Description Logic

Lin Zuoquan
Information Science Department
Peking University
lz@is.pku.edu.cn
http://www.is.pku.edu.cn/~lz/teaching/stm/saswws.html

Outline

- Introduction
- Description Logic
- Algorithm
- Complexity
- Implementation
- Application
- Logical Foundations of Semantic Web

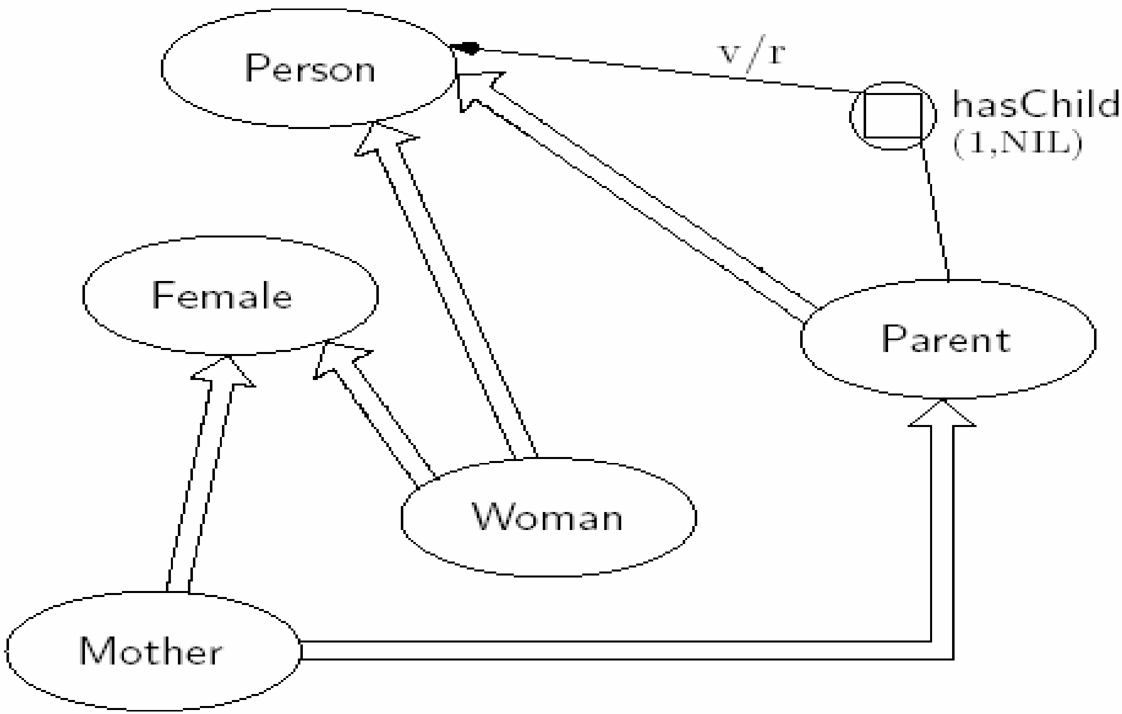
Introduction

Description Logics

- DL: Logics for the representation of and reasoning about
  - terminological knowledge
  - ontologies
  - database schemata
    - schema design, evolution, and query optimization
    - source integration in heterogeneous databases/data warehouses
    - conceptual modelling of multidimensional aggregation
  - …
- Historically, descendants of semantics networks, frame-based systems, and KL-ONE
- A.k.a., terminological logics, terminological KR systems, concept languages, attributive languages, etc.

Network → DL

- Network-based representation, referred as terminology,
  - the generality/specificity of the concepts, in particular, IS-A relationship
- "Parent" can be read as
  - "A parent is a person having at least one child, and all of his/her children are persons."



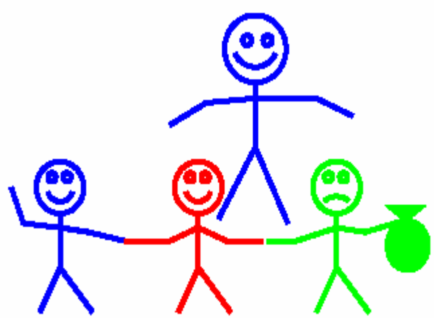
Network → DL (contd.)

- *Nodes:* concepts, i.e., sets or classes of individual objects
  - concepts can have properties (attributes)
- *Links:* relationships among concepts.
  - Beyond IS-A, and more complex relationships are themselves represented as nodes
- *Roles:* by a link from the concept to a node for the role, e.g., hasChild
  - *value restriction* (v/r): a limitation on the range of types of objects that can fill that role
  - *number restriction:* e.g., (1,NIL)

Network → DL (contd.)

