Part II : Non standard reasoning in DL

Naouel KARAM
karam@isima.fr

- Introduction
- Non standard inferences
  - LCS and MSC
  - Matching
  - Rewriting
    - Approximation
  - Difference
    - Semantic difference
    - Syntactic difference

Introduction

- “Non-standard” by opposition to “standard” inferences (subsumption, satisfiability,...)
- Standard inferences are not sufficient when:
  - It comes to generate new concept descriptions from given ones
  - Concepts are specified using different vocabularies
  - Concepts are described in different levels of abstraction
  - ...
- Non-standard inferences needed for the construction and maintenance of large DL knowledge bases.
- First ad hoc implementations integrated into the CLASSIC DL-system

Part II : Non standard reasoning in DL

Naouel KARAM
karam@isima.fr

- Introduction
- Non standard inferences
  - LCS and MSC
  - Matching
  - Rewriting
    - Approximation
  - Difference
    - Semantic difference
    - Syntactic difference

LCS

- Least Common Subsumer. The ***lcs*** of a given sequence of concept descriptions is a description that represent their common properties.

  Formally:

  C is the lcs of $C_1, ..., C_k$ iff:

  1. $C_i \sqsubseteq C$ for all $i = 1, ..., k$; and
  2. for a description E, if $C_i \sqsubseteq E$ for all $i = 1, ..., k$, then $C \sqsubseteq E$

- The ***lcs*** is unique up to equivalence.
- The ***lcs*** need-not always exist:
  - There may be several subsumption incomparable minimal descriptions satisfying 1 (This case cannot occur for DLs allowing for conjunction).
  - There may be an infinite chain of more and more specific descriptions satisfying 1.

MSC

- Most Specific Concept. The ***msc*** of individuals described in an ABox is a description that represent their common properties.

  Formally:

  C is the msc of $a_1, ..., a_k$ iff:

  1. $C(a_i) \in \mathcal{A}$ for all $i = 1, ..., k$; and
  2. for a description E, if $E(a_i) \in \mathcal{A}$ for all $i = 1, ..., k$, then $C \sqsubseteq E$

- Close connection between the ***msc*** and the ***lcs***
  - Given the ABox: $C_1(a_1), ..., C_n(a_n)$
    $msc(a_1, ..., a_n)$ is equivalent to $lcs(C_1, ..., C_n)$
  - Reduction to ***lcs*** and unary ***msc*** operation
    $msc(a_1, ..., a_n) \equiv lcs(msc(a_1), ..., msc(a_n))$

Applications of the LCS and the MSC

- Learning from examples. Find the most specific concept that generalizes a set of examples.
- Used as an alternative to disjunction. Replace $C_1 \sqcup ... \sqcup C_n$ by $lcs(C_1, ..., C_n)$ which represents the best approximation of the disjunction within a DLs not allowing for it.
- Support for the “bottom-up” construction of DL knowledge. Derive concept descriptions from typical examples in the ABox.

Part II : Non standard reasoning in DL

Naouel KARAM
karam@isima.fr

- Introduction
- Non standard inferences
  - LCS and MSC
  - Matching
  - Rewriting
    - Approximation
  - Difference
    - Semantic difference
    - Syntactic difference

Matching

- Given a concept pattern D (a concept description with variables) and a concept description C,
  - a matching problem modulo equivalence asks for a substitution σ (of variables by concept descriptions) such that $C \equiv \sigma(D)$
  - a matching problem modulo subsumption asks for a substitution σ such that $C \sqsubseteq \sigma(D)$
- Applications of matching
  - Filter unimportant aspects of complicated concepts in KB.

    $D :=$ ∀research-interests.X

    $C :=$ ∀pets.Cat ⊓ ∀research-interests.AI ⊓ ∀hobbies.Gardening

    assigns AI to the variable X
  - Detect redundancies in KB.

