

Tema 3 Sistemas Expertos

**Razonamiento y Representación del
Conocimiento**

Introducción

¿Qué es un experto?

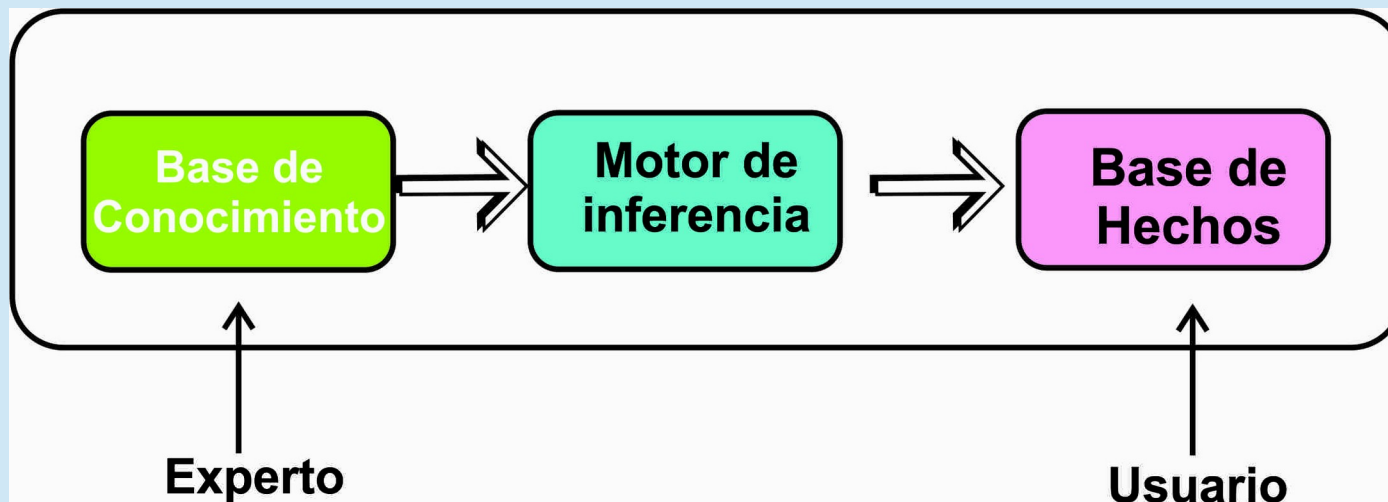
- Objetivo: clonar el conocimiento de un experto
- Los sistemas automáticos de deducción no funcionan (funcionaban) bien para cuestiones generales
- Concretar el campo de trabajo ofrece mejores resultados

Introducción

- Capacidades de un sistema experto
 - Resolver problemas difíciles y apoyar la toma de decisiones en base a un proceso de razonamiento simbólico
 - Soportar incertidumbre
 - Ofrecer una interfaz eficaz y cómoda (natural)
 - Incorporar la posibilidad de aprendizaje
 - Ofrecer soluciones alternativas y justificar la línea de razonamiento seguida para alcanzarlas

Esquema de un S.E.

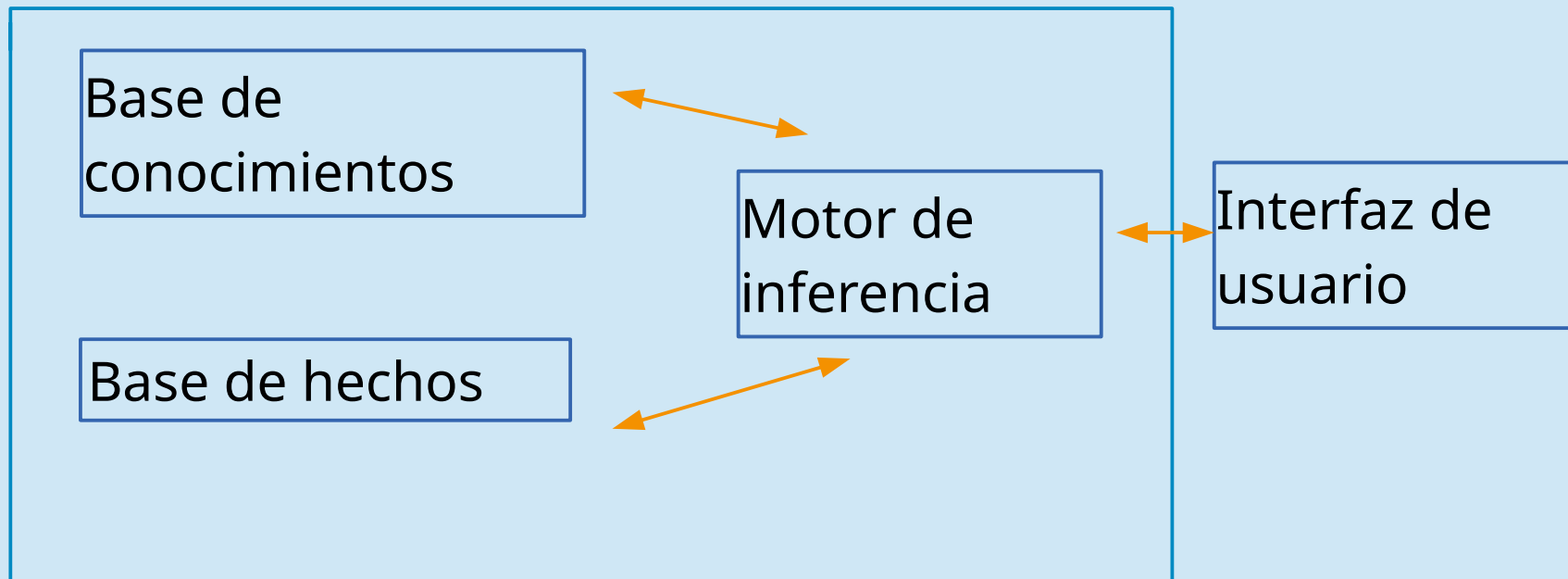
- Conocimiento del experto → Base de conocimiento
- Razonamiento → Motor de inferencia



Esquema de un S.E.

- Base de conocimiento
 - Contiene conocimiento altamente especializado
 - Proporcionado por el(los) experto(s) humano(s)
 - Hechos, reglas, conceptos y relaciones
- Motor de inferencia
 - Procesador del conocimiento
 - Obtiene conclusiones o recomendaciones

Arquitectura básica



Base de conocimiento

- Contiene el conocimiento sobre el dominio
- Típicamente reglas IF/THEN

Regla 1

IF el coche no arranca

THEN El problema puede estar en el sistema eléctrico

Regla 2

IF El problema puede estar en el sistema eléctrico

AND El voltaje de la batería está por debajo de 10 voltios

THEN El fallo es una batería defectuosa

Base de hechos

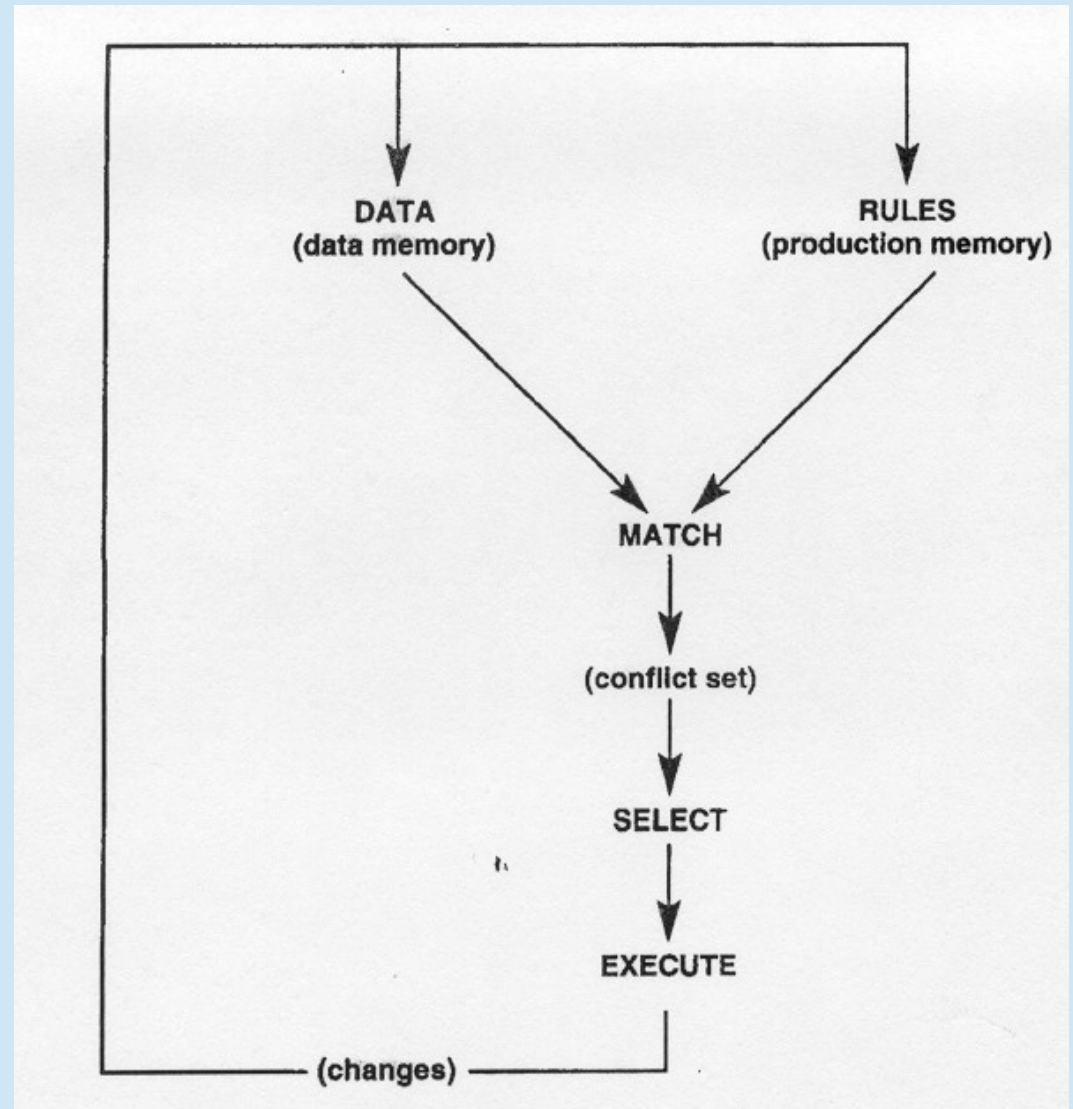
- También llamada memoria de trabajo
- Contiene los hechos que se han descubierto durante una consulta
- Durante una consulta el usuario* introduce información del problema actual en la base de hechos

Motor de inferencia

- Realiza el razonamiento a partir de los hechos y el conocimiento en la base de conocimiento: las reglas
- Deduce nuevos hechos
- Obtiene conclusiones

Motor de inferencia

1. Buscar las reglas cuya parte izda. Se cumpla según los hechos → búsqueda muchos-a-muchos
2. Se forma el conjunto de conflicto
3. Se escoje una regla usando algún tipo de heurística
4. Se ejecuta la regla y se modifica la base de hechos o la base de conocimiento acorde a su parte derecha
5. Volvemos al paso 1



Motor de inferencia

- Problema:
 - La búsqueda muchos-a-muchos es lenta
 - Puede ralentizarse tanto que haga imposible aplicarse para problemas reales con muchas reglas y muchos hechos

Motor de inferencia

- Solución

1. Sistemas de índices y listas enlazadas → ayudan a reducir (algo) el tiempo
2. El algoritmo de *matching* Rete → reduce el tiempo de búsqueda de satisfacción de la parte izda de las reglas en varios órdenes de magnitud incrementando el consumo de memoria

Algoritmo de matching Rete

- Características
 - Constituye una red de nodos → grafo dirigido acíclico
 - Cada nodo representa un hecho en la parte izda de una regla
 - Explota las **redundancias** en la parte izda de las reglas para construir una red compacta de nodos
 - Mantiene la evidencia (hechos) en los nodos

Algoritmo de matching Rete

- ¿Cómo funciona?
 - Veamos un ejemplo: un sistema experto de una fábrica de golosinas. Dos de las reglas de producción:
 - Regla Red__Round__Ones: Si el objetivo es identificar una golosina, y si la golosina es roja, y si la golosina es redonda, entonces es una gominola
 - Regla Red__cylindrical__Ones: Si el objetivo es identificar una golosina, y si la golosina es roja, y si la golosina es cilíndrica, entonces es una barra de regaliz

Algoritmo de matching Rete

- Ejemplo. Formalizándolo en un lenguaje lógico (OPS5)
 - Reglas de producción

```
(P Red__Round__Ones
  (Goal | Type Identify | Object {N})
  (Candy | Name {N} | Color Red | Shape Round | Company {X})
->   then clause, which is not important in this context)

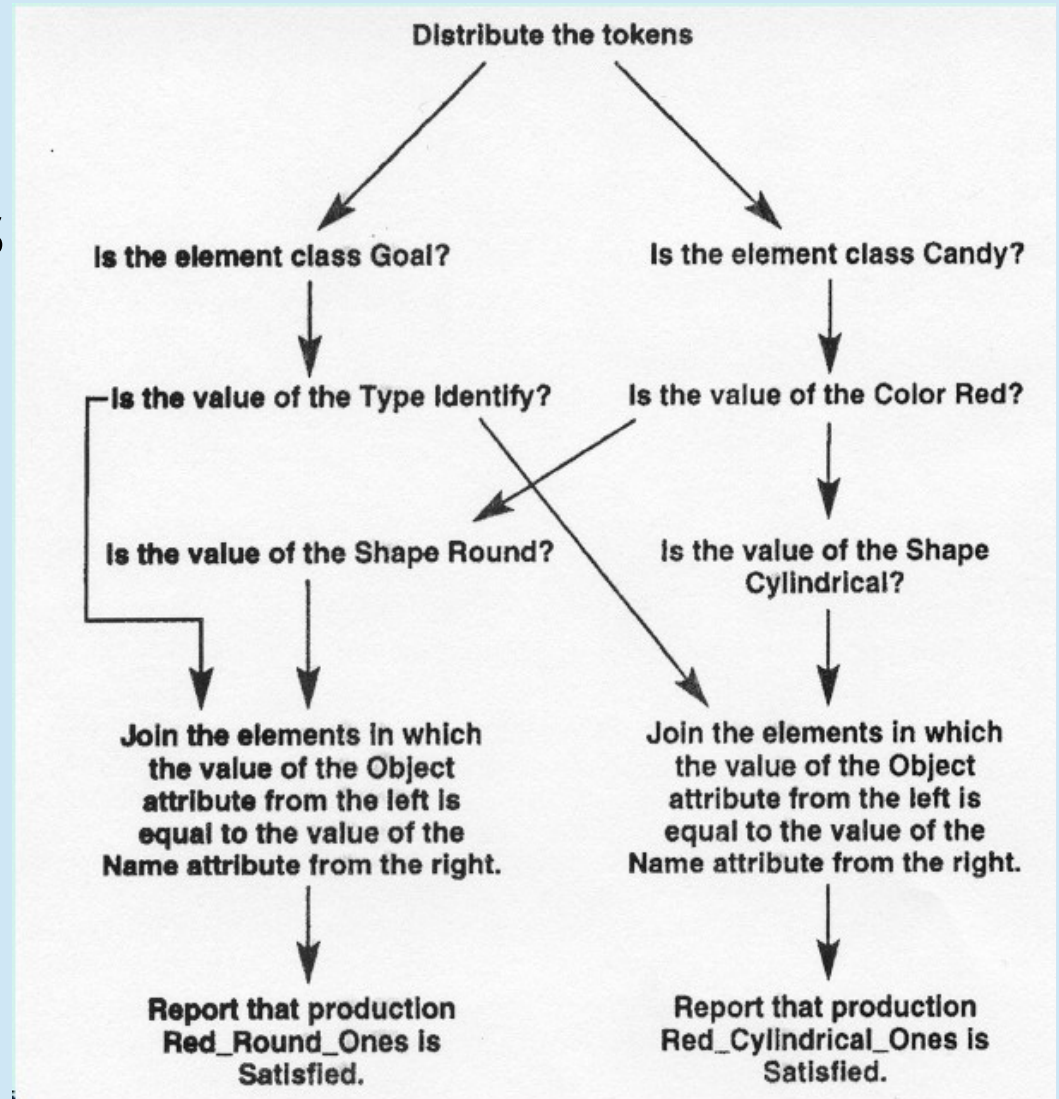
(P Red__Cylindrical__Ones
  (Goal | Type Identify | Object {N})
  (Candy | Name{N} | Color Red | Shape Cylindrical | Company {X})
->   then clauses
```

- Ejemplo de memoria de datos

```
1 (Goal | Type Identify | Object Sample7)
2 (Candy | Name Sample7 | Color Red | Shape Round | Company FerraraPan)
```

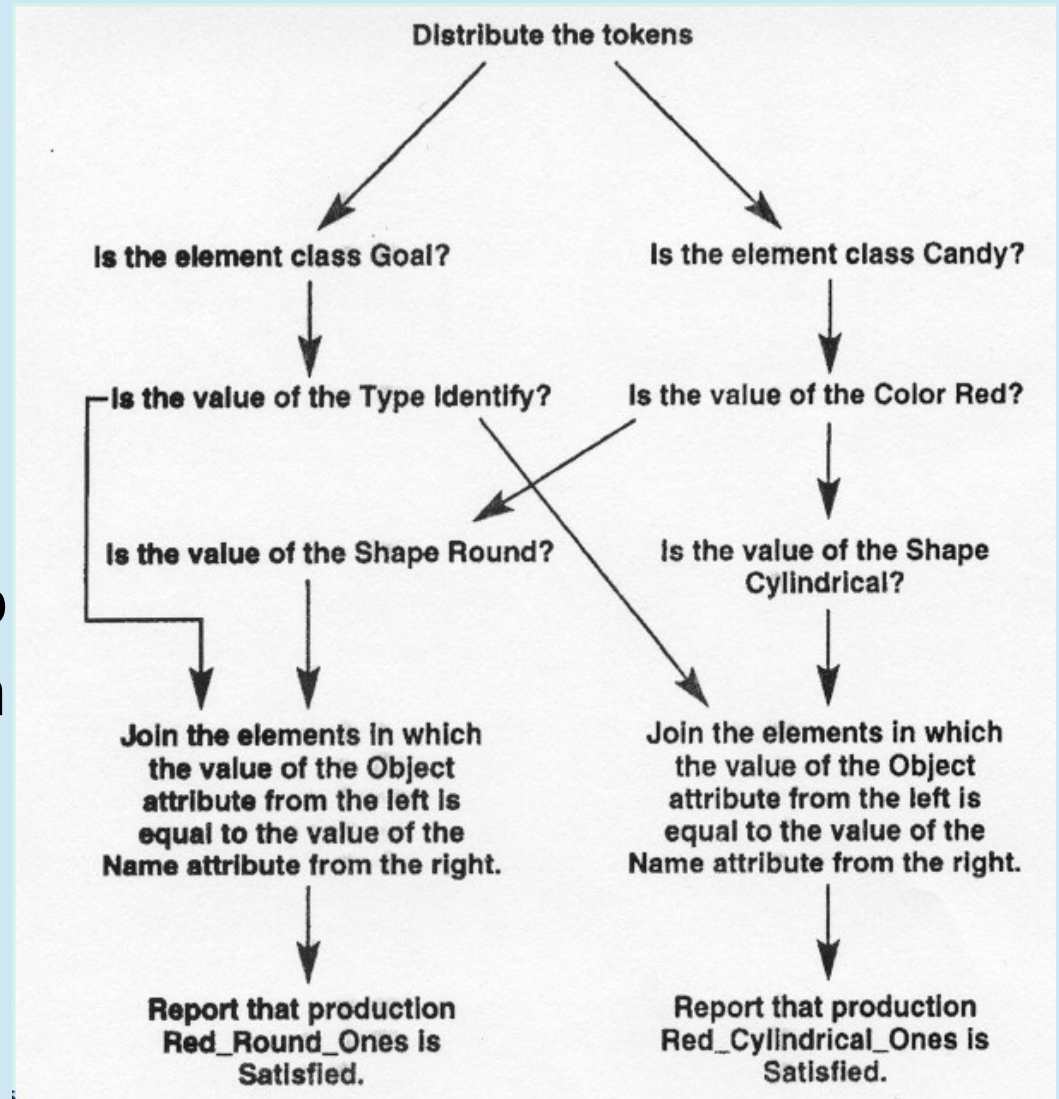
Algoritmo de matching Rete

- Red de nodos
 - Dos tipos de nodos
 - Con 1 entrada
 - Con 2 entradas
 - Las hojas indican que la regla de producción se satisface



