Representación del Conocimiento

Ontologías

Dr. Alejandro Guerra-Hernández

Instituto de Investigaciones en Inteligencia Artificial
Universidad Veracruzana

Campus Sur, Calle Paseo Lote II, Sección Segunda No 112,
Nuevo Xalapa, Xalapa, Ver., México 91097

mailto:aguerra@uv.mx
https://www.uv.mx/personal/aguerra/rc

Maestría en Inteligencia Artificial 2024

Universidad Veracruzana

Razonando sobre objetos I

- Los objetos se acomodan naturalmente en categorías.
- Ej. Mi perro es una mascota, soy un profesor, etc.
- Pero comúnmente los objetos se conciben como miembros de múltiples categorías.
- Ej. Soy profesor, empleado y padre.
- Una categoría puede ser más general o más específica que otras.
- Ej. Pastor catalán es un tipo de perro; reumatólogo es un tipo de doctor; etc.

Universidad Veracruzana

Razonando sobre objetos II

- La generalización es común también entre categorías con descripciones más complejas.
- Ej. Un empleado de tiempo parcial es un empleado; una familia francesa con tres hijos es una familia; etc.
- Los objetos tienen partes, algunas veces muchas.
- Ej. Los libros tienen títulos; las mesas tienen patas; los automóviles tienen llantas; etc.
- Las relaciones entre las partes de un objeto son esenciales al considerarlo como parte de una categoría.

Universidad Veracruzana

Expresividad

- Hasta ahora nos hemos concentrado en los enunciados declarativos para expresar lo que sabemos.
- Ahora nos concentraremos en las frases nominales, i.e., grupos de palabras cuyo núcleo está constituido por un sustantivo y sus modificadores directos o indirectos.
- Ej. La casa amarilla.

Universidad Veracruzana

Frases nominales

- Hemos representado categorías de objetos como predicados de aridad uno, usando sustantivos comunes como $casa(X)$, $amarilla(X)$, etc.
- Pero existen otras frases nominales aparte de los sustantivos.
- Ej. El hombre cuyos hijos son todas mujeres.
- Para expresarlas necesitamos predicados con estructura interna.
- Ej. Es de esperar que si $hijo(X, Y)$ y $padre\_de\_puras\_niñas(X)$ son el caso, entonces Y tiene que ser una niña, por definición.
- Se trata de un predicado compuesto.

Universidad Veracruzana

Lógica Descriptiva

- La lógica de primer orden no provee ninguna herramienta para contender con predicados compuestos de este estilo.
- De cierta manera, las únicas frases nominales en la lógica de primer orden son los sustantivos.
- Como una lógica que nos permitiría manipular predicados complejos estaría trabajando principalmente con descripciones, llamamos al sistema lógico resultante de estas ideas una lógica descriptiva ($\mathcal{DL}$) [2].

Universidad Veracruzana

Conceptos y Roles

- Las descripciones involucran dos clases de sustantivos:

  Categóricos. Describen clases básicas de objetos, p. ej. Adolescente, Niña, etc.

  Relacionales. Describen objetos que son parte, atributo o propiedad de otros objetos, p. ej., Hijo_de, Edad, etc.

- En $\mathcal{DL}$ a los sustantivos categóricos se les suele llamar conceptos, mientras que a los relacionales se les llama roles.

Universidad Veracruzana

Generalización

- Los conceptos están organizados en una jerarquía de generalizaciones.
- Ej. *Profesor&Investigador* es una especialización de *Profesor*.
- En $\mathcal{DL}$ mucha de la información jerárquica se sigue lógicamente del significado de los conceptos compuestos involucrados.
- Mucho del razonamiento llevado a cabo en $\mathcal{DL}$ tiene que ver con computar automáticamente la relación de generalización.

Universidad Veracruzana

Constantes

- Aunque mucho del razonamiento el $\mathcal{DL}$ tiene que ver con categorías, es deseable que nuestras descripciones apliquen también a individuos.
- De modo que $\mathcal{DL}$ incluirá también constantes, p. ej., *alexGuerra*

Universidad Veracruzana

Símbolos lógicos

- Son aquellos que tienen un significado fijo, independiente de la aplicación que tengamos en mente.
- Hay cuatro clases de símbolos lógicos en $\mathcal{DL}$:

Puntuación.	[,], (,).
Enteros positivos.	1,2,3,. . .
Formación de conceptos.	ALL, EXISTS, FILLS, AND
Conectivos.	$\sqsubseteq$, $\doteq$, $\rightarrow$

Universidad Veracruzana

Símbolos no lógicos

- Son aquellos cuyo significado depende de la aplicación en turno.
- Hay tres clases de símbolos no lógicos en $\mathcal{DL}$:

Conceptos atómicos. Se escriben con mayúsculas iniciales, p. ej., `Persona`, `VinoBlanco`. $\mathcal{DL}$ incluye el concepto atómico especial `Thing`.

Roles. Se escriben como los conceptos, pero precedidos por dos puntos, p. ej., `:Hijo`, `:Edad`, etc.

Constantes. Se escriben con minúscula inicial como `alex`.

Universidad Veracruzana

Fórmulas bien formadas (fbf)

- Hay cuatro tipos de fbfs en $\mathcal{DL}$:
  - Constantes.
  - Roles.
  - Conceptos.
  - Enunciados.
- Usaremos las meta-variables c y r para denotar constantes y roles; d y e para conceptos y enunciados; y a para conceptos atómicos.
- Estas meta-variables son análogas a las que denotábamos con letras griegas en las sesiones anteriores, i.e., pueden tomar como valor cualquier fbf, pero ellas mismas no son fbfs.

Universidad Veracruzana

Conceptos

- El conjunto de conceptos en $\mathcal{DL}$ es el conjunto mínimo que satisface:
  - Todo concepto atómico es un concepto;
  - Si r es un rol y d es un concepto, entonces [ALL r d] es un concepto;
  - Si r es un rol y n es un entero positivo, entonces [EXISTS n r] es un concepto;
  - Si r es un rol y c es una constante, entonces [FILLS r c] es un concepto;
  - Si $d_1, \ldots, d_n$ son conceptos, entonces [AND $d_1, \ldots, d_n$] es un concepto.

Universidad Veracruzana

Enunciados

- Existen tres tipos de enunciados en $\mathcal{DL}$:
  - Si d_1 y d_2 son conceptos, entonces $(d_1 \sqsubseteq d_2)$ es un enunciado;
  - Si d_1 y d_2 son conceptos, entonces $(d_1 \doteq d_2)$ es un enunciado;
  - Si c es una constante y d es un concepto, entonces $(c \rightarrow d)$ es un enunciado.

Universidad Veracruzana

Significado de las fbfs

- Las constantes denotan objetos del universo de discurso, como en la lógica de primer orden.
- Los conceptos denotan categorías o clases de individuos.
- Los roles denotan relaciones binarias sobre individuos.
- El significado de los conceptos complejos se deriva de sus partes.

Universidad Veracruzana

Ejemplos de Conceptos Atómicos

- [EXISTS 1 :*Hijos*] denota a alguien que tiene hijos.
- [FILLS :*Primo juan*] denota a alguien cuyo primo es Juan.
- [ALL :*Empleado Sindicalizado*] describe algo cuyos empleados, si tiene, son todos sindicalistas.

Universidad Veracruzana

Ejemplos de Conceptos Complejos

- [AND [FILLS :*Color rojo*] [EXISTS 2 :*TipoUva*]] denota la categoría de mezcla de tintos (un vino cuyo color es rojo y está hecho con al menos dos tipos de uva).
- O una definición basada en un concepto complejo:

$$
\begin{aligned}
(\mathit{EmpresaProgre} \doteq [\text{AND } \mathit{Empresa} \\
&\quad [\text{EXISTS } 7 : \mathit{Director}] \\
&\quad [\text{ALL } : \mathit{Admin} \ [\text{AND } \mathit{Mujer} \\
&\qquad\qquad [\text{FILLS } : \mathit{Grado}\ \mathit{phD}]]] \\
&\quad [\text{FILLS } : \mathit{SalarioMin}\ \$24{,}00/\mathit{hr}]])
\end{aligned}
$$

Universidad Veracruzana

Definiciones completas

- Si el último ejemplo se incluye en una base de conocimientos, decimos que `EmpresaProgre` está completamente definido.
- Establecimos las condiciones suficientes y necesarias para ser una empresa progresista en el lado derecho de la definición.
- Si usamos $\sqsubseteq$ en lugar de $\doteq$ solo definimos las condiciones necesarias, por lo que no tendríamos manera de saber reconocer a un individuo como miembro de esta categoría; aunque si podríamos razonar sobre un individuo que ha sido declarado como tal.

Universidad Veracruzana

Ejemplos de enunciados

- $(\mathit{Cirujano} \sqsubseteq \mathit{Doctor})$ expresa que todo cirujano es, entre otras cosas, también un doctor.
- $(\mathit{nicandro} \rightarrow \mathit{SNII})$ expresa que el Dr. Nicandro es un SNII.

Universidad Veracruzana

Interpretaciones

- Una interpretación $\mathcal{I}$ para $\mathcal{DL}$ es un par $\langle D, I \rangle$ donde D es el universo de discurso, i.e., el conjunto de objetos sobre los que queremos expresarnos; y I es un mapeo de los símbolos no lógicos en $\mathcal{DL}$ a elementos y relaciones en D, donde:
  - Para toda constante c, $\mathcal{I}[c] \in D$;
  - Para todo concepto atómico a, $\mathcal{I}[a] \subseteq D$;
  - Para todo rol r, $\mathcal{I}[r] \subseteq D \times D$.
- El conjunto $\mathcal{I}[d]$, asociado al concepto d, es una interpretación llamada su extensión.

Universidad Veracruzana

Conceptos no atómicos

- Los conceptos no atómicos son interpretados en función de sus componentes:
  - $\mathcal{I}[Thing] = D$;
  - $\mathcal{I}[[\text{ALL } r\ d]] = \{x \in D \mid \forall y, \langle x, y\rangle \in \mathcal{I}[r] \implies y \in \mathcal{I}[d]\}$;
  - $\mathcal{I}[[\text{EXISTS } n\ r]] = \{x \in D \mid$ hay al menos n distintos y t.q. $\langle x, y\rangle \in \mathcal{I}[r]\}$;
  - $\mathcal{I}[[\text{FILLS } r\ c]] = \{x \in D \mid \langle x, \mathcal{I}[c]\rangle \in \mathcal{I}[r]\}$;
  - $\mathcal{I}[[\text{AND } d_1, \ldots, d_n]] = \mathcal{I}[d_1] \cap \cdots \cap \mathcal{I}[d_n]$.
- Estas reglas nos permiten computar la extensión de cualquier concepto.

Universidad Veracruzana

Verdad en una interpretación

- Ahora podemos especificar que enunciados $\mathcal{DL}$ son verdaderos de acuerdo a una interpretación dada:
  - $\mathcal{I} \models (c \rightarrow d)$ ssi $\mathcal{I}[c] \in \mathcal{I}[d]$;
  - $\mathcal{I} \models (d \sqsubseteq d')$ ssi $\mathcal{I}[d] \subseteq \mathcal{I}[d']$;
  - $\mathcal{I} \models (d \doteq d')$ ssi $\mathcal{I}[d] = \mathcal{I}[d']$.
- $\mathcal{I} \models S$ denota que los enunciados en el conjunto S son verdaderos en la interpretación $\mathcal{I}$.

Universidad Veracruzana

Consecuencia Lógica

- Sea S un conjunto de enunciados y α un enunciado, $S \models \alpha$ denota que α es consecuencia lógica de S, i.e., para toda interpretación $\mathcal{I}$, siempre que $\mathcal{I}[S]$ es el caso, también lo es que $\mathcal{I}[\alpha]$.
- Decimos que el enunciado α es válido ssi es consecuencia lógica del conjunto vacío, $\models \alpha$.

