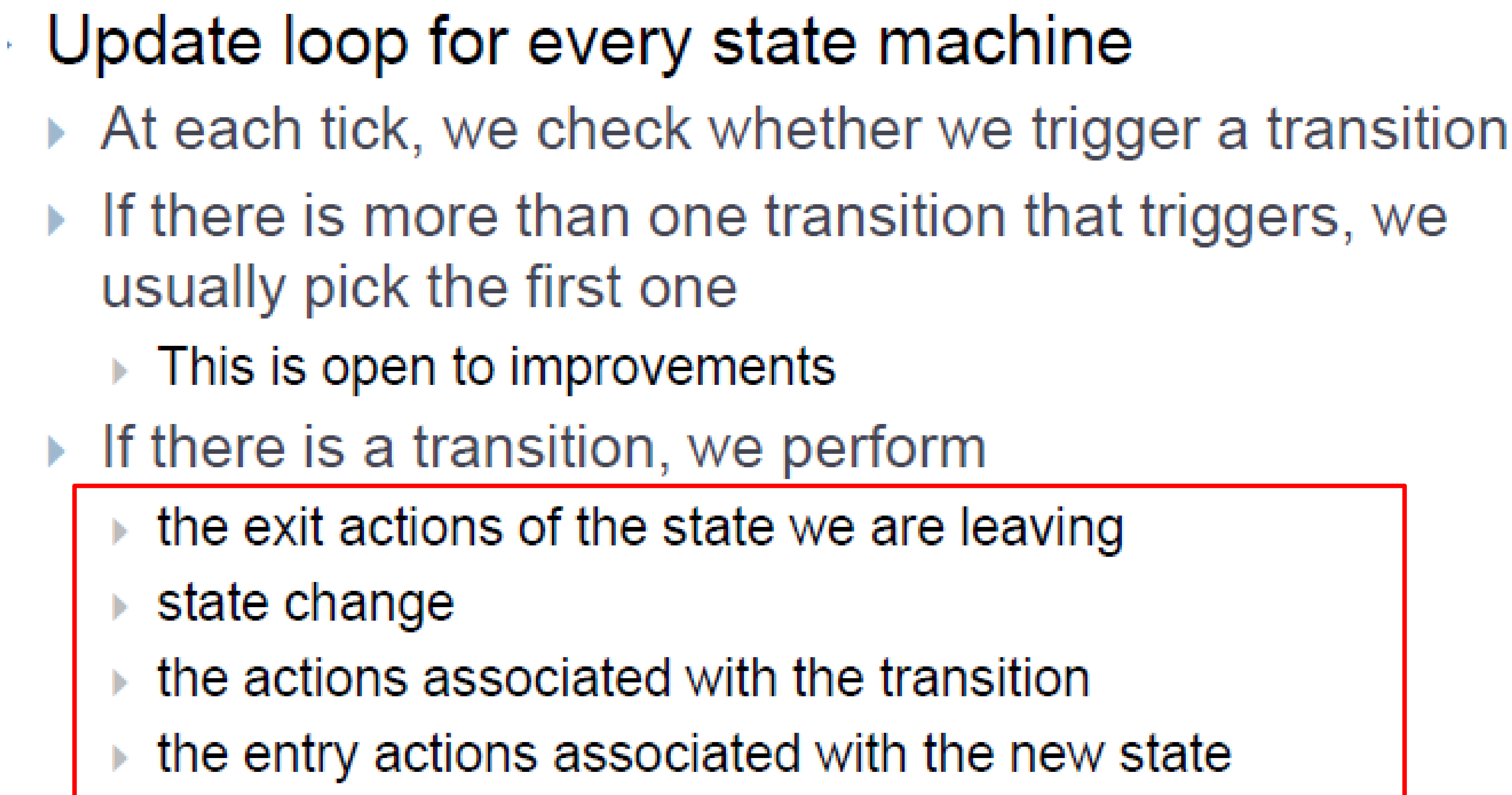
Transitions

- change state
- have actions associated to them
- actions depend on the nature of the transition

