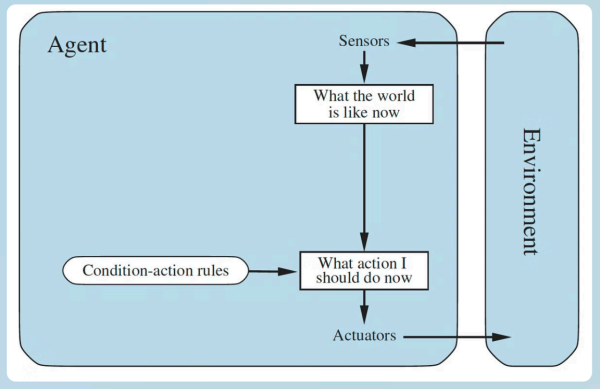
2. agentes reativos

introduction

- make a direct mapping between perceptions and actions
- have a local view of the environment
- may have memory (small)



production systems

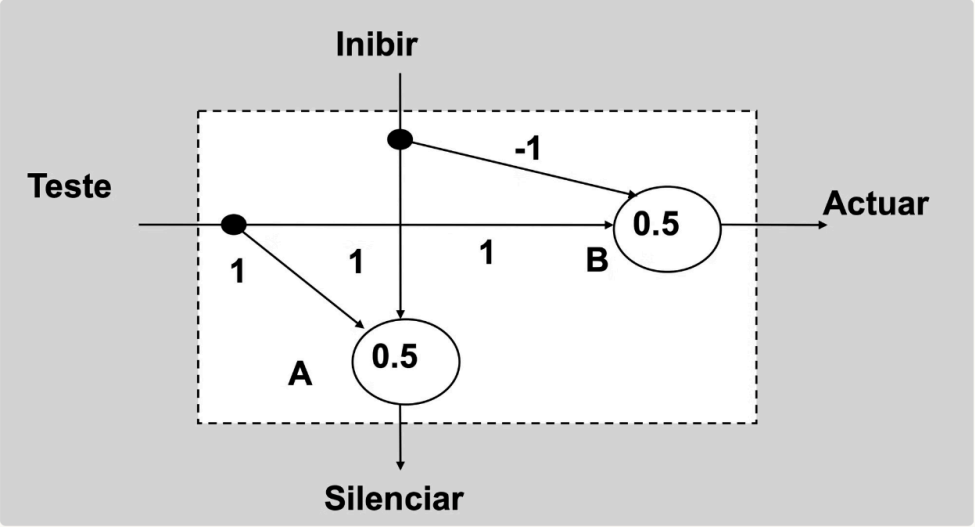
- the **symbolic** approach is used: rules map perceptions in actions and may or not have memory
- **production systems**
- the rules are chosen BY THE ORDER THEY ARE DEFINED

```
¬O1 → A1
O2 → NIL
```

