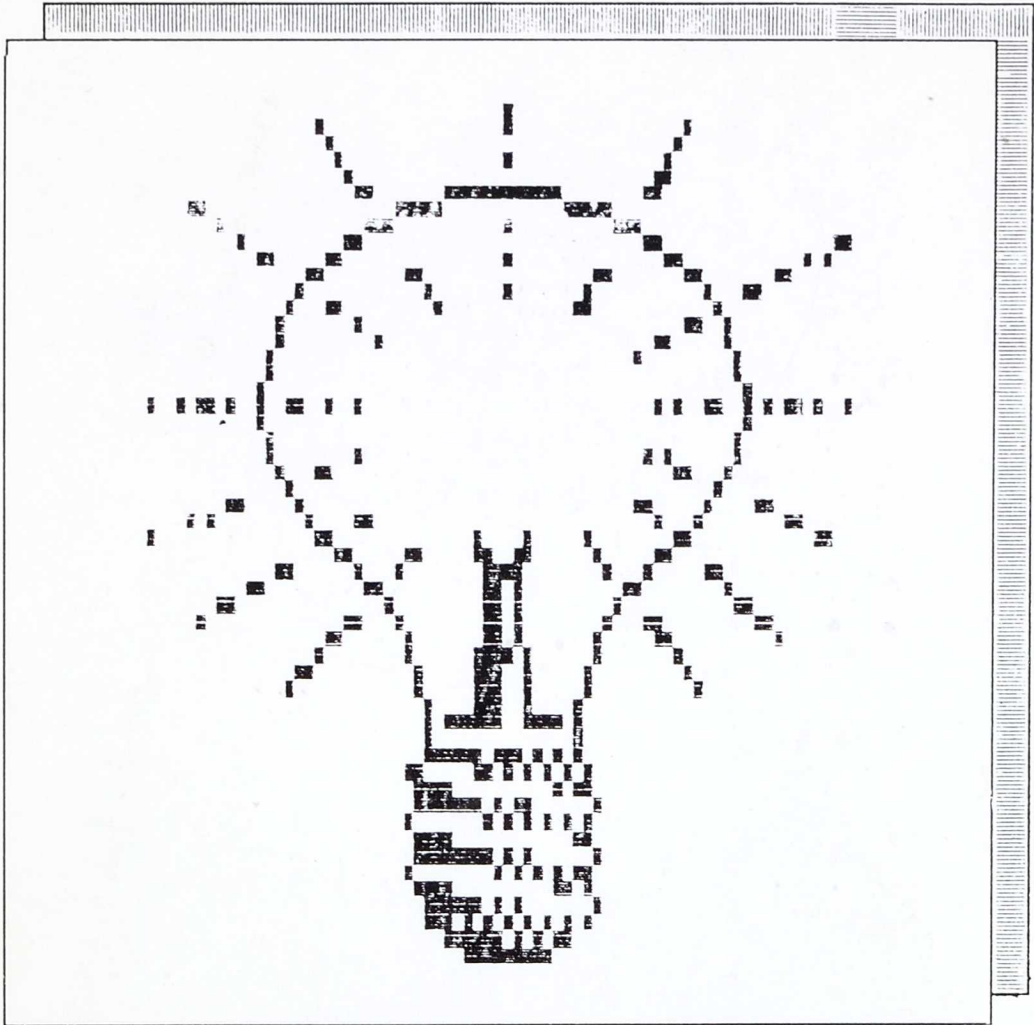


PROGRAMA DE APRENDIZAJE DE CONCEPTOS



JUAN ARANDA CARRILERO

PUBLICACIONES

CEP

ALBACETE

SERIE : NUEVAS TECNOLOGIAS

C 1075/8



C 1075/8

C 1075/8

Programa de Aprendizaje de Conceptos

AUTOR: Juan Aranda Carrilero

BIBLIOMEC



024222



A-79.068

INDICE

PARTE I: APRENDIZAJE DE MAQUINA

- 1.1) Una revisión de la noción de Aprendizaje.
 - 1.1.1) Tres perspectivas sobre el Aprendizaje.
 - 1.1.2) Un Modelo simple de Aprendizaje.
- 1.2) Aprendizaje de Descripciones de Clases a partir de Ejemplos.
 - 1.2.1) Un Caso de Inducción.
 - 1.2.2) Cuasiejemplos. Algoritmo de Especialización
 - 1.2.3) Ejemplos correctos. Algoritmo de generalización.
 - 1.2.4) Un Algoritmo general de Aprendizaje.

PARTE II: P.A.C.: UNA INTRODUCCION

- 2.1) Método de Aprendizaje utilizado.
- 2.2) Sistema de introducción de ejemplos.
- 2.3) Representación interna de la Red Semántica.
- 2.4) Un caso práctico.

PARTE III: MANUAL DE USUARIO DE P.A.C.

- 3.1) Requerimientos del Sistema.
- 3.2) Comenzando una sesión de Aprendizaje.
- 3.3) El Menú de usuario.
- 3.4) Máscara de introducción de ejemplos.
- 3.5) Obtención del Concepto aprendido.

PARTE I: APRENDIZAJE DE MÁQUINA

- 1.1) Una revisión de la noción de Aprendizaje.
 - 1.1.1) Tres perspectivas sobre el Aprendizaje.
 - 1.1.2) Un Modelo simple de Aprendizaje.
- 1.2) Aprendizaje de Descripciones de Clases a Partir de Ejemplos:
 - 1.2.1) Un Caso de Inducción.
 - 1.2.2) Casos de los Algoritmos de Especialización.
 - 1.2.3) Ejemplos de Algoritmos de generalización.
 - 1.2.4) Un Algoritmo general de Aprendizaje.

PARTE I

- 1.1) Una revisión de la noción de Aprendizaje.

La mayoría de la gente considera la habilidad de aprender como una característica fundamental de la noción de inteligencia, ya que es precisamente la que más nos diferencia del resto de los animales. Además, de todos los mecanismos mentales que demostramos inteligencia, es el más evasivo en relación a ser implementado mediante procedimientos. No obstante, algunos mecanismos de aprendizaje pueden ser explicados e incorporados en rutinas.