Universidad Veracruzana

Razonamientos

- Hay dos tipos básicos de razonamientos cuando usamos $\mathcal{DL}$:
  - Determinar si es el caso, o no, que alguna constante c satisface un concepto dado d: $\Delta \models (c \rightarrow d)$;
  - Determinar si el el caso, o no, que un concepto dado d subsume a otro concepto d': $\Delta \models (d \sqsubseteq d')$.
- El razonamiento se reduce a computar estas consecuencias lógicas.
- Comenzaremos por el segundo caso, que es necesario para resolver el primero.

Universidad Veracruzana

Observaciones I

- Algunas fbf son validas por su estructura, al igual que en FOL:

$$([\text{AND } \mathit{Doctor}\ \mathit{Mujer}] \sqsubseteq \mathit{Doctor})$$

$$(\mathit{juan} \rightarrow \mathit{Thing})$$

- Normalmente la consecuencia lógica depende de Δ.
- Ej. Si $(\mathit{Cirujano} \sqsubseteq \mathit{Doctor}) \in \Delta$ entonces:

$$\Delta \models ([\text{AND } \mathit{Cirujano}\ \mathit{Mujer}] \sqsubseteq \mathit{Doctor})$$

Universidad Veracruzana

Observaciones II

- Lo mismo se sigue si Δ incluye:

$$(Cirujano \doteq [\text{AND } Doctor\ [\text{FILLS } :Espec\ cirugia]])$$

- Al igual que en el caso de la resolución para FOL, esta computación será correcta (robusta y completa), dada la definición de consecuencia lógica

Universidad Veracruzana

Simplificación de la Base de Conocimientos

- La subsumción no involucra enunciados del tipo $(c \rightarrow d)$: Si Δ' es como Δ, excepto que las fbfs de la forma $(c \rightarrow d)$ han sido removidas, $\Delta \models (d \sqsubseteq e)$ ssi $\Delta' \models (d \sqsubseteq e)$.
- Para computar $\sqsubseteq$ asumimos que Δ no contiene este tipo de fbfs.
- Podemos remplazar las fbfs de la forma $(d \sqsubseteq e)$ por $(d \doteq [\text{AND } e\ a])$, donde a es un nuevo concepto atómico que no se usa en ningún otro sitio.
- Restricciones:
  - El operador izquierdo de $\doteq$ debe ser un concepto atómico diferente de *Thing*.
  - Tal operador debe tener una ocurrencia única en Δ.
  - Asumimos que las fbfs que involucran $\doteq$ son acíclicas.

Universidad Veracruzana

Cómputo de $\sqsubseteq$

- Asumiendo las simplificaciones precedentes, para computar $\Delta \models (d \sqsubseteq e)$ debemos:
  1. Usando las declaraciones de definiciones ($\doteq$), poner d y e es una forma especial normalizada, similar es espíritu a la CNF;
  2. Determinar si cada parte de la e normalizada se toma en cuenta en alguna parte normalizada de d.
- De forma que subsumir se reduce a una relación estructural entre dos conceptos normalizados.

Universidad Veracruzana

Normalización I

1. Expandir definiciones. Todo concepto atómico que aparezca a la izquierda de $\doteq$ es remplazado por su definición donde aparezca.
2. Aplanar operadores AND. Todo concepto de la forma

$$[\text{AND} \ldots [\text{AND } d_1 \ldots d_n] \ldots]$$

se simplifica a $[\text{AND} \ldots d_1 \ldots d_n \ldots]$.

3. Combinar los operadores ALL. Todo sub-concepto de la forma

$$[\text{AND} \ldots [\text{ALL } r\ d_1] \ldots [\text{ALL } r\ d_2] \ldots]$$

se simplifica a $[\text{AND} \ldots [\text{ALL } r\ [\text{AND } d_1\ d_2]]]$.

Universidad Veracruzana

Normalización II

4. Combinar los operadores EXISTS. Todos los sub-conceptos de la forma

$$[\text{AND} \ldots [\text{EXISTS } n_1 \ r] \ldots [\text{EXISTS } n_2 \ r] \ldots]$$

se puede simplificar a [AND ... [EXISTS n r]] donde n es el máximo de n_1 y n_2.

5. Contender con *Thing*. Algunos conceptos deben removerse como argumentos de AND: *Thing*, [ALL r *Thing*] y [AND]. *Thing* solo debe aparecer si la expresión se simplifica a ello.

6. Remover expresiones redundantes. Eliminar toda expresión que es un duplicado de otra dentro de la misma expresión AND.

Universidad Veracruzana

Observaciones I

- Estas reglas se aplican repetidamente, en cualquier orden y a cualquier nivel de anidamiento para normalizar un concepto.
- El proceso termina cuando ninguna regla puede aplicarse.

Universidad Veracruzana

Observaciones II

- El resultado de la normalización es *Thing*, un concepto atómico o un concepto de la siguiente forma:

$$
\begin{aligned}
&[\text{AND } a_1 \ldots a_m \\
&\quad [\text{FILLS } r_1\ c_1] \ldots [\text{FILLS } r_{m'}\ c_{m'}] \\
&\quad [\text{EXISTS } n_1\ s_1] \ldots [\text{EXISTS } n_{m''}\ s_{m''}] \\
&\quad [\text{ALL } t_1\ e_1] \ldots [\text{ALL } t_{m'''}\ e_{m'''}]]
\end{aligned}
$$

donde a_i con conceptos atómicos primitivos distintos a *Thing*; r_i, s_i y t_i son roles; c_i son constantes; n_i son enteros positivos; y e_i son a su vez conceptos normalizados.

Universidad Veracruzana

Observaciones III

- Los argumentos de AND en un concepto normalizado se conocen como sus componentes.
- De hecho podemos pensar en *Thing* como un AND sin componentes; y en los conceptos atómicos con AND con un solo componente.

Universidad Veracruzana

Ejemplo

- Asumamos que Δ incluye las siguientes definiciones:

$$\textit{EmpresaEquilibrada} \doteq [\text{AND } \textit{Empresa} \; [\text{ALL } :\textit{Admin}[\text{AND } \textit{GEscolarLic} \; [\text{EXISTS } 1 :\textit{GEscolarTecnico}]]]]$$

$$\textit{EmpresaTech} \doteq [\text{AND } \textit{Empresa} \; [\text{FILLS } :\textit{Mercado } \textit{nasdaq}] \; [\text{ALL } :\textit{AdminTechie}]]$$

$$\textit{Techie} \doteq [\text{EXISTS } 2{:}\textit{GEscolarTecnico}]$$

- ¿Cómo normalizar el siguiente concepto?

$$[\text{AND } \textit{EmpresaEquilibrada } \textit{EmpresaTech}]$$

Universidad Veracruzana

Expansión de Conceptos

[AND [AND *Empresa*
[ALL :*Admin* [AND *GEscolarLic*
[EXISTS 1 :*GEscolarTecnico*]]]]
[AND *Empresa*
[FILLS :*Mercado nasdaq*]
[ALL :*Admin* [EXISTS 2:*GEscolarTecnico*]]]]

Universidad Veracruzana

Aplanado de Conjunciones

[AND *Empresa*
[ALL :*Admin* [AND *GEscolarLic*
[EXISTS 1 :*GEscolarTecnico*]
[EXISTS 2 :*GEscolarTecnico*]]]
Empresa
[FILLS :*Mercado nasdaq*]]

Universidad Veracruzana

Removiendo Redundancia

[AND *Empresa*
[ALL :*Admin* [AND *GEscolarLic* [EXISTS 2 :*GEscolarTecnico*]]]
[FILLS :*Mercado nasdaq*]]

Universidad Veracruzana

Correspondencia estructural

- Para computar $\Delta \models (d \sqsubseteq e)$ necesitamos comparar las versiones normalizadas de d y e.
- La forma normalizada de d debe satisfacer de alguna forma todos los componentes de la forma normalizada de e.
- Ej. Si e contiene el componente [ALL r e'], entonces d debe contener algo como [ALL r d'] tal que $d' \sqsubseteq e'$.

Universidad Veracruzana

Algoritmo

- Sean d y e dos conceptos normalizados de la forma [AND $d_1 \ldots d_m$] y [AND $e_1 \ldots e_{m'}$] respectivamente.
- Regresar *si* ssi para cada componente $e_j, 1 \leq j \leq m'$ existe un componente $d_i, 1 \leq i \leq m''$ t.q. d_i se corresponde con e_j como sigue:
  1. Si e_j es un concepto atómico, entonces d_i debe ser idéntico a e_j;
  2. Si e_j es de la forma [FILLS r c], entonces d_i debe ser idéntico a éste;
  3. Si e_j es de la forma [EXISTS n r], entonces el d_i correspondiente debe ser de la forma [EXISTS n' r], para algún $n' \geq n$; en el caso de $n = 1$, d_i puede ser de la forma [FILLS r c], para cualquier constante c.
  4. Si e_j es de la forma [ALL r e'], entonces el d_i correspondiente debe ser de la forma [ALL r d'], donde recursivamente d' es subsumido por e'.
- En cualquier otro caso, regresar *no*.

Universidad Veracruzana

Ejemplo

- Consideren el siguiente concepto normalizado:

$$[\text{AND } \textit{Empresa} \\ [\text{ALL } :\textit{Admin } \textit{GEscolarLic}] \\ [\text{EXISTS } 1 :\textit{Mercado}]]$$

- Este concepto (d) subsume al del ejemplo anterior (d').

Universidad Veracruzana

Observaciones

- *Empresa* es un concepto atómico que aparece como componente de d';
- Para los componentes ALL de d, cuya restricción es *GEscolarLic*, hay componentes ALL en d' t.q. la restricción en todos esos componentes es subsumida por *GEscolarLic*, i.e., la conjunción [AND *GEscolarLic* [EXISTS 2 :*GEscolarTecnico*]].
- Para el componente [EXISTS 1 :*Mercado*] de d, existe el componente correspondiente FILLS en d'.

Universidad Veracruzana

Correctez I

- Si Δ está simplificada, todos los conceptos en ella se pueden normalizar.
- Cada paso de la normalización preserva la equivalencia entre conceptos.
- Una vez normalizada, $\Delta \models (d \sqsubseteq e)$ ssi $(d' \sqsubseteq e')$ es válida.
- *si*: Supongamos que todo componente de e' tiene un componente correspondiente en d'. Imaginemos una interpretación $\mathcal{I} = \langle D, I \rangle$ y una $x \in D$, t.q., $x \in \mathcal{I}[d']$ y por lo tanto en todos sus componentes, d_i.
- Para probar que $x \in \mathcal{I}[e']$ observamos que $\forall j, x \in \mathcal{I}[e_j]$ porque i tiene un componente correspondiente en $d_i \in d'$ y $x \in \mathcal{I}[d_i]$.

