

Sistemas Basados en Comportamientos

CI-2657 Robótica

Dra. Kryscia Ramírez Benavides







Definiciones

- 🤖 **Comportamiento individual.** Es un par estímulo-respuesta para un ajuste del medio ambiente, dado que es modulado por la atención y determinado por la intención.
- 🤖 **Atención.** Prioriza las tareas y se centra en los recursos sensoriales y está determinada por el contexto del entorno actual.
- 🤖 **Intención.** Determina qué conjunto de comportamientos deben ser activados basados en las metas y objetivos internos del agente robótico.







Definiciones (cont.)

-  **Comportamiento emergente.** El comportamiento global observado; una consecuencia de la interacción de los comportamientos individuales activos.
-  **Conducta reflexiva (comportamiento puramente reactivo).** El comportamiento que se genera por comportamientos reactivos mediante conexiones entre sensores y efectores.





Comportamiento Animal

Ideas

-  El comportamiento animal define la inteligencia (o estupidez).
-  La prueba de que el comportamiento inteligente es posible.
-  Puede proporcionar modelos que pueden ser utilizados para crear máquinas inteligentes.
-  Los estudios biológicos no están vistos, necesariamente, como limitante para los robots.


Problemas.

-  Hardware biológico es diferente.
-  Nuestro conocimiento del funcionamiento del hardware biológico es a menudo insuficiente.



Estudios del Comportamiento Animal

Neurociencia

 El estudio de la anatomía del sistema nervioso, la fisiología, la bioquímica y la biología molecular.

Psicología

 El estudio de la mente y el comportamiento.

Etología

 El estudio del comportamiento animal en condiciones naturales.



Comportamiento del Robot en Sistemas Reactivos

- 🤖 **En el control reactivo:**
 - 🤖 La percepción y la acción están estrechamente acopladas, normalmente en el contexto de conductas motoras.
 - 🤖 Produce una respuesta robótica oportuna en mundos dinámicos y no estructurados.
 - 🤖 Sin el uso de intervención en la representación abstracta o historial de tiempo.



Sistema Robótico Puramente Reactivo

- 🤖 Los comportamientos sirven como bloques de construcción básicos para las acciones robóticas.
- 🤖 Uso de conocimiento explícito de la representación abstracta se evita en la generación de una respuesta.
- 🤖 Los modelos animales de comportamiento a menudo sirven como base.
- 🤖 Intrínsecamente modular desde la perspectiva del diseño de software.



Robótica Basada en el Comportamiento

- 🤖 La robótica basada en el comportamiento surgió del reconocimiento de que la planificación, no importa cuán intencionada, es a menudo una pérdida de tiempo.
- 🤖 Los sistemas robóticos basados en el comportamiento proporcionan un medio para que un robot navegue en un mundo incierto e imprevisible sin planificación, dotando al robot con los comportamientos que tienen que ver con los objetivos específicos de forma independiente y coordinarlos de manera intencionada.



Robótica Basada en el Comportamiento (cont.)

- 🤖 Los comportamientos son el módulo subyacente del sistema.
- 🤖 La descomposición del comportamiento:
 - 🤖 Los sistemas consisten en módulos secuenciales logrando funciones independientes.



Comportamiento Robótico

- 🤖 Generar una respuesta motora de un determinado estímulo perceptual.
- 🤖 Base en los estudios biológicos.
 - 🤖 Sirve como fuente de inspiración para el diseño.



Comportamiento vs Acción

Comportamiento

- 🤖 Basado en proceso dinámico.
 - 🤖 Funcionamiento en paralelo.
 - 🤖 La falta de un control central.
 - 🤖 Acoplamientos rápidos entre los sensores y motores.
- 🤖 Explotación de emergencia.
 - 🤖 Efectos secundarios de los procesos combinados.
 - 🤖 Utilización de las propiedades del entorno.
- 🤖 Reactivo.

Acción

- 🤖 Discreto en el tiempo.
 - 🤖 Bien definidos puntos inicial y final.
 - 🤖 Permite pre y post condiciones.
- 🤖 No hay efectos secundarios.
 - 🤖 Sólo una acción o acciones de unos pocos a la vez.
 - 🤖 Los conflictos no son deseados y se evitan.
- 🤖 Deliberativo.



Reactividad

- 🤖 Los comportamientos sirven como bloques de construcción para las acciones.
- 🤖 La representación abstracta se evita.
- 🤖 A menudo, el modelo de comportamiento de los animales.
- 🤖 Intrínsecamente modular.



Estímulos

- 🤖 La presencia de estímulos es necesaria pero no suficiente en la robótica basada en el comportamiento.
- 🤖 El estímulo debe alcanzar el valor umbral antes de que se genera una respuesta.



Representación

- 🤖 Los comportamientos pueden ser representados / almacenados en una red, con relaciones entre ellos.
- 🤖 Multiplicador de fuerza, o ganancia, puede desactivar las conductas o aumentar la respuesta.