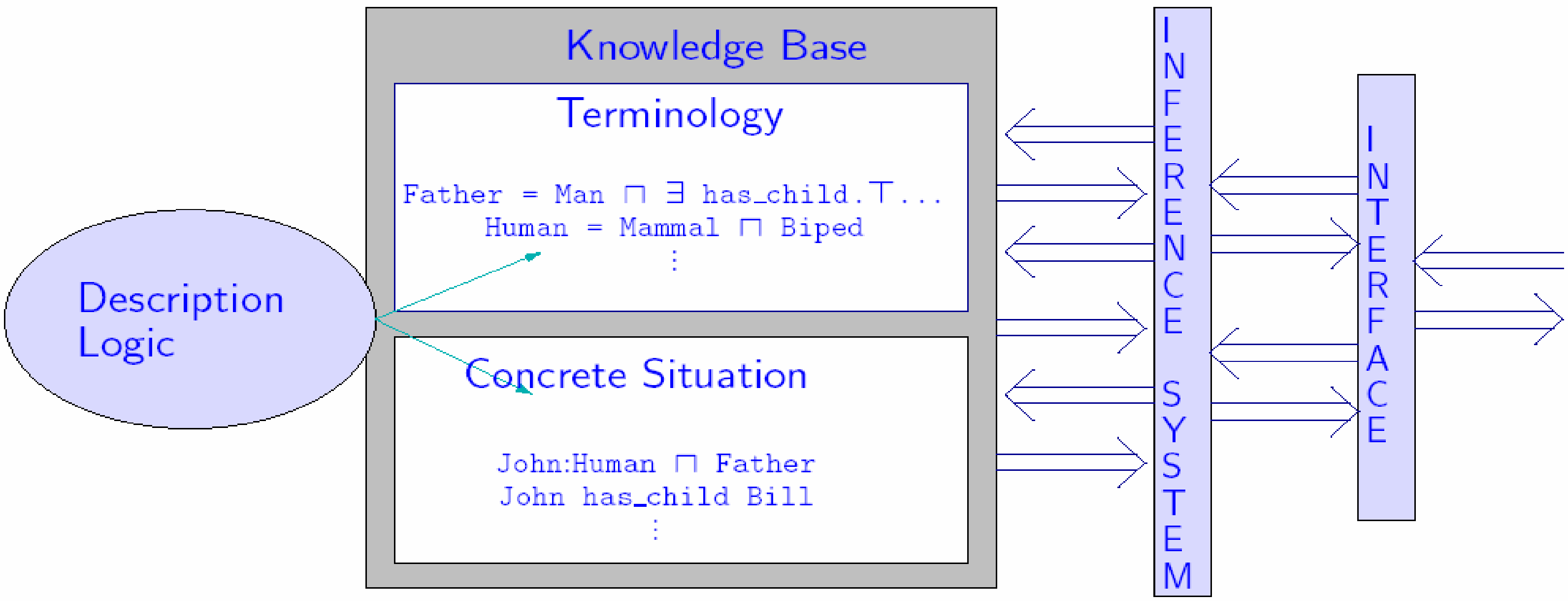
- A Description Logic - mainly characterized by a set of *constructors* that allow to build complex concepts and roles from atomic ones
  - concepts correspond to classes / are interpreted as sets of objects
  - roles correspond to relations / are interpreted as binary relations on objects

Example: Happy Father in the DL $\mathcal{ALC}$



$$\texttt{Man} \sqcap (\exists \texttt{has-child}.\texttt{Blue}) \sqcap (\exists \texttt{has-child}.\texttt{Green}) \sqcap (\forall \texttt{has-child}.\texttt{Happy} \sqcup \texttt{Rich})$$

Basic Architecture



Architecture of a knowledge representation system based on Description Logic.

Description Logic

Description Languages

- The family of *AL*-languages (*AL*=attributive language)
  - A,B stand for atomic concepts
  - R stand for atomic roles
  - C,D stand for concept descriptions
- Constuctors
  - Operations for defining complex concepts

Syntax: *AL*

C,D :: = A	|	(atomic concept)
⊤	|	(universal concept)
⊥	|	(bottom concept)
¬A	|	(atomic negation)
C∩D	|	(intersection)
∀R.C	|	(value restriction)
∃R. ⊤		(limited existential quantification)

