Ontologías y Web Semántica
Clase 3-Parte2: Description Logic

Laura Cecchi Germán Braun
{lcecchi,german.braun}@fi.uncoma.edu.ar

Depart. de Teoría de la Computación - Facultad de Informática
Universidad Nacional del Comahue

Marzo 2020

Contenido

Ingredientes de las Lógicas Descriptivas

Una DL está caracterizada por:

- Un Lenguaje de Descripción: cómo formar conceptos y roles

$$Human \sqcap Male \sqcap \exists hasChild \sqcap \forall hasChild.(Doctor \sqcup Lawyer)$$

✓

- Un mecanismo para especificar conocimiento sobre los conceptos y los roles, i.e., un TBox

$$\mathcal{T} = \{Father \equiv Human \sqcap Male \sqcap \exists hasChild;$$

$$HappyFather \sqsubseteq Father \sqcap \exists hasChild.(Doctor \sqcup Lawyer)\}$$

✓

Ingredientes de las Lógicas Descriptivas

Una DL está caracterizada por:

- Un mecanismo para especificar propiedades de los objetos, i.e., un ABox

$$\mathcal{A} = \{\mathit{HappyFather}(\mathit{john}); \mathit{hasChild}(\mathit{john}; \mathit{mary})\}$$

✓

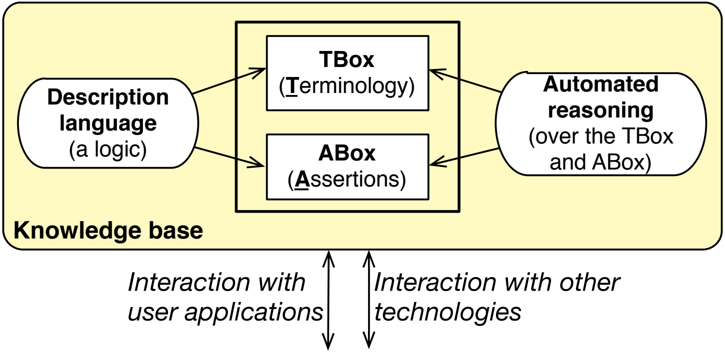
- Un conjunto de servicios de inferencia: cómo razonar sobre una KB dada

$$\mathcal{T} \models \mathit{HappyFather} \sqsubseteq \mathit{Father} \sqcap \exists \mathit{hasChild}.(\mathit{Doctor} \sqcup \mathit{Lawyer})$$

$$\mathcal{T} \cup \mathcal{A} \models (\mathit{Doctor} \sqcup \mathit{Lawyer})(\mathit{mary})$$

✗

Description Logic knowledge base



Ontología como una Base de Conocimiento de las Lógicas Descriptivas

Es un par $O = \langle \mathcal{T}; \mathcal{A} \rangle$, donde $\mathcal{T}$ es un TBox and $\mathcal{A}$ es un ABox.

Description Logics TBox

Consiste de un conjunto de asercciones sobre conceptos y roles:

- Aserciones de inclusión sobre conceptos: $C1 \sqsubseteq C2$
- Aserciones de inclusión sobre roles: $R1 \sqsubseteq R2$
- Aserciones de propiedades sobre roles atómicos:
  (transitive P) (symmetric P) (domain P C)
  (functional P) (reflexive P) (range P C) ...

Description Logics ABox

Consiste de un conjunto de aserciones sobre individuos:

- Aserciones de membresía para conceptos: A(c)
- Aserciones de miembros de roles: P(c1; c2)
- Aserciones de igualdad y desigualdad: $c1 \approx c2$, $c1 \not\approx c2$

Semántica de $\mathcal{ALC}$

Basada en la teoría de modelos.