Propiedades de los Comportamientos

- 🤖 Lograr tareas / objetivos específicos.
- 🤖 Se ejecuta en paralelo / concurrente.
- 🤖 Se puede almacenar el estado y ser usado para construir modelos / representación del mundo.
- 🤖 Se puede conectar directamente sensores y efectores (es decir, tomar las entradas de los sensores y enviar salidas a los efectores).
- 🤖 Puede tomar los aportes de otros comportamientos y enviar sus productos a otros comportamientos (esto permite la creación de redes).
- 🤖 Por lo general acciones de alto nivel.
- 🤖 Controladores de retroalimentación.
- 🤖 Normalmente circuito cerrado.
- 🤖 Cuando está montado en representaciones distribuidas, los comportamientos pueden ser utilizados para mirar hacia adelante, pero en una escala de tiempo comparable con el resto del sistema basado en el comportamiento.






Clasificación Genérica de Comportamientos del Robot

- Comportamientos de exploración / direccionales.
- Orientadas hacia metas de conductas apetitivas.
- Comportamientos aversivo / de protección.
- Comportamientos siguiendo ruta.
- Conductas posturales (equilibrio, estabilidad).
- Conductas sociales / cooperativa.
- Comportamientos teleautonomous (influencia, modificación del comportamiento).
- Conductas perceptivas.
- Comportamientos de caminar (marcha de control).
- Manipulador de comportamientos específicos.
- Comportamientos de mano / pinza (agarrar, envolver).



Propiedades Principales

-  Capacidad para actuar en tiempo real.
-  Capacidad para utilizar representaciones y generar comportamiento eficiente (y no sólo reactivo).
-  Capacidad para utilizar una estructura uniforme y una representación en todo el sistema (por lo que no hay capa intermedia).



Tipos de Notación de Comportamientos

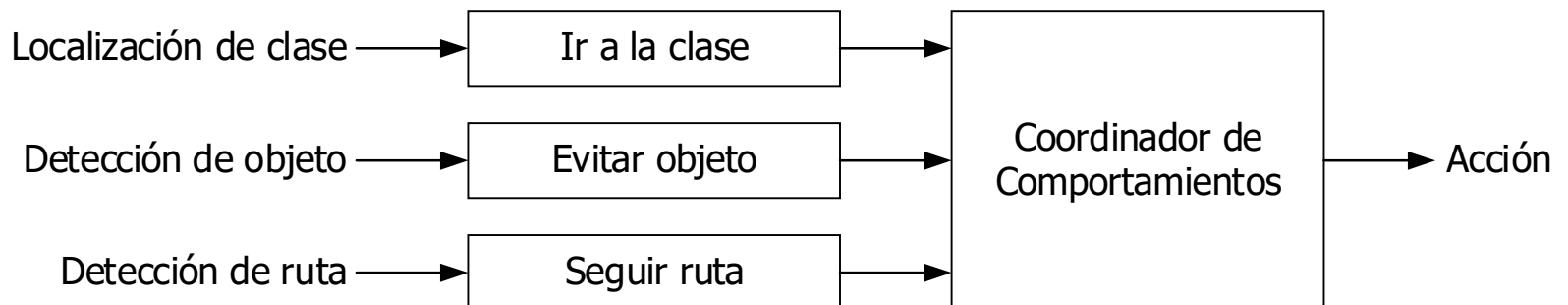
- 🤖 Diagramas de estímulo-respuesta.
- 🤖 Comportamientos de notación funcional.
- 🤖 Diagramas de aceptación de estados finitos.

Diagramas de Estímulo-Respuesta

- Los diagramas de estímulo-respuesta (SR) son los más intuitivos y el método menos formal de expresión.