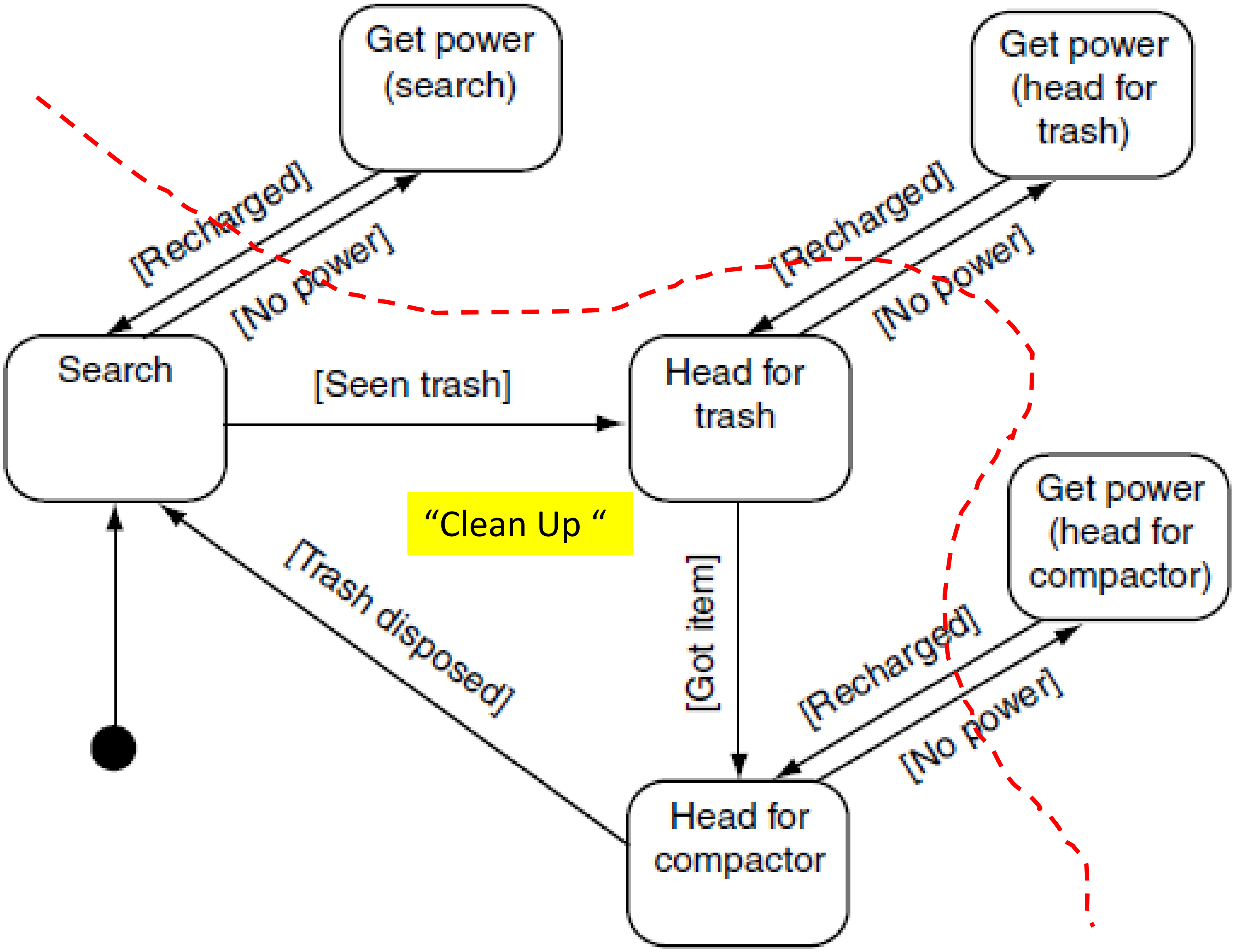
Can associate entry and exit actions to the state