    Woman ⊓ haschild.Woman

    Female ⊓ Human ⊓ haschild.(Female ⊓ Human)
  - Find interschema assertions in KB integration

LIMOS NEOS

Part II : Non standard reasoning in DL

Naouel KARAM
karam@isima.fr

- Introduction
- Non standard inferences
  - LCS and MSC
  - Matching
  - Rewriting
    - Approximation
  - Difference
    - Semantic difference
    - Syntactic difference

Rewriting

- General framework
  - Let $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$ desription languages, C an $\mathcal{L}_1$-concept description and $\mathcal{T}$ an $\mathcal{L}_2$-TBox

    Rewrite C into an $\mathcal{L}_3$-concept description D such that

    1. C $\rho_{\mathcal{T}}$ D
    2. D satisfies an optimality criteria (ex. minimal in size)
- Instances
  - The minimal rewriting problem
    - The DLs are the same language $\mathcal{L}$
    - The TBox $\mathcal{T}$ is acyclic
    - The binary relation $\rho_{\mathcal{T}}$ is equivalence modulo the TBox
    - D is minimal w.r.t size of concept descriptions

  Applications

    - Increase the readability of large concept descriptions by using defined concept names from a TBox

Rewriting

- The problem of rewriting queries using views
  - $\mathcal{L}_1 = \mathcal{L}_2 = \mathcal{ALCNR}$ $\mathcal{L}_3 = \{\sqcap, \sqcup\}$
  - The binary relation $\rho_{\mathcal{T}}$ is subsumption (C $\sqsupseteq$ D)
  - D is maximal w.r.t subsumption

Application

- Databases: optimize the runtime of queries by using cashed views

- The approximation of concept descriptions

  exact approximation

  Upper/lower approximations

  - The TBox $\mathcal{T}$ is empty
  - The binary relation $\rho_{\mathcal{T}}$ is equivalence, subsume $\sqsubseteq$, subsumed by $\sqsupseteq$
  - D is minimal (*upper* case) / maximal (*lower* case) w.r.t subsumption

Application

- Translation of knowledge bases from an expressive DL into a less expressive one

Part II : Non standard reasoning in DL

Naouel KARAM
karam@isima.fr

- Introduction
- Non standard inferences
  - LCS and MSC
  - Matching
  - Rewriting
    - Approximation
  - Difference
    - Semantic difference
    - Syntactic difference

Approximation

- Upper approximation of $\mathcal{ALC}$-concept descriptions by $\mathcal{ALE}$-concept descriptions
- Given an $\mathcal{ALC}$-concept description $C = E \sqcup F$, its $\mathcal{ALE}$-approximation is $lcs(E, F)$.
- A naïve approach: replace every disjunction by the lcs of its disjuncts

$$C_{ex1} = (\forall r.B \sqcup (\exists r.B \sqcap \forall r.A)) \sqcap \exists r.A$$
$$C_{ex2} = \exists r.A \sqcap \exists r.B \sqcap \forall r.(\neg A \sqcup \neg B)$$

$\mathcal{ALC}$ normal from

- An $\mathcal{ALC}$-concept description C is in $\mathcal{ALC}$-normal form iff
  1. C $\equiv \bot$, then C = $\bot$ or C $\equiv \top$, then C = $\top$
  2. C is of the form $C = C_1 \sqcup \ldots \sqcup C_n$ with