APRENDIZAJE DE MÁQUINA

Herbert Simon define el aprendizaje como cualquier proceso por medio del cual un sistema mejora su actuación. Esta definición supone que el sistema intenta realizar alguna tarea. Esta puede ser mejorada aplicando nuevas técnicas y conocimiento o haciéndolo más rápido, más robusto o más exacto.

Una visión más restringida, adoptada por mucha gente que trabaja en el campo de los Sistemas Expertos, es que el aprendizaje consiste en la adquisición explícita de conocimiento. Muchos Sistemas Expertos representan su experiencia como grandes colecciones de reglas que necesitan ser adquiridas, organizadas y, en su caso, aumentadas. Esta visión pone el énfasis en la importancia de hacer explícito el conocimiento adquirido, de manera que puede ser fácilmente verificado, modificado y explicado. Los investigadores trabajan activamente sobre sistemas de adquisición del conocimiento que sean capaces de descubrir nuevas reglas a partir de ejemplos o aceptar reglas de los expertos en lenguajes próximos al natural e incorporarlas en una Base de Conocimiento.



Parte I: Aprendizaje de Máquina

PARTE I: APRENDIZAJE DE MAQUINA

- 1.1) Una revisión de la noción de Aprendizaje.
 - 1.1.1) Tres perspectivas sobre el Aprendizaje.
 - 1.1.2) Un Modelo simple de Aprendizaje.
- 1.2) Aprendizaje de Descripciones de Clases a partir de Ejemplos.
 - 1.2.1) Un Caso de Inducción.
 - 1.2.2) Cuasiejemplos. Algoritmo de Especialización
 - 1.2.3) Ejemplos correctos. Algoritmo de generalización.
 - 1.2.4) Un Algoritmo general de Aprendizaje.

1.1) Una revisión de la noción de Aprendizaje.

La mayoría de la gente considera la *habilidad de aprender* como una característica fundamental de la noción de inteligencia, ya que es precisamente la que más nos diferencia del resto de los animales. Además, de todos los mecanismos mentales que demuestran inteligencia, es el más evasivo en relación a ser implementado mediante procedimientos. No obstante, algunos mecanismos de aprendizaje pueden ser explicados e incorporados en rutinas.

1.1.1) Tres perspectivas sobre el Aprendizaje.

Herbert Simon define el aprendizaje como *cualquier proceso por medio del cual un sistema mejora su actuación*. Esta definición supone que el sistema intenta realizar alguna tarea. Esta puede ser mejorada aplicando nuevos métodos y conocimiento o haciéndolos más rápidos, más robustos o más exactos.

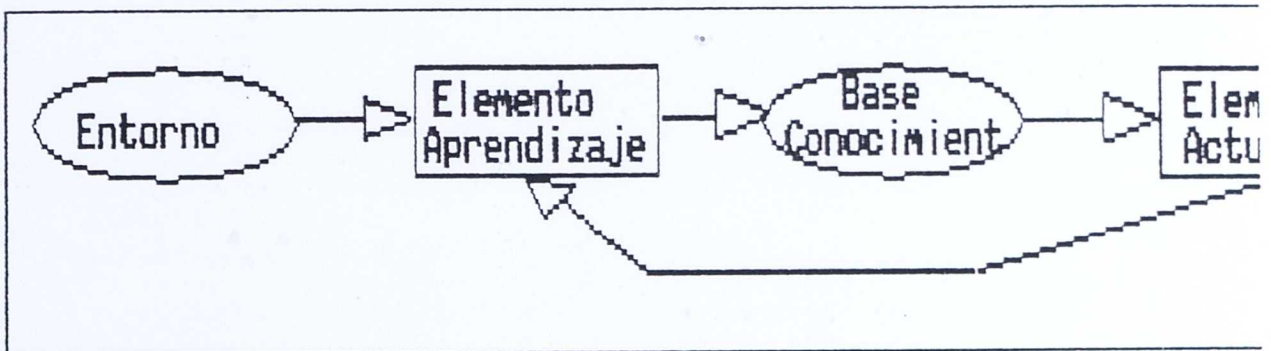
Una visión más restringida, adoptada por mucha gente que trabaja en el campo de los Sistemas Expertos, es que el aprendizaje consiste en la *adquisición explícita de conocimiento*. Muchos Sistemas Expertos representan su experiencia como grandes colecciones de reglas que necesitan ser adquiridas, organizadas y, en su caso, aumentadas. Esta visión pone el énfasis en la importancia de hacer explícito el conocimiento adquirido, de manera que pueda ser fácilmente verificado, modificado y explicado. Los investigadores trabajan activamente sobre sistemas de adquisición del conocimiento que sean capaces de descubrir nuevas reglas a partir de ejemplos o aceptar reglas de los expertos en lenguajes próximos al natural e incorporarlas en una Base de Conocimiento.

Parte I: Aprendizaje de Máquina

Una tercera visión afirma que el aprendizaje es la *adquisición de destrezas*. Los sicólogos han apuntado que mucho tiempo después de que a una persona se le haya dicho cómo realizar una tarea, tal como tocar el piano o programar un ordenador, su destreza continua aumentando por medio de la práctica. Parece que aunque sea sencillo comprender los mecanismos básicos de una tarea todavía es necesaria mucha práctica para que la tarea se realice con un nivel de rendimiento aceptable.

1.1.2) Un modelo simple de aprendizaje.

Tomando la primera visión como punto de partida, podemos elaborar un modelo simple de un sistema de aprendizaje, tal como se muestra en la figura adjunta:



En el modelo, los círculos significan bloques de información declarativa (hechos representados en forma de cálculo de predicados, reglas de experiencia de un experto, etc.), mientras que los rectángulos significan procedimientos. Las flechas muestran la dirección predominante del flujo de datos por el sistema de aprendizaje. El entorno suministra alguna información al elemento de aprendizaje; éste usa esta información para realizar mejoras en la Base de Conocimiento y el elemento de actuación usa el conocimiento para realizar su tarea. Finalmente, la información obtenida durante la realización de la tarea puede realimentar en el elemento de aprendizaje para modificar la Base de Conocimiento. Este modelo es primitivo, pero permite clasificar muchos sistemas de aprendizaje, de acuerdo a cómo éstos se adaptan a las cuatro unidades funcionales.