FL

- FL^{-}

  $A \mid \top \mid \bot \mid C \sqcap D \mid \forall R.C \mid \exists R.\top$

  - the sublanguage of *AL* obtained by disallowing atomic negation

- $FL_{\circ}$

  $A \mid \top \mid \bot \mid C \sqcap D \mid \forall R.C$

  - the sublanguage of *FL* obtained by disallowing limited existential quantification

Example

- $person \cap \forall hasChild.\bot$

  "those persons without a child"

- $person \cap \exists hasChild.\top$

  "those persons who have a child"

Semantics

- An interpretations *I* consisting of
  - a non-empty domain
  - an function which assigns to every atomic concept a set of the domain and to every atomic role R a binary relation over domains

Semantics: *AL*

$$A^I \subseteq \Delta^I, R^I \subseteq \Delta^I \times \Delta^I$$

$$\mathrm{T}^{\mathrm{I}} = \Delta^I, \bot^I = \varnothing$$

$$(\neg A)^I = \Delta^I \setminus A^I$$

$$(C \cap D)^I = C^I \bigcap D^I$$

$$(\forall R.C)^I = \{a \in \Delta^I \mid \forall b.(a,b) \in R^I \rightarrow b \in C^I\}$$

$$(\exists R.\mathrm{T})^I = \{a \in \Delta^I \mid \exists b.(a,b) \in R^I\}$$

Semantics: Additional Constructors

- $\mathcal{U}$:union of concepts
- ε : full existent
- $\mathcal{N}$:number restrictions
- $\mathcal{C}$:negation of arbitrary concepts

$$(C \cup D)^I = C^I \bigcup D^I$$

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

$$(\geq nR)^I = \{a \in \Delta^I \mid \left|\{b \mid (a,b) \in R^I\}\right| \geq n\}$$

$$(\leq nR)^I = \{a \in \Delta^I \mid \left|\{b \mid (a,b) \in R^I\}\right| \leq n\}$$

$$(\neg C)^I = \Delta^I \setminus C^I$$

Example

$$\mathsf{Person} \sqcap (\leqslant 1\, \mathsf{hasChild} \sqcup (\geqslant 3\, \mathsf{hasChild} \sqcap \exists \mathsf{hasChild}.\mathsf{Female}))$$

- "those persons that have either not more than one child or at least three children, one of which is female."

AL Family

- $\mathcal{AL}[\mathcal{U}][\mathcal{E}][\mathcal{N}][\mathcal{C}]$

  Extending *AL* by any subset of the additional constructors yields a particular *AL*-language.

- Union and full existential quantification can be expressed using negation (and vice versa)
  - $\mathcal{ALC}$ instead of $\mathcal{ALUE}$
  - $\mathcal{ALCN}$ instead of $\mathcal{ALUEN}$

$$C \cup D \equiv \neg(\neg C \cap \neg D)$$

$$\exists R.C \equiv \neg \forall R.\neg C$$

Syntax and Semantics of $\mathcal{ALC}$

Semantics given by means of an **interpretation** $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$:

Constructor	Syntax	Example	Semantics
atomic concept	A	Human	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
atomic role	R	likes	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
For C, D concepts and R a role name			
conjunction	$C \sqcap D$	Human $\sqcap$ Male	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	Nice $\sqcup$ Rich	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
negation	$\neg C$	$\neg$ Meat	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
exists restrict.	$\exists R.C$	$\exists$has-child.Human	$\{x \mid \exists y.\langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
value restrict.	$\forall R.C$	$\forall$has-child.Blond	$\{x \mid \forall y.\langle x, y\rangle \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$

Other Constructors

- Many other DL constructors have been introduced.

Constructor	Syntax	Example	Semantics
number restriction	$(\geq n\ R)$	($\geq$ 7 has-child)	$\{x \mid \lvert\{y.\langle x,y\rangle \in R^{\mathcal{I}}\}\rvert \geq n\}$
($\rightsquigarrow \mathcal{ALCN}$)	$(\leq n\ R)$	($\leq$ 1 has-mother)	$\{x \mid \lvert\{y.\langle x,y\rangle \in R^{\mathcal{I}}\}\rvert \leq n\}$
inverse role	R^{-}	has-child^{-}	$\{\langle x,y\rangle \mid \langle y,x\rangle \in R^{\mathcal{I}}\}$
trans. role	R^{*}	has-child*	$(R^{\mathcal{I}})^{*}$
concrete domain	$u_1,\ldots,u_n.P$	h-father·age, age. $>$	$\{x \mid \langle u_1^{\mathcal{I}}(x),\ldots,u_n^{\mathcal{I}}(x)\rangle \in P\}$
etc.			

