Sensory robotics

Lecture 03.

i.) Behavior based robotics

György Cserey
02.22.2021.

3. Behaviour based robotics

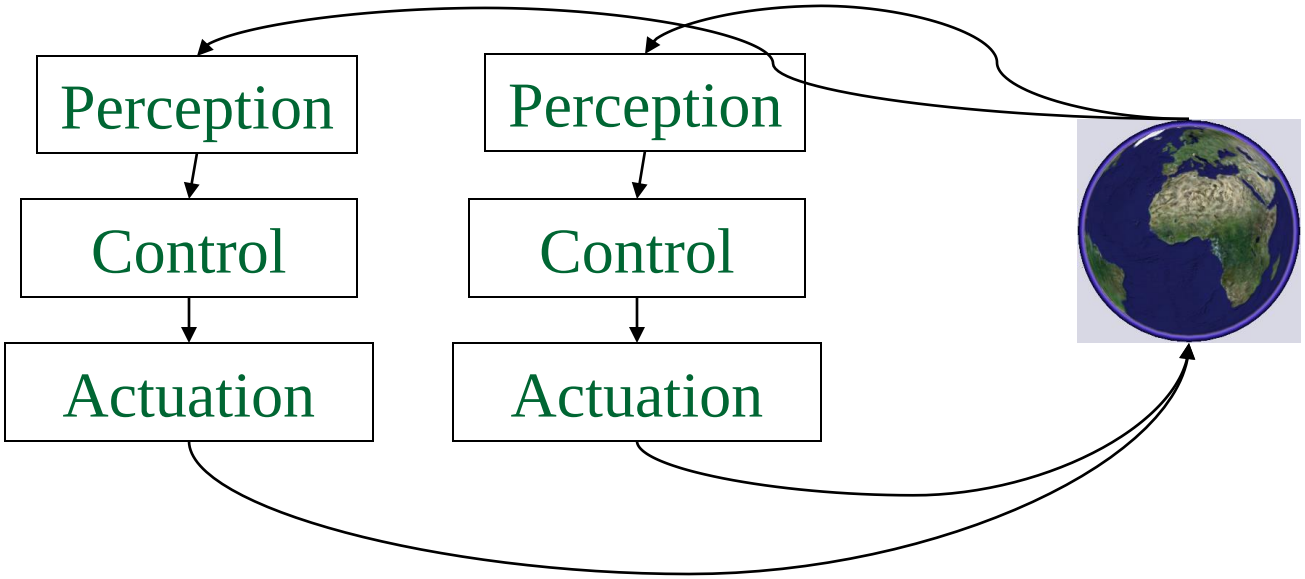
- Behaviour based robotics, introducing behaviour methods in robotics, deliberative and reactive systems, description and coding of behaviours, behaviour design and coordination , design decisions
- Behaviour coordiantion, emergence, fusion and synchronization methods of behaviours

Behavior based robotics and AI

- Contrary to classical AI
- Brooks (1990):
  - „Planning just delays the next actuation step”
  - "Elephants do not play chess"
- The focus is on sensing the environment and intervening (actuating) into it (or feed back)
- Knowledge base and planning is less important

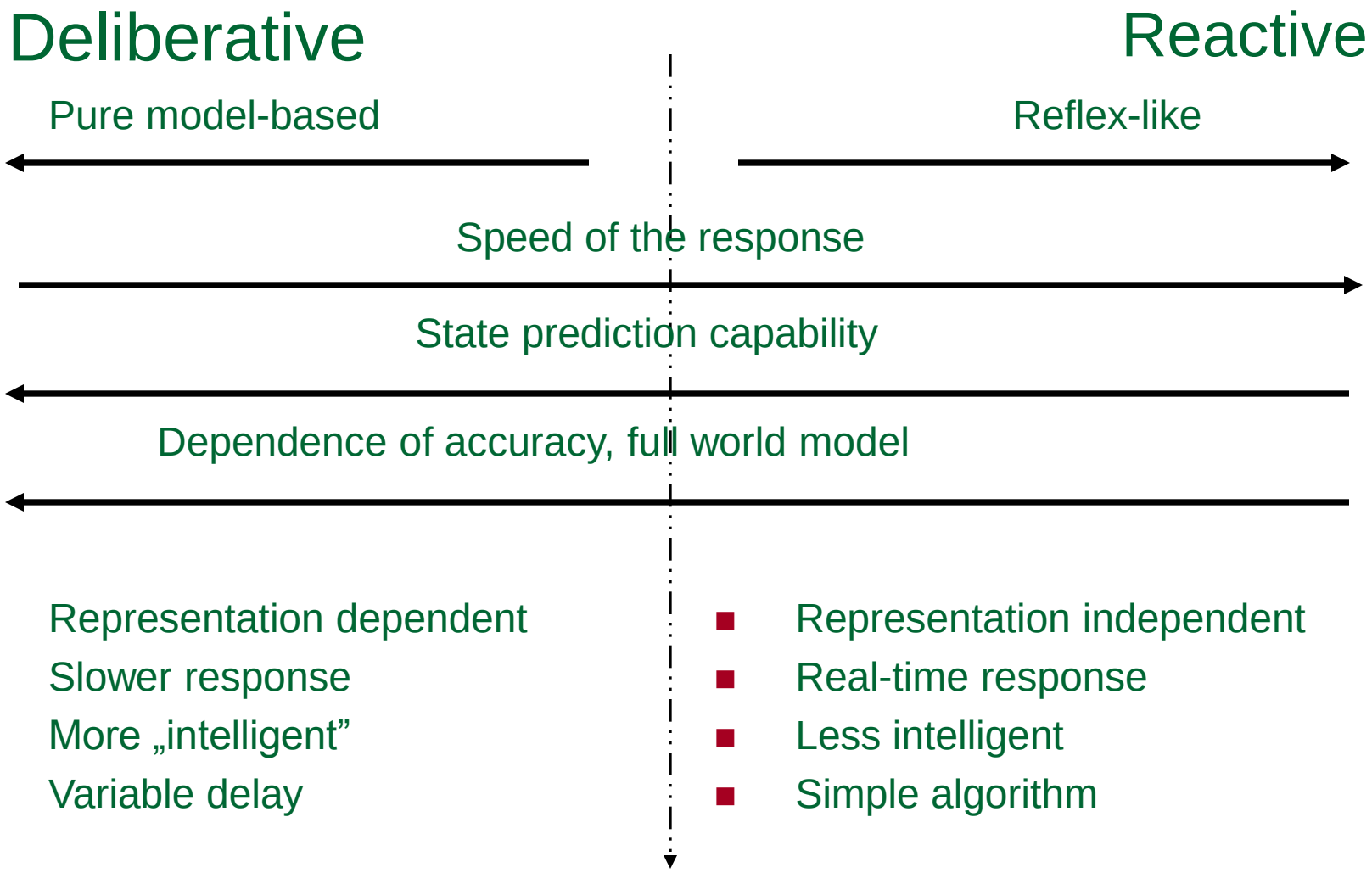
Biological paradigm

"The real world is the best model, we should not build another one."



The results show that this approach is often successful.

Deliberative vs. reactive control



- Representation dependent
- Slower response
- More „intelligent”
- Variable delay

- Representation independent
- Real-time response
- Less intelligent
- Simple algorithm

Deliberative control

- General properties
  - Hierarchical
  - Functionality can be identified
  - The communication and control can be calculated and predetermined
  - Higher levels provide subtasks for lower levels
  - The field of view of the planning varies in the hierarchy levels
  - Symbolic representation based
- It fits well organized and predictable environments (eg. production lines, industrial)

Deliberative systems

- „sensing→ planning→ actuation” model
- Basically sequential
- Search is needed for planning, which is slow
- World model is needed for search
- The drawback of world models is that they require frequent updating
- The planning and search take a long time

Reactive systems

- behavior = mapping of the sensor signals into movement or actuation patterns, which perform tasks

  ie = response of the stimulus

- Sensing and actuation is closely linked in reactive systems, there is no abstract inner representation.
- Thus, the reactive paradigm:

Reactive systems

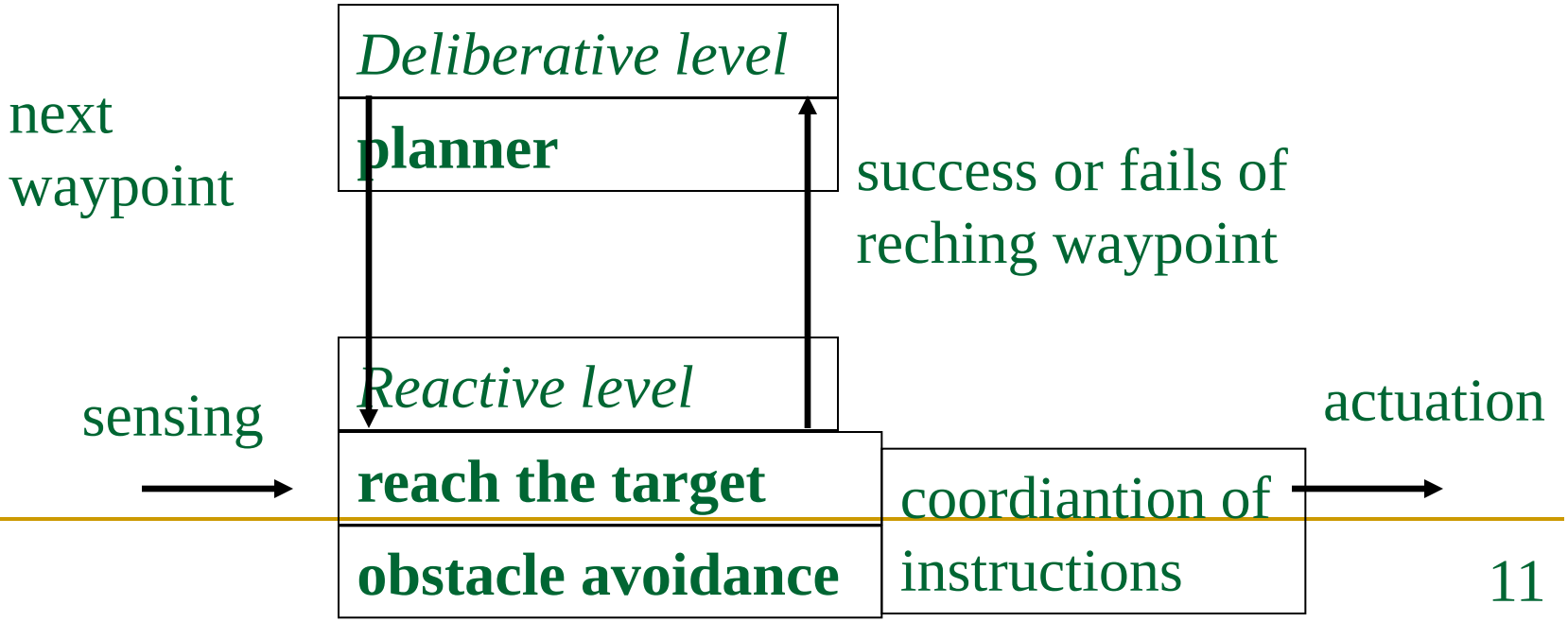
- The features of sensing-actuation (stimulus-response)
  - Parallel by its nature
  - Few states
  - No memory
  - Very quick response
  - It is not able to plan
  - It is not able to learn

Hybrid systems

- The combination of the two extremes
  - Reactive system is below
  - Deliberative system is on top
  - There can be internal levels
  - It is often called three-level system
  - The levels must work simultaneously
  - Various representations and time-scale on the levels

Example (hybrid systems)

- Deliberative behavior: trail planning, navigation
  - Global representation: Trajectory, trail, waypoints,
- Reactive behavior: obstacle avoidance
  - Local representation: distance from the obstacles and direction

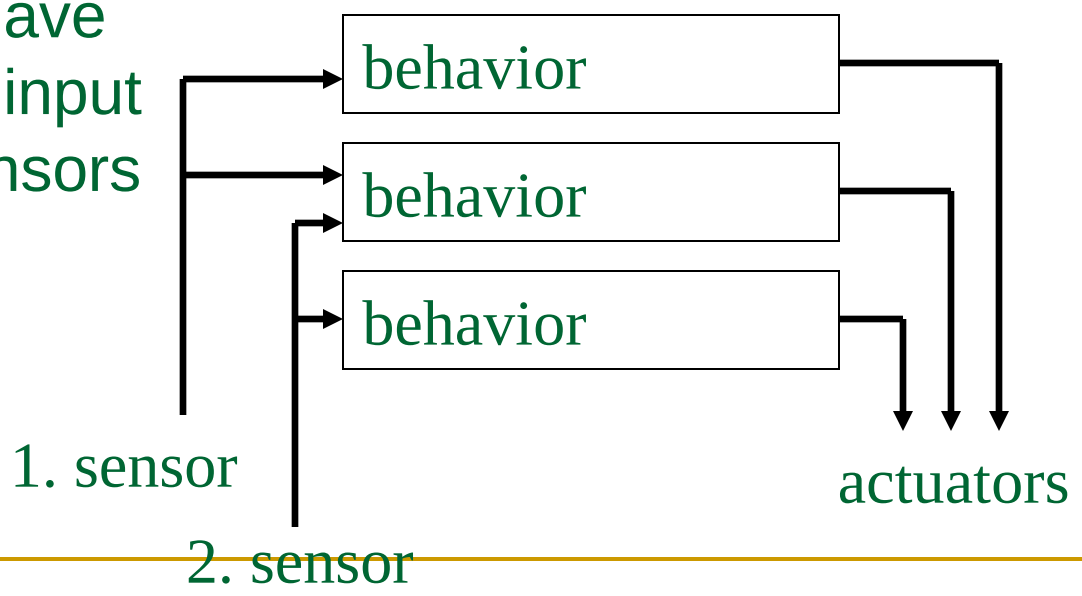


Behavior based systems

- Alternatives of hybrid systems
- The ability of deliberative and reactive actuation as well
- There are no intermediate levels
- Uniform and consistent representation in the entire system => parallel behaviors
- Solve the sequence and timing problem

Reactive behoviour based systems

- The behaviors are direct mapping between sensory input and motor movement in order to achieve a specified goal.
- Local perception
- Different behaviors can use the same senory input
- A behavior can have multiple sensory input from different sensors



