Introduction to Logic in Computer Science: Autumn 2006

Description Logics Crash Course

Andreas Witzel

Institute for Logic, Language and Computation
University of Amsterdam

Why Description Logics?

The goal of AI is to create systems with intelligent behaviour.

Crucial for this is

- acquiring knowledge about the world / application domain
- representing the knowledge
- reasoning with the knowledge

Conclusion: We need a logic formalism to represent knowledge and facilitate reasoning.

Outline

In this lecture, we will introduce one example of such a description logic formalism, see how to reason with it, and discuss some properties and modifications. The topics are:

- The Attributive Language with Complement ($\mathcal{ALC}$)
- Reasoning with $\mathcal{ALC}$
- $\mathcal{ALC}$ with Number Restrictions ($\mathcal{ALCN}$)
- Computational Issues
- Summary

Outline

Syntax of $\mathcal{ALC}$

Let N_C and N_R be disjoint sets of concept names and role names.

$\mathcal{ALC}$ concept terms are defined inductively:

1. Each concept name $\mathtt{A} \in N_C$ is an $\mathcal{ALC}$ concept term
2. If $\mathtt{C}, \mathtt{D}$ are $\mathcal{ALC}$ concept terms and $\mathtt{r} \in N_R$ is a role name, then the following are also $\mathcal{ALC}$ concept terms:
   - $\bot, \top$ (Bottom, Top)
   - $\mathtt{C} \sqcap \mathtt{D}, \mathtt{C} \sqcup \mathtt{D}, \neg\mathtt{C}$ (Boolean Operators)
   - $\forall\mathtt{r}.\mathtt{C}, \exists\mathtt{r}.\mathtt{C}$ (value restriction and existential restriction)

Example

- $\mathtt{Human} \sqcap \neg\mathtt{Female}$ — describes "male"
- $\exists\mathtt{has\text{-}child}.\top$ — describes "parent"
- $\mathtt{Human} \sqcap \forall\mathtt{has\text{-}child}.\mathtt{Female}$ — describes "humans having only daughters"

Semantics of $\mathcal{ALC}$

An interpretation $\mathcal{I}$ over a non-empty domain $\Delta^{\mathcal{I}}$ assigns

- to each concept name $\mathtt{A} \in N_C$ a subset $\mathtt{A}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
- to each role name $\mathtt{r} \in N_R$ a binary relation $\mathtt{r}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

and is inductively extended to all $\mathcal{ALC}$ concept terms:

$$
\begin{aligned}
\bot^{\mathcal{I}} &:= \emptyset \\
\top^{\mathcal{I}} &:= \Delta^{\mathcal{I}} \\
(\neg \mathtt{C})^{\mathcal{I}} &:= \Delta^{\mathcal{I}} \setminus \mathtt{C}^{\mathcal{I}} \\
(\mathtt{C} \sqcap \mathtt{D})^{\mathcal{I}} &:= \mathtt{C}^{\mathcal{I}} \cap \mathtt{D}^{\mathcal{I}} \\
(\mathtt{C} \sqcup \mathtt{D})^{\mathcal{I}} &:= \mathtt{C}^{\mathcal{I}} \cup \mathtt{D}^{\mathcal{I}} \\
(\forall \mathtt{r}.\mathtt{C})^{\mathcal{I}} &:= \left\{ d \in \Delta^{\mathcal{I}} | \forall e \in \Delta^{\mathcal{I}}.(d,e) \in \mathtt{r}^{\mathcal{I}} \Rightarrow e \in \mathtt{C}^{\mathcal{I}} \right\} \\
(\exists \mathtt{r}.\mathtt{C})^{\mathcal{I}} &:= \left\{ d \in \Delta^{\mathcal{I}} | \exists e \in \Delta^{\mathcal{I}}.(d,e) \in \mathtt{r}^{\mathcal{I}} \wedge e \in \mathtt{C}^{\mathcal{I}} \right\}
\end{aligned}
$$