TBox

For terminological knowledge: **TBox** contains

Concept definitions $A \doteq C$ (A a concept name, C a complex concept)

$$\mathsf{Father} \doteq \mathsf{Man} \sqcap \exists\mathsf{has\text{-}child.Human}$$

$$\mathsf{Human} \doteq \mathsf{Mammal} \sqcap \forall\mathsf{has\text{-}child}^{-}.\mathsf{Human}$$

$\leadsto$ introduce macros/names for concepts, can be (a)cyclic

Axioms $C_1 \sqsubseteq C_2$ (C_i complex concepts)

$$\exists\mathsf{favourite.Brewery} \sqsubseteq \exists\mathsf{drinks.Beer}$$

$\leadsto$ restrict your models

An interpretation $\mathcal{I}$ **satisfies**

a concept definition $A \doteq C$ iff $A^{\mathcal{I}} = C^{\mathcal{I}}$

an axiom $C_1 \sqsubseteq C_2$ iff $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$

a TBox $\mathcal{T}$ iff $\mathcal{I}$ satisfies all definitions and axioms in $\mathcal{T}$

$\leadsto$ $\mathcal{I}$ is a **model** of $\mathcal{T}$

ABox

For assertional knowledge: **ABox** contains

Concept assertions $a : C$ (a an individual name, C a complex concept)

John : Man $\sqcap$ $\forall$has-child.(Male $\sqcap$ Happy)

Role assertions $\langle a_1, a_2 \rangle : R$ (a_i individual names, R a role)

$\langle$John, Bill$\rangle$: has-child

An interpretation $\mathcal{I}$ **satisfies**

a **concept assertion** $a : C$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$

a **role assertion** $\langle a_1, a_2 \rangle : R$ iff $\langle a_1^{\mathcal{I}}, a_2^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$

an **ABox** $\mathcal{A}$ iff $\mathcal{I}$ satisfies all assertions in $\mathcal{A}$

$\rightsquigarrow$ $\mathcal{I}$ is a **model** of $\mathcal{A}$

Basic Inference Problems

Subsumption: $C \sqsubseteq D$ — Is $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all interpretations $\mathcal{I}$?

w.r.t. TBox $\mathcal{T}$: $C \sqsubseteq_{\mathcal{T}} D$ — Is $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all models $\mathcal{I}$ of $\mathcal{T}$?

$\rightsquigarrow$ structure your knowledge, compute taxonomy

Consistency: Is C consistent w.r.t. $\mathcal{T}$? — Is there a model $\mathcal{I}$ of $\mathcal{T}$ with $C^{\mathcal{I}} \neq \emptyset$?

of ABox $\mathcal{A}$: Is $\mathcal{A}$ consistent? — Is there a model of $\mathcal{A}$?

of KB $(\mathcal{T},\mathcal{A})$: Is $(\mathcal{T},\mathcal{A})$ consistent? — Is there a model of both $\mathcal{T}$ and $\mathcal{A}$?

Inference Problems are **closely related:**

$C \sqsubseteq_{\mathcal{T}} D$ iff $C \sqcap \neg D$ is **in**consistent w.r.t. $\mathcal{T}$,
(no model of $\mathcal{I}$ has an instance of $C \sqcap \neg D$)

C is consistent w.r.t. $\mathcal{T}$ iff **not** $C \sqsubseteq_{\mathcal{T}} A \sqcap \neg A$

$\rightsquigarrow$ **Decision Procdures for consistency (w.r.t. TBoxes) suffice**

Inference Problems

- For most DLs, the basic inference problems are decidable, with complexities between P and ExpTime.
- Why is decidability important? Why does semi-decidability not sufice?
  - If subsumption (and hence consistency) is undecidable, and
    - subsumption is semi-decidable, then consistency is not semi-decidable
    - consistency is semi-decidable, then subsumption is not semi-decidable
  - Quest for a highly expressive DL with decidable\practicable inference problems
    - expressiveness depends on the application
    - practicability changed over the time

DLs as Decidable Fragments of FOL

a unary predicate ϕ_A for a concept name A
a binary relation ρ_R for a role name R

Translate complex concepts C, D as follows:

$$
\begin{aligned}
t_x(A) &= \phi_A(x), & t_y(A) &= \phi_A(y), \\
t_x(C \sqcap D) &= t_x(C) \wedge t_x(D), & t_y(C \sqcap D) &= t_y(C) \wedge t_y(D), \\
t_x(C \sqcup D) &= t_x(C) \vee t_x(D), & t_y(C \sqcup D) &= t_y(C) \vee t_y(D), \\
t_x(\exists R.C) &= \exists y.\rho_R(x,y) \wedge t_y(C), & t_y(\exists R.C) &= \exists x.\rho_R(y,x) \wedge t_x(C), \\
t_x(\forall R.C) &= \forall y.\rho_R(x,y) \Rightarrow t_y(C), & t_y(\forall R.C) &= \forall x.\rho_R(y,x) \Rightarrow t_x(C).
\end{aligned}
$$