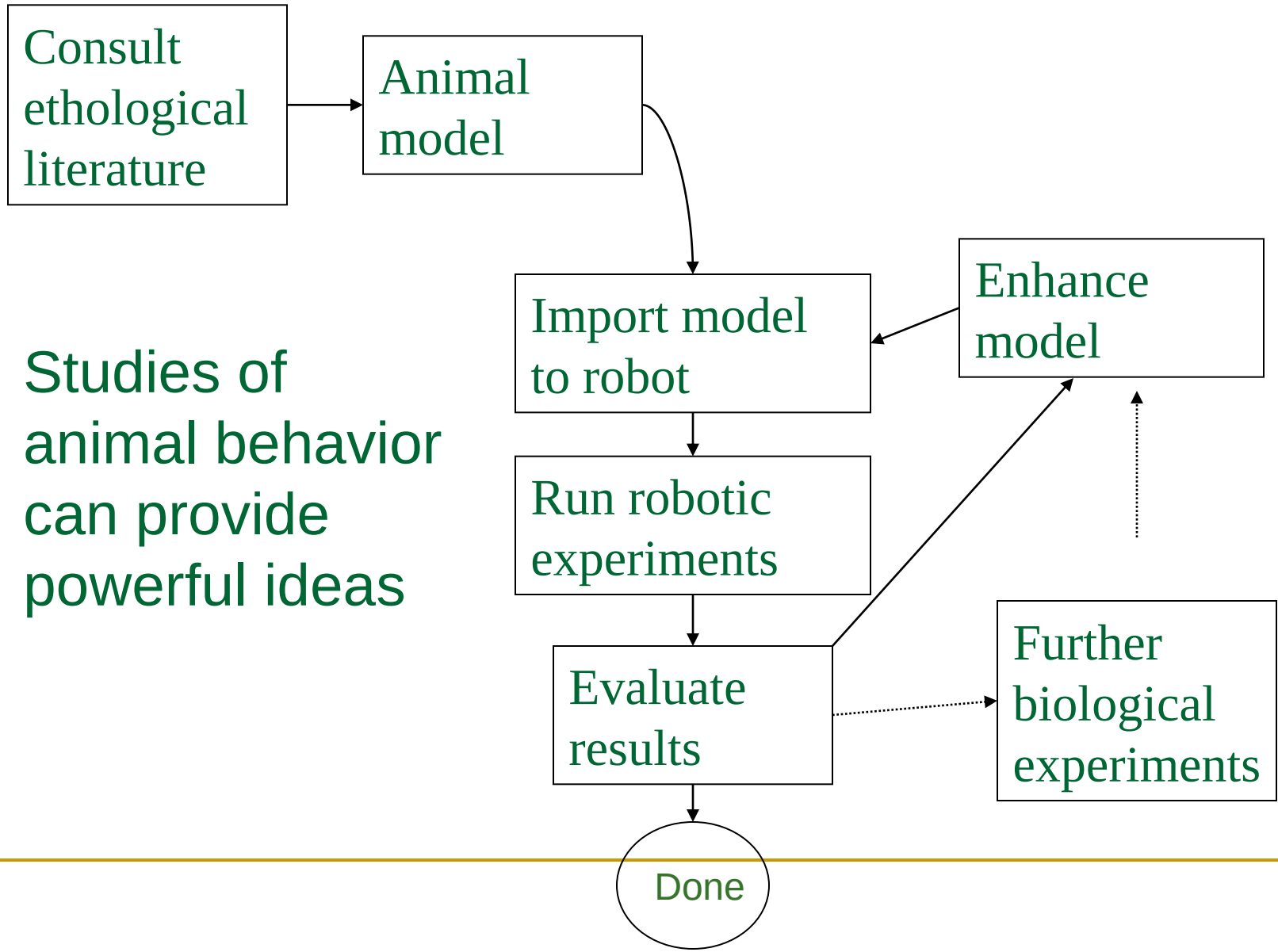
Features of reactive behavior based robot systems

- The behaviors are the basic building blocks of the action and actuation of the robot
  - Tipically, the behaviors are simple sensor-motor / sensor-actuation pairs
- Usage of abstract representation (or world model) is not typical
- Its model is often based on biological behavior
- They are modular by their nature, which is beneficial for software design considerations

Features of behavior based control

- Situatedness: the robot operates in the real world, there is no abstact representation
- Embodiment: the robot has a physical presence (a body)
- Emergence: the intelligence of the robot arises from the interactions with the environment
- Grounding in reality: to avoid symbol grounding problem
- Ecological dynamics: not typical
- It is not known whether the behavior-based control's ability will ever achieve human intelligence

Etologically guided design

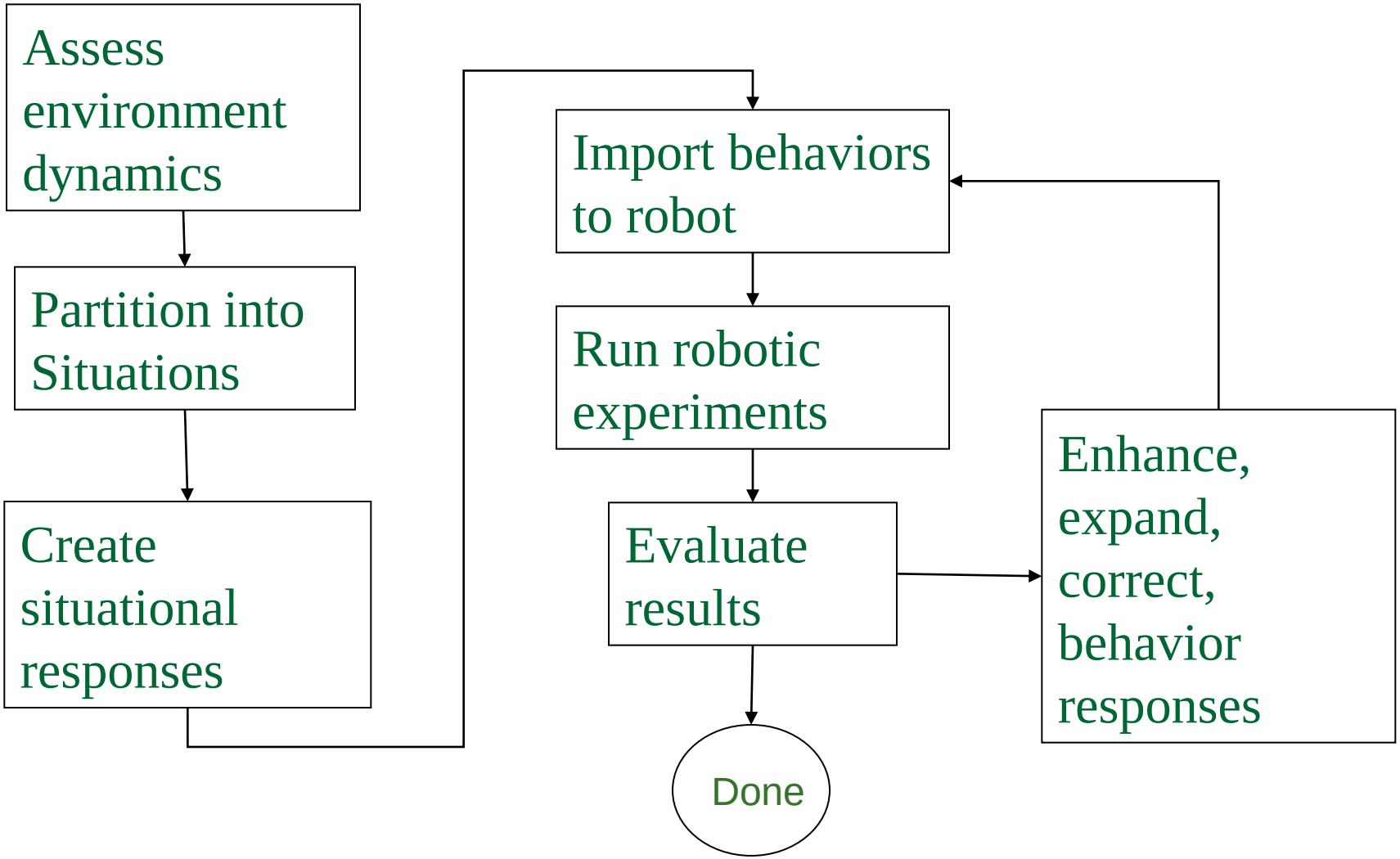


Studies of animal behavior can provide powerful ideas

Situated activity-based design

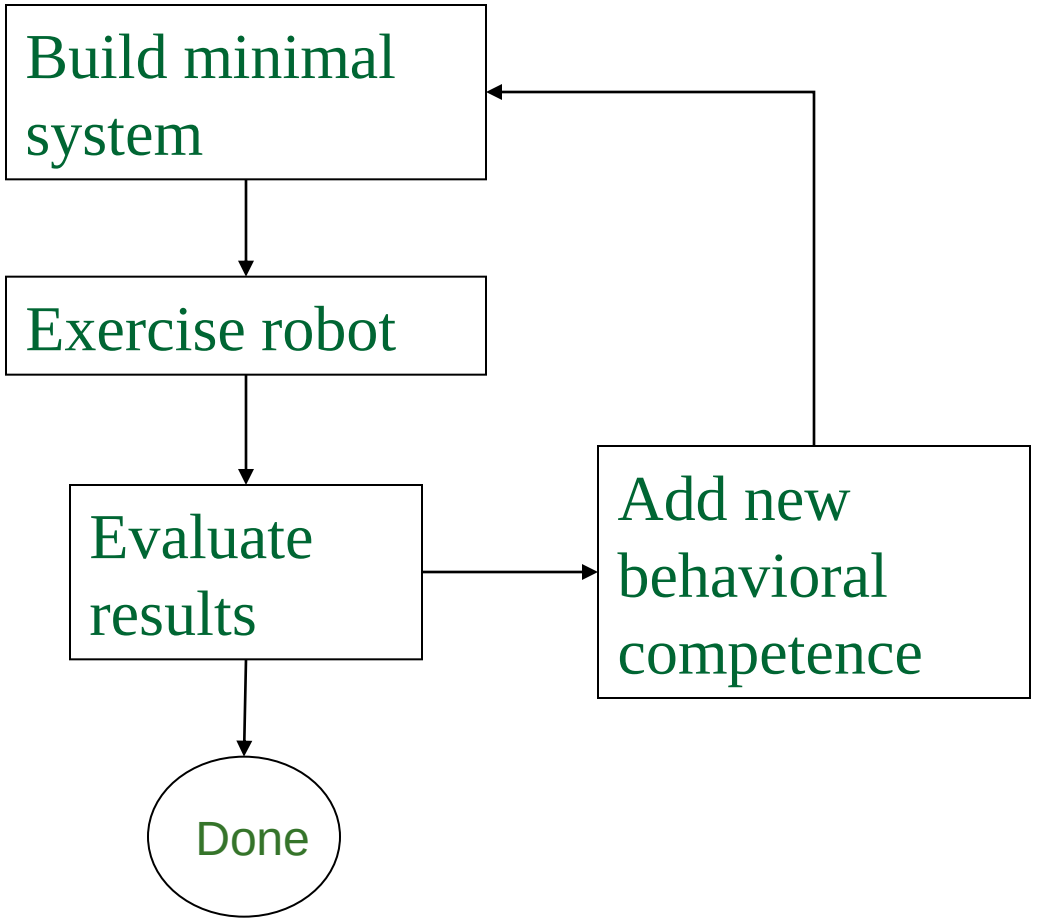
- Situated activity: a robot's actions are predicted upon the situations in which it finds itself.
- Hence the perception problem is reduced to recognizing what situations the robot is in and then choosing one action to undertake.
- If a robot is in a new situation, it selects a new and more appropriate action.
- Desinging a robot based on this methodology requires a solid understanding of the relationship between the robot and its environment.

Situated activity design methodology



Experimentally driven methodology

- Create in a bottom-up manner
- Iterative method



General classification of robot behaviors

- Exploration/directional behaviors
  - Heading based
  - Wandering
- Goal-oriented appetitive behaviors (move towards an attractor)
  - Discrete object, attractor
  - Area attractor
- Aversive/protective behaviors (prevent collision)
  - Avoid stationary objects
  - Elude moving objects (Dodge, escape)
  - Aggression

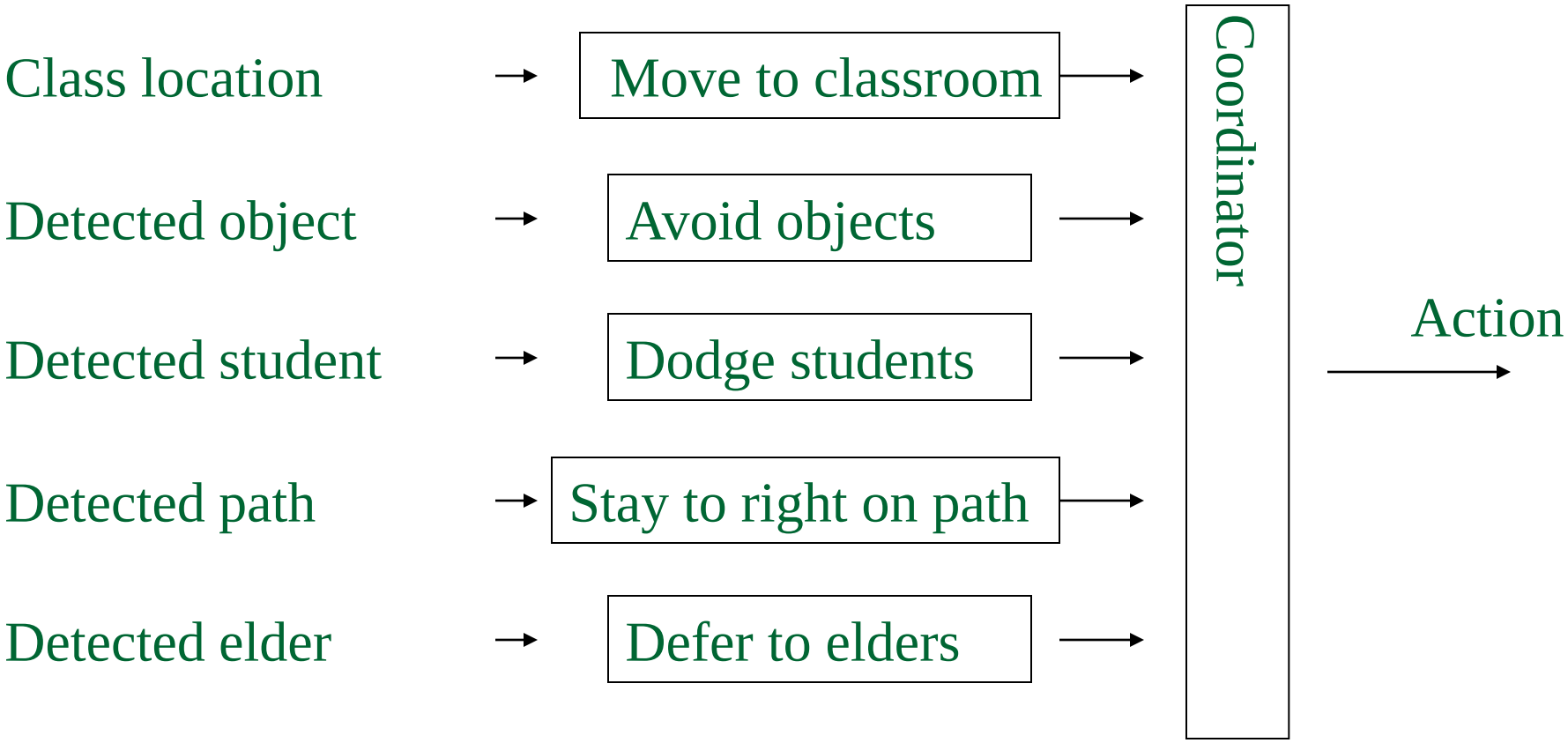
General classification of robot behaviors