Algoritmo de matching Rete

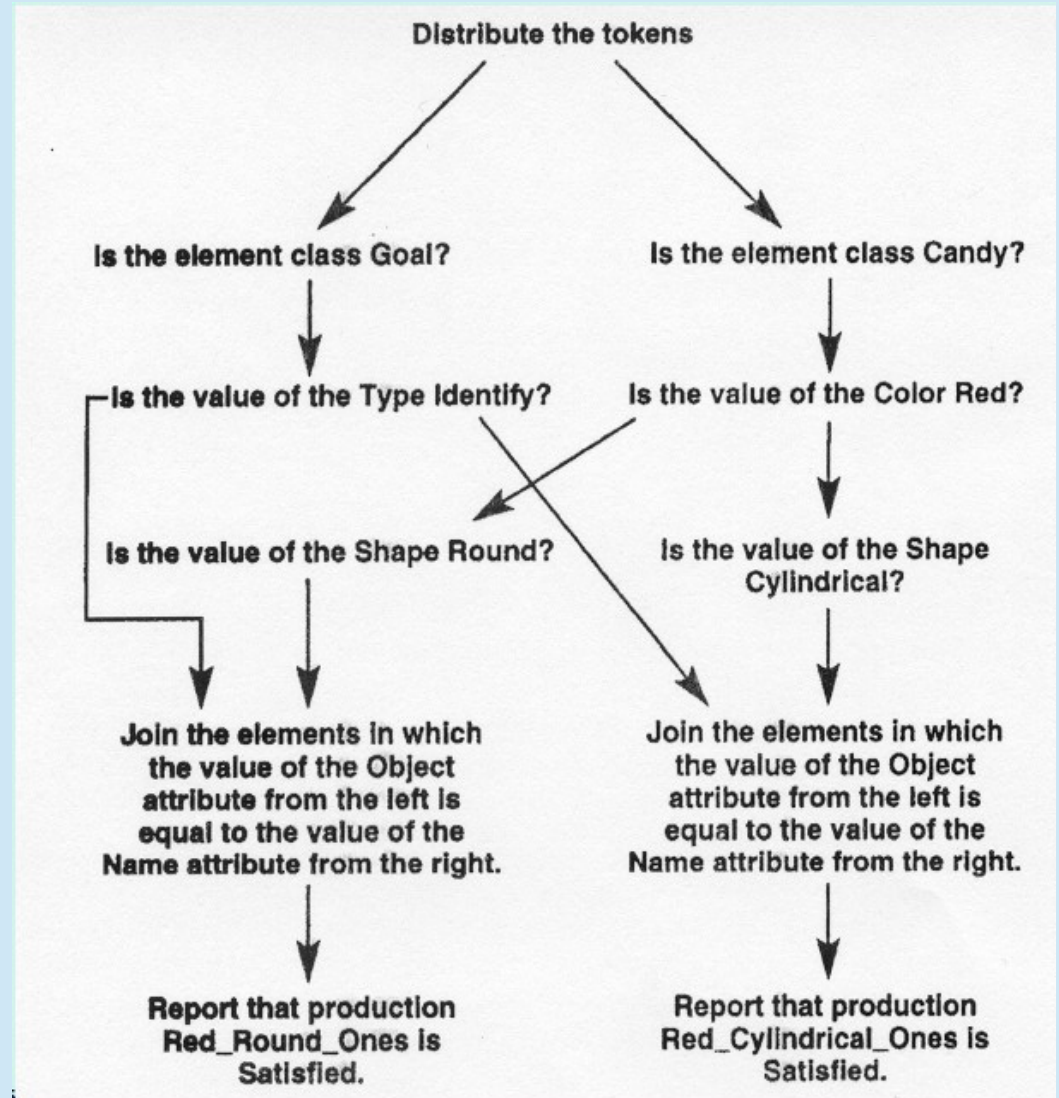
- ¿cómo funciona?
 - Los nodos de 1 entrada evalúan su atributo
 - Los de 2 entradas hacen lo mismo, pero si solo llega evidencia por una entrada la guardan para más adelante



Algoritmo de matching Rete

- ¿cómo funciona?