- Ejemplo: Diagrama de estímulo-respuesta para un Robot que va a un aula





Comportamientos de Notación Funcional

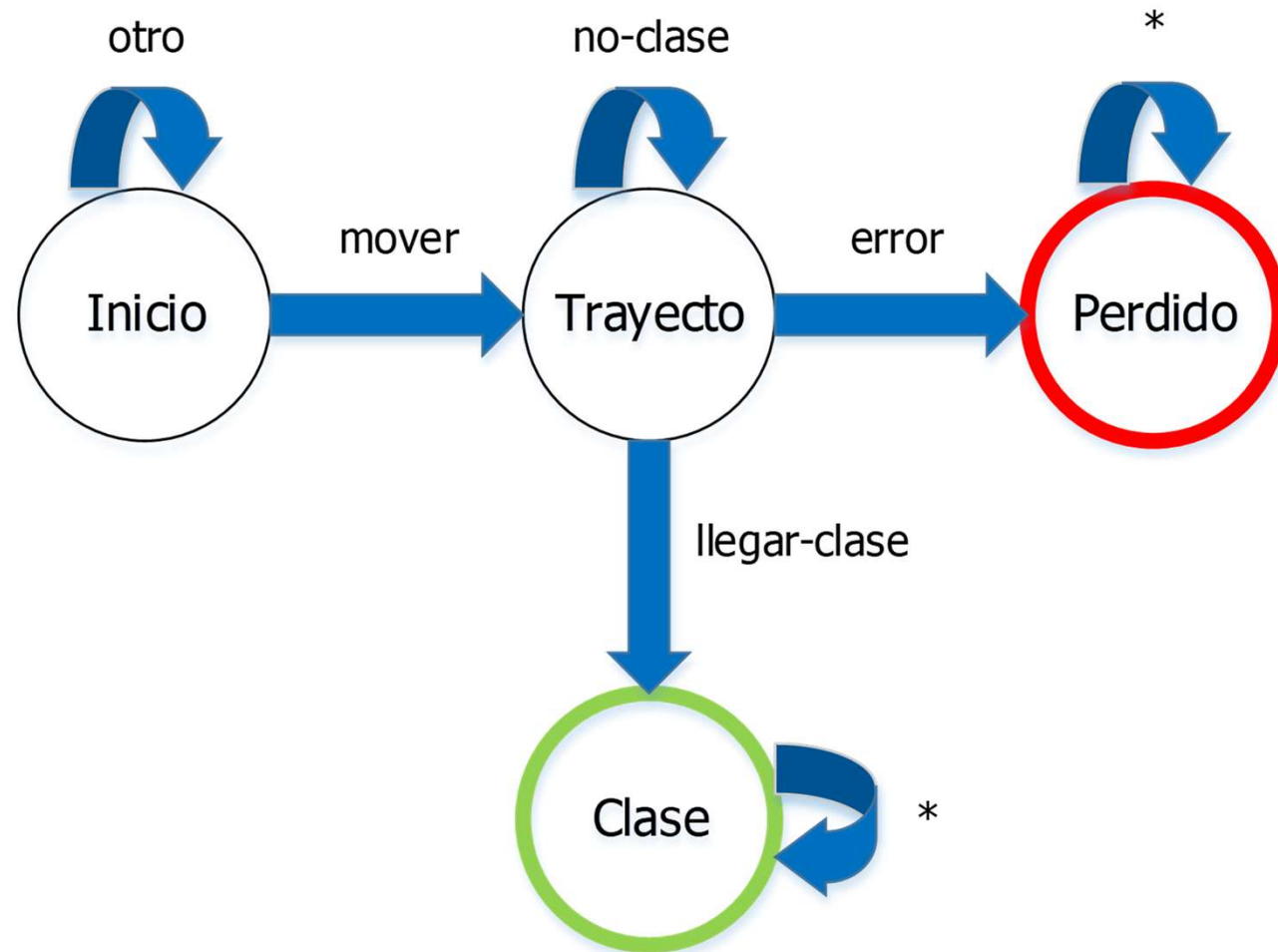
- 🤖 Podemos utilizar la notación funcional para describir comportamientos: $b(s) = r$.
 - 🤖 Donde b es el comportamiento, s es el estímulo y r la respuesta.
- 🤖 En un sistema puramente reactivo, el tiempo no es un argumento para la conducta b .
- 🤖 Ejemplo:
 - coordinar-comportamientos [
ir-a-clase (localizacion-deteccion-clase)
evitar-objeto (deteccion-clase)
esquivar-estudiante (deteccion-estudiante)
seguir-ruta (deteccion-ruta)
] = respuesta-motor



Diagramas de Aceptación de Estados Finitos

- 🤖 Diagramas de estados finitos (FSD) o aceptor de estados finitos (FSA) se pueden utilizar para describir agregaciones y las secuencias de comportamientos (también llamado como ensambles de comportamiento).
- 🤖 FSA son los mejores para especificar sistemas complejos de control de la conducta en grupos completos de comportamientos primitivos que se intercambian dentro y fuera de su ejecución durante el cumplimiento de una meta de alto nivel.
- 🤖 En algunos casos, los lenguajes formales, como el lenguaje de especificación y descripción (SDL) se puede utilizar para crear / generar código para las implementaciones reales.

Diagramas de Aceptación de Estados Finitos (cont.)






Comportamientos Emergentes

- 🤖 **Comportamiento emergente:**
 - 🤖 A menudo, una cosa mística, el robot parece hacer más que lo programado! Una verdadera sorpresa: la suma es considerablemente mayor que sus partes!
- 🤖 **¿Por qué no podemos predecir el comportamiento exactamente?**
 - 🤖 El mundo mismo resiste un modelado analítico.
 - 🤖 Modelos con sensores precisos no existen.
- 🤖 **El comportamiento emergente es una consecuencia de la complejidad subyacente del mundo en el que el agente robótico reside y la complejidad adicional de percibir ese mundo.**

Características Comunes de las Arquitecturas Basadas en el Comportamiento

- 
- 🤖 Hay muchas arquitecturas basadas en el comportamiento, pero comparten muchas características comunes:
 - 🤖 Énfasis en la importancia de la detección de acoplamiento y la acción hermética.
 - 🤖 Evitar el conocimiento simbólico representacional.
 - 🤖 El sistema global de control se divide en unidades.
 - 🤖 Pero también hay diferencias:
 - 🤖 La granularidad de la descomposición del comportamiento.
 - 🤖 La base de la especificación del comportamiento (actividad etológica, situado, experimental).
 - 🤖 El método de codificación de la respuesta (*null*, discreto y continuo).
 - 🤖 El método de coordinación (competitivo o cooperativo).
 - 🤖 Métodos de programación, lenguajes, reusabilidad.