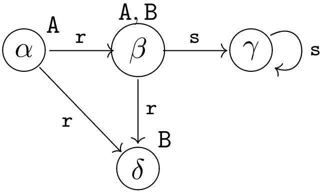
Semantics of $\mathcal{ALC}$

Example

For a language with concept names A, B and role names r, s, we consider an example interpretation $\mathcal{I}$ with $\Delta^{\mathcal{I}} := \{\alpha, \beta, \gamma, \delta\}$ and

$$\mathtt{A}^{\mathcal{I}} := \{\alpha, \beta\} \qquad \mathtt{r}^{\mathcal{I}} := \{(\alpha, \beta), (\alpha, \delta), (\beta, \delta)\}$$

$$\mathtt{B}^{\mathcal{I}} := \{\beta, \delta\} \qquad \mathtt{s}^{\mathcal{I}} := \{(\beta, \gamma), (\gamma, \gamma)\}$$



$$(\neg\mathtt{A})^{\mathcal{I}} = \{\gamma, \delta\} \qquad (\neg\mathtt{A} \sqcap \mathtt{B})^{\mathcal{I}} = \{\delta\} \qquad (\forall\mathtt{r}.\mathtt{B})^{\mathcal{I}} = \{\alpha, \beta, \gamma, \delta\}$$

$$(\forall\mathtt{r}.\mathtt{A})^{\mathcal{I}} = \{\gamma, \delta\} \qquad (\exists\mathtt{s}.\neg\mathtt{A})^{\mathcal{I}} = \{\beta, \gamma\} \qquad (\exists\mathtt{s}.\mathtt{A})^{\mathcal{I}} = \emptyset$$

Terminology (TBox): Syntax

In a realistic setting there will commonly be complex concepts, i.e. concepts built from simpler ones. We want to be able to give them abbreviating names, in this way defining the terminology of the setting.

A TBox allows us to do just that. It contains (finitely many) pairs of concept names and complex concept terms defining them.

Example (TBox)

$$\begin{bmatrix} \texttt{Male} \doteq \neg\texttt{Female} \\ \texttt{Woman} \doteq \texttt{Human} \sqcap \texttt{Female} \\ \texttt{Man} \doteq \texttt{Human} \sqcap \texttt{Male} \\ \texttt{Mother} \doteq \texttt{Woman} \sqcap \exists\texttt{has-child}.\texttt{Human} \\ \texttt{Father} \doteq \texttt{Man} \sqcap \exists\texttt{has-child}.\texttt{Human} \end{bmatrix}$$

Note: Multiple definitions and cycles are not allowed!

Terminology (TBox): Semantics

An interpretation $\mathcal{I}$ is a model of a TBox $\mathcal{T}$ iff we have:

$$\mathtt{A}^{\mathcal{I}} = \mathtt{C}^{\mathcal{I}} \qquad \text{for all } \mathtt{A} \doteq \mathtt{C} \in \mathcal{T}$$

Two TBoxes are equivalent iff they have the same models.

For every TBox $\mathcal{T}$ there is an equivalent unfolded TBox $\hat{\mathcal{T}}$ where only primitive concept names (i.e. names which are not themselves being defined in $\hat{\mathcal{T}}$) occur on right-hand sides.

An interpretation of the primitive concept names and role names in $\mathcal{T}$ can be uniquely extended to a model of $\mathcal{T}$.

Example

$$\begin{bmatrix} \mathtt{Woman} \doteq \mathtt{Human} \sqcap \mathtt{Female} \\ \mathtt{Mother} \doteq \mathtt{Human} \sqcap \mathtt{Female} \sqcap \exists \mathtt{has\text{-}child.Human} \end{bmatrix}$$

Note: This may result in exponential blowup!

Assertions (ABox): Syntax

Having fixed the terminology of a setting, we may want to assign names to individuals and make assertions about them.
That is what an ABox is used for. It contains (again finitely many) assertions over individual names N_I (disjoint from N_C and N_R).

