A Description Logic Journey

Franz Baader

Theoretical Computer Science

TU Dresden, Germany

Navy Art Collection

TECHNISCHE UNIVERSITÄT DRESDEN

DFG Deutsche Forschungsgemeinschaft

Knowledge Representation

general goal

"develop formalisms for providing high-level descriptions of the world that can be effectively used to build intelligent applications"

[Brachman & Nardi, 2003]

- formalism: well-defined syntax and formal, unambiguous semantics
- high-level description: only relevant aspects represented, others left out
- intelligent applications: must be able to reason about the knowledge, and infer implicit knowledge from the explicitly represented knowledge
- effectively used: need for practical reasoning tools and efficient implementations

Dresden

Description Logics

- Family of logic-based knowledge representation languages tailored towards representing terminological knowledge
- Many DLs are decidable fragments of first-order logic
- Close relationship to propositional modal logics
- Design goal: good compromise between expressiveness and complexity
- Decidability and complexity results for a great variety of DLs and various inference problems, but also implementation of practical systems
  - very expressive DLs of high worst-case complexity, but with highly optimized "practical" reasoning procedures — *FaCT, Racer, Pellet, HermiT, ..., Konclude, MORe*
  - inexpressive DLs with tractable inference problems, which are expressive enough for certain applications — *CEL, Snorocket, ELK, QuOnto, Mastro, ontop*
- Applications: natural language processing, configuration, databases, modelling in engineering domains, ontologies (Web ontology language OWL, biomedical ontologies)

Dresden

Description Logics

from a general point of view

Concepts

- Constructors for building complex concept descriptions out of atomic concepts (unary predicates) and roles (binary predicates).
- Interpretation $\mathcal{I}$ assigns sets $C^{\mathcal{I}}$ to concept descriptions C according to the semantics of the constructors.

Ontology

TBoxes

- Finite set of general concept inclusions (GCIs) of the form $C \sqsubseteq D$ where C, D are concept descriptions.
- The interpretation $\mathcal{I}$ is a model of a TBox $\mathcal{T}$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for all GCIs $C \sqsubseteq D$ in $\mathcal{T}$.

ABoxes

- Finite set of assertions of the form $C(a)$ and $r(a, b)$ where C is a concept description, r a role, and a, b individual names.
- The interpretation $\mathcal{I}$ is a model of an ABox $\mathcal{A}$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$ holds for all assertions $C(a)$ and $r(a, b)$ in $\mathcal{A}$.

Dresden

Description Logics

from a general point of view

Concepts

- Constructors for building complex concept descriptions out of atomic concepts (unary predicates) and roles (binary predicates).
- Interpretation $\mathcal{I}$ assigns sets $C^{\mathcal{I}}$ to concept descriptions C according to the semantics of the constructors.

Ontology

Restricted TBoxes

- Finite set of general concept inclusions (GCIs) of the form $A \equiv D$ where A is a concept name occurring only once as left-hand side.
- The interpretation $\mathcal{I}$ is a model of a retricted TBox $\mathcal{T}$ if $A^{\mathcal{I}} = D^{\mathcal{I}}$ holds for all definitions $A \equiv D$ in $\mathcal{T}$.

ABoxes

- Finite set of assertions of the form $C(a)$ and $r(a, b)$ where C is a concept description, r a role, and a, b individual names.
- The interpretation $\mathcal{I}$ is a model of an ABox $\mathcal{A}$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$ holds for all assertions $C(a)$ and $r(a, b)$ in $\mathcal{A}$.

Constructors of the DL $\mathcal{ALC}$

top concept $\top$, negation $\neg C$
conjunction $C \sqcap D$, disjunction $C \sqcup D$,
existential restriction $\exists r.C$,
value restriction $\forall r.C$

An advanced course that	$\mathit{Course} \sqcap \mathit{Advanced} \sqcap$
has a smart or studious student,	$\exists \mathit{has\_student}.(\mathit{Smart} \sqcup \mathit{Studious}) \sqcap$
no easy topic,	$\forall \mathit{has\_topic}.\neg \mathit{Easy} \sqcap$
and a teacher	$\exists \mathit{has\_teacher}.\top$