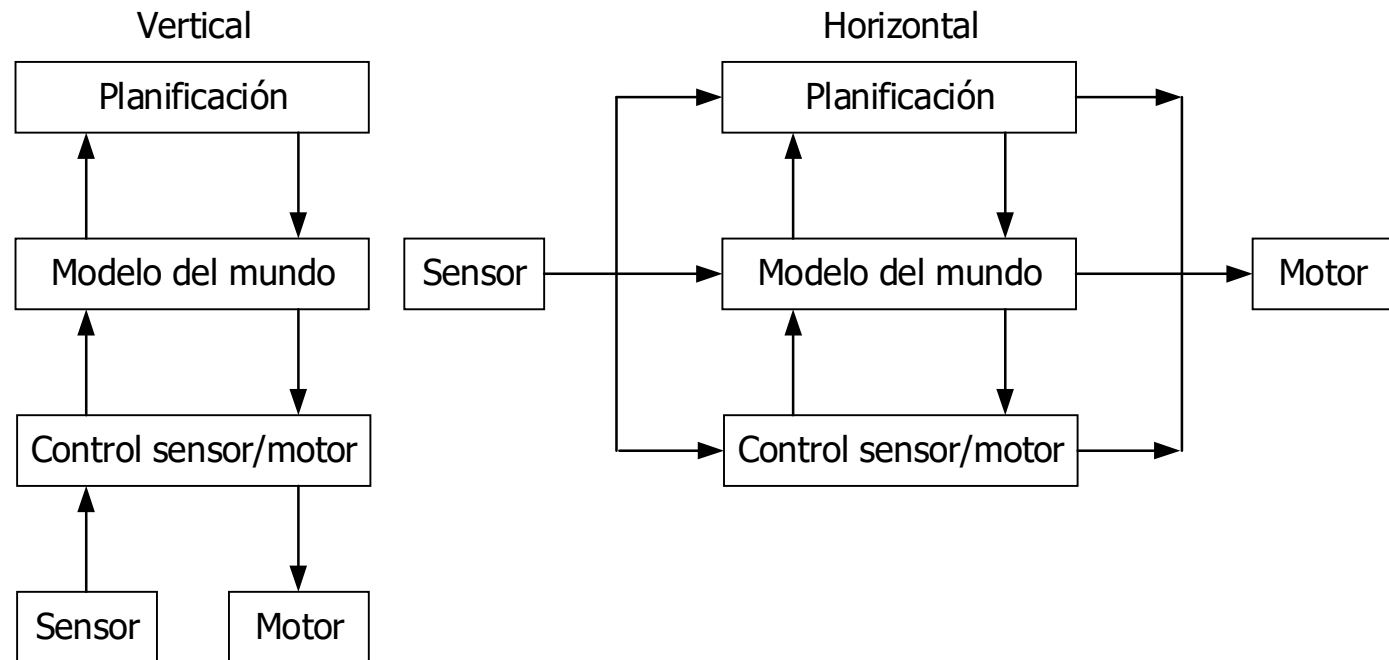


Evaluación de las Arquitecturas Basadas en el Comportamiento

- 🤖 ¿Apoyo para el paralelismo?
- 🤖 ¿Capacidad del hardware de destino?
¿Mapeo de SW a HW?
- 🤖 ¿Capacidad del lugar de destino?
- 🤖 ¿Soporte a la modularidad? ¿Encapsular abstracciones del comportamiento?
- 🤖 ¿Robustez? ¿Tolerancia a fallos?
- 🤖 ¿Puntualidad en el desarrollo? ¿Herramientas de desarrollo? ¿Simuladores?
- 🤖 ¿Flexibilidad en tiempo de ejecución?
- 🤖 ¿Efectividad del rendimiento?

Arquitectura de Subsunción

- Logro de tareas de comportamientos en la arquitectura de subsunción se representan como capas separadas que funcionan en paralelo.





Arquitectura de Subsunción (cont.)

- 🤖 Descomposición de controladores en **niveles de competencia.**
- 🤖 Añade comportamientos incrementalmente.
- 🤖 Las capas de alto nivel se sustentan en las de bajo nivel.
- 🤖 Las capas de alto nivel pueden **subsumir** las de bajo nivel.
- 🤖 Las capas se activan en **paralelo.**
- 🤖 Cada capa trabaja **independientemente.**
- 🤖 **No** existe un control **jerárquico.**
- 🤖 Acoplamiento directo entre los sensores y efectores con un pequeño procesamiento interno.



El Mejor Modelo es el Mundo

- 🤖 Este es un principio fundamental de los sistemas reactivos y arquitectura de subsunción:
 - 🤖 El mundo es el mejor modelo.
 - 🤖 Si el mundo puede proporcionar la información directamente (a través de sensores), lo mejor es conseguirlo de esa manera, que almacenar internamente una representación (que puede ser grande, lenta, costosa y anticuada).