- These actions depend on the nature of the state

State Machines

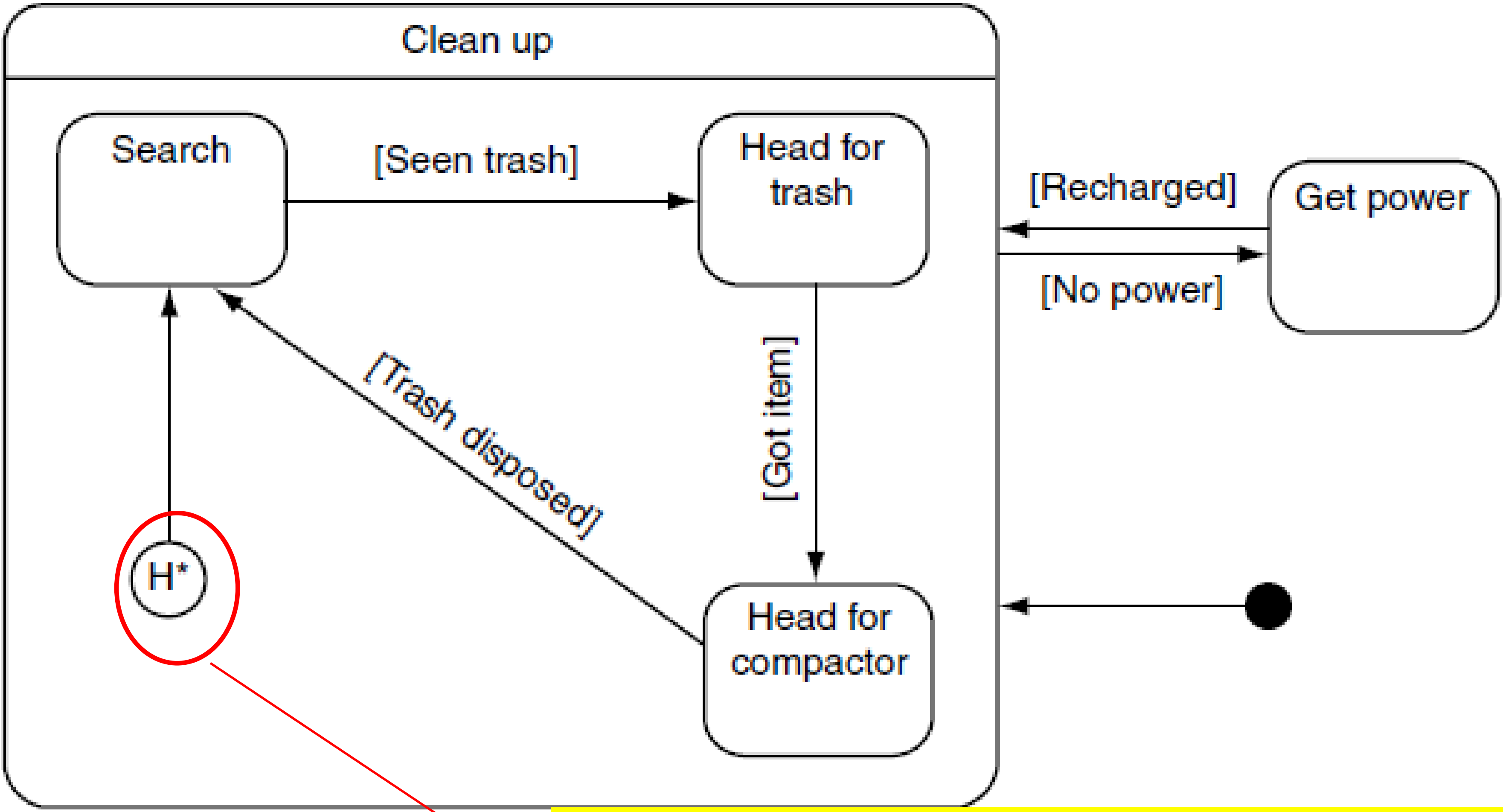


- Update loop for every state machine
  - At each tick, we check whether we trigger a transition
  - If there is more than one transition that triggers, we usually pick the first one
    - This is open to improvements
  - If there is a transition, we perform
    - the exit actions of the state we are leaving
    - state change
    - the actions associated with the transition
    - the entry actions associated with the new state



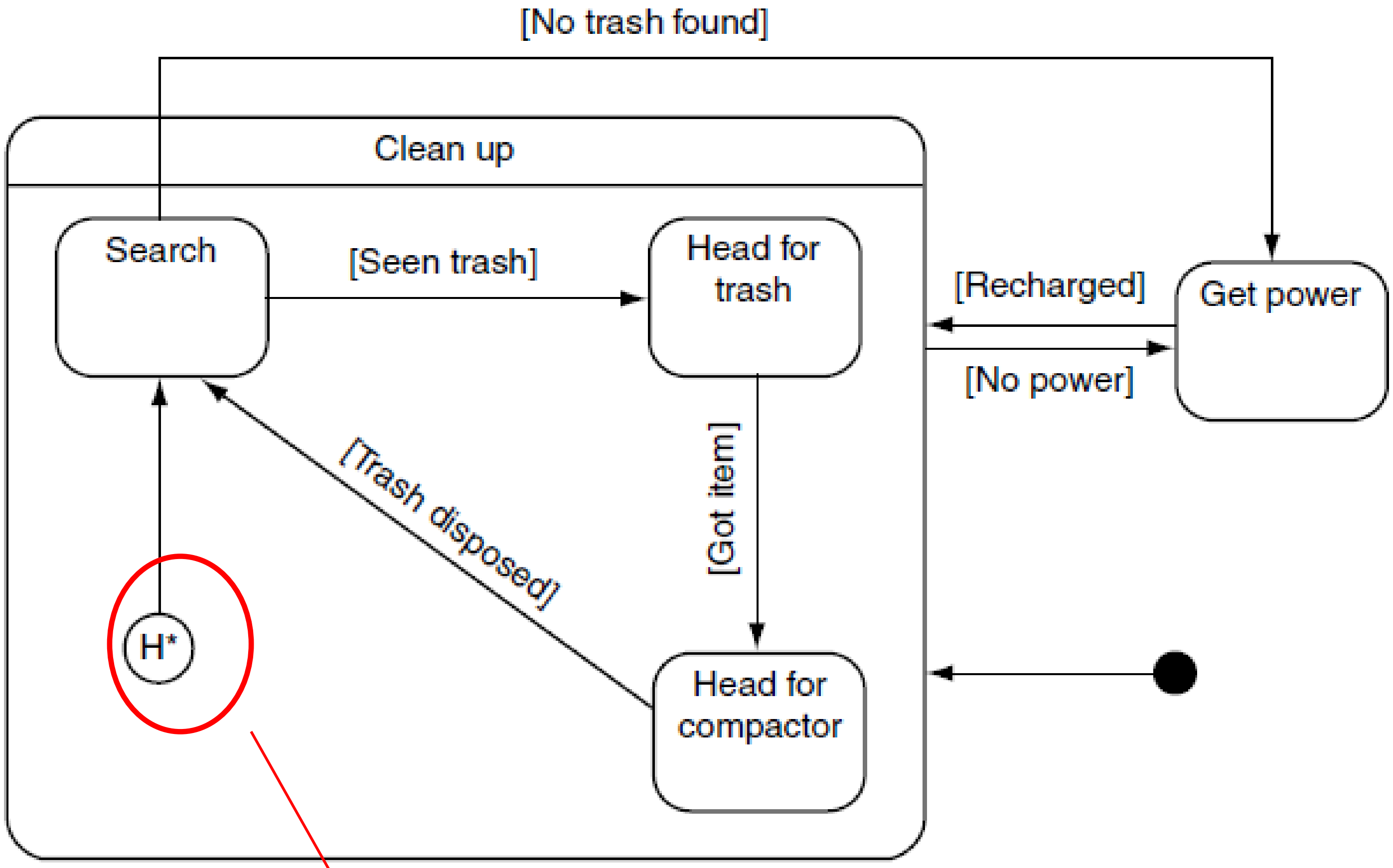
An alarm mechanism in a standard state machine