- Path following behavious (move on a designated path)
  - Road following / hallway navigation
  - Stirpe following
- Postural behaviors
  - Balance
  - Stability
- Social / cooperative behaviors
  - Sharing
  - Foraging
  - Flocking
- Teleautonomous behaviors (coordinate with a human operator)
  - Influence
  - Behavioral modification

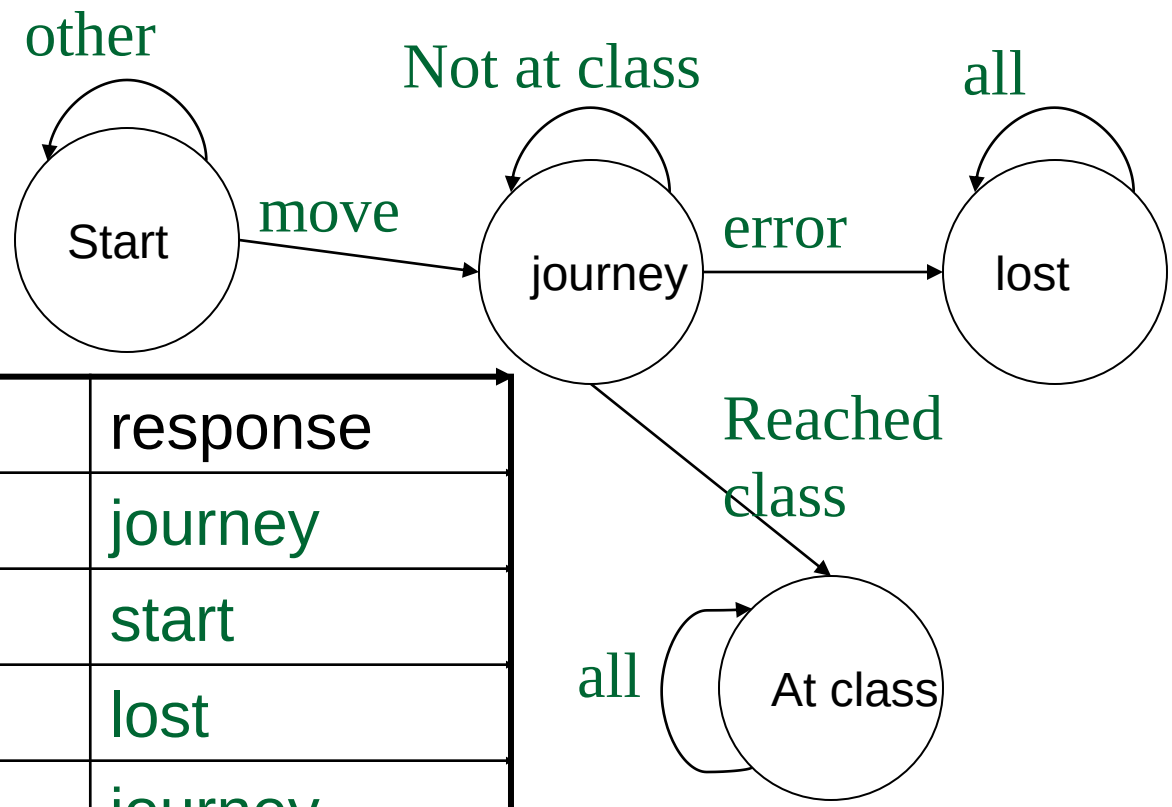
General classification of robot behaviors

- Perceptual behaviors
  - Saccades
  - Visual search
  - Ocular reflexes
- Walking behaviors (for legged robots)
  - Gait control
- Manipulator-specific behaviors (for arm control)
  - Reaching
- Gripper / dextrous hand behaviors (for object acquisition)
  - Grasping
  - Enveloping

Stimulus-response diagrams



Finite State Acceptor diagramm



state	input	response
start	move	journey
start	other	start
journey	error	lost
journey	not at class	journey
journey	reached class	at class
at class	all	at class
lost	all	lost

Behavioral encoding

- Behavioral mapping: the mapping between the stimulus domain and response range
- behavior can be expressed: (S,R,β)

  where:

  S denotes the domain of all interpretable stimuli,

  R denotes the range of possible responses,

  β denotes the mapping, β: S->R

Behavioral encoding

- A behavior can be expressed as a triple: (S,R,β)

  β(s)=r, where:

  β = behavior,

  s = stimuli,

  r = response
- g is a scalar gain value (strength mulitplier), further modifying the magnitude of the overall response r, hence r'=gr
- Encoding can be:
  - Discrete
  - Continuous

R: responses

- R denotes the range of possible responses
- The robot's motor response has two orthogonal components:
- Strength:
  - Denotes the magnitude of the response
  - Velocity, force, orthogonal velocity
- Orientation:
  - Denotes the direction of action for the response (direction, rotational speed)
  - Distancing from a source of negative stimuli, or approaching an attractor

General categories of behavioral mapping

- Null: the stimulus produces no motor response
- Discrete: The stimulus produces a response from an enumerable set of prescribed choices, e.g., turn-right, go-straight, stop, travel-atspeed-5
- Continuous: The stimulus domain produces a motor response that is continuous over **R**'s range

Discrete encoding

- Situated action, β = consists of a finite set of (situation, response) pairs
  - (neares-object < 5, turn-right-10)
- Rule-based systems, with If-Then rules

  If (nearest-object > 20) omega = 0

  Else If (nearest-object-direction< -30) omega = 10

  Else If (nearest-object-direction < 0) omega = 20

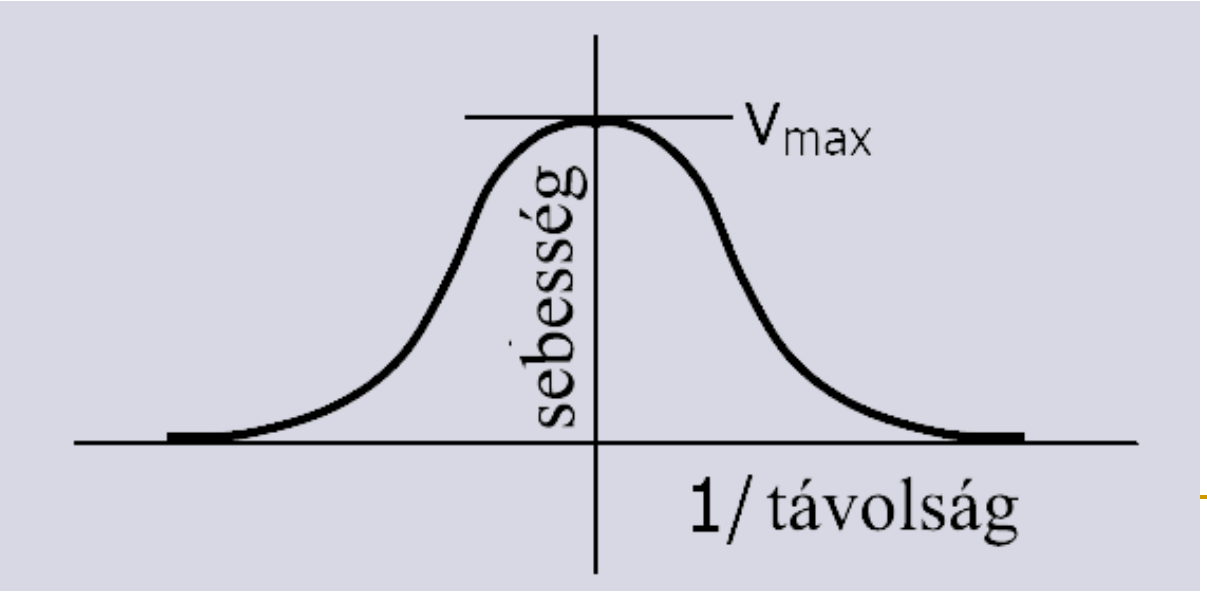
  Else If (nearest-object-direction < 30) omega = -20

  Else omega = -10

Continuous Functional Encoding

- A continuous functional transformation between the sensoral input and behavioral action
- Common example: ~$1/x^2$

  The force is proportional to the 1/distance^2



Sample behaviors

- Movement
  - Movement with constant speed
- Approaching to the target
  - Towards to the target object (eg. light source)
- Stop
  - Decrease its speed near to the object
- Avoidance
  - Obstacle avoidance

Basis for robotic behavior

- The primary questions:
  - What are the right behavioral building blocks for robotic systems?
  - What really is a primitive behavior?
  - How are these behaviors effectively coordinated?
  - How are these behaviors grounded to sensors and actuators?
- There are currently no universally agreed-upon answers to these questions
- The ultimate judge is the appropriateness of the robotic response to a given task and environment.

Summary

- Robotic behaviors generate a motor response from a given perceptual stimulus
- Purely reactive systems avoid the use of explicit representational knowledge or world model
- Three design paradigms
  - ethologically guided/constrained design
  - Situated activity based design
  - experimentally driven design
- behaviors can be described in three different ways
  - SR diagrams (stimulus-response diagram)
  - Functional notation
  - FSA diagrams (Finite State Acceptor diagramm)
- Functional notation: (S,R,β) represented as triples

Behavioral coordination

- If we have more than behaviors how can we combine and cooperate them?
- Competitive (decision is required)
  - Eg.: one of the outputs of the given behaviors will be chosen
- Cooperative
  - Mixture of the outputs, which fits to the overal purpose of the robot
  - Eg.: vector addition in potential field

„Emergence”

- „ 'the appearance of novel properties in whole systems” (Moravec 1988)
- „ Global functionality emerges from the parallel interaction of local behaviors” (Steels 1990)
- „ Intelligence emerges from the interaction of the components of the system” (Brooks 1991)
- …
- In our case, emergence is a property of a collection of interacting components.

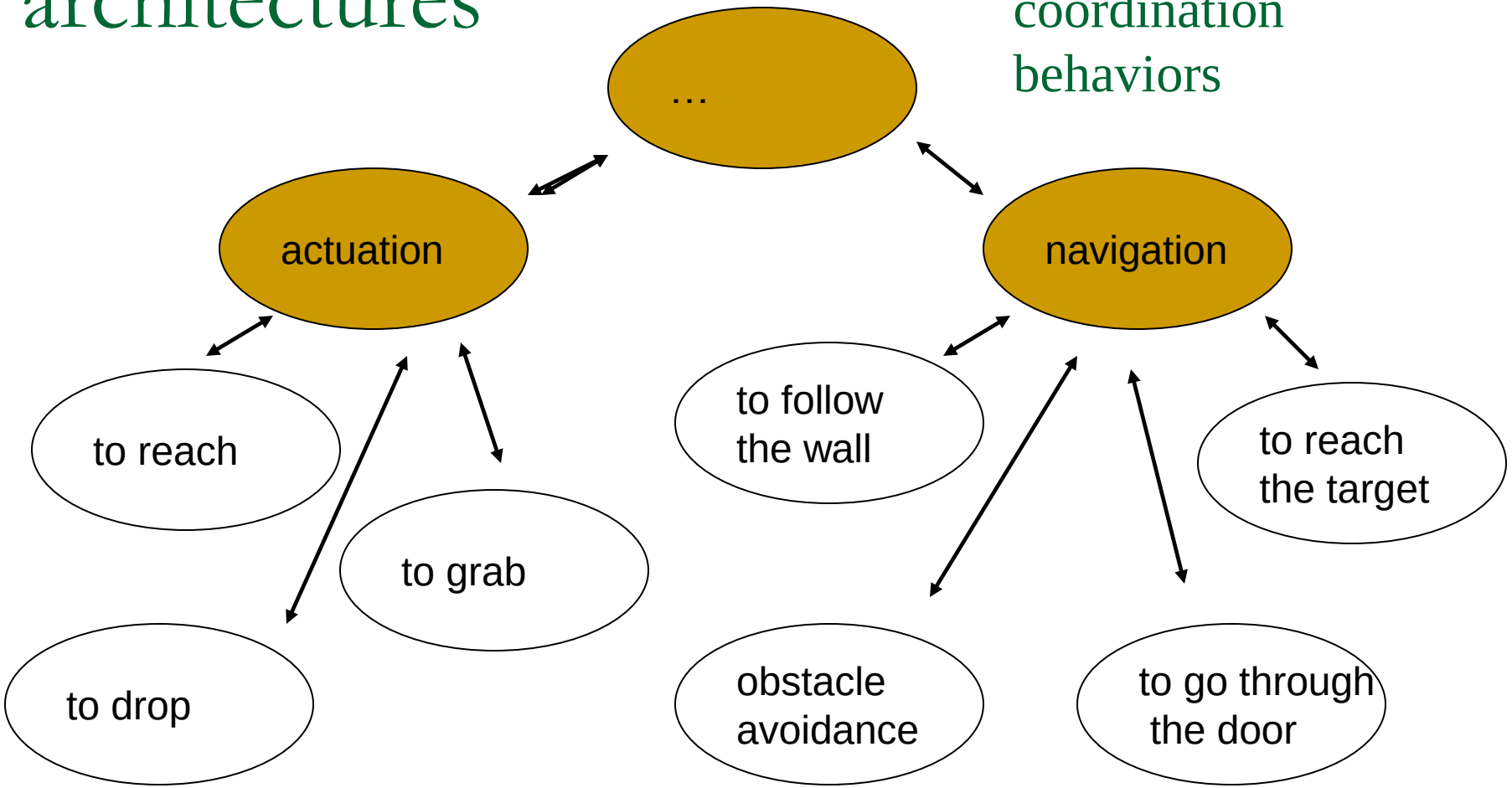
„Emergence”

- The question arises however: Individual behaviors are well defined functionally, so why should a coordinated collection of them produce novel or unanticipated results?
- The coordination can be
  - straightforward, such as choosing the highest ranked or most dominant behavior
  - may be more complex, involving fusion of multiple active behaviors
- Deterministic functions and certainly computable
- Why can not we predict their behavior exactly?