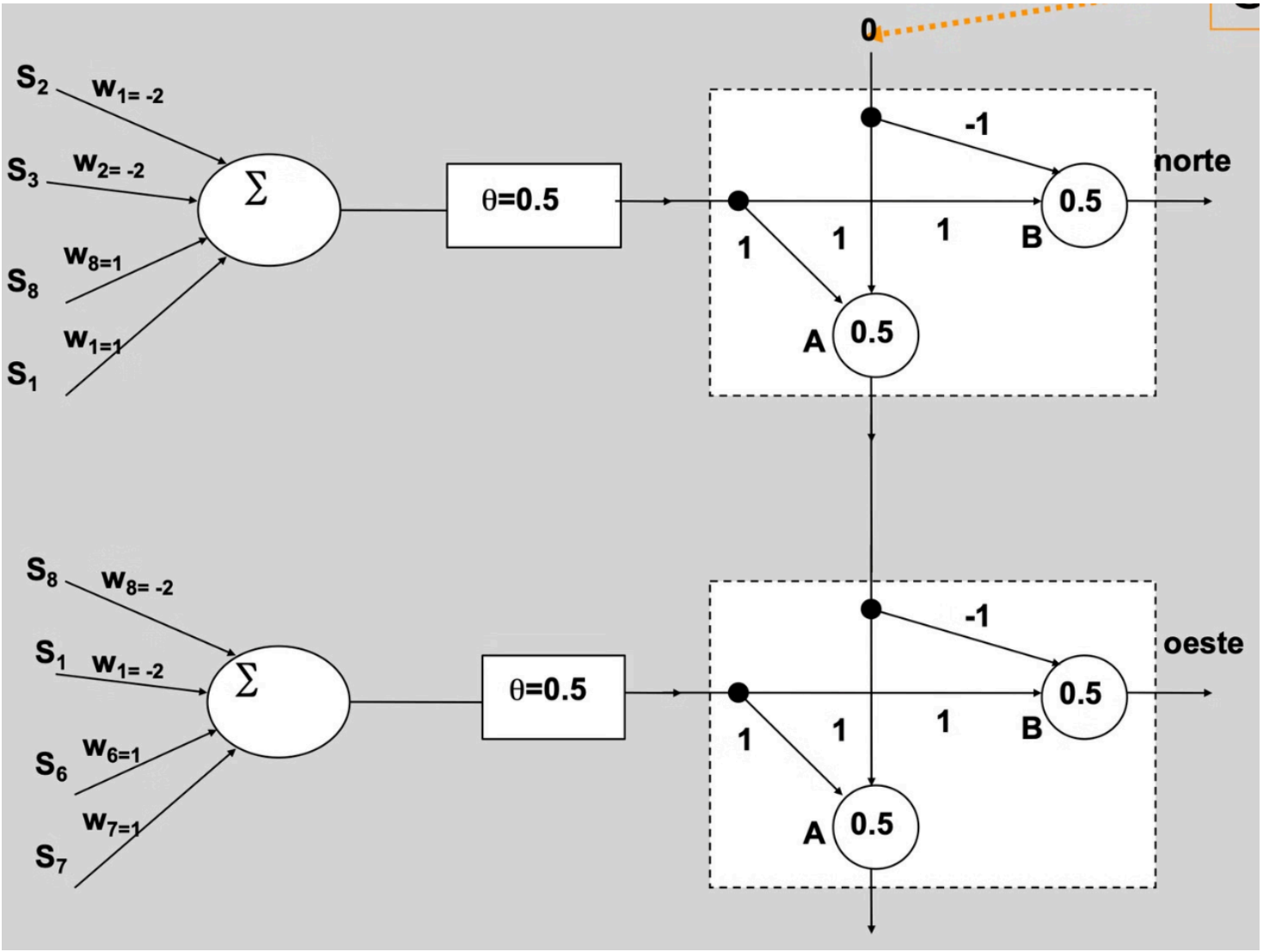
TLU's

- the **connectionist** approach is used
- **linear threshold units**: médias pesadas das observações podem fazer disparar ou não a TLU's, espoletando ou não a ação correspondentes
- duas TLU's podem ser usadas para criar uma **unidade TISA (testar, inibir, silenciar, atuar)**
- **TLU A**: *OR* entre as entradas *Inibir* e *Testar*, ou seja, *Silenciar* apenas será 0 se ambas *Inibir* e *Testar* forem 0, não executando a ação e passando à próxima regra
- **TLU B**: *Atuar* será 1 quando *Testar* for 1, a menos que *Inibir* seja também 1
- a entrada é multiplicada pelo valor na seta e à chegada do limiar, esses produtos são somados: se forem superiores à TLU (0.5), devolve 1

Chaining Rules



- várias podem ser encadeadas (a saída *Silenciar* da primeira liga-se à entrada *Inibir* da segunda), sendo que a primeira entrada *Inibir* na sequência (ou seja, a entrada *Inibir* da primeira regra) é sempre 0



- assim, se a primeira regra das produções for cumprida, a entrada *Testar* na primeira TISA será 1 (e em *Inibir* será sempre zero), pelo que a saída em *Atuar* será $1 * 1 + 0 * -1 = 1$ e em *Silenciar* será $1 * 1 + 0 * 1 = 1$, passando 1 para a próxima entrada *Inibir* das seguintes regras, prevenindo-as de serem ativadas

subsumption architecture

- the **nature inspired** approach is used

It emphasizes a **decentralized** and **hierarchical design**, enabling complex behavior to **emerge** from simple components without relying on a centralized controller or extensive symbolic reasoning.

Layered Design

The system is composed of layers, each responsible for a specific behavior or task

Higher layers can "subsume" the functionality of lower layers, overriding or modifying their output when needed.

Behavior-Based Approach

Each layer corresponds to a specific behavior, such as obstacle avoidance, path following, or goal seeking

Behaviors operate independently and in parallel, with simpler behaviors typically handled by lower layers.

Reactive and Bottom-Up

Focuses on direct interaction with the environment through sensors and actuators.
Avoids reliance on high-level symbolic reasoning or internal world models.

Decentralization:

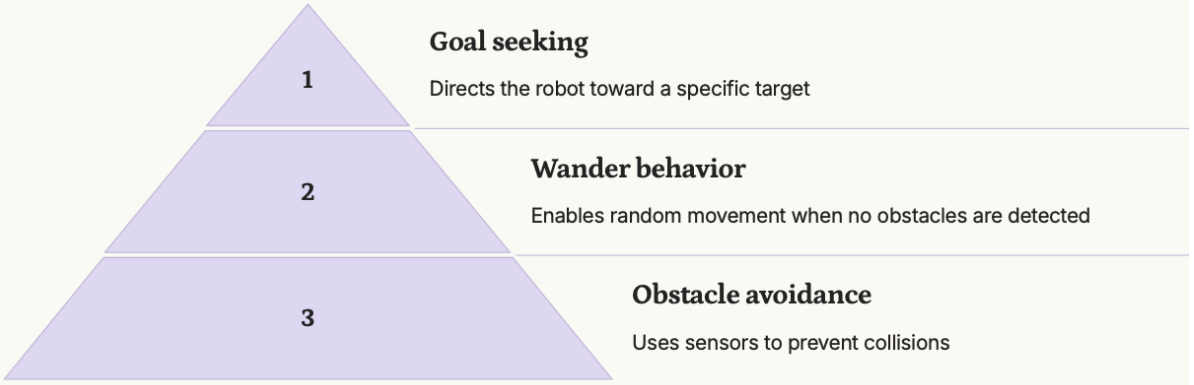
No central controller manages the entire system; behaviors are modular and self-contained

Emergent Behavior

Complex and intelligent actions emerge from the interaction of simple behaviors and the environment.

The sum is greater than the parts.

Consider a robot navigating a room:



Goal seeking

Directs the robot toward a specific target

Wander behavior

Enables random movement when no obstacles are detected

Obstacle avoidance

Uses sensors to prevent collisions

If an obstacle is detected, the obstacle avoidance layer overrides (or subsumes) the higher-level behaviors (wandering or goal seeking), ensuring safe operation.

- cada camada é implementada como um autómato finito