Search, “Head for trash”, “Head for compactor” 는 “Clean Up “ state의 sub state 들 임.



Search 가 “Clean Up” state 내의 시작 sub-state 임을 나타냄

A hierarchical state machine for the robot



A hierarchical state machine with a cross-hierarchy transition

Search 가 “Clean Up” state 내의 시작 sub-state 임을 나타냄

Behavior Trees

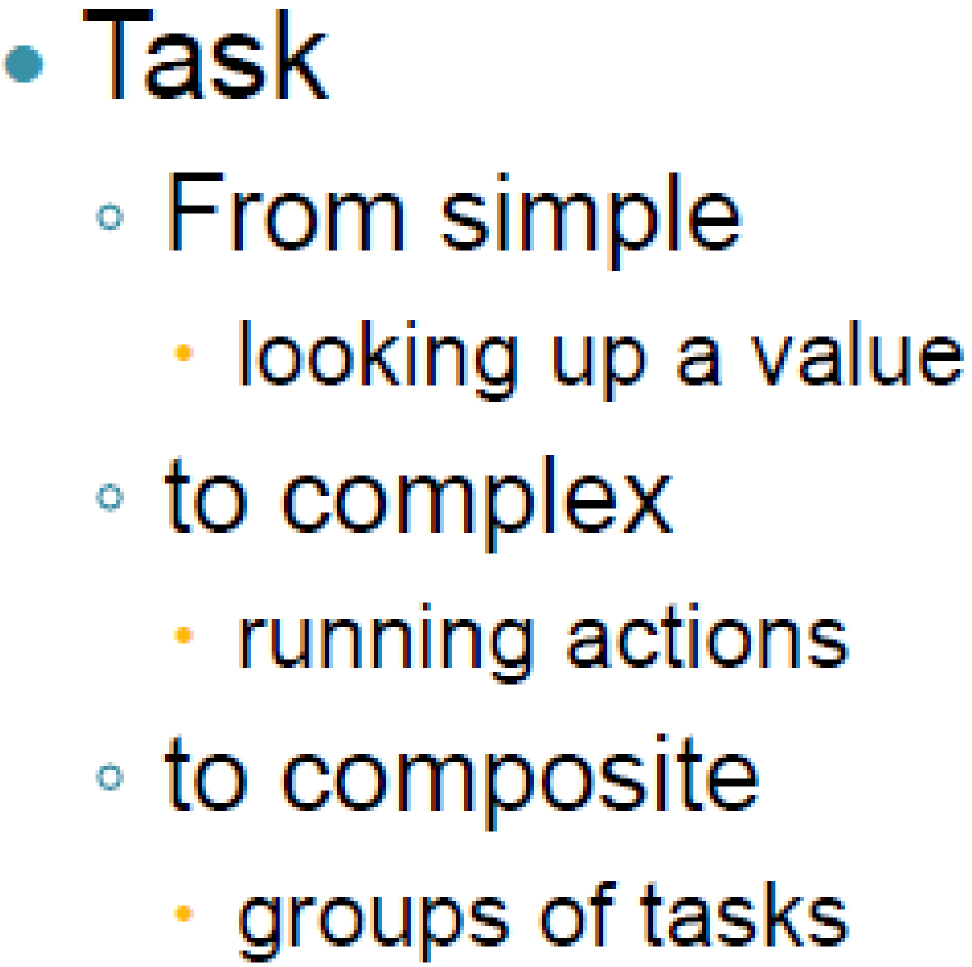
Behavior Trees

Becomes important since Halo 2 (2004)