„Emergence”

- The answer to this question lies in the relationship between the robot and its environment.
- The real world
  - resists analytical modeling
  - nondeterminism
  - filled with uncertainty and dynamic properties
  - the perception process itself is also poorly characterized
- If a world model could be created that accurately captured all of its properties
  - Emergence would not exist
  - Accurate predictions could be made
- But it is the nature of the world to resist just such characterization, hence we cannot predict a priori, with any degree of confidence, in all but the simplest of worlds, how the world will present itself.

Hierarchical behavior based architectures



The action-selection problem

- How can the robot chose the most appropriate and most important action in a given time and even unusual situation? (Maes).
- Satisfactory behavior: The behavior which responses with a quite good action is better than the behavior which striving for optimum with limited sensing opportunities. (Simon)

The criteria of action-selection

- Goal-oriented: the behaviors work together to achieve one or more objectives
- Situatedness: the actions fit into the appropriate situation
- Continuous: the actions cooperate to achieve the goal
- Planning: foresight to avoid dangerous situations
- Robustness: in case of a faulty behavior, slight reduction in performance
- Reactivity: quick and timely response

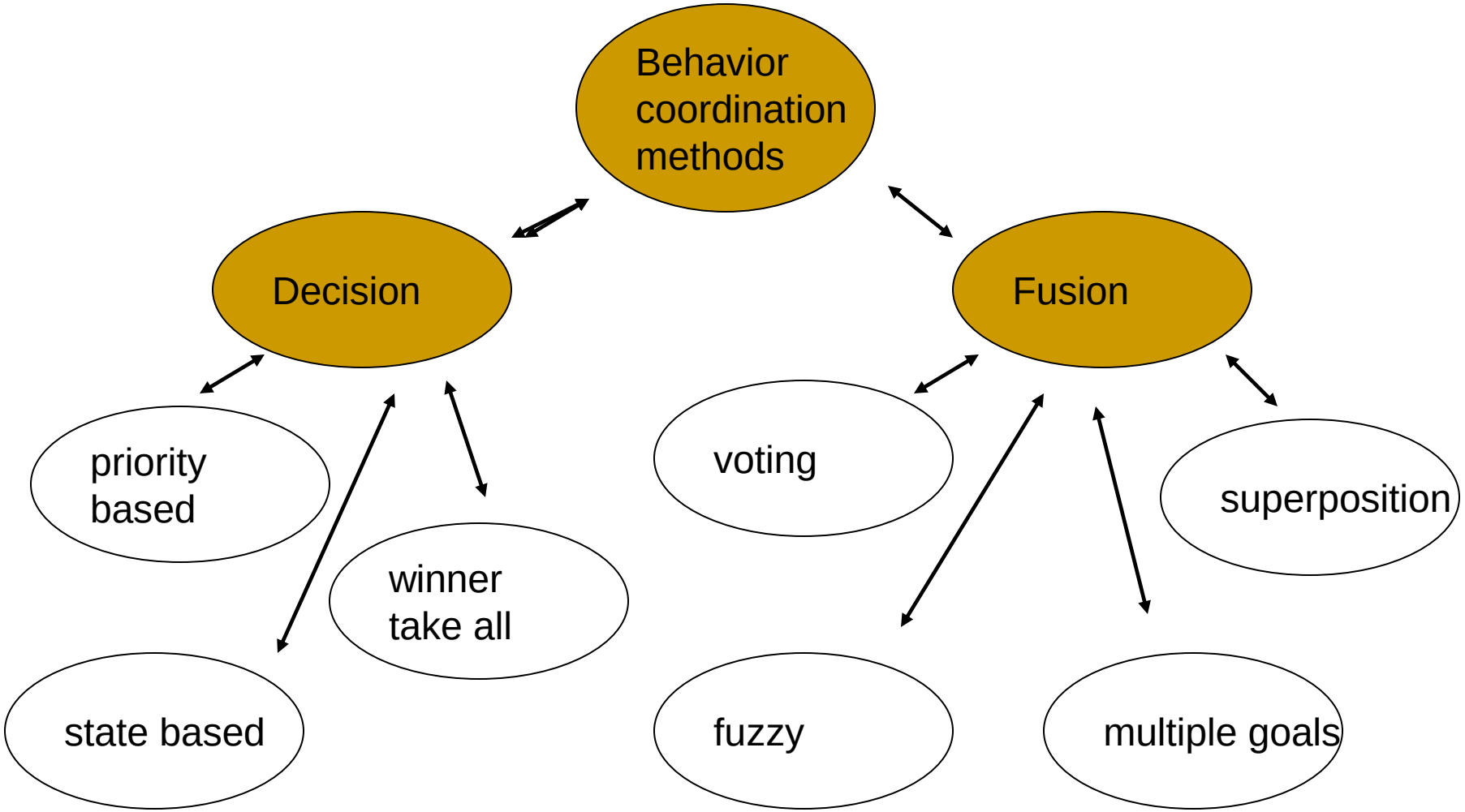
Behavioral coordination and behavioral fusion

- Behavioral coordination is interested in how to decide which behaviors should be activated at a given moment
- Behavioral fusion is responsible how to unite the actions of several, which will be forwarded to the actuators of the robot

Behavioral fusion

- The behavioral fusion has three steps:
- Proposing actions
  - All behavior, proposes an action based on the status and purpose of the required action
- Combining behaviors
  - Combining the proposed actions in accordance with given rules
- Action selection
  - Selecting the appropriate action based on the coordinated requirements

Behavior coordination methods



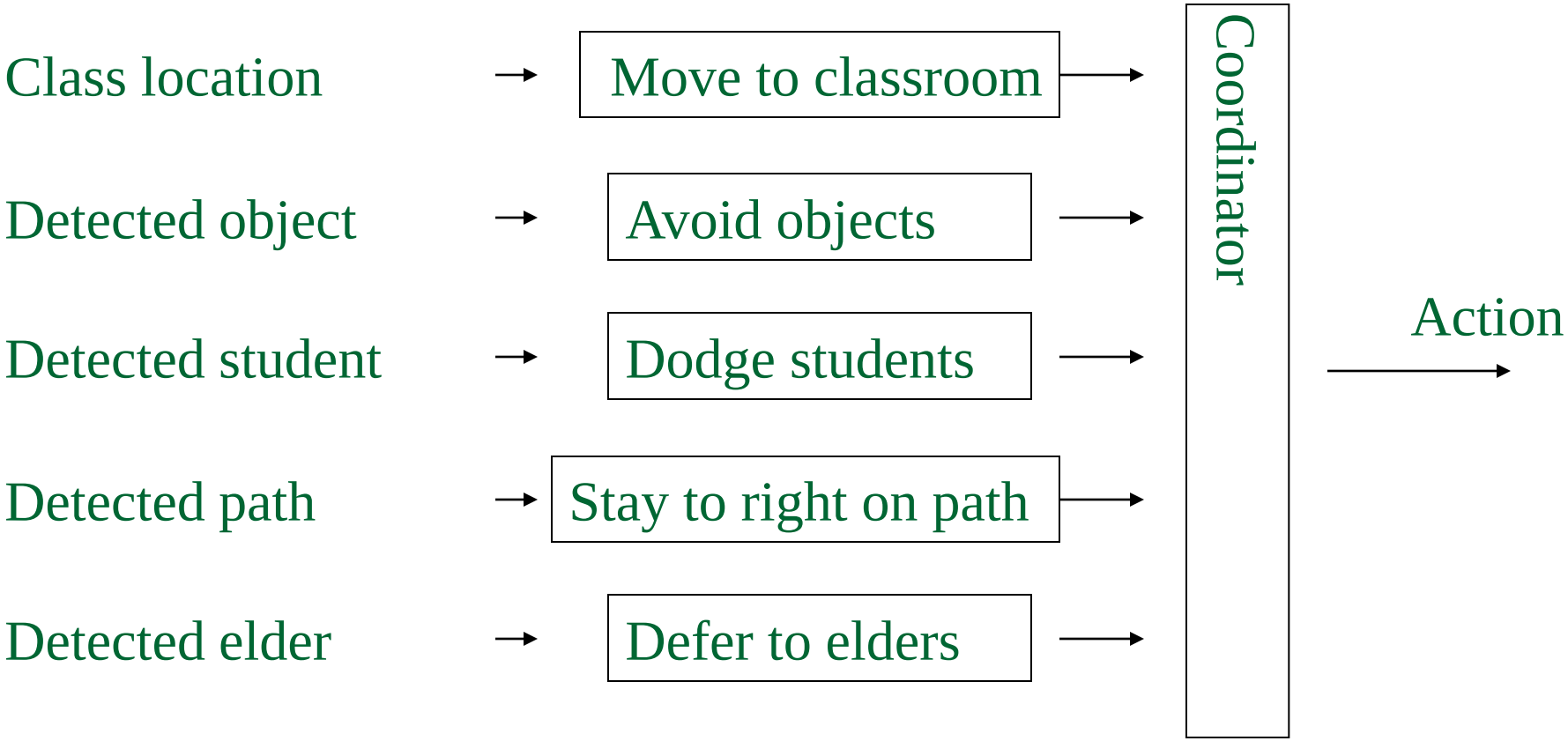
Behavior coordination methods

- Decision
  - Priority based (Brooks)
  - State-based (Kosecka, Arkin)
  - Winner-take-all (Maes)
- Behavioral fusion
  - Voting (Rosenblatt & Payton, Bekey)
  - Superposition (Khatib, Arkin, Schoener)
  - Environment dependent mixture (Ruspini & Saffioni & Konolige, Tunstel, Bonarini)
  - Multiple goals (Pirjanian & Mataric)

Navigation example

If we have more than one behaviors, how can we coordinate them?

Example:



Behavioral encoding

- behaviors can be described in three different ways
  - SR diagrams (stimulus-response diagram)
  - Functional notation
  - FSA diagrams (Finite State Acceptor diagramm)
- A behavior can be expressed as a triple: (S,R,β)

  β(s)=r, where:

  β = behavior,

  s = stimuli,

  r = response
- g is a scalar gain value (strength mulitplier), modifying the magnitude of the overall response r, hence r'=gr
- Encoding can be:
  - Discrete
  - Continuous

Notation in case of multiple behaviors

- **S** denote a vector of all stimuli s_{it} relevant for each behavior β_i detectable at time t.
- **B** denote a vector of all active behaviors β_i at a given time t.
- **G** denote a vector encoding the relative strength or gain gi of each active behavior β_i
- **R** denote a vector of all responses r_i generated by the set of active behaviors

Notation in case of multiple behaviors

- A new behavioral coordination function, C, is now defined such that: ρ=**C**(**G***B(**S**)) or alternatively ρ=**C**(**G***R) where

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \dots \\ \mathbf{r}_n \end{bmatrix}, \mathbf{S} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \dots \\ \mathbf{s}_n \end{bmatrix}, \mathbf{G} = \begin{bmatrix} g_1 \\ g_2 \\ \dots \\ g_n \end{bmatrix}, \textit{and } \mathbf{B} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_n \end{bmatrix}$$

ρ = is the vector encoding the global response that therobot

Classroom example

- Given the robot's current perceptions at time t:

$$S = \begin{bmatrix} (move - to - class, 1.0) \\ (avoid - object, 0.2) \\ (dodge - student, 0.8) \\ (stay - right, 1.0) \\ (defer - to - elder, 0.0) \end{bmatrix}$$

- for each $s_i = (p, \lambda)$, where λ is the stimulus p's percentage of maximum strength.

Classroom example

- Behavioral response:

$$B(S) = \begin{bmatrix} \beta_{movetoclass}(s_1) \\ \beta_{avoidobject}(s_2) \\ \beta_{dodgestudent}(s_3) \\ \beta_{stayright}(s_4) \\ \beta_{defertoelder}(s_5) \end{bmatrix}$$

the direction and the strenght of R can be computed:

- $R = \begin{bmatrix} r_{movetoclass} \\ r_{avoidobject} \\ r_{dodgestudent} \end{bmatrix}$ $R_{strenght} = \begin{bmatrix} 1.0 \\ 0.0 \\ 0.8 \end{bmatrix}$ ← The strength of the stimulus is below a given threshold

Classroom example

- ρ=**C**(**G***R)
- Before the coordination function C is applied, R is multiplied by the gain vector G, G denots the importance of the behavior.

$$G = \begin{bmatrix} g_{movetoclass} \\ g_{avoidobject} \\ g_{dodgestudent} \\ g_{stayright} \\ g_{defertoelder} \end{bmatrix} = \begin{bmatrix} 0.8 \\ 1.2 \\ 1.5 \\ 0.5 \\ 0.8 \end{bmatrix}$$

- For G, stay-right is least important (0.5), dodge-student is deemed the most important behavior (1.5)

Classroom example

$$R_{strenght} = \begin{bmatrix} 1.0 \\ 0.0 \\ 0.8 \\ 1.0 \\ 0.0 \end{bmatrix} \qquad G = \begin{bmatrix} g_{movetoclass} \\ g_{avoidobject} \\ g_{dodgestudent} \\ g_{stayright} \\ g_{defertoelder} \end{bmatrix} = \begin{bmatrix} 0.8 \\ 1.2 \\ 1.5 \\ 0.5 \\ 0.8 \end{bmatrix}$$

$$\rho = C(G * R) = C(R') = C \begin{bmatrix} g_1 * r_1 \\ g_2 * r_2 \\ g_3 * r_3 \\ g_4 * r_4 \\ g_5 * r_5 \end{bmatrix} \qquad R'_{strenght} = \begin{bmatrix} 0.8 \\ 0 \\ 1.2 \\ 0.5 \\ 0 \end{bmatrix}$$

Classroom example

- Action-selection (winner-take-all)
  - A single component vector g (based on some metric such as greatest magnitude) is chosen by C, assigned to ρ and executed
  - This is the component with a value of 1.2, associated with the dodge-student behavior.
- Voting
  - The action, which receives the highest number of votes, will be implemented.
  - Assuming that stay-right and move-to-class behaviors vote r_1 behavior, and avoid-object votes to re, then r_1 will have the highest value of 1.3.

