ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE CUADROS

Parte I

LÓGICAS DE DESCRIPCIÓN

1 INTRODUCCIÓN

Cuando pensamos en el concepto de simetría, quizá lo primero que se nos viene a la cabeza puede ser un polígono regular, un número capicúa o incluso alguna imagen del Test de Rorschach (Figura 1.1).

Figure 1.1: Lámina 8 del Test de Rorschach

Entonces pensamos en algo que se puede *partir sobre un eje*, y repite el mismo patrón, aunque quizá con direcciones diferentes, en ambas partes. En la resolución de problemas matemáticos es muy usual recurrir a las simetrías en términos de razonamiento, ya que se puede analizar un caso en detalle y dicho análsis sería igual para sus pares simétricos.

Esto se lo robé a Ezequiel

En el contexto de razonamiento automático, se puede definir como simetría de una fórmula a una permutación de sus variables (literales) que mantenga su estructura y sus modelos. De esta manera, si somos capaces de demostrar que una fórmula tiene simetrías, podemos guiar al algoritmo de búsqueda para que solo busque soluciones en las partes no simétricas del espacio de búsqueda.

1.1 QUÉ VAMOS A HACER EN ESTA TESIS

En esta tesis vamos a trabajar con un tipo de base de datos muy particulares, llamadas *bases de conocimiento*. Y como en cualquier base de datos, el objetivo es poder hacer consultas y obtener de la manera más rápida posible, una respuesta correcta. Nos concetraremos en buscar métodos para agilizar en cómo se realizan las consultas a nuestra base de conocimientos. Más concretamente estos métodos tienen como objetivo encontrar simetrías en la base de datos, por lo que si, por ejemplo, queremos hacer una pregunta sobre algún concepto muy *complicado*, A, pero nuestro algoritmo encuentra que A es simétrico a B, que es un concepto mucho más simple, entonces podremos hacer nuestra consulta sobre B.

CONSULTA DE USUARIO

↓

ALGORITMO DE CONSULTA ←TRABAJAREMOS HACIENDO ESTA PARTE MÁS EFICIENTE

↓

BASE DE CONOCIMIENTO

1.2 POR QUÉ

1.3 ANTECEDENTES SIMILARES, LA ESTRUCTURA DE ESTA TESIS

Trabajos similares [1] han sido desarrollados para lógicas modales. Trabajo en el cual esta tesis se apoya fuertemente ya que usa herramientas implementadas para dicho proyecto.

1.4 SOBRE SIMETRÍAS EN AR

2 SOBRE DL Y ONTOLOGÍAS

Empezaremos con una descripción informal de las lógicas de descripción y su relación con las ontologías.

2.1 LÓGICAS DE DESCRIPCIÓN

Como sabemos la lógica de primer orden es una lógica en la cual la satisfacibilidad es indecidible, cuando no se aplica ninguna restricción. Las lógicas de descripción son fragmentos decidibles de la lógica de primer orden.

Más precisamente, las lógicas de descripción son una familia de la formalización de la Representación de Conocimiento basado en lógica. Describe dominios en términos de conceptos, roles e individuos. El foco principal de investigar en Lógicas de Descripción es cómo usar sus constructores para aplicarlos al mundo real y cuá es su impacto en la complejidad del razonamiento. Los sistemas de representación basados en Lógicas de Descripción ($\mathcal{DL}$) consisten en dos componentes: por un lado la componente que describe las clases, o conceptos, del universo que estamos intentando modelar. Dicha componente se llama *TBOX* (por *Terminological Box*). Y por otro lado, una componente que expresa restricciones sobre estos conceptos y de qué manera se relacionan entre sí, llamada *ABOX* (por *Assertional Box*).

El trabajo de esta tesis se enfocará particularmente en la TBOX.

2.1.1 *Sintaxis y Semántica*

En la Tabla 2.1 podemos observar la sintaxis y semántica de una lógica muy simple llamada *AL* (por *Attribute Language*). En dicha tabla, A, C y D representan Conceptos y R representa un Rol atómico. La semántica es definida usando una función de interpretación I, que consiste de un conjunto no vacío $\Delta^{\mathcal{I}}$ (el dominio dela interpretación) y una función de interpretación, que asigna un conjunto $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ a cada concepto atómico A, y asigna una relación binaria $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ a cada Rol Atómico R. En la Tabla 2.2 se extiende la función de interpretación I para dar lugar a más constructores y así enriquecer la expresividad.