Synthesis of a number of techniques

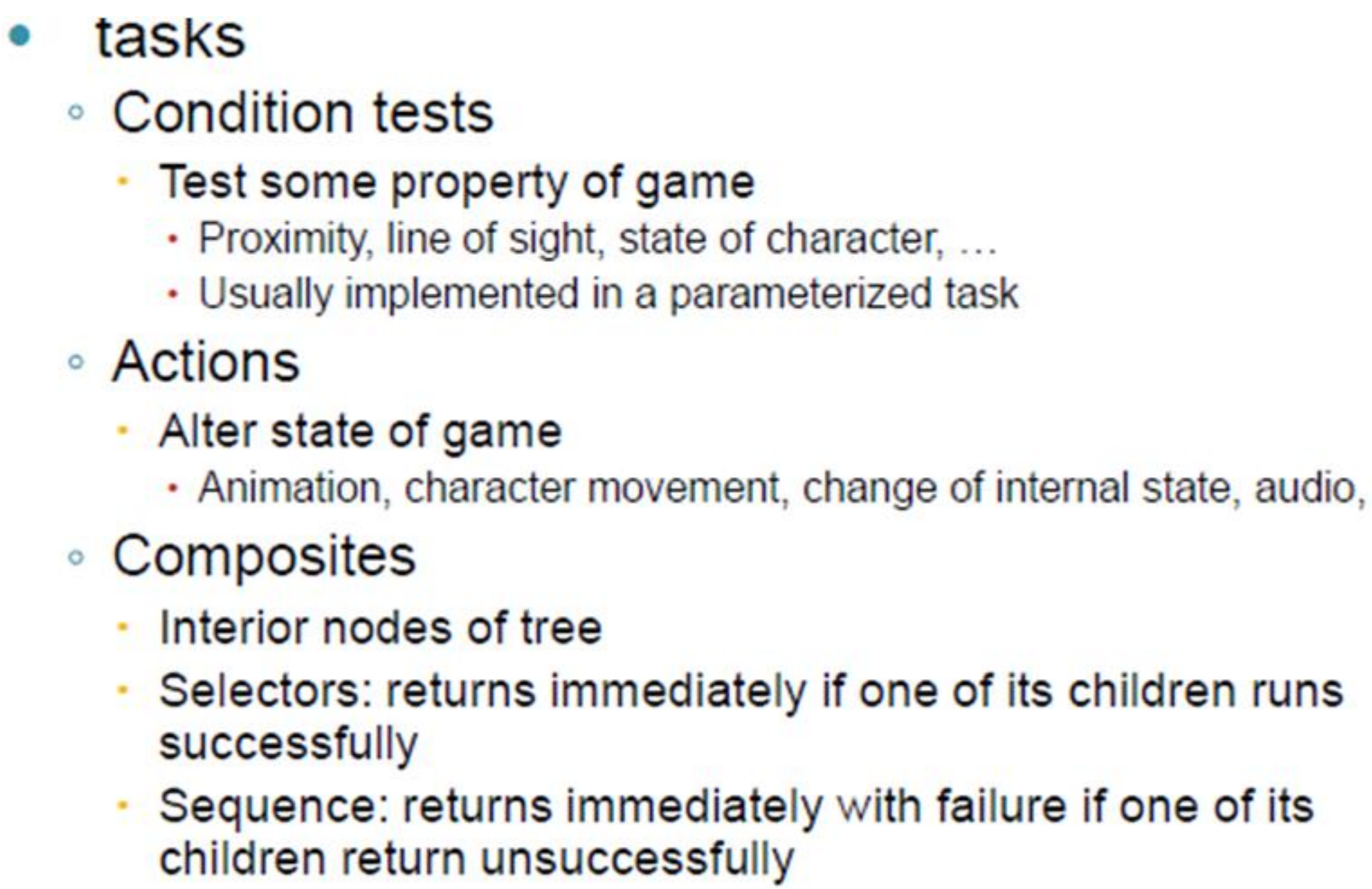
Tree Editor (GUI based) ➔
level designers, technical artists
can author AI,