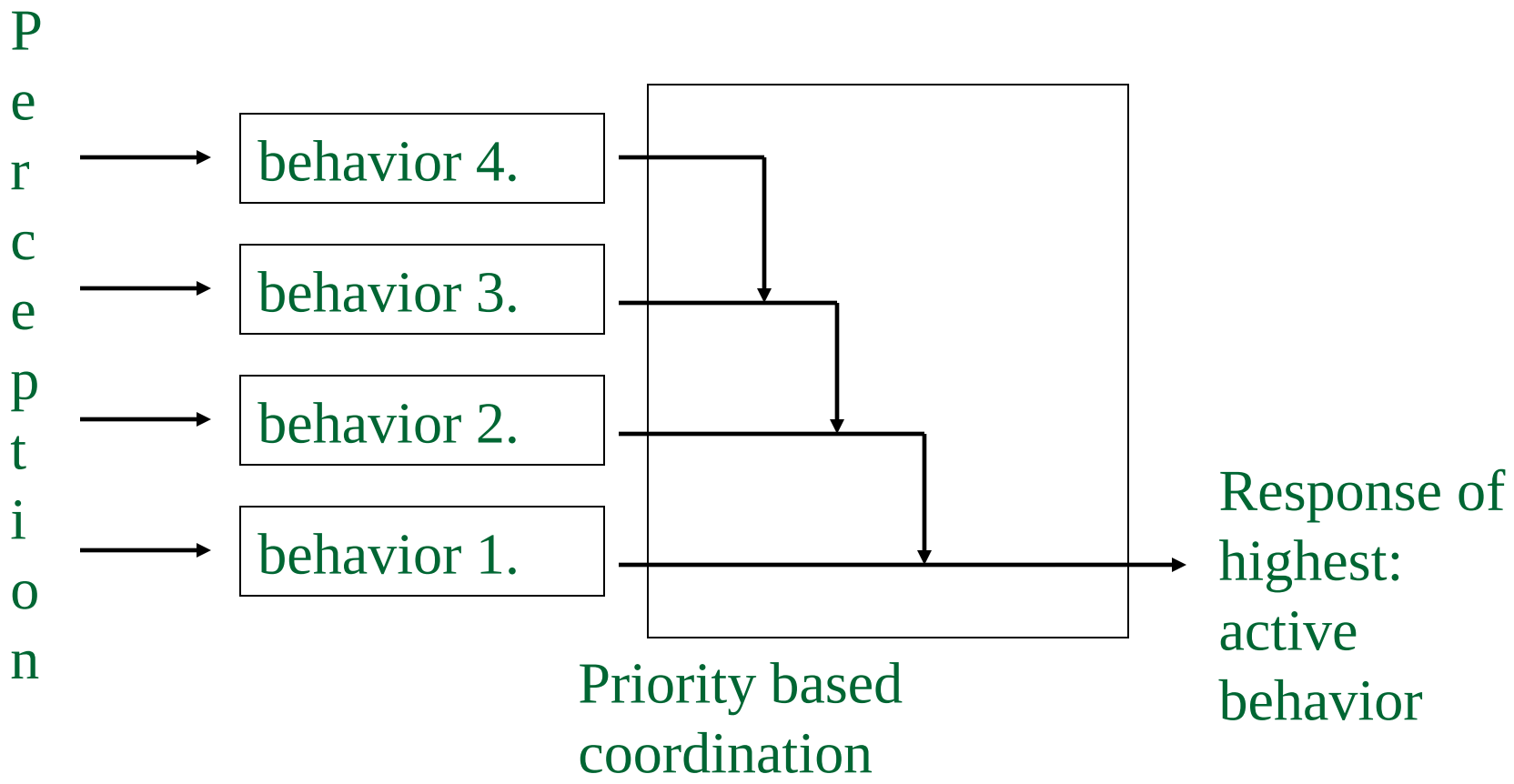
Classroom example

- Priority based arbitration system
  - The highest-ranked component behavior encoded in G (above threshold) would be chosen and executed, independent of the individual components' relative magnitudes $R_{strength}$.
  - The dodge-student would also be chosen using this method, assuming the stimulus is above threshold, since godge-student(1.5) is the highest-ranked behavior.

Competitive methods

- Competitive methods provide a means of coordinating behavioral response for conflict resolution.
- winner take all
- The decision can be:
  - priority based
  - action selection

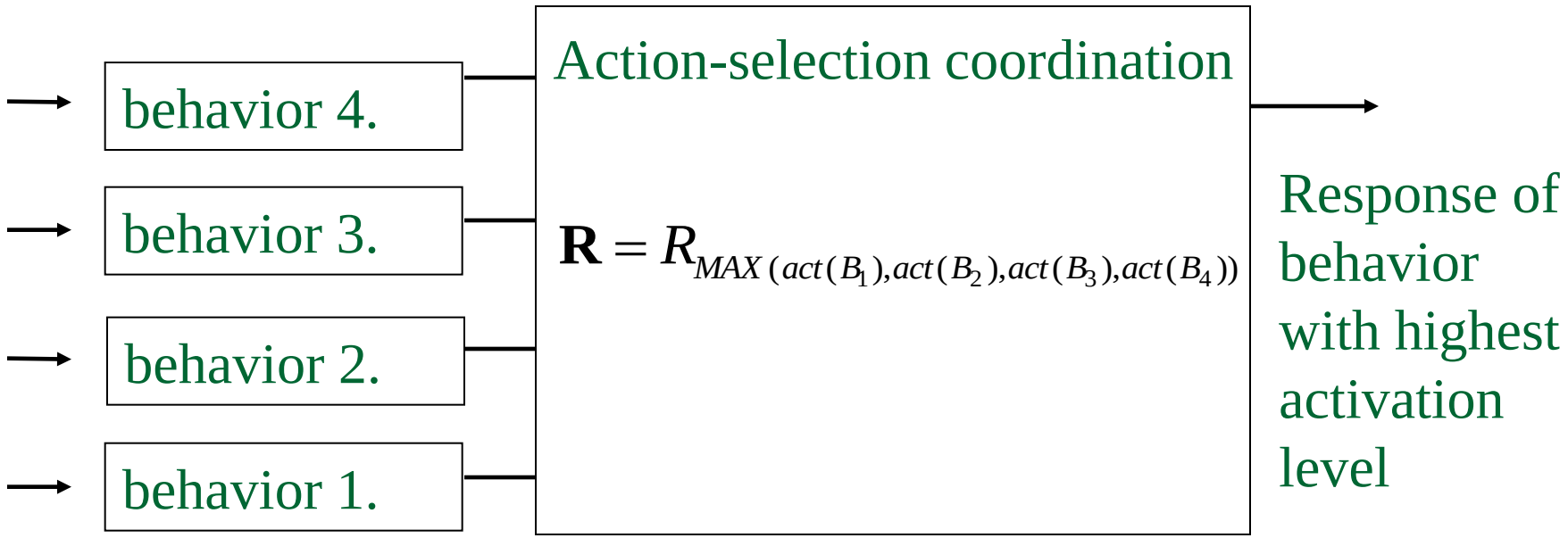
Priority-based coordination



- Fixed prioritization network in execution time

Action-selection coordination

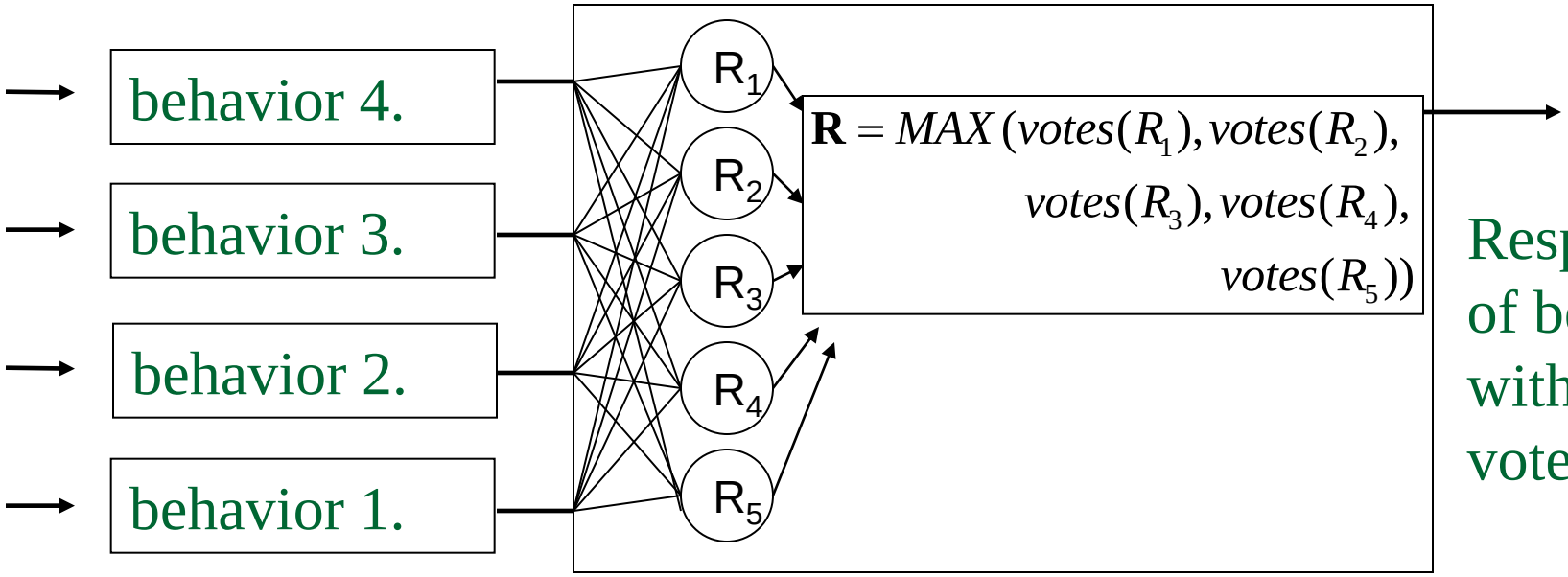
Perception



- The behaviors actively compete with each other through the use of activation levels driven by both the agent's goals (or intentionality) and incoming sensory information.
- The concept of lateral inhibition can easily be implemented using this method where one behavior's strong output negatively affects another's output
- No fixed hierarchy is established

Perception

Voting based coordination



Voting based coordination

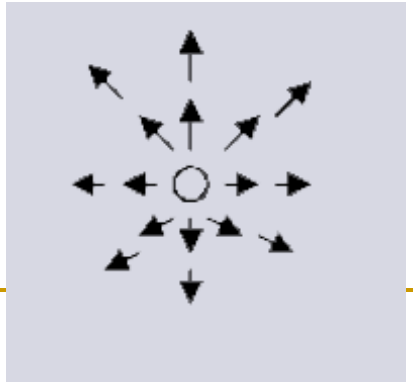
- Predefined set of discrete motor responses
- All behavior votes for all motor response
- The output will be the motor response which gets the most votes
- The priority dynamically changes in execution time

Voting

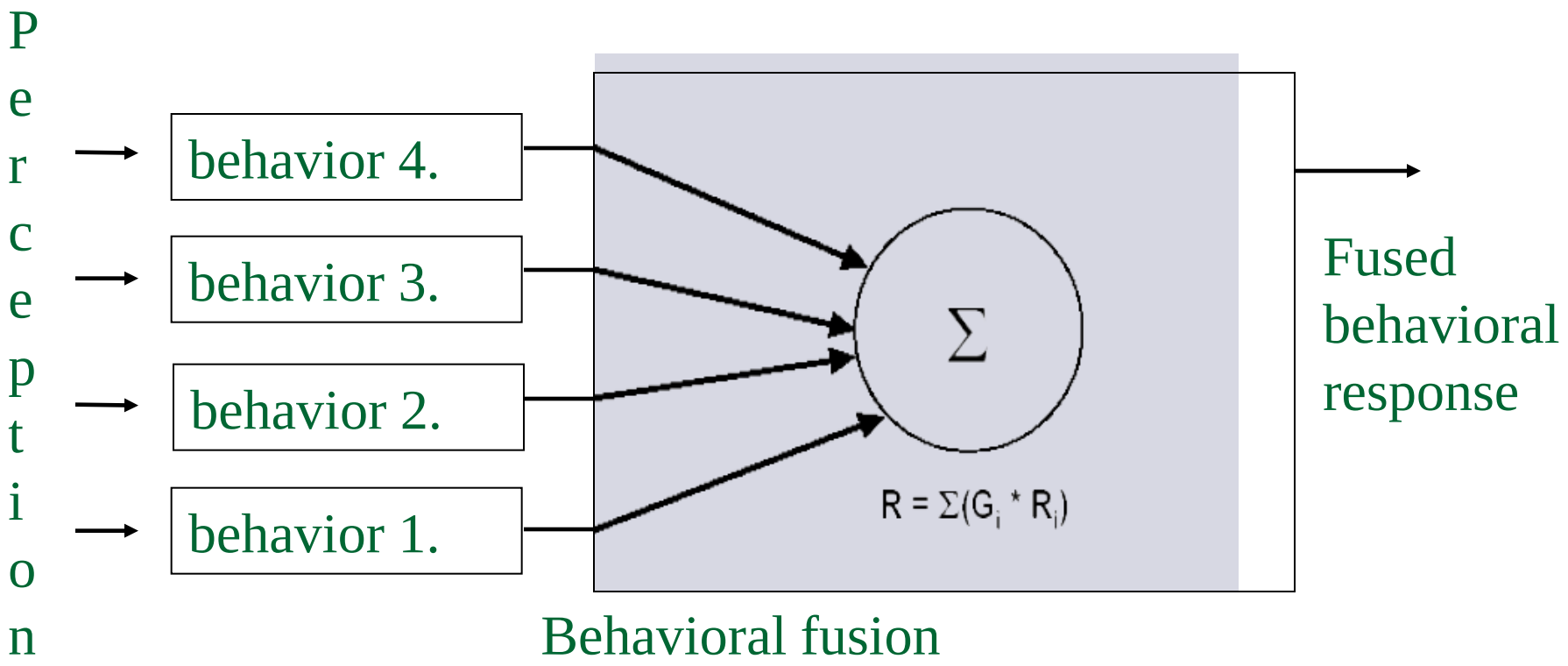
- Response set: {hard-left, soft-left, straight-ahead, soft-right, hard-right}
- Behaviors: {e.g., goal-seeking, obstacle-avoidance, road-following, cross-country, teleoperation, map-based navigation}
- Each active behavior has a certain number of votes and a user-provided distribution for allocating those votes.
- Arbitration takes place through a winner-take-all strategy in which the single response with the most votes is enacted
- This allows a level of behavioral cooperation, but the method is still arbitrary in its choice of a single response.