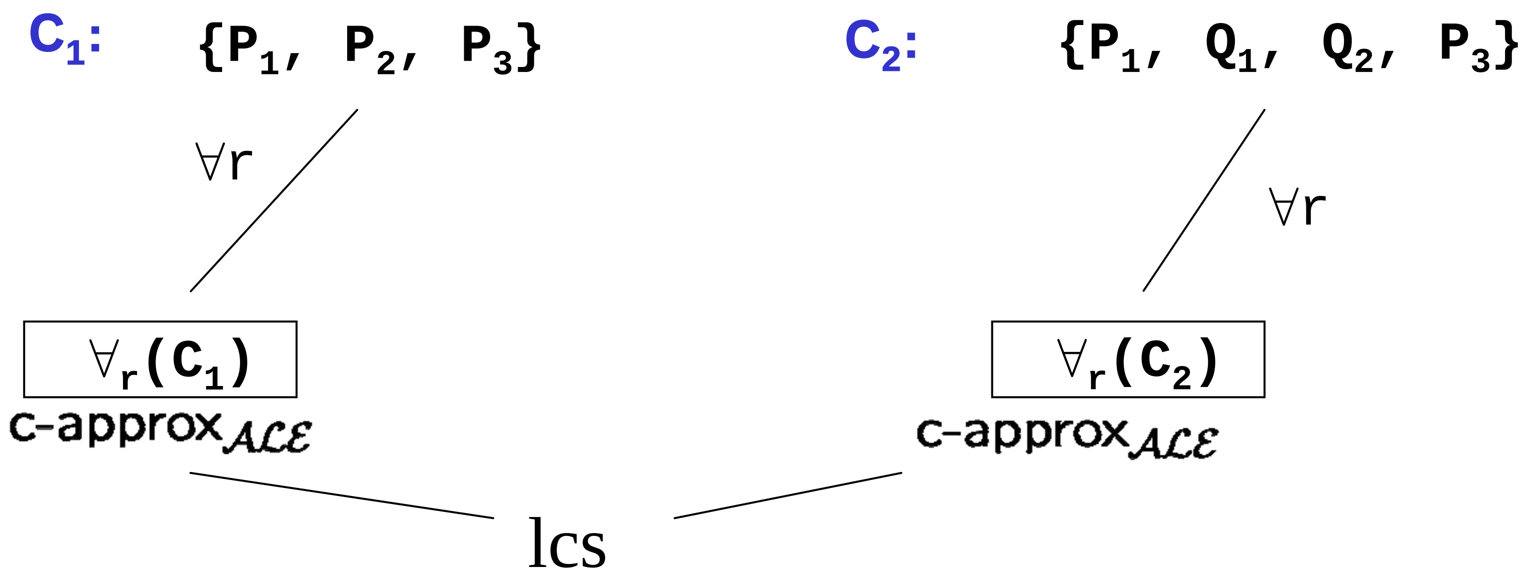
$$C_i = \bigsqcap_{A \in \mathrm{prim}(C_i)} A \sqcap \forall r.\forall_r(C_i) \sqcap \bigsqcap_{C' \in \exists_r(C_i)} \exists r.C'$$

in $\mathcal{ALC}$-normal form

- $\mathcal{ALC}$-normal form can be of size exponential

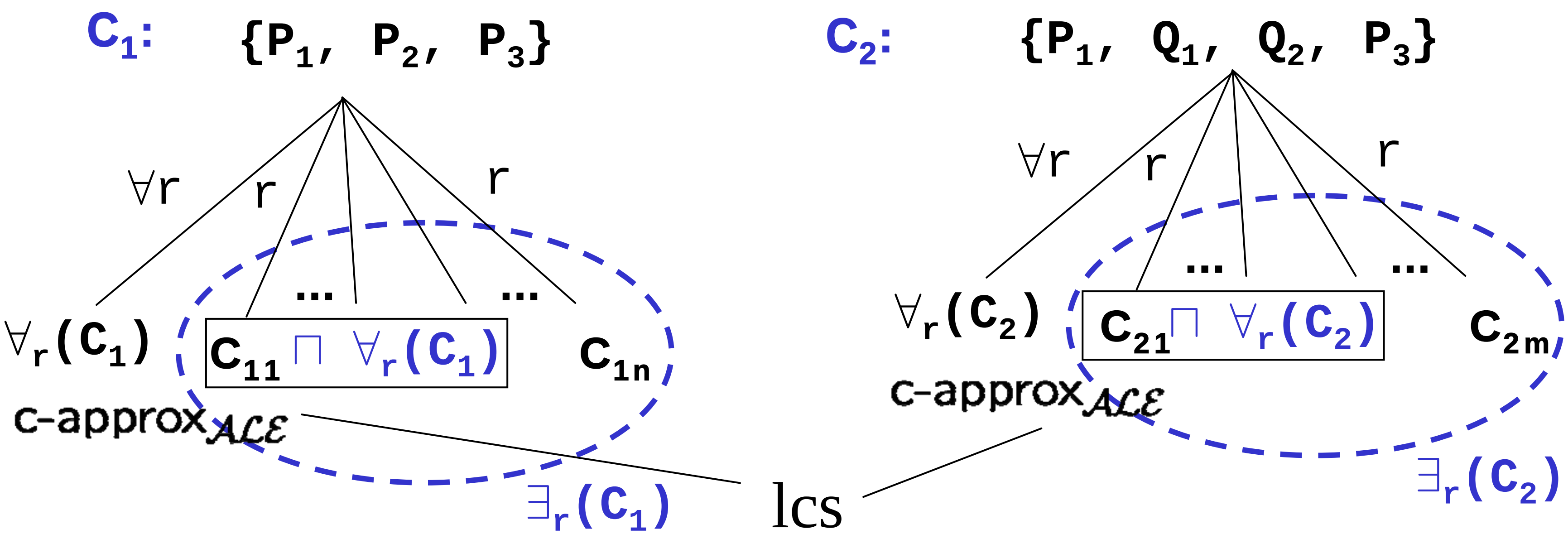
$$(A_1 \sqcup A_2) \sqcap \ldots \sqcap (A_{2n-1} \sqcup A_{2n})$$

