



Representación del conocimiento

Métodos estructurados: Marcos





Contenido

1. Introducción a los sistemas de marcos.
2. Elementos de un sistema de marcos.
3. Un modelo básico.
4. Herencia simple sin excepciones.
5. Herencia simple.
6. Facetas.
7. Herencia múltiple.
8. Marcos y Reglas.
9. Resumen.



1. Introducción a los sistemas de marcos



Origen

- Minsky, 75, "A Framework for representing knowledge"
- "... Esta es la esencia de la teoría de los marcos: cuando nos enfrentamos a una nueva situación, o cuando realizamos un cambio sustancial en nuestra visión de un problema, seleccionamos en nuestra memoria una estructura denominada marco. Este es un esquema que nosotros recordamos y que debemos adaptar a la realidad cambiando los detalles según sea necesario..."
- Estructuras de datos utilizadas para representar elementos bien conocidos (prototípicos).
- Para adaptarnos a la situación actual, accedemos a la estructura que más se asemeja, y modificamos los detalles necesarios.



Características

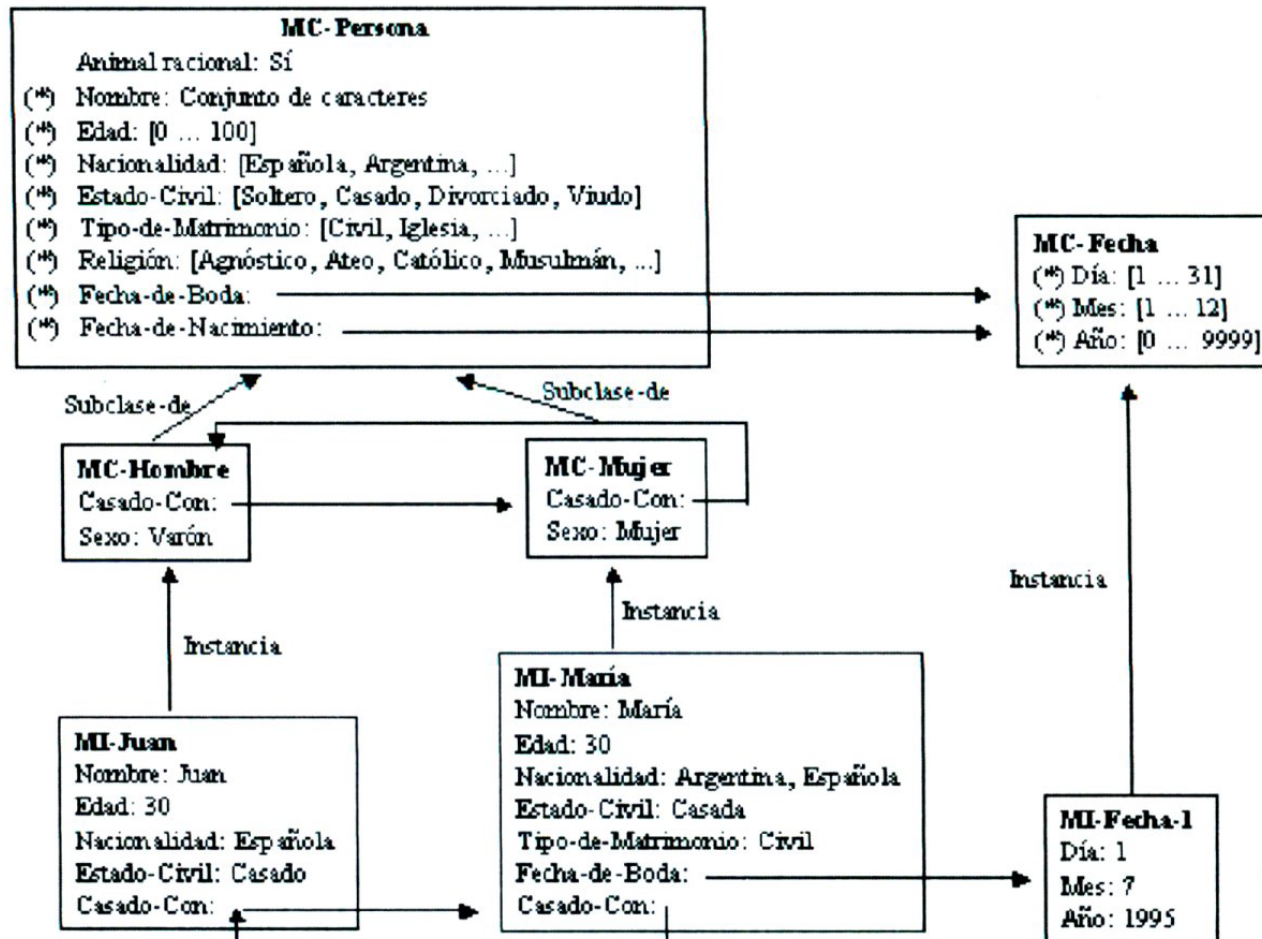
- Como las redes semánticas, hacen explícitas las relaciones del dominio que proporcionan la estructura de la red.
- A diferencia de las redes semánticas, los nodos tienen estructura:
 - Permiten agrupar las propiedades de los elementos del dominio en una unidad denominada Marco (*Frame*).
- Representación principalmente **declarativa**.
 - Propiedades de conceptos, individuos.
 - Pero permiten añadir elementos operacionales a propiedades individuales (mediante facetas).
- Estructuran el dominio en **jerarquías** (árboles, grafos) de herencia.
- Inferencia: Equiparación, **Herencia**, métodos operacionales.



En la actualidad

- Principal formalismo de representación del conocimiento cuando este se organiza en clases.
- Principal formalismo para la representación del conocimiento estructural.
- La organización en jerarquías de clases y la herencia se mantienen en la mayoría de los lenguajes de Ontologías.

Ejemplo de Sistema de Marcos





2. Elementos de un sistema de marcos



Elementos de un sistema de marcos

- Marcos: clases e instancias.
- Relaciones: dependencias entre marcos.
- Propiedades: describen los marcos.
- Facetas: descripción adicional de los valores que pueden tomar las propiedades.

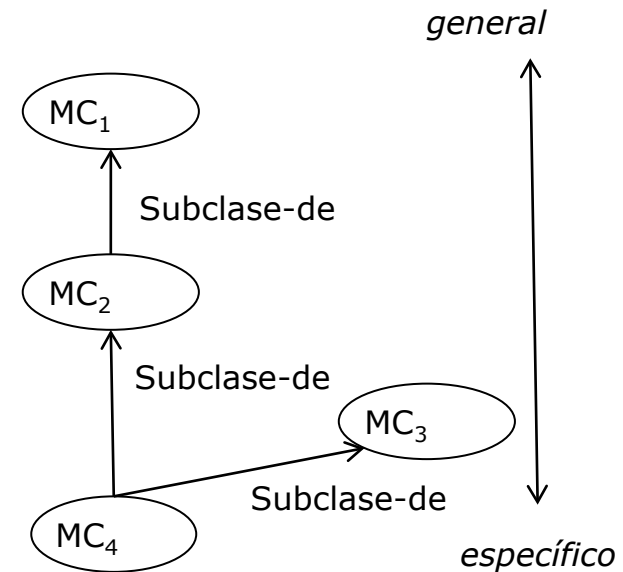


Marcos

- Marcos de clase
 - Conocimiento relativo a clases.
 - Hombre, Puerta-Lógica, Alarma, Plan-Evacuación, Arteria.
- Marcos de instancia
 - Conocimiento relativo a elementos individuales.
 - Juan, AND-1, Alarma-Incendio-327, Plan-Evacuacion-ETSII-2310, Aorta.

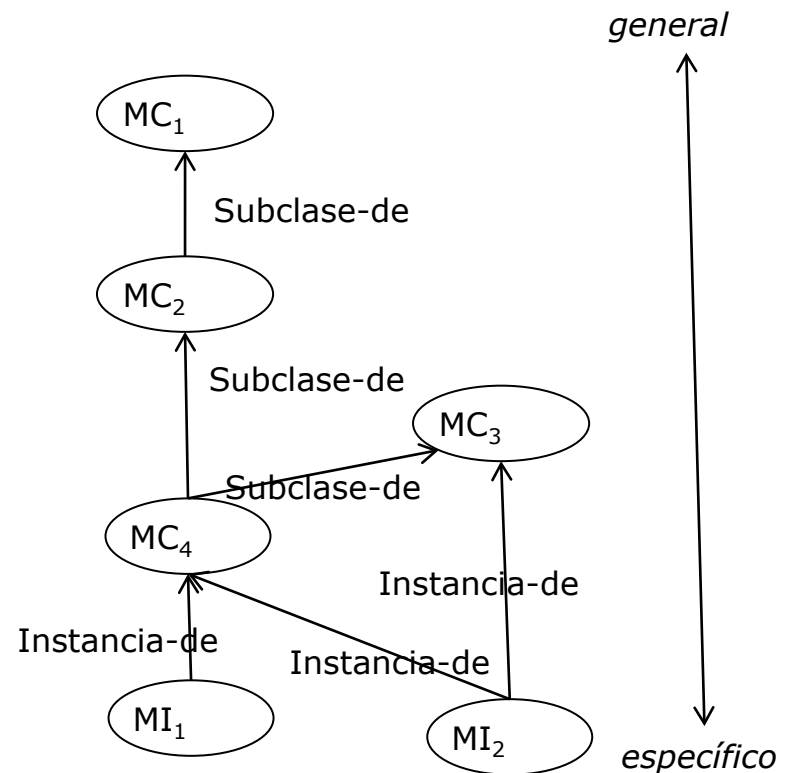
Relaciones estándar (1)

- Subclase-de:
 - Relación binaria entre marcos de clase.
 - Dirigida.
 - De orden parcial: reflexiva, anti simétrica, transitiva.
 - Inversa: superclase.
- Jerarquía de clases:
 - Generalización, especialización.