TBox

Concept definition

$\mathit{Good\_course} \equiv \mathit{Course} \sqcap \ldots$

General concept inclusion (GCI)

$\exists \mathit{has\_student}.\top \sqsubseteq \mathit{Course}$

ABox

properties of `individuals`

$\mathit{Good\_Course}(\texttt{Course123})$
$\mathit{has\_teacher}(\texttt{Course123}, \texttt{Franz})$
$\mathit{has\_topic}(\texttt{Course123}, \texttt{DL})$

Dresden

Constructors of the DL $\mathcal{ALC}$

top concept $\top$, negation $\neg C$
conjunction $C \sqcap D$, disjunction $C \sqcup D$,
existential restriction $\exists r.C$,
value restriction $\forall r.C$

An advanced course that	$Course \sqcap Advanced \sqcap$
has a smart or studious student,	$\exists has\_student.(Smart \sqcup Studious) \sqcap$
no easy topic,	$\forall has\_topic.\neg Easy \sqcap$
and a teacher	$\exists has\_teacher.\top$

The semantics of the constructors is defined through identities:

- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$,
- $(\exists r.C)^{\mathcal{I}} = \{d \mid \exists e.(d,e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$,
- $(\forall r.C)^{\mathcal{I}} = \{d \mid \forall e.(d,e) \in r^{\mathcal{I}} \rightarrow e \in C^{\mathcal{I}}\}$,
- $\ldots$

Reasoning

makes implicitly represented knowledge explicit,
provided as service by the DL system, e.g.:

polynomial reductions

Subsumption: Is C a subconcept of D?

$C \sqsubseteq_{\mathcal{T}} D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of the TBox $\mathcal{T}$.

Satisfiability: Is the concept C non-contradictory?

C is satisfiable w.r.t. $\mathcal{T}$ iff $C^{\mathcal{I}} \neq \emptyset$ for some model $\mathcal{I}$ of $\mathcal{T}$.

Consistency: Is the ABox $\mathcal{A}$ non-contradictory?

$\mathcal{A}$ is consistent w.r.t. $\mathcal{T}$ iff it has a model that is also a model of $\mathcal{T}$.

Instantiation: Is e an instance of C?

$\mathcal{A} \models_{\mathcal{T}} C(e)$ iff $e^{\mathcal{I}} \in C^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$.

in presence of negation

Focus of DL research

develop and investigate reasoning procedures

Expressivity of language

- application relvant concepts must be definable
- some domains require very expressive DLs

Complexity of reasoning

- reasoning must be decidable/of low complexity
- requires restricted description languages

Practical algorithms

- easy to implement and optimize
- behave well in practice

Dresden

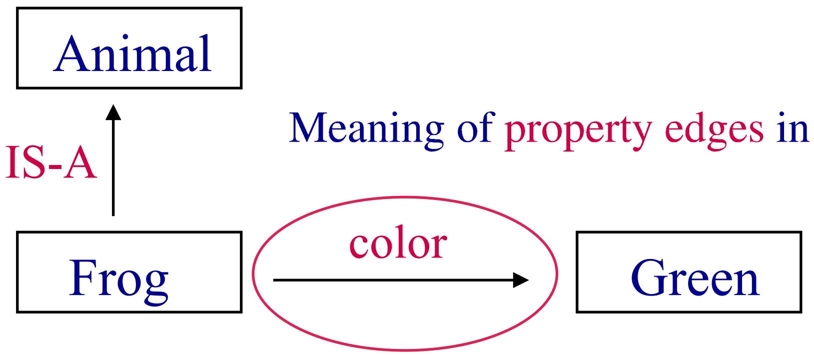
A Description Logic Journey

Starting small

The Description Logic $\mathcal{FL}_0$

$C \sqcap D, \ \forall r.C, \ \top$

In the early days of DL research, $\mathcal{FL}_0$ was considered to be the smallest possible DL.



Meaning of property edges in semantic networks?

- value restriction: green is the only possible color;

  $Frog \sqsubseteq Animal \sqcap \forall color.Green$

- existential restriction: green is one of its colors.