$\mathtt{a} : \mathtt{C}$	(concept assertion)
$(\mathtt{a}, \mathtt{b}) : \mathtt{r}$	(role assertion)
$\mathtt{a} \neq \mathtt{b}$	(distinct individuals) $\rightsquigarrow$ generalized ABox

Example ($N_I = \{\mathtt{gunther}, \mathtt{gundula}, \mathtt{gisbert}\}$)

$$\left\{\begin{array}{r@{\,:\,}l} \mathtt{gunther} & \mathtt{Man} \\ \mathtt{gundula} & \mathtt{Woman} \\ \mathtt{gisbert} & \mathtt{Man} \\ (\mathtt{gunther}, \mathtt{gisbert}) & \mathtt{has\text{-}child} \\ (\mathtt{gundula}, \mathtt{gisbert}) & \mathtt{has\text{-}child} \end{array}\right\}$$

Assertions (ABox): Semantics

An interpretation $\mathcal{I}$ now additionally assigns, to each individual name $\mathtt{a} \in N_I$, an element $\mathtt{a}^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.

Remark: Normally, we make the unique name assumption, implicitly requiring that $\mathtt{a}^{\mathcal{I}} \neq \mathtt{b}^{\mathcal{I}}$ for any two (unequal) individual names $\mathtt{a}$ and $\mathtt{b}$. However, later on we need to use generalized ABoxes, where these statements are made explicitly.

An interpretation $\mathcal{I}$ is a model of an ABox $\mathcal{A}$ iff

$$
\begin{aligned}
\mathtt{a}^{\mathcal{I}} \in \mathtt{C}^{\mathcal{I}} \qquad & \text{for all } \mathtt{a} : \mathtt{C} \in \mathcal{A}, \text{ and} \\
(\mathtt{a}^{\mathcal{I}}, \mathtt{b}^{\mathcal{I}}) \in \mathtt{r}^{\mathcal{I}} \qquad & \text{for all } (\mathtt{a}, \mathtt{b}) : \mathtt{r} \in \mathcal{A}
\end{aligned}
$$

Given a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$, one is often interested in common models of $\mathcal{T}$ and $\mathcal{A}$.

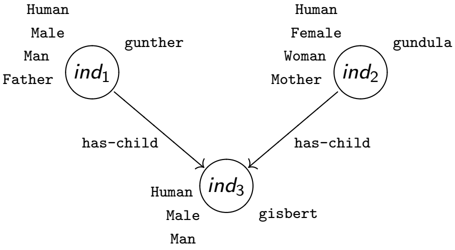
Example

$$\begin{bmatrix} \texttt{Male} \doteq \neg\texttt{Female} \\ \texttt{Woman} \doteq \texttt{Human} \sqcap \texttt{Female} \\ \texttt{Man} \doteq \texttt{Human} \sqcap \texttt{Male} \\ \texttt{Mother} \doteq \texttt{Woman} \sqcap \exists\texttt{has-child.Human} \\ \texttt{Father} \doteq \texttt{Man} \sqcap \exists\texttt{has-child.Human} \end{bmatrix} \qquad \begin{Bmatrix} \texttt{gunther} : \texttt{Man} \\ \texttt{gundula} : \texttt{Woman} \\ \texttt{gisbert} : \texttt{Man} \\ (\texttt{gunther}, \texttt{gisbert}) : \texttt{has-child} \\ (\texttt{gundula}, \texttt{gisbert}) : \texttt{has-child} \end{Bmatrix}$$

$$\begin{aligned} \Delta^{\mathcal{I}} := \texttt{Human}^{\mathcal{I}} &:= \{ind_1, ind_2, ind_3\} \\ \texttt{Female}^{\mathcal{I}} &:= \{ind_2\} \\ \texttt{has-child}^{\mathcal{I}} &:= \{(ind_1, ind_3), (ind_2, ind_3)\} \end{aligned} \qquad \begin{aligned} \texttt{gunther}^{\mathcal{I}} &:= ind_1 \\ \texttt{gundula}^{\mathcal{I}} &:= ind_2 \\ \texttt{gisbert}^{\mathcal{I}} &:= ind_3 \end{aligned}$$