Interpretación en $\mathcal{ALC}$

Una interpretación $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consiste de

- Un conjunto no vacío de objetos $\Delta^{\mathcal{I}}$ llamado *dominio de la interpretación*
- Una *función de la interpretation* $\cdot^{\mathcal{I}}$ que mapea:
  - Cada concepto atómico A a un subconjunto $A^{\mathcal{I}}$ de $\Delta^{\mathcal{I}}$, i.e. $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
  - Cada role R a un subconjunto $R^{\mathcal{I}}$ de $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, i.e. $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
  - Cada nombre de individuo a a un elemento de $\Delta^{\mathcal{I}}$, i. e. $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
- Note: $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$ and $\bot^{\mathcal{I}} = \emptyset$

Semántica de $\mathcal{ALC}$

- $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
- $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
- $(\forall R.C)^{\mathcal{I}} = \{o | \forall o'.(o, o') \in R^{\mathcal{I}} \rightarrow o' \in C^{\mathcal{I}}\}$
- $(\exists R.)^{\mathcal{I}} = \{o | \exists o'.(o, o') \in R^{\mathcal{I}}\}$

Semántica de $\mathcal{ALC}$

Sean C y D conceptos, R un rol, y a y b individuos

- Una interpretación $\mathcal{I}$ satisface la sentencia $C \sqsubseteq D$, $\mathcal{I} \models C \sqsubseteq D$, si

$$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$$

- Una interpretación $\mathcal{I}$ satisface la sentencia $C \equiv D$, $\mathcal{I} \models C \equiv D$, si $C^{\mathcal{I}} = D^{\mathcal{I}}$
- Una interpretación $\mathcal{I}$ satisface una propiedad P, $\mathcal{I} \models (prop\ P)$, si $P^{\mathcal{I}}$ es una relación que tiene la propiedad prop.
- $C(a)$ es satisfecho por $\mathcal{I}$, $\mathcal{I} \models C(a)$, si $a^{\mathcal{I}} \in C^{\mathcal{I}}$
- $R(a, b)$ es satisfecho por$\mathcal{I}$, $\mathcal{I} \models R(a, b)$, si $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$
- $\mathcal{I} \models c1 \approx c2$ si $c_1^{\mathcal{I}} = c_2^{\mathcal{I}}$
- $\mathcal{I} \models c1 \not\approx c2$ si $c_1^{\mathcal{I}} \neq c_2^{\mathcal{I}}$

Semántica de $\mathcal{ALC}$

Modelo de un Concepto

Una interpretación $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ es un modelo de un concepto C si $C^{\mathcal{I}} \neq \emptyset$.

Modelo de una Base de Conocimiento

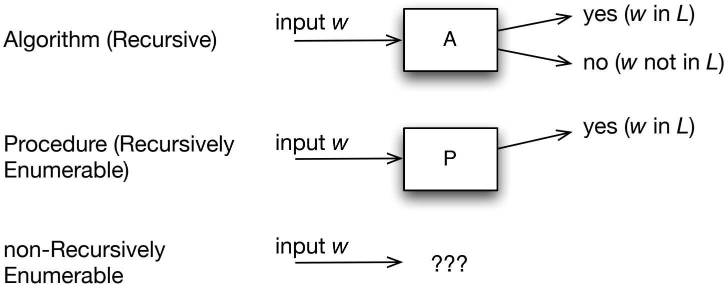
Una interpretación $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ es un modelo de la base de conocimiento $\mathcal{KB}$ si cada axioma de $\mathcal{KB}$ es satisfecho por $\mathcal{I}$.

Satisfacible

Una base de conocimiento $\mathcal{KB}$ se dice satisfacible si admite un modelo.

DLs son fragmentos de FOL

- Recordemos que FOL es **indecidible**
- Esto no es bueno para el razonamiento automático



DLs son fragmentos de FOL

- Enfoque: encuentre un fragmento —un *sublenguaje*— de FOL que es decidible
- Tome algunas características, pruebe la complejidad computacional de algunos problemas
- Pero primero demostremos que FOL y DL están relacionadas

Correspondencia entre DL y FOL

- $C \sqsubseteq D$
  - $\forall x(C(x) \rightarrow D(x))$
- $C \sqsubseteq D \sqcap E$
  - $\forall x(C(x) \rightarrow D(x) \wedge E(x))$
- $C \sqsubseteq \exists R.D$
  - $\forall x(C(x) \rightarrow \exists y(R(x, y) \wedge D(y))$
- $C \equiv \exists R.D \sqcup \exists S.D$
  - $\forall x(C(x) \leftrightarrow \exists y((R(x, y) \vee S(x, y)) \wedge D(y))$

Conceptos Esenciales del Razonamiento Automático

- Elegir la clase de problemas que queremos resolver (Problema de decisión)
- El lenguaje formal en el que representar los problemas
- El modo en que los programas computan la solución
- Cómo hacerlo eficientemente

Implicación Lógica

Es el problema de decisión o servicio de razonamiento fundamental del que los otros son derivados

Implicación Lógica

Una base de conocimiento $\mathcal{KB}$ implica lógicamente una aserción ϕ, y lo denotaremos $\mathcal{KB} \models \phi$, si cada modelo de $\mathcal{KB}$ es un modelo de ϕ.

Podemos dar una definición análoga para el TBox $\mathcal{T}$ en vez de $\mathcal{KB}$.

Servicios de Razonamiento para ontologías basadas en DL

Recordemos que

Modelo de un Concepto

Una interpretación $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ es un modelo de un concepto C si $C^{\mathcal{I}} \neq \emptyset$.

El servicio Concept (and role) Satisfiability se define como

$$\mathcal{KB} \not\models C \sqsubseteq \bot$$

- ¿existe un modelo de $\mathcal{KB}$ en el cual C (resp. R) tenga una extensión no vacía?

Servicios de Razonamiento para ontologías basadas en DL

- Consistency of the knowledge base ($\mathcal{KB} \not\models \top \sqsubseteq \bot$)
  - Is the $\mathcal{KB} = (\mathcal{T}, \mathcal{A})$ consistent (non-selfcontradictory), i.e., is there at least a model for $\mathcal{KB}$?
- Concept (and role) subsumption ($\mathcal{KB} \models C \sqsubseteq D$)
  - i.e., is the extension of C (resp. R) contained in the extension of D (resp. S) in every model of $\mathcal{T}$?
- Concept (and role) equivalence ($\mathcal{KB} \models C \equiv D$)
  - i.e., is the extension of C (resp. R) equal in the extension of D (resp. S) in every model of $\mathcal{T}$?
- Concept (and role) disjointness ($\mathcal{KB} \models C \sqcap D \equiv \bot$)
  - i.e., $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ for every model of $\mathcal{KB}$.

Servicios de Razonamiento para ontologías basadas en DL

- KB Satisfiability: Verify whether a $\mathcal{KB}$ is satisfiable, i.e., whether $\mathcal{KB}$ admits at least one model.
- Instance checking ($\mathcal{KB} \models C(a)$ or $\mathcal{KB} \models R(a, b)$)
  - is a (resp. (a, b)) a member of concept C (resp. R) in $\mathcal{KB}$, i.e., is the fact $C(a)$ (resp. $R(a, b)$) satisfied by every interpretation of $\mathcal{KB}$?
- Instance retrieval ($\{a \mid \mathcal{KB} \models C(a)\}$)
  - find all members of C in $\mathcal{KB}$, i.e., compute all individuals a s.t. $C(a)$ is satisfied by every interpretation of $\mathcal{KB}$

Técnica de Razonamiento Automático

Razonamiento por Tableaux

Misma idea que para FOL

Complejidad de razonar sobre Conceptos

Complejidad de Concept satisfiability: [Donini et al., 1997]

$\mathcal{AL}, \mathcal{ALN}$	PTime
$\mathcal{ALU}, \mathcal{ALUN}$	NP-complete
$\mathcal{ALE}$	coNP-complete
$\mathcal{ALC}, \mathcal{ALCN}, \mathcal{ALCI}, \mathcal{ALCQI}$	PSpace-complete

Observaciones:

- Dos fuentes de complejidad:
  - Unión ($\mathcal{U}$) de tipo NP,
  - Cuantificación existencial cualificado ($\mathcal{E}$) de tipo coNP.
- Restricciones numéricas ($\mathcal{N}$) no agrega complejidad.

Complejidad de razonar la Base de Conocimiento

Razonar sobre la $\mathcal{KB}$ es mucho más complejo que razonar sobre conceptos.

Malas noticias:

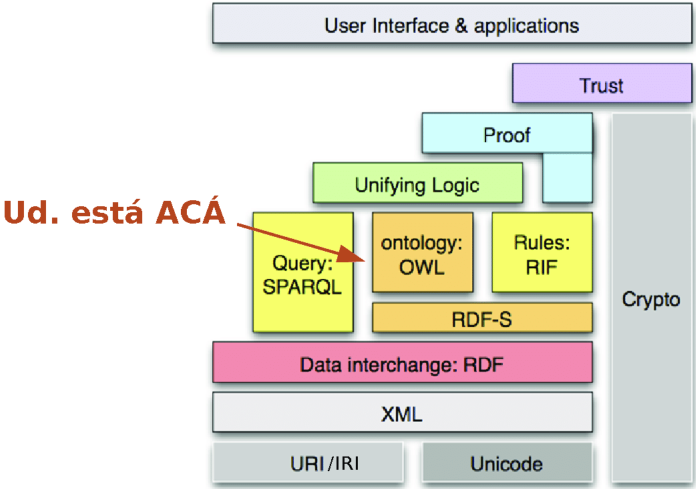
Sin restricciones de forma sobre las aserciones del TBox, razonar sobre la $\mathcal{KB}$ es ExpTime-hard, aún para DLs simples [Donini, 2003].

Buenas noticias:

- Podemos agregar mucha expresividad (i.e., esencialmente todos los constructores DL vistos), manteniendo ExpTime como cota superior.
- Existen razonadores que ejecutan sobre DL razonablemente bien en la práctica para tales DLs (ej., Racer, Pellet, Fact++, . . .) [Möller and Haarslev, 2003].

The Wedding Cake...

La “torta” de la Web Semántica



Lenguajes Ontológicos de la Web

Diferentes versiones del estándar W3C de Web Ontology Language (OWL) han sido definidos como variantes sintácticas de las DLs.

- OWL1 Lite es una variante de la DL $\mathcal{SHIF}(D)$, donde:
  - $\mathcal{S}$ representa $\mathcal{ALC}$ extendida con roles transitivos,
  - $\mathcal{H}$ representa roles jerárquicos (i.e., aserciones de inclusión de roles),
  - $\mathcal{I}$ representa roles inversos,
  - $\mathcal{F}$ representa funcionalidad de roles,
  - (D) representa tipos de datos, los que son necesarios en cualquier lenguaje de representación de conocimiento.
- OWL1 DL es una variante de la DL $\mathcal{SHOIN}(D)$, donde:
  - $\mathcal{O}$ representa los nominales, lo que significa la posibilidad de usar individuos en el TBox (i.e., la parte intensionalde la ontología),
  - $\mathcal{N}$ representa restricciones numéricas (no cualificadas).

https://www.w3.org/TR/2017/REC-owl-time-20171019/

Lenguajes Ontológicos de la Web

La última versón estandarizada por W3C es OWL2:
W3C Recommendation of 11/12/2012
`http://www.w3.org/2007/OWL/wiki/OWL_Working_Group`

- OWL2 DL es una variante de la DL $\mathcal{SROIQ}(D)$, que agrega a OWL1 DL algunos constructores, preservando la decidibilidad del razonamiento:
  - $\mathcal{Q}$ representa restricciones de números cualificadas
  - $\mathcal{R}$ representa jerarquía de roles regular.
- OWL2 define tres profiles: OWL2 QL, OWL2 EL, OWL2 RL.
  - Cada profile corresponde a un fragmento sintáctico (i.e., a un sub-languaje) de OWL2 DL, que está orientado a un uso específico.
  - La restricción en cada profile garantiza mejores propiedades computacionales que las de OWL2 DL.
  - El OWL2 QL profile es derivado de las DLs de la familia DL-Lite.

`https://www.w3.org/TR/owl2-profiles/#Profile_Specification`

Constructores DL vs OWL

OWL constructor	DL constructor	Example
ObjectIntersectionOf	$C_1 \sqcap \cdots \sqcap C_n$	Human $\sqcap$ Male
ObjectUnionOf	$C_1 \sqcup \cdots \sqcup C_n$	Doctor $\sqcup$ Lawyer
ObjectComplementOf	$\neg C$	$\neg$Male
ObjectOneOf	$\{a_1\} \sqcup \cdots \sqcup \{a_n\}$	{john} $\sqcup$ {mary}
ObjectAllValuesFrom	$\forall P.C$	$\forall$hasChild.Doctor
ObjectSomeValuesFrom	$\exists P.C$	$\exists$hasChild.Lawyer
ObjectMaxCardinality	$(\leq n\ P)$	($\leq$ 1 hasChild)
ObjectMinCardinality	$(\geq n\ P)$	($\geq$ 2 hasChild)