Universidad Veracruzana

Correctez II

- *no*: Debemos construir una interpretación donde alguna $x \in D$ cumple con $x \in \mathcal{I}[d']$ pero $x \notin \mathcal{I}[e']$.
- El caso más simple es cuando no hay EXISTS involucrados:
  1. Sea $\mathcal{D}$ el conjunto de todas las constantes en Δ junto con el conjunto de cadenas de roles, definido como todas las secuencias de roles (incluida la secuencia vacía).
  2. Para toda constante c, $\mathcal{I}[c] = c$; y para todo concepto atómico a sea $\mathcal{I}[a]$ todas las constantes y todas las cadenas de roles σ donde $\sigma = r_1 \ldots r_k$ para algún $k \geq 0$ t.q., d' es de la forma:

$$[\text{AND} \ldots [\text{ALL } r_1 \ldots [\text{AND} \ldots [\text{ALL } r_k \ a] \ldots] \ldots] \ldots]$$

Universidad Veracruzana

Correctez III

3. Para todo rol r, sea $\mathcal{I}[r]$ todo par de constantes, junto con todo par $(\sigma, \sigma \cdot r)$ donde σ es una cadena de roles, junto con el par (σ, c) donde c es una constante, $\sigma = r_1 \ldots r_k$ para algún $k \geq 0$ tal que d' es de la forma:

$$[\text{AND} \ldots [\text{ALL } r_1 \ldots [\text{AND} \ldots [\text{ALL } r_k \ [\text{FILLS } r \ c]] \ldots] \ldots] \ldots]$$

4. Asumiendo que el procedimiento regrese *no*, se puede probar que para esta interpretación la cadena de roles vacía pertenece a la extensión de d' pero no a la de e' y en consecuencia d' no subsume a e'.

Universidad Veracruzana

Satisfacción

- Computar si es el caso que un individuo denotado por una constante satisface un concepto, resulta muy similar a computar si un concepto subsume a otro.
- La principal diferencia es que debemos tomar en cuenta los enunciados con $\rightarrow$ en Δ.
- En general queremos determinar si es el caso que $\Delta \models (b \rightarrow e)$ donde b es una constante y e es un concepto.
- Ej. Si Δ contiene $(b \rightarrow d)$ y $\Delta \models (d \sqsubseteq e)$ es claro que $\Delta \models (b \rightarrow e)$.

Universidad Veracruzana

Ideas

- Lo anterior sugiere que debemos colectar todos los enunciados de la forma $(b \rightarrow d_i)$ en Δ y responder *si* cuando $(e \sqsubseteq [\text{AND } d_1 \ldots d_n])$.
- Pero esto puede perder algunas inferencias necesarias.

Universidad Veracruzana

Ejemplo

- Asumamos que Δ contiene:

$$\begin{aligned} \textit{jose} &\rightarrow \textit{Persona} \\ \textit{empresaCan} &\rightarrow [\text{AND } \textit{Empresa} \\ &\qquad [\text{ALL } :\textit{Director } \textit{Canadiense}] \\ &\qquad [\text{FILLS } :\textit{Director } \textit{jose}]] \end{aligned}$$

- Es fácil ver que $\Delta \models (\textit{jose} \rightarrow \textit{Canadiense})$, aunque el enunciado $\rightarrow$ acerca de *jose* no nos lleve a esa conclusión.

Universidad Veracruzana

Propagación

- Es necesario propagar la información implicada por lo que sabemos acerca de otros individuos (nombrados por constantes, o anónimos que satisfacen roles) antes de subsumir.
- Esto se consigue con una forma de encadenamiento hacía adelante similar al que usamos con las cláusulas de Horn.
- En lo que sigue, asumimos que no hay términos EXISTS en ningún concepto involucrado.

Universidad Veracruzana

Algoritmo

1. Construir una lista S de pares (b, d), donde b es cualquier constante en Δ y d es la versión normalizada del AND de todos los conceptos d', t.q., $(b \rightarrow d') \in \Delta$.
2. Tratar de encontrar dos constantes b_1 y b_2, t.q., (b_1, d_1) y (b_2, d_2) pertenecen a S, y para algún rol r, [FILLS r b_2] y [ALL r e] son ambos componentes de d_1, pero $\Delta \not\models (d_2 \sqsubseteq e)$.
3. Si no se puede encontrar tal par de constantes, salir. En otro caso, remplazar el par $(b_2, d_2) \in S$ por (b_2, d_2'), donde d_2' es la versión normalizada de [AND d_2 e]. Ir al paso 2.

Universidad Veracruzana

Observaciones

- Este procedimiento computa para cada constante b, el concepto más específico t.q., $\Delta \models (b \rightarrow d)$.
- Una vez ejecutado, para verificar si es el caso que $\Delta \models (b \rightarrow e)$ solo debemos verificar si $\Delta \models (d \sqsubseteq e)$.
- El encadenamiento hacía adelante termina en tiempo polinomial en función del tamaño de Δ.
- En el peor caso, para cada constante b, terminaremos con un par (b, d) donde d es el AND de todo componente mencionado en Δ, tras lo cual no hay más propagaciones posibles.

Universidad Veracruzana

Existenciales

- Para contender con términos de la forma [EXISTS 1 r] se puede usar una idea similar.
- En lugar de tener pares $(b, d) \in S$, construiremos pares de la forma $(b \cdot \sigma, d)$, donde σ es una cadena de roles.
- Intuitivamente, $b \cdot r_1 \cdot r_2$ puede entenderse como un individuo que es un r_2 de un r_1 de b, posiblemente anónimo.
- Cuando σ es vacío, esto corresponde con b mismo.

Universidad Veracruzana

Encadenamiento hacía adelante extendido

- Agregamos los siguientes pasos al final del algoritmo original:
  - Tratar de encontrar una constante b y una cadena de roles σ (posiblemente vacía) y un rol r, t.q., $(b \cdot \sigma, d_1) \in S$ y algún $(b \cdot \sigma \cdot r, d_2) \in S$ (o si no existe tal par, tomar *Thing* como d_2), y [EXISTS 1 r] y [ALL r e] son ambos componentes de d_1, pero $\Delta \not\models (d_2 \sqsubseteq e)$.
  - Si el paso anterior tiene éxito, remover los pares $(b \cdot \sigma \cdot r, d_2)$ de S (si aplica) y agregar los pares $(b \cdot \sigma, d_2')$, donde d_2' es la versión normalizada de [AND d_2 e]. Repetir.

Universidad Veracruzana

Observaciones

- El algoritmo anterior extiende el procedimiento de propagación a los individuos anónimos: Iniciamos con alguna propiedad del individuo $b \cdot \sigma$, y concluimos algo nuevo acerca del individuo $b \cdot \sigma \cdot r$.
- Eventualmente esto puede llevarnos a concluir algo nuevo sobre un individuo nombrado por una constante.

Universidad Veracruzana

Ejemplo

- Sea Δ:

$$\begin{aligned} \textit{elena} \rightarrow [\text{AND } [\text{EXISTS 1:}\textit{Hijo}] \\ [\text{ALL :}\textit{Hijo } [\text{AND } [\text{FILLS :}\textit{Pediatra maria}] \\ [\text{ALL :}\textit{Pediatra Escandinavo}]]]] \end{aligned}$$

- Se puede concluir correctamente que ($\textit{maria} \rightarrow \textit{Escandinavo}$).
- La versión original fallaría porque el programa no declara hijos de *elena*.
- El caso más general para $n > 1$ se procesa de la misma forma (el resto de los individuos anónimos tienen exactamente las mismas propiedades en el encadenamiento hacía adelante).

Universidad Veracruzana

Ontologías

- Las ontologías se usan para capturar el conocimiento acerca de un dominio de interés.
- Ej. Vigilancia en la epidemia del COVID-19
- Se describen los conceptos en el dominio, así como las relaciones entre esos conceptos.
- Se ha propuesto muchos lenguajes para definir ontologías, nosotros utilizaremos OWL, basado en las ideas revisadas sobre lógica descriptiva [1]:

`https://www.w3.org/TR/owl2-overview/`

Universidad Veracruzana

Individuos

- Representan objetos del universo de discurso.
- OWL no asume el supuesto de nombre único (UNA), i.e., nombres diferentes pueden denotar diferentes individuos.
- Ej. AMLO, Andrés Manuel y Presidente de México.
- Se debe especificar si los individuos son diferentes de otros, o iguales a otros.
- A los individuos también se les conoce en inglés como Instances.

Universidad Veracruzana

Propiedades

- Son relaciones binarias entre individuos.
- Ej. `tieneDiagnostico` puede ser una relación entre un paciente y su diagnóstico.
- Las propiedades pueden tener inversos.
- Ej. `diagnosticoDe`
- También pueden ser transitivas, simétricas, funcionales, etc.
- En $\mathcal{DL}$ les llamamos roles y en Protégé se conocen como slots.

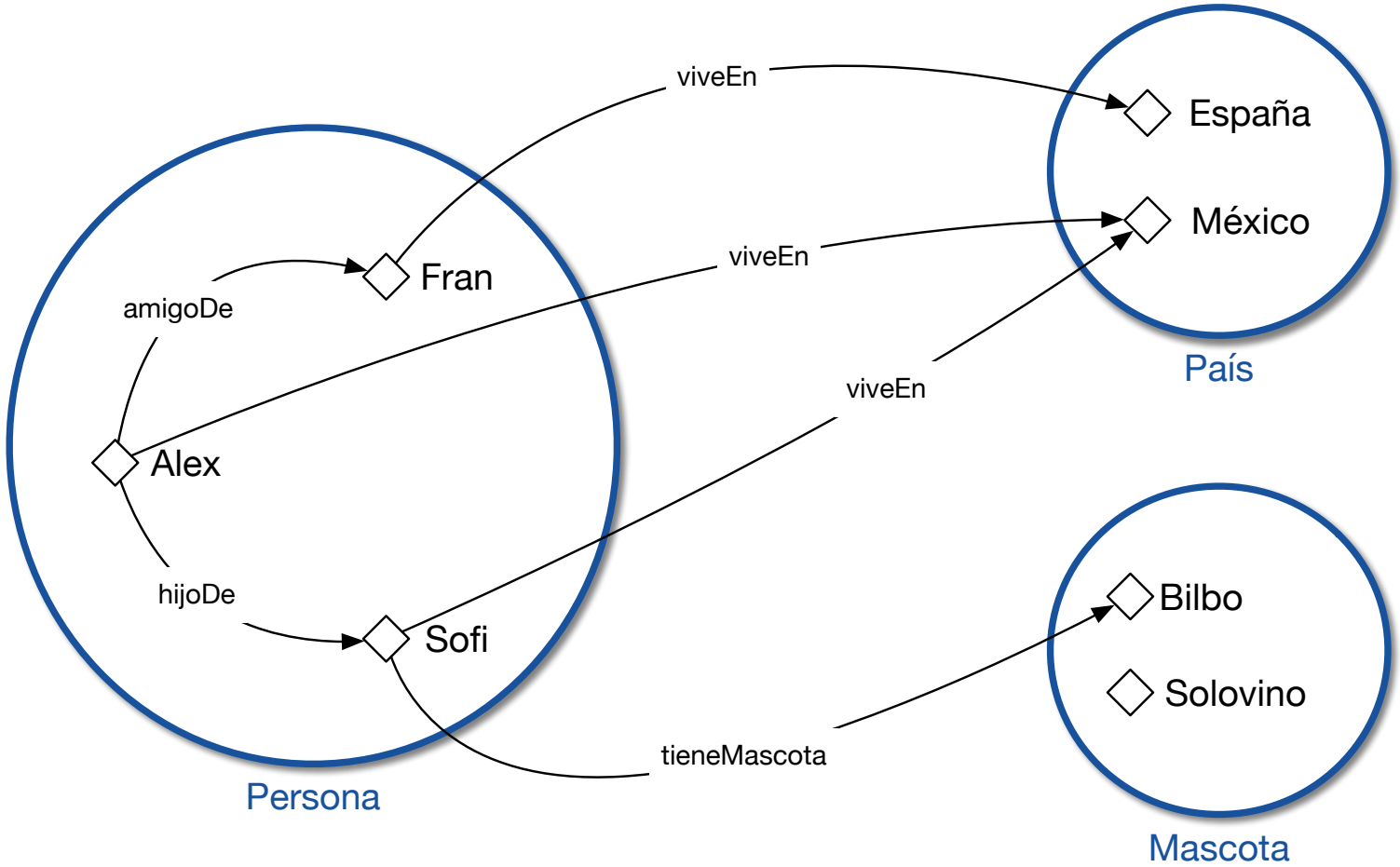
Universidad Veracruzana

Clases

- Las clases son conjuntos de individuos.
- Se describen formalmente para definir las condiciones de membresía.
- Se escriben con mayúscula inicial, p. ej., `Gato` es el conjunto de todos los individuos en el dominio de discurso que son gatos.
- Se organizan jerárquicamente en sub-clases y super-clases.
- El término concepto se usa a veces en lugar de clase.