This uniquely defines the remaining concepts:

$$\begin{aligned} \texttt{Male}^{\mathcal{I}} &= \{ind_1, ind_3\} \\ \texttt{Woman}^{\mathcal{I}} &= \{ind_2\} \\ \texttt{Man}^{\mathcal{I}} &= \{ind_1, ind_3\} \\ \texttt{Mother}^{\mathcal{I}} &= \{ind_2\} \\ \texttt{Father}^{\mathcal{I}} &= \{ind_1\} \end{aligned}$$



Outline

Terminological Reasoning Tasks

Purpose: Extract implicit terminological knowledge from explicitly given one

Terminological reasoning:

- *Satisfiability*: is there an interpretation $\mathcal{I}$ with $\mathtt{C}^{\mathcal{I}} \neq \emptyset$?
- *Satisfiability wrt $\mathcal{T}$*: is there a model $\mathcal{I}$ of $\mathcal{T}$ with $\mathtt{C}^{\mathcal{I}} \neq \emptyset$?
- *Subsumption ($C \sqsubseteq D$)*: is $\mathtt{C}^{\mathcal{I}} \subseteq \mathtt{D}^{\mathcal{I}}$ in all interpretations $\mathcal{I}$?
- *Subsumption wrt $\mathcal{T}$ ($C \sqsubseteq_{\mathcal{T}} D$)*: is $\mathtt{C}^{\mathcal{I}} \subseteq \mathtt{D}^{\mathcal{I}}$ in all models $\mathcal{I}$ of $\mathcal{T}$?

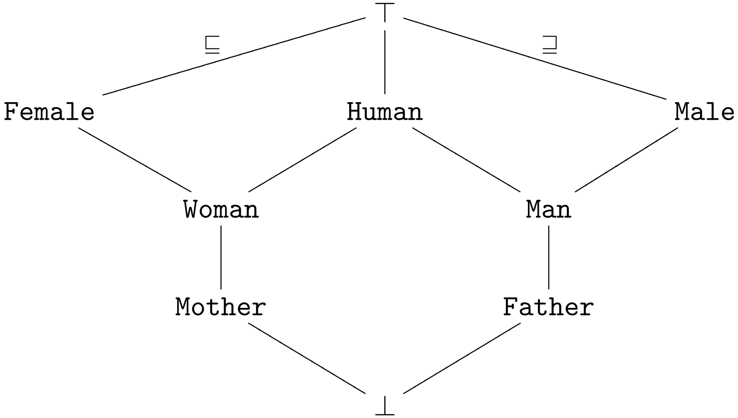
Example

With respect to the TBox from the earlier examples, we have

- $\mathtt{Mother} \sqcap \mathtt{Man}$ is not satisfiable (it can be unfolded to $\ldots \sqcap \mathtt{Female} \sqcap \ldots \sqcap \neg\mathtt{Female} \sqcap \ldots$)
- $\mathtt{Mother} \sqcap \exists\mathtt{has\text{-}child}.\neg\mathtt{Human}$ is satisfiable (this would be a $\mathtt{Woman}$ with a $\mathtt{Human}$ and a $\neg\mathtt{Human}$ $\mathtt{child}$...)
- $\mathtt{Mother} \sqsubseteq_{\mathcal{T}} \mathtt{Human}$
- $\mathtt{Man} \not\sqsubseteq_{\mathcal{T}} \mathtt{Father}$

TBox Classification

Description Logic systems usually offer various services, for example computing the concept hierarchy (wrt the subsumption relation) of all occurring concept names.



Assertional Reasoning Tasks

Purpose: Extract implicit knowledge about individuals from explicitly given one

Assertional reasoning:

- *Consistency*: does $\mathcal{A}$ have a model?
- *Consistency wrt $\mathcal{T}$*: do $\mathcal{A}$ and $\mathcal{T}$ have a common model?
- *Instance wrt $\mathcal{A}$ ($\mathcal{A} \models a : C$)*: is $\mathtt{a}^{\mathcal{I}} \in \mathtt{C}^{\mathcal{I}}$ in all models $\mathcal{I}$ of $\mathcal{A}$?
- *Instance wrt $\mathcal{A}$ and $\mathcal{T}$ ($\mathcal{T}, \mathcal{A} \models a : C$)*: is $\mathtt{a}^{\mathcal{I}} \in \mathtt{C}^{\mathcal{I}}$ in all common models $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$?