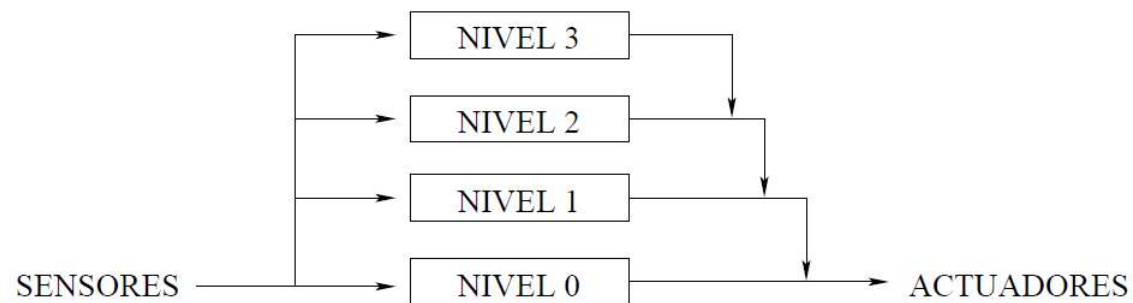


Proceso de Diseño de la Arquitectura de Subsunción

- 🤖 Especificar cualitativamente el comportamiento global necesario para la tarea.
- 🤖 Descomponer en comportamientos específicos e independientes (capas).
- 🤖 Las capas deben ser de abajo hacia arriba y consistir en acciones inconexas.
- 🤖 Comportamientos de bajo nivel en los sensores del robot y efectores.
- 🤖 Incremental en la manera de construir, probar y agregar.

Niveles de competencia

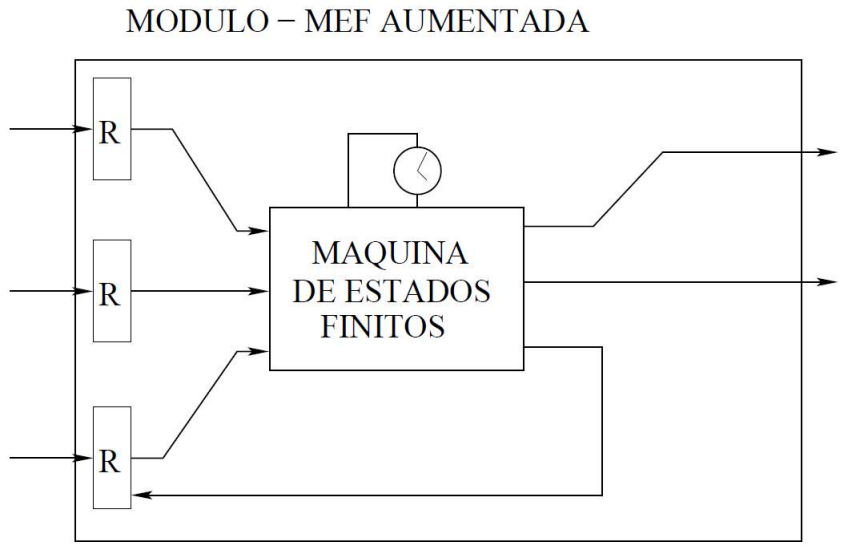
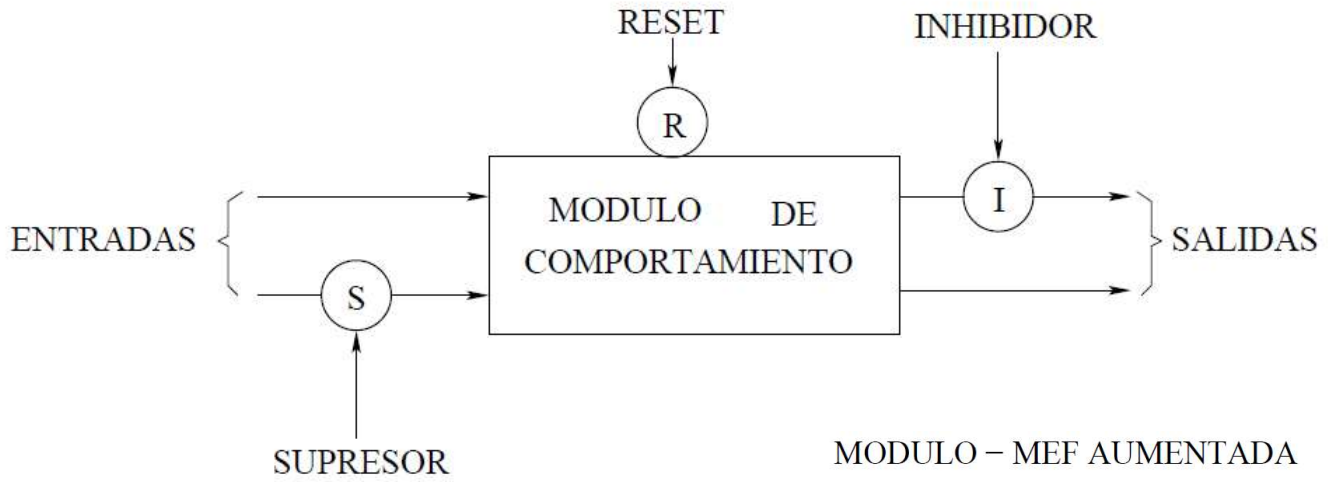
- 🤖 Define los comportamientos deseados: Evitar Obstáculos, Explorar, traer comida, etc.
- 🤖 Cada nivel se puede implementar independientemente.
- 🤖 Permite una gran escalabilidad.
- 🤖 Compuestos de varios módulos.





Módulos

🤖 Máquinas de estados finitos aumentadas.





Módulos (cont.)