SINTAXIS	SEMÁNTICA	COMENTARIO
A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$	Concepto Atómico
R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \chi \Delta^{\mathcal{I}}$	Rol Atómico
$\top$	$\Delta^{\mathcal{I}}$	Top, concepto más general
$\bot$	$\emptyset$	Bottom, concepto más específico
$\neg A$	$\Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$	Negación de concepto atómico
$C \prod D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$	Intersección de Conceptos
$\forall R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \forall b.(a,b) \in R^{\mathcal{I}} \Rightarrow b \in C^{\mathcal{I}}\}$	Restricción de valor
$\exists R.\top$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b.(a,b) \in R^{\mathcal{I}}\}$	Cuantificación existencial

Table 2.1

slbsfdb
bslbdsf
bsdlfbsd
lsdfbsdbfdbsfdfb fsbsfb
sbsdbfl

NOMBRE	SINTAXIS	SEMÁNTICA	COMENTARIO
$\mathcal{U}$	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$	Unión de dos Conceptos
$\mathcal{E}$	$\exists R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b.(a,b) \in R^{\mathcal{I}} \wedge b \in C\}$	Cuantificación Completa
$\mathcal{N}$	$\geq nR$ $\leq nR$	$\{a \in \Delta^{\mathcal{I}} \mid\mid \{b.(a,b) \in R^{\mathcal{I}} \mid\geq n\}$ $\{a \in \Delta^{\mathcal{I}} \mid\mid \{b.(a,b) \in R^{\mathcal{I}} \mid\leq n\}$	Restricción de cardinalidad
$\mathcal{C}$	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$	Negación de un concepto Arbitrario

Table 2.2

<Sintaxis / Semántica>
<Ejemplos>

AGREGAR EJEMPLOS

2.1.2 *Clases y Complejedida*

$\mathcal{SHOIQ}$ (http://www.cs.man.ac.uk/~ezolin/dl/)

2.2 ONTOLOGÍAS

Habiendo dado ya una formalización de las lógicas de descripción y de cómo construir sus fórmulas, ya contamos con las herramientas para poder definir algún sistema de interés. Y una ontología es una especificación explícita de dicho sistema o conceptualización (http://www.obitko.com/tutorials/ontologies-semantic-web/ontologies.html)

2.3 SU RELACIÓN CON LÓGICAS MODALES

Las principales herramientas para realizar los experimentos sobre Lógicas de Descripción fueron desarrolladas principalmente para Lógicas Modales, un fragmento particular (y decidible) de Lógica de Primer Orden.

A continuación, una breve introducción sobre Lógicas Modales y luego su relación con Lógicas de Descripción.

2.3.1 *Lógicas Modales*

DEFINICIÓN:

(Sintaxis) Sea $\mathsf{PROP} = \{p_1, p_2, ...\}$ *un conjunto contable infinito de variables proposicionales y* $\mathsf{MOD} = \{m, m''\}$ *un cojunto de símbolos de modalidades. Denominamos como* signatura modal *al par* $\mathcal{S} = \langle \mathsf{PROP}, \mathsf{MOD} \rangle$. *El conjunto de las fórmulas modales básicas* FORM *sobre la signatura* $\mathcal{S}$ *se define como*

$$\mathsf{FORM} ::= p \mid \neg\varphi \mid \varphi \vee \psi \mid [m]\varphi,$$

donde $p \in \mathsf{PROP}$, $m \in \mathsf{MOD}$, *y* $\varphi, \psi \in \mathsf{FORM}.\top$*y* $\perp$ *representan una tautología arbitraria y la contradicción, respectivamente. También utilizaremos los conectivos clásicos como* $\wedge, \rightarrow$ *definidos de la forma usual. Por cada* $m \in \mathsf{MOD}$ *tenemos un* operador dual $\langle m \rangle$ *(diamante) definido como* $\langle m \rangle \varphi = \neg[m]\neg\varphi$. *Cuando* MOD *es un singleton, es decir, en el caso monomodal, simplemente escribimos* $\Box$ *y* $\Diamond$.