Parte I: Aprendizaje de Máquina

El factor que más afecta al diseño del sistema de aprendizaje es el *Nivel de la información suministrada*.

Este se refiere al grado de generalidad de la información suministrada en relación a la representación de la misma información que es utilizada por el elemento de actuación.

Información de alto nivel es información abstracta que es significativa para una amplia clase de problemas. Información de bajo nivel es información detallada que es significativa solamente para un único problema. La tarea del elemento de aprendizaje puede ser contemplada como el establecimiento de un puente entre el nivel en el que la información es suministrada por el entorno y el nivel en el que el elemento de actuación puede usar esa información para realizar su función.

En general tres situaciones diferentes pueden darse:

a) Aprendizaje memorístico:

El entorno proporciona información exactamente en el nivel del elemento de actuación.

b) Aprendizaje por instrucción:

La información proporcionada por el entorno es más general o abstracta que la usada por el elemento de actuación.

c) Aprendizaje a partir de ejemplos:

La información suministrada es más específica y detallada que la usada por el elemento de actuación.

1.2) Aprendizaje de Descripciones de Clases a partir de ejemplos.

Un tipo particular de aprendizaje a partir de ejemplos es el Aprendizaje de Descripciones de Clases a partir de ejemplos positivos y negativos, desarrollado por el profesor Patrick Henry Winston a mediados de los años setenta.

El método utiliza tanto ejemplos correctos como cuasi-ejemplos (near miss samples) que son ejemplos que NO son correctos por un pequeño detalle.

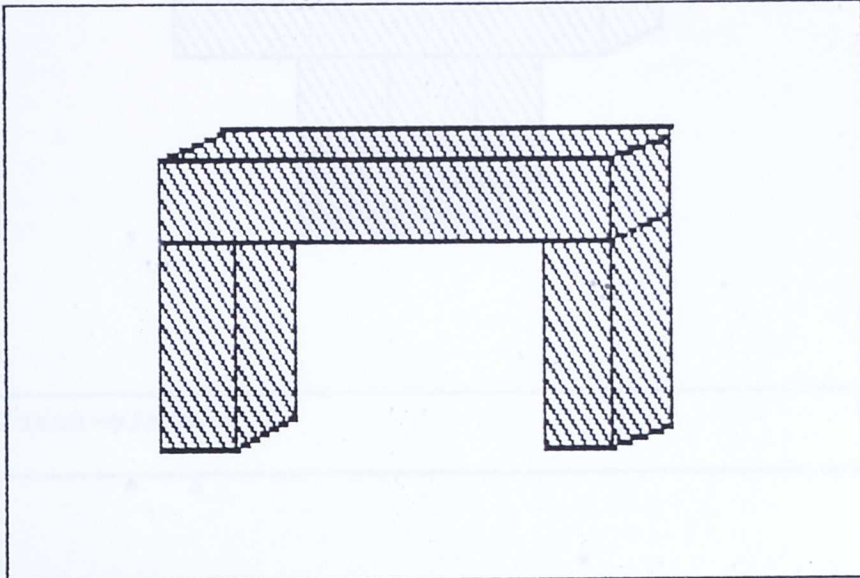
El algoritmo de aprendizaje necesita la colaboración de un profesor (que realiza las funciones del entorno) que presentan tanto los ejemplos correctos como los cuasi-ejemplos en un orden apropiado y uno después de otro.

El objetivo del método de aprendizaje es obtener la descripción de un Concepto que representa a una Clase de Objetos. Cuando el Concepto ha sido suficientemente elaborado puede ser utilizado (elemento de actuación) para clasificar otros objetos como pertenecientes o no pertenecientes a la Clase aprendida.

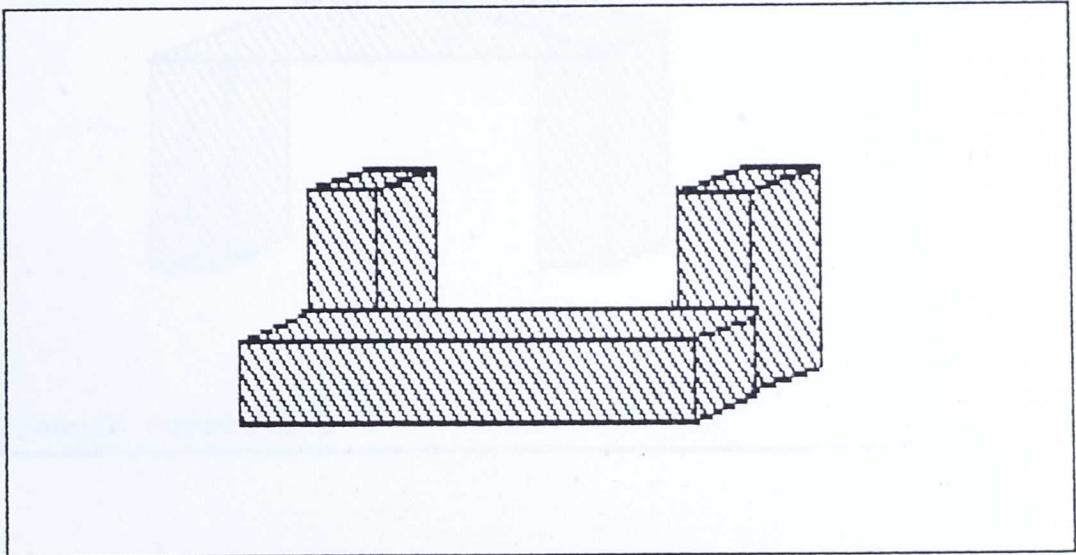
Parte I: Aprendizaje de Máquina

1.2.1) Un Caso de inducción.

