



ugr

Universidad de Granada

Departamento de Ciencias de la Computación  
e Inteligencia Artificial



## Grado en Ingeniería Informática

### Diseño y Desarrollo de Sistemas de Información

#### Resolución del examen de contenidos teóricos, Septiembre de 2016

##### Ejercicio 1

La toma de decisión resulta un proceso habitual en la vida cotidiana del ser humano. En multitud de ocasiones nos encontramos ante diferentes alternativas entre las que debemos seleccionar la que, a nuestro juicio, nos parece mejor o la que satisface el mayor número de necesidades requeridas. El Análisis de Decisión Multicriterio es una herramienta que brinda varios métodos para ayudar al decisor durante el proceso de toma de decisiones. En este proceso se definen varios conceptos:

- a) Decisor es el ente responsable de la selección de una posible alternativa, y se caracteriza por un identificador y el número de años de experiencia.
- b) Las alternativas son las posibles acciones o soluciones a seleccionar por parte del decisor. Una alternativa se identifica por un identificador, y esta tiene varios criterios que se utilizan para describir las cualidades de las alternativas a evaluar. Un criterio se identifica por una descripción y el identificador de la alternativa y cada criterio se asocia a una única alternativa.
- c) Un decisor evalúa los criterios de las alternativas dando una evaluación numérica y definiendo un valor de peso de cada uno de ellos. Cada criterio de una alternativa debe ser evaluado, al menos, por un experto.
- d) Un método multicriterio evalúa las alternativas consideradas por los decisores según las valoraciones asignadas a los criterios asociados, y con ello obtiene un ranking de las alternativas.
- e) Existen diferentes métodos multicriterios que pueden usarse para ello. Estos se identifican por un identificador y pueden ser de tres tipos diferentes: de Escuela Americana, de Escuela Europea y Otros. Los métodos de la Escuela Americana se caracterizan por usar alguna función de utilidad, Los de la Escuela Europea por usar alguna medida de sobreclasificación, y los de Otros por usar una distancia a la solución ideal y la no ideal.

Diseña un diagrama entidad-relación que recoja todas estas restricciones y permita que el sistema realice todas las funcionalidades especificadas.



