



Universidad
de Huelva



Universidad de Huelva

GRADO EN INGENIERÍA INFORMÁTICA

TEMA 3. COMPORTAMIENTOS INTELIGENTES BÁSICOS

Resumen

Autor: Alberto Fernández Merchán
Asignatura: Sistemas Inteligentes

1. Tipos de Algoritmos

Existen dos niveles en el campo de la inteligencia artificial: El nivel **básico**, en el que se encuentran las técnicas de toma de decisiones reactivas y el nivel **avanzado**, donde se encuentran las técnicas de razonamiento y planificación.

1.1. Sin Búsqueda

En los algoritmos sin búsqueda se puede predecir el coste computacional. Estos son las reglas, los árboles de decisión.

El conocimiento se encuentra en la lógica del código o en tablas externas.

1.2. Con Búsqueda

En estos algoritmos el coste computacional depende del tamaño del espacio de búsqueda. Estos son los algoritmos de minimización y maximización, de planificación y de *pathfinding*.

El conocimiento se encuentra en la evaluación de los estados.

2. Sistemas basados en Reglas

Se conocen como sistemas expertos o sistemas de producción.

2.1. Componentes

- Memoria de Trabajo: En ella se guardan los hechos ciertos.
- Reglas de Producción: Son las reglas que permiten el funcionamiento.
- Motor de inferencia: Programa que calcula las consecuencias de aplicar las reglas.
- Resolución de conflictos: En caso de que se puedan aplicar varias reglas, decide cuál elegir.

2.2. Razonamiento

Se dice que una regla se dispara si sus antecedentes coinciden con los hechos de la base de hechos. El disparo de la regla añade a la base de hechos el consecuente de esta.

2.2.1. Encadenamiento hacia delante

Partiendo de un conjunto de antecedentes utilizamos la deducción para llegar a una conclusión.

2.2.2. Encadenamiento hacia atrás

A partir de una conclusión, la demostramos siguiendo un camino lógico desde la conclusión hasta un conjunto de antecedentes que se encuentran en la base de hechos.

2.3. Resolución de conflictos

Existen varias formas de solucionar los conflictos.

- Ordenar físicamente las reglas: Es difícil añadir reglas a estos sistemas.
- Ordenar los datos: Ordena los elementos del problema en colas de prioridad y utiliza la regla que se ocupa de los elementos más prioritarios.
- Especificidad o máxima especificidad: Elige la regla que tenga más antecedentes que coinciden.
- Selección aleatoria.

- Disparar todas las reglas.
- Buscar la regla más adecuada.

2.4. Ventajas y Desventajas

- Ventajas
 - Son fácilmente exportables a otros lenguajes.
 - Es un método muy expresivo.
 - Aplica un conocimiento modular.
 - Aportan legibilidad.
- Desventajas
 - No todo el conocimiento se ajusta al formato de las reglas
 - Debe existir un consenso de expertos
 - Puede requerir mucha memoria y capacidad de procesamiento.
 - Puede ser difícil de depurar
 - Requiere un conocimiento detallado.
 - Difícil olvidar hechos

3. Árboles de Decisión

3.1. Características

- Técnica de IA básica
- Fácil de implementar
- Ejecución Rápida

- Fácil comprensión

3.2. Creación

Se utiliza la programación orientada a objetos (POO).

Árboles de Decisión OO (Pseudo-Código)



```
class Node
  def decide() // return action/decision

class Decision : Node // interior
  def getBranch() // return a node
  def decide()
    return getBranch().decide()

class Action : Node // leaf
  def decide() return this
```

```
class Boolean : Decision // if yes/no
  yesNode
  noNode

class MinMax : Boolean // if range
  minValue
  maxValue
  testValue

def getBranch()
  if maxValue >= testValue >= minValue
    return yesNode
  else
    return noNode
```

```
// Define root as start of tree
Node *root

// Calls recursively until action
Action * action = root → decide()
action → doAction()
```

3.3. Funcionamiento

- Test de Nodos individuales (getBranch). Tiempo constante.
- Depende de la profundidad del árbol.
- Lo ideal es construir árboles balanceados.

4. Máquinas de Estado Finitas

4.1. Características

- Fácil representación, programación y depuración.
- Formalmente está compuesta por:
 - Conjunto de Estados
 - Estado Inicial
 - Alfabeto de entrada
 - Transiciones

4.2. Implementación

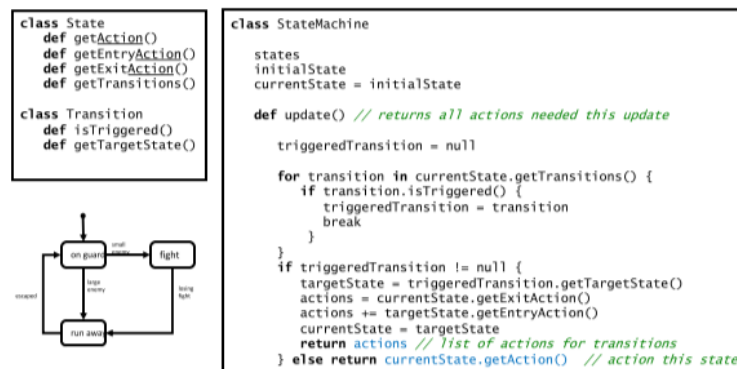


Figura 1: Implementación POO. Más flexible y limpia.

4.3. Combinar Árboles de Decisión con Máquinas de Estados

Se puede cambiar el conjunto de reglas por un árbol de decisión y así evitar comprobaciones duplicadas en las máquinas de estado.

5. Máquinas de Estado Jerárquicas

Tienen memoria de por donde se habían quedado ejecutándose.