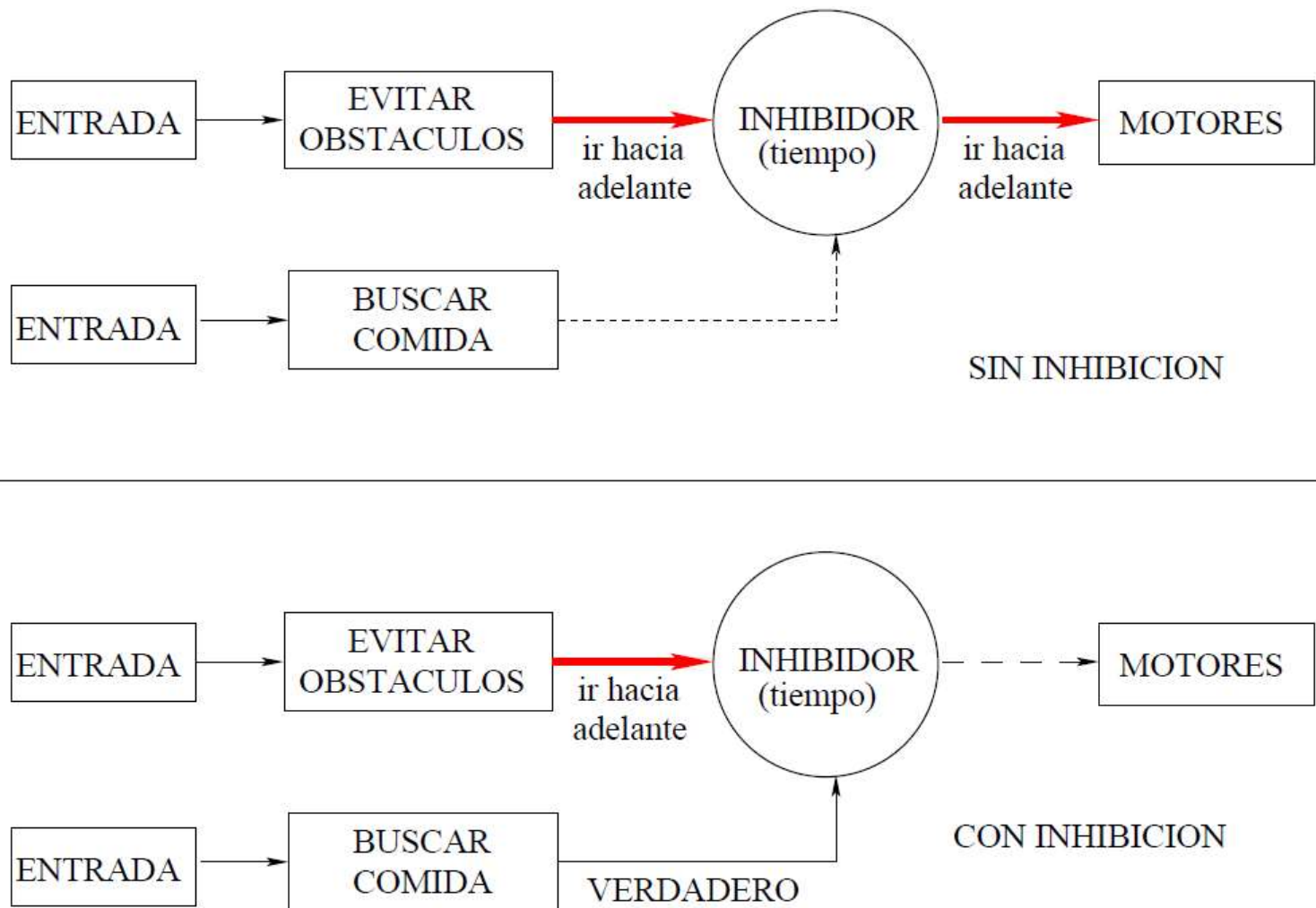
- 🤖 Los aspectos más importantes son:
 - 🤖 Las **salidas** son **funciones** sencillas de las **entradas** y variables locales.
 - 🤖 Las entradas/salidas se pueden inhibir/suprimir.
 - 🐞 Los módulos de los niveles de competencia superiores pueden inhibir a los inferiores.
 - 🐞 Los niveles de competencia inferiores siguen funcionando como si no existieran los niveles superiores.