1. (Goals | Type Identify | Object Sample7)
2. (Candy | Name Sample8 | Color Red | Shape Round | Company FerraraPan)
3. (Goal | Type Identify | Object Sample8)



Otras partes de un Sistema Experto

- Subsistema de explicación
- Las conclusiones deben ser explicadas
- Proporciona información de por qué plantea nuevas preguntas al usuario y de cómo ha llegado a sus conclusiones
- Interfaz de usuario
- La interacción se ha de producir en lenguaje natural
- Habilidad para hacer preguntas
- Presentar resultados gráficamente

Sistema experto vs experto humano

Factor	Experto humano	Sistema experto
Disponibilidad de tiempo	Día laborable	Siempre
Geografía	Local	Cualquier lugar
Seguridad	Irremplazable	Reemplazable
Caducidad	Sí	No
Ejecución	Variable	Constante
Velocidad	Variable	Constante
Coste	Alto	Abordable

Tipos de Sistemas Expertos

- Control: gobiernan el comportamiento de un sistema
- Diseño: configuran objetos bajo un conjunto de restricciones
- Diagnóstico: deducen fallos del sistema desde información observable
- Instrucción: guían la educación de un estudiante en un tema dado
- Interpretación: producen una descripción de una situación desde información disponible, normalmente ruidosa o incompleta

Tipos de Sistemas Expertos

- Monitorización: comparan información del comportamiento de un sistema con estados “cruciales” en su operación
- Planificación: diseñan un plan de acciones para realizar una meta dada
- Predicción: deducen consecuencias probables desde una situación dada y un modelo del problema
- Selección: identifican la mejor elección de una lista de posibilidades
- Simulación: Modelan un proceso o sistema para permitir estudios operacionales bajo diversas condiciones

Fases en el desarrollo de un Sistema Experto

- Cualquier desarrollo software tiene las siguientes fases:
 - Investigación → Diseño → Prueba → Documentación → Mantenimiento
- Un SE requiere de una fase adicional: **Adquisición del conocimiento**, que se realizará en las etapas iniciales
 - Adquirir el conocimiento necesario sobre el problema
 - Descubrir los conceptos clave y los métodos de resolución de problemas empleados por los expertos

Ventajas de los Sistemas Expertos

- Reutilización del motor de inferencia para distintos problemas
- Facilita el proceso de aprendizaje de nuevo conocimiento experto (y se puede automatizar)
- Facilita el manejo de conocimiento incompleto
- Se centra en un problema específico apartando problemas referentes a otros conocimientos
- Lo ya comentado: permanencia, transportabilidad, coherencia, coste, estabilidad

Inconvenientes de los Sistemas Expertos

- Resolución y búsqueda menos eficientes al separar el conocimiento sobre el dominio y sobre el proceso
- No capturan* algunas buenas cualidades de los expertos humanos: creatividad, sentido común, adaptabilidad, miras amplias, etc.
- Comprobación costosa de inconsistencias
- (Cada día tienen menos inconvenientes)

Ejemplos de Sistemas Expertos

- Sistemas de diagnóstico médico
- Sistemas de diagnóstico en talleres de automóviles
- Sistemas de apoyo en la toma de decisiones (inversiones, bolsa, etc.)
- ¿Se os ocurre alguno más?
- ¿Habéis utilizado alguna vez uno?

Bibliografía

- Inteligencia Artificial. Un enfoque Moderno. Stuart Russell, Peter Noving. Ed. Prentice Hall. 2004
- Fundamentos de inteligencia artificial. M.A Cazorla y otros. Ed. Publicaciones de la Universidad de Alicante.
- The Rete Matching Algorithm. Dr Dobb's site:
<https://drdobbs.com>