Universidad Veracruzana

Ejemplo gráfico



Universidad Veracruzana

Sintaxis de Manchester

- `https://www.w3.org/TR/owl2-manchester-syntax/`:

$\mathcal{DL}$	Manchester
Thing	`owl:Thing`
	`owl:Nothing`
Concepto	`Class`
:*Rol*	`Object property`
[AND d_1 d_2]	d_1 `and` d_2
	d_1 `or` d_2
	`not` d
[FILLS r c]	r `some` C
	r `only` C
[EXISTS n r]	r `exactly` n C
	r `min` n C
	r `max` n C

Universidad Veracruzana

Ejemplos

- $\mathcal{DL}$
  1. [AND *Persona* [FILLS: *Sexo masculino*]]
  2. [AND [FILLS: *Color rojo*] [EXISTS 2 : *TipoUva*]]
- Manchester
  1. `Persona and (sexo some Masculino)`
  2. `(color some Rojo) and (min 2 TipoUva Vino)`

Universidad Veracruzana

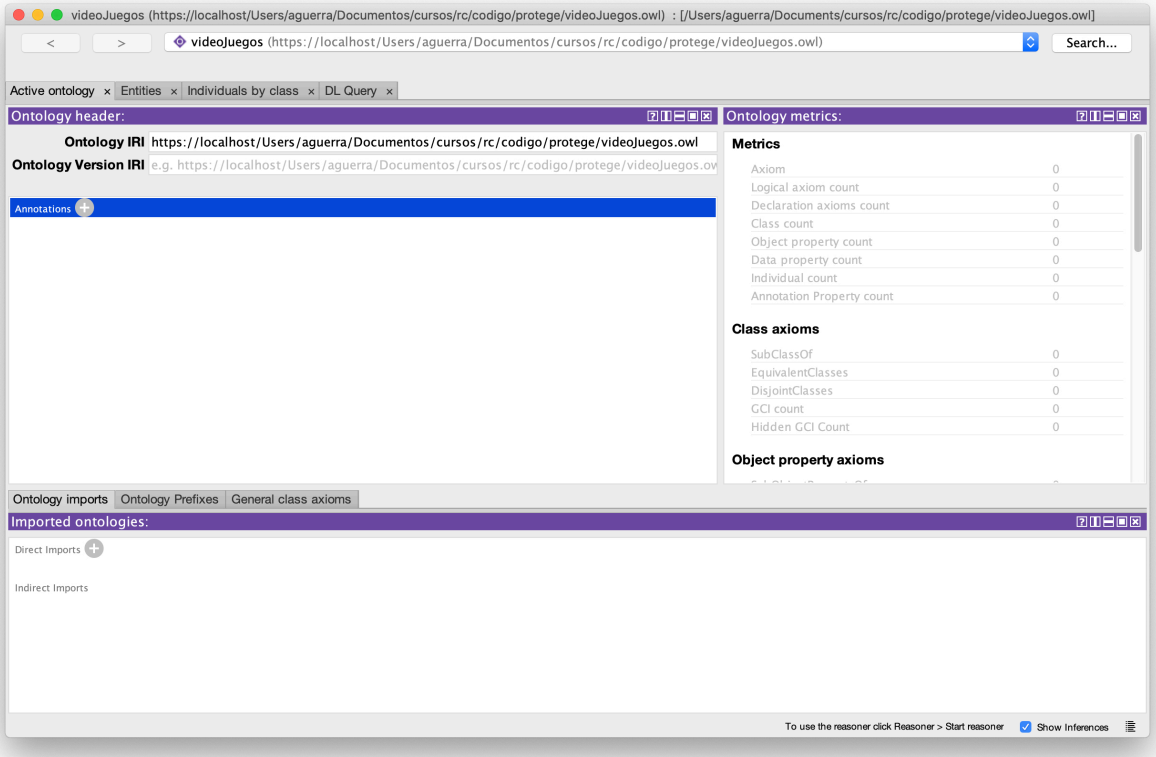
Videojuegos

- Vamos a desarrollar una ontología sobre vídeo juegos.
- Los conceptos involucrados incluyen: juegos, plataformas.
- Las propiedades incluyen genero, dificultad, etc.
- Las relaciones son que los juegos tienen una dificultad, plataforma y género definidos.
- Los conceptos definibles incluyen: juego multi-plataforma, juego difícil, juego de macOS, etc.

Universidad Veracruzana

Creando una nueva ontología I

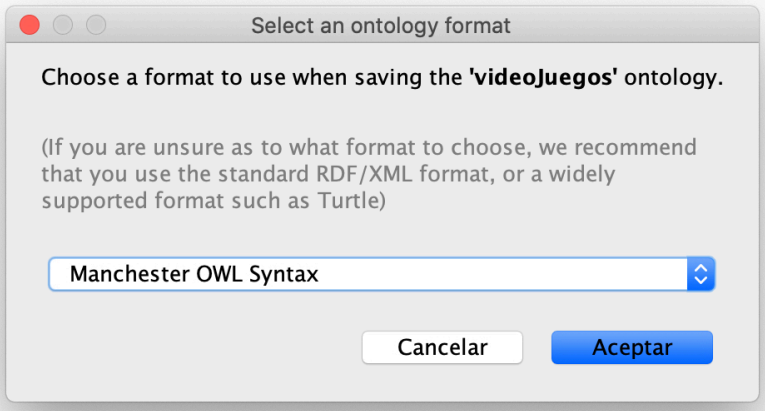
- Ejecute Protégé [3] y asegúrese de estar en la pestaña `Active Ontology`.
- Cambie el IRI de la ontología a algo como:



Universidad Veracruzana

Creando una nueva ontología II

- Guarde estos cambios, Protégé le preguntará en que formato desea guardar la ontología:

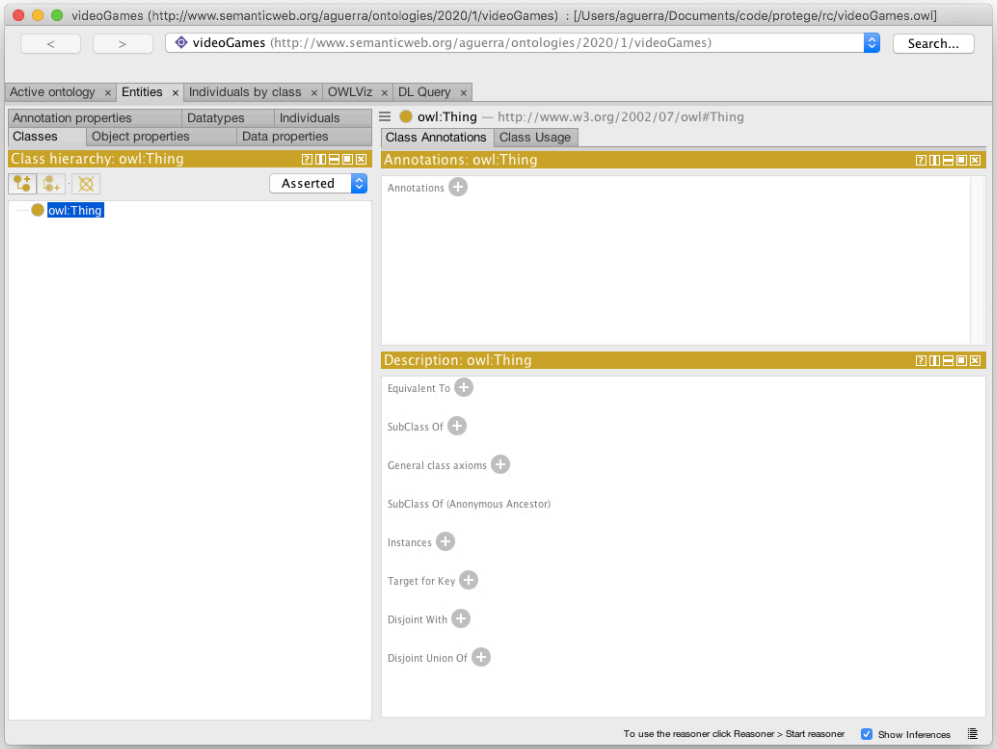


- Debería tener un archivo `videoJuegos.owl` en el directorio seleccionado.

Universidad Veracruzana

Clases I

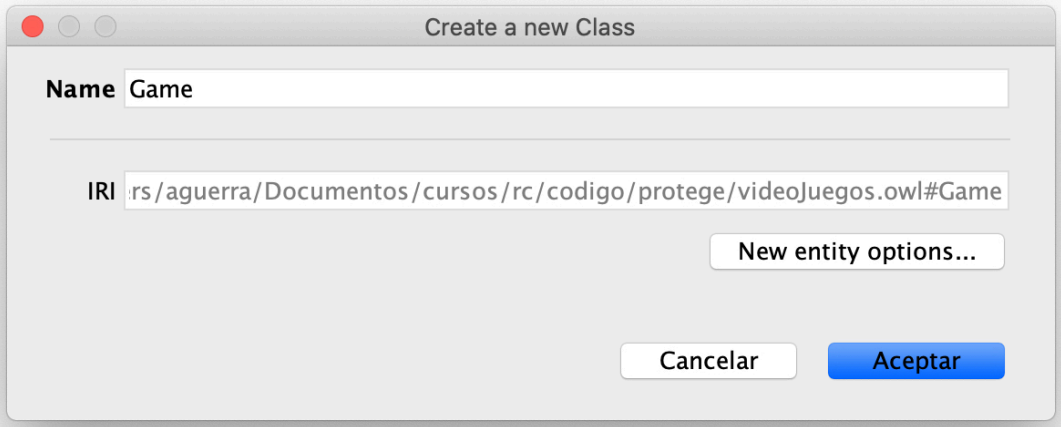
- Para trabajar con las clases hay que ir a la pestaña `Entities`.



- `owl:Thing` es la súper clase por defecto.

