

Description Logics for Knowledge Representation

DLs are a family of knowledge-representation formalisms

- **object-centered, roles and features (binary relations)**
- **necessary vs. sufficient attributes**
- **inference services**
 - subsumption check
 - consistency check
 - classification
 - abstraction
 - default reasoning
 - spatial and temporal reasoning
- **guaranteed correctness, completeness, decidability and complexity properties**
- **highly optimized implementations (e.g. RACER)**

1

Development of Description Logics

There exist several commercial and experimental developments of DLs, among them

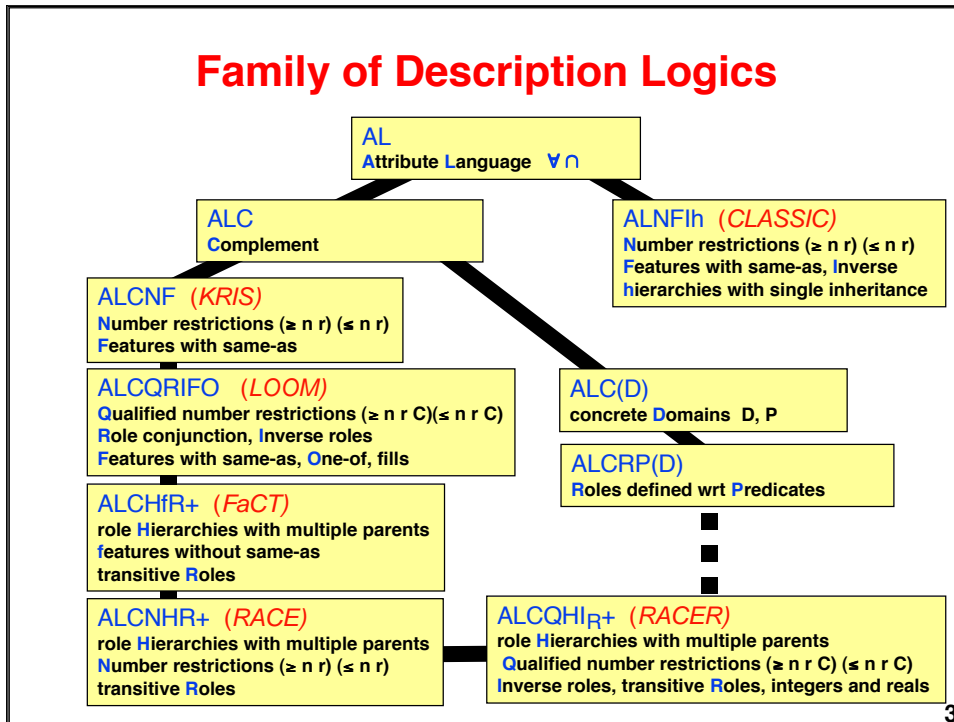
- KL-ONE first conception of a DL (1985)
- CLASSIC commercial implementation by AT&T
- LOOM experimental system at USC
- FaCT experimental and commercial system (Horrocks, Manchester)
- RACER experimental system in Hamburg and Montreal (Haarslev & Moeller)

There is active research on DLs:

- **extending the expressivity of concept languages**
- **decidability and tractability of inference services**
- **integration of predicates over concrete domains (e.g. numbers)**
- **highly optimized implementations**
- **developing new inference services (e.g. for scene interpretation)**

2

Family of Description Logics



3

RACER Concept Language

<p>C concept term CN concept name R role term RN role name</p> <p>C -> CN *top* *bottom* (not C) (and C1 ... Cn) (or C1 ... Cn) (some R C) (all R C) (at-least n R) (at-most n R) (exactly n R) (at-least n R C) (at-most n R C) (exactly n R C) CDC</p>	<p><i>concept definition</i> (equivalent CN C)</p> <p><i>concept axioms</i> (implies CN C) (implies C1 C2) (equivalent C1 C2) (disjoint C1 ... Cn)</p> <p><i>roles</i> R -> RN (inv RN)</p>	<p><i>concrete-domain concepts</i> AN attribute name</p> <p>CDC -> (a AN) (an AN) (no AN) (min AN integer) (max AN integer) (> aexpr aexpr) (>= aexpr aexpr) (< aexpr aexpr) (<= aexpr aexpr) (= aexpr aexpr)</p> <p>aexpr -> AN real (+ aexpr1 aexpr1*) aexpr1</p> <p>aexpr1 -> real AN (* real AN)</p>
---	---	--

4

Primitive and Defined Concepts

Concept expressions of a DL describe sets of entities within terms of properties (unary relations) and the roles (binary relations).

The main building blocks are primitive oder defined concepts.

Primitive concepts: concept \Rightarrow satisfied properties and relations

satisfied properties and relations are necessary conditions for an object to belong to a class

Defined concepts: concept \Leftrightarrow satisfied properties and relations

satisfied properties and relations are necessary and sufficient conditions for an object to belong to a classt

Primitive concept "person":
 (implies person (and human (some has-gender (or female male))))

Defined concept "parent":
 (equivalent parent (and person (some has-child person)))