The approximation algorithm



$$\text{c-approx}_{\mathcal{ALE}}(C) := P_1 \sqcap P_3$$

The approximation algorithm



$$\text{c-approx}_{\mathcal{ALE}}(C) := \text{P}_1 \sqcap \text{P}_3 \sqcap \forall r.lcs\{\text{c-approx}_{\mathcal{ALE}}(\forall_r(C_i)) \mid 1 \leq i \leq n\} \sqcap$$

The approximation algorithm

Require: $\mathcal{ALC}$ – concept description C

Ensure: upper $\mathcal{ALE}$-approximation of C

if $C \equiv \bot$ **then**
    c-approx$_{\mathcal{ALE}}(C) := \bot$
else
    if $C \equiv \top$ **then**
        c-approx$_{\mathcal{ALE}}(C) := \top$
    else
        C $\mathcal{ALC}$-normal from $C_1 \sqcup \ldots \sqcup C_n$

$$\text{c-approx}_{\mathcal{ALE}}(C) := \bigsqcap_{A \in \bigcap_{i=1..n} prim(C_i)} A \sqcap \forall r.lcs\{\text{c-approx}_{\mathcal{ALE}}(\forall_r(C_i)) \mid 1 \leq i \leq n\} \sqcap$$

$$\bigsqcap_{(C'_1,\ldots,C'_n) \in \exists_r(C_1) \times \ldots \times \exists_r(C_n)} \exists r.lcs\{\text{c-approx}_{\mathcal{ALE}}(C'_i \sqcap \forall_r(C_i)) \mid 1 \leq i \leq n\}$$

    end if
end if

Examples

- Compute the ALE-approximations of

$$C_{ex1} = (\forall r.B \sqcup (\exists r.B \sqcap \forall r.A)) \sqcap \exists r.A$$

$$C_{ex2} = \exists r.A \sqcap \exists r.B \sqcap \forall r.(\neg A \sqcup \neg B)$$

Part II : Non standard reasoning in DL

Naouel KARAM
karam@isima.fr

- Introduction
- Non standard inferences
  - LCS and MSC
  - Matching
  - Rewriting
    - Approximation
  - Difference
    - Semantic difference
    - Syntactic difference

Difference

- The difference operator allows to remove from a given description the information contained into another description.
- Two definitions of the difference:

  - The semantic difference

$$C - D := \max_{\sqsubseteq}\{E \mid E \sqcap D \equiv C\}$$

where C is required to subsume D.

$$\mathsf{Father} - \mathsf{Man} := \exists \mathsf{hasChild}.\mathsf{Human}$$

  - The “syntactic” difference

$$C - D := min_{\preceq_d}\{E \mid E \sqcap D \equiv C \sqcap D\}$$

$$\mathsf{Uncle} - \mathsf{Father} := \exists \mathsf{hasSibling}.\mathsf{Father}$$