  $Frog \sqsubseteq Animal \sqcap \exists color.Green$

Source: Wikimedia Author: LiquidGhoul

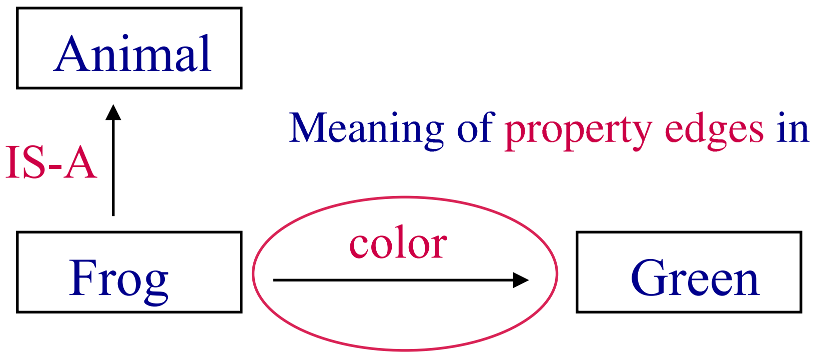
Source: Wikimedia Author: Carey James Balboa

Dresden

The Description Logic $\mathcal{FL}_0$

$C \sqcap D, \ \forall r.C, \ \top$

In the early days of DL research, $\mathcal{FL}_0$ was considered to be the smallest possible DL.



Meaning of property edges in semantic networks?

- value restriction: green is the only possible color;

$\mathcal{FL}_0$ $\quad Frog \sqsubseteq Animal \sqcap \forall color.Green$

Chosen by KL-ONE and other early DL systems

- existential restriction: green is one of its colors.

$$Frog \sqsubseteq Animal \sqcap \exists color.Green$$

Dresden

Some bad news

in the late 1980ies

Source: Wikimedia Author: Paul Fürst

A commonly held belief in the 1980ies:

reasoning in KR systems should be tractable, i.e., of polynomial time complexity

- KL-ONE and its early successor systems (BACK, MESON, K-Rep, ...) employed polynomial-time algorithms ← sound, but incomplete

- reasoning in KL-ONE is undecidable

[Schmidt-Schauß; 1989]

- reasoning w.r.t. a TBox is intractable even in the minimal DL $\mathcal{FL}_0$

[Nebel; 1989]

Complexity

of subsumption reasoning $C \sqsubseteq D$

restricted TBox

	no TBox	acyclic TBox	cyclic TBox	general TBox
$\mathcal{FL}_0$	P [Brachman& Levesque; 1987]	coNP [Nebel; 1989]	PSpace [Baader; 1990]	ExpTime [Baader et al.; 2005] [Hofmann; 2005]
$\mathcal{ALC}$	PSpace [Schmidt-Schauß& Smolka; 1988]	PSpace [Lutz; 1999]	ExpTime [Schild; 1991]	ExpTime [Schild; 1991]

$$P \subsetneq coNP \subseteq PSpace \subsetneq ExpTime$$

P	coNP ⊆ PSpace	ExpTime
tractable	not tractable *unless* $P \neq NP$	not tractable

⇒ *no reason for restricting to* $\mathcal{FL}_0$

Dresden

A Description Logic Journey

Getting larger

$\mathcal{ALC}$ and beyond

Extensions of $\mathcal{ALC}$ motivated by applications

mechanical engineering

- Concrete domains: refer to concrete objects (e.g., numbers) and predicates on these objects (e.g., numerical comparisons) when defining concepts:

$$Teenager \equiv Human \sqcap (age \geq 10) \sqcap (age \leq 19)$$

$$Human \sqsubseteq (mother \circ age > age) \sqcap (father \circ age > age)$$

[Baader & Hanschke; 1991, 1992] [Lutz; 2002] [Lutz & Milicic; 2007]

Dresden

Extensions of $\mathcal{ALC}$ motivated by applications

- Concrete domains: refer to concrete objects (e.g., numbers) and predicates on these objects (e.g., numerical comparisons) when defining concepts.

mechanical engineering

- Local and global cardinality constraints: restrict the number of role successors of an object (number restrictions) or the cardinality of a concept:

configuration

At most two sons and at least one daughter:
$(\leq 2\, child.Male) \sqcap (\geq 1\, child.Female)$