Cooperative methods

- Behavioral fusion provides the ability to use concurrently the output of more than one behavior at a time.
- The central issue in combining the outputs of behaviors is finding a representation to fusion.
- Usual method:
  - Vector addition in potential-fields
  - Potential-field example:

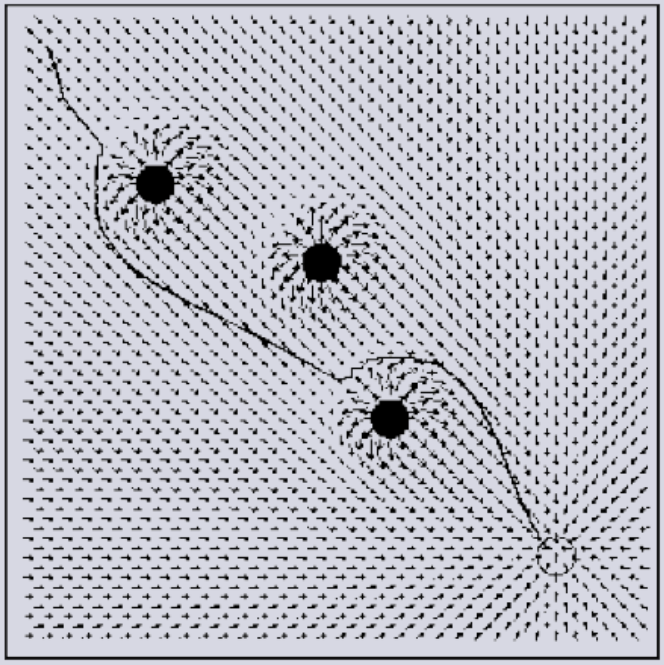


Behavioral fusion with vector addition

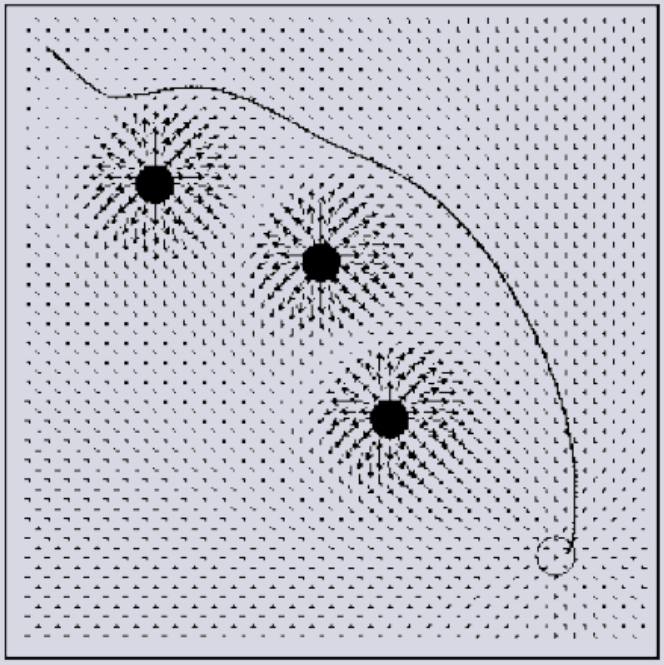


Behavioral fusion - Gain

$g_{goal\text{-}attractor} = 2 * g_{obstacle\text{-}avoidance}$

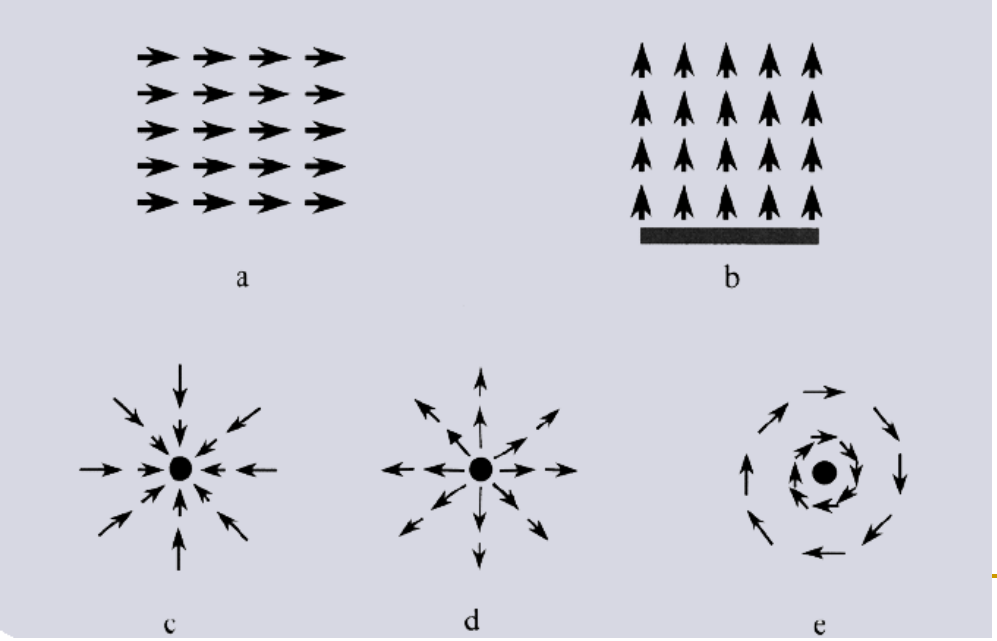


$g_{goal\text{-}attractor} = 0.5 * g_{obstacle\text{-}avoidance}$



Visualization of potential fields

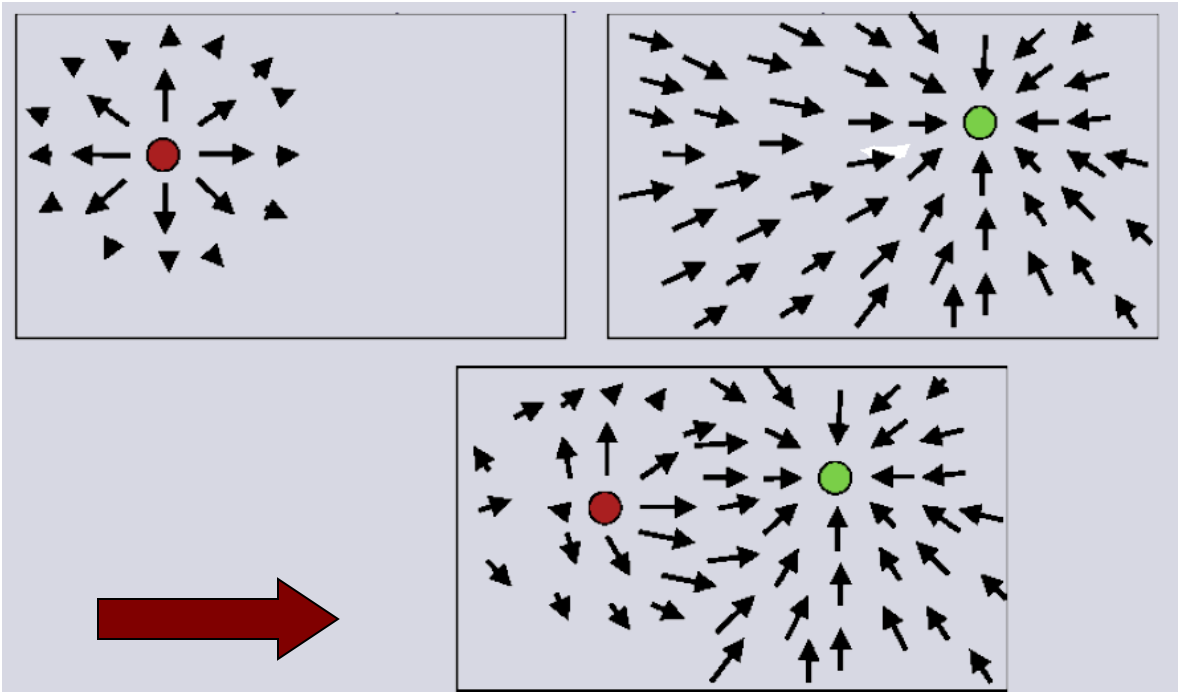
- Similar as the robot moves in a force field
- 3D surfaces (such as gravity) - marble (robot)
- Vectors point from the mountain peaks towards the valleys



a.) uniform
b.) perpendicular
c.) attraction
d.) repulson
e.) tangential

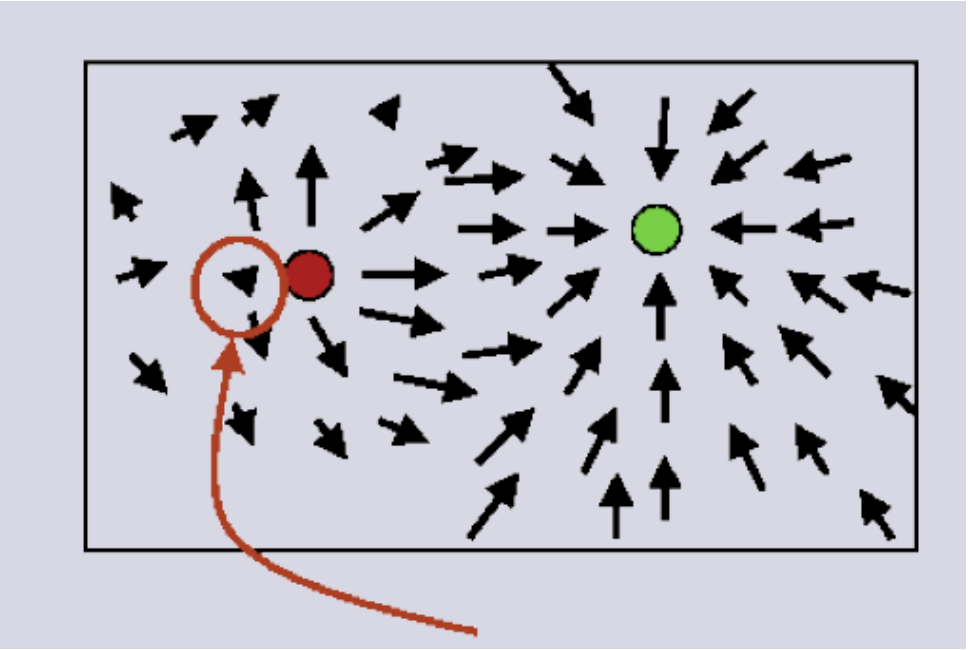
Combination of fields or behaviors

- Calculate the potential field of all behavior
- Aggregate the vectors in the robot position to obtain the output vector



Problem of local minimum

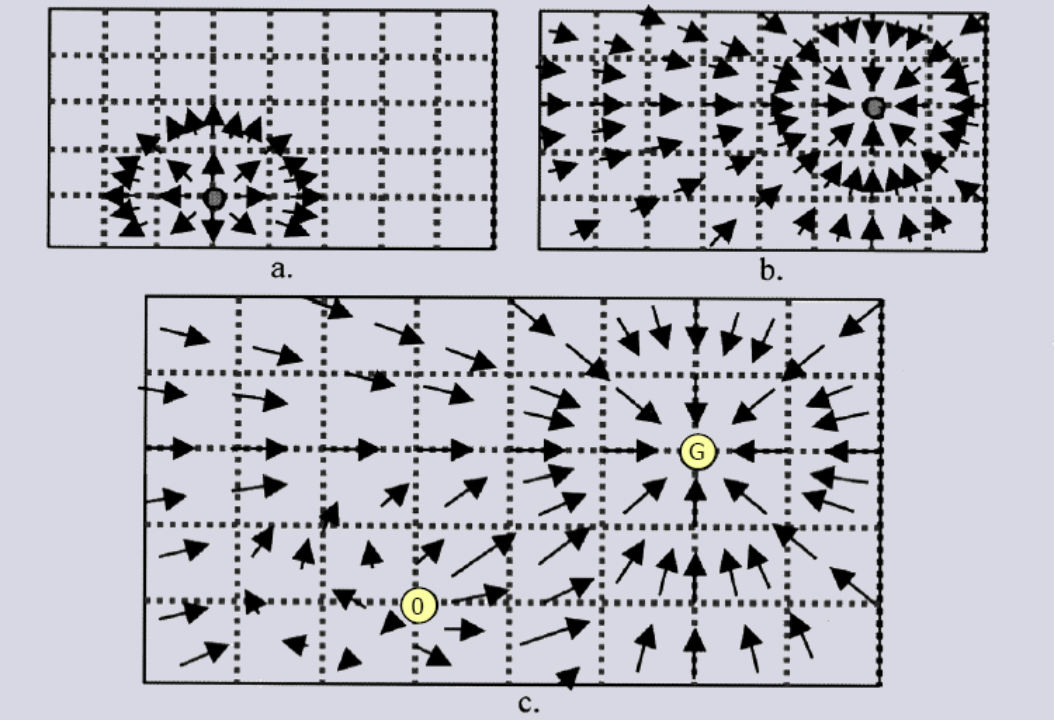
- When the robot reaches a local minimum point, it can not do anything, it stays there