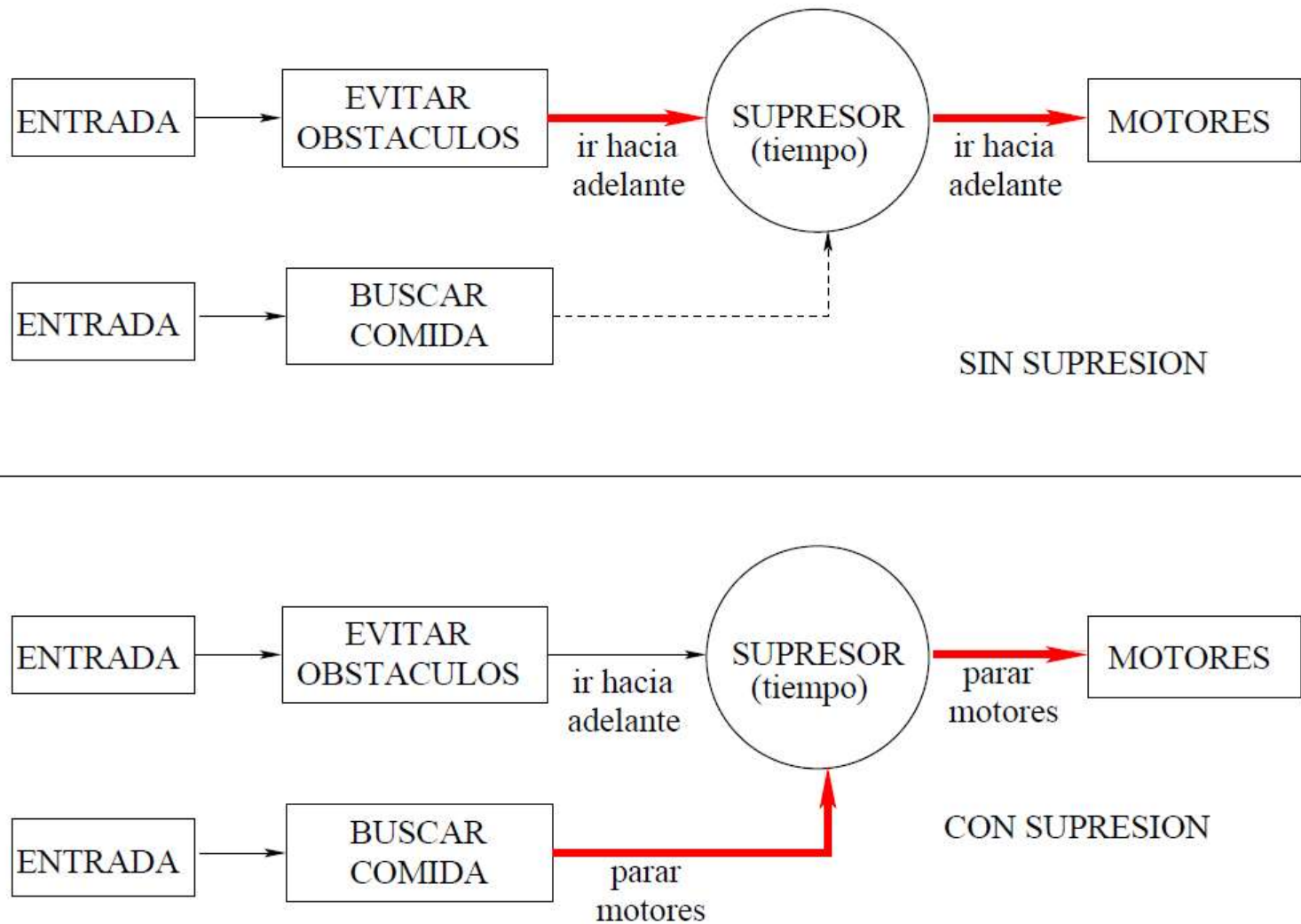
El Arbitraje en Subsunción

- 🤖 **Arbitraje.** Decidir quién tiene el control.
- 🤖 **Inhibición.** Evita que las señales de salida lleguen a los efectores.
- 🤖 **Supresión.** Reemplaza la señal de entrada con el mensaje de la supresión.
- 🤖 Las dos anteriores son los únicos mecanismos de coordinación.
 - 🤖 Resultados de prioridad basado en el arbitraje.
 - 🤖 La regla o una capa con una prioridad más alta se hace cargo, es decir, tiene el control de la AFSM.