[Hollunder & Baader; 1991]

[Tobies; 2000]

At most 45 million cars are registered all over Germany:
$(\leqslant 45000000\, (Car \sqcap \exists registered\_in.German\_district))$

[Baader et al.; 1996]

Dresden

Extensions of $\mathcal{ALC}$ motivated by applications

- Concrete domains: refer to concrete objects (e.g., numbers) and predicates on these objects (e.g., numerical comparisons) when defining concepts.

  mechanical engineering

- Local and global cardinality constraints: restrict the number of role successors of an object (number restrictions) or the cardinality of a concept.

  configuration

- Transitive roles, subroles, and inverse roles: describe complex objects that are composed of different parts:

  $$Engine \sqcap \exists part\_of.Car \sqcap \exists has\_part.Distributor \sqcap \ldots$$

  The role has_part is transitive, the inverse of $part\_of$, and has has_strict_part as a subrole.

  [Sattler; 1996] [Horrocks & Sattler; 1999]

  chemical process engineering

Dresden

Extensions of $\mathcal{ALC}$ motivated by applications

- Concrete domains: refer to concrete objects (e.g., numbers) and predicates on these objects (e.g., numerical comparisons) when defining concepts.

mechanical engineering

- Local and global cardinality constraints: restrict the number of role successors of an object (number restrictions) or the cardinality of a concept.

configuration

- Transitive roles, subroles, and inverse roles: describe complex objects that are composed of different parts.

chemical process engineering

- These and some additional features are available in the Web Ontology Language OWL 2 DL.

  Highly optimized reasoning systems: *FaCT, Racer Pellet, HermiT, Sequoia, . . .*

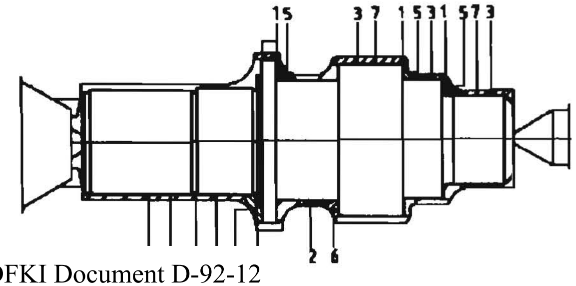
Dresden

Concrete domains

our original motivation

[Baader & Hanschke; 1991]

describe geometric structure of
rotational-symmetric lathe workpieces



DFKI Document D-92-12

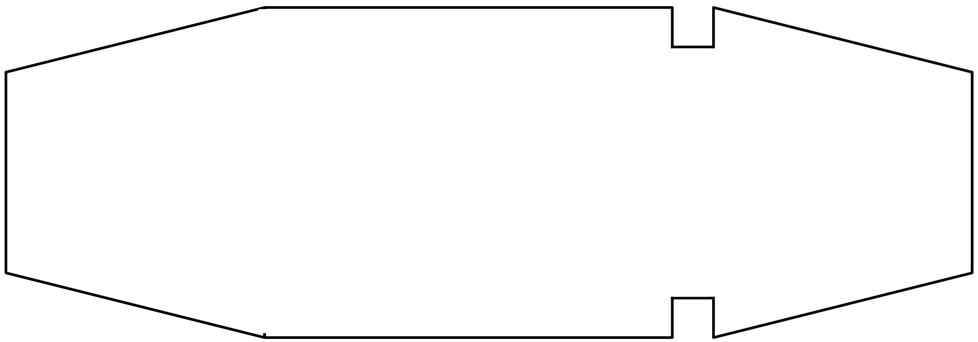
Source: Wikimedia Author: Greudin

Concrete domains

our original motivation

[Baader & Hanschke; 1991]

describe geometric structure of
rotational-symmetric lathe workpieces



- decompose workpiece into simple geometric components

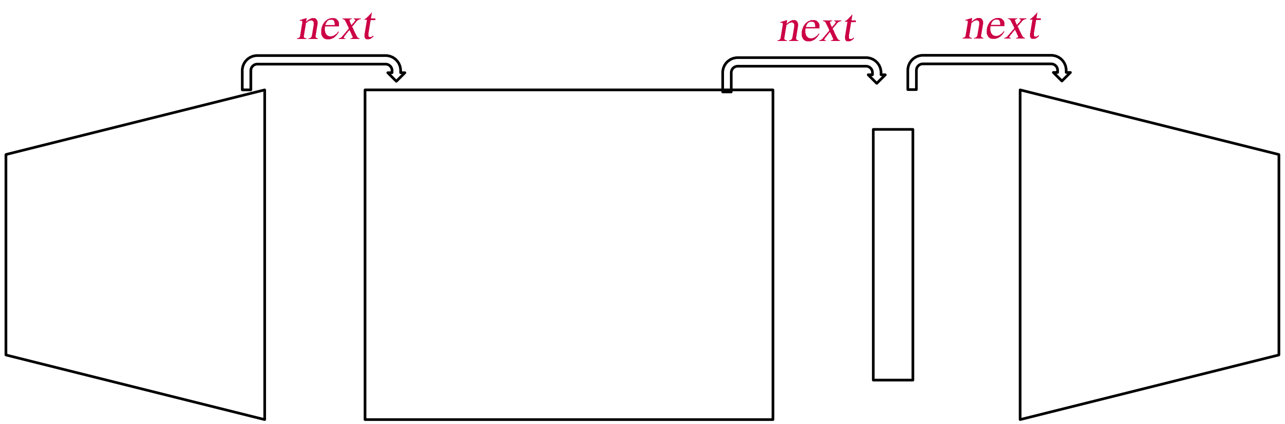
Dresden

Concrete domains

our original motivation

[Baader & Hanschke; 1991]

describe geometric structure of
rotational-symmetric lathe workpieces



- decompose workpiece into simple geometric components
- described geometric shape of single components and how neighbouring components fit together using concrete domain predicates *real arithmetics*
- described whole workpiece as sequence of its components using transitive closure

Dresden

Good news

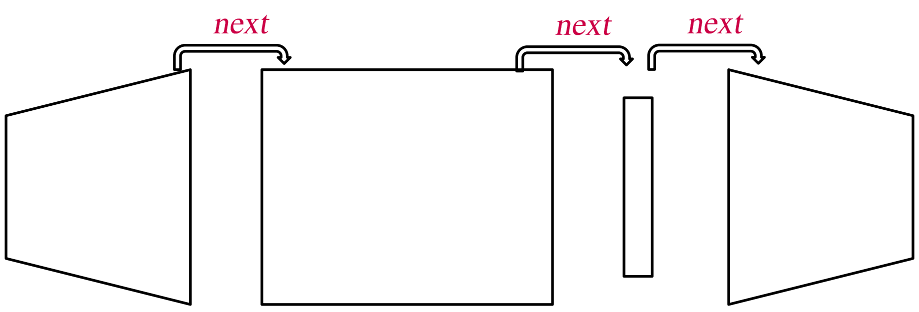
satisfiability of constraints in the concrete domain decidable

- extending $\mathcal{ALC}$ with "admissible" concrete domains leaves reasoning (without TBox or w.r.t. an acyclic TBox) decidable [Baader & Hanschke; 1991]
- extending $\mathcal{ALC}$ with transitive closure of roles leaves reasoning decidable [Baader; 1991]

Bad news

[Baader & Hanschke; 1992]

- combining the two extensions causes undecidability



More bad news

[Lutz; 2001]

- adding an acyclic TBox to $\mathcal{ALC}$ with an admissible concrete domain may increase the complexity considerably
- adding a cyclic or general TBox may cause undecidability
- even for quite simple admissible concrete domains

Some good news

combination of several
rather complex conditions

- extending $\mathcal{ALC}$ with ω-admissible concrete domains leaves reasoning decidable even in the presence of general TBoxes

  [Lutz; 2002] [Lutz & Milicic; 2007]

- model-theoretic chracterizations of ω-admissible concrete domains that facilitate finding new ω-admissible concrete domains

  [Baader & Rydval; 2020]