A TBox $\mathcal{T} = \{C_i \sqsubseteq D_i\}$ is translated as

$$\Phi_{\mathcal{T}} = \forall x. \bigwedge_{1 \leq i \leq n} t_x(C_i) \Rightarrow t_x(D_i)$$

As Fragments of FOL

$$
\begin{aligned}
C \text{ is consistent } &\text{ iff } \text{ its translation } t_x(C) \text{ is satisfiable,}\\
C \text{ is consistent w.r.t. } \mathcal{T} &\text{ iff } \text{ its translation } t_x(C) \wedge \Phi_{\mathcal{T}} \text{ is satisfiable,}\\
C \sqsubseteq D &\text{ iff } t_x(C) \Rightarrow t_x(D) \text{ is valid}\\
C \sqsubseteq_{\mathcal{T}} D &\text{ iff } \Phi_t \Rightarrow \forall x.(t_x(C) \Rightarrow t_x(D)) \text{ is valid.}
\end{aligned}
$$

⇝ $\mathcal{ALC}$ is a fragment of FOL with 2 variables (L2), known to be decidable

⇝ $\mathcal{ALC}$ with inverse roles and Boolean operators on roles is a fragment of L2

⇝ further adding number restrictions yields a fragment of C2 (L2 with "counting quantifiers"), known to be decidable

- ✦ in contrast to most DLs, adding transitive roles/transitive closure operator to L2 leads to **undecidability**
- ✦ many DLs (like many modal logics) are fragments of the **Guarded Fragment**
- ✦ most DLs are less complex than L2:
  L2 is NExpTime-complete, most DLs are in ExpTime

DLs as Modal Logics

DLs and Modal Logics are closely related:

$\mathcal{ALC} \rightleftharpoons$ multi-modal K:

$$C \sqcap D \rightleftharpoons C \wedge D, \qquad C \sqcup D \rightleftharpoons C \vee D$$

$$\neg C \rightleftharpoons \neg C \quad ,$$

$$\exists R.C \rightleftharpoons \langle R \rangle C \;, \qquad \forall R.C \rightleftharpoons [R]C$$

transitive roles $\stackrel{\cdot}{\rightleftharpoons}$ transitive frames (e.g., in $\mathrm{K4}$)

regular expressions on roles $\stackrel{\cdot}{\rightleftharpoons}$ regular expressions on programs (e.g., in PDL)

inverse roles $\stackrel{\cdot}{\rightleftharpoons}$ converse programs (e.g., in C-PDL)

number restrictions $\stackrel{\cdot}{\rightleftharpoons}$ deterministic programs (e.g., in D-PDL)

- no TBoxes available in modal logics
  - $\rightsquigarrow$ "internalise" axioms using a universal role u: $C \doteq D \rightleftharpoons [u](C \Leftrightarrow D)$
- no ABox available in modal logics $\rightsquigarrow$ use nominals

Algorithm

Tableaux

- Resoning procedure: tableau algorithm

  - works on a tree (semantics through viewing tree as an ABox):

    nodes represent elements of $\Delta^{\mathcal{I}}$, labelled with sub-concepts of C_0

    edges represent role-successorships between elements of $\Delta^{\mathcal{I}}$

  - works on concepts in **negation normal form**: push negation inside using de Morgan' laws and

$$\neg(\exists R.C) \rightsquigarrow \forall R.\neg C \qquad \neg(\forall R.C) \rightsquigarrow \exists R.\neg C$$

$$\neg(\leq n\ R) \rightsquigarrow (\geq (n+1) R) \qquad \neg(\geq n\ R) \rightsquigarrow (\leq (n-1) R) \quad (n \geq 0)$$

$$\neg(\geq 0\ R) \rightsquigarrow A \sqcap \neg A$$

  - is initialised with a tree consisting of a single (root) node x_0 with $\mathcal{L}(x_0) = \{C_0\}$:

$$x_0 \bullet \{C_0\}$$

  - a tree T contains a **clash** if, for a node x in T,

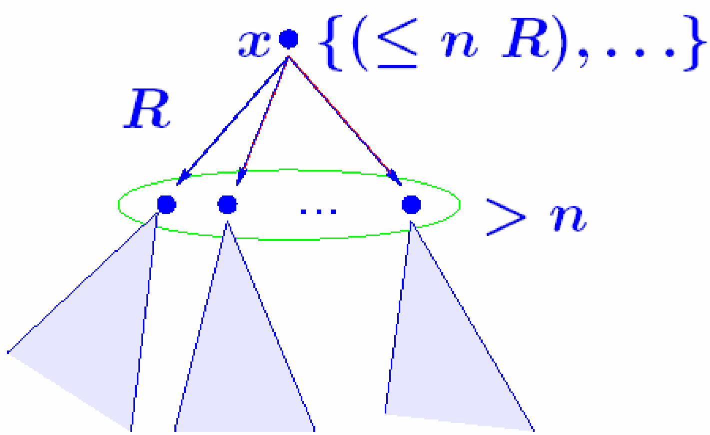
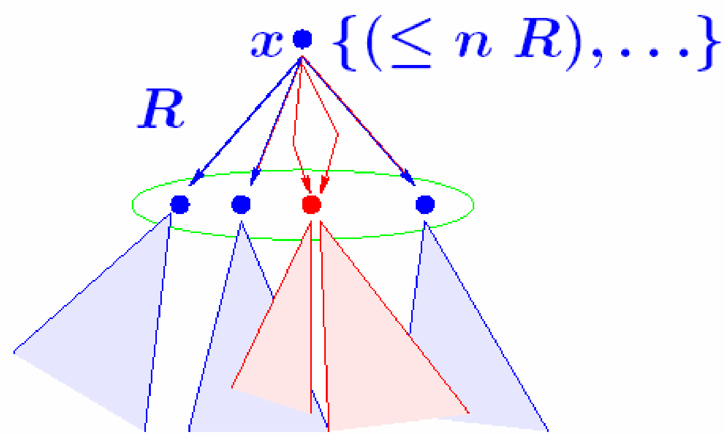
$$\{A, \neg A\} \subseteq \mathcal{L}(x) \text{ or}$$

$$\{(\geq m\ R), (\leq n\ R)\} \subseteq \mathcal{L}(x) \text{ for } n < m$$

$\mathcal{ALC}$ Tableau Rules

$x \bullet \{C_1 \sqcap C_2, \ldots\}$	$\rightarrow_\sqcap$	$x \bullet \{C_1 \sqcap C_2, C_1, C_2, \ldots\}$
$x \bullet \{C_1 \sqcup C_2, \ldots\}$	$\rightarrow_\sqcup$	$x \bullet \{C_1 \sqcap C_2, C, \ldots\}$ for $C \in \{C_1, C_2\}$
$x \bullet \{\exists R.C, \ldots\}$	$\rightarrow_\exists$	$x \bullet \{\exists R.C, \ldots\}$ $R \downarrow$ $y \bullet \{C\}$
$x \bullet \{\forall R.C, \ldots\}$ $R \downarrow$ $y \bullet \{\ldots\}$	$\rightarrow_\forall$	$x \bullet \{\forall R.C, \ldots\}$ $R \downarrow$ $y \bullet \{C, \ldots\}$

N Tableau Rules

$x \bullet \{(\geq n\ R), \ldots\}$ x has no R-succ.	$\rightarrow_{\geq}$	$x \bullet \{(\geq n\ R), \ldots\}$ $\downarrow R$ $y \bullet \{\}$
	$\rightarrow_{\leq}$	 merge two R-succs.

Soundeness and Completeness

Lemma

- Let C0 be an ALCN concept and T obtained by applying the tableau rules to C0. Then

  1. the rule application terminates,
  2. if T is consistent and ! is applicable to T, then ! can be applied such that it yields consistent T0,
  3. if T contains a clash, then T has no model, and
  4. if no more rules apply to T, then T denes (canonical) model for C0.