Clases II

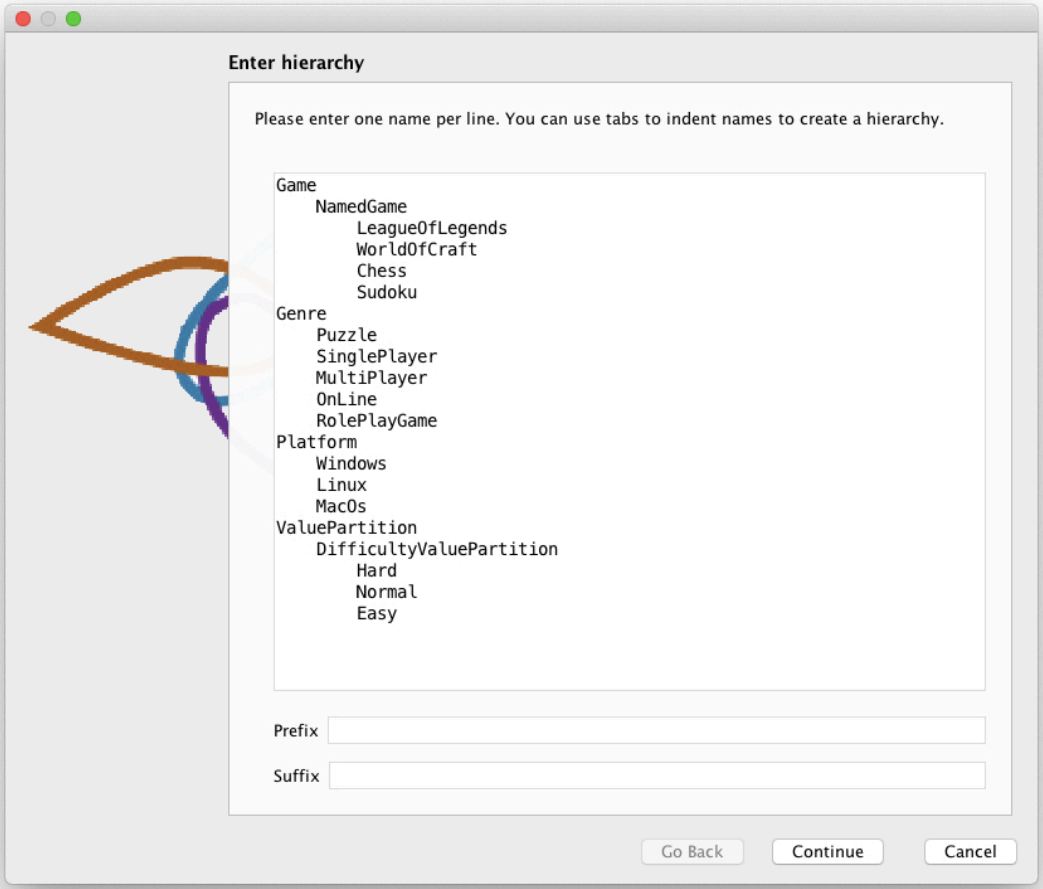
- Se pueden agregar clases individualmente:



- El `IRI` se actualiza automáticamente.

Clases III

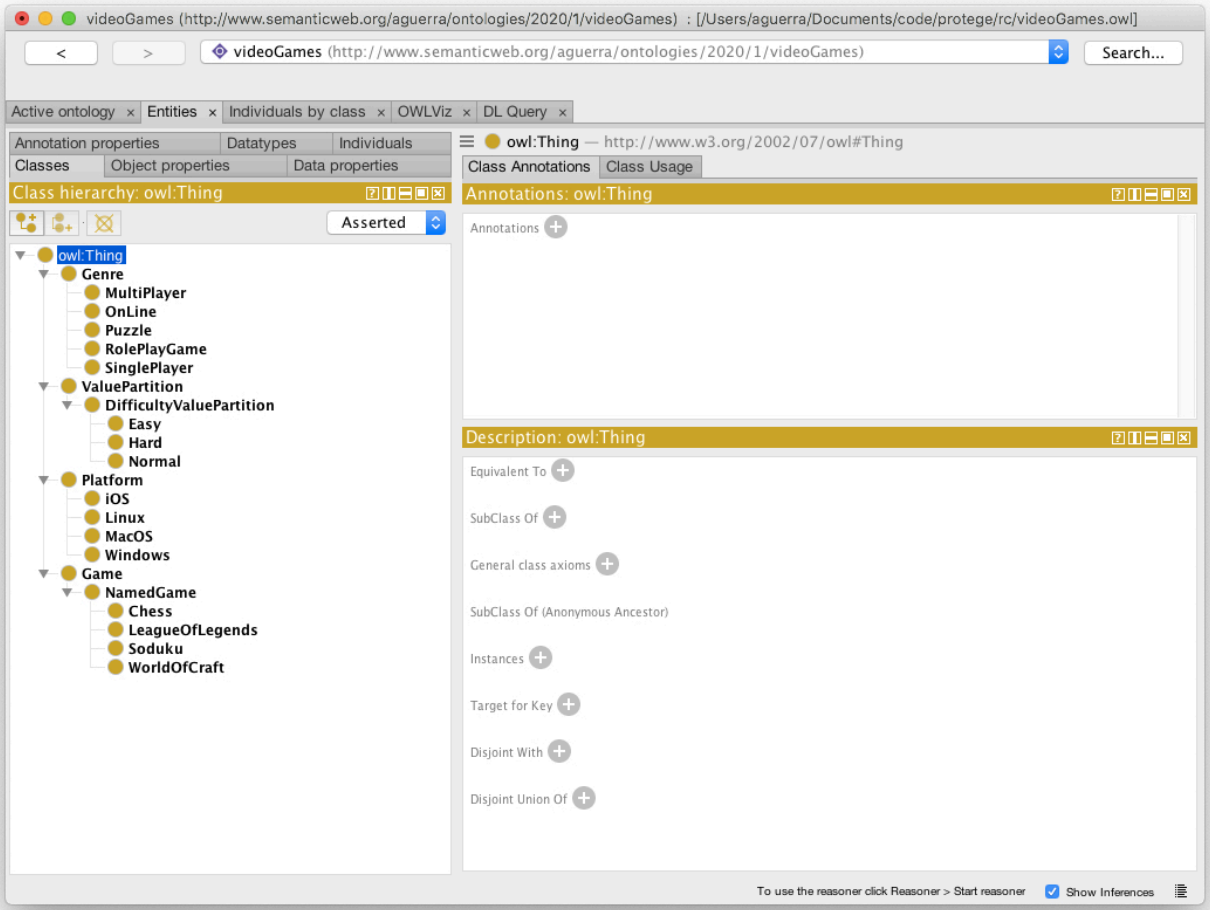
- O se puede usar la herramienta `Create class hierarchy...`:



Universidad Veracruzana

Clases IV

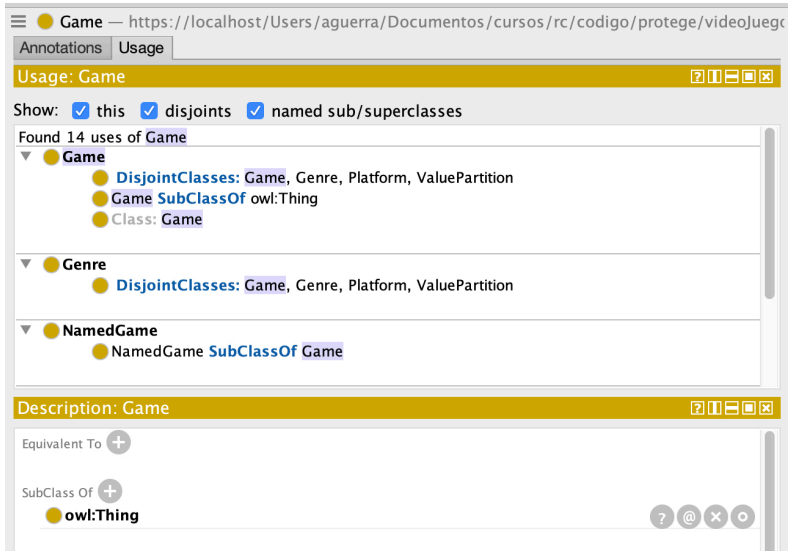
- Que produce la siguiente jerarquía:



Universidad Veracruzana

Axiomas

- Para ver los axiomas que definen las clases, haga clic en `Usage`.
- Agregue a la clase `Game` el axioma de que es sub-clase de `owl:Thing`:



- $Game \sqsubseteq Thing$
- $[\text{AND } Game\ Genre] \sqsubseteq Nothing$

Universidad Veracruzana

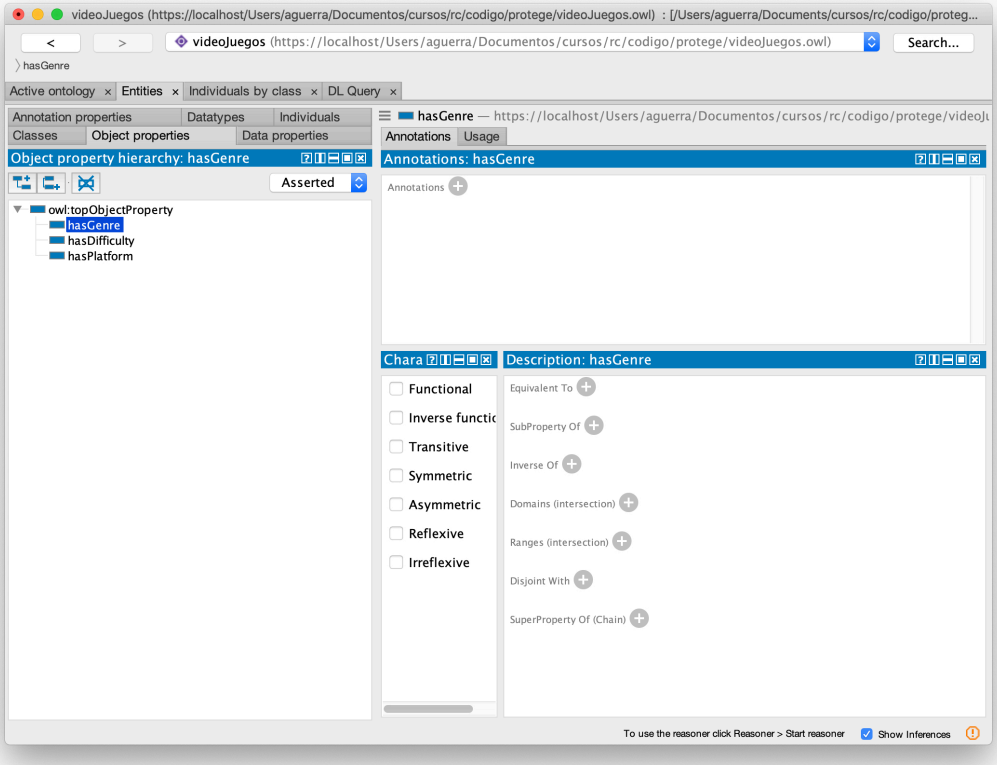
¿Qué sigue?

- ¿Qué tenemos?
  - Todas las clases no definibles en la ontología.
  - Una jerarquía inicial de clases, i.e., una taxonomía.
  - Axiomas disjuntos básicos (entre hermanos).
- ¿Qué nos hace falta?
  - Propiedades de objetos.
  - Relaciones entre clases.
  - Clases definibles.