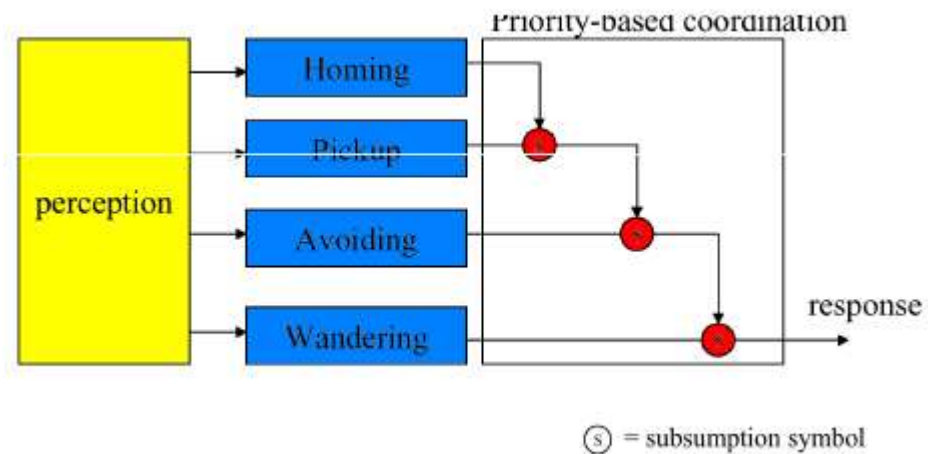
Inhibición



Supresión



Ejemplo: Diagrama para un robot de búsqueda de alimento





Subsunción

Interacciones más generales de comportamiento

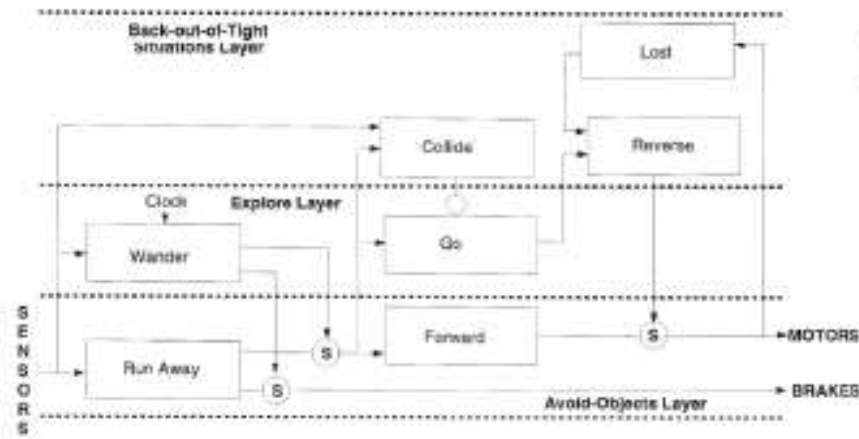
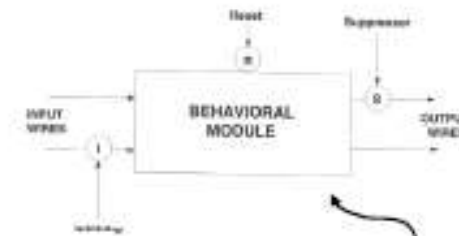


Figure 4.5
AFSMs for a simple three-layered robot (Brooks 1987b).







Los bloques de construcción

Arbitra entre las capas de una manera ad-hoc





Evaluación de Subsunción

Fortalezas:

-  Reactividad (velocidad, tiempo real de la naturaleza).
-  Paralelismo.
-  Diseño incremental => robustez.
-  Generalidad.

Debilidades:

-  Falta de flexibilidad en tiempo de ejecución.
-  Experiencia necesaria en el diseño.



Comportamiento Emergente

- 🤖 Fenómeno importante, pero no se entiende bien.
- 🤖 Se encuentra a menudo en robótica basada en el comportamiento.
- 🤖 Comportamientos del robot "emergen" de
 - 🤖 Las interacciones de las normas.
 - 🤖 Las interacciones de los comportamientos.
 - 🤖 Las interacciones de ambos con el medio ambiente.



Distinción

- 🤖 Codificando comportamiento.
 - 🤖 En el esquema de programación.
- 🤖 Comportamiento observado.
 - 🤖 En los ojos del observador.
 - 🤖 Emerge.
- 🤖 No hay asignación de uno-a-uno entre los dos.
- 🤖 ¿Por qué?
 - 🤖 La suma es mayor que las partes.
 - 🤖 El comportamiento emergente es más que el controlador que produce.

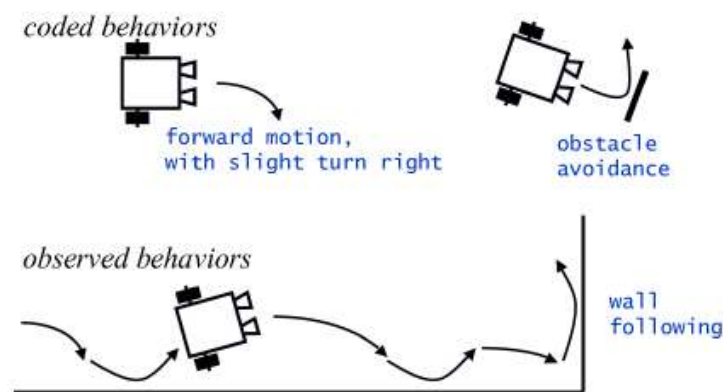


Interacción y Emergencia

- 🤖 Las interacciones entre las normas, comportamientos y medio ambiente.
- 🤖 Fuente de poder expresivo de un diseñador.
- 🤖 Es decir, los sistemas pueden ser diseñados para tomar ventaja del comportamiento emergente.

Ejemplo: Siguiendo a la pared

- 🤖 Puede ser implementado con estas reglas:
 - 🤖 Si está demasiado lejos, moverse más cerca.
 - 🤖 Si está muy cerca, se aleja.
 - 🤖 De lo contrario, sigue.
- 🤖 Con el tiempo, en un entorno con paredes, esto se traducirá en la pared siguiente.





Ejemplo: Siguiendo a la pared (cont.)

- 🤖 ¿Es este comportamiento emergente?
- 🤖 Se argumenta que sí porque:
 - 🤖 El robot en sí mismo no tiene conocimiento de una pared, sólo reacciona a lecturas de distancia.
 - 🤖 Los conceptos de "muro" y "siguiente" no se almacenan en el controlador del robot.
 - 🤖 El sistema es sólo una colección de reglas.



Referencias Bibliográficas

- 🤖 Fu, K.S.; González, R.C. y Lee, C.S.G. Robotics: Control, Sensing, Vision, and Intelligence. McGraw-Hill. 1987.





¡Gracias!



Dra. Kryscia Daviana Ramírez Benavides
Profesora e Investigadora
Universidad de Costa Rica
Escuela de Ciencias de la Computación e Informática

Sitio Web: <http://www.kramirez.net/>
E-Mail: kryscia.ramirez@ucr.ac.cr
kryscia.ramirez@eccu.ucr.ac.cr

Redes Sociales:

