Description logics

Description logics

- A family of knowledge representation languages
- Uses in different application areas (e.g., software management, configuration management, natural language processing, clinical information systems, information retrieval)
- Key technology for Ontologies and the Semantic Web

Ontologies, Description Logics and OWL terminology

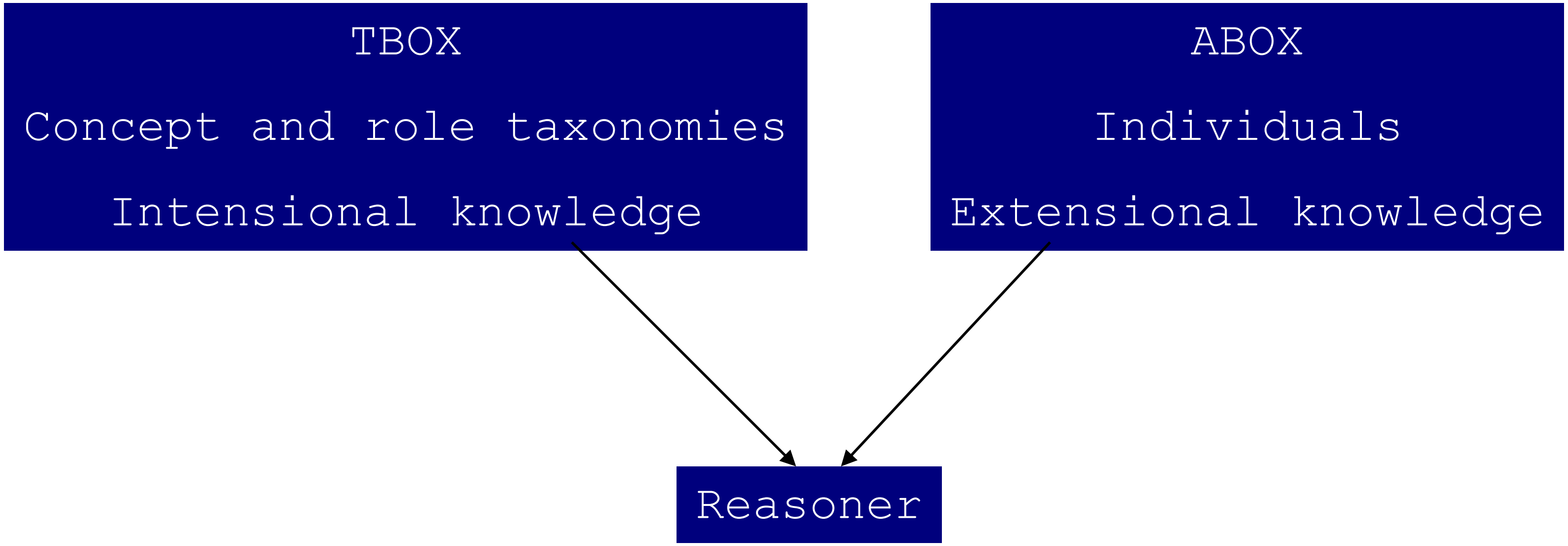
Ontologies	DL	OWL
concept	concept	class
relation	role (binary)	property
axiom	axiom	axiom
instance	individual	individual

Outline

- DL languages
  - syntax and semantics
- DL reasoning services
  - algorithms, complexity
- DL systems
- DLs for the web

DL SYNTAX

Tbox and Abox



$\mathcal{AL}$

R atomic role, A atomic concept

C,D → A | (atomic concept)

T | (universal concept, top) *owl:thing*

⊥ | (bottom concept) *owl:nothing*

¬A | (atomic negation) *owl:complementOf*

C ∩ D | (conjunction) *owl:intersectionOf*

∀R.C | (value restriction) *owl:allValuesFrom*

∃R.T (limited existential quantification) *owl:someValuesFrom*

$\mathcal{AL}[\mathcal{X}]$

$\mathcal{C}$ ¬C (concept negation) *owl:complementOf*

$\mathcal{U}$ C U D (disjunction) *owl:unionOf*

$\mathcal{E}$ ∃R.C (existential quantification)

owl:someValuesFrom

$\mathcal{N}$ ≥ n R, ≤ n R (number restriction)

owl:maxCardinality, owl:minCardinality

$\mathcal{Q}$ ≥ n R.C, ≤ n R.C (qualified number restriction)

owl:maxQualifiedCardinality,owl:minQualifiedCardinality

Concepts and relations

Team ← Concept/class

(Team)

$\neg$Team

(**not** Team)

Relation/role/property

Team $\cap \geq 10$ hasMember

(Team **and at least** 10 members)

Team $\cap \leq 10$ hasMember

(Team **and at most** 10 members)

Concepts and relations

Team $\cap$ $\forall$ hasMember.Football-player

(Team **and all** members **are** football players)

Team $\cap$ $\exists$ hasMember.Football-player

(Team **and there is a** member **that is a** football player)

$\mathcal{AL}[\mathcal{X}]$

$\mathcal{R}$ R $\cap$ S (role conjunction)

$\mathcal{I}$ R- (inverse roles)

$\mathcal{H}$ (role hierarchies)

$\mathcal{F}$ $u_1 = u_2$, $u_1 \neq u_2$ (feature (dis)agreements)

$\mathcal{S}[\mathcal{X}]$

$\mathcal{S}$ $\mathcal{ALC}$ + transitive roles

$\mathcal{SHIQ}$ $\mathcal{ALC}$ + transitive roles

+ role hierarchies

+ inverse roles

+ number restrictions

Tbox - Terminological axioms

- $C = D$ ($R = S$)
  owl:equivalentClass / owl:equivalentProperty

- $C \subseteq D$ ($R \subseteq S$)
  rdfs:subClassOf / rdfs:subPropertyOf
  Football-player $\subseteq$ Athlete
  (**Every** football player **is a**n athlete)

- (disjoint C D)
  owl:disjointWith

Tbox

- An equality whose left-hand side is an atomic concept is a definition.
- A finite set of definitions T is a Tbox (or terminology) if no symbolic name is defined more than once.

Example

$Team \subseteq \geq 2\ hasMember$

$Large\text{-}Team = Team \cap \geq 10\ hasMember$

$Sports\text{-}team = Team \cap \forall\ hasMember.Athlete$

$Football\text{-}Team = Team \cap \geq 11\ hasMember$

$\qquad \cap \forall\ hasMember.Football\text{-}player$

$Football\text{-}player \subseteq Athlete$

DL as sublanguage of FOPL

Team(this)

$\wedge$

($\exists$ $x_1,\ldots,x_{11}$:

hasMember(this,x1) ^ … ^ hasMember(this,x11)

$\wedge\ x_1 \neq x_2 \ \wedge \ldots \wedge\ x_{10} \neq x_{11}$)

$\wedge$

($\forall$ x: hasMember(this,x) → Football-player(x))

Abox

- Assertions about individuals:
  - C(a)
  - R(a,b)

Example

Football-Team(Real_Madrid)
hasMember(Real_Madrid, Eden_Hazard)

Knowledge base

A knowledge base is a tuple $< T, A >$ where T is a Tbox and A is an Abox.

Example

Team $\sqsubseteq \geq 2$ hasMember
Large-Team = Team $\sqcap \geq 10$ hasMember
Sports-team = Team $\sqcap \forall$ hasMember.Athlete
Football-Team = Team $\sqcap \geq 11$ hasMember
$\sqcap \forall$ hasMember.Football-player
Football-player $\sqsubseteq$ Athlete

Football-Team(Real_Madrid)
hasMember(Real_Madrid, Eden_Hazard)

Example - OWL

```
<Declaration>  <ObjectProperty IRI="#hasmember"/> </Declaration>

<Declaration>  <Class IRI="#football-player"/> </Declaration>
<Declaration>  <Class IRI="#athlete"/> </Declaration>
<Declaration>  <Class IRI="#team"/> </Declaration>
<Declaration>  <Class IRI="#large-team"/> </Declaration>
<Declaration>  <Class IRI="#sports-team"/> </Declaration>
<Declaration>  <Class IRI="#football-team"/> </Declaration>

<Declaration>  <NamedIndividual IRI="#Real_Madrid"/> </Declaration>
<Declaration>  <NamedIndividual IRI="#Eden_Hazard"/> </Declaration>
```

Example - OWL

Large-Team = Team $\cap$ $\geq$ 10 hasMember

```
<EquivalentClasses>
    <Class IRI="#large-team"/>
    <ObjectIntersectionOf>
      <Class IRI="#team"/>
      <ObjectMinCardinality cardinality="10">
        <ObjectProperty IRI="#hasmember"/>
      </ObjectMinCardinality>
    </ObjectIntersectionOf>
  </EquivalentClasses>
```

Example - OWL

Football-Team = Team $\cap \geq 11$ hasMember $\cap \forall$ hasMember.Football-player

```
<EquivalentClasses>
    <Class IRI="#football-team"/>
    <ObjectIntersectionOf>
      <Class IRI="#team"/>
      <ObjectAllValuesFrom>
        <ObjectProperty IRI="#hasmember"/>
        <Class IRI="#football-player"/>
      </ObjectAllValuesFrom>
      <ObjectMinCardinality cardinality="11">
        <ObjectProperty IRI="#hasmember"/>
      </ObjectMinCardinality>
    </ObjectIntersectionOf>
  </EquivalentClasses>
```

DL SEMANTICS

$\mathcal{AL}$ (Semantics)

An interpretation $\mathcal{I}$ consists of a non-empty set $\Delta^{\mathcal{I}}$ (the domain of the interpretation) and an interpretation function $.^{\mathcal{I}}$ which assigns to every atomic concept A a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to every atomic role R a binary relation

$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

The interpretation function is extended to concept definitions using inductive definitions.

$\mathcal{AL}$ (Semantics)

C,D → A |	(atomic concept)	
T |	(universal concept)	$T^{\mathcal{I}} = \Delta^{\mathcal{I}}$
⊥ |	(bottom concept)	$\bot^{\mathcal{I}} = \emptyset$
¬A |	(atomic negation)	$(\neg A)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$
C ∩ D |	(conjunction)	$(C \cap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
∀R.C |	(value restriction)	$(\forall R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \forall b.(a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$
∃R.T |	(limited existential quantification)	$(\exists R.T)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b.(a,b) \in R^{\mathcal{I}}\}$

$\mathcal{ALC}$ (Semantics)

$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$

$(C \cup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$

$(\geq n\ R)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \#\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\} \geq n\}$

$(\leq n\ R)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \#\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\} \leq n\}$

$(\exists R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b \in \Delta^{\mathcal{I}} : (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$

Semantics

Individual i

$i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$

Unique Name Assumption:

if $i_1 \neq i_2$ then $i_1^{\mathcal{I}} \neq i_2^{\mathcal{I}}$

Semantics

An interpretation $\mathcal{I}$ is a model for a terminology *T* iff

$C^{\mathcal{I}} = D^{\mathcal{I}}$ for all C = D in *T*

$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all a $C \subseteq D$ in *T*

$C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ for all (disjoint C D) in *T*

Semantics

An interpretation $.^{\mathcal{I}}$ is a model for a knowledge base <*T*, *A* > iff

$.^{\mathcal{I}}$ is a model for *T*

$a^{\mathcal{I}} \in C^{\mathcal{I}}$ for all C(a) in *A*

$\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$ for all R(a,b) in *A*

Semantics - acyclic Tbox

Bird = Animal $\cap$ $\forall$ Skin.Feather

$\Delta^{\mathcal{I}}$ = {tweety, goofy, fea1, fur1}

Animal$^{\mathcal{I}}$ = {tweety, goofy}

Feather$^{\mathcal{I}}$ = {fea1}

Skin$^{\mathcal{I}}$ = {<tweety,fea1>, <goofy,fur1>}

→ Bird$^{\mathcal{I}}$ = {tweety}

Exercise

Create an interpretation for:

Team $\subseteq \geq 2$ hasMember
Large-Team = Team $\cap \geq 10$ hasMember
Sports-team = Team $\cap \forall$ hasMember.Athlete
Football-Team = Team $\cap \geq 11$ hasMember
$\cap \forall$ hasMember.Football-player
Football-player $\subseteq$ Athlete

Football-Team(Real_Madrid)
hasMember(Real_Madrid, Eden_Hazard)

Semantics - cyclic Tbox

$QuietPerson = Person \cap \forall$ Friend.QuietPerson

(A = F(A))

$\Delta^{\mathcal{I}}$ = {john, sue, andrea, bill}

$Person^{\mathcal{I}}$ = {john, sue, andrea, bill}

$Friend^{\mathcal{I}}$ = {<john,sue>, <andrea,bill>, <bill,bill>}

→ $QuietPerson^{\mathcal{I}}$ ={john, sue}

→ $QuietPerson^{\mathcal{I}}$ ={john, sue, andrea, bill}

Semantics - cyclic Tbox

Descriptive semantics: A = F(A) is a constraint stating that A has to be some solution for the equation.

- Not appropriate for defining concepts
- Necessary and sufficient conditions for concepts

Human = Mammal ∩ ∃ Parent
∩ ∀ Parent.Human

Semantics - cyclic Tbox

Least fixpoint semantics: A = F(A) specifies that A is to be interpreted as the smallest solution (if it exists) for the equation.

- Appropriate for inductively defining concepts

DG = EmptyDG U Non-Empty-DG

Non-Empty-DG = Node ∩ ∀ Arc.Non-Empty-DG

Human = Mammal ∩ ∃ Parent ∩ ∀ Parent.Human → Human = ⊥

Semantics - cyclic Tbox

Greatest fixpoint semantics: A = F(A) specifies that A is to be interpreted as the greatest solution (if it exists) for the equation.