Universidad Veracruzana

Propiedades I

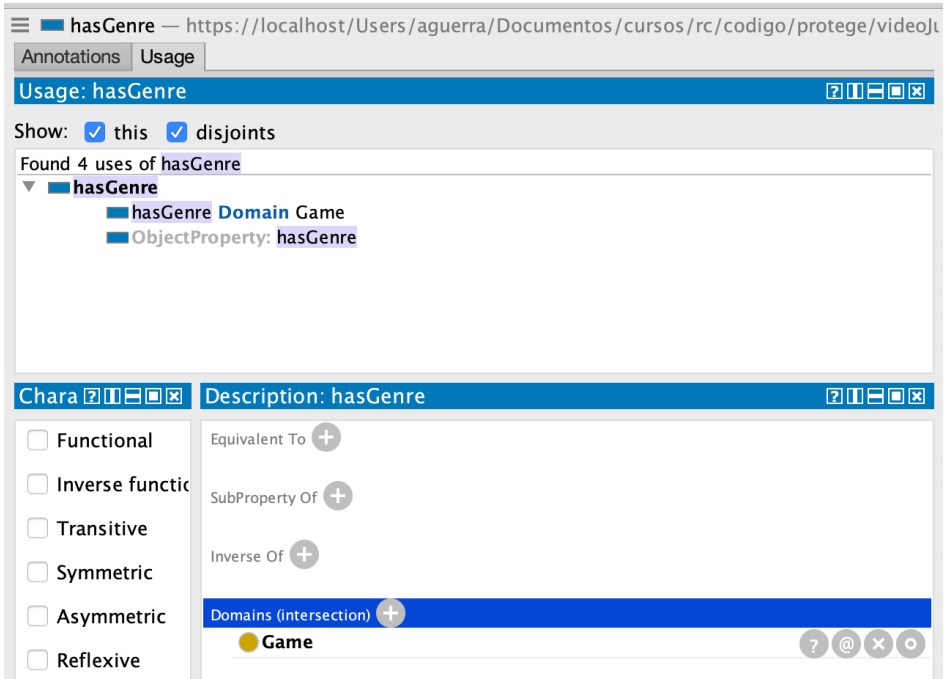
- En la pestaña `Object properties` agregue las propiedades de tener plataforma, dificultad y género:



Universidad Veracruzana

Propiedades II

- Agregue a `hasPlatform` su dominio:



- [FILLS :*hasPlatform Thing*] $\sqsubseteq$ *Game*

Universidad Veracruzana

Propiedades III

- Ahora agregue como rango `Platform`.
- $\mathit{Thing} \sqsubseteq [\text{ALL } :\mathit{hasPlatform}\ \mathit{Platform}]$

Universidad Veracruzana

Axiomas

- ¿Qué tipo de axiomas podemos agregar?
  - $A \sqsubseteq C$, una condición necesaria para A;
  - $A \doteq C$, una condición necesaria y suficiente para A.
- Para cada sub-clase de `namedGame` necesitamos expresar:
  - `Chess` puede instalarse en cualquier plataforma;
  - `LeagueOfLegends` es un juego online.
- `DifficultyValuePartition` necesita definirse correctamente, i.e., sus valores solo pueden ser `Hard`, `Normal` e `Easy`.
- Agregar clases definibles.

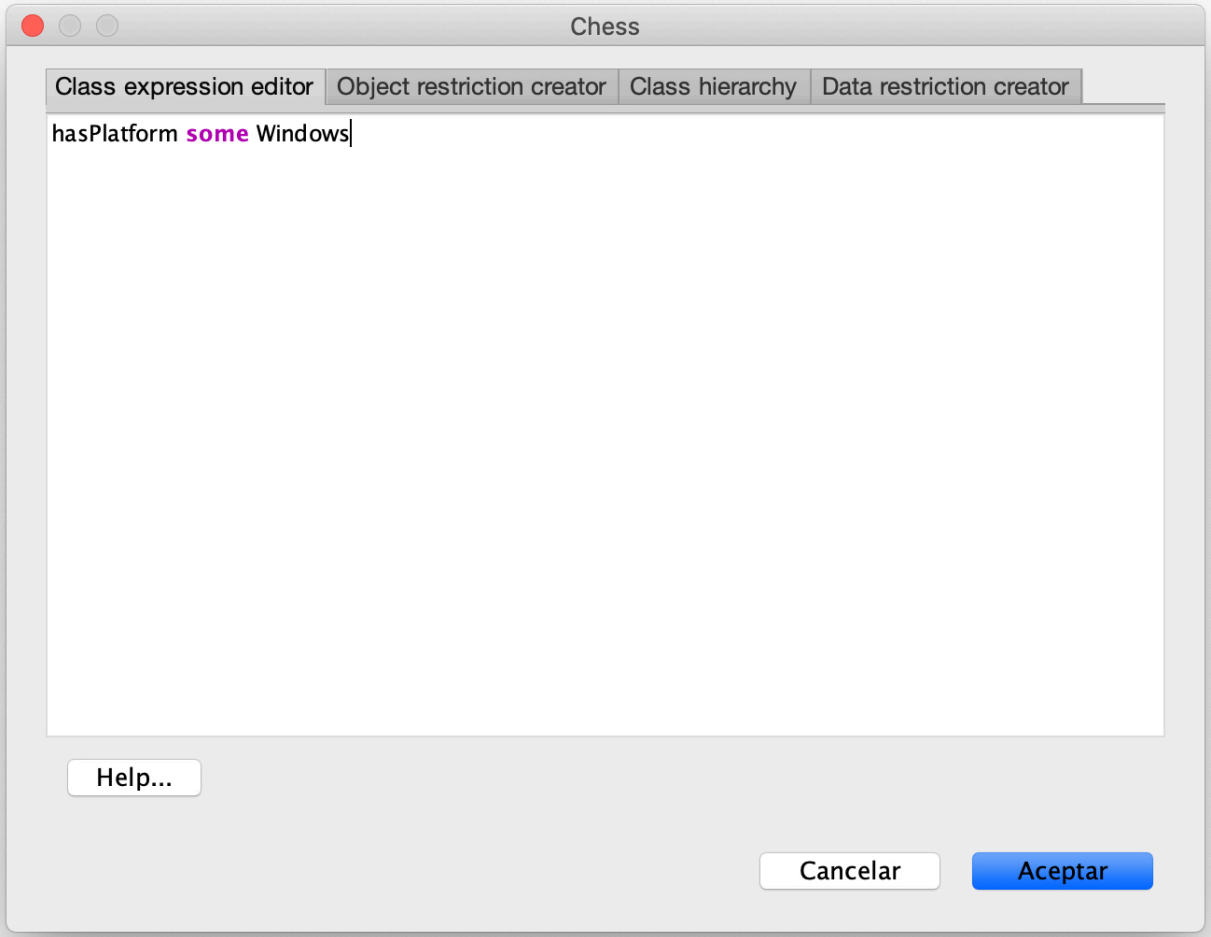
Universidad Veracruzana

Ejemplo $A \sqsubseteq C$

- Lenguaje natural: `Chess` puede jugarse en cualquier plataforma.
- En términos de la ontología: `Chess` tiene como plataforma Window, macOS y Linux.
- Formalmente: *Chess* $\sqsubseteq$ [FILLS :*hasPlatform Windows*]
- Manchester: `hasPlatform some Windows`.