Consideremos la secuencia de arcos y cuasi-arcos mostrada en la figura adjunta:

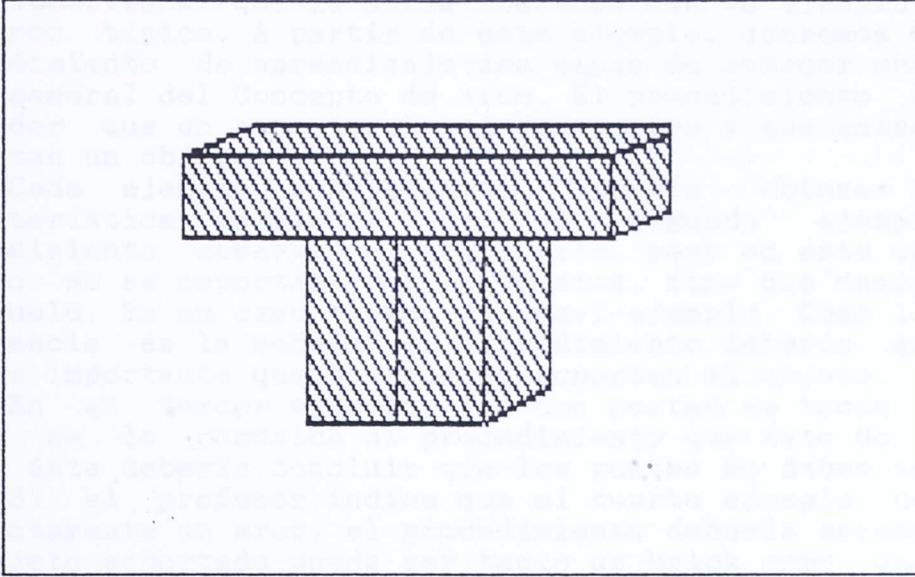


Modelo Inicial

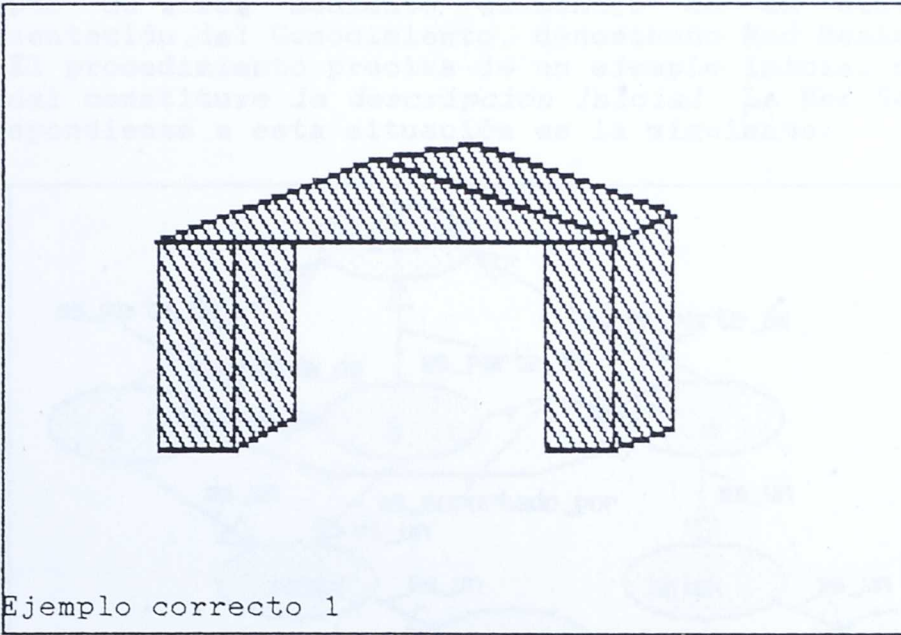


Cuasi-ejemplo 1

Parte I: Aprendizaje de Máquina



Cuasi-ejemplo 2



Ejemplo correcto 1

Observemos que la serie comienza con un ejemplo que es un arco típico. A partir de este ejemplo, queremos que un procedimiento de aprendizaje sea capaz de obtener una idea muy general del Concepto de arco. El procedimiento debería aprender que un arco involucra dos postes y que entre ambos soportan un objeto.

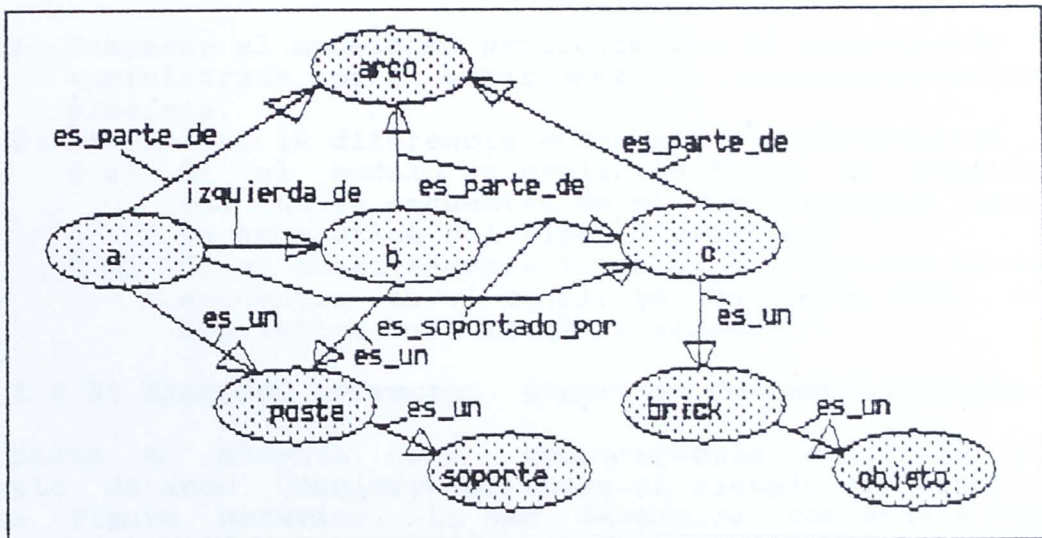
Cada ejemplo subsiguiente debería obtener alguna característica adicional. En el segundo ejemplo, el procedimiento observa el mismo arco, pero en este caso el objeto no es soportado por los postes, sino que descansa en el suelo. Es un caso típico de cuasi-ejemplo. Como la única diferencia es la señalada, el procedimiento debería aprender que es importante que los postes soporten al objeto.