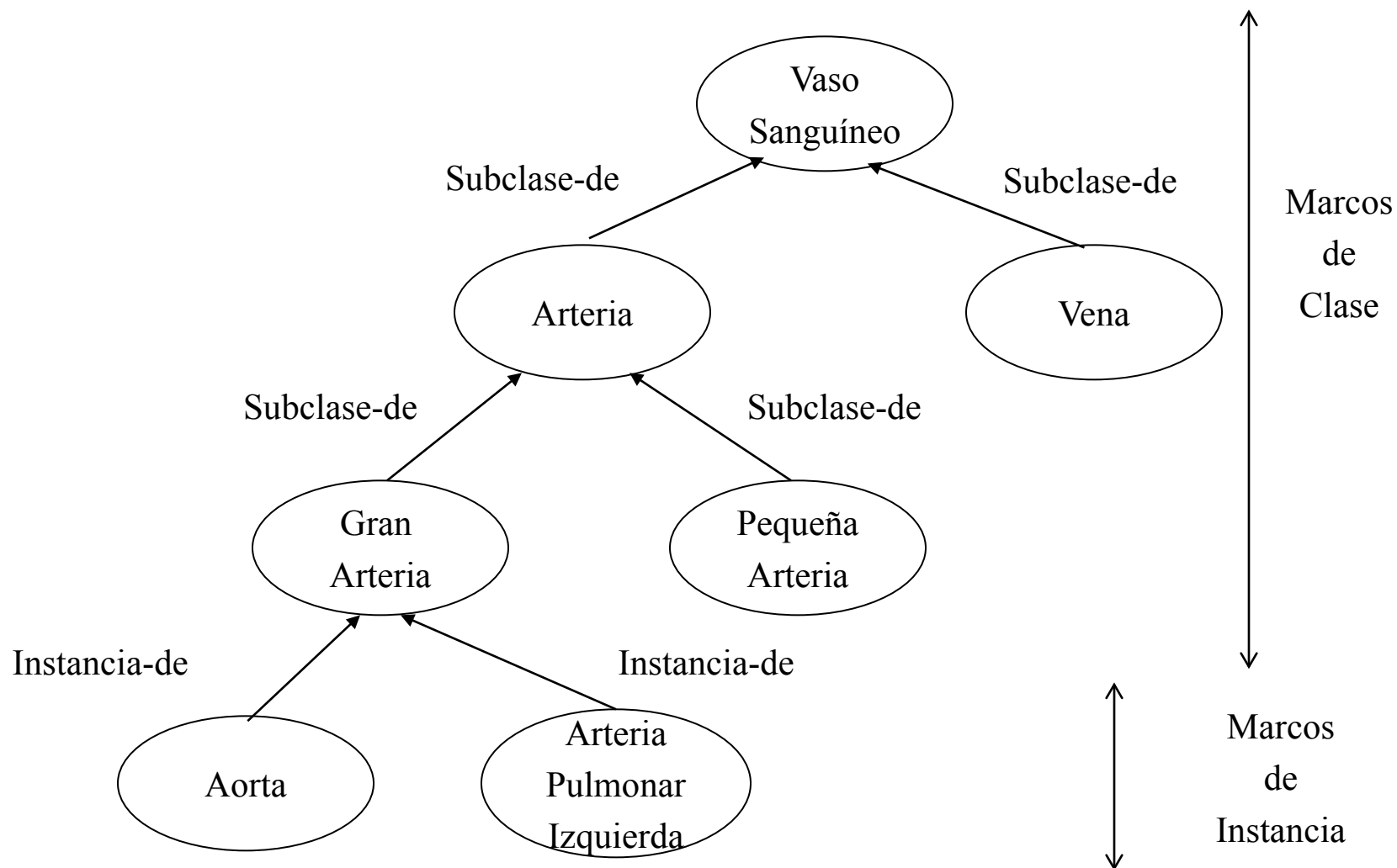


Relaciones estándar (2)

- Instancia-de:
 - Relación entre marcos de instancia y de clase.
 - Dirigida.
 - Inversa: representa.
- Jerarquía de clases.

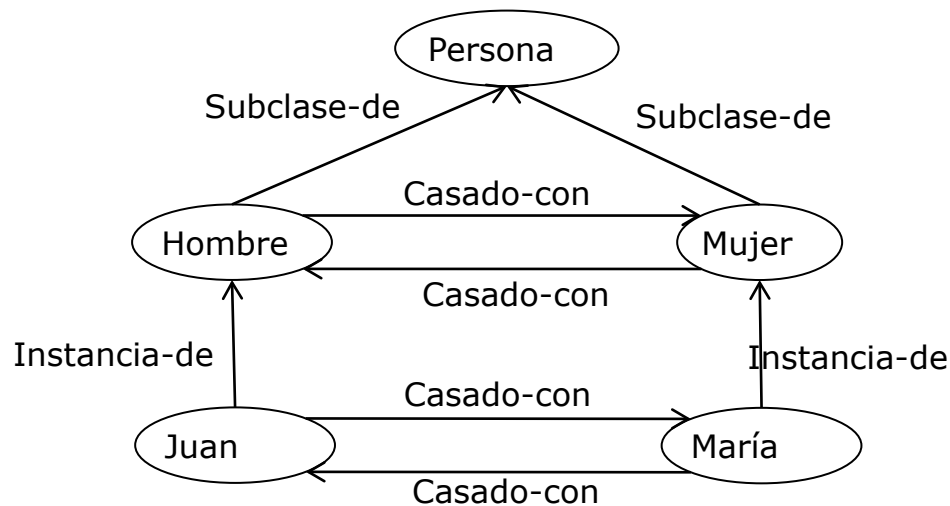


Jerarquía Marcos



Relaciones no estándar

- Cualquier relación dependiente del dominio
 - Hermano-de, Conectado-con, Casado-con
 - Binarias, dirigidas.
 - Inversa.
 - Se han de definir primero entre marcos de clase.





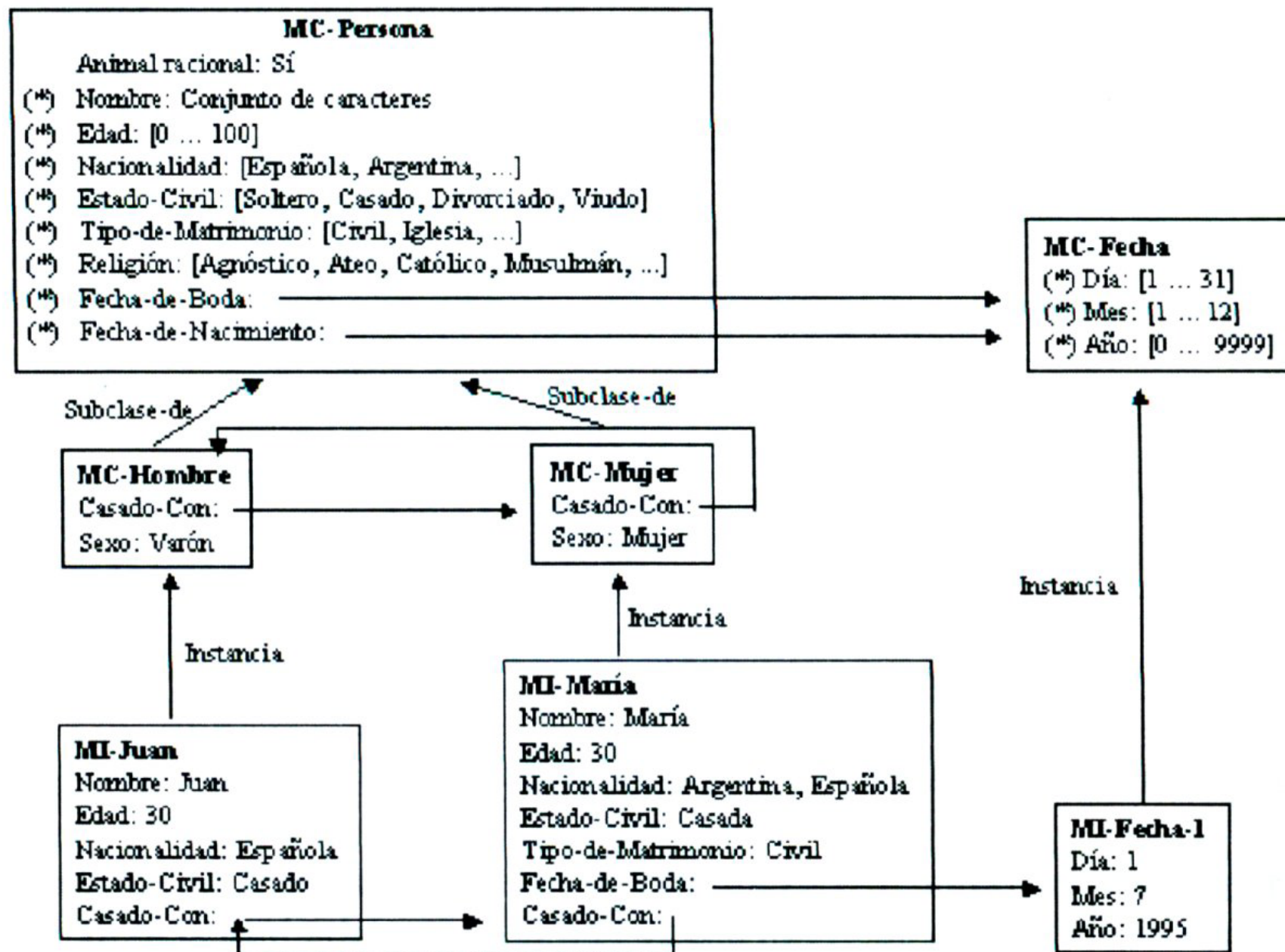
Propiedades

- Permiten describir clases e instancias.
- Propiedades de clase:
 - Describen propiedades de una clase.
 - Reciben valor en la clase que las define.
 - Son comunes a todas las instancias (salvo excepciones).
- Propiedades de instancia:
 - Describen características específicas de las instancias.
 - Se definen en los marcos de clase.
 - Reciben valor en los marcos de instancia.
 - Típicamente, valores distintos en cada instancia.



Facetas

- Permiten definir características adicionales de relaciones y propiedades:
 - Información de tipo.
 - Cardinalidad.
 - Demonios: procedimientos que se invocan automáticamente al acceder/alterar el valor de una propiedad.





3. Un modelo básico



Simplificaciones modelo básico

- Jerarquía de herencia simple: La jerarquía es un árbol.
- Solo relaciones estándar.
- Sin facetas.
- Nombre único de cada marco.
- Propiedades univaluadas.



Sintaxis Marcos

(Jerarquía de Herencia: Árbol)

<marco>	::=	<clase> <instancia>
<clase>	::=	Clase <nombre-clase> es subclass-de <espec-clase-superior> ; [<propiedades-clase> [; <declaración-propiedades-instancia>]] <declaración-propiedades-instancia> end
<instancia>	::=	Instancia <nombre-instancia> es instancia-de <nombre-clase> [; <propiedades-instancia>] end
<espec-clase-superior>	::=	<nombre-clase> T
<propiedades-clases>	::=	<par-propiedad-valor> {;<par-propiedad-valor>}* ε
<propiedades-instancia>	::=	<par-propiedad-valor> {;<par-propiedad-valor>}* ε
<declaración-propiedades-instancia>	::=	*<nombre-propiedad> {;<nombre-propiedad>}* ε
<par-propiedad-valor>	::=	<nombre-propiedad> = <valor>
<valor>	::=	<constante-elemental> <nombre-instancia>



Restricciones del modelo básico

- Cada clase debe definir, al menos, una propiedad de clase o una propiedad de instancia.
- Las propiedades de una instancia han de estar declaradas en alguna de sus superclases.
- Los nombres de las propiedades son locales al marco que las define (aunque puedan heredarse).
- El valor de una propiedad puede ser:
 - Una constante elemental.
 - El nombre de una instancia.



Ejemplo

Clase Persona **es subclase-de** T;
Animal-racional= si;
*Nombre;
*Edad;
*Nacionalidad;
*Estado-civil;
*Tipo-de-Matrimonio;
*Religión;
*Fecha-Nacimiento

end

Clase Hombre **es subclase-de** Persona;
Sexo=varón;
*Fecha-Boda;
*Casado-con

end



Ejemplo (continuación 1)

Clase Mujer **es subclase-de** Persona;
 Sexo=mujer;
 *Fecha-Boda;
 *Casado-con
end

Clase Fecha **es subclase-de** T;
 *Día;
 *Mes;
 *Año
end



Ejemplo (continuación 2)

Instancia Juan es **instancia-de** Hombre;
Nombre=Juan;
Edad=30;
Nacionalidad=Española;
Estado-Civil=Casado;
Fecha-Nacimiento=Fecha-N-Juan;
Fecha-Boda=Fecha-Boda-J-M;
Casado-con=María
end

Instancia María **es instancia-de** Mujer
Nombre=María;
Edad=30;
Nacionalidad=Española;
Estado-Civil=Casada;
Religión=Católica;
Fecha-Nacimiento=Fecha-N-María;
Fecha-Boda=Fecha-Boda-J-M;
Casado-con=Juan
end

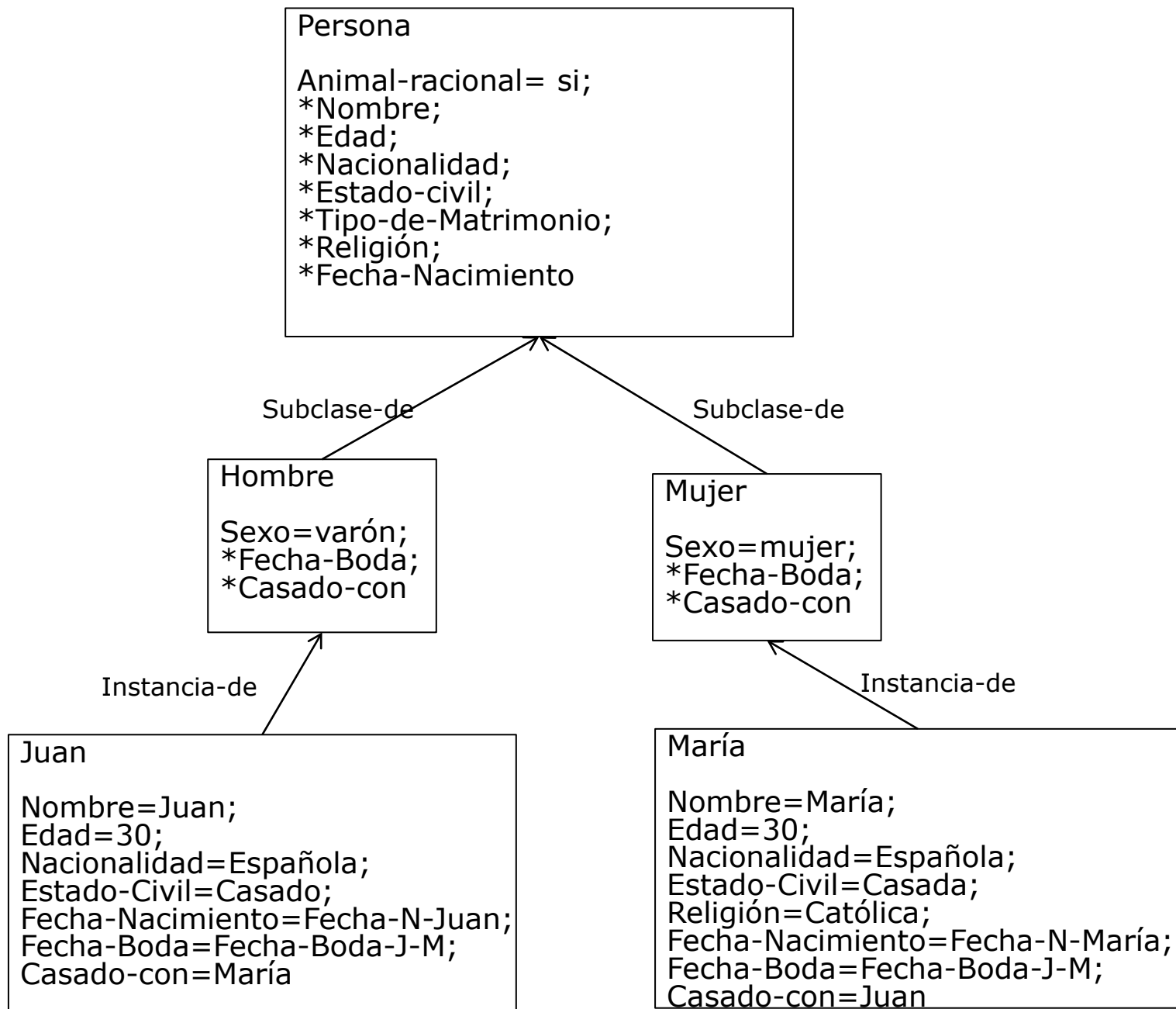


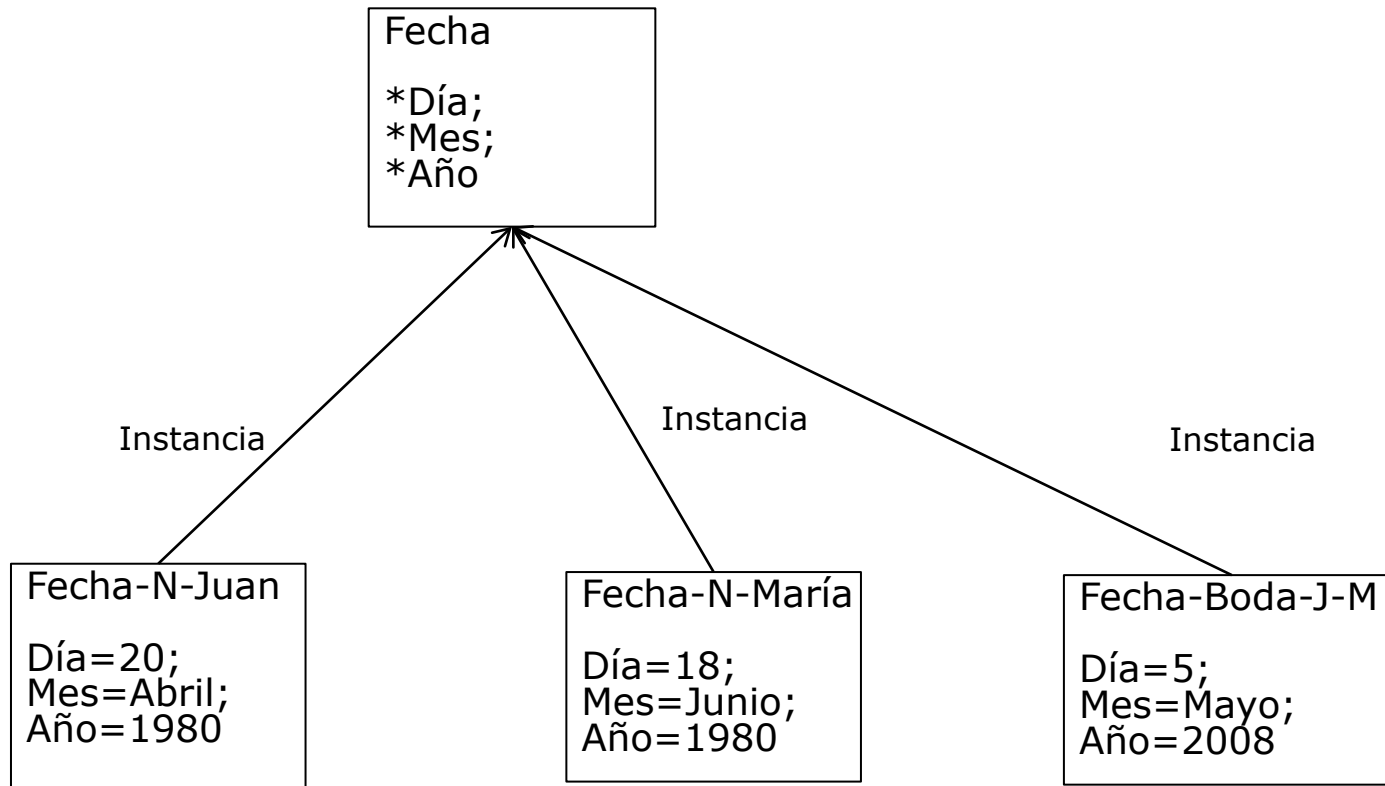
Ejemplo (continuación 3)

Instancia Fecha-N-Juan **es instancia-de** Fecha;
 Día=20;
 Mes=Abril;
 Año=1980
end

Instancia Fecha-N-María **es instancia-de** Fecha;
 Día=15;
 Mes=Junio;
 Año=1980
end

Instancia Fecha-Boda-J-M **es instancia-de** Fecha;
 Día=5;
 Mes=Mayo;
 Año=2008
end







Ejercicio

Representar la siguiente descripción de los vasos sanguíneos mediante un sistema de marcos

1. Los vasos sanguíneos tienen forma tubular y transportan sangre.
2. Los vasos sanguíneos se subdividen en tres categorías: arterias, capilares y venas. Estas categorías se subdividen como indica la figura 1.
3. La aorta, la arteria y vena pulmonares y la arteria cubital son ejemplos de vasos sanguíneos específicos.
4. Las arterias transportan sangre desde el corazón hasta los capilares de los tejidos y se distinguen de otros vasos por poseer una pared gruesa. En la mayoría de los casos, las arterias transportan sangre con un elevado contenido de oxígeno.
5. Contrariamente a las arterias, las venas transportan sangre desde los capilares de los tejidos al corazón. Tienen una pared relativamente delgada. Usualmente, las venas contienen sangre pobre en oxígeno.
6. La presión sanguínea media en las arterias es relativamente elevada (40-100 mmHg), frente a una presión media inferior a 10 mmHg en la mayoría de las venas.



Ejercicio (continuación)

Representar la siguiente descripción de los vasos sanguíneos mediante un sistema de marcos

7. Las arterias pulmonares son un ejemplo de excepción a la descripción anterior. Estas arterias transfieren sangre del corazón a los pulmones y poseen una gruesa pared muscular. Por ello se las considera arterias. Sin embargo, estas arterias transfieren sangre con bajo contenido en oxígeno y su presión media es más bien baja (13 mmHg).
8. Las grandes arterias tienen un diámetro entre 1 y 2,5 cm. Las pequeñas arterias tienen un diámetro de 0,4 cm. y las arteriolas de 0,003 cm.
9. Las grandes venas tienen un diámetro entre 3 y 1,5 cm. y las pequeñas venas tienen un diámetro de 0,5 cm.
10. La arteria aorta tiene un diámetro de 2,5 cm.
11. La arteria pulmonar izquierda tiene un diámetro de 1,4 cm.
12. La vena cava tiene un diámetro de 3 cm.

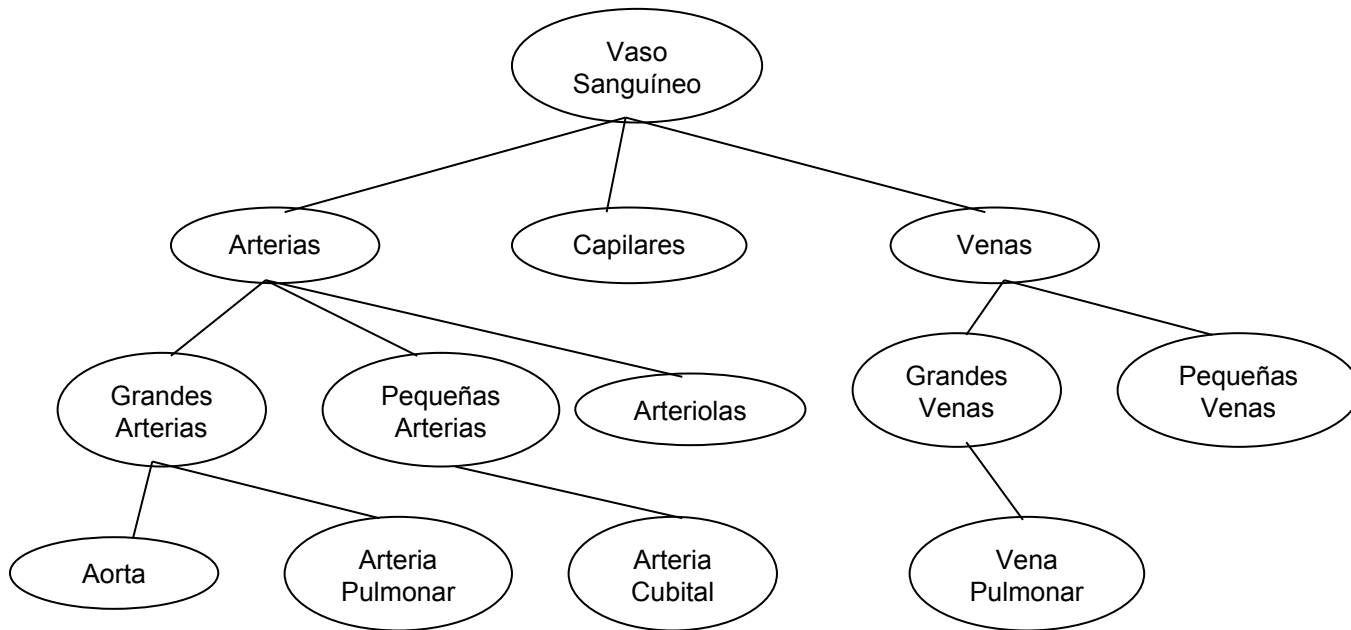


Figura 1: Clasificación de los vasos sanguíneos.



Recomendaciones

- Definir las propiedades de clase en la clase más general posible.
- Al introducir una subclase asegurarse de que
 - Posee una propiedad que no está en su clase padre.
 - Posee una propiedad que la diferencia de sus clases hermanas.
- Recordar que todas las propiedades de las instancias han de estar declaradas en alguna de sus superclases.
- Proporcionar valor solo a las propiedades de las instancias conocidas en tiempo de diseño.



4. Herencia simple sin excepciones



4. Herencia simple sin excepciones

- Herencia simple: la jerarquía de herencia tiene estructura de árbol.
- Modelo básico sin facetas.
- Restricción adicional: sin excepciones
 - cada <par-propiedad-valor> ocurre una sola vez en cada camino de la jerarquía para cada propiedad.



Inferencia mediante herencia simple sin excepciones

- Herencia simple sin excepciones.
- Regla de inferencia que consiste en proporcionar a cada marco todas las propiedades de clase de sus generalizaciones en la jerarquía de herencia.
- Propiedades de un marco
 - Específicas: definidas en el propio marco.
 - Heredadas: definidas en sus superclases.
 - Transparente desde el punto de vista de una consulta.



Ejemplo herencia simple sin excepciones

Clase Vaso-sanguíneo **es subclase-de** T;
 contiene= fluido-sanguíneo; forma = tubular
end

Clase Arteria **es subclase-de** Vaso-sanguíneo;
 sangre = rica-oxigeno; *diámetro
end

Instancia Aorta **es instancia-de** Arteria;
 diámetro = 2,5
end

- Marco Aorta
 atrib. Específicos: diámetro.
 atrib. Heredados: contiene, forma, sangre.



Descripción operacional de la herencia simple sin excepciones

Procedimiento Hereda(nombre-marco, pares-propiedad-valor)
 if nombre-marco=T **then**
 return(pares-propiedad-valor)
 end;
 pares-propiedad-valor \leftarrow pares-propiedad-valor \cup
 Propiedades-clase(nombre-marco);
 return(Hereda(ClaseSuperior(nombre-marco), pares-propiedad-valor))
end



Semántica

herencia simple sin excepciones

- Semántica basada en LPO
 - Inspirada en *Modus Ponens*
 - Transformar sistema de marcos en conjunto de fórmulas LPO.
- Directrices:
 - Nombre instancia: símbolo constante.
 - Nombre clases: símbolo predicado unario.
 - Relación instancia-de: predicado unario (clase), término constante (instancia).
 - Relación subclase-de: implicación lógica (salvo T) y cuantificación universal.
 - Nombre propiedad: símbolo función.
 - Par propiedad-valor: igualdad entre término y constante (derecha implicación si clase).
 - Obviar declaraciones de propiedades de instancias.
- Axiomas: igualdad + nombre único (nombre marcos y constantes).

Esquema general cambio representación

Clase C es subclase-de S;
 $a_1=b_1; a_2=b_2; \dots a_n=b_n$
end

$\forall x (C(x) \supset S(x))$
 $\forall x (C(x) \supset a_1(x)=b_1)$
 $\forall x (C(x) \supset a_2(x)=b_2)$
 .
 .
 .
 $\forall x (C(x) \supset a_n(x)=b_n)$

Instancia I es instancia-de C
 $j_1=k_1; j_2=k_2; \dots j_m=k_m$
end

$C(I)$
 $j_1(I)=k_1$
 $j_2(I)=k_2$
 .
 .
 .
 $j_m(I)=k_m$

- Restricción ocurrencia única par propiedad-valor:
 Garantiza existencia de un modelo.

Ejemplo Herencia simple sin excepciones

Instancia Aorta

I ARTERIA(Aorta)

II diámetro(Aorta) = 2,5

Clase Arteria

III $\forall x (ARTERIA(x) \supset VASO-SANGUINEO(x))$

IV $\forall x (ARTERIA(x) \supset sangre(x)=rica-oxigeno)$

Clase Vaso-sanguíneo

V $\forall x (VASO-SANGUINEO(x) \supset$
contiene(x)=fluido-sanguíneo)

VI $\forall x (VASO-SANGUINEO(x) \supset$
forma(x)=tubular)

Instancia Aorta es

instancia-de Arteria;
diámetro = 2,5

end

Clase Arteria es

subclase-de Vaso-sanguíneo;
sangre = rica-oxigeno

end

Clase Vaso-sanguíneo es

subclase-de T;
contiene= fluido-sanguíneo;
forma = tubular

end

Ejemplo derivación

- De I y III, aplicando IU (sobre III) y MP al resultado (con I):
 - I ARTERIA(Aorta)
 - III $\forall x (ARTERIA(x) \supset VASO-SANGUINEO(x))$
 - VII VASO-SANGUINEO(Aorta)

- De VII y V, aplicando IU (sobre V) y MP al resultado (con VII)
 - VII VASO-SANGUINEO(Aorta)
 - V $\forall x (VASO-SANGUINEO(x) \supset contiene(x)=fluido-sanguíneo)$
 - VIII contiene(Aorta)=fluido-sanguíneo

- De modo similar
 - IX sangre(Aorta)=rica-oxigeno
 - X forma(Aorta)=tubular



5. Herencia simple



5. Herencia simple

- Herencia simple: la jerarquía de herencia tiene estructura de árbol.
- Modelo básico sin facetas.
- Se permiten múltiples ocurrencias de un <par propiedad-valor> en un camino de herencia para la misma propiedad.



5. Herencia simple

- Excepción: se produce una excepción cuando el valor de una propiedad específica de un marco es distinto del valor de la misma propiedad en una de sus generalizaciones.
 - Valor excepcional: el del marco más específico.
- Modificación de la herencia: se hereda el valor más específico.
- Aporta: mayor flexibilidad diseño jerarquía, excepciones.




Ejemplo excepciones

Clase Vaso-sanguíneo **es subclase-de** T;
 contiene= fluido-sanguíneo;
 forma = tubular
end

Clase Arteria **es subclase-de** Vaso-sanguíneo;
 sangre = rica-oxígeno; *diámetro
end

Instancia Aorta **es instancia-de** Arteria;
 diámetro = 2,5
end

Instancia Arteria-pulmonar-izquierda **es instancia-de** Arteria;
 diámetro = 1,4;
 sangre=pobre-oxígeno
end





Descripción operacional de la herencia simple

Procedimiento Hereda(nombre-marco, pares-propiedad-valor)
 if nombre-marco=T **then**
 return(pares-propiedad-valor)
 end;
 pares \leftarrow Propiedades-clase(nombre-marco)
 pares-propiedad-valor \leftarrow pares-propiedad-valor \cup
 Nuevas-Propiedades(pares, pares-propiedad-valor);
 return(Hereda(ClaseSuperior(nombre-marco), pares-propiedad-valor))
end



Interés de la herencia simple

- Economía de la representación:
 - Las propiedades de clase solo se definen en la clase pertinente.
- Solución operacional al problema de la cualificación:
 - Las clases e instancias excepcionales modifican el valor de la propiedad de clase heredada.
 - ¿Semántica?
 - No es posible con la lógica clásica
 - Posible con lógicas no-monotónicas, para las que no existen métodos eficientes de derivación con carácter general.



6. Facetas



Facetas

- Permiten definir características adicionales de las propiedades y sus valores.
- Dos familias principales
 - Restricción: limitaciones sobre los valores
 - Tipo, Multivaluado, Default, Value
 - Demonios (valores activos o disparadores). Procedimientos invocados automáticamente por el sistema de marcos cuando se satisface alguna condición sobre el valor de un atributo
 - if-needed se necesita un valor, pero no existe valor actual
 - if-added se proporciona un valor al facet value
 - if-removed se elimina un valor del facet value



Sintaxis Marcos

Herencia simple y facetas (I)

<marco>	::=	<clase> <instancia>
<clase>	::=	Clase <nombre-clase> es subclass-de <espec-clase-superior> ; [<propiedades-clase> [; <declaración-propiedades-instancia>]] <declaración-propiedades-instancia> end
<instancia>	::=	Instancia <nombre-instancia> es instancia-de <nombre-clase> [; <propiedades-instancia>] end
<espec-clase-superior>	::=	<nombre-clase> T
<propiedades-clases>	::=	<propiedad-facetas-clase> {; <propiedad-facetas-clase> }* ε
<propiedades-instancia>	::=	<propiedad-facetas-instancia> {; <propiedad-facetas-instancia>}* ε
<declaración-propiedades-instancia>	::=	*<propiedad-facetas-declaración-instancia> REVISAR {;* <propiedad-facetas-declaración-instancia>}* ε



Sintaxis Marcos

Herencia simple y facetas (II)

$\langle \text{propiedades-clases} \rangle ::= \langle \text{propiedad-facetado-clase} \rangle \{ ; \langle \text{propiedad-facetado-clase} \rangle \}^* | \epsilon$

$\langle \text{propiedades-instancia} \rangle ::= \langle \text{propiedad-facetado-instancia} \rangle \{ ; \langle \text{propiedad-facetado-instancia} \rangle \}^* | \epsilon$

$\langle \text{declaración-propiedades-instancia} \rangle ::= * \langle \text{propiedad-facetado-declaración-instancia} \rangle \{ ; * \langle \text{propiedad-facetado-declaración-instancia} \rangle \}^* | \epsilon$

$\langle \text{propiedad-facetado-clase} \rangle ::= \langle \text{nombre-propiedad} \rangle : (\langle \text{faceta-clase} \rangle \{ , \langle \text{faceta-clase} \rangle \}^*)$

$\langle \text{propiedad-facetado-instancia} \rangle ::= \langle \text{nombre-propiedad} \rangle : (\langle \text{faceta-instancia} \rangle \{ , \langle \text{faceta-instancia} \rangle \}^*)$

$\langle \text{propiedad-facetado-declaración-instancia} \rangle ::= \langle \text{nombre-propiedad} \rangle : (\langle \text{faceta-declaración-instancia} \rangle \{ , \langle \text{faceta-declaración-instancia} \rangle \}^*)$



Facetas de propiedades de clase

`<faceta-clase>` ::= `<faceta-tipo>` | `<faceta-multivaluada>` | `<faceta-value>`

`<faceta-tipo>` ::= **type** `<tipo>`
`<tipo>` ::= **boolean** | **int** | **real** | **string** | `<enumerado>` | `<nombre-clase>`

`<faceta-multivaluada>` ::= **multivalue**

`<faceta-value>` ::= **value** `<valor>`

`<valor>` ::= `<constante-elemental>` | `<nombre-instancia>`



Facetas de declaración de propiedades de instancias

<faceta-declaración-instancia> ::= <faceta-tipo> | <faceta-multivaluada> | <faceta-default> |
| <faceta-value> | <faceta-demon>

<faceta-tipo> ::= **type** <tipo>
<tipo> ::= **boolean** | **int** | **real** | **string** | <enumerado> | <nombre-clase>

<faceta-multivaluada> ::= **multivalued**
<faceta-default> ::= **default** <valor>
<faceta-value> ::= **value** <valor>

<faceta-demon> ::= **demon** <tipo-demon> <llamada-demon>

<tipo-demon> ::= **if-needed** | **if-added** | **if-removed**

<llamada-demon> ::= <nombre-procedimiento> (<arg> {, <arg>}* | ε)

<arg> ::= a:<nombre-propiedad> | <valor>

<valor> ::= <constante-elemental> | <nombre-instancia>



Facetas de propiedades de instancias

<faceta-instancia> ::= <faceta-default> | <faceta-value>

<faceta-default> ::= **default** <valor>

<faceta-value> ::= **value** <valor>

<valor> ::= <constante-elemental> | <nombre-instancia>



Restricción adicional

- Para poder asignar un nombre de instancia como valor, es **necesario** que una marco superior haya declarado el tipo de la propiedad con un nombre de clase.
 - La instancia asignada ha de ser una instancias de la clase declarada.



Ejemplo facetas

Clase Persona **es subclase-de** T;
Animal-racional: (**type** {si, no}, **value** si);
*Nombre: (**type** string);
*Edad: (**type** {0,... 100});
*Nacionalidad: (**type** string, **multivalued**);
*Estado-civil: (**type** {soltero, casado, divorciado, viudo});
*Tipo-de-Matrimonio: (**type** {civil, religioso});
*Religión: (**type** string);
*Fecha-Nacimiento: (**type** Fecha)

end

Clase Hombre **es subclase-de** Persona;
Sexo: (**type** {varón, mujer}, **value** varón);
*Fecha-Boda: (**type** Fecha);
*Casado-con: (**type** Mujer)

end



Ejemplo facetas (continuación 1)

Clase Mujer **es subclase-de** Persona;
Sexo : (**type** {varón, mujer}, **value** mujer);
*Fecha-Boda: (**type** Fecha);
*Casado-con: (**type** Hombre)

end

Clase Fecha **es subclase-de** T;
*Día: (**type** {1, 2, ... 30});
*Mes: (**type** {Enero, Febrero, ... Diciembre});
*Año: (**type** {1900, 1901, ... 2030})

end



Ejemplo facetas (continuación 2)

Instancia Juan es **instancia-de** Hombre;
Nombre: (**value** Juan);
Edad: (**value** 30);
Nacionalidad: (**value** Española);
Estado-Civil: (**value** Casado);
Fecha-Nacimiento: (**value** Fecha-N-Juan);
Fecha-Boda: (**value** Fecha-Boda-J-M);
Casado-con: (**value** María)

End

Instancia María es **instancia-de** Mujer
Nombre: (**value** María);
Edad: (**value** 30);
Nacionalidad: (**value** Española, Argentina);
Estado-Civil: (**value** Casada);
Religión: (**value** Católica);
Fecha-Nacimiento: (**value** Fecha-N-María);
Fecha-Boda: (**value** Fecha-Boda-J-M);
Casado-con: (**value** Juan)

end

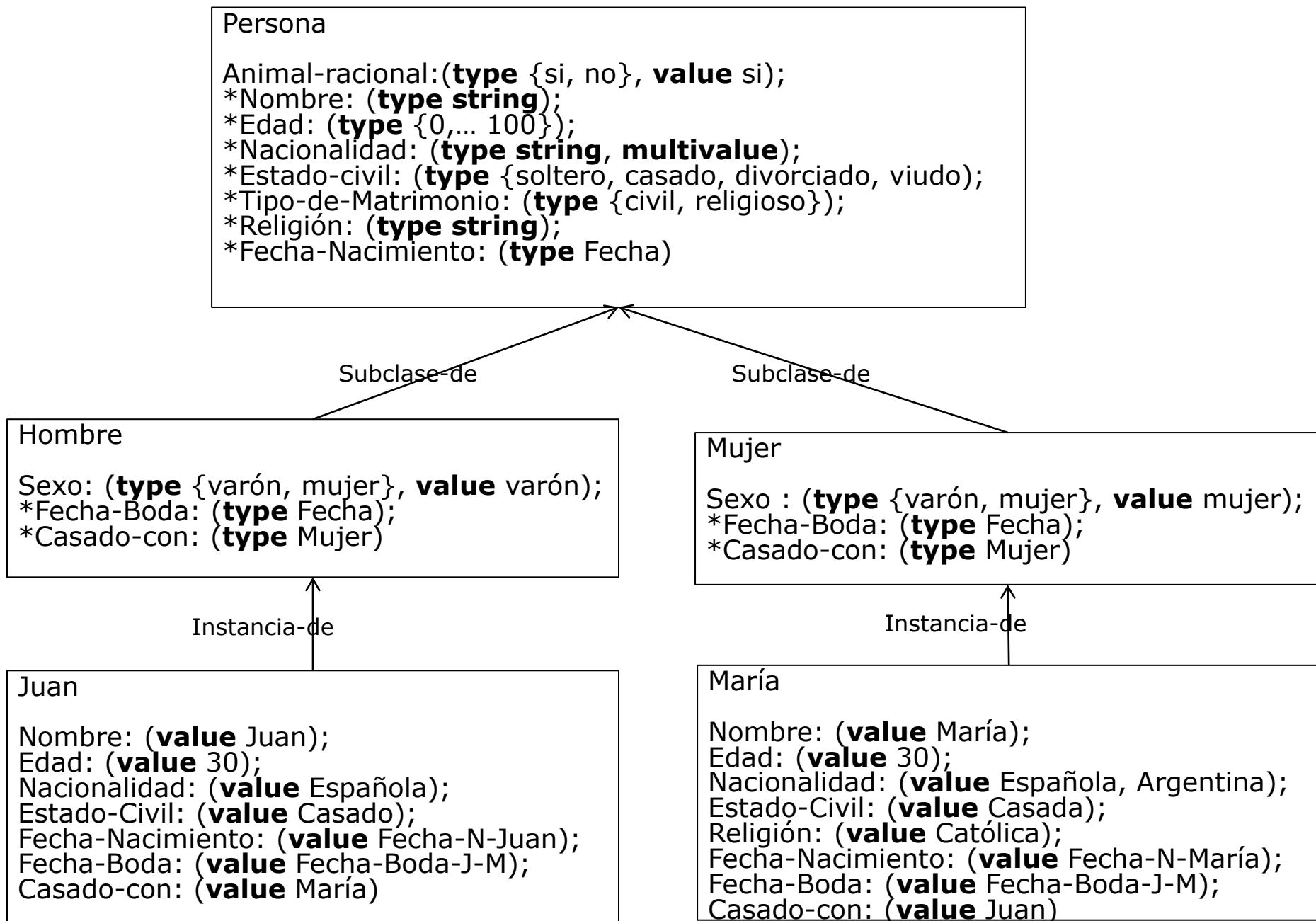


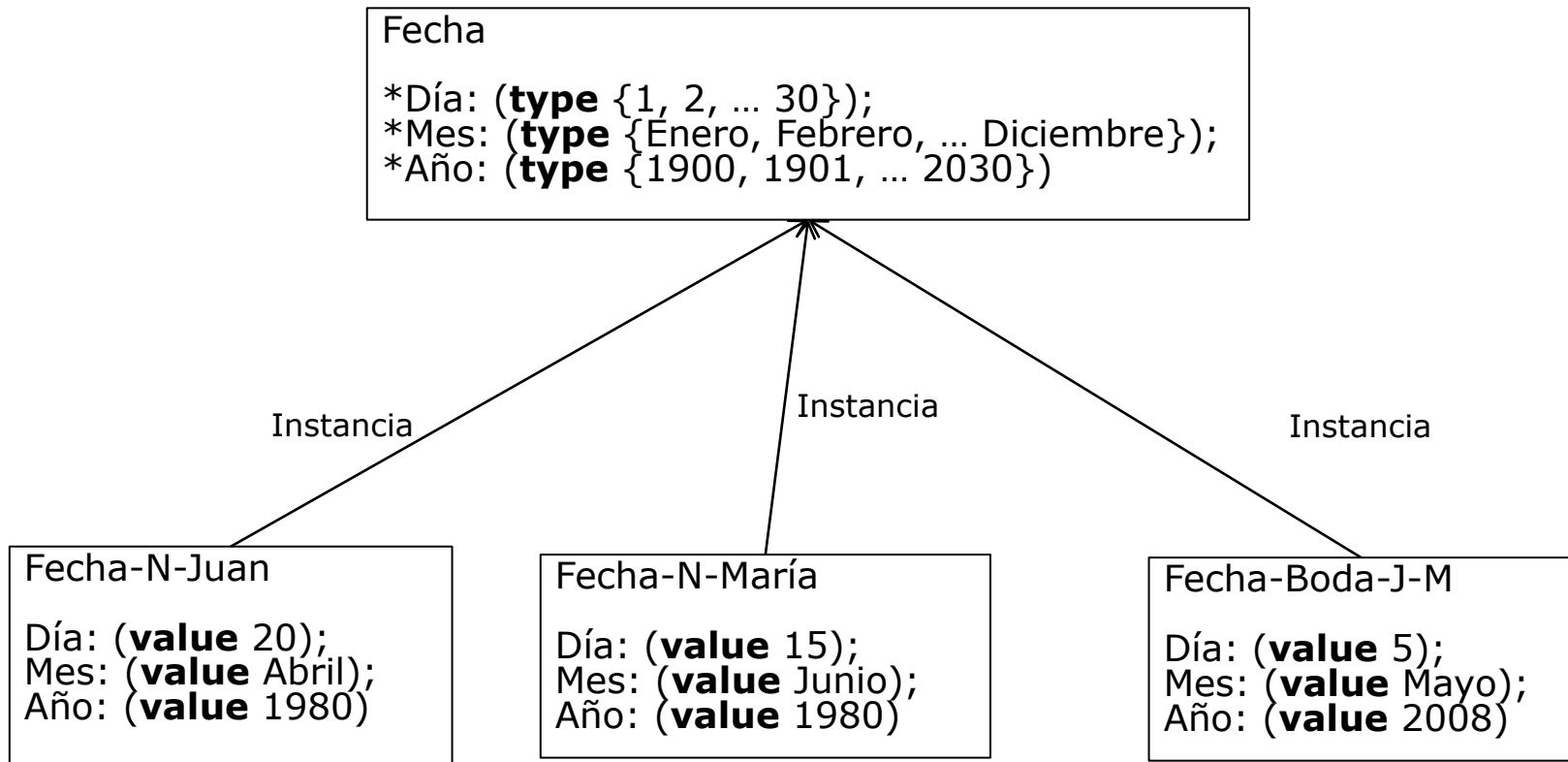
Ejemplo facetas (continuación 3)

Instancia Fecha-N-Juan **es instancia-de** Fecha;
 Día: (**value** 20);
 Mes: (**value** Abril);
 Año: (**value** 1980)
end

Instancia Fecha-N-María **es instancia-de** Fecha;
 Día: (**value** 15);
 Mes: (**value** Junio);
 Año: (**value** 1980)
end

Instancia Fecha-Boda-J-M **es instancia-de** Fecha;
 Día: (**value** 5);
 Mes: (**value** Mayo);
 Año: (**value** 2008)
end







Ejemplo facetas: demonios (I)

Clase Polígono **es subclase-de** T;
definición: (**type string, value** «Un polígono es una figura geométrica cerrada y plana limitada por tres o más líneas rectas que se cortan en sus vértices.»)
end

Clase Cuadrilátero **es subclase-de** Polígono;
numero-de-lados: (**type int, value 4**)
end

Clase Cuadrado **es subclase-de** Cuadrilátero;
*lado: (**type real**);
*área = (**type real, demon if-needed** producto(a:lado, a:lado))
end

Instancia Cuadrado-35 **es instancia-de** Cuadrado;
lado: (**value 2**)
end

¿Área del cuadrado-35?



¿Área del cuadrado-35?

- Consulta del valor del área del cuadrado-35:
 - No es una propiedad específica: se invoca el mecanismo de herencia.
 - Se hereda el demonio **if-needed** de la clase superior que declara el atributo de la instancia (clase cuadrado).
 - Se llama el procedimiento *producto* desde la instancia cuadrado-35.
 - Valor de los argumentos: valor de la propiedad de la instancia que hace la llamada, 2.



```
Instancia Arteria-pulmonar-izquierda es instancia-de Arteria;  
    diámetro: (value 1,4); sangre: (value pobre-oxigeno)  
    flujo-sanguíneo: (value 10); resistencia: (default 1)  
end
```



Valores de algunas propiedades (I)

- ¿Presión sanguínea de la aorta?: 21.
 - La faceta **value** tiene preferencia sobre **default**.
- ¿Flujo sanguíneo de la aorta?: 7.
 - La faceta **value** tiene preferencia sobre **default**.
- ¿Resistencia de la aorta?: 3.
 - Calculado por el demonio **if-needed**, heredado.
 - Invocando el procedimiento *R*.
 - Con los valores de las propiedades de la instancia que lo invoca: 21, 7.



Valores de algunas propiedades (II)

- ¿Presión sanguínea de la arteria pulmonar izquierda?: 16.
 - Heredado de la faceta **default**.
- ¿Flujo sanguíneo de la arteria pulmonar izquierda?: 10.
 - La faceta **value** tiene preferencia sobre **default**.
- ¿Resistencia de la arteria pulmonar izquierda?:
 - Depende de como se defina la herencia.

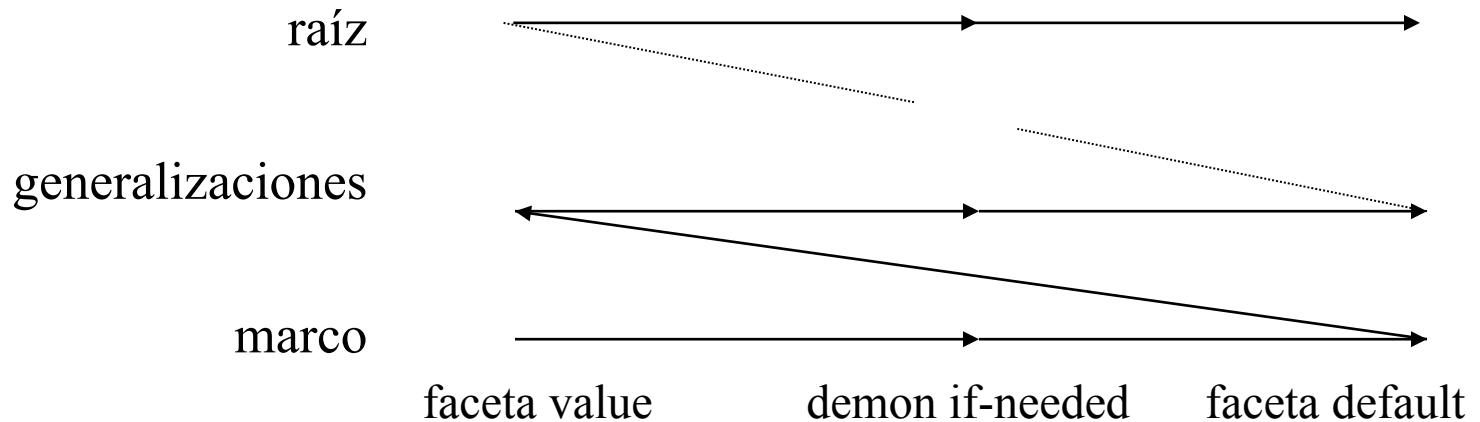


Herencia y facetas

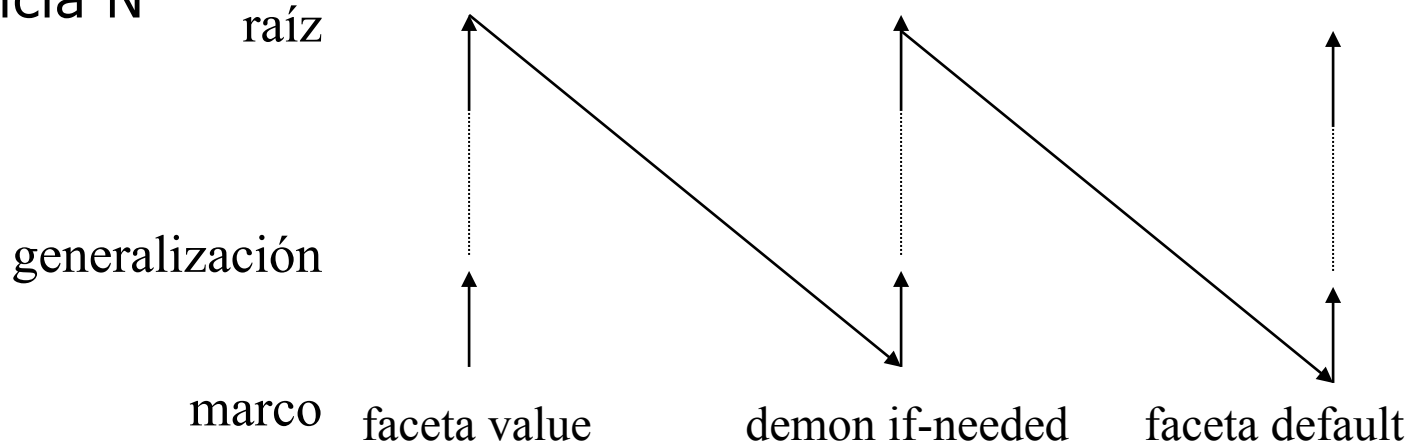
- Interacción entre la Herencia y las facetas.
- Tenemos tres formas de proporcionar valor a una propiedad:
 - Facetas **value**, **if-needed** y **default**.
- Todas ellas se pueden heredar.
- Opciones:
 - Priorizar las facetas locales frente a las heredadas: herencia Z.
 - Priorizar las facetas heredadas frente a las locales: herencia N.

Facetas y Herencia: Herencia N, Z

■ Herencia Z



■ Herencia N





```

Clase Arteria es subclase-de Vaso-sanguíneo;
sangre: (value rica-oxigeno);
*diámetro: (type real);
*presión-sanguínea: (type real, default 16);
*flujo-sanguíneo: (type real, default 8);
*resistencia: (type real, demon if-needed R(a:presión-sanguínea,
a:flujo-sanguíneo), default 2)
end

```

```
Instancia Aorta es instancia-de Arteria;  
    diámetro: (value 2,5);  
    presión-sanguínea: (value 21); flujo-sanguíneo: (value 7)  
end
```

```
Instancia Arteria-pulmonar-izquierda es instancia-de Arteria;  
    diámetro: (value 1,4); sangre: (value pobre-oxigeno)  
    flujo-sanguíneo: (value 10); resistencia: (default 1)  
end
```



Valores de algunas propiedades (III)

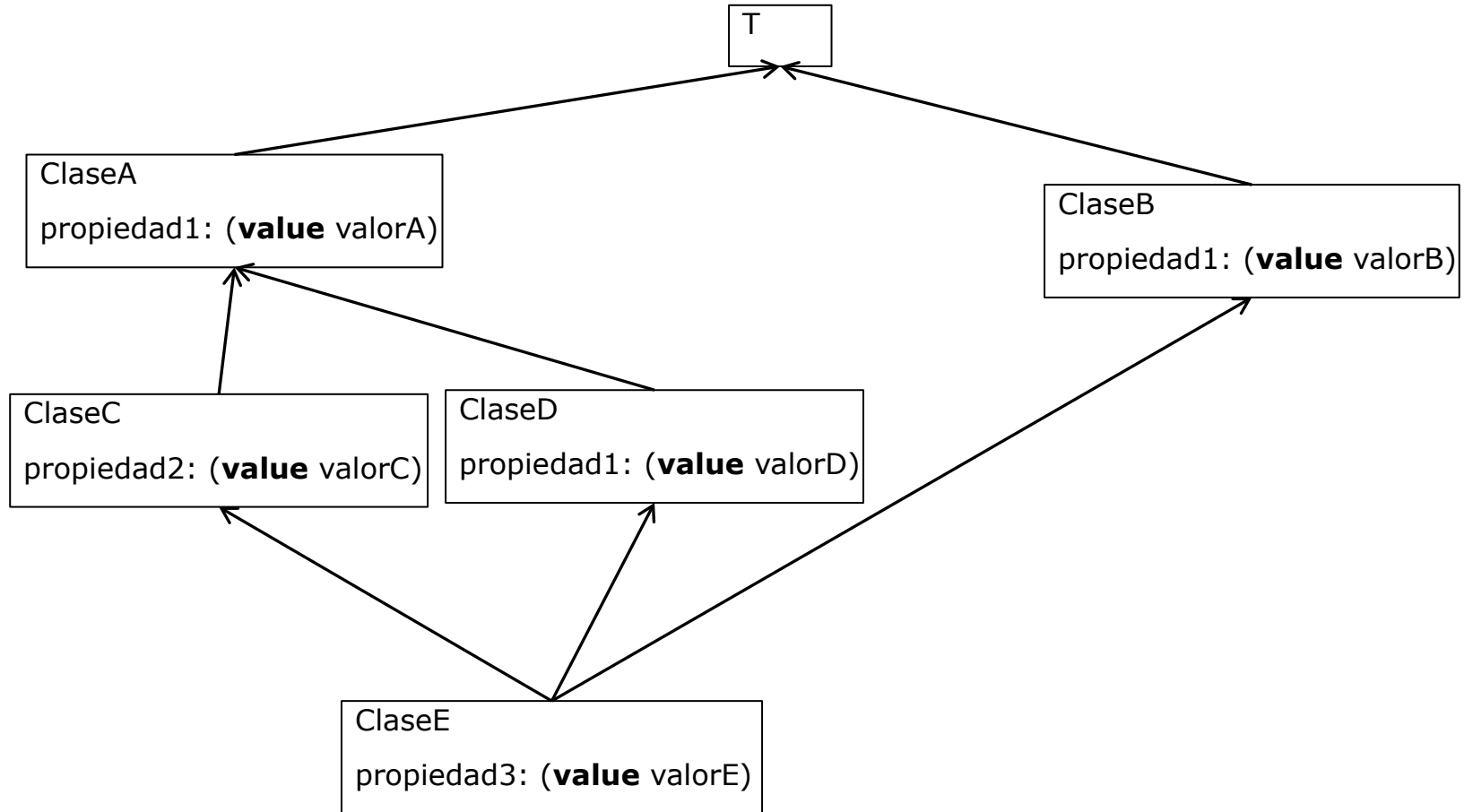
- ¿Presión sanguínea de la arteria pulmonar izquierda?: 16.
 - Heredado de la faceta **default**.
- ¿Flujo sanguíneo de la arteria pulmonar izquierda?: 10.
 - La faceta **value** tiene preferencia sobre **default**.
- ¿Resistencia de la arteria pulmonar izquierda?:
 - Herencia Z: 1
 - Por la faceta **default** definida en la instancia.
 - Herencia N: 1,6
 - Por la faceta **if-needed** heredada.



Herencia múltiple

- Un marco puede tener más de un padre:
 - Declarar una instancia de varias clases.
 - Declarar una clase de varias superclases.
- Jerarquía de herencia: grafo dirigido acíclico.
- Herencia de propiedades:
 - Sin excepciones: no hay ocurrencia de pares propiedad-valor con distinto valor para la misma propiedad:
 - Similar a la herencia simple sin excepciones.
 - Con excepciones: hay ocurrencias de pares propiedad-valor con distinto valor para la misma propiedad:
 - Dificultades.

Ejemplo herencia múltiple con excepciones





Ejemplo

herencia múltiple con excepciones

Clase ClaseA **es subclase-de** T;
 propiedad1: (**value** valorA)
end

Clase ClaseB **es subclase-de** T;
 propiedad1: (**value** valorB)
end

Clase ClaseC **es subclase-de** ClaseA;
 propiedad2: (**value** valorC)
end

Clase ClaseD **es subclase-de** ClaseA;
 propiedad1: (**value** valorD)
end

Clase ClaseE **es subclase-de** ClaseC, ClaseD, ClaseB;
 propiedad3: (**value** valorE)
end

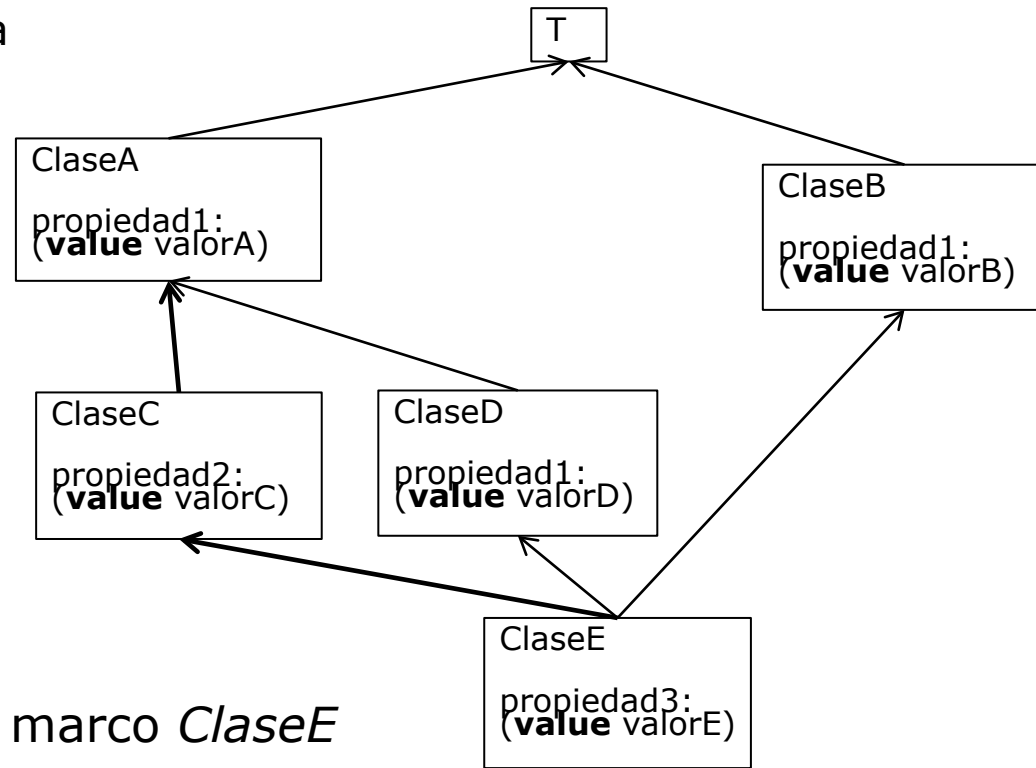


Herencia múltiple con excepciones

- Ocurrencias de pares propiedad-valor con distinto valor para la misma propiedad.
- El valor heredado de la propiedad depende del algoritmo de búsqueda que recorra el grafo.
 - Se hereda el primer valor encontrado para la propiedad.
- Algunos algoritmos:
 - Primero en profundidad.
 - Primero en anchura.

Valor de *propiedad1* en marco *ClaseE*

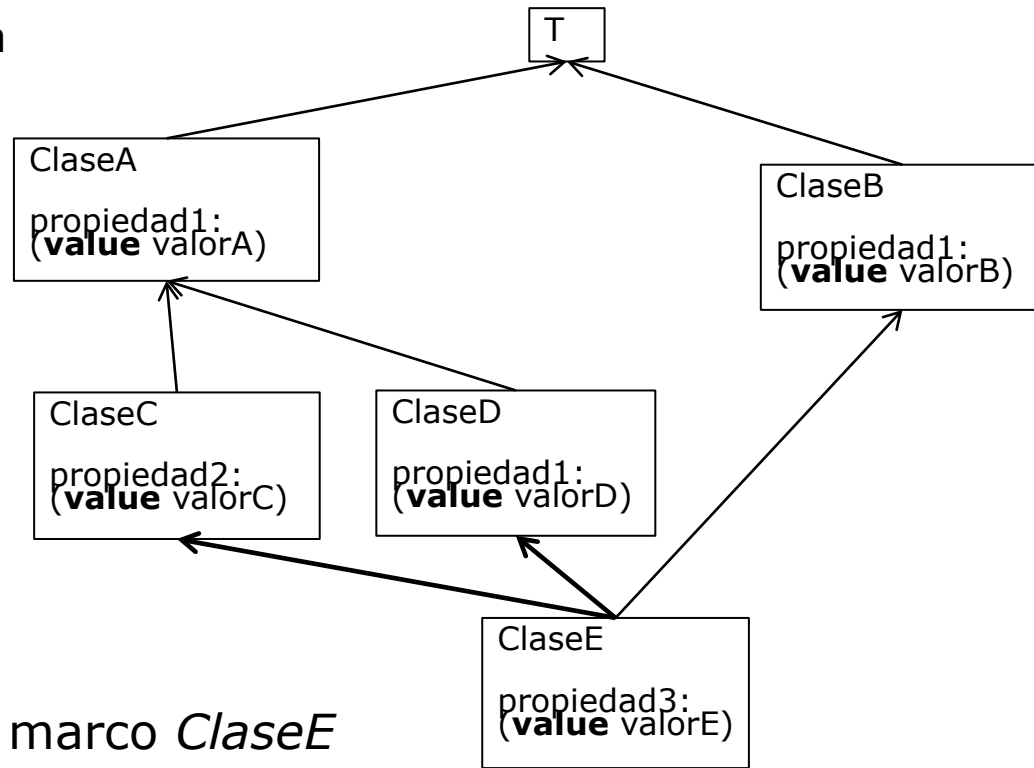
- Primero en profundidad
 - De izquierda a derecha



- Valor de *propiedad1* en marco *ClaseE*
 - *valorA*

Valor de *propiedad1* en marco *ClaseE*

- Primero en anchura
 - De izquierda a derecha



- Valor de *propiedad1* en marco *ClaseE*
 - *valorD*

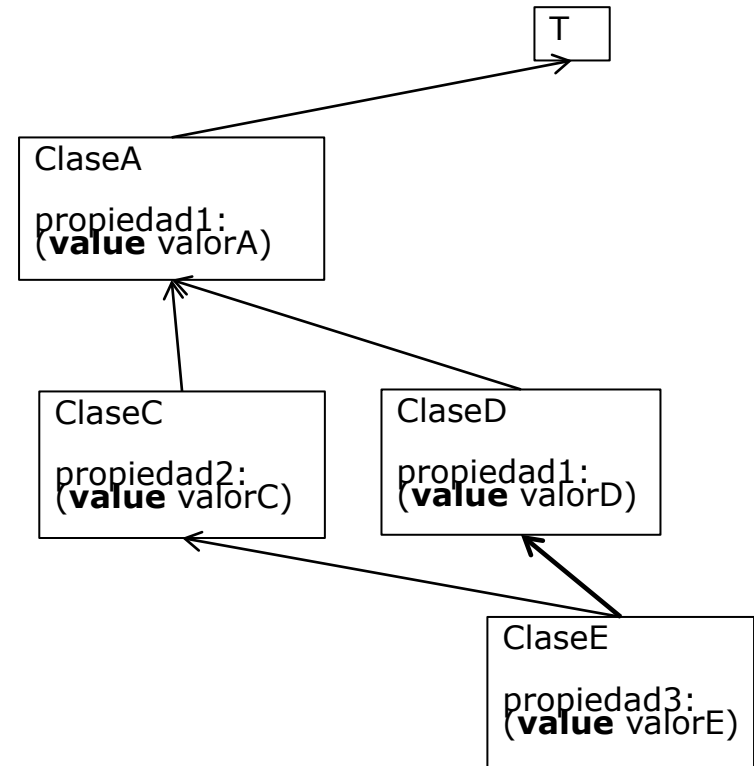


Algunas soluciones (I)

- Distancia Inferencial de Touretzky
 - La condición necesaria y suficiente para que la $clase_1$ esté más cercana a la $clase_2$ que a la $clase_3$ es que la $clase_1$ tenga un camino de inferencia a través de la $clase_2$ hacia la $clase_3$.
 - Si la $clase_2$ está entre la $clase_1$ y la $clase_3$.
 - Criterio: heredar el valor de la clase más cercana.
- Establece un orden parcial: solo entre marcos conectados por un camino de herencia.
 - Soluciona problema de excepciones entre marcos conectados.
 - No evita contradicciones entre marcos no conectados.

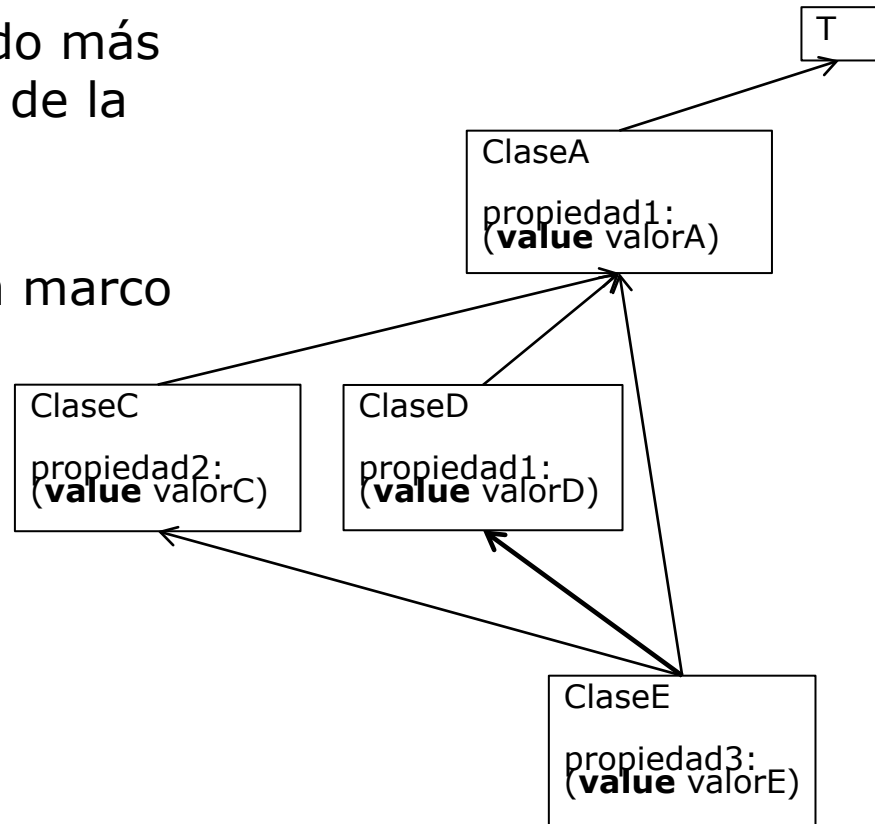
Distancia Inferencial de Touretzky (I)

- La *ClaseE* está más cerca de la *ClaseD* que de la *ClaseA*
- Valor de *propiedad1* en marco *ClaseE*
 - *valorD*



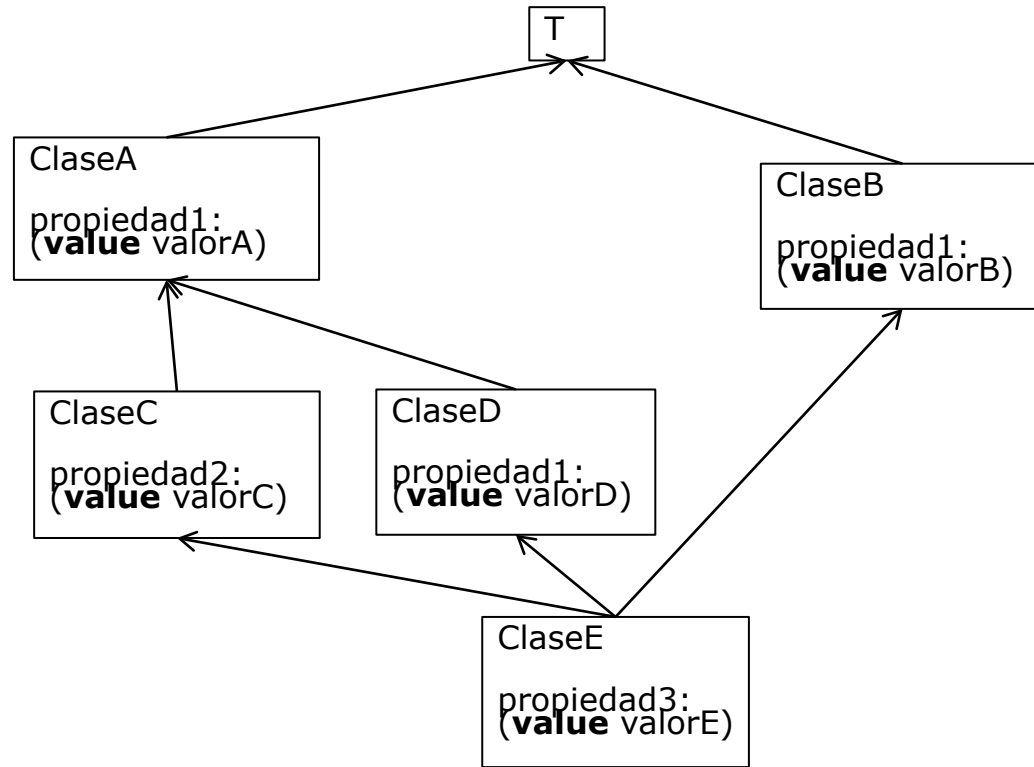
Distancia Inferencial de Touretzky (II)

- La *ClaseE* sigue estando más cerca de la *ClaseD* que de la *ClaseA*
- Valor de *propiedad1* en marco *ClaseE*
 - *valorD*



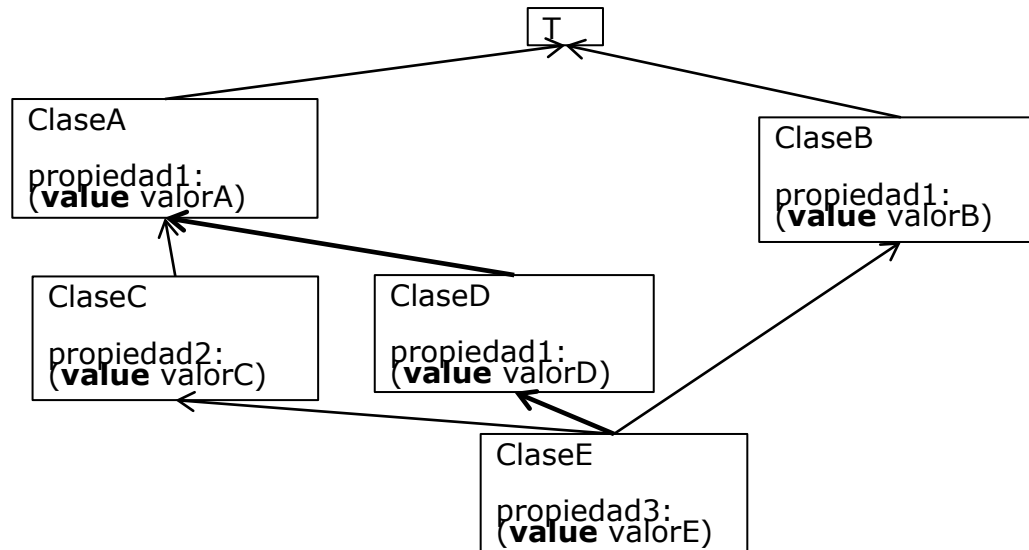
Distancia Inferencial de Touretzky (III)

- La *ClaseE* está más cerca de la *ClaseD* que de la *ClaseA*.
- La clase *ClaseB* no está ordenadas respecto a la clase *ClaseD*.
- No proporciona ningún criterio para heredar *propiedad1* en *ClaseE*



Algunas soluciones (II)

- Proporcionar un orden total.
 - El ingeniero de conocimiento en el diseño de clases.
 - Asociada a algún algoritmo de búsqueda.
 - Por ejemplo, primero en profundidad con el criterio de la distancia inferencial de Touretzky: ClaseE, ClaseD, ClaseA, ClaseC, ClaseB.
 - Valor heredado: valorD





Algunas soluciones (III)

- Permitir excepciones, pero no contradicciones
 - No permitir ocurrencias de pares propiedad-valor con distinto valor para la misma propiedad en marcos no conectados por un camino de herencia.



8. Marcos y Reglas



Integración de marcos y reglas

- Interés
 - Marcos: conocimiento estructural.
 - Reglas: conocimiento heurístico.
- Integración conceptualmente sencilla: permitir que las reglas accedan a los valores de las propiedades de los marcos.
- Alternativas:
 - Ampliar sistema de marcos con clase reglas.
 - Sistema de producción con sistema de marcos como memoria de trabajo.



Propuesta de integración

- Marcos: modelo estándar.
 - Herencia (simple o múltiple).
 - Facetas.
- Reglas: lenguaje O-A-V con variables.
 - O ::= <nombre-clase> | <nombre-instancia> | <variable>
 - A ::= <nombre-propiedad>
 - V ::= <constante-elemental> | <nombre-instancia> | <variable>
 - <variable> ::= ?<nombre-variable>
 - <condición> ::= <predicado> (O, A, V)
 - <conclusión> ::= <acción>(O, A, V | =llamada-función)

Ejemplo puerta OR exclusivo (I)

Clase PuertaLógica **es subclase-de** T;

*en1: (**type** Terminal);

*en2: (**type** Terminal);

*sal: (**type** Terminal)

end

Clase PuertaXOR **es subclase-de** PuertaLógica;

Comentario :(**value** "Puerta OR exclusivo")

end

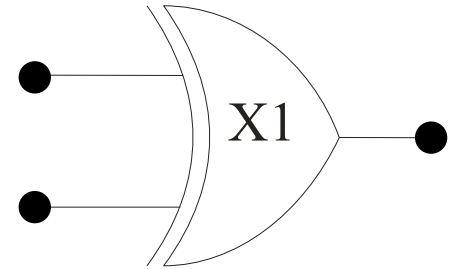
Instancia X1 **es instancia-de** PuertaXOR;

en1: (**value** en1X1);

en2: (**value** en2X1);

sal: (**value** salX1)

end



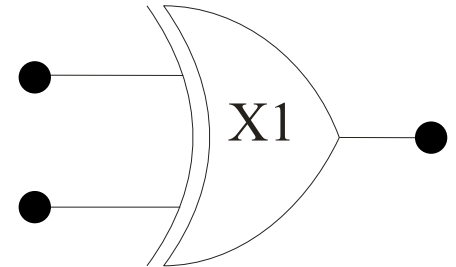
Ejemplo puerta OR exclusivo (II)

Clase Terminal **es subclase-de** T;
 *valor: (**type boolean**)
end

Instancia en1X1 **es instancia-de** Terminal
end

Instancia en2X1 **es instancia-de** Terminal
end

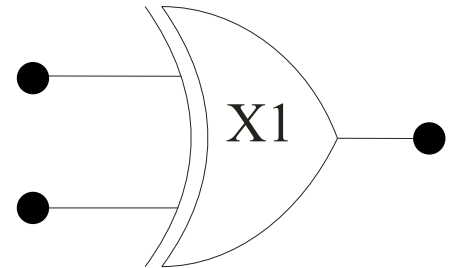
Instancia salX1 **es instancia-de** Terminal
end



Ejemplo puerta OR exclusivo (III)

- Regla para obtener el valor de la señal en el terminal de salida de una puerta XOR conocido el valor de la señal en los terminales de entrada.

```
if iguales(?x, instancia-de, PuertaXOR) and  
    iguales(?x, en1, ?tin1) and  
    iguales(?tin1, valor, ?val1) and  
    iguales(?x, en2, ?tin2) and  
    iguales(?tin2, valor, ?val2) and  
    iguales(?x, sal, ?tsal)  
then añadir(?tsal, valor, =xor(?val1, ?val2)) fi
```





9. Resumen

- Permiten estructurar el dominio.
 - Conocimiento estructural.
- El mecanismo de herencia permite una representación concisa del dominio:
 - Conocimiento general: clases.
 - Descripción instancias específica: instancias.
 - Obtención de conocimiento derivado a partir del conocimiento general: herencia de propiedades.
 - Resuelve problema de la cualificación en LPO, a cambio de dificultades semánticas si herencia múltiple.
- Los demonios permiten representar conocimiento operacional.