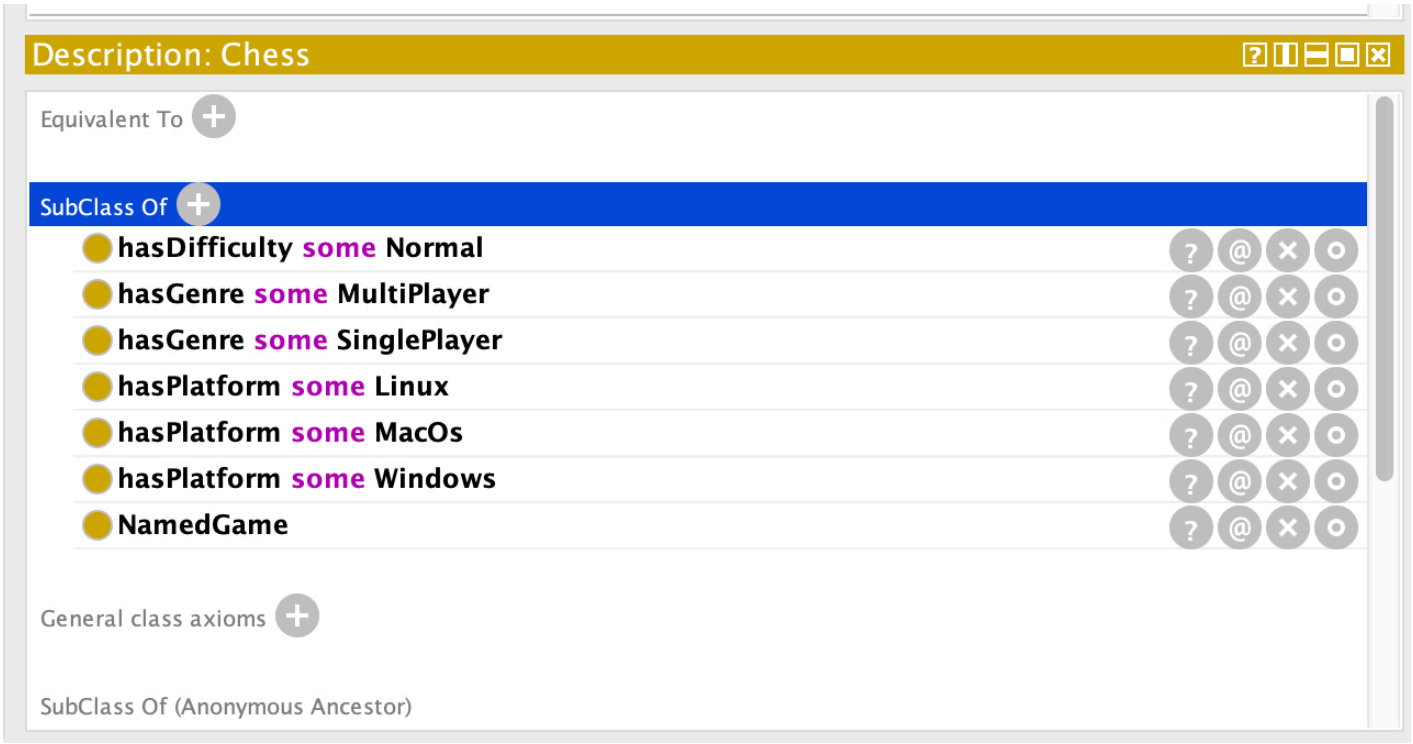
Universidad Veracruzana

Agregar subclase en Protégé



Universidad Veracruzana

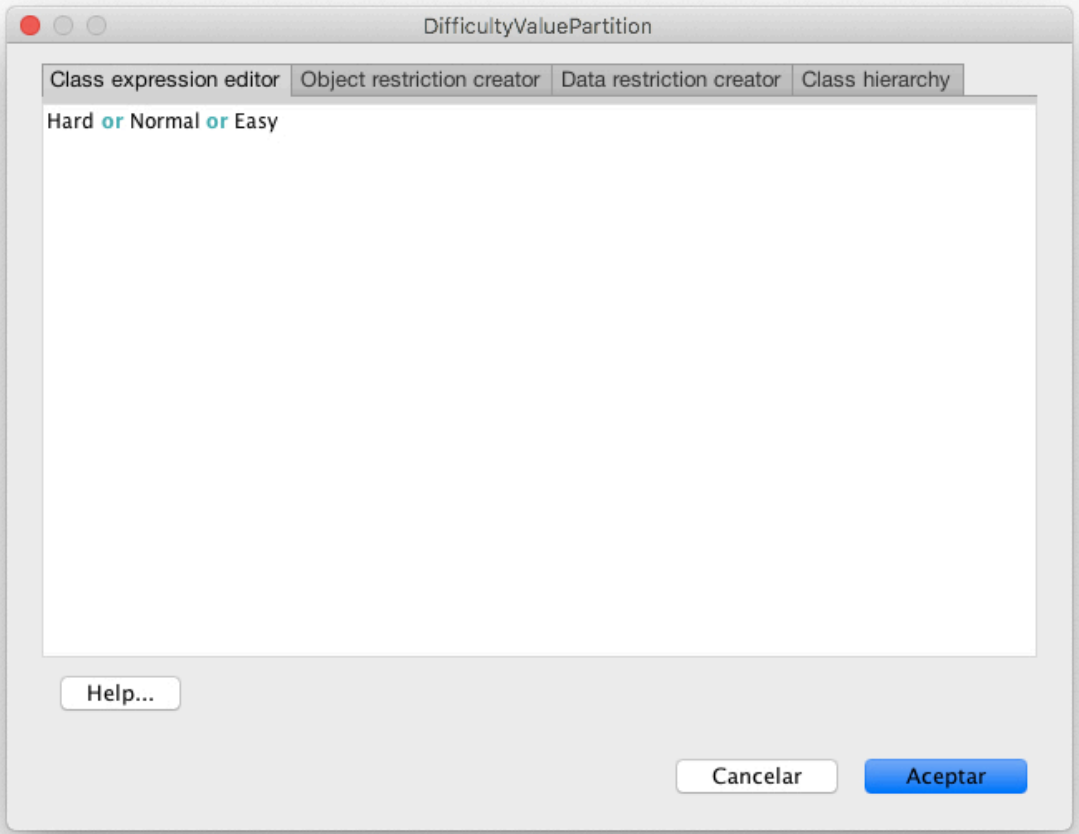
Resultado



Universidad Veracruzana

Equivalencias

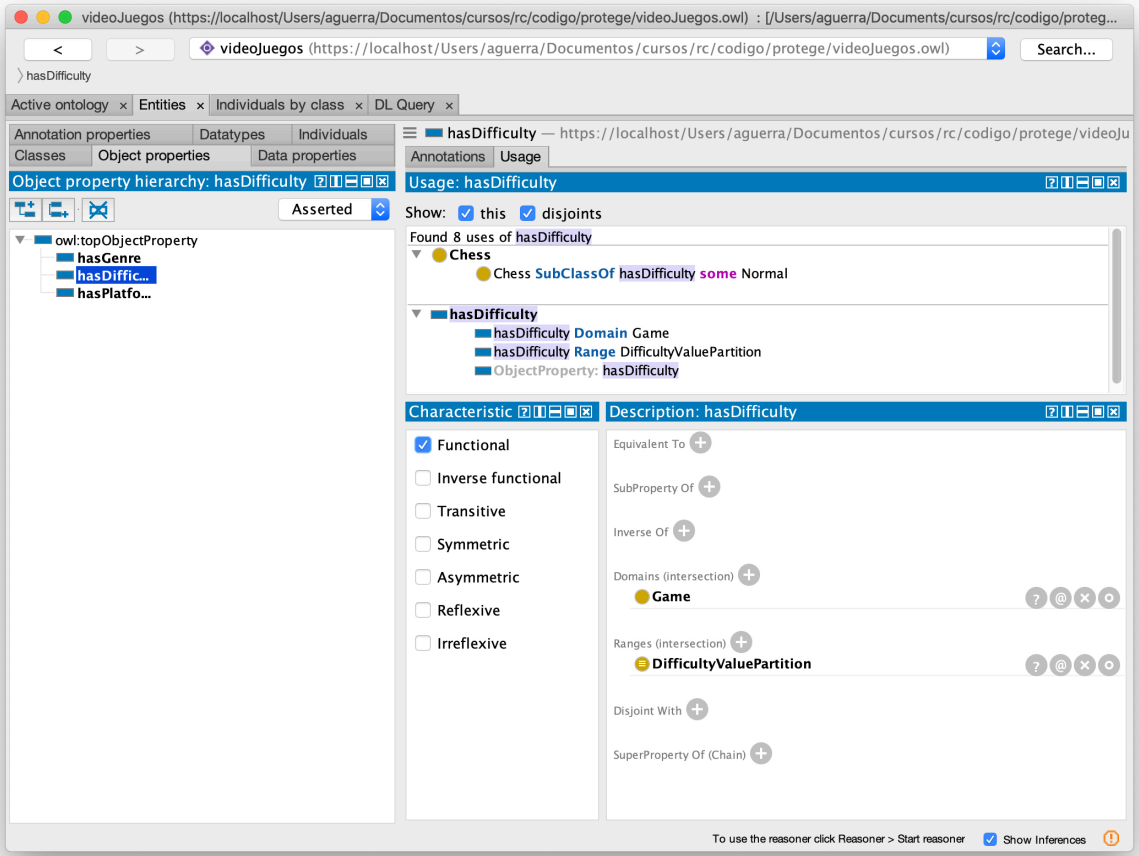
- Se puede mejorar `DifficultyValuePartition`:



Universidad Veracruzana

Cambiando las características de una Propiedad

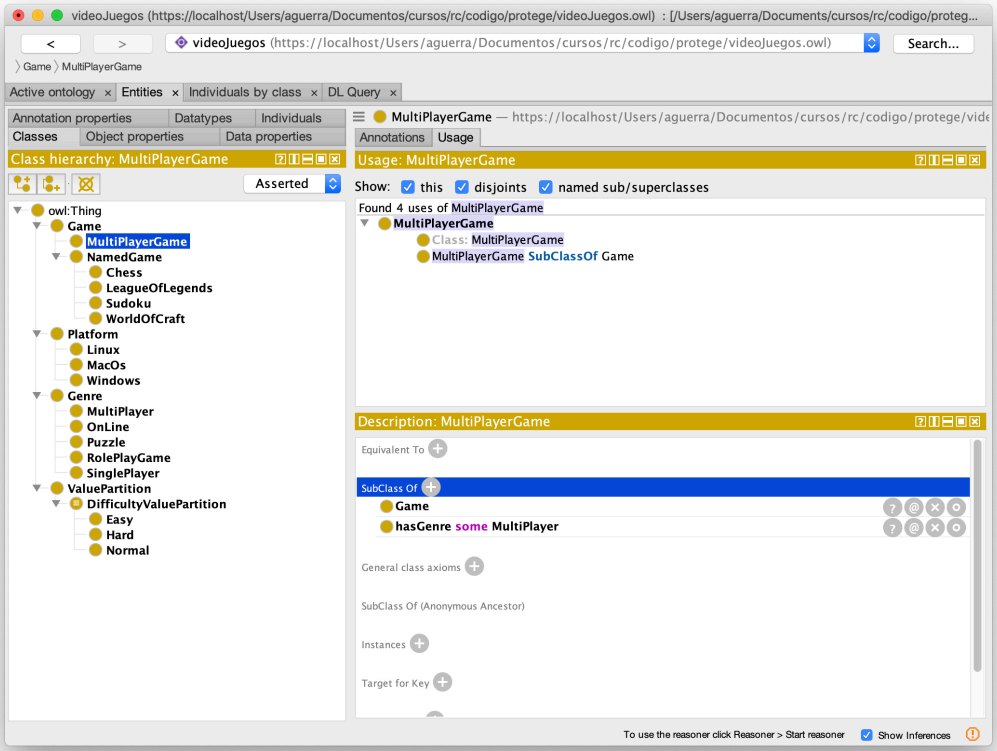
- Vamos a hacer que `hasDifficulty` sea funcional:



Universidad Veracruzana

Clases definibles I

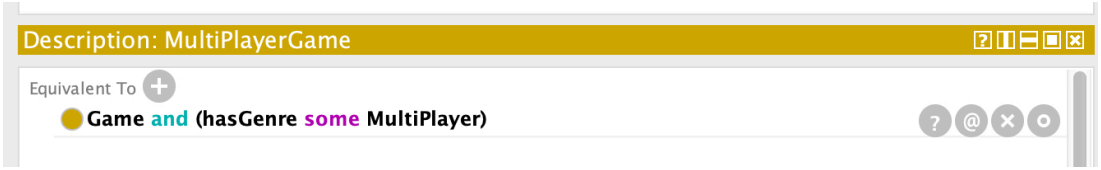
- Vamos a agregar una clase definible `MultiPlayerGame`:



Universidad Veracruzana

Clases definibles II

- Luego en el menú: `Edit > Convert to Defined Class`

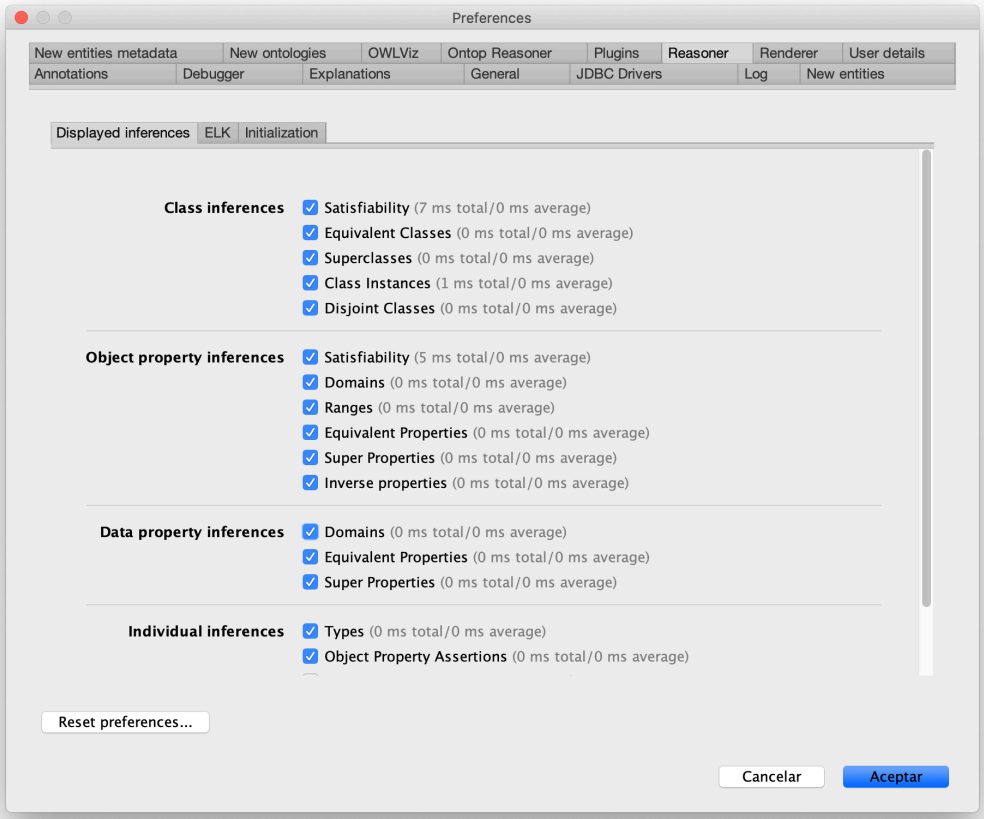


- $MultiPlayerGame \doteq [\text{AND } Game\ [\text{FILLS } :hasGenre\ MultiPlayer]]$

Universidad Veracruzana

Razonamiento I

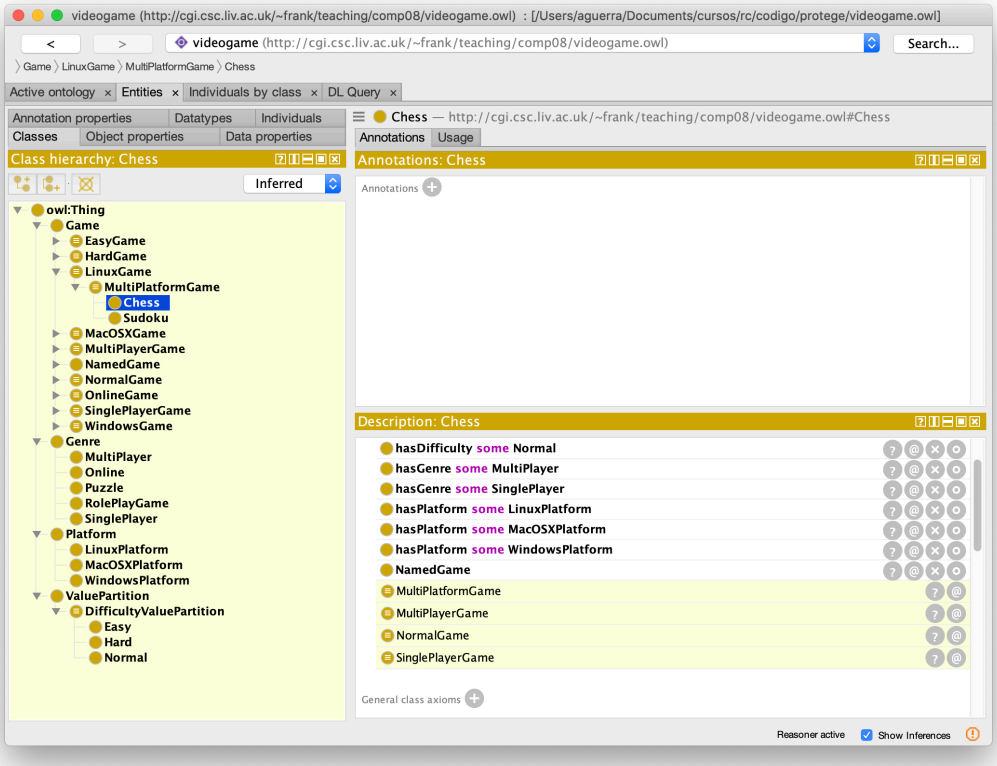
- Seleccionar un razonador, p. ej., en el menú `Reasoner > HermiT`
- Configurarlo:



Universidad Veracruzana

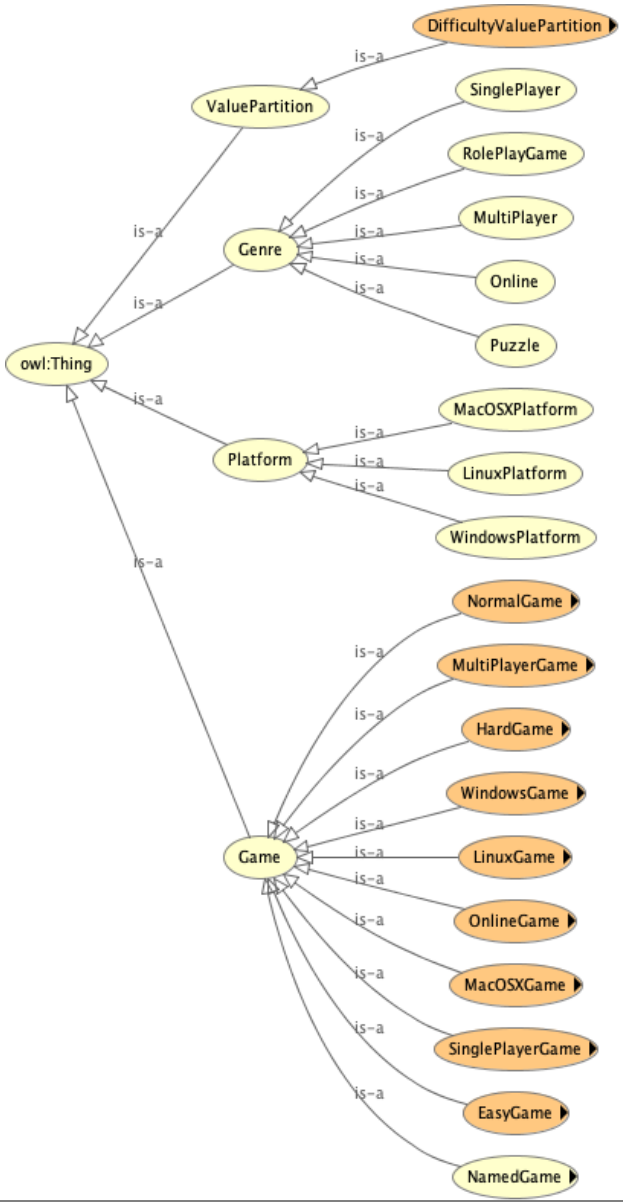
Razonamiento II

- Finalmente `Reasoner > Start reasoner`
- Inferencias:



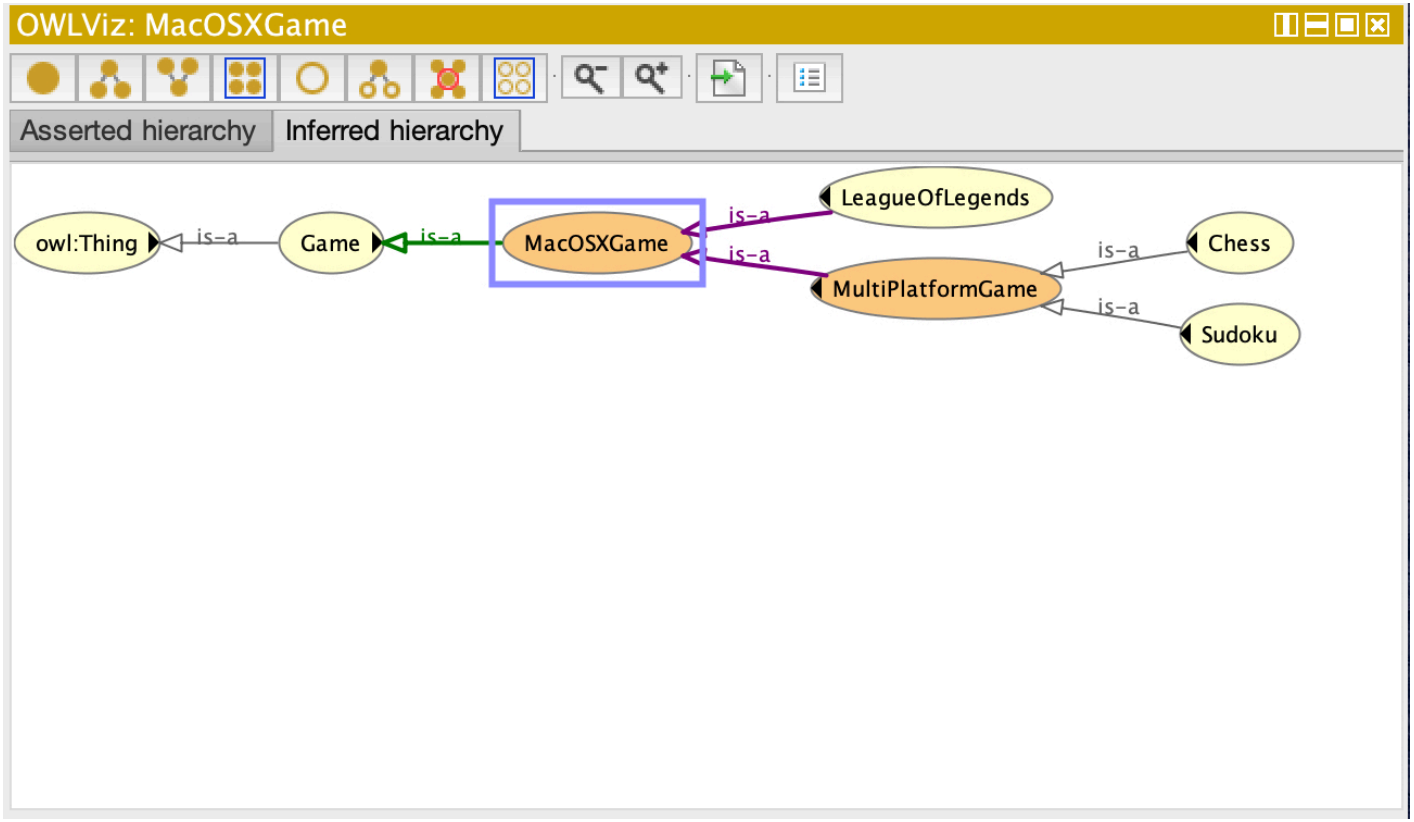
Universidad Veracruzana

Gráficamente (owlViz)



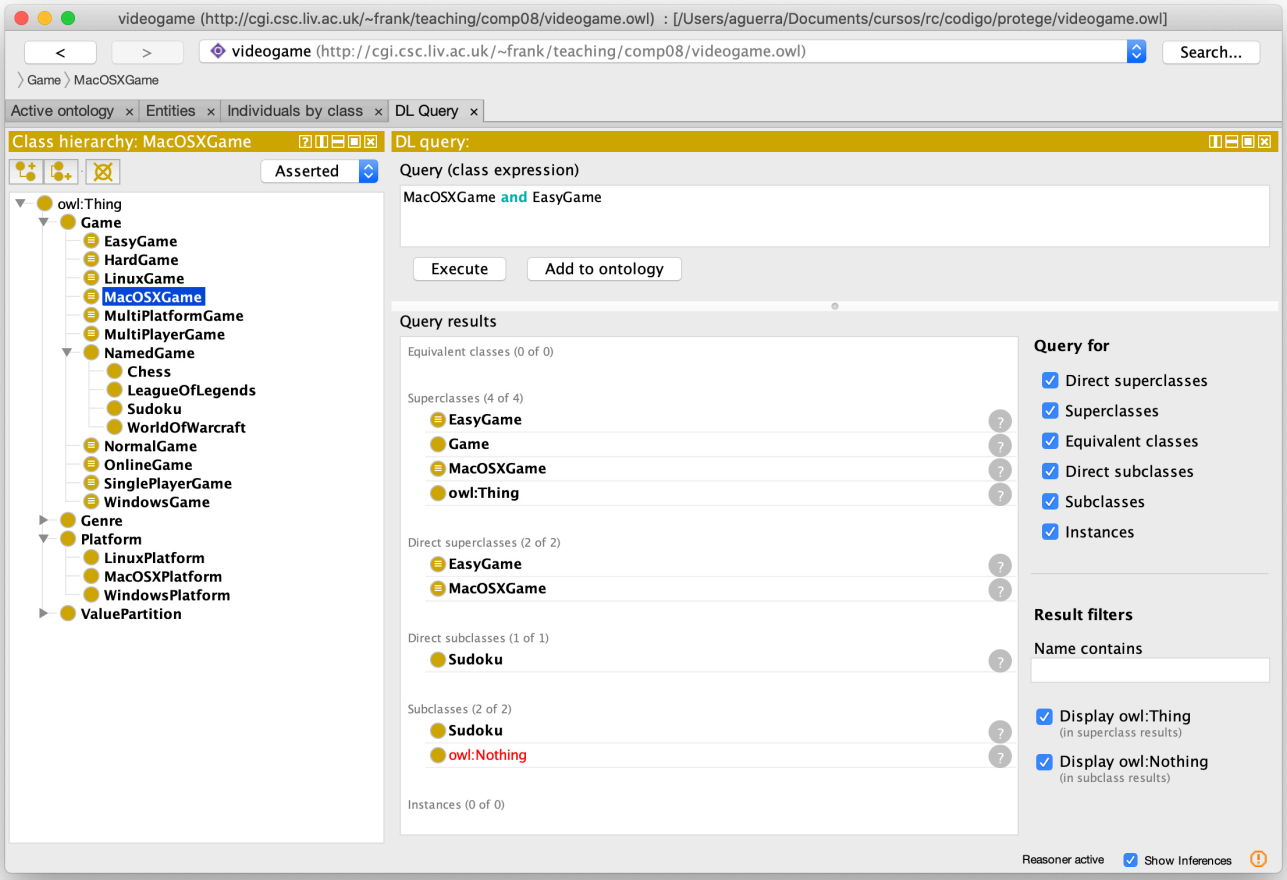
Universidad Veracruzana

owlViz es navegable



Universidad Veracruzana

Consultas



Universidad Veracruzana

Referencias

[1] F Baader et al. *The Description Logic Handboook: Theory, Implementation and Applications*. Second. New York: Cambridge University Press, 2007.

[2] RJ Brachman y HJ Levesque. *Knowledge representation and reasoning*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 2004.

[3] MA Musen. "The Protégé project: A look back and a look forward". En: *AI Matters* 1.4 (2015), págs. 4-12.

Universidad Veracruzana