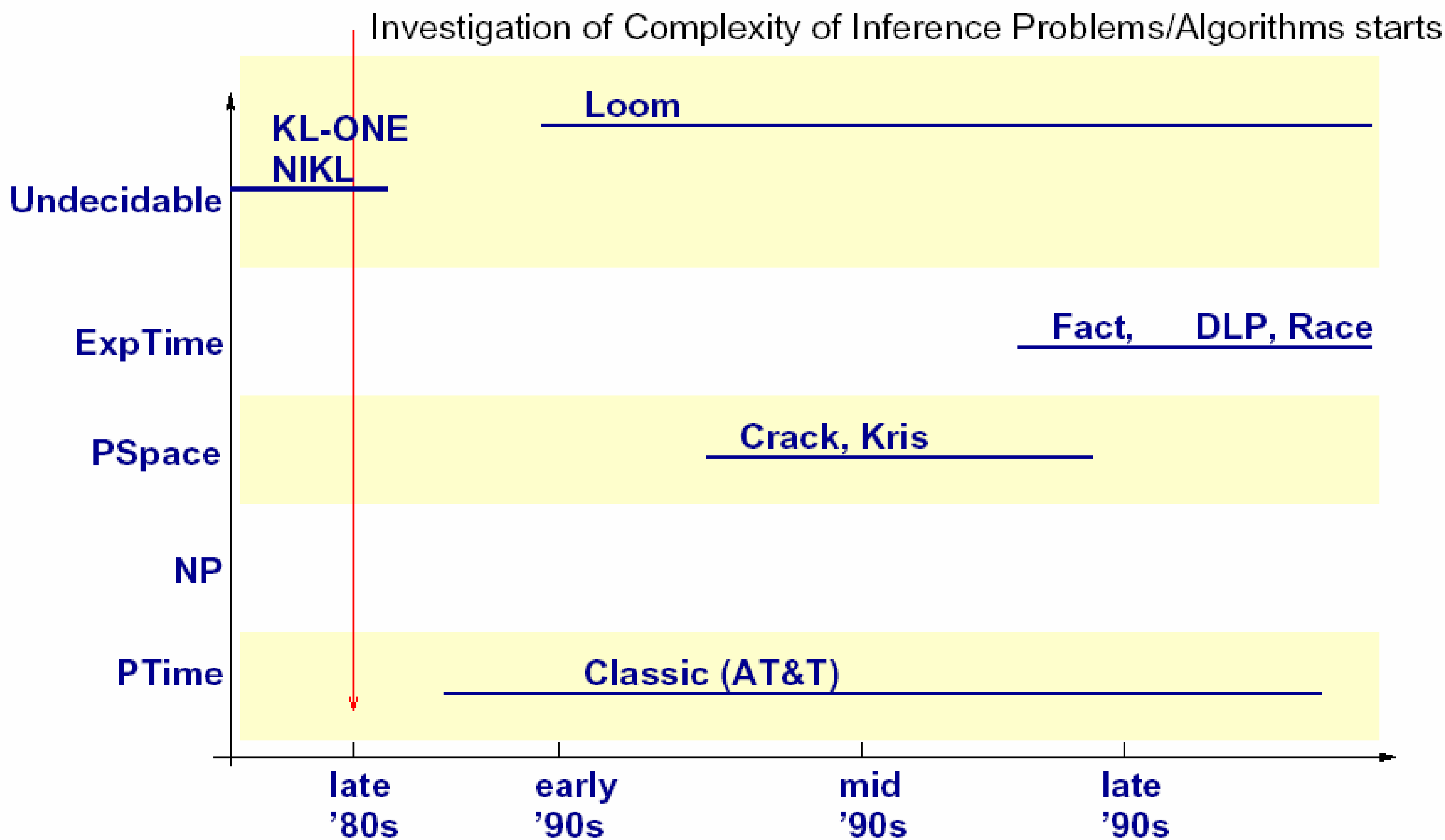
Corollary

(1) The tableau algorithm is a PSpace decision procedure for consistency (and subsumption) of ALCN concepts