5

Example of a TBox

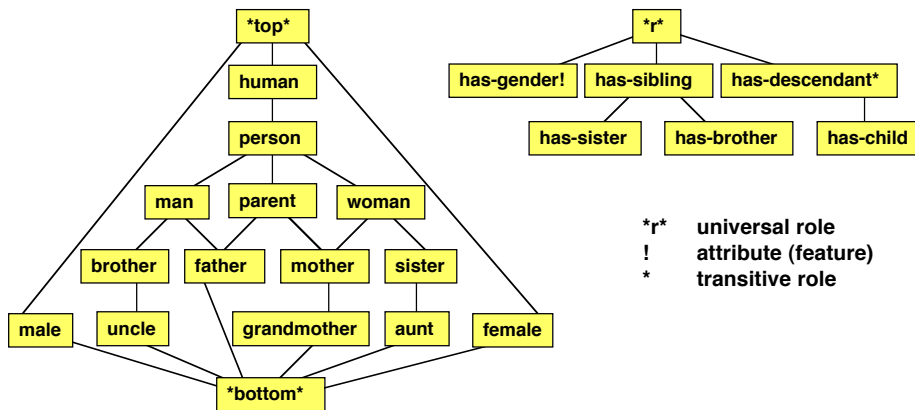
(signature	:atomic-concepts (person human female male woman man parent mother father grandmother aunt uncle sister brother)	
:roles	((has-child :parent has-descendant) (has-descendant :transitive t) (has-sibling) (has-sister :parent has-sibling) (has-brother :parent has-sibling) (has-gender :feature t))	Signature of TBox

	(implies *top* (all has-child person)) (implies (some has-child *top*) parent) (implies (some has-sibling *top*) (or brother sister)) (implies *top* (all has-sibling (or sister brother))) (implies *top* (all has-sister (some has-gender female))) (implies *top* (all has-brother (some has-gender male)))	domain and range restrictions for roles
--	---	--

	(implies person (and human (some has-gender (or female male)))) (disjoint female male) (implies woman (and person (some has-gender female))) (implies man (and person (some has-gender male))) (equivalent parent (and person (some has-child person))) (equivalent mother (and woman parent)) (equivalent father (and man parent)) (equivalent grandmother (and mother (some has-child (some has-child person)))) (equivalent aunt (and woman (some has-sibling parent))) (equivalent uncle (and man (some has-sibling parent))) (equivalent brother (and man (some has-sibling person))) (equivalent sister (and woman (some has-sibling person)))	concepts
--	---	-----------------

3

Concept and Role Hierarchies Implied by TBox



7

TBox Inferences

A DL system offers several inference services. At the core is a consistency test:

$$C \stackrel{?}{\models} \text{*bottom*} \text{ (the empty concept)}$$

Example: $(\text{and} (\text{at-least } 1 \text{ has-child}) (\text{at-most } 0 \text{ has-child})) \models \text{*bottom*}$

Consistency checking is the basis for several other inference services:

- **subsumption**
 $(\text{implies } C1 \ C2) \iff (\text{and } C1 \ (\text{not } C2)) \models \text{*bottom*}$
- **classification of a concept expression**
 searches the existing concept hierarchy for the most special concept which subsumes the concept expression

8

ABox of a Description Logic System

TBox = terminological knowledge (concepts and roles)

ABox = assertional knowledge (facts)

An ABBox contains:

- concept assertions (instance IN C)
individual IN is instance of a concept expression C
- role assertions (related IN_1 IN_2 RN)
individual IN_1 is related to IN_2 by role RN
- An ABBox always refers to a particular TBox.
- An ABBox requires unique names
- ABBox facts are assumed to be incomplete (OWA).
 - OWA = Open World Assumption
(new facts may be added, hence inferences are restricted)
 - CWA = Closed World Assumption
(no facts may be added)

9

ABox Inferences

ABox inferences = inferring facts about ABBox individuals

Typical queries:

- consistency *is ABBox consistent?*
- retrieval *which individuals satisfy a concept expression?*
- classification *what are the most special concept names which describe an individual?*

ABox consistency checking is in general more complicated than TBox consistency checking

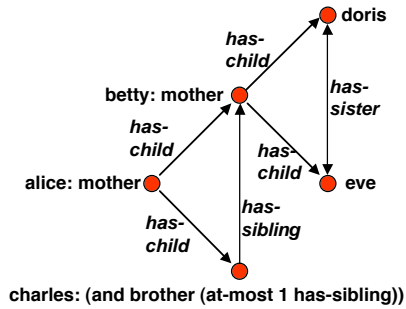
ABox consistent \Leftrightarrow there exists a "model" for ABBox and TBox

All ABBox inferences are based on the ABBox consistency check.

10

Example of ABox Queries

Contents of ABox
 (instance alice mother)
 (related alice betty has-child)
 (related alice charles has-child)
 (instance betty mother)
 (related betty doris has-child)
 (related betty eve has-child)
 (instance charles brother)
 (related charles betty has-sibling)
 (instance charles (at-most 1 has-sibling))
 (related doris eve has-sister)
 (related eve doris has-sister)



Questions and answers

- | | |
|--|---|
| (individual-instance? doris woman)
T | Is doris <i>instance of the concept</i> woman? |
| (individual-types eve)
((sister) (woman) (person) (human) (*top*)) | Of which <i>concept names</i> is eve an instance? |
| (individual-fillers alice has-descendant)
(doris eve charles betty) | What are the <i>descendants</i> of eve? |
| (concept-instances sister)
(doris betty eve) | Which <i>instances</i> has the concept sister? |

11

Abstraction with Description Logics

Abstraction = omission of properties or relations, extending a concept, generalization

Examples:

- **Superordinate concept name of a concept expression (= concept classification)**
 (and person (some has-size tall)) → person
- **Generalization of concept expressions**
 (and (some has-occupation professor) (at-least 3 has-child))
↓
 (and (some has-occupation civil-servant) (at-least 1 has-child))
- **Concept expression which subsumes several individuals**
 1. classify individuals
 2. determine least common subsumer (LCS)
 - for RACER: trivial solution in terms of (OR C₁ ... C_n)
 - for DLs without OR: special abstraction operator LCS

12