Part II : Non standard reasoning in DL

Naouel KARAM
karam@isima.fr

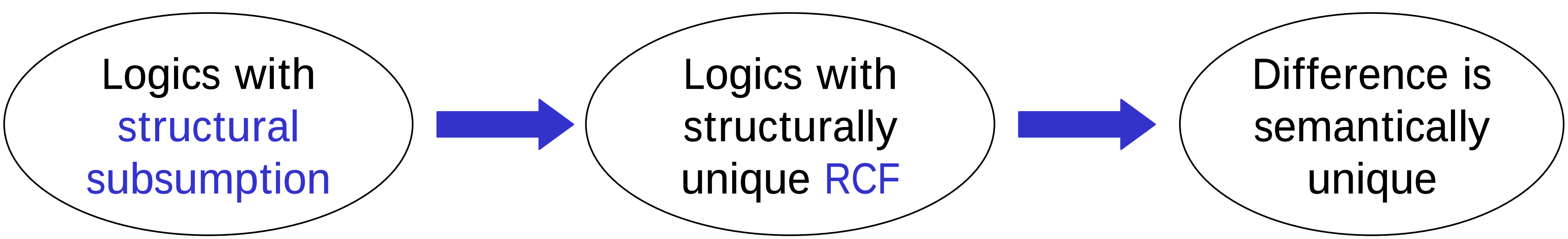
- Introduction
- Non standard inferences
  - LCS and MSC
  - Matching
  - Rewriting
    - Approximation
  - Difference
    - Semantic difference
    - Syntactic difference

The semantic difference

- Some important properties
  - If the DL allows for negation $C - D$ is always equivalent to $\neg(D \sqcap \neg C)$ (not useful in practice)
  - For certain DLs the difference may not be unique

$$\bot - (P \sqcap Q) = \{\neg P, \neg Q\}$$

  - Teege characterizes the necessary conditions for a logic to have a semantically unique difference.



Semantically unique difference

- RCF (Reduced Clause Form)
  - A clause is a description that cannot be decomposed into a conjunction of descriptions

    $\forall r.(P \sqcap Q)$ is not a clause because $\equiv \forall r.P \sqcap \forall r.Q$
  - A set of clauses is reduced if no clause subsumes the conjunction of the others.

$$A \doteq \forall r_1.P_1 \sqcap \forall r_2.(P_2 \sqcap \forall r_3.\exists r_4.(P_3 \sqcap P_4)) \sqcap \forall r_2.\forall r_3.\exists r_4.P_3$$

$$\{\forall r_1.P_1, \forall r_2.P_2, \forall r_2.\forall r_3.\exists r_4.(P_3 \sqcap P_4)\}$$

- Structural subsumption
  - A a clause and $B = B_1 \sqcap ... \sqcap B_n$ given by its RCF

$$A \sqsupseteq B \Leftrightarrow \exists 1 \leq i \leq n : A \sqsupseteq B_i$$

Implementation of the semantic difference operator

Logics with structural subsumption → Logics with structurally unique RCF → Difference is semantically unique

C - D implemented by removing in the RCF of C the equivalent clauses present in the RCF of D

- The cases involving a non unique difference
  - Use of certain operators
  - Combination of operators
- The set of constructors of the logic $\mathcal{L}_1$ is The maximal set supporting structurally unique RCFs

  - $\sqcap, \sqcup, \top, \bot, (\exists r.C), (\exists f.C), (\geq nR)$ for concepts,
  - $\bot$, $\circ$, $|$ for roles,
  - $\bot$, $\circ$ for functional roles.

Cases of non unique differences

- Decomposition of $\bot$

$$A \sqcap \neg A \equiv \bot$$
$$\forall f.\bot \sqcap \exists f.\top \equiv \bot$$
$$(\geq (n+1)r) \sqcap (\leq nr) \equiv \bot$$
$$\forall r.\bot \sqcap \exists r.\top \equiv \bot$$

  - Compute the following differences:

$$(\exists f_3.\top \sqcap \forall f_3.\bot) - (\exists f_1.\top \sqcap \exists f_2.\top)$$
$$(\geq (p+1)r_3) \sqcap (\leq nr_3) - (\geq (n+1)r_1) \sqcap \leq mr_2)$$
$$(\forall \mathrm{r}.\bot) - (\forall \mathrm{r}.\mathrm{P} \sqcap \forall \mathrm{r}.(\exists \mathrm{r}.\top))$$

- Roles with fixed number of images
- Feature agreement or role value map

Part II : Non standard reasoning in DL

Naouel KARAM
karam@isima.fr

- Introduction
- Non standard inferences
  - LCS and MSC
  - Matching
  - Rewriting
    - Approximation
  - Difference
    - Semantic difference
    - Syntactic difference

The syntactic difference

$$C - D := min_{\preceq_d}\{E \mid E \sqcap D \equiv C \sqcap D\}$$

- Proposed for $\mathcal{ALE}$-descriptions, $\mathcal{ALC}$ and $\mathcal{ALE}$-descriptions
- The syntactic order $\preceq_d$