Example

With respect to $\mathcal{A}$ and $\mathcal{T}$ from the earlier examples, we have

- $\mathcal{A}$ is consistent and consistent wrt $\mathcal{T}$
- $\mathcal{A} \models \mathtt{gundula} : \mathtt{Woman}$
- $\mathcal{A} \not\models \mathtt{gundula} : \mathtt{Human}$
- $\mathcal{T}, \mathcal{A} \models \mathtt{gundula} : \mathtt{Human}$

Realization and Retrieval

Services offered by Description Logic systems in the context of ABoxes include:

- Realization of an individual name a: What are the (minimal) concept names of which a is an instance?

  $\texttt{gunther} \rightsquigarrow \texttt{Father}, \texttt{Man}, \texttt{Human}, \texttt{Male}$ (all)

  $\texttt{gunther} \rightsquigarrow \texttt{Father}$ (minimal)

- Retrieval: Which individuals are instances of C?

  $$\texttt{Mother} \rightsquigarrow \texttt{gundula}$$

  $$\texttt{Man} \sqcap \neg\texttt{Father} \rightsquigarrow \texttt{gisbert}$$

Reducing Reasoning Tasks

Instead of writing reasoners for each single reasoning task, one can try to reduce some tasks to other ones. For example,

Is concept C satisfiable?

can be reduced as follows:

$\Leftrightarrow$ *Is there a model $\mathcal{I}$ such that $C^{\mathcal{I}}$ is non-empty?*
$\Leftrightarrow$ *Is it <u>not</u> the case that for all models $\mathcal{I}$, $C^{\mathcal{I}}$ is empty?*
$\Leftrightarrow$ *Is concept C <u>not</u> subsumed by the empty concept $\bot$?*

That is, $\mathtt{C}$ is satisfiable iff $\mathtt{C} \not\sqsubseteq \bot$.

Similarly,

- $\mathtt{C} \sqsubseteq \mathtt{D}$ iff $\mathtt{C} \sqcap \neg\mathtt{D}$ is unsatisfiable
- $\mathtt{C}$ is satisfiable iff the ABox $\{\mathtt{a} : \mathtt{C}\}$ is consistent
- $\mathcal{A} \models \mathtt{a} : \mathtt{C}$ iff $\mathcal{A} \cup \{\mathtt{a} : \neg\mathtt{C}\}$ is inconsistent

The Reduction Hierarchy

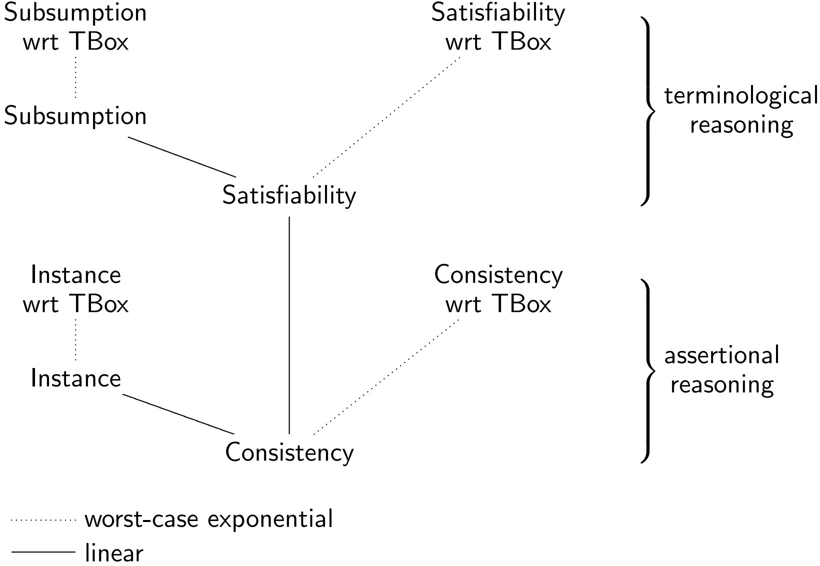


Tableau Algorithm

We present a tableau algorithm to decide ABox consistency.

