Logics for Data and Knowledge Representation

5. Reasoning in $\mathcal{ALC}$

Luciano Serafini

FBK-irst, Trento, Italy

October 14, 2012

$\mathcal{ALC}$ Basic Inference Problems

The basic inference problems on concepts and T-boxes are the following:

Concept subsumption

C is subsumed by D, or equivalently, D subsumes C, in symbols $\models C \sqsubseteq D$, if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all interpretations $\mathcal{I}$

$\mathcal{ALC}$ Basic Inference Problems

The basic inference problems on concepts and T-boxes are the following:

Concept subsumption

C is subsumed by D, or equivalently, D subsumes C, in symbols $\models C \sqsubseteq D$, if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all interpretations $\mathcal{I}$

Concept Subsumption w.r.t. T-Box

C is subsumed by D w.r.t., T-box $\mathcal{T}$, or equivalently, D subsumes C in $\mathcal{T}$, in symbols $\models C \sqsubseteq_{\mathcal{T}} D$, (an alternative notation $\mathcal{T} \models C \sqsubseteq D$) if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all interpretations $\mathcal{I}$ that satisfies $\mathcal{T}$.

$\mathcal{ALC}$ Basic Inference Problems

The basic inference problems on concepts and T-boxes are the following:

Concept subsumption

C is subsumed by D, or equivalently, D subsumes C, in symbols $\models C \sqsubseteq D$, if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all interpretations $\mathcal{I}$

Concept consistency

C is consistent if and only if there exists an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$.

Concept Subsumption w.r.t. T-Box

C is subsumed by D w.r.t., T-box $\mathcal{T}$, or equivalently, D subsumes C in $\mathcal{T}$, in symbols $\models C \sqsubseteq_{\mathcal{T}} D$, (an alternative notation $\mathcal{T} \models C \sqsubseteq D$) if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all interpretations $\mathcal{I}$ that satisfies $\mathcal{T}$.

$\mathcal{ALC}$ Basic Inference Problems

The basic inference problems on concepts and T-boxes are the following:

Concept subsumption

C is subsumed by D, or equivalently, D subsumes C, in symbols $\models C \sqsubseteq D$, if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all interpretations $\mathcal{I}$

Concept Subsumption w.r.t. T-Box

C is subsumed by D w.r.t., T-box $\mathcal{T}$, or equivalently, D subsumes C in $\mathcal{T}$, in symbols $\models C \sqsubseteq_{\mathcal{T}} D$, (an alternative notation $\mathcal{T} \models C \sqsubseteq D$) if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all interpretations $\mathcal{I}$ that satisfies $\mathcal{T}$.

Concept consistency

C is consistent if and only if there exists an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$.

Concept consistency w.r.t a Tbox

C is consistent w.r.t. $\mathcal{T}$ if and only if there a model $\mathcal{I}$ of $\mathcal{T}$ with $C^{\mathcal{I}} \neq \emptyset$

$\mathcal{ALC}$ Basic Inference Problems

The basic inference problems on concepts and T-boxes are the following:

Concept subsumption

C is subsumed by D, or equivalently, D subsumes C, in symbols $\models C \sqsubseteq D$, if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all interpretations $\mathcal{I}$

Concept Subsumption w.r.t. T-Box

C is subsumed by D w.r.t., T-box $\mathcal{T}$, or equivalently, D subsumes C in $\mathcal{T}$, in symbols $\models C \sqsubseteq_{\mathcal{T}} D$, (an alternative notation $\mathcal{T} \models C \sqsubseteq D$) if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all interpretations $\mathcal{I}$ that satisfies $\mathcal{T}$.

Concept consistency

C is consistent if and only if there exists an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$.

Concept consistency w.r.t a Tbox

C is consistent w.r.t. $\mathcal{T}$ if and only if there a model $\mathcal{I}$ of $\mathcal{T}$ with $C^{\mathcal{I}} \neq \emptyset$

Consistency of a T-box

A T-box $\mathcal{T}$ is consistent, if there is an interpretation $\mathcal{I}$ that satisfies $\mathcal{T}$, i.e., $\mathcal{I} \models \mathcal{T}$.

$\mathcal{ALC}$ Dependencies between basic inference problems

Concept subsumption ⇔ concept consistency

$$\models C \sqsubseteq D \iff C \sqcap \neg D \text{ is not consistent} \tag{1}$$

$$\mathcal{T} \models C \sqsubseteq D \iff C \sqcap \neg D \text{ is not consistent w.r.t., } \mathcal{T} \tag{2}$$

Proof.