DEFINICIÓN:

(Literales y CNF modal). Un literal proposicional *l es una variable proposicional p o su negación* $\neg p$. *El conjunto de literales sobre* PROP *es* $\mathsf{PLIT} = \mathsf{PROP} \cup \{\neg p \mid p \in \mathsf{PROP}\}$. *Una fórmula modal está en* forma normal conjuntiva modal (CNF modal) *si es una conjunción de cláusulas en CNF modal. Una cláusula* en CNF modal *es una disyunción de literales proposicionales*

y modales. Un literal modal *es una fórmula de la forma* $[m]C$ *o* $\neg[m]C$*, donde C es una cláusula en CNF modal.*

DEFINICIÓN:

(Modelos) Un modelo (o modelo de Kripke) $\mathcal{M}$ *es una tripla* $\mathcal{M} = \langle W, \{R^m\}_{m \in \mathsf{MOD}}, V \rangle$*, donde:*

- *W, el* dominio*, es un conjunto no vacío. Los elementos de W se denominan* puntos, estados, mundos*, etc.*
- *Cada* R^m *es una relación binaria sobre W.*
- V*, la* valuación*, es una función que asigna a cada elemento* $w \in W$ *un subconjunto* $V(w) \subseteq \mathsf{PROP}$*. Informalmente, podemos pensar a* $V(w)$ *como el conjunto de variables proposicionales que son verdaderas en* w*.*

DEFINICIÓN:

(Modelos Puntuados) Un modelo puntuado *es un modelo con un elemento distinguido, e.g., dado un modelo* $\mathcal{M} = \langle W, R, V \rangle$ *y un elemento* w *en W, el modelo puntuado correspondiente es la tupla* $\mathcal{M} = \langle w, W, R, V \rangle$*. Si un modelo puntuado* $\mathcal{M}$ *satisface una fórmula* φ*escribimos* $\mathcal{M} \models \varphi$*.*

DEFINICIÓN:

(Semántica) Sea $\mathcal{M} = \langle w, W, R, V \rangle$ *un modelo puntuado y* φ *una fórmula en CNF modal. Definamos recursivamente cuándo una fórmula* φ *es* satisfecha *en* $\mathcal{M}$ *de la siguiente manera:*

$$
\begin{array}{lll}
\mathcal{M} \models \varphi & \textit{sii} & \textit{para todas las cláusulas} C \in \varphi,\ \mathcal{M} \models C, \\
\mathcal{M} \models C & \textit{sii} & \textit{existe un literal } l \in C \textit{ tal que } \mathcal{M} \models l, \\
\mathcal{M} \models p & \textit{sii} & p \in V(w) \textit{ para } p \in \mathsf{PROP}, \\
\mathcal{M} \models \neg p & \textit{sii} & p \notin V(w) \textit{ para } p \in \mathsf{PROP}, \\
\mathcal{M} \models \Box C & \textit{sii} & \langle v, W, R, V \rangle \models C, \textit{ para todo } v \textit{ tal que } wRv, \\
\mathcal{M} \models \Box \neg C & \textit{sii} & \mathcal{M} \not\models \Box C
\end{array}
$$

2.4 HACIA DL

Nos enfocaremos por ahora en la lógica $\mathcal{ALC}$:

- En vez de p_i escribiremos C_i
- En vez de $\varphi \vee \psi$ escribiremos $\varphi \sqcup \psi$
- En vez de $\langle R \rangle \varphi$ escribiremos $\exists R.\varphi$

2.4.1 *Extensión de una fórmula φ en $\mathcal{M}$ ($\| \varphi \|^{\mathcal{M}}$)*

Analizaremos una fórmula de Lógica Modal φ recursivamente:

- $\| p \|^{\mathcal{M}} = \{w \mid w \in \mathcal{I}(p)\}$, o sea, $\| p \|^{\mathcal{M}} = \mathcal{I}(p)$
- $\| \neg\varphi \|^{\mathcal{M}} = \Delta \setminus \| \varphi \|^{\mathcal{M}}$
- $\| \varphi \vee \psi \|^{\mathcal{M}} = \| \varphi \|^{\mathcal{M}} \cup \| \psi \|^{\mathcal{M}}$
- $\| \langle R \rangle \varphi \|^{\mathcal{M}} = \{w \mid \exists v.R(w,v) \wedge v \in \| \varphi \|^{\mathcal{M}}\}$