Given an ABox $\mathcal{A}$ with unique name assumption, we

- drop the unique name assumption and add $\mathtt{a} \neq \mathtt{b}$ for all $\mathtt{a}, \mathtt{b} \in N_I$
- transform the resulting ABox into Negation Normal Form (i.e. negation occurs only in front of concept names)

We then exhaustively apply the tableau rules on the following slide. $\mathcal{A}$ is consistent iff there remains an open branch. As with earlier Tableau systems, such a branch can be used to build a model of the ABox.

In order to guarantee termination, rules cannot be applied if the resulting formulas are already on the branch under consideration (note the special case with $\exists$).

Tableau Rules

$\sqcap$ and $\sqcup$ rules:

$$\frac{\mathtt{a} : \mathtt{C} \sqcap \mathtt{D}}{\begin{array}{c}\mathtt{a} : \mathtt{C} \\ \mathtt{a} : \mathtt{D}\end{array}} \qquad \frac{\mathtt{a} : \mathtt{C} \sqcup \mathtt{D}}{\mathtt{a} : \mathtt{C} \mid \mathtt{a} : \mathtt{D}}$$

$\forall$ and $\exists$ rules:

$$\frac{\begin{array}{c}(\mathtt{a}, \mathtt{b}) : \mathtt{r} \\ \mathtt{a} : \forall \mathtt{r}.\mathtt{C}\end{array}}{\mathtt{b} : \mathtt{C}} \qquad \frac{\mathtt{a} : \exists \mathtt{r}.\mathtt{C}}{\begin{array}{c}(\mathtt{a}, \mathtt{c}) : \mathtt{r} \\ \mathtt{c} : \mathtt{C}\end{array}}$$

Closure rules:

$$\frac{\begin{array}{c}\mathtt{a} : \mathtt{C} \\ \mathtt{a} : \neg \mathtt{C}\end{array}}{\times} \qquad \frac{\mathtt{a} \neq \mathtt{a}}{\times}$$

In the $\exists$ rule, $\mathtt{c}$ is a new individual name unused in $\mathcal{A}$. Furthermore, the $\exists$ rule is not applicable if there is any individual name $\mathtt{b}$ such that $(\mathtt{a}, \mathtt{b}) : \mathtt{r}$ and $\mathtt{b} : \mathtt{C}$ are already on the branch.

Outline

Number Restrictions: Syntax and Semantics

Sometimes it is important not only to make statements about existence of roles, but also about their quantity. $\mathcal{ALCN}$ introduces number restrictions for any role name $\mathtt{r} \in N_R$:

- $(\geq n\ \mathtt{r})$ (at least restriction)
  Semantics: $(\geq n\ \mathtt{r})^{\mathcal{I}} := \left\{ d \in \Delta^{\mathcal{I}} \middle| \left|\{e \in \Delta^{\mathcal{I}} | (d,e) \in \mathtt{r}^{\mathcal{I}}\}\right| \geq n \right\}$
- $(\leq n\ \mathtt{r})$ (at most restriction)
  Semantics: $(\leq n\ \mathtt{r})^{\mathcal{I}} := \left\{ d \in \Delta^{\mathcal{I}} \middle| \left|\{e \in \Delta^{\mathcal{I}} | (d,e) \in \mathtt{r}^{\mathcal{I}}\}\right| \leq n \right\}$

$\mathcal{ALCN}$ concept terms are all concept terms built analogously to $\mathcal{ALC}$ concept terms, where now additionally these two new basic concept terms can be used.

Example

- $(\leq 3\ \mathtt{has\text{-}child}) \sqcap \mathtt{Woman}$ describes "mother of at most 3 children"
- $\exists \mathtt{has\text{-}child}.(\geq 2\ \mathtt{has\text{-}child})$ describes "grandparent of at least 2 siblings"

Number Restrictions: Tableau Rules

The tableau rules for number restrictions are a bit more involved. In particular, the $\leq$ rule requires renaming of individuals along the branch, which isn't really compatible with our notation. It also requires constraints on the rule application order to preserve termination.