Behavior Trees

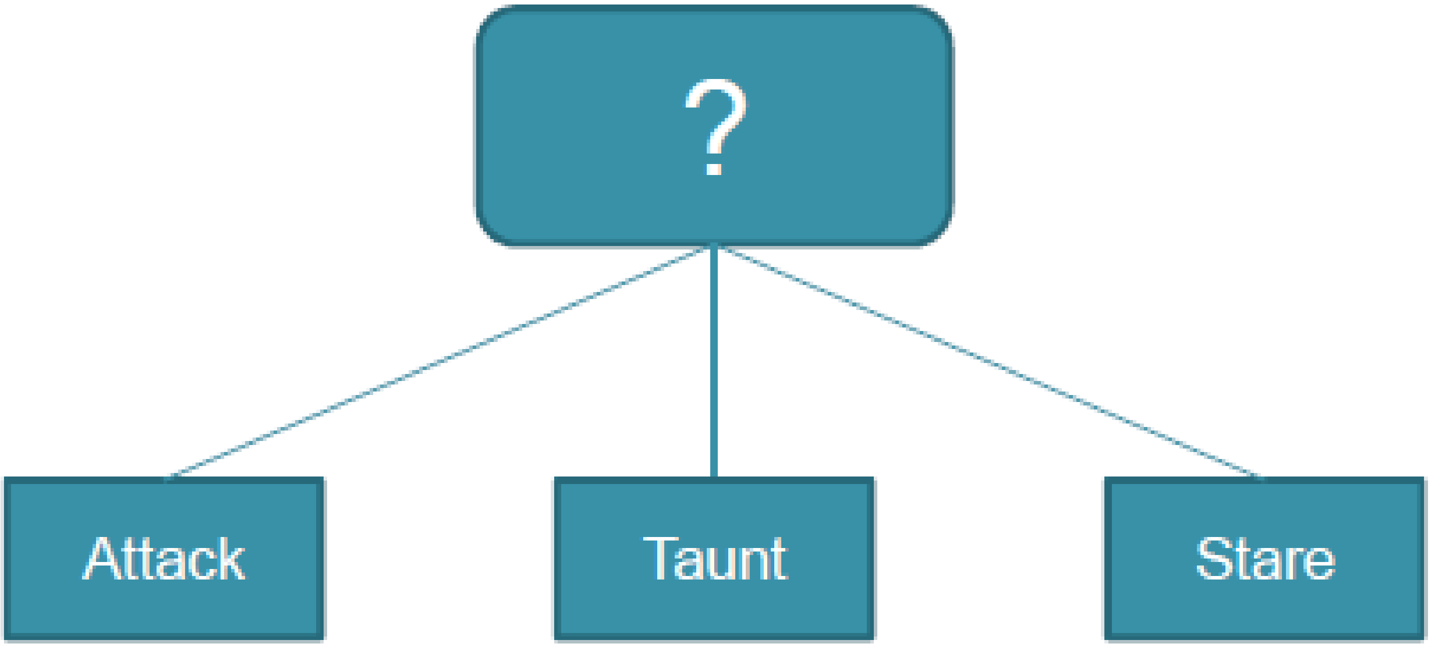


- Task
  - From simple
    - looking up a value
  - to complex
    - running actions
  - to composite
    - groups of tasks