En el tercer ejemplo, los dos postes se tocan. Si de nuevo se le comunica al procedimiento que éste No es un arco, éste debería concluir que los postes No deben tocarse.

Si el profesor indica que el cuarto ejemplo describe correctamente un arco, el procedimiento debería aprender que el objeto soportado puede ser tanto un brick como una cuña.

Comenzaremos a explicar un procedimiento de aprendizaje, denominado W, mostrando cómo aprende el Concepto de arco mediante el manejo de un sistema de representación del Conocimiento, denominado Red Semántica.

El procedimiento precisa de un ejemplo inicial de arco, el cual constituye *la descripción inicial*. La Red Semántica correspondiente a esta situación es la siguiente:



Durante el aprendizaje, la descripción inicial es incrementada por medio de información de otras Redes Semánticas que indican qué vínculos son importantes y cuáles no para la correcta identificación del Concepto. Esta descripción cambiante es lo que se denomina *Modelo*.

1.2.2) Cuasi-ejemplos. Algoritmo de Especialización.

Un *cuasi-ejemplo* es una descripción del Concepto que no se puede considerar un ejemplo correcto por un pequeño número de razones.

En la Red Semántica que describiera el cuasi-ejemplo 1 de la figura anterior, no aparecerían vínculos que describieran la situación en la cual el objeto es soportado por los postes. Por ello el procedimiento de aprendizaje debería obtener la conclusión de que esta situación es importante para que la descripción correspondiera a un arco. Por tanto el modelo obtenido de este paso debería contener la forma enfática de que el objeto debe ser soportado por los postes. Esta situación es un exponente de la heurística denominada *vínculo precisado*.

En la Red semántica que describe el cuasi-ejemplo 2 aparecería un vínculo que indicara que un poste toca a otro, y que no se encontraría en la descripción inicial. Ahora el cuasi-ejemplo no es un ejemplo correcto de arco, por un motivo diferente al caso anterior. En este caso el problema no es que haya vínculos ausentes sino que un presente vínculo no debería aparecer. El procedimiento de aprendizaje debería concluir que un arco no debe tocar a otro. Es un caso de la heurística *vínculo olvidado*.

El denominado *procedimiento de especialización* es capaz de manejar estos dos casos:

ESPECIALIZAR:

- 1) Comparar el modelo en evolución con la descripción suministrada para establecer las correspondencias precisas.
- 2) Determinar la diferencia entre ambas descripciones.
 - 2.a) Si el modelo en evolución tiene un vínculo que no se encuentra en el cuasi-ejemplo, usar la heurística del vínculo precisado
 - 2.b) Si el cuasi-ejemplo tiene un vínculo que no se encuentra en el modelo en evolución, usar la heurística del vínculo olvidado.

1.2.3) Ejemplos correctos. Algoritmo de generalización.

Hasta el momento cada cuasi-ejemplo restringe el Concepto de arco. Consideremos ahora el ejemplo-correcto 1 de la figura anterior. La Red Semántica contendría un vínculo del objeto soportado a una cuña. Si ésto se le confirma que es un arco, el procedimiento debería proveer algún mecanismo para poder generalizar el objeto soportado a algo más general que un brick o una cuña.

Parte I: Aprendizaje de Máquina

Si la Red Semántica provee vínculos del tipo es un que conectan objetos con objetos más generales, se debería realizar un seguimiento de estos vínculos para obtener el objeto más general posible. Si es así, diremos que estamos haciendo uso de la heurística de *ascensión en árboles*.

A veces en cambio la Red Semántica no contiene estos árboles de objetos generalizados mediante vínculos *es un*. En este caso, aún sería posible definir un nuevo objeto que fuera una lista de los casos particulares encontrados hasta el momento. En este caso diremos que estamos usando la heurística del *conjunto incremental*.

Si una descripción en forma de Red Semántica de un ejemplo correcto contiene vínculos que no están presentes en el modelo en evolución podremos pensar que estos vínculos no son estrictamente necesarios para describir correctamente el concepto. Entonces usaremos la heurística de *eliminación de vínculo*.

El procedimiento de aprendizaje que resume el mecanismo de todas estas situaciones se denomina *procedimiento de generalización* y se puede describir en la siguiente forma:

GENERALIZAR:

- 1) Comparar el modelo en evolución con la descripción suministrada para establecer las correspondencias precisas.
- 2) Determinar la diferencia entre ambas descripciones.
 - 2.a) Si la diferencia es que el vínculo apunta a un objeto diferente en el ejemplo correcto y en el modelo en evolución, entonces:
 - 2.a.1) Si los objetos son parte de un árbol de de clasificación, usar la heurística de *ascensión en árboles*.
 - 2.a.2) Si no usar la heurística del *conjunto incremental*
 - 2.b) Si la diferencia es que un vínculo se encuentra perdido ya sea en el ejemplo correcto o en el modelo en evolución, usar la heurística de *eliminación de vínculo*.

1.2.4) Un Algoritmo general de Aprendizaje.

El algoritmo W usa ejemplos suministrados por un profesor. Estos pueden ser tanto ejemplos correctos como cuasi-ejemplos, pero la descripción inicial tiene que ser siempre un ejemplo correcto. El profesor decide el orden de presentación. El algoritmo es el siguiente:

Parte I: Aprendizaje de Máquina

ALGORITMO W:

- 1) Proporcionar la descripción del primer ejemplo, el cual tiene que ser correcto.
- 2) Para el resto de las descripciones presentadas:
 - 2.a) Si la descripción es un cuasi-ejemplo, usar el procedimiento ESPECIALIZAR.
 - 2.b) Si la descripción es un ejemplo correcto, usar el procedimiento GENERALIZAR.

PARTE II

P.A.C. : Una Introducción

PARTY II: P.A.C. : UNA INTRODUCCION

- 2.1) Método de Aprendizaje utilizado.
- 2.2) Sistema de introducción de ejemplos.
- 2.3) Representación interna de la Red Conexionista.
- 2.4) Un caso práctico.