For these reasons, we only consider the $\geq$ rule here.

$\geq$ rule:

$$\frac{\mathtt{a} : (\geq n\ \mathtt{r})}{\begin{array}{c}(\mathtt{a}, \mathtt{c}_1) : \mathtt{r} \\ \dots \\ (\mathtt{a}, \mathtt{c}_n) : \mathtt{r} \\ \mathtt{c}_i \neq \mathtt{c}_j \qquad (1 \leq i < j \leq n)\end{array}}$$

Again, the $\mathtt{c}_i$ are new individual names unused in $\mathcal{A}$. Furthermore, the rule is not applicable if there are individual names $\mathtt{b}_1, \dots, \mathtt{b}_n$ such that all $(\mathtt{a}, \mathtt{b}_i) : \mathtt{r}$ and all $\mathtt{b}_i \neq \mathtt{b}_j$ are already on the branch.

Outline

The Attributive Language with Complement ($\mathcal{ALC}$)

Reasoning with $\mathcal{ALC}$

$\mathcal{ALC}$ with Number Restrictions ($\mathcal{ALCN}$)

Computational Issues
- $\mathcal{ALC}$ is hard
- Useful Restrictions

Summary

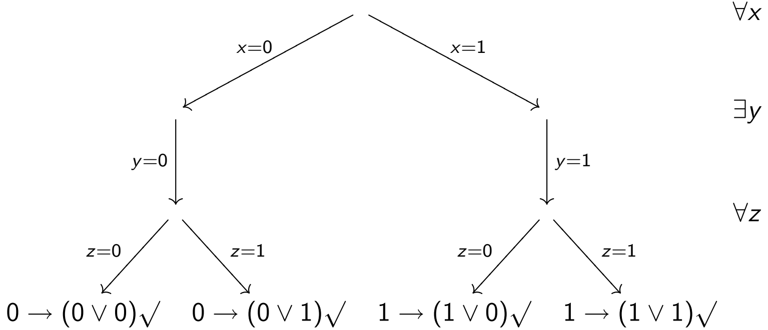
Quantified Boolean Formulas (QBF)

Idea: Reduce QBF validity to $\mathcal{ALC}$ concept satisfiability, thus showing **PSPACE**-hardness ('in **PSPACE**' also holds, but not shown here).

Reminder from last lecture: A QBF is a propositional formula preceded by either $\forall x$ or $\exists x$ for each occurring propositional variable x. For example:

$$\forall x \exists y \forall z. (x \rightarrow (y \vee z))$$

A QBF is valid iff it has a quantifier tree:



Reducing QBF validity to $\mathcal{ALC}$ concept satisfiability

Idea: Use $\mathcal{ALC}$ concepts to describe quantifier trees

Given: QBF $Q = Q_1 x_1 \dots Q_n x_n.\varphi$.

Find: $\mathtt{C}_Q$ which is satisfiable iff Q has a quantifier tree

Concepts: $\mathtt{X}_1, \dots, \mathtt{X}_n$ ($\hat{=}$ the x_i)
$\mathtt{L}_1, \dots, \mathtt{L}_n$ ($\hat{=}$ nodes of level i in the tree)

Roles: $\mathtt{r}$ ($\hat{=}$ edges from nodes to children)

We define $\mathtt{C}_Q := \mathtt{L}_1 \sqcap \forall \mathtt{r}.(\mathtt{L}_2 \sqcap \forall \mathtt{r}.(\mathtt{L}_3 \sqcap \dots \forall \mathtt{r}.(\mathtt{L}_n \sqcap \forall \mathtt{r}.\mathtt{C}_\varphi) \dots))$, where:

- $\mathtt{L}_i := \mathtt{D}_i \sqcap \begin{cases} \exists \mathtt{r}.\top & \text{if } Q_i = \exists \\ \exists \mathtt{r}.\mathtt{X}_i \sqcap \exists \mathtt{r}.\neg \mathtt{X}_i & \text{if } Q_i = \forall \end{cases}$
- $\mathtt{D}_i := \bigsqcap_{j<i} (\mathtt{X}_j \Rightarrow \forall \mathtt{r}.\mathtt{X}_j) \sqcap (\neg \mathtt{X}_j \Rightarrow \forall \mathtt{r}.\neg \mathtt{X}_j)$
- $\mathtt{Y} \Rightarrow \mathtt{Z} := \neg \mathtt{Y} \sqcup \mathtt{Z}$
- $\mathtt{C}_\varphi$ is obtained from φ by replacing all x_i by $\mathtt{X}_i$, $\wedge$ by $\sqcap$, $\vee$ by $\sqcup$

Useful Restrictions

If we want to guarantee reasoning tasks to be tractable, we can consider sub-Boolean fragments of $\mathcal{ALC}$. These typically allow for conjunction, but prohibit disjunction and/or negation.

Two examples for such logics are:

- $\mathcal{FL}_0$, featuring $\sqcap$, $\forall$ and $\top$
- $\mathcal{EL}$, featuring $\sqcap$, $\exists$ and $\top$

Satisfiability for such logics without negation is often trivial (i.e. all concept terms are satisfiable). Therefore, we are now interested in other notions such as least common subsumers (LCS).

LCS are intended to describe the commonalities of concept terms. We call $\mathtt{E}$ the least common subsumer of $\mathtt{C}$ and $\mathtt{D}$ if

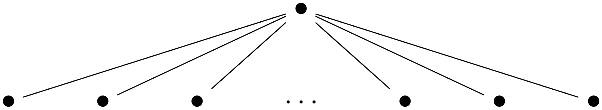
(i) $\mathtt{C} \sqsubseteq \mathtt{E}$ and $\mathtt{D} \sqsubseteq \mathtt{E}$

(ii) $\mathtt{E} \sqsubseteq \mathtt{F}$ for all $\mathtt{F}$ with $\mathtt{C} \sqsubseteq \mathtt{F}$ and $\mathtt{D} \sqsubseteq \mathtt{F}$

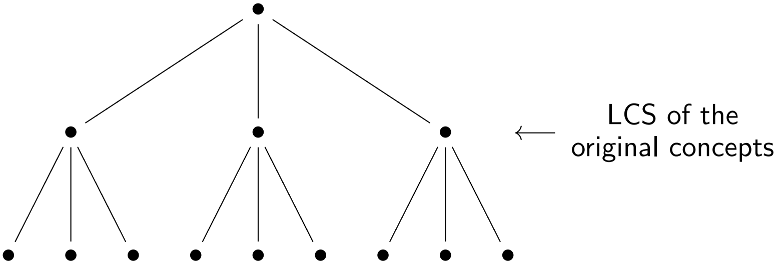
In $\mathcal{FL}_0$ and $\mathcal{EL}$, the LCS always exists.

Least Common Subsumer Application Example

The LCS can be used to add more structure to "flat" knowledge bases by introducing "meaningful" intermediate concepts into the concept hierarchy.



becomes



Outline

- The Attributive Language with Complement ($\mathcal{ALC}$)
- Reasoning with $\mathcal{ALC}$
- $\mathcal{ALC}$ with Number Restrictions ($\mathcal{ALCN}$)
- Computational Issues
- Summary

Summary

We have introduced and discussed Description Logics, in particular

- Syntax and Semantics of $\mathcal{ALC}$ as an exemplary language
- TBoxes and ABoxes for terminology and assertions
- Common reasoning tasks
- Tableau rules
- Computational complexity of $\mathcal{ALC}$
- An extension and two restrictions of $\mathcal{ALC}$

Related topics include

- Historical approaches (semantic networks, frames, non-monotonic inheritance networks)
- First description logic system (KL-ONE)
- Modal Logic ($\mathcal{ALC}$ is a syntactic variant of K_n)
- State of the art (OWL, FaCT, RACER, KAON 2)

Based on "Logic-based Knowledge Representation" by F. Baader.
`http://lat.inf.tu-dresden.de/teaching/ss2006/lbkr/`