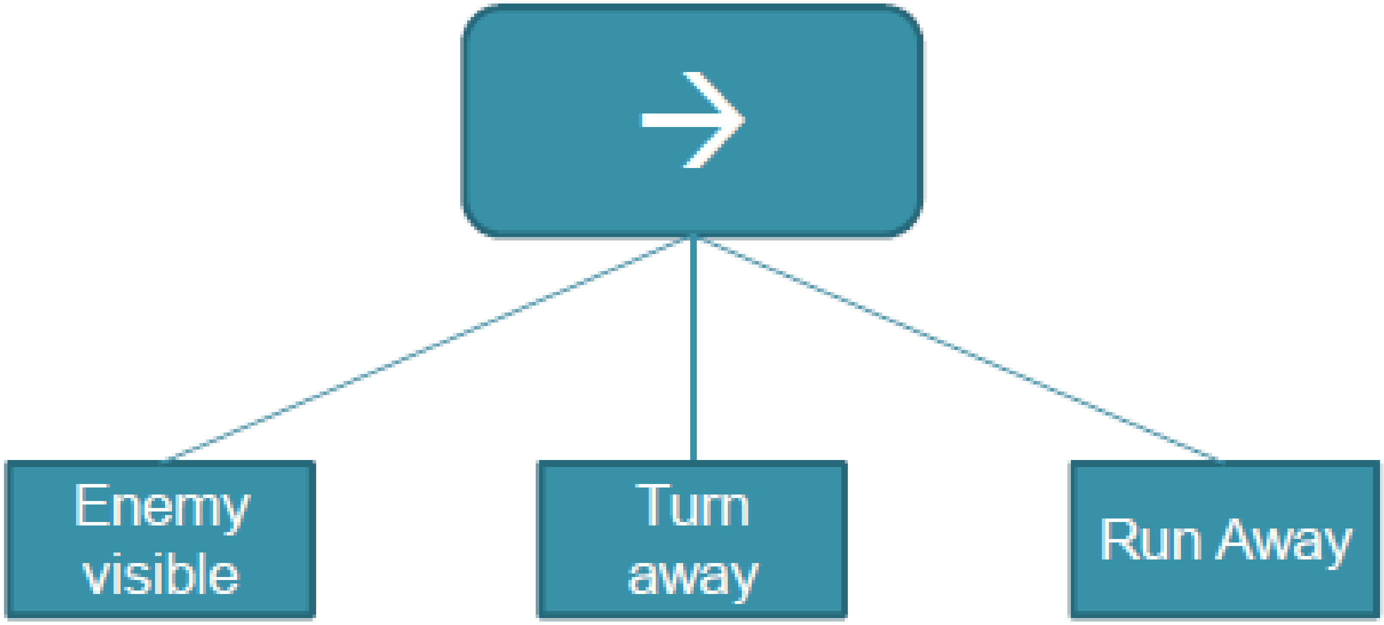
Behavior Trees



- tasks
  - Condition tests
    - Test some property of game
      - Proximity, line of sight, state of character, …
      - Usually implemented in a parameterized task
  - Actions
    - Alter state of game
      - Animation, character movement, change of internal state, audio,
  - Composites
    - Interior nodes of tree
    - Selectors: returns immediately if one of its children runs successfully
    - Sequence: returns immediately with failure if one of its children return unsuccessfully