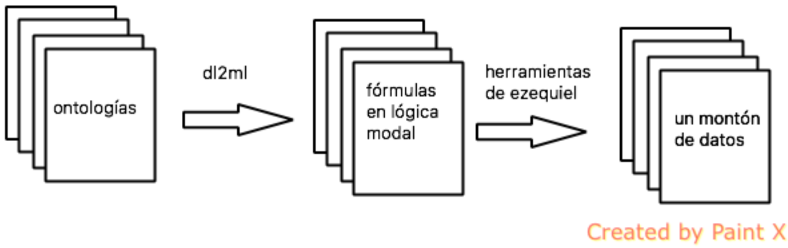
Sea $\mathcal{M} = \langle \Delta, \mathcal{I} \rangle$, con Δ un conjunto no vacío e $\mathcal{I}$ una función de interpretación, decimos que:

$$
\begin{array}{llll}
\langle \Delta, \mathcal{I} \rangle \models & C_i \sqsubseteq C_j & \textit{sii} & \| C_i \|^{\mathcal{M}} \subseteq \| C_j \|^{\mathcal{M}} \\
\langle \Delta, \mathcal{I} \rangle \models & C_i \equiv C_j & \textit{sii} & \| C_i \|^{\mathcal{M}} = \| C_j \|^{\mathcal{M}} \\
\langle \Delta, \mathcal{I} \rangle \models & a : C_i & \textit{sii} & \mathcal{I}(a) \in \| C_i \|^{\mathcal{M}} \\
\langle \Delta, \mathcal{I} \rangle \models & (a,b) : R & \textit{sii} & (\mathcal{I}(a), \mathcal{I}(a)) \in \mathcal{I}(R)
\end{array}
$$

3 SIMETRÍAS EN ONTOLOGÍAS

Ahora que sabemos que los modelos de Lógica Modal son los mismos que los de Lógicas de Descripción, podemos aplicar las mismas herramientas a ambas lógicas. Como las herramientas principales están desarrolladas para Lógica Modal, daremos traducciones de fórmulas en lógica de descripción a fórmulas en lógicas modales.

La arquitectura del sistema a granularidad baja es como sigue:



3.1 LA VERSIÓN ABSTRACTA - DL2ML

In the eagerness of finding symmetries on Description Logics ($\mathcal{DL}$), a set of tools for finding symmetries in Modal Logics ($\mathcal{ML}$) are to be used. But first, a bridge between the two logis needs to be build: the translations are the approach to build such bridge.

TRANSLATIONS

Let $\mathcal{R}$ be a relation and $\mathcal{C}_i$ a concept.

$$\Psi : D_\Psi \subseteq \mathcal{F}_{\mathcal{DL}} \to \mathcal{F}_{\mathcal{ML}}$$

$$\Psi(\mathcal{R}\, someValuesFrom\, \mathcal{C}) \doteq \langle\mathcal{R}\rangle\Psi(\mathcal{C})$$
$$\Psi(\mathcal{R}\, allValuesFrom\, \mathcal{C}) \doteq [\mathcal{R}]\Psi(\mathcal{C})$$
$$\Psi(\mathcal{R}\, hasValue\, \mathcal{C}) \doteq [\mathcal{R}]\Psi(\mathcal{C}) \wedge \langle\mathcal{R}\rangle\Psi(\mathcal{C})$$
$$\Psi(\mathcal{R}^{-} someValuesFrom\, \mathcal{C}) \doteq \langle\mathcal{R}^{-}\rangle\Psi(\mathcal{C})$$
$$\Psi(\mathcal{C}_1 \sqsubseteq \mathcal{C}_2) \doteq \forall(\Psi(\mathcal{C}_1) \implies \Psi(\mathcal{C}_2))$$
$$\Psi(\mathcal{C}_1 \equiv \mathcal{C}_2) \doteq \Psi(\mathcal{C}_1 \sqsubseteq \mathcal{C}_2) \wedge \Psi(\mathcal{C}_2 \sqsubseteq \mathcal{C}_1)$$
$$\Psi(\mathcal{C}_1 \cup \mathcal{C}_2) \doteq \Psi(\mathcal{C}_1) \vee \Psi(\mathcal{C}_2)$$
$$\Psi(\mathcal{C}_1 \cap \mathcal{C}_2) \doteq \Psi(\mathcal{C}_1) \wedge \Psi(\mathcal{C}_2)$$
$$\Psi(\mathcal{R}\, ObjectDomain\, \mathcal{C}) \doteq \forall(\langle\mathcal{R}\rangle\top \implies \Psi(\mathcal{C}))$$
$$\Psi(\mathcal{R}\, ObjectRange\, \mathcal{C}) \doteq \forall(\langle\mathcal{R}^{-}\rangle\top \implies \Psi(\mathcal{C}))$$
$$\Psi(functional\, \mathcal{R}) \doteq \forall(\langle\mathcal{R}\rangle\top \implies [\mathcal{R}]\top)$$
$$\Psi(inverseFunctional\, \mathcal{R}) \doteq \forall(\langle\mathcal{R}^{-}\rangle\top \implies [\mathcal{R}^{-}]\top)$$
$$\Psi(\mathcal{C}_1\, Disjoint\, \mathcal{C}_2) \doteq \forall(\Psi(\mathcal{C}_1) \Leftrightarrow \neg\Psi(\mathcal{C}_2))$$