PARTE II

2.1) Método de Aprendizaje utilizado

El Programa de aprendizaje de Nociones (P.A.C.) usa una versión reducida del Algoritmo P. desarrollado por el profesor Patrick Henry Winston. En ella se contemplan las siguientes heurísticas:

- Heurísticas del vínculo predefinido.
- Heurísticas del vínculo dividido.
- Heurísticas de inversión de árboles.
- Heurísticas de eliminación de vínculos.

y se se utilizan la heurísticas del principio incremental. El sistema de desarrollo utilizado es el Sistema Arity Párola de esta Corporación.

P.A.C. : Una Introducción

2.2) Sistema de introducción de ejemplos.

En la versión original, el programa dispone de una interfaz de reconocimiento de formas que convierte los ejemplos y sus correspondientes representaciones en forma de árbol a una representación interna de forma de los nodos. El desarrollo de esta interfaz fue desarrollado por el equipo del aprendizaje de Nociones. Esta interfaz está basada en el programa de desarrollo de árboles de aprendizaje bastante simple y eficiente.

Las descripciones serán introducidas por ejemplo en una forma que sea fácilmente transitable al formato de los ejemplos. En estas los ejemplos serán convertidos en los ejemplos y los vínculos binarios que los conectan. En algunos casos usaremos el siguiente formato:

Objetos: {frutas} {animales} {plantas}

PARTE II: P.A.C.: UNA INTRODUCCION

- 2.1) Método de Aprendizaje utilizado.
- 2.2) Sistema de introducción de ejemplos.
- 2.3) Representación interna de la Red Semántica.
- 2.4) Un caso práctico.

2.1) Método de Aprendizaje utilizado

El *Programa de Aprendizaje de Conceptos (P.A.C.)* usa una versión reducida del Algoritmo W, desarrollado por el profesor Patrick Henry Winston. En ella se contemplan las siguientes heurísticas:

- Heurística del vínculo precisado.
- " del vínculo olvidado.
- " de ascensión de árboles.
- " de eliminación de vínculo.

y no se utiliza la heurística del conjunto incremental.

El sistema de desarrollo utilizado es el compilador Arity Prolog de Arity Corporation.

2.2) Sistema de introducción de ejemplos.

En la versión original, el programa disponía de una interfase de reconocimiento de formas que convertía los ejemplos y cuasi-ejemplos presentados en forma de dibujo a una representación interna en forma de Red Semántica. El desarrollo de esta interfase cae evidentemente fuera del ámbito del Aprendizaje de Máquina. Para suplir esta carencia el programa aquí desarrollado utiliza un mecanismo bastante menos sofisticado.

Las descripciones serán introducidas por teclado de una forma que sea fácilmente trasladable al formato de Red Semántica. En éstas los elementos principales son los nodos y los vínculos binarios que los unen. En nuestro caso usaremos el siguiente formato:

<Sujeto> <Predicado> <Objeto>

Así, para indicar que una columna es un soporte adoptaremos la convención de expresarlo de la siguiente manera:

columna es_un soporte

De esta manera cada vínculo vendrá representado por un Predicado y los nodos que están unidos por éste serán respectivamente el Sujeto y el Objeto.

2.3) Representación interna de la Red Semántica.

Para representar internamente la Red Semántica utilizaremos los mecanismos que proporciona el Prolog. En concreto cada vínculo binario asociado a un par de nodos será representado como un átomo Prolog con estructura de functor a excepción de aquellos vínculos relacionados con el árbol de ascensión que serán representados por medio de clausulas.

De hecho se utilizarán un conjunto de predicados Prolog para indicar la situación en la que se encuentra cada una de las ternas formada por un vínculo y los nodos que enlaza.

Estos son los siguientes:

- a) **modelo_concepto**
Contiene parte de la descripción del modelo en evolución. En concreto aquellos vínculos y sus nodos asociados que no han sufrido ninguna modificación por el Programa de Aprendizaje.
- b) **ejemplo_correcto**
Contiene la descripción del ejemplo correcto que se encuentra en ese momento en proceso.
- c) **cuasi_ejemplo**
Contiene la descripción del cuasi-ejemplo que se encuentra en ese momento en proceso.
- d) **debe_tener**
Contiene la descripción de aquellos vínculos a los que se les ha aplicado la heurística del vínculo precisado.
- e) **no_debe_tener**
Contiene la descripción de aquellos vínculos a los que se les ha aplicado la heurística del vínculo olvidado.

Como ya se ha comentado anteriormente los vínculos y nodos implicados en el árbol de ascensión se representan internamente como clausulas, utilizando todas ellas el predicado común **es_un**.

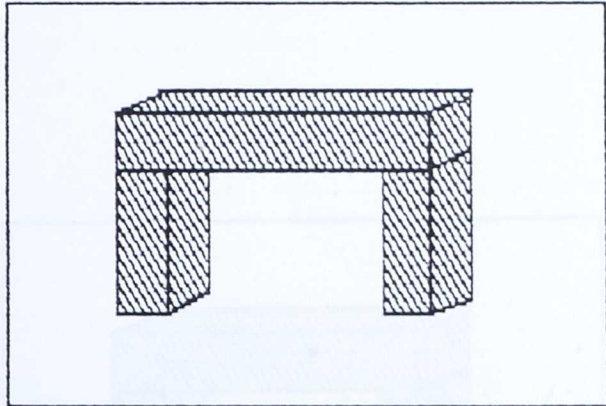
2.4) Un Caso práctico.

En este momento podría ser interesante presentar un caso práctico que nos permitiera comprender con mayor claridad cuál será el mecanismo de presentación de ejemplos y cuasi-ejemplos. El Concepto que se intentará aprender será el de arco y deberá recordarse que es responsabilidad del profesor presentar cada descripción en un orden adecuado para que el proceso de aprendizaje pueda tener lugar satisfactoriamente.