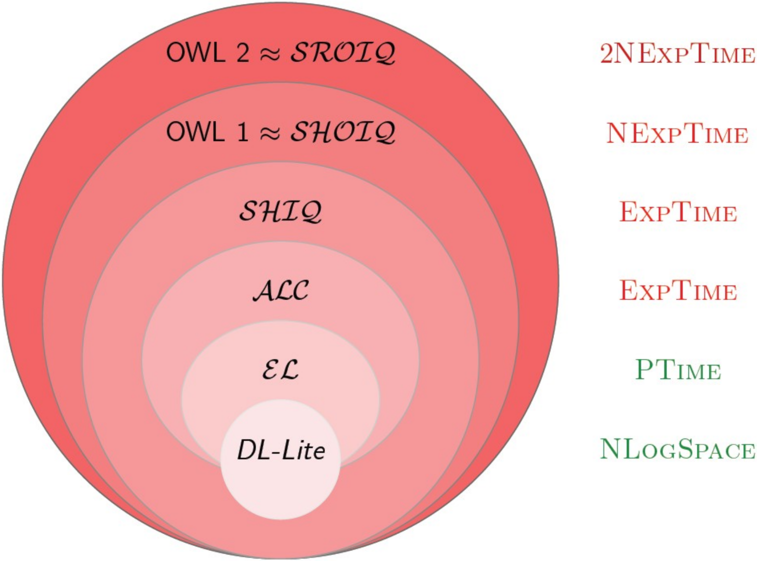
...

Axiomas DL vs OWL

OWL axiom	DL syntax	Example
SubClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
EquivalentClasses	$C_1 \equiv C_2$	Man $\equiv$ Human $\sqcap$ Male
DisjointClasses	$C_1 \sqsubseteq \neg C_2$	Man $\sqsubseteq$ $\neg$Female
SameIndividual	$\{a_1\} \equiv \{a_2\}$	{presBush} $\equiv$ {G.W.Bush}
DifferentIndividuals	$\{a_1\} \sqsubseteq \neg\{a_2\}$	{john} $\sqsubseteq$ $\neg${peter}
SubObjectPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
EquivalentObjectProperties	$P_1 \equiv P_2$	hasCost $\equiv$ hasPrice
InverseObjectProperties	$P_1 \equiv P_2^-$	hasChild $\equiv$ hasParent$^-$
TransitiveObjectProperty	$P^+ \sqsubseteq P$	ancestor$^+$ $\sqsubseteq$ ancestor
FunctionalObjectProperty	$\top \sqsubseteq (\leq 1\, P)$	$\top \sqsubseteq (\leq 1$ hasFather$)$

...

Complejidad de las DL



Resumiendo

- An ontology, as a conceptualization of a domain of interest, provides the mechanisms for modeling the domain and reasoning upon it, and has to be represented in terms of a well-defined language.
- Description Logics are logics specifically designed to represent structured knowledge and to reason upon it, and as such are perfectly suited as languages for representing ontologies.
- Proveen un método para razonar
- Existen diferentes lenguajes de ontologías, llamados Web Ontology Language OWL2 y sus variantes (llamados profiles)
- El razonamiento tiene cota superior en las clases exponenciales.

Bibliografía

- An Introduction to Ontology Engineering. v1
  Keet, C. Maria - 2020
- Formal Ontology, Conceptual Analysis and Knowledge Representation - Nicola Guarino
  International journal of human-computer studies- 43(5) -Elsevier - 1995
- Representing and Reasoning over a Taxonomy of Part-Whole Relations. Keet, C.M., Artale, A. *Applied Ontology*, 2008, 3(1-2):91-110.
- The Description Logic Handbook: Theory, Implementation and Applications.
  Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors.
  Cambridge University Press, 2003.
- Francesco M. Donini.
  Complexity of reasoning.
  In Baader et al.,chapter 3, pages 96-36, 2003.
- Ralf Möller and Volker Haarslev.
  Description logic systems.
  In Baader et al., chapter 8, pages 282-305,2003.

¿Preguntas?

¡Descanso!