$$\Psi(Complement\, \mathcal{C}) \doteq \neg\Psi(\mathcal{C})$$
$$\Psi(Cardinality\; type\, \mathcal{N}\, \mathcal{C}) \doteq \langle str(type\mathcal{N}\mathcal{C})\rangle\Psi(\mathcal{C})$$

Parte II

EXPERIMENTACIÓN

4 HERRAMIENTAS [4-5 HOJAS] IMPLEMENTACIÓN

4.1 SCRIPTS

Los principales:

- init.sh:
  - mkdirs.sh: crea los directorios para las salidas de las herramientas sobre lógica modal
  - unzip.sh: descomprime las ontologías para trabajar
  - wipe.sh: borra datos previos (se puede correr init.sh luego de haber obtenido algunos resultados)
  - buid.sh: compila todas las herramientas para lógica modal
- sbt.sh:
  - Se encarga de realizar las traducciones de todas las ontologías que encuentre en el directorio *ontologies/* a lógica modal. También genera un mapeo entre variables proposicionales y los nombres de los conceptos
- doall.sh:
  - k.sh: todas las fórmulas en lógica modal que resolvió nuestro traductor, las pasa a formato CNF
  - s.sh: genera los grafos (no?)
  - b.sh: busca automorfismos en los grafos generados por el script anterior
  - bp.sh: bliss-proc.py
  - map.sh: mapea de vuelta las variables proposicionales a los nombres de los conceptos

Algunos scripts adicionales:

- info.sh:
  - Se encarga de generar archivos de información sobre las ontologías, como tamaño (MB), tbox count, abox count, etc. La información se guarda en <*dir*>
- generate_sheet.py:
  - Genera un archivo con toda la información obtenida por los scripts, de forma estructurada.

4.2 BLISS

Aprender

4.3 MANUAL DE USO

https://github.com/giovanniresciа/thesis

4.4 PIPELINE

Nuestro input será una ontología, por lo que primero empezaremos trabajando con lógicas de descripción. El primer paso para poder aplicar todas las herramientas para encontrar simetrías es hacer uso de las traducciones definidas en la sección 3.1 y luego ir aplicando cada uno de los scripts.

4.4.1 *dl2ml (traducción)*

Este script traduce de DL a LM.

4.4.2 *kcnf*

Este script transforma una fórmula de lógica modal a una versión CNF.

4.4.3 *sy4ncl*

4.4.4 *bliss*

finding automorphism
http://www.tcs.hut.fi/Software/bliss/

4.4.5 *bliss proc*

5 ANÁLISIS DE SIMETRIAS EN ONTOLOGIAS EXISTENTES [5 HOJAS]

5.1 DESCRIPCIÓN DE LAS ONTOLOGÍAS UTILIZADAS

sda Tabla 5.1
hola Tabla 5.2

5.2 TABLA DE DATOS

5.3 ANÁLISIS DE DATOS

5.4 CONCLUSIÓN: ANDA

Ontología	TBOX	Time	Generators	total time
Family	1	0.2	22	2
Galen	2	0.4	33	2

Table 5.1

ONTOLOGÍA	TBOX	GENERATOR
Family	1	3
Galen	2	4
Dolce	3	5

Table 5.2: Resultados

Parte III

CONCLUSIÓN

6 CONCLUSIONES & TRABAJO FUTURO [3 HOJAS]

6.1 resumen de lo hecho (dar los limites del trabajo)

6.2 “opiniones”

6.3 qué más se puede hacer a partir de acá

7 CONCLUSIONES PERSONALES SOBRE EL TRABAJO (OPTATIVA)

BIBLIOGRAFÍA

[1] Ezequiel Orbe. «SLM». Tesis doct. FaMAF, 2014.