El orden de presentación podría ser el siguiente:

a) Ejemplo correcto inicial.

a es_parte_de arco
b es_parte_de arco
c es_parte_de arco
a es_un poste
b es_un poste
c es_un brick
c es_soportado_por a
c es_soportado_por b
a izquierda_de b

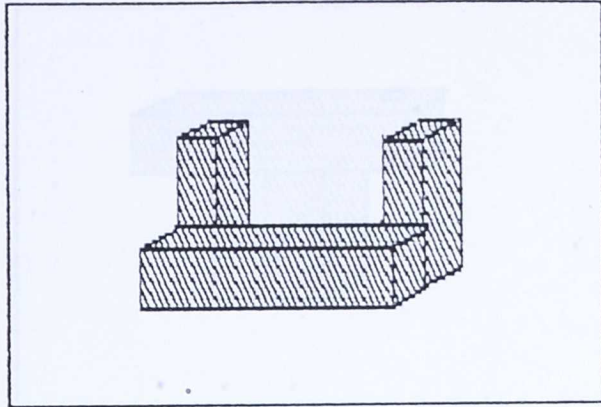


b) Arbol de ascensión

poste es_un soporte
columna es_un soporte
brick es_un objeto
cuña es_un objeto

c) Cuasi ejemplo 1.

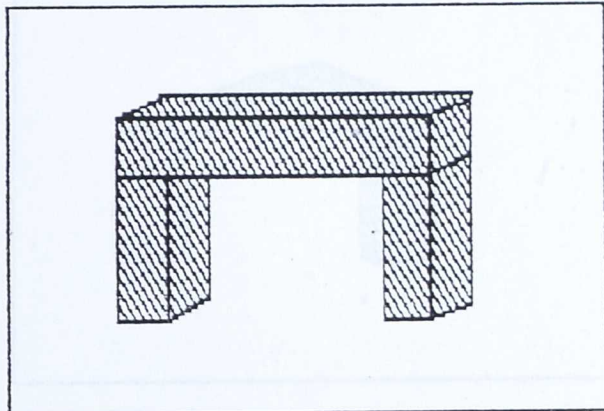
a es_parte_de arco
b es_parte_de arco
c es_parte_de arco
a es_un poste
b es_un poste
c es_un brick
a izquierda_de b



Notemos que a este cuasi-ejemplo le faltan las descripciones correspondientes al vínculo es_soportado_por. Debido a ello entrará en funcionamiento la heurística del vínculo precisado.

d) Ejemplo correcto 1.

a es_parte_de arco
b es_parte_de arco
c es_parte_de arco
a es_un poste
b es_un poste
c es_un brick
c es_soportado_por a
c es_soportado_por b

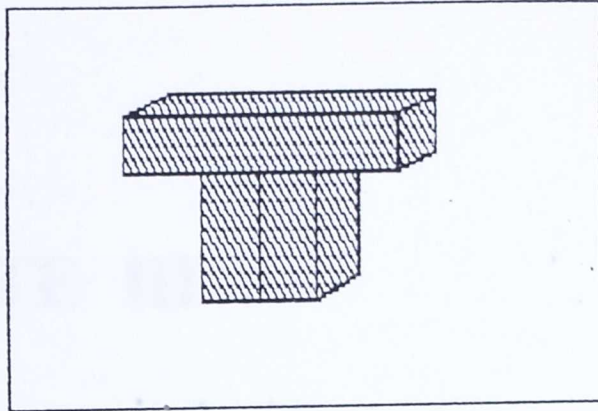


En este caso el vínculo que no aparece en el ejemplo correcto es el correspondiente a izquierda_de permitiendo la aplicación de la heurística de eliminación de vínculos.



e) Cuasi ejemplo 2.

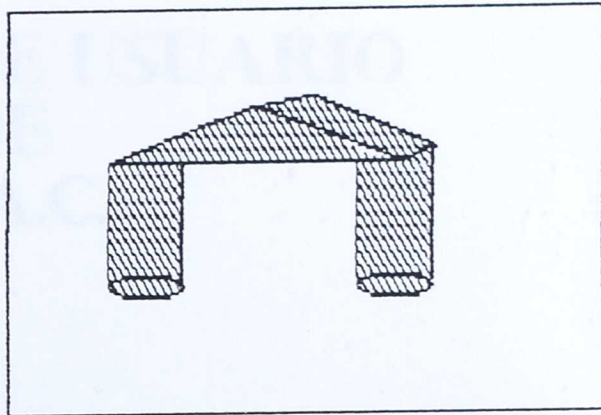
a es_parte_de arco
b es_parte_de arco
c es_parte_de arco
a es_un poste
b es_un poste
c es_un brick
c es_soportado_por a
c es_soportado_por b
a toca_a b



La aparición del vínculo toca_a permite el uso de la heurística del vínculo olvidado. Además hemos de señalar el hecho de que una aplicación juiciosa de las descripciones obliga al profesor a no incluir en este cuasi-ejemplo el vínculo izquierda_de pues ello falsearía el resultado del aprendizaje.

f) Ejemplo correcto 2

a es_parte_de arco
b es_parte_de arco
c es_parrte_de arco
a es_un columna
b es_un columna
c es_un cuña
c es_soportado_por a
c es_soportado_por b



En este cuasi-ejemplo cambiamos los siguientes Objetos:

- poste por columna
- brick por cuña

Por ello se aplica la heurística de ascensión de árboles. Notemos además que en los dos últimos casos hemos debido de introducir la descripción de los vínculos es_soportado_por, dado que ésta es necesaria para la correcta identificación del Concepto de arco.

PARTY III: MANUAL DE USUARIO DE P.A.C.

- 3.1) Resumen
- 3.2) Comenzando una sesión de aprendizaje
- 3.3) El menú
- 3.4) Especificación de introducción de ejemplos
- 3.5) Utilización del Concepto Aprendido

PARTE III

3.1) Resumen del sistema

El Programa de aprendizaje de Conceptos puede ejecutarse en cualquier ordenador compatible IBM compatible PC/XT que disponga al menos de una unidad de disco. En algunos casos puede ser necesario disponer de un monitor en color y siempre se recomienda que se tenga una memoria disponible sea de 640 K.

MANUAL DE USUARIO DE P.A.C.