$\widehat{C} \preceq_d C$ if $\widehat{C}$ is obtained by removing subdescriptions from C

$$C_{ex} = \text{Q} \sqcap \text{Q} \sqcap \forall \text{r.P} \sqcap \exists \text{r.(P} \sqcap \exists \text{r.Q)}$$

$$\widehat{C}_{ex} = \text{Q} \sqcap \forall \text{r.P} \sqcap \exists \text{r.(}\exists \text{r.Q)}$$ ← reduced

if $\widehat{C} \equiv C$ and $\widehat{C}$ is the most specific description verifying this property
$\widehat{C}$ is called reduced

- The difference algorithm principle:
  - Remove from C the information in D or already present in C

A difference algorithm for $\mathcal{ALE}$

Syntactic difference algorithm

$\mathcal{G}_C$: {~~P_1~~, P_2, ~~P_3~~}

$\forall r$

$\forall_r(C)$

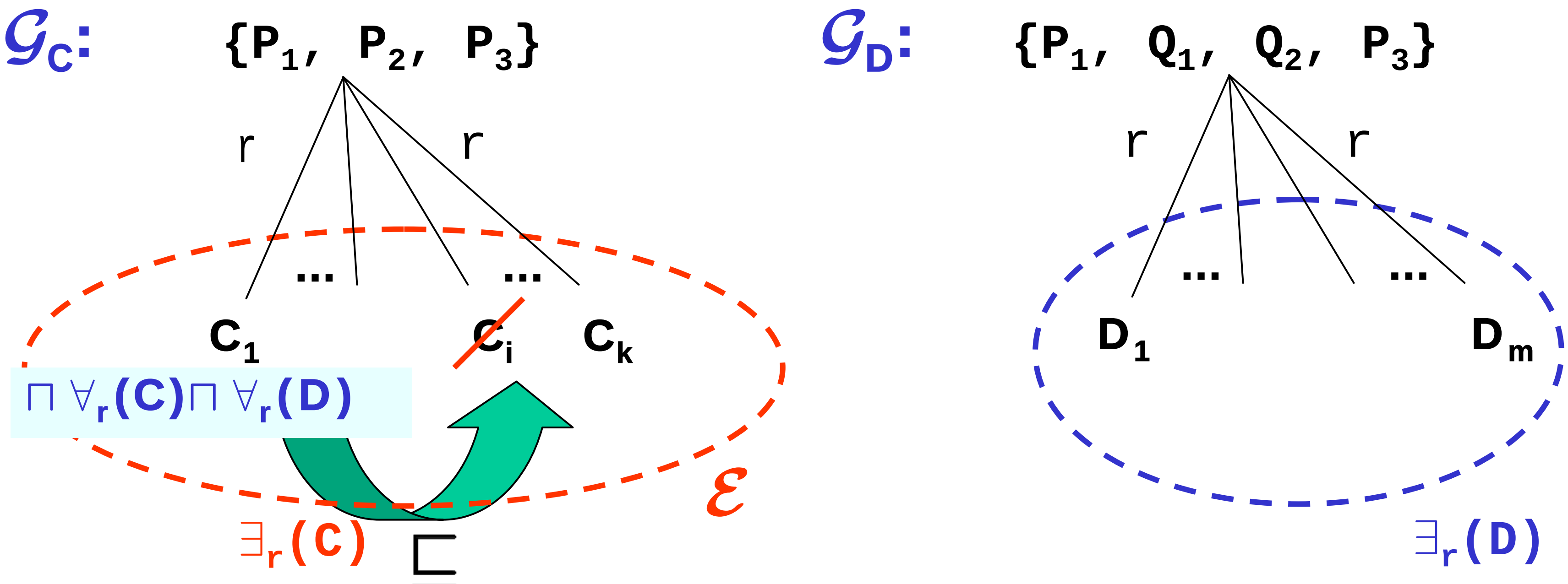
$\mathcal{G}_D$: {P_1, Q_1, Q_2, P_3}

$\forall r$

$\forall_r(D)$

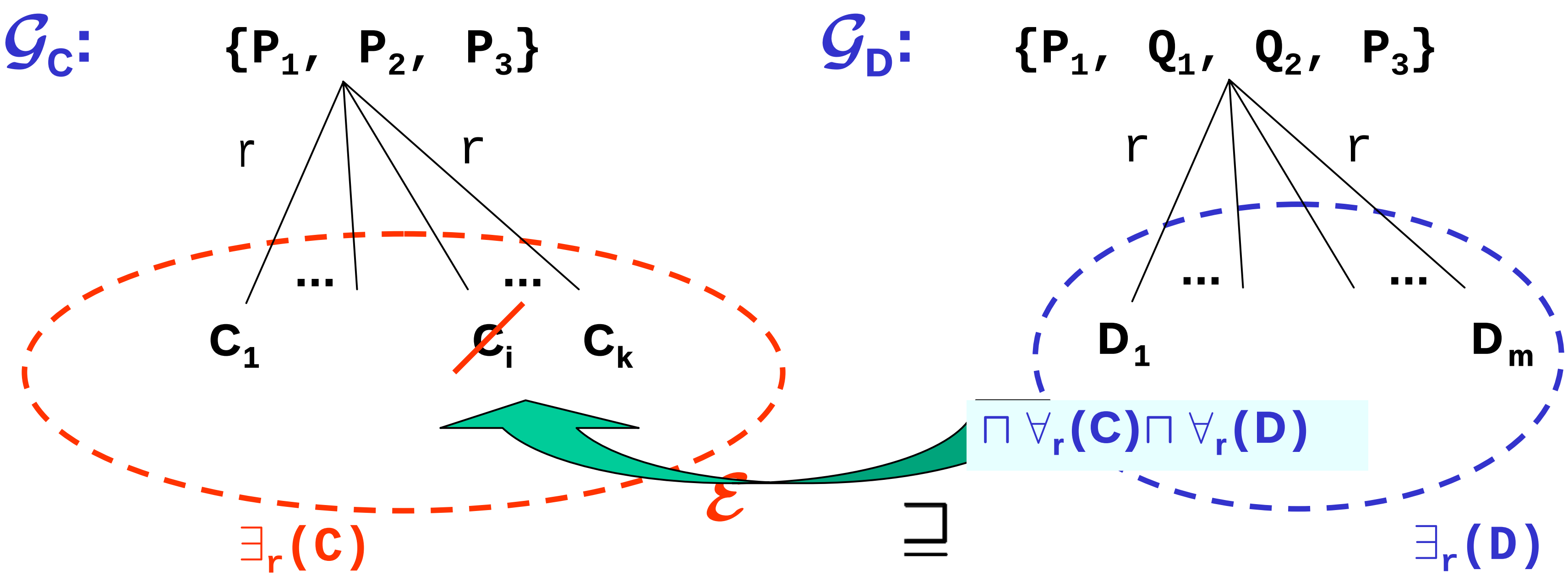
$$\mathsf{diff}(C, D) := P_2 \sqcap \forall r.\mathsf{diff}(\forall_r(C), \forall_r(D))$$

Syntactic difference algorithm



$$\text{diff}(C, D) := P_2 \sqcap \forall r.\text{diff}(\forall_r(C), \forall_r(D))$$

Syntactic difference algorithm



$$\text{diff}(C, D) := P_2 \sqcap \forall r.\text{diff}(\forall_r(C), \forall_r(D)) \sqcap \textstyle\prod_{E \in \mathcal{E}} \exists r.E$$

An example

$$C \doteq \exists r.P \sqcap \forall r.(P \sqcap Q) \sqcap \forall s.(\neg P \sqcap Q)$$

$$D \doteq \exists r.(P \sqcap \neg Q) \sqcap \forall r.P \sqcap \forall s.Q$$

Another philosophical question

- The link between the semantic and the syntactic operators?

Bibliography

- “The Description Logic Handbook: Theory, Implementation and Applications”, Cambridge: University Press, 2003
- R. Küsters, “Non-Standard Inferences in Description Logics", Springer-Verlag, 2001
- S. Brandt, R. Küsters and A.-Y. Turhan, "Approximation and Difference in Description Logics” in KR 2002.
- G. Teege, "Making the Difference: A Subtraction Operation for Description Logics” in KR 1994.