Description Logics

1

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)**

2

Development of Description Logics

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

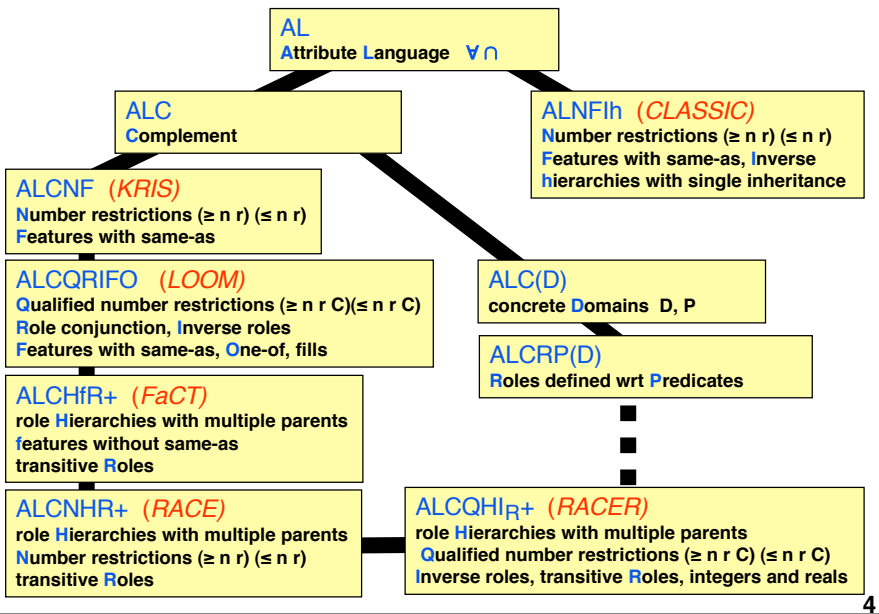
- KL-ONE — first conception of a DL (1985)
- CLASSIC — commercial implementation by AT&T
- LOOM — experimental system at USC
- FaCT — experimental and commercial system (Horrocks, Manchester)
- RACER — experimental system in Hamburg and Montreal (Haarslev & Moeller)

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)**

3

Family of Description Logics



4

RACER Concept Language

C concept term
CN concept name
R role term
RN role name

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

concept definition
(equivalent *CN C*)

concept axioms
(implies *CN C*)
(implies *C1 C2*)
(equivalent *C1 C2*)
(disjoint *C1 ... Cn*)

roles
R -> *RN*
(inv *RN*)

concrete-domain concepts
AN *attribute name*
CDC -> (a *AN*)
(an *AN*)
(no *AN*)
(min *AN integer*)
(max *AN integer*)
(> *aexpr aexpr*)
(>= *aexpr aexpr*)
(< *aexpr aexpr*)
(<= *aexpr aexpr*)
(= *aexpr aexpr*)
aexpr -> *AN*
real
(+ *aexpr1 aexpr1**)
aexpr1
aexpr1 -> *real*
AN
(* *real AN*)

5

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 => satisfied properties and relations**

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

Defined concepts: **concept <=> 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)))

6

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)))
```

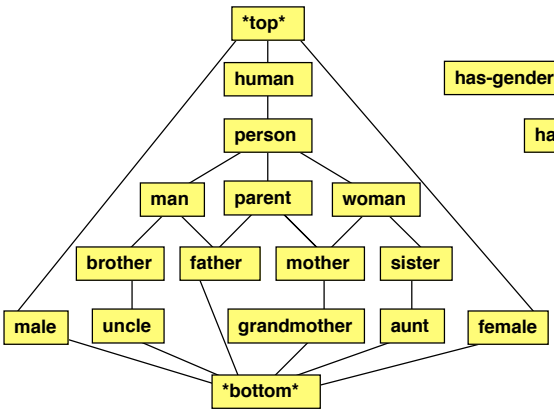
domain and range restrictions for roles

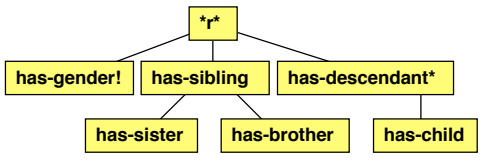
```
(implies person (and human (some has-gender (or female male))))
(disjoint female male)
(implies woman (and person (some has-gender female)))
(implies man (and person (some has-gender male)))
(equivalent parent (and person (some has-child person)))
(equivalent mother (and woman parent))
(equivalent 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

7

Concept and Role Hierachies Implied by TBox





r universal role
! attribute (feature)
* transitive role

8

TBox Inferences

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

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

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

Consistency checking is the basis for several other inference services:

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

9

ABox of a Description Logic System

TBox = terminological knowledge (concepts and roles)
ABox = assertional knowledge (facts)

An ABox 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 ABox always refers to a particular TBox.**
- **An ABox requires unique names**
- **ABox 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)

10

ABox Inferences

ABox inferences = inferring facts about ABox individuals

Typical queries:

- **consistency** — ***is ABox 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 <=> there exists a "model" for ABox and TBox

All ABox inferences are based on the ABox consistency check.

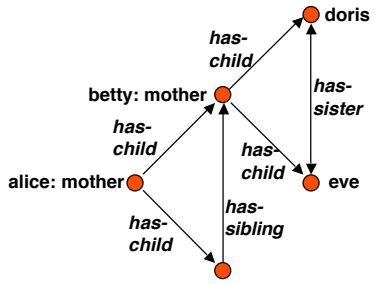
Example of ABox Queries

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)



charles: (and brother (at-most 1 has-sibling))

Questions and answers

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

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 C_1 ... C_n)**
  - **for DLs without OR: special abstraction operator LCS**

13