Local minimum, the sum of the vectors is 0

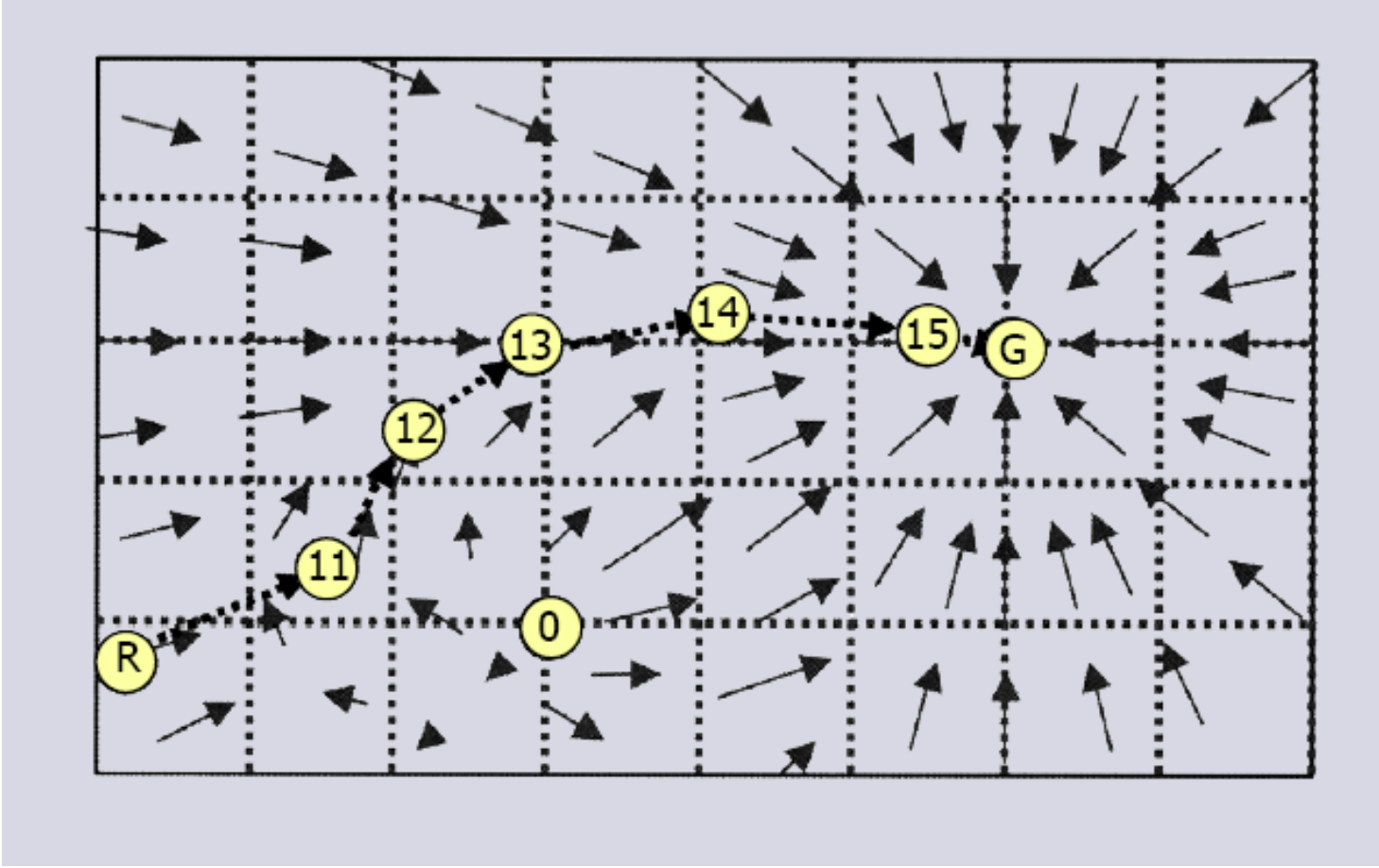
Potential fields method

- Combination of behaviors and fields



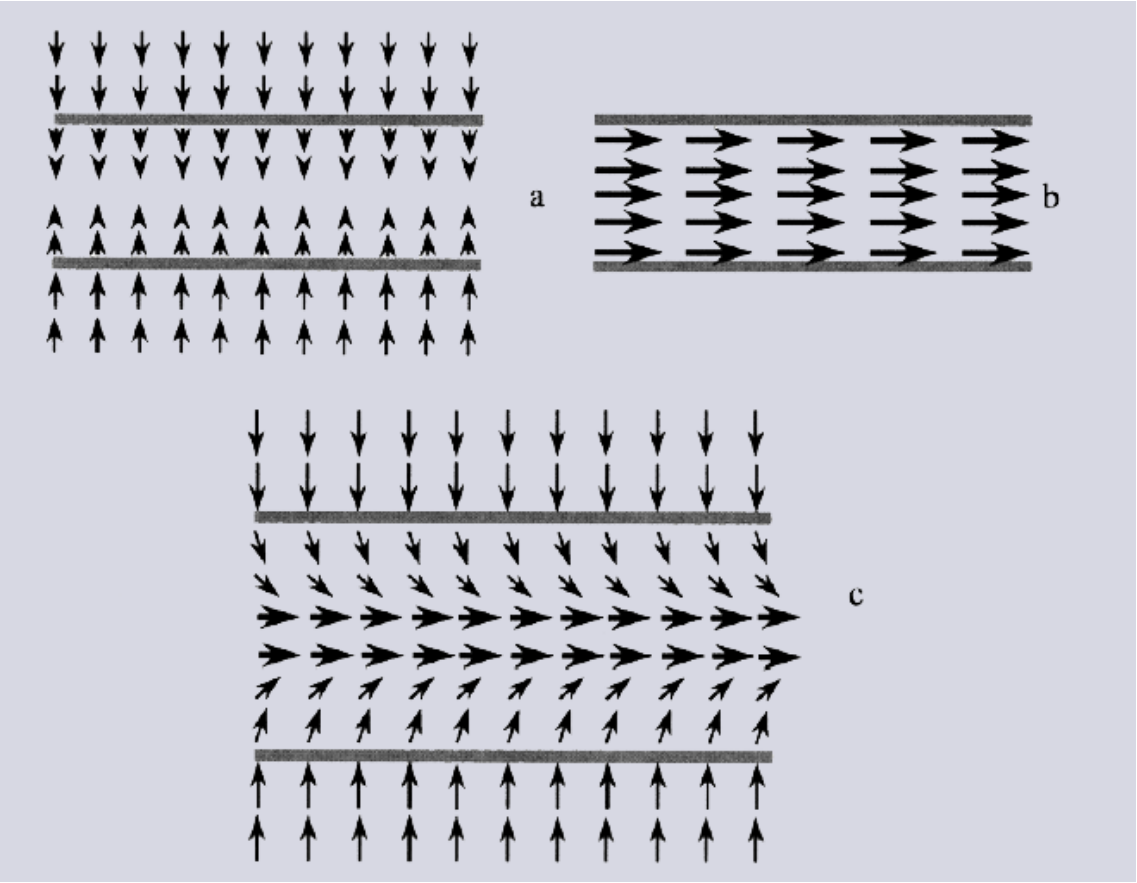
a.) repulsion from the object b.) attraction towards the goal c.) a combination of both

Potential fields method



The robot trajectory

Potential fields method

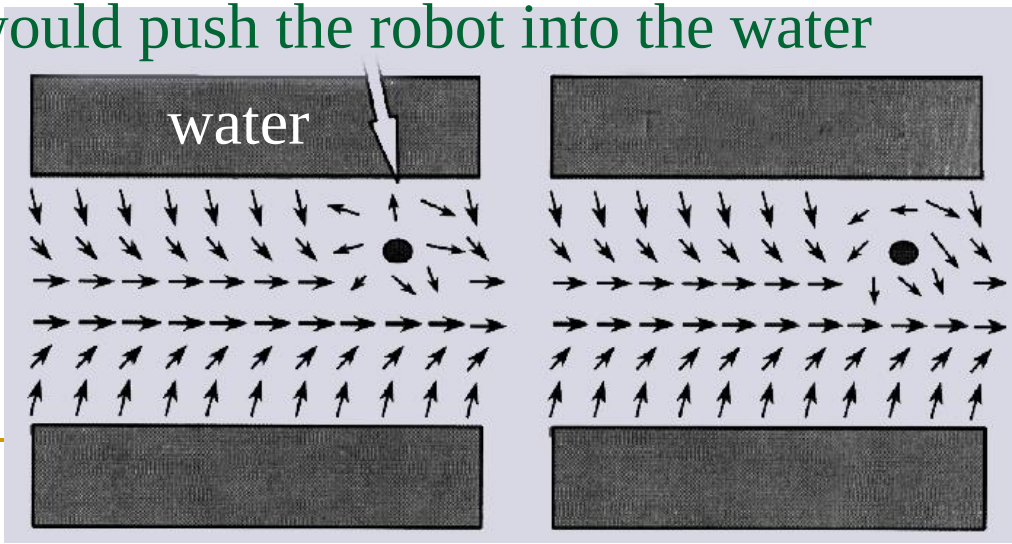


a.) perpendicular, b.) field which helps to follow the corridor c.) combination of uniform fields

Advantages and disadvantages

- The problem of potential fields: local minimum
- Solution: Navigation patterns
- The repulsive field is modified by a tangential field. The direction of the trajectory depends on wether the robot is on the left or the right side of the center line.

it would push the robot into the water



Advantages and disadvantages

- Behaviors contain one or more motor patterns. The motor patterns are always potential fields.
- All behavior is processes concurrently and the output vectors are summed.
- Although all behavior is handled equally, the behaviors can contribute differently to the ultimate robot's behavior. A behavior can change the strength of another behavior, and influence its output. This means that a behavior can be inhibited or excited by different behaviors, although this method is rarely used.

Weighted coordination

- Behaviors have constant priority and state dependent strength
- The responses of the behaviors will sum up in priority descending order
- The coordination adjusts the strength of the response of the next active behavior in proportion to the overall response.
- The method further reduces the strength of lower priority behaviors.
- Actions with same priority will be averaged before they would be added to the response.

Weighted coordination

- Behaviors: A, B, C, D
- Priorities: p_A=4, p_B=3, p_C=3, p_D=1
- Responses: R_A=-400, R_B=-100, R_C=200, R_D=500
- Strength: S_A=0.25, S_B=1.0, S_C=0.5, S_D=0.5

behavior	priority	response	strength	valid response	valid strength
A	4	-400	0.25	-400	0.25
B	3	-100	1.0	R_B & R_C	
C	3	200	0.5	R_B & R_C	
B&C	3	0	0.75	-100	1.0

- R_B and R_C have same priority therefore their responses have to be avareged before combining them with response R_A
- R_D has no effect, because the valid strength is already 1.0

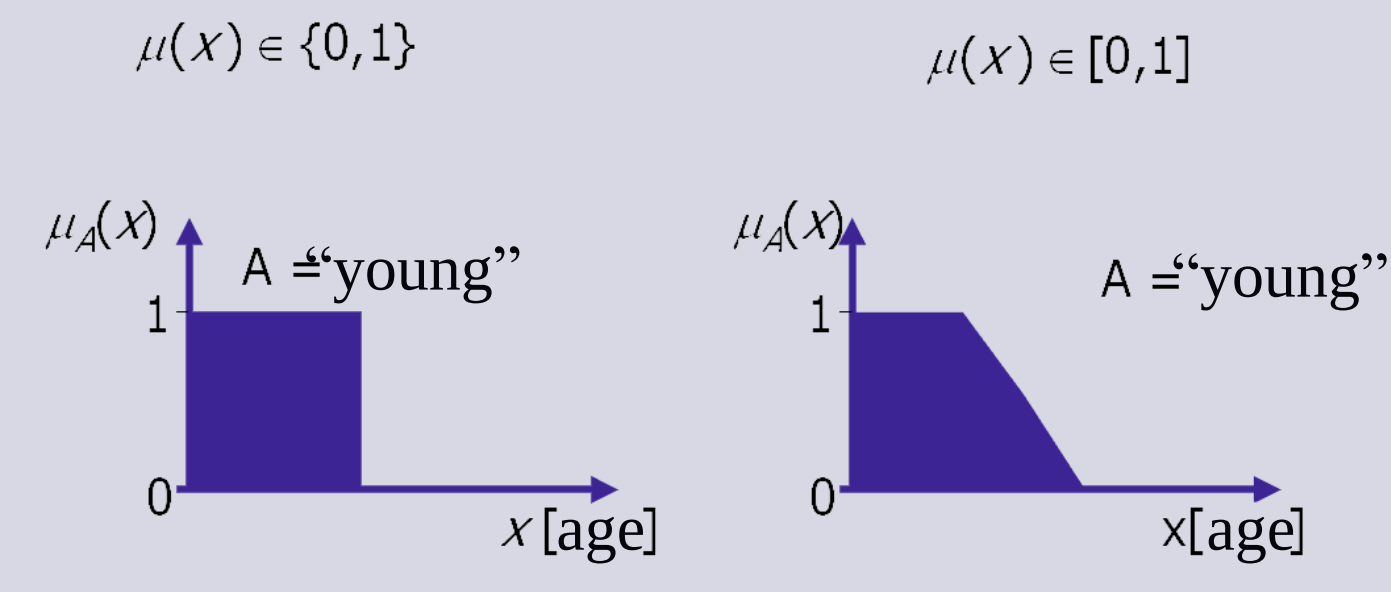
Fuzzy sets

Boolean logic

x is an element of A, or not

Fuzzy logic

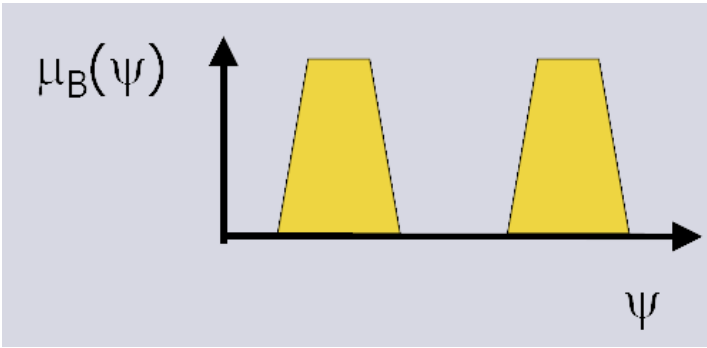
value x is an element of a given range



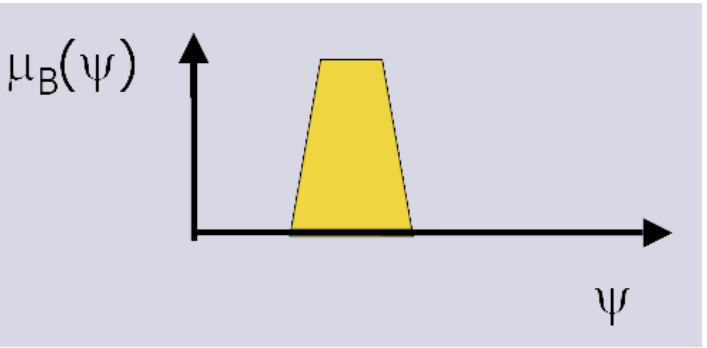
Fuzzy based responses

- Instead of a definite response, all behavior expresses demand for all possible responses.
- For every action the claims are given with fuzzy sets, where $\mu_B(R)$ denotes the behavior's demand to give R response