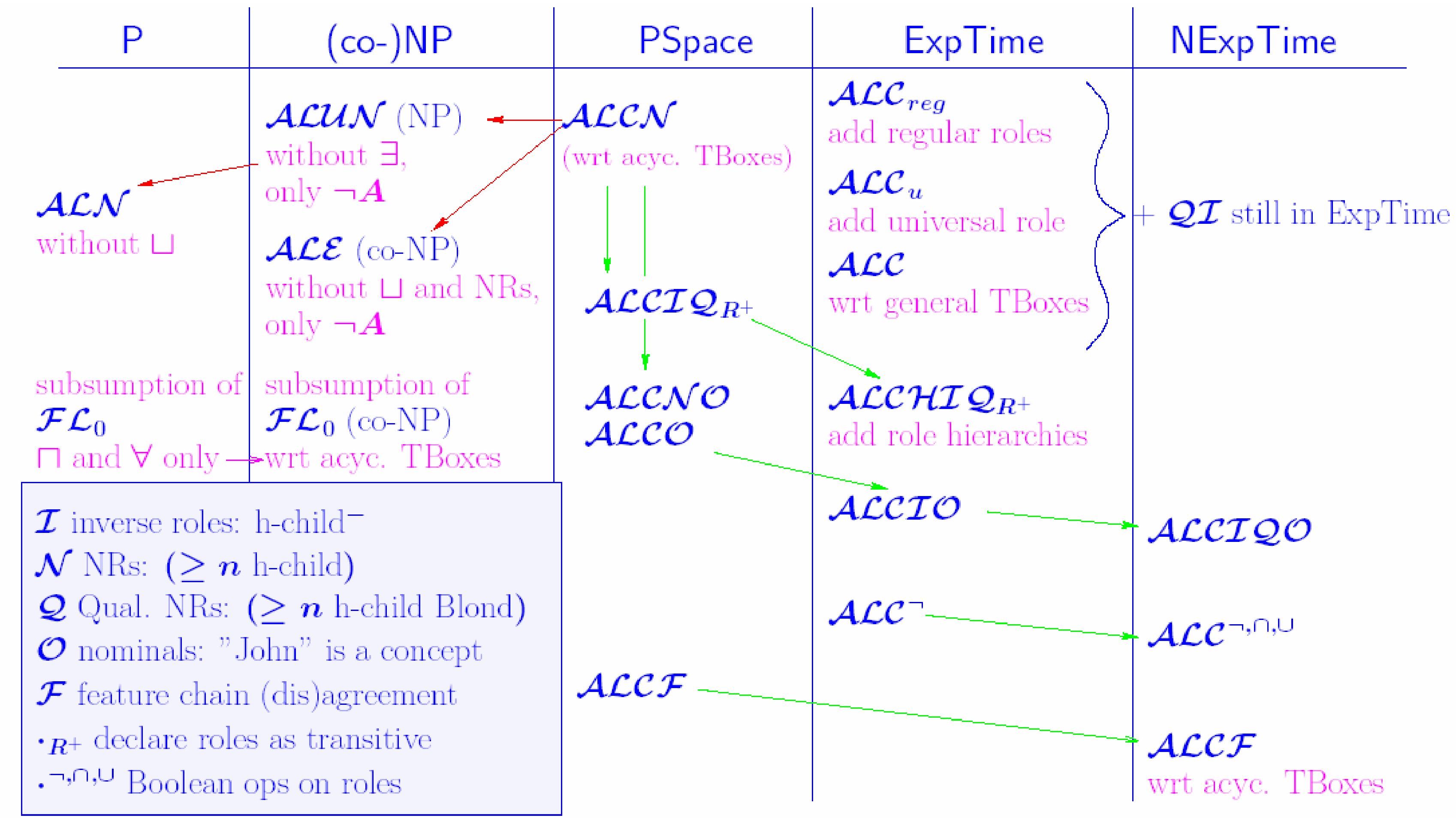
(2) ALCN has the tree model property

Complexity

Complexity


Investigation of Complexity of Inference Problems/Algorithms starts
Undecidable
ExpTime
PSpace
NP
PTime
KL-ONE
NIKL
Loom
Fact,
DLP, Race
Crack, Kris
Classic (AT&T)
late
'80s
early
'90s
mid
'90s
late
'90s

Complexity of Concept Consistency



Implementation

Implementation

In the last 5 years, DL-based systems were built that

- can handle DLs far more expressive than ALC (close relatives of converse-DPDL)
  - Number restrictions: “people having at most 2 children"
  - Complex roles: inverse (“has-child" – “child-of")
    transitive closure (“ospring" – “has-child")
    role inclusion (“has-daughter" – “has-child"), etc.
- implement provably sound and complete inference algorithms
  (for ExpTime-complete problems)
- can handle large knowledge bases
  (e.g., Galen medical terminology ontology: 2,740 concepts, 413 roles, 1,214 axioms)
- are highly optimised versions of tableau-based algorithms
- perform (surprisingly well) on benchmarks for modal logic reasoners

Application

Application Areas

- Terminological KR and Ontologies
  - DLs initially designed for terminological KR (and reasoning)
  - Natural to use DLs to build and maintain ontologies
- Semantic Web
- Configuration
- Software information systems
- Database applications
- . . .

Semantic Web

- **Semantic** markup will be added to web resources
- Markup will use **Ontologies** to provide common terms of reference with clear semantics
- Requirement for web based ontology language
  - Well defined semantics
  - Builds on existing Web standards (XML, RDF, RDFS)
- Resulting language (DAML+OIL) is **based on a DL** (SHIQ)
- DL **reasoning** can be used to, e.g.,
  - Support ontology design and maintenance
  - Classify resources w.r.t. ontologies

Logical Foundations of Semantic Web