3.2) Comenzando una sesión de aprendizaje

Para ejecutar el programa desde la línea del prompt del DOS, debemos teclear `CDPAC` y posteriormente pulsar la tecla `ENTER`. En la pantalla, podrá ver la siguiente:

```
C:\DOS>CDPAC
```

En ese momento aparecerá la pantalla de presentación del programa, tal como se muestra a continuación:

PARTE III: MANUAL DE USUARIO DE P.A.C.

- 3.1) Requerimientos del Sistema.
- 3.2) Comenzando una sesión de Aprendizaje.
- 3.3) El Menú de usuario.
- 3.4) Máscara de introducción de ejemplos.
- 3.5) Obtención del Concepto aprendido.

3.1) Requerimientos del Sistema.

El Programa de Aprendizaje de Conceptos puede ejecutarse en cualquier ordenador compatible IBM o compatible PS/2 que disponga al menos de una unidad de diskette. En algunos casos puede ser necesario disponer de un monitor en color y siempre es recomendable que la Memoria RAM disponible sea de 640 K.

Dado que el programa usa el controlador de dispositivo ANSI es necesario que en el directorio raíz se encuentre el fichero ANSI.SYS y que en el fichero de configuración del Sistema Operativo MS-DOS, denominado CONFIG.SYS, exista una línea que contenga lo siguiente:

```
DEVICE=ANSI.SYS
```

3.2) Comenzando una sesión de Aprendizaje.

Para ejecutar el programa desde la línea del prompt del DOS, deberemos teclear APREND2 y posteriormente pulsar la tecla Return. Así la situación, podría ser la siguiente:

```
C>APREND2(Ret)
```

En ese momento aparecerá la pantalla de presentación del programa, tal como se muestra a continuación:

Programa de Aprendizaje de Conceptos

Según el método del Profesor Patrick Henry Winston
Juan Aranda Carrilero (1.998)
Versión Arity-Prolog

<PgDn> Continuar visualización

3.3) El Menú de Usuario.

Pulsando la tecla <Pg Ab>, pasamos a la pantalla que contiene el Menú de Usuario, tal como puede observarse a continuación:

MENU PRINCIPAL DEL PROGRAMA DE APRENDIZAJE

Modelo inicial Conocimiento_inicial Aprendizaje Solución Final
Permite introducir la descripción del primer ejemplo positivo

<Ret> Acept.opc. <-> Opc.sig. <<-> Opc.ant. <Esc> Abandonar

En éste podemos elegir una de las siguientes opciones:

a) Modelo Inicial.

Será obligatoriamente la primera opción que debemos elegir, dado que en ella identificaremos el Concepto que debe ser aprendido. Esto se hará recurriendo a la siguiente máscara:

IDENTIFICACION DEL CONCEPTO A SER APRENDIDO
<p>Has de introducir el nombre del Concepto que debe ser aprendido</p> <p>Nombre = <input type="text"/></p>
<Ret> Aceptar Bor.car. <<> Backspace <Esc> Abandonar

Posteriormente se nos pedirá la descripción del Ejemplo Inicial Correcto del Concepto para lo cual se utilizará una Máscara de Introducción de ejemplos, cuya operativa se explicará en la siguiente Sección.

b) Conocimiento inicial.

Servirá para introducir las descripciones correspondientes al árbol de ascensión. Por este motivo el predicado que tendrá sentido utilizar será es_un.

c) Aprendizaje.

Nos introducirá en un submenú del cual podremos elegir cualquiera de las dos siguientes opciones:

c.1) Ejemplo correcto.

c.2) Cuasi_ejemplo

ambas con el significado obvio.

d) Solución.

Nos proporcionará el Concepto aprendido utilizando tres listas de descripciones:

d.1) Las descripciones que *puede tener* el Concepto.

d.2) Las descripciones que *debe tener* el Concepto.

d.3) Las descripciones que *no debe tener* el Concepto.

e) Final.

Nos permitirá abandonar el programa y regresar al Sistema Operativo.

3.4) Máscara de Introducción de ejemplos.

Las descripciones de los vínculos presentes en la Red Semántica se realizarán mediante el formato ya comentado anteriormente:

<Sujeto> <Predicado> <Objeto>

y de acuerdo con la máscara de pantalla de la siguiente figura:

INTRODUCCION DE LOS PREDICADOS DEL CONCEPTO		
Introduciremos los predicados con el siguiente formato <SUJETO> <PREDICADO> <OBJETO>		
SUJETO	PREDICADO	OBJETO
_____	_____	_____
¿Desea Continuar?(s/n) <input checked="" type="checkbox"/> no		
NOTA: Se comenzará por minúscula y no se admite el carácter blanco		
<Ret> Aceptar_valor <Shift-Tab> Camp. ant. <Esc> Abandonar		

Esta misma máscara nos servirá para introducir las descripciones correspondientes al ejemplo inicial, al árbol de ascensión, a los ejemplos correctos y a los cuasi-ejemplos.

Es de destacar que los valores introducidos deberán comenzar obligatoriamente por una letra minúscula y que no podrán contener el carácter blanco en su interior. Para paliar esta última circunstancia podrá utilizarse el carácter subrayado "_".

3.5) Obtención del Concepto Aprendido.

Como ya hemos comentado al explicar la Opción Solución, el programa nos proporciona tres listados diferentes, correspondientes a las descripciones que puede, debe o no debe tener el Concepto. En la siguiente figura se observa el resultado correspondiente al ejemplo práctico desarrollado en la Parte II para la lista de descripciones que puede tener el Concepto de arco.

Resultados correspondientes al Concepto de: arco

Los predicados que PUEDE TEMER son los siguientes:

a es_parte_de arco
b es_parte_de arco
c es_parte_de arco
a es_un soporte
b es_un soporte
c es_un objeto

<PgDn> Continuar visualización

Impreso en el CENTRO DE PROFESORES DE ALBACETE
en Octubre de 1.991
DEPOSITO LEGAL AB-122-1.991



MINISTERIO DE EDUCACION Y CIENCIA
Subdirección General de Formación del Profesorado