We prove property (2). Indeed (1) is a special case of (2) with $\mathcal{T} = \emptyset$.

$$\begin{aligned}
\mathcal{T} \models C \sqsubseteq D &\iff \text{for all } \mathcal{I} \text{ such that } \mathcal{I} \models \mathcal{T},\ C^{\mathcal{I}} \subseteq D^{\mathcal{I}} \\
&\iff \text{for all } \mathcal{I} \text{ s.t. } \mathcal{I} \models \mathcal{T},\ (C \sqcap \neg D)^{\mathcal{I}} = \emptyset \\
&\iff \text{there is no } \mathcal{I} \models \mathcal{T},\ (C \sqcap \neg D)^{\mathcal{I}} \neq \emptyset \\
&\iff C \sqcap \neg D \text{ is not satisfiable in } \mathcal{T}
\end{aligned}$$

□

Dependencies between basic inference problems

Concept consistency w.r.t., T-box ⇔ T-box consistency

$$C \text{ is consistent w.r.t. } \mathcal{T} \Longleftrightarrow \mathcal{T} \cup \{\exists P_{new}.C\} \text{ is consistent} \quad (3)$$

Where P_{new} is a "fresh" role, i.e., a role symbol not appearing in $\mathcal{T}$

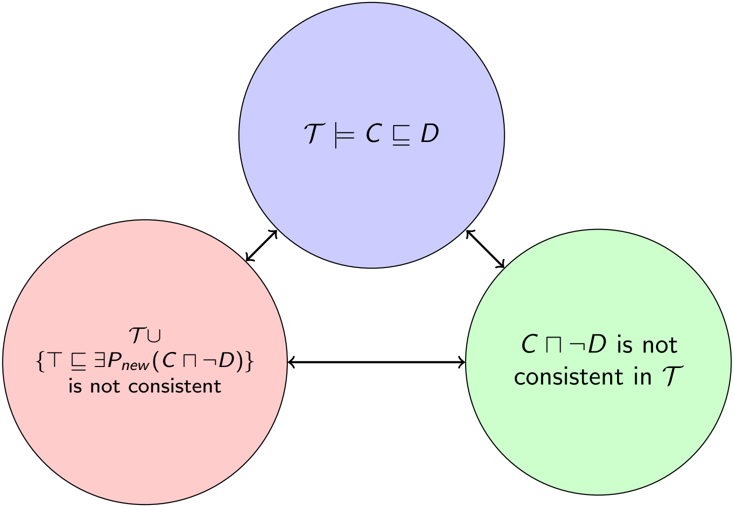
Proof.

$\Longrightarrow$ If C is consistent w.r.t. $\mathcal{T}$, there is an interpretation $\mathcal{I}$ that satisfies C and such that $C^{\mathcal{I}} \neq \emptyset$. Let $\mathcal{I}'$ be the extension of $\mathcal{I}$ where $(P_{new}) = \Delta \times C^{\mathcal{I}}$. Since $C^{\mathcal{I}}$ is not empty we have that for all $d \in \Delta^{\mathcal{I}}$ there is a $d' \in C^{\mathcal{I}}$ such that $(d, d') \in (P_{new})^{\mathcal{I}}$, this implies that $d \in (\exists P_{new}.C$. Since this holds for every $d \in \Delta^{\mathcal{I}}$, we have that $\mathcal{I} \models \top \sqsubseteq \exists P_{new}.C$, and therefore $\mathcal{I}$ is a model for $\mathcal{T} \cup \{\top \sqsubseteq \exists P_{new}.C\}$.

$\Longleftarrow$ If $\mathcal{T} \cup \{\top \sqsubseteq \exists P_{new}.C\}$ is consistent then there is a model $\mathcal{I}$ that satisfies $\top \sqsubseteq \exists P_{new}.C$. Since $\top^{\mathcal{I}}$ is not empty, this implies that there is a $d \in \exists P_{new}.C$, which implies that there is a d', with $(d, d') \in P_{new}$ and $d' \in C^{\mathcal{I}}$, i.e., C is consistent.

□

Dependencies between basic inference problems



(un)satisfiability general properties - exercises

Exercise

Show that $\models C \sqsubseteq D$ implies $\models \exists R.C \sqsubseteq