- Appropriate for defining concepts whose individuals have circularly repeating structure

FoB = Blond $\cap$ $\exists$ Child.FoB

Human = Mammal $\cap$ $\exists$ Parent $\cap$ $\forall$ Parent.Human

Horse = Mammal $\cap$ $\exists$ Parent $\cap$ $\forall$ Parent.Horse

$\rightarrow$ Human = Horse

Open world vs closed world semantics

Databases: closed world reasoning

database instance represents one interpretation

→ absence of information interpreted as negative information

"complete information"

query evaluation is finite model checking

DL: open world reasoning

Abox represents many interpretations (its models)

→ absence of information is lack of information

"incomplete information"

query evaluation is logical reasoning

Open world vs closed world semantics

hasChild(Jocasta, Oedipus)

hasChild(Jocasta, Polyneikes)

hasChild(Oedipus, Polyneikes)

hasChild(Polyneikes, Thersandros)

patricide(Oedipus)

$\neg$ patricide(Thersandros) ***(not represented in DB)***

Does it follow from the Abox that

$\exists$hasChild.(patricide $\cap$ $\exists$hasChild. $\neg$ patricide)(Jocasta) ?

DL REASONING

Example

Teams have at least two members, while large teams have at least 10 members. Sports teams are teams which have only athletes as members. A football team is a team which has at least 11 members and all the members are football players. Football players are athletes. Real Madrid is a football team that has Eden Hazard as a member.

Example

$\text{Team} \subseteq \geq 2\ \text{hasMember}$
$\text{Large-Team} = \text{Team} \cap \geq 10\ \text{hasMember}$
$\text{Sports-team} = \text{Team} \cap \forall\ \text{hasMember.Athlete}$
$\text{Football-Team} = \text{Team} \cap \geq 11\ \text{hasMember}$
$\cap \forall\ \text{hasMember.Football-player}$
$\text{Football-player} \subseteq \text{Athlete}$

Football-Team(Real_Madrid)
hasMember(Real_Madrid, Eden_Hazard)

Example

Every team has at least 2 members
Every large team is a team and has at least 10 members
Every sports team is a team and has only athletes as members
Every football team is a team and has at least 11 members and has only football players as members
Every football player is an athlete

Example

Every team has at least 2 members
Every large team is a team and has at least 10 members
Every sports team is a team and has only athletes as members
Every football team is a team and has at least 11 members and has only football players as members
Every football player is an athlete

Reasoning:

Every football team is a large team
Every football team is a sports team

Example

Real Madrid is an instance of football team

Real Madrid has member Eden Hazard

Example

Reasoning:

Real Madrid is an instance of football team
Real Madrid is an instance of large team
Real Madrid is an instance of team
Real Madrid is an instance of sports team
Real Madrid has at least 11 members
All members in Real Madrid are football players
All members in Real Madrid are athletes

Real Madrid has member Eden Hazard

Eden Hazard is an instance of football player
Eden Hazard is an instance of athlete

Reasoning services

- Satisfiability of concept
- Subsumption between concepts
- Equivalence between concepts
- Disjointness of concepts

- Classification

- Instance checking
- Realization
- Retrieval
- Knowledge base consistency

Reasoning services

- Satisfiability of concept
  - C is satisfiable w.r.t. $\mathcal{T}$ if there is a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}}$ is not empty.
- Subsumption between concepts
  - C is subsumed by D w.r.t. $\mathcal{T}$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every model $\mathcal{I}$ of $\mathcal{T}$.
- Equivalence between concepts
  - C is equivalent to D w.r.t. $\mathcal{T}$ if $C^{\mathcal{I}} = D^{\mathcal{I}}$ for every model $\mathcal{I}$ of $\mathcal{T}$.
- Disjointness of concepts
  - C and D are disjoint w.r.t. $\mathcal{T}$ if $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ for every model $\mathcal{I}$ of $\mathcal{T}$.

Reasoning services

- Reduction to subsumption
  - C is unsatisfiable iff C is subsumed by $\perp$
  - C and D are equivalent iff C is subsumed by D and D is subsumed by C
  - C and D are disjoint iff C $\cap$ D is subsumed by $\perp$
- The statements also hold w.r.t. a Tbox.

Reasoning services

- Reduction to unsatisfiability
  - C is subsumed by D iff C $\cap \neg$D is unsatisfiable
  - C and D are equivalent iff

    both (C $\cap \neg$D) and (D $\cap \neg$C) are unsatisfiable
  - C and D are disjoint iff C $\cap$ D is unsatisfiable

- The statements also hold w.r.t. a Tbox.

Tableau algorithms

- To prove that C subsumes D:
  - If C subsumes D, then it is impossible for an individual to belong to D but not to C.
  - Idea: Create an individual that belongs to D and not to C and see if it causes a contradiction.
  - If **always** a contradiction (clash) then subsumption is proven. Otherwise, we have found a model that contradicts the subsumption.

Tableau algorithms

- Based on constraint systems.

  - $S = \{ x: \neg C \cap D \}$
  - Add constraints according to a set of propagation rules
  - Until clash or no constraint is applicable

Tableau algorithms – de Morgan rules

$\neg \neg C \rightarrow C$

$\neg (A \cap B) \rightarrow \neg A \cup \neg B$

$\neg (A \cup B) \rightarrow \neg A \cap \neg B$

$\neg (\forall R.C) \rightarrow \exists R.(\neg C)$

$\neg (\exists R.C) \rightarrow \forall R.(\neg C)$

Tableau algorithms – constraint propagation rules

- $S \rightarrow_{\cap} \{x{:}C_1, x{:}C_2\} \cup S$

  if x: $C_1 \cap C_2$ in S
  and either $x{:}C_1$ or $x{:}C_2$ is not in S

- $S \rightarrow_{\cup} \{x{:}D\} \cup S$

  if x: $C_1 \cup C_2$ in S and neither $x{:}C_1$ or $x{:}C_2$ is in S, and $D = C_1$ or $D = C_2$

Tableau algorithms – constraint propagation rules

- $S \rightarrow_{\forall} \{y{:}C\} \cup S$

  if x: ∀ R.C in S and xRy in S and y:C is not in S

- $S \rightarrow_{\exists} \{xRy, y{:}C\} \cup S$

  if x: ∃ R.C in S and y is a new variable and there is no z such that both xRz and z:C are in S

Example

- ST: Tournament
  $\cap \exists$ hasParticipant.Swedish
- SBT: Tournament
  $\cap \exists$ hasParticipant.(Swedish $\cap$ Belgian)

Example 1

- SBT => ST?
- S = { x:

$\neg$(Tournament $\cap$ $\exists$ hasParticipant.Swedish)

$\cap$ (Tournament

$\cap$ $\exists$ hasParticipant.(Swedish $\cap$ Belgian))

}

Example 1

- S = { x:

  ($\neg$Tournament

  $\cup$ $\forall$ hasParticipant.$\neg$ Swedish)

  $\cap$ (Tournament

  $\cap$ $\exists$ hasParticipant.(Swedish $\cap$ Belgian))

  }

Example 1

$\cap$-rule:

- S = {

  x: ($\neg$Tournament

  $\sqcup$ $\forall$ hasParticipant.$\neg$ Swedish)

  $\cap$ (Tournament

  $\cap$ $\exists$ hasParticipant.(Swedish $\cap$ Belgian)),

  x: $\neg$Tournament

  $\sqcup$ $\forall$ hasParticipant.$\neg$ Swedish,

  x: Tournament,

  x: $\exists$ hasParticipant.(Swedish $\cap$ Belgian)

  }

Example 1

∃ -rule:

- S = {
  x: (¬Tournament U ∀ hasParticipant.¬ Swedish)

  ∩ (Tournament

  ∩ ∃ hasParticipant.(Swedish ∩ Belgian)),

  x: ¬Tournament

  U ∀ hasParticipant.¬ Swedish,

  x: Tournament,

  x: ∃ hasParticipant.(Swedish ∩ Belgian),

  x hasParticipant y, y: (Swedish ∩ Belgian)

  }

Example 1

$\cap$-rule:

- S= {x: ($\neg$Tournament U $\forall$ hasParticipant.$\neg$ Swedish)

  $\cap$ (Tournament

  $\cap$ $\exists$ hasParticipant.(Swedish $\cap$ Belgian)),

  x: $\neg$Tournament U $\forall$ hasParticipant.$\neg$ Swedish,

  x: Tournament,

  x: $\exists$ hasParticipant.(Swedish $\cap$ Belgian),

  x hasParticipant y, y: (Swedish $\cap$ Belgian),

  y: Swedish, y: Belgian }

Example 1

U-rule, choice 1

- S = { x: (¬Tournament U ∀ hasParticipant.¬ Swedish) ∩ (Tournament ∩ ∃ hasParticipant.(Swedish ∩ Belgian)),
  x: ¬Tournament U ∀ hasParticipant.¬ Swedish,
  x: Tournament,
  x: ∃ hasParticipant.(Swedish ∩ Belgian),
  x hasParticipant y, y: (Swedish ∩ Belgian),
  y: Swedish, y: Belgian,
  x: ¬Tournament
  }

→ clash

Example 1

U-rule, choice 2

- S = {x: (¬Tournament U ∀ hasParticipant.¬ Swedish)
  ∩ (Tournament
  ∩ ∃ hasParticipant.(Swedish ∩ Belgian)),
  x: ¬Tournament U ∀ hasParticipant.¬ Swedish,
  x: Tournament,
  x: ∃ hasParticipant.(Swedish ∩ Belgian),
  x hasParticipant y, y: (Swedish ∩ Belgian),
  y: Swedish, y: Belgian,
  x: ∀ hasParticipant.¬ Swedish

}

Example 1

choice 2 – continued

∀-rule

- S = {

x: (¬Tournament U ∀ hasParticipant.¬ Swedish)

∩ (Tournament ∩ ∃ hasParticipant.(Swedish ∩ Belgian)),

x: ¬Tournament U ∀ hasParticipant.¬ Swedish,

x: Tournament,

x: ∃ hasParticipant.(Swedish ∩ Belgian),

x hasParticipant y, y: (Swedish ∩ Belgian),

y: Swedish, y: Belgian,

x: ∀ hasParticipant.¬ Swedish,

y: ¬ Swedish

}

→ clash

Example 2

- ST => SBT?
- S = { x:

$\neg$ (Tournament

$\cap \exists$ hasParticipant.(Swedish $\cap$ Belgian))

$\cap$ (Tournament $\cap \exists$ hasParticipant.Swedish)

}

Example 2

- S = { x:

(¬Tournament

U ∀ hasParticipant.(¬ Swedish U ¬ Belgian))

∩ (Tournament ∩ ∃ hasParticipant.Swedish)

}

Example 2

∩-rule

- S = {

x: (¬Tournament

U ∀ hasParticipant.(¬ Swedish U ¬ Belgian))

∩ (Tournament ∩ ∃ hasParticipant.Swedish),

x: (¬Tournament

U ∀ hasParticipant.(¬ Swedish U ¬ Belgian)),

x: Tournament,

x: ∃ hasParticipant.Swedish

}

Example 2

∃ -rule

- S = {
  x: (¬Tournament
  U ∀ hasParticipant.(¬ Swedish U ¬ Belgian))
  ∩ (Tournament ∩ ∃ hasParticipant.Swedish),
  x: (¬Tournament
  U ∀ hasParticipant.(¬ Swedish U ¬ Belgian)),
  x: Tournament,
  x: ∃ hasParticipant.Swedish,
  x hasParticipant y, y: Swedish
  }

Example 2

U –rule, choice 1

- S = {
  x: (¬Tournament
  U ∀ hasParticipant.(¬ Swedish U ¬ Belgian))
  ∩ (Tournament ∩ ∃ hasParticipant.Swedish),
  x: (¬Tournament
  U ∀ hasParticipant.(¬ Swedish U ¬ Belgian)),
  x: Tournament,
  x: ∃ hasParticipant.Swedish,
  x hasParticipant y, y: Swedish,
  x: ¬Tournament
  }

→ clash

Example 2

U –rule, choice 2

- S = {

  x: (¬Tournament

  U ∀ hasParticipant.(¬ Swedish U ¬ Belgian))

  ∩ (Tournament ∩ ∃ hasParticipant.Swedish),

  x: (¬Tournament

  U ∀ hasParticipant.(¬ Swedish U ¬ Belgian)),

  x: Tournament,

  x: ∃ hasParticipant.Swedish,

  x hasParticipant y, y: Swedish,

  x: ∀ hasParticipant.(¬ Swedish U ¬ Belgian)

  }

Example 2

choice 2 continued

∀–rule

- S = {

  x: (¬Tournament

  U ∀ hasParticipant.(¬ Swedish U ¬ Belgian))

  ∩ (Tournament ∩ ∃ hasParticipant.Swedish),

  x: (¬Tournament

  U ∀ hasParticipant.(¬ Swedish U ¬ Belgian)),

  x: Tournament,

  x: ∃ hasParticipant.Swedish,

  x hasParticipant y, y: Swedish,

  x: ∀ hasParticipant.(¬ Swedish U ¬ Belgian),

  y: (¬ Swedish U ¬ Belgian)

  }

Example 2

choice 2 continued

U–rule, choice 2.1

- S = {

  x: (¬Tournament
  U ∀ hasParticipant.(¬ Swedish U ¬ Belgian))
  ∩ (Tournament ∩ ∃ hasParticipant.Swedish),
  x: (¬Tournament
  U ∀ hasParticipant.(¬ Swedish U ¬ Belgian)),
  x: Tournament,
  x: ∃ hasParticipant.Swedish,
  x hasParticipant y, y: Swedish,
  x: ∀ hasParticipant.(¬ Swedish U ¬ Belgian),
  y: (¬ Swedish U ¬ Belgian),
  y: ¬ Swedish
  } → clash

Example 2

choice 2 continued

U–rule, choice 2.2

- S = {

  x: (¬Tournament
  U ∀ hasParticipant.(¬ Swedish U ¬ Belgian))
  ∩ (Tournament ∩ ∃ hasParticipant.Swedish),
  x: (¬Tournament
  U ∀ hasParticipant.(¬ Swedish U ¬ Belgian)),
  x: Tournament,
  x: ∃ hasParticipant.Swedish,
  x hasParticipant y, y: Swedish,
  x: ∀ hasParticipant.(¬ Swedish U ¬ Belgian),
  y: (¬ Swedish U ¬ Belgian),
  y: ¬ Belgian
  } → ok, model

Complexity - languages

- Overview available via the DL home page at http://dl.kr.org

Example tractable language:

A, T,⊥ , ¬A, C ∩ D, ∀R.C, ≥ n R, ≤ n R

Reasons for intractability:

choices, e.g. C U D

exponential size models,

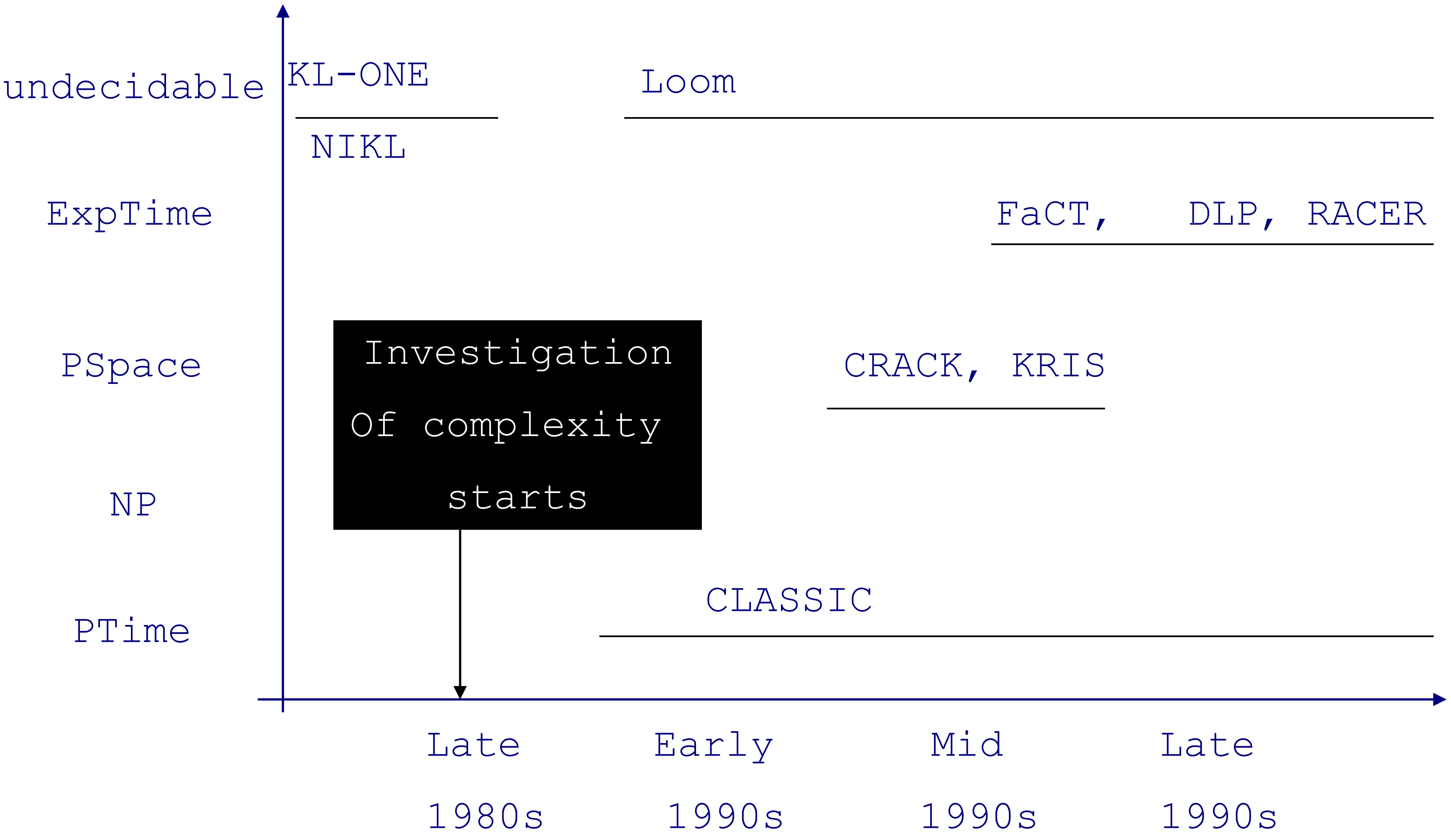
e.g interplay universal and existential quantification

Reasons for undecidability:

e.g. role-value maps R=S

DL SYSTEMS

Systems


undecidable
KL-ONE
NIKL
Loom
ExpTime
FaCT, DLP, RACER
PSpace
Investigation
Of complexity
starts
CRACK, KRIS
NP
PTime
CLASSIC
Late 1980s
Early 1990s
Mid 1990s
Late 1990s

Systems

- Overview available via the DL home page at http://dl.kr.org
- Current systems include: CEL, Cerebra Enginer, FaCT++, fuzzyDL, HermiT, KAON2, MSPASS, Pellet, QuOnto, RacerPro, SHER

Extensions

- Time
- Defaults
- Part-of
- Knowledge and belief
- Uncertainty (fuzzy, probabilistic)

DL AND THE WEB

DAML+OIL Class Constructors

Constructor	DL Syntax	Example
intersectionOf	$C_1 \sqcap \ldots \sqcap C_n$	Human $\sqcap$ Male
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	Doctor $\sqcup$ Lawyer
complementOf	$\neg C$	$\neg$Male
oneOf	$\{x_1 \ldots x_n\}$	{john, mary}
toClass	$\forall P.C$	$\forall$hasChild.Doctor
hasClass	$\exists P.C$	$\exists$hasChild.Lawyer
hasValue	$\exists P.\{x\}$	$\exists$citizenOf.{USA}
minCardinalityQ	$\geqslant n P.C$	$\geqslant$2hasChild.Lawyer
maxCardinalityQ	$\leqslant n P.C$	$\leqslant$1hasChild.Male
cardinalityQ	$= n\ P.C$	$=$1 hasParent.Female

- XMLS **datatypes** as well as classes
- Arbitrarily complex **nesting** of constructors
  - E.g., Person $\sqcap$ $\forall$hasChild.(Doctor $\sqcup$ $\exists$hasChild.Doctor)

DAML+OIL Axioms

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
sameClassAs	$C_1 \equiv C_2$	Man $\equiv$ Human $\sqcap$ Male
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
samePropertyAs	$P_1 \equiv P_2$	cost $\equiv$ price
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	{President_Bush} $\equiv$ {G_W_Bush}
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq$ $\neg$Female
differentIndividualFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} $\sqsubseteq$ $\neg${peter}
inverseOf	$P_1 \equiv P_2^-$	hasChild $\equiv$ hasParent$^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor$^+$ $\sqsubseteq$ ancestor
uniqueProperty	$\top \sqsubseteq \leqslant 1 P$	$\top \sqsubseteq \leqslant 1$hasMother
unambiguousProperty	$\top \sqsubseteq \leqslant 1 P^-$	$\top \sqsubseteq \leqslant 1$isMotherOf$^-$

☞ Axioms (mostly) **reducible to subClass/PropertyOf**

OWL

- OWL-Lite, OWL-DL, OWL-Full: increasing expressivity
- A legal OWL-Lite ontology is a legal OWL-DL ontology is a legal OWL-Full ontology
- OWL-DL: expressive description logic, decidable
- XML-based
- RDF-based (OWL-Full is extension of RDF, OWL-Lite and OWL-DL are extensions of a restriction of RDF)

OWL-Lite

- **Class**, subClassOf, equivalentClass
- intersectionOf (only named classes and restrictions)
- **Property**, subPropertyOf, equivalentProperty
- domain, range (global restrictions)
- inverseOf, TransitiveProperty (*), SymmetricProperty, FunctionalProperty, InverseFunctionalProperty
- allValuesFrom, someValuesFrom (local restrictions)
- minCardinality, maxCardinality (only 0/1)
- **Individual**, sameAs, differentFrom, AllDifferent

(*) restricted

OWL-DL

- **Type separation (**class cannot also be individual or property, property cannot be also class or individual), Separation between DatatypeProperties and ObjectProperties
- **Class –complex classes**, subClassOf, equivalentClass, *disjointWith*
- intersectionOf, *unionOf, complementOf*
- **Property**, subPropertyOf, equivalentProperty
- domain, range (global restrictions)
- inverseOf, TransitiveProperty (*), SymmetricProperty, FunctionalProperty, InverseFunctionalProperty
- allValuesFrom, someValuesFrom (local restrictions), *oneOf, hasValue*
- minCardinality, maxCardinality
- **Individual**, sameAs, differentFrom, AllDifferent

(*) restricted

OWL2

- OWL2 Full and OWL2 DL
- OWL2 DL compatible with SROIQ

- Punning
  - IRI may denote both class and individual
  - For reasoning they are considered separate entities

OWL2 profiles

- OWL2 EL (based on EL++)
  - Essentially intersection and existential quantification
  - SNOMED CT, NCI Thesaurus
- OWL2 QL (“query language”)
  - Can be realized using relational database technology
  - RDFS + small extensions
- OWL2 RL (“rule language”)

References

- Baader, Calvanese, McGuinness, Nardi, Patel-Schneider. *The Description Logic Handbook*. Cambridge University Press, 2003.
- Donini, Lenzerini, Nardi, Schaerf, Reasoning in description logics. *Principles of knowledge representation*. CSLI publications. pp 191-236. 1996.
- dl.kr.org

References

- https://www.w3.org/TR/2004/REC-owl-features-20040210/
- https://www.w3.org/TR/2012/REC-owl2-quick-reference-20121211/
- https://www.w3.org/TR/2012/REC-owl2-primer-20121211/