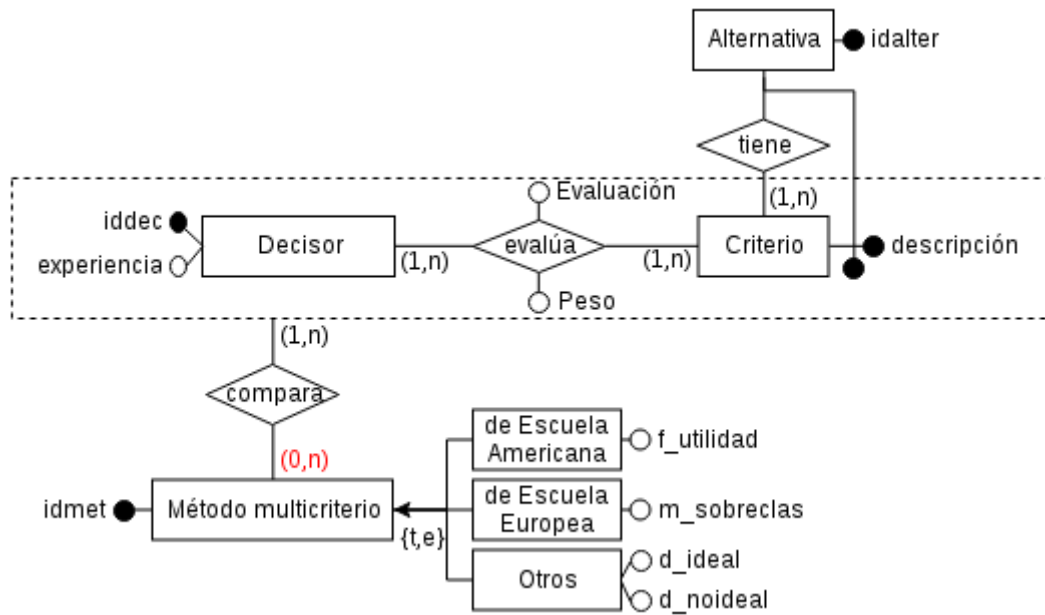
ugr

Universidad de Granada

Departamento de Ciencias de la Computación  
e Inteligencia Artificial



## Resolución





ugr

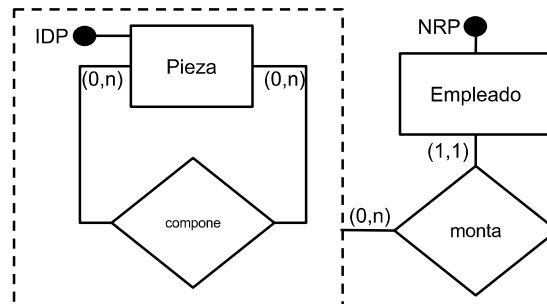
Universidad de Granada

Departamento de Ciencias de la Computación  
e Inteligencia Artificial



## Ejercicio 2

Propón un esquema relacional (conjunto de tablas) capaz de almacenar la información representada en el siguiente Diagrama Entidad-Relación, incluyendo todas las restricciones necesarias.



## Resolución

El esquema relacional propuesto, en el que los atributos que forman parte de la llave primaria aparecen en negrita y las llaves externas y restricciones adicionales (*NOT NULL*) se especifican explícitamente para cada tabla, es el siguiente:

- Pieza (**idp**)
- Empleado (**NRP**)
- Compone\_monta (**idp1**, **idp2**, **NRP**), con *idp1* e *idp2* llaves externas al atributo *idp* de la tabla *pieza*, y *NRP* llave externa a la tabla *empleado* y con la restricción *Not Null* sobre dicho atributo.



ugr

Universidad de Granada

Departamento de Ciencias de la Computación  
e Inteligencia Artificial



### Ejercicio 3

Sea  $R$  una relación de esquema  $\{A, B, C, D, E, F\}$  que verifica el conjunto  $F$  de dependencias funcionales  $\{D \rightarrow C, DE \rightarrow F, B \rightarrow D, AF \rightarrow C, DF \rightarrow A, D \rightarrow B\}$ . Determinar las claves candidatas de la relación y justificar por qué son las únicas existentes.

### Resolución

En primer lugar, sería recomendable establecer los tipos de atributos en función de si aparecen a la izquierda en todas las dependencias funcionales en las que aparece, a la derecha en todas las dependencias funcionales en las que aparece, a la izquierda de alguna y a la derecha en otra o en ninguna dependencia funcional:

- atributos independientes (no aparecen en ninguna dependencia): ninguno
- atributos equivalentes: la pareja  $B-D$
- atributos que aparecen a la izquierda en todas las dependencias en las que aparecen:  $\{E\}$
- atributos que aparecen a la izquierda de alguna dependencia funcional y a la derecha de otra:  $\{A, B, D, F\}$
- atributos que aparecen a la derecha en todas las dependencias en las que aparecen:  $\{C\}$

Establecidos los tipos, podemos aplicar el algoritmo de extracción de claves candidatas:

1º Eliminación de atributos independientes:

Al no haber atributos independientes, la relación sin independientes queda  $RSI = R = \{A, B, C, D, E, F\}$

2º Eliminación de equivalencias:

Al existir una equivalencia entre parejas de atributos, seleccionamos un atributo de la pareja para eliminarlo (habitualmente, el que menos veces aparece en el conjunto de dependencias, es decir, en este caso,  $B$ ) y sustituimos todas las apariciones de ese atributo en el nuevo conjunto por el atributo que se queda:  $RSIE = RSI = \{A, C, D, E, F\}$  y  $FSIE = \{D \rightarrow C, DE \rightarrow F, D \rightarrow D, AF \rightarrow C, DF \rightarrow A, D \rightarrow D\} = \{D \rightarrow C, DE \rightarrow F, AF \rightarrow C, DF \rightarrow A\}$ .

3º Procesamiento de atributos que aparecen sólo a la izquierda:

$$K_p = \{E\} = E$$

Habiendo un candidato, tenemos que probar si es una clave:

$E^+ = \{E\} \neq RSIE$ , luego  $E$  no es clave y no hemos conseguido una clave en el paso 3, lo que nos obliga a pasar al paso 4.

4º Procesamiento de atributos que aparecen a la izquierda y a la derecha:

Dado que el candidato del paso 3 tiene que estar incluido en todas las claves (por estar formado por atributos que están sólo a la izquierda), cualquier candidato que se genere en este paso debe incluir todos los atributos del candidato del paso 3 (menos aquellos que se puedan derivar de dicho candidato), de modo que debemos combinar el candidato del paso 3 con cualquier atribu-



ugr

Universidad de Granada

Departamento de Ciencias de la Computación  
e Inteligencia Artificial



to que esté a la izquierda y a la derecha pero no sea generado por el candidato del paso 3, es decir:

$$Kp' = \{EA, ED, EF\}$$

Dado que existen candidatos, debemos comprobar si el primero es clave, comprobando si es capaz de generar toda la relación RSIE:

$EA+ = \{E, A\} \neq RSIE$ , de modo que EA no es una clave, pero una extensión de EA podría serlo. Para calcular las extensiones directas de EA es necesario combinar EA con cualquier atributo que esté a la izquierda y la derecha, pero no sea generado por el propio EA, es decir, EAD y EAF. Sin embargo, si generamos el candidato EAD mientras ED es un candidato posible, y se demuestra que ED es clave, EAD no podría serlo. Por lo tanto, no generamos EAD por ser una extensión de un candidato. Lo mismo ocurre con EAF. Por ello, este candidato no genera ningún candidato adicional.

$$ED+ = \{E, D, C, F, A\} = RSIE, \text{ de modo que ED es una clave, por lo que, } CKSIE = \{ED\}$$

Puesto que quedan candidatos, debemos seguir explorando:

$EF+ = \{E, F\} \neq RSIE$ , de modo que EF no es una clave, pero una extensión de EF podría serlo. Para calcular las extensiones directas de EF es necesario combinar EF con cualquier atributo que esté a la izquierda y la derecha, pero no sea generado por el propio EF, es decir, EFA y EFD. EFD no puede ser un candidato puesto que es una extensión de ED que es una clave, pero EFA no es extensión de ninguna clave ni de ningún candidato que quede, por lo que:

$$Kp' = \{EFA\}$$

Como quedan candidatos, tenemos que seguir explorando si son clave o no:

$EFA+ = \{E, F, A, C\} \neq RSIE$ , por lo que EFA no es una clave, pero puede haber alguna extensión con un atributo que esté a la izquierda y la derecha que lo sea. La única extensión que queda es EFAD pero se trata de una extensión de la clave ED, por lo que EFAD no es candidata.

Dado que el conjunto de candidatos está vacío, no quedan candidatos que explorar y  $CKSIE = \{ED\}$

##### 5º Incorporación de atributos independientes:

Puesto que no existían atributos independientes,  $CK' = \{ED\}$

##### 6º Incorporación de atributos equivalentes:

Al existir una pareja de atributos equivalentes, es necesario duplicar cada clave que incluya un atributo equivalente (y por cada pareja, por si hubiera más de una) y sustituir en la copia el atributo que dejamos por el que eliminamos, de modo que,  $CK = \{ED, EB\}$

El hecho de haber aplicado el algoritmo de extracción de claves candidatas, garantiza que las únicas claves que hay son ED y EB.



ugr

Universidad de Granada

Departamento de Ciencias de la Computación  
e Inteligencia Artificial



## Ejercicio 4

Sea  $R$  una relación de esquema  $\{A, B, C, D, E\}$ , con un conjunto  $CK$  de claves candidatas  $\{AD, AE\}$  y que verifica un conjunto  $F$  de dependencias funcionales  $\{C \rightarrow B, D \rightarrow E, BE \rightarrow D, AE \rightarrow C\}$ . Comprobar si la relación está en Forma Normal de Boyce y Codd (sin comprobar si está en segunda forma normal o en tercera forma normal) y si no lo está, realizar una descomposición hasta que todas las relaciones de la descomposición estén en dicha forma normal.

## Resolución

La relación  $R$  estaría en Forma Normal de Boyce y Codd si, y sólo si, todas las dependencias tienen a su izquierda una clave candidata. Según esta norma, podemos decir que la relación no está en BCNF porque  $C \rightarrow B$ ,  $D \rightarrow E$  y  $BE \rightarrow D$  forman parte de  $F$  y ni  $C$  ni  $D$  ni  $BE$  son claves candidatas de la relación.

Para conseguir una normalización, debemos aplicar el Algoritmo de Heath sobre una de las dependencias que no cumplen la forma normal. Sin embargo, se sabe que si se considera la importancia de las dependencias según el atributo que figura a su derecha, se reduce el riesgo de la pérdida de dependencias en una descomposición. La importancia de las dependencias para la normalización se establece, de menor a mayor, en dependencias que tienen a la derecha un atributo determinado y no determinante (no existe ninguna), dependencias que tienen a la derecha un atributo determinado y determinante que no forma parte de ninguna clave ( $C \rightarrow B$ ) y dependencias que tienen a la derecha un atributo determinado y determinante que forma parte de alguna clave ( $D \rightarrow E$  y  $BE \rightarrow D$ ). Escogiendo entre las dependencias “menos importantes para la normalización” minimizamos riesgos, así que normalizamos por  $C \rightarrow B$ :

$R_1 = \{C, B\}$ ,  $F_1 = \{C \rightarrow B\}$ ,  $CK_1 = \{C\}$ ,  **$R_1$  está en FNBC** puesto que todas las dependencias funcionales de  $F_1$  tienen una clave candidata de  $R_1$  a su izquierda.

$R_2 = \{A, C, D, E\}$ ,  $F_2 = \{D \rightarrow E, AE \rightarrow C\}$ ,  $CK_2 = \{AD, AE\}$ ,  $R_2$  no está en BCNF porque  $D \rightarrow E$  forma parte de  $F_2$  pero  $D$  no es clave candidata de  $R_2$ .

Es necesario normalizar la relación  $R_2$  mediante la aplicación del Teorema de Heath. Sin embargo, no hay dependencias en  $F_2$  que tengan un atributo determinante y no determinado a la derecha. Sólo queda una dependencia que no cumple la forma normal de modo que normalizamos por  $D \rightarrow E$ :

$R_{2,1} = \{D, E\}$ ,  $F_2 = \{D \rightarrow E\}$ ,  $CK_2 = \{D\}$ ,  **$R_{2,1}$  está en BCNF** porque todas sus dependencias tienen a la izquierda una llave candidata de  $R_{2,1}$ .

$R_{2,2} = \{A, C, D\}$ ,  $F_2 = \emptyset$ ,  $CK_2 = \{AD\}$ ,  **$R_{2,2}$  está en BCNF** porque todas sus dependencias (no tiene ninguna explícita aunque sí tiene que la que se deriva de la clave candidata) tienen a la izquierda una llave candidata de  $R_{2,2}$ .

El resultado de nuestra normalización es el conjunto de relaciones:

$$\{(\{C, B\}, r_1), (\{D, E\}, r_{2,1}), (\{A, C, D\}, r_{2,2})\}$$