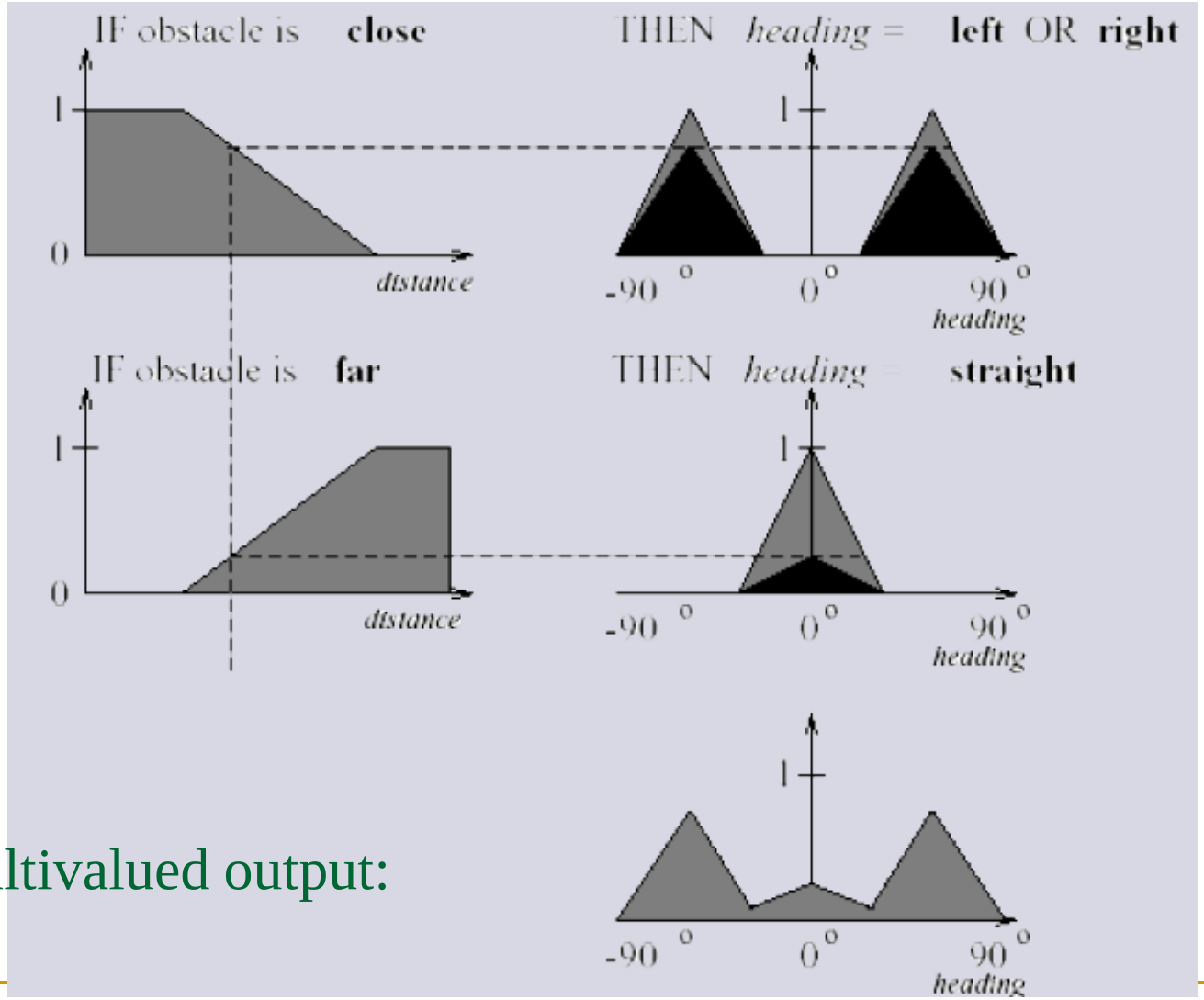
obstacle avoidance



toward the goal

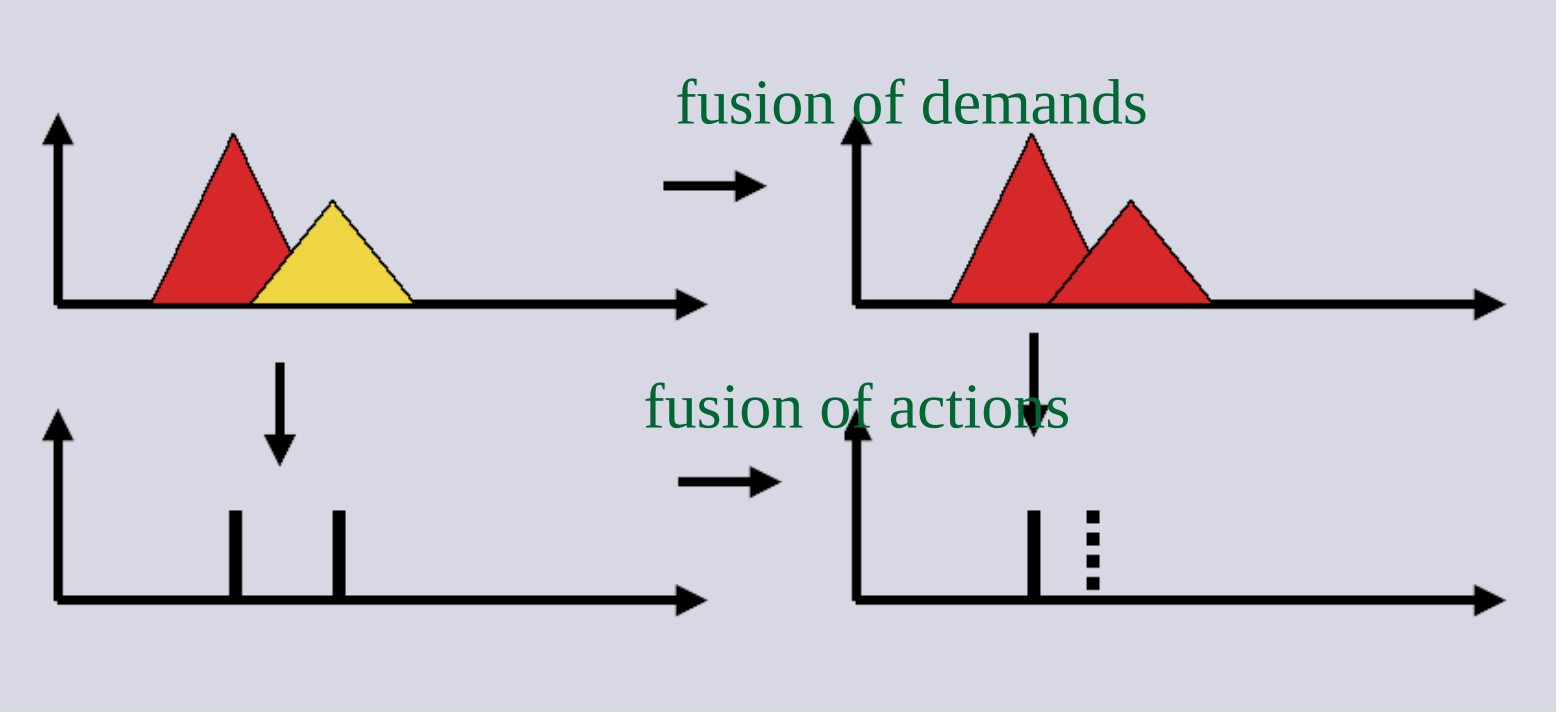


Obstacle avoidance - Fuzzy behavior



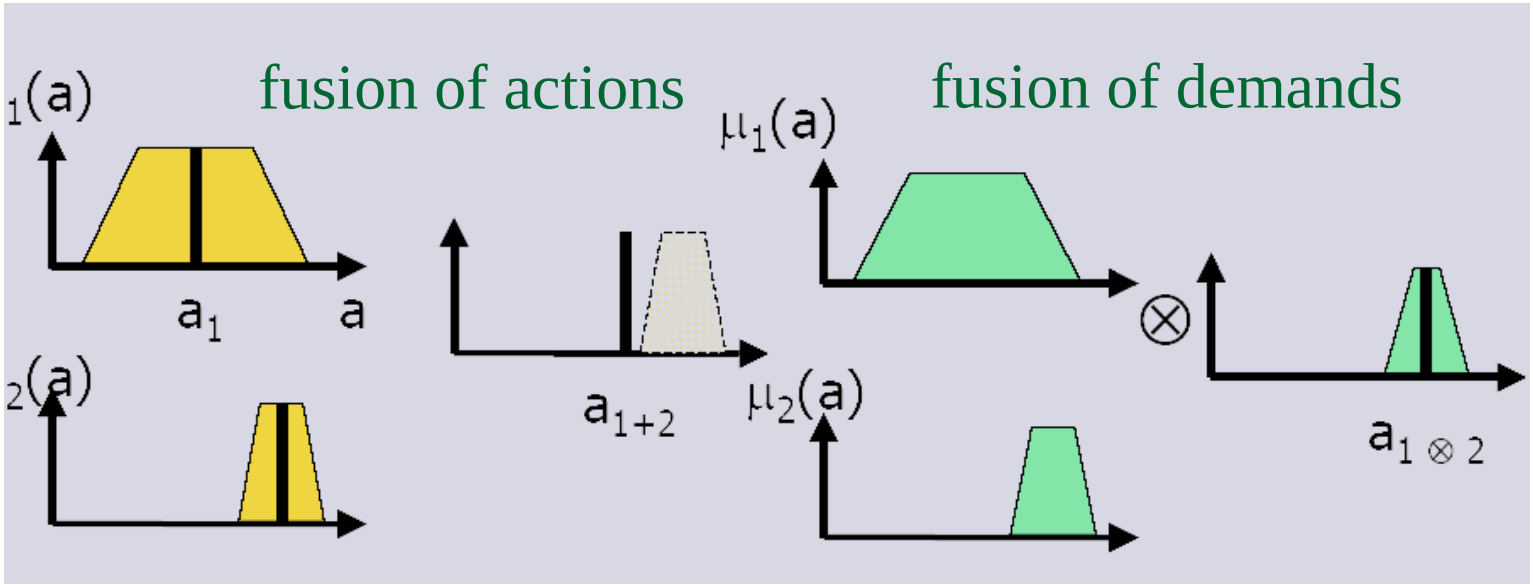
Multivalued output:

Action fusion vs. demand fusion



Fuzzy fusion

- It uses multivalued logic to summarize the demands of the responses of active behaviors

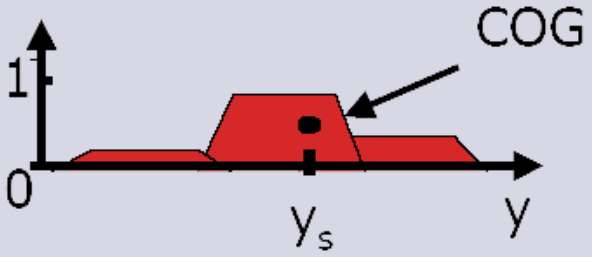


Defuzzification

- center of gravity, mean of maxima, centroid of largest

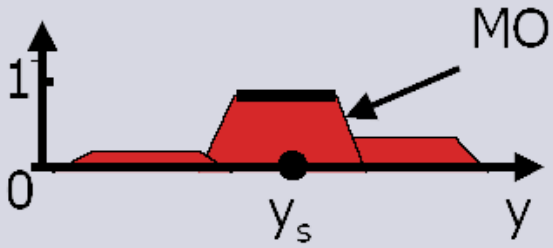
Center of gravity (COG)

$$y_s = \frac{\int \mu(y)\ y\ dy}{\int \mu(y)\ dy}$$

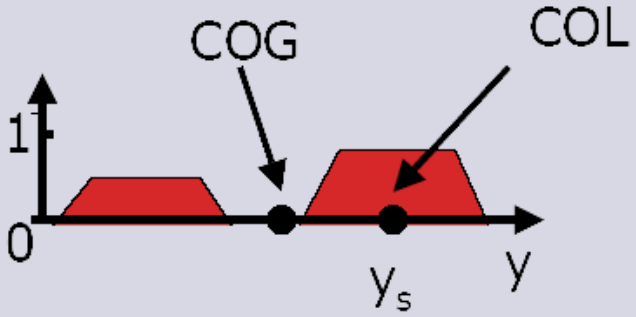


Mean of maxima (MOM)

$$y_s = \frac{\int_{\mu(y)=\max_{y'}\{\mu(y')\}} y\ dy}{\int_{\mu(y)=\max_{y'}\{\mu(y')\}} dy}$$

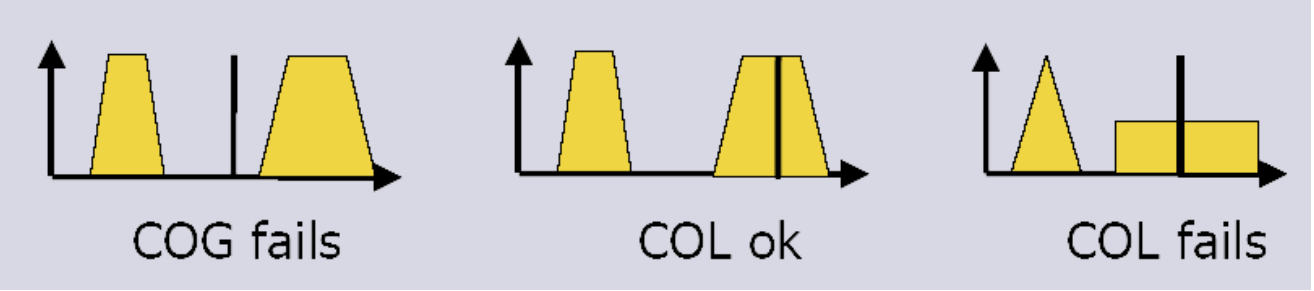


Centroid of largest (COL)



Fuzzy fusion

- In case of contrary demands defuzzification may give non-desired results
- Further restrictions may be required
- We separate the rules which require mutually exclusive output demands and then we apply winner-take-all heuristics



End of Lecture 03.

i.) Behavior based robotics

György Cserey
02.22.2021.