Selector Node



→

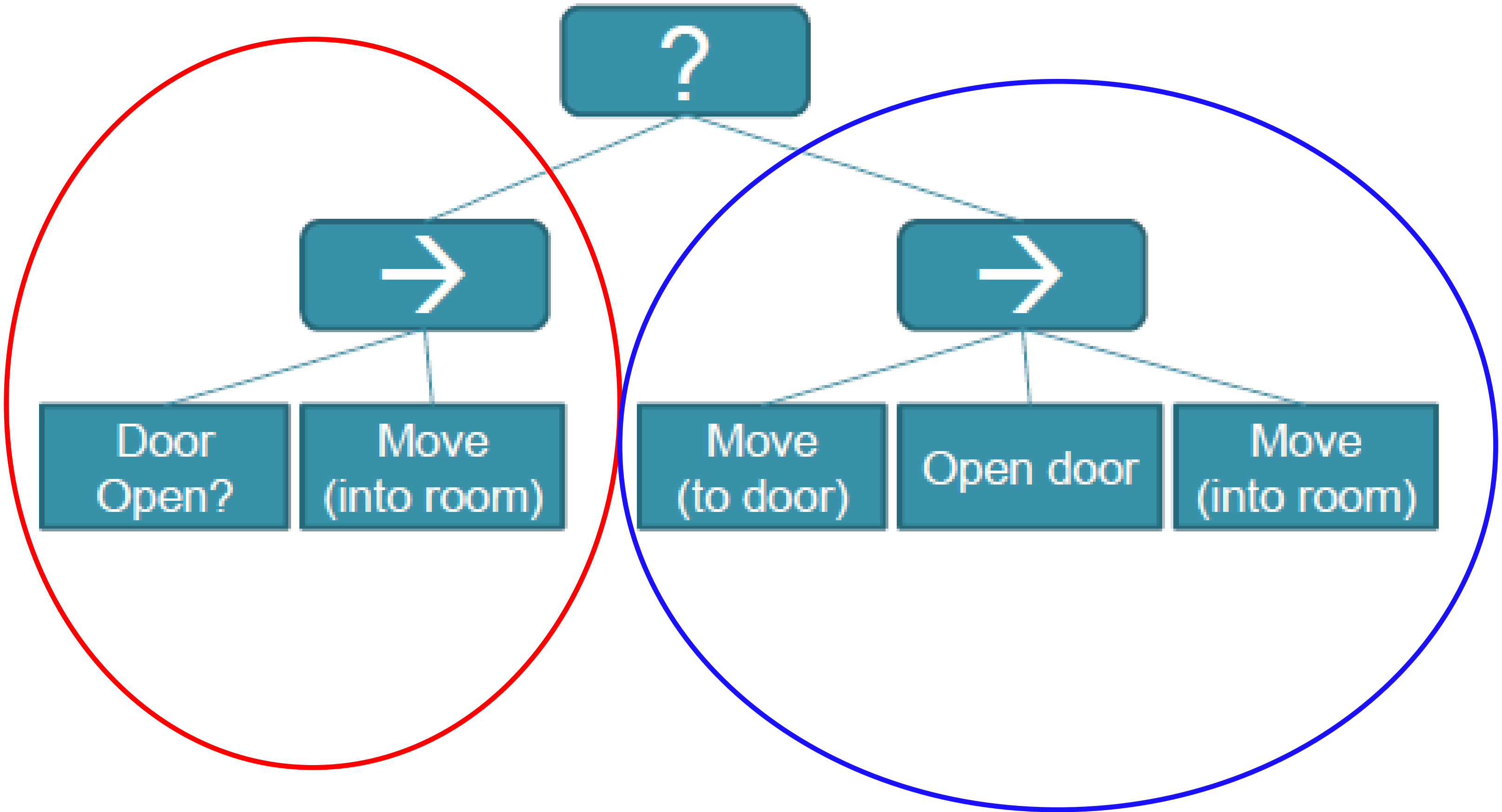
Enemy visible

Turn away

Run Away

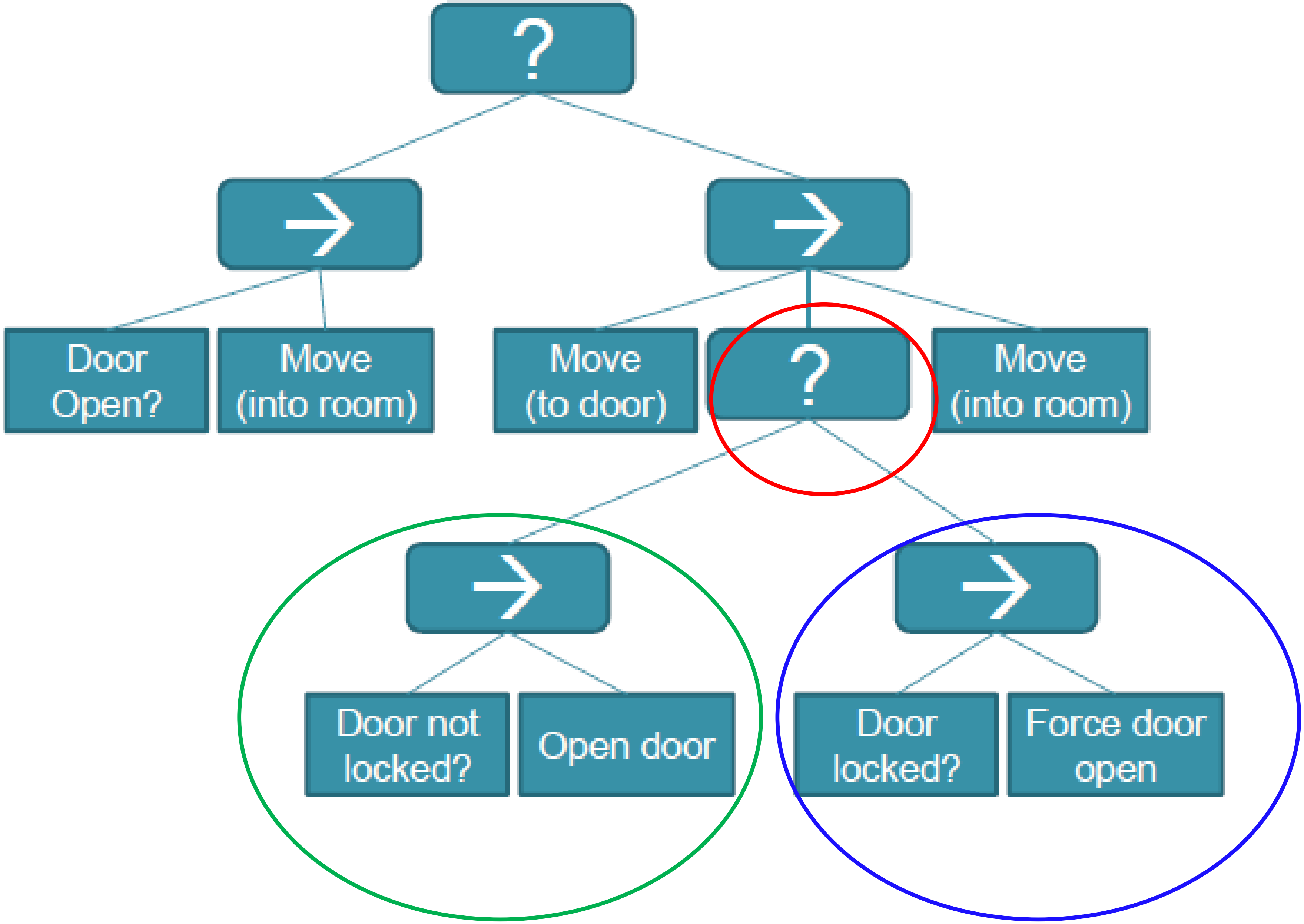
Sequence Node

Example: Entering a room, opening a door if necessary



Condition action in a Sequence is like an if condition:

- If testing fails, Remaining actions are not carried out


?
→
→
Door Open?
Move (into room)
Move (to door)
?
Move (into room)
→
→
Door not locked?
Open door
Door locked?
Force door open

음성 강의 종료