Dresden

A Description Logic Journey

Getting smaller again

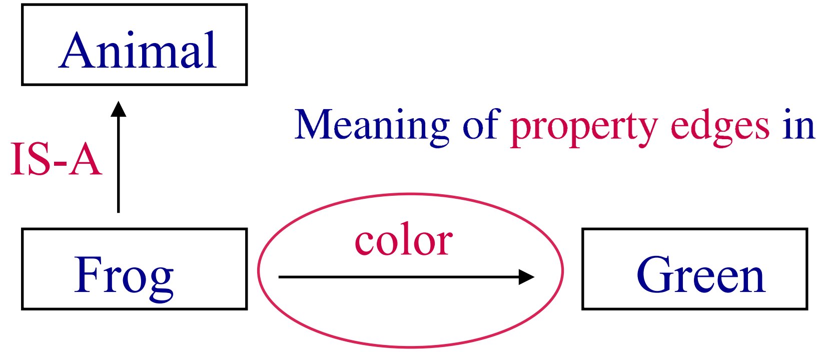
$\mathcal{EL}$

The Description Logic $\mathcal{EL}$

$C \sqcap D, \quad \exists r.C, \quad \top$

In the early days of DL research, $\mathcal{FL}_0$ was considered to be the smallest possible DL.

Meaning of property edges in semantic networks?



- value restriction: green is the only possible color;

$\mathcal{FL}_0$ $\quad Frog \sqsubseteq Animal \sqcap \forall color.Green$

- existential restriction: green is one of its colors.

$\mathcal{EL}$ $\quad Frog \sqsubseteq Animal \sqcap \exists color.Green$

Chosen by KL-ONE and other early DL systems

Source: Wikimedia Author: Carey James Balboa

Complexity

of subsumption reasoning $C \sqsubseteq D$

restricted TBox

	no TBox	acyclic TBox	cyclic TBox	general TBox
$\mathcal{FL}_0$	P	coNP	PSpace	ExpTime
$\mathcal{ALC}$	PSpace	PSpace	ExpTime	ExpTime
$\mathcal{EL}$	P [Baader, Küsters Molitor; 1999]	P [Baader; 2003]	P [Baader; 2003]	P [Brandt; 2004]

Is $\mathcal{EL}$ useful in practice?

Dresden

$\mathcal{EL}$

a practically useful DL for which
TBox and ABox reasoning is tractable

- subsumption in $\mathcal{EL}$ is polynomial
  even in the presence of general TBoxes;

- $\mathcal{EL}$ is used in several biomedical ontologies
  (e.g., SNOMED CT, The Gene Ontology);

- tractability even holds for ABox reasoning and for interesting extensions of $\mathcal{EL}$;
  [Brandt; 2004]
  [Suntisrivaraporn; 2009]
  [Baader, Brandt, Lutz; 2005]

- a maximal such extension is the basis of the OWL 2 EL profile
  of the Web Ontology Language standard of the W3C;

- availability of several highly efficient $\mathcal{EL}$ reasoners
  CEL
  Snorocket, ELK, Konclude

A Description Logic Journey

Making in larger again

Horn DLs and beyond

Horn DLs and beyond

- The algorithmic approach (consequence-based reasoning) employed for $\mathcal{EL}$ can be extended to non-tractable DLs that share a certain model-theoretic property with $\mathcal{EL}$

  Horn $\approx$ existence of canonical models

Horn DLs and beyond

- The algorithmic approach (consequence-based reasoning) employed for $\mathcal{EL}$ can be extended to non-tractable DLs that share a certain model-theoretic property with $\mathcal{EL}$

$\mathcal{ELI}$ extension of $\mathcal{EL}$ with inverse roles

  - The subsumption problem in $\mathcal{ELI}$ is ExpTime-complete.
    [Baader, Brandt, Lutz; 2008]
  - The consequence-based reasoner CB is much faster on $\mathcal{ELI}$ than highly-optimized reasoners for more expressive DLs
    [Kazakov; 2009]

- This approach has recently been extended also to non-Horn DLs up to OWL 2 DL

  Consequence-based reasoner Sequoia [Cucala, Grau, Horrocks; 2021]

Dresden

End of Journey

- Great variety of DLs with different expressive power
- Formal and algorithmic properties well investigated
- Highly optimized reasoning system
- Formal basis of OWL 2 standard
- Employed in diverse application domains