

Tema 4: Sistemas de representación estructurados

Anabel Gómez Ríos

18 de mayo de 2016

1. Introducción

Son una representación del conocimiento mediante grafos (conceptos, relaciones). Facilitan la representación del conocimiento humano y hay dos tipos: redes semánticas y frames.

2. Redes Semánticas

Las redes semánticas (introducidas por R. Quillian, 1968) son una representación en procesamiento de lenguaje natural. Son un formalismo muy limitado para dominios más complejos y limitado para tratar con formas de inferencia sofisticada. Fueron las precursoras de los frames.

Una red semántica se representa como un grafo dirigido etiquetado (en algunos casos se exige que dicho grafo sea acíclico), constituido por **nodos**, que representan conceptos (un objeto individual o una clase de objetos) y **arcos**, que representan relaciones binarias entre los conceptos.

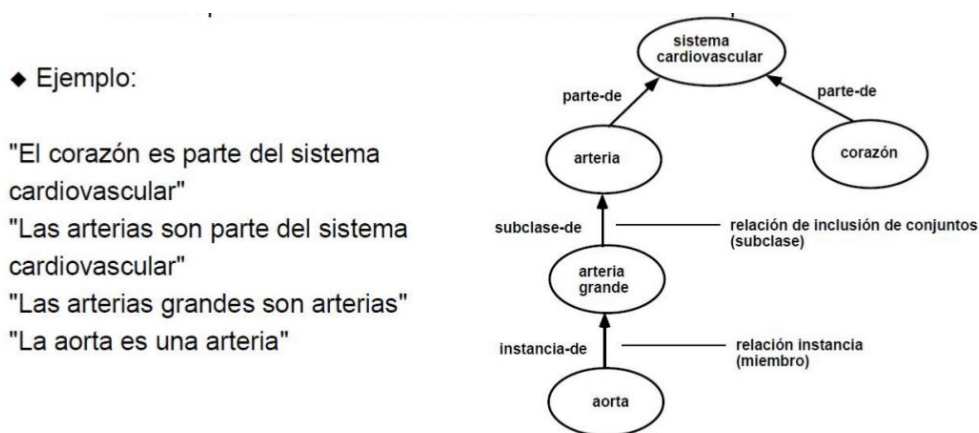


Figura 1: Ejemplo básico de red semántica

♦ Ejemplo:

"Las arterias pequeñas son arterias"

"La arteria branquial izquierda es una arteria grande"

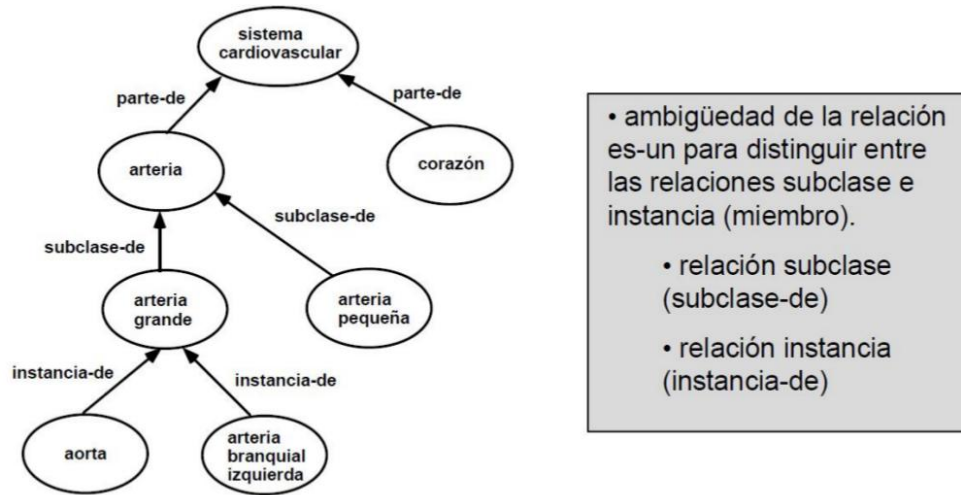


Figura 2: Ejemplo de red semántica

2.1. Herencia en Redes Semánticas

La herencia es el mecanismo de razonamiento utilizado en redes semánticas. Un concepto (nodo) hereda las propiedades de los conceptos *más altos en jerarquía* a través de las relaciones subclase-de e instancia-de.

Ejemplo:

1. Un vaso sanguíneo es parte del sistema cardiovascular.
2. Las arterias contienen sangre rica en oxígeno.
3. Las arterias tienen pared muscular.
4. La arteria pulmonar izquierda es una arteria grande.

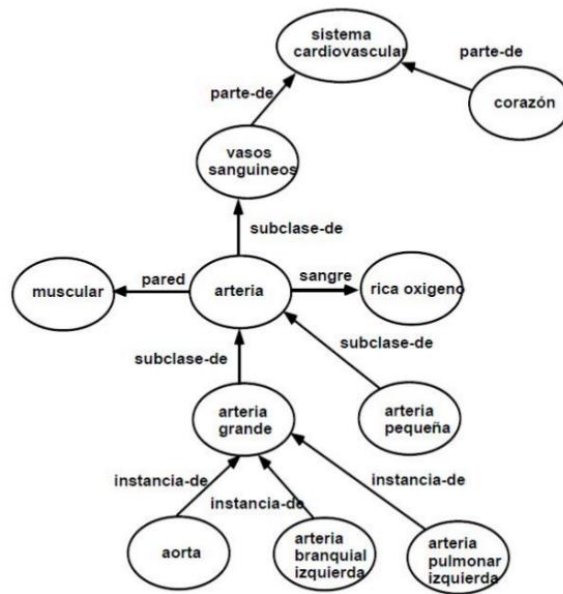


Figura 3: Ejemplo de red semántica con herencia

A partir de la red semántica podemos deducir:

1. Las arterias grandes son ricas en oxígeno.
2. Las arterias grandes tienen pared muscular
3. La aorta contiene sangre rica en oxígeno.
4. La aorta tiene pared muscular.

2.2. Excepciones en la Herencia

- a) No heredar propiedades que producen inconsistencias. Por ejemplo en el ejemplo anterior, *La arteria pulmonar izquierda contiene sangre pobre en oxígeno* y *La arteria pulmonar izquierda tiene pared muscular y es rica en oxígeno*. La propiedad *Las arterias transportan sangre rica en oxígeno* no debe ser heredada (excepción) por la arteria pulmonar izquierda.

Una posible solución es almacenar la propiedad como información explícita en cada concepto en el que se cumple la propiedad, eliminando la propiedad general.

- b) No heredar propiedades que consideramos relevantes para una clase, pero no para sus especializaciones, como por ejemplo *Existen 100 arterias*.

3. Redes Semánticas Extendidas

Las redes semánticas extendidas (A. Deliyanni y R. A. Kowalski) son un formalismo de representación alternativo a la forma clausal de la lógica con la restricción de sólo poder utilizar símbolos de predicado binarios. Debido a la equivalencia sintáctica entre redes semánticas

extendidas y la forma clausal de la lógica, las reglas de inferencia definidas para la forma clausal de la lógica pueden ser aplicadas para manipular arcos y nodos de una red semántica extendida.

Un predicado binario puede ser traducido en una red en la que los nodos representan términos y el arco representa la relación (predicado)

La restricción a símbolos de predicado binarios no es crítica, ya que cualquier átomo que contenga un símbolo de predicado n-ario puede ser reemplazado por una conjunción de átomos que contengan sólo símbolos de predicado binarios.

- Si $n > 2$, se requieren $n+1$ nuevos predicados.
- Si $n = 1$, sólo se requiere un nuevo predicado

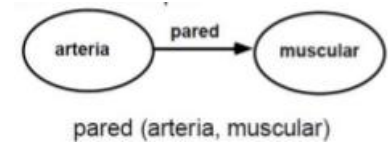


Figura 4: Ejemplo de red semántica extendida

3.1. Ejemplos Redes Semánticas

3.1.1. Ejemplo 1: Traducción predicado terciario a binario

$\text{PresiónSangre}(x, y, z) = \text{"la presión sanguínea de } x \text{ varía entre } y \text{ mmHg y } z \text{ mmHg"}$.

El predicado $\text{PresiónSangre}(\text{arteria}, 40, 80)$ puede ser reemplazado por la conjunción de predicados binarios:

```

instancia-de(presión1, presiónsangre)
sujeto(presión1, arteria)
Límiteinferior(presión1, 40)
Límitesuperior(presión1, 80)

```

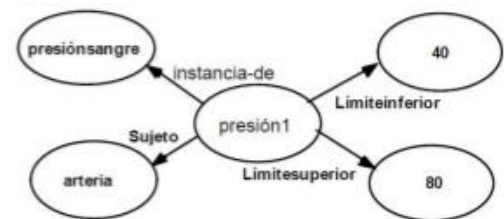


Figura 5: Ejemplo de red semántica extendida

3.1.2. Ejemplo 2: Traducción predicado unario a binario

Supongamos el siguiente predicado unario $\text{Arteria}(x) = \text{"x es una arteria"}$ y las cláusulas $\text{Arteria}(\text{aorta})$ y $\text{Arteria}(\text{arteria-grande})$. Estas cláusulas pueden ser reemplazadas por las cláusulas $\text{instancia-de}(\text{aorta}, \text{arteria})$ y $\text{subclase-de}(\text{arteria-grande}, \text{arteria})$

3.1.3. Ejercicio

Representar mediante redes semánticas la siguiente información:

Una persona tiene dos brazos y dos piernas.

Las personas pueden ser hombres y mujeres.

Un jugador de baloncesto es un hombre.

Michael Jordan es un jugador de baloncesto y juega de escolta.

Shaquille O'Neil es un jugador de baloncesto y juega de pivot.

La media de puntos de un escolta es 20.

La media de puntos de Jordan es 20.

La media de puntos de un pivot es 20.
El peso de un jugador de baloncesto es 120 kilos.
ordan pertenece al equipo de los Bulls.
O' Neil pertenece al equipo de los Lakers.

4. Frames

Los frames (introducidos por Minsky en 1975) se basan en el concepto de considerar la resolución de problemas humana como el proceso de rellenar huecos de descripciones parcialmente realizadas (O. Selz).

La idea subyacente en un sistema basado en frames es que el conocimiento concerniente a individuos o clases de individuos, incluyendo las relaciones entre los mismos, es almacenada en una entidad compleja de representación llamada frame (unidad, objeto, concepto).

Un conjunto de frames que representan el conocimiento de un dominio de interés es organizada jerárquicamente en lo que es llamado una taxonomía (asociada a un método de razonamiento automático llamado herencia).

El conocimiento relevante de un concepto (objeto individual o clase de objetos) es representado mediante entidad compleja de representación llamada frame, constituida por un conjunto de propiedades (atributos). Los frames proporcionan un formalismo para agrupar explícitamente todo el conocimiento concerniente a las propiedades de objetos individuales o clases de objetos.

Tipos de frames:

1. Frames clase, o frames genéricos, que representan conocimiento de clases de objetos.
2. Frames instancia, que representan conocimientos de objetos individuales

4.1. Jerarquía o taxonomía de frames

El conocimiento de un dominio de interés es organizado jerárquicamente en una jerarquía o taxonomía de frames. La taxonomía es representada mediante un grafo dirigido acíclico (generalmente un árbol) en el que **sólo** se dan las relación de instancia-de y de subclase-de, y cada nodo denota un frame.

La raíz del árbol es la descripción más general del dominio y las hojas del árbol descripciones de conceptos más específicos.

Las especializaciones (instancias, subclases, subframes) se representan como descendientes de un frame en la taxonomía. No se pueden definir especializaciones de los frames instancia (con excepción de las metaclases). Las generalizaciones (superclases, superframes) se representan como antecesores de un frame en la taxonomía.

Las propiedades de los frame más generales son heredadas por sus especializaciones (herencia).

4.2. Definición de frame



Cada frame de una taxonomía tiene un nombre único. Un frame sólo puede tener una superclase (herencia simple). La información (propiedades) específica al concepto representado por un frame es representada mediante atributos o slots. Los atributos ofrecen un medio de representar las propiedades de objetos individuales o clases de objetos.

Figura 6: Definición de frame

4.2.1. Sintaxis de frames

<code><frame> ::=</code>	<code><clase> <instancia></code>
<code><clase> ::=</code>	<code>clase <nombre-clase> es</code> <code> subclase-de <espec-super>;</code> <code> <atributos-clase></code> <code>end</code>
<code><instancia> ::=</code>	<code>instancia <nombre-instancia> es</code> <code> instancia-de <espec-super>;</code> <code> <atributos-instancia></code> <code>end</code>
<code><espec-super> ::=</code>	<code><nombre-clase> nil</code>
<code><atributos-clase> ::=</code>	<code><declaración> {; <declaración>* <vacío></code>
<code><atributos-instancia> ::=</code>	<code><par-atributo-valor> {; <par-atributo-valor>* <vacío></code>
<code><declaración> ::=</code>	<code><par-atributo-tipo> <par-atributo-valor></code>
<code><par-atributo-tipo> ::=</code>	<code><nombre-atributo> : <tipo></code>
<code><par-atributo-valor> ::=</code>	<code><nombre-atributo> = <valor></code>
<code><tipo> ::=</code>	<code>entero real string <conjunto> <nombre-clase></code>
<code><valor> ::=</code>	<code><constante> <nombre-instancia></code>
	<code><vacío> ::=</code>

Figura 7: Sintaxis de un frame

El símbolo nil denota que un frame es la raíz de la taxonomía. <conjunto> denotará un conjunto enumerado de constantes elementales y/o nombres de instancias. Se asume que un par atributo/tipo o atributo/valor ocurre una única vez en una taxonomía (posteriormente se elimina esta suposición).

4.2.2. Ejemplo

<pre>instancia arteria-branquial-izquierda es instancia-de arteria; diámetro = 0.4; situación = brazo; sangre = rica-oxigeno end instancia brazo es instancia-de miembro; posición = superior end</pre>	<pre>clase arteria es subclase-de vasos-sanguíneos; situación: {brazo, cabeza, pierna, tronco} end clase miembro es subclase-de nil; posición:{superior, inferior} end</pre>
---	---

Figura 8: Ejemplo de frame

La información especificada en las partes atributo de los frames instancia sigue las siguientes reglas:

1. Todos los atributos que ocurren en las instancias de un frame clase deben haber sido declarados en dicha frame clase o en una de sus generalizaciones.
2. Los valores asignados a los atributos de la instancia deben ser del tipo de datos definido en alguna de sus generalizaciones.

En la declaración de un frame clase también se puede asignar valor a un atributo (están permitidos los pares atributo/valor).

<pre>clase arteria es subclase-de vasos-sanguíneos; situación : {brazo, cabeza, pierna, tronco}; sangre = rica-oxigeno end</pre>	<pre>instancia arteria-branquial-izquierda es instancia-de arteria; diámetro = 0.4; situación = brazo end</pre>
--	---

“ se considera adecuado especificar el valor rica-oxigeno del atributo sangre en la especificación de la clase, con lo que no es necesario especificarlo en sus subclases o instancias, ya que esta propiedad será heredada por todos los descendientes de la clase”

Figura 9: Ejemplo de frame

4.3. Equivalencia frames / redes semánticas

clases, instancias y valores atributos = conceptos (nodos)
atributos = relaciones (arcos)

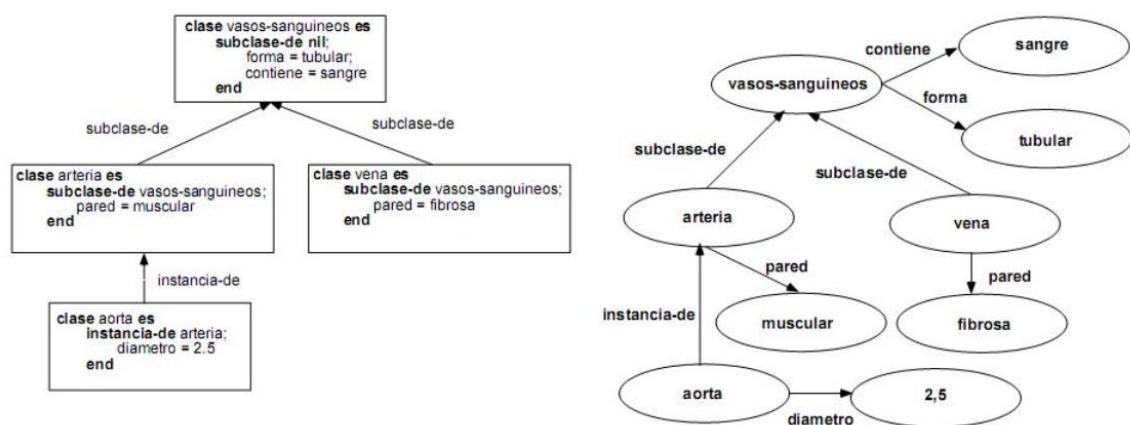


Figura 10: Equivalencia con redes semánticas

4.4. Herencia simple

Herencia simple: frames con una única superclase, taxonomías de tipo árbol. Herencia múltiple: frame con más de una superclase, taxonomías de tipo grafo. Tipos de información sobre atributos en una taxonomía de frames: información sobre tipo de atributo e información sobre valor de atributo. Los enlaces instancia-de y subclase-de definen una ordenación parcial de frames clase de la taxonomía, que puede ser utilizado para razonar sobre los valores de los atributos de la misma forma que en las redes semánticas (las especializaciones 'heredan' los valores de los atributos de las generalizaciones). El enlace subclase-de puede ser considerado como una relación que restringe los contenidos semánticos en la taxonomía de frames (los valores de un atributo de una subclase están restringidos por el tipo de atributo especificado en la superclase).

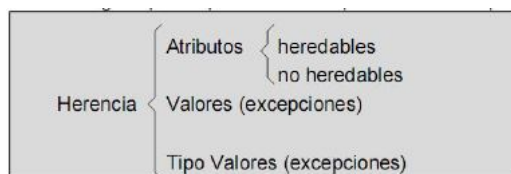


Figura 11: Herencia

La herencia simple consiste en que un frame hereda todos los atributos de sus superclases, así como los valores de estos atributos.

Consideremos lo siguiente desde un punto de vista de la lógica de preidcados de primer orden: $\{arteria(aorta), \forall x (arteria(x) \rightarrow pared(x) = muscular)\}$. Aplicando el Modus Ponens obtenemos: $pared(aorta) = muscular$. De forma similar derivaríamos: $vasos-sanguíneos(aorta)$. Esta información es heredada por la áorta desde una información general de las arterias.

Este tipo de razonamiento es modelado en un método de inferencia para frames llamado *herencia simple*. Informalmente podemos definir el procedimiento de herencia simple de la siguiente forma: Recorrer la taxonomía desde un frame específico hasta la raíz de la misma y coleccionar sucesivamente los atributos de los frames encontrados y sus valores asociados.

4.4.1. Excepciones de la herencia simple

Unicidad atributos en taxonomía disminuye expresividad.

Asumiremos atributos mono-valor.

Un atributo puede ocurrir en más de un frame.

Solución: concepto de *excepción* (propiedad general que no se cumple para algún/os objeto/s del dominio): Si en una especificación de una subclase o instancia, se especifica un valor de un atributo que ha sido especificado también en una generalización de la misma, se mantiene dicho valor y no se hereda el valor especificado en las generalizaciones (los descendientes de este nuevo frame heredarán un nuevo valor especificado como excepción).

4.5. Extensión de la definición de Frame

El formalismo de frames descrito no permite saber si el valor del atributo de una instancia ha sido heredado o ha sido especificado explícitamente ni permite calcular los valores de un atributo a partir de los valores de otros atributos.

Muchos lenguajes de frames proporcionan constructores especiales del lenguaje llamados facetas, que permiten manejar las funcionalidades anteriores.

Una faceta es considerada como una propiedad asociada a un atributo.

1. **faceta valor**, es la más común y referencia el valor real del atributo.
2. **faceta valor por defecto**, denota el valor inicial del atributo en caso de que no se especifique lo contrario.
3. **faceta tipo valor**, especifica el tipo de dato del valor del atributo.
4. **faceta cardinalidad**, especifica si se trata de un atributo uni o multi-valuado.
5. **faceta máxima cardinalidad**, sólo es válida para atributos multi-valorados y especifica el máximo número de valores asociados al atributo.
6. **facetas demonio**, permiten la integración de conocimiento declarativo y procedural. Un demonio o valor activo es un procedimiento que es invocado en un momento determinado durante la manipulación del atributo donde ha sido especificado (si-necesario, si-añadido, si-eliminado).
7. **faceta tipo atributo**, especifica si se trata de un atributo heredable o no heredable.
8. **faceta herencia**, especifica el tipo de herencia del atributo.

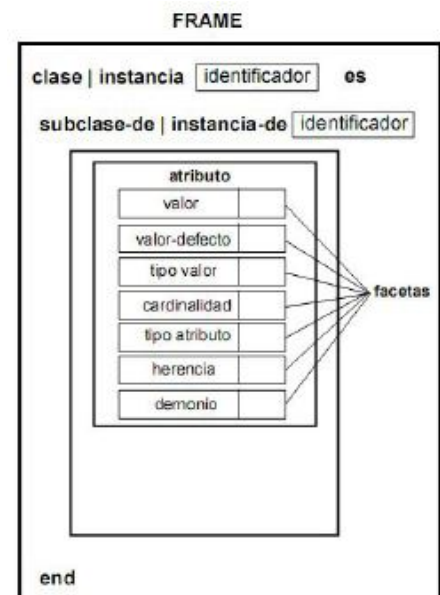


Figura 12: Extensión de frame

(extensión del lenguajes de frames)

<espec-super>::=	<nombre-clase> nil
<atributos>::=	<atributo-faceta-par> [{; <atributo-faceta-par>}* <vacío>
<atributo-faceta-par>::=	<nombre-atributo>=(<espec-faceta> {, <-espec-faceta>}*)
<espec-faceta>::=	<faceta> demonio <tipo-demonio> <llamada-demonio>
<faceta>::=	valor <valor> defecto <valor> tipo-valor : <tipo> max-card entero card <tipo-card> herencia <tipo-herencia> tipo-atributo <tipo-atributo>
<tipo-demonio>::=	si-necesario si-añadido si-eliminado
<tipo>::=	entero real string <conjunto> <nombre-clase> <rango>
<valor>::=	<constante> <nombre-instancia>
<vacío>::=	
<tipo-atributo>::=	heredable no_heredable
<tipo-herencia>::=	Z N
<tipo-card>::=	multi uni
<conjunto>::=	{ <valor> {, <valor>}* }
<rango>::=	[<constante> .. <constante>]

Figura 13: Sintaxis de frames

1. **Tipos de herencia de valores:** Dependiendo de cómo se recorre la taxonomía para determinar los valores del atributo considerado. En la N-herencia va antes la herencia que el procedimiento para calcular los valores del atributo y en la Z-herencia va antes el procedimiento para calcular los valores que la herencia:



Figura 14: Tipos de herencia

2. **Herencia de tipo de atributos:** Una especialización hereda el tipo de valor de su generalización a no ser que se especifique lo contrario. Excepciones: en la especificación de un frame se puede restringir el tipo de valor de un atributo a un subtipo del tipo de valores de su generalización.

4.6. Herencia múltiple

Taxonomía representada por un grafo dirigido acíclico. Una especificación hereda los atributos de todas sus generalizaciones (su conjunto de atributos será la unión de los conjuntos de atributos de sus superclases).

Excepciones:

1. Debidas a inconsistencias entre generalizaciones y especializaciones se resuelven mediante herencia simple.
2. Debidas a inconsistencias entre superclases de una misma especialización, se necesitan métodos para decidir qué valores de facetas heredar de entre los de las superclases.
3. Explícitas.

4.7. Ejemplo Frames

Representar mediante un método basado en frames, detallando: clases, subclases e instancias, slots o atributos de los frames (distinguir entre miembros y propios), clase de valores de los atributos y valores de los atributos, para aquellos que sean conocidos.

Categoría \rightarrow **Rango de presión media (mmHG)**

Arterias grandes \rightarrow 90-100

Arterias pequeñas \rightarrow 80-90

Arteriolas \rightarrow 40-80

Venas \rightarrow <10

Vena pequeña \rightarrow <10

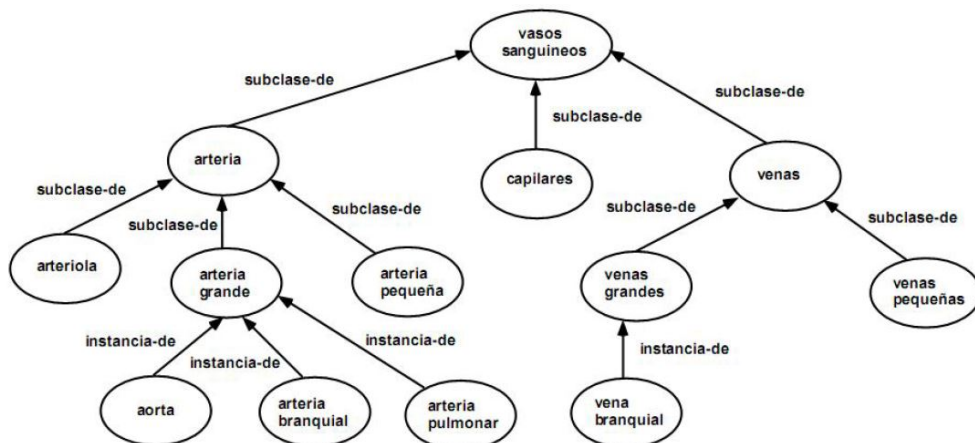


Figura 15: Ejemplo

<pre> clase vasos-sanguíneos es subclase-de nil; pared = (tipo-valor {muscular, fibroso}, tipo-atributo heredable); oxígeno = (tipo-valor {rico, pobre}, tipo-atributo heredable); presión-media = (tipo-valor real, tipo-atributo heredable, demonio si-necesario media(presión-máxima, presión-mínima)); presión-mínima = (tipo-valor real, tipo-atributo heredable); presión-máxima = (tipo-valor real, tipo-atributo heredable) end end clase arterias es subclase-de vasos-sanguíneos; pared = (valor muscular); end end </pre> <p style="text-align: center;">(a)</p>	<pre> clase capilares es subclase-de vasos-sanguíneos; presión-media = (tipo-valor [0..10]) end clase arterias-grandes es subclase-de arterias; presión-media = (tipo-valor [90..100]) end clase arterias-pequeñas es subclase-de arterias; oxígeno = (valor rico); presión-media = (tipo-valor [80..90]) end end </pre> <p style="text-align: center;">(b)</p>
--	---

Figura 16: Ejemplo parte 1

```

clase arteriolas es
  subclase-de arterias;
    oxígeno = (valor rico);
    presión-media = (tipo-valor [40..80])
  end

clase venas es
  subclase-de vasos-sanguíneos;
    pared = (valor fibroso);
    oxígeno = (valor pobre);
    presión-media = (tipo-valor [0..10])
  end
end

```

Figura 17: Ejemplo parte 2

4.7.1. Ejemplo: organización de un congreso

Representa mediante una estructura de Frames la siguiente información acerca de la organización de un congreso:

1. En dicho congreso se debe poder almacenar información acerca de las presentaciones que se van a realizar que serán bien artículos aceptados, conferencias invitadas o posters. De cada una de estas presentaciones se desea conocer su títulos, número de referencia, autor/es, su lista de descriptores y si está confirmada su presentación en el congreso.
2. Se desea también almacenar información de los diferentes autores con datos como nombre, apellidos, universidad o centro donde trabajan y número de artículos presentados.
3. Por otro lado se debe mantener una lista de las personas inscritas, indicando su nombre, cantidad abonada, número de tarjeta de crédito y si es estudiante o no. En el caso de ser estudiante se deberá guardar información acerca de la universidad donde está estudiando.

4. Se quiere disponer de una estructura que refleje las sesiones del congreso por días. El congreso dura 3 días (miércoles, jueves y viernes) y hay 3 sesiones diarias (mañana1, mañana2 y tarde1), donde en cada sesión puede haber o bien 3 artículos o 1 conferencia invitada o un número indeterminado de posters (no puede haber mezclas de presentaciones diferentes).
5. Cada uno de los descriptores del congreso debe asociarse a una descripción del mismo que explique el significado del descriptor.

5. Notas finales

En este último ejemplo vimos que lo que le falta a los frames para ser una ontología es poder referenciar en un atributo de un frame a otro frame.

Además, hay una cosa que se puede hacer con frames y no con redes semánticas: las restricciones que ponemos sobre los atributos (aunque en realidad sí se puede, pero la red semántica aumenta demasiado en tamaño y queda antinatural). Por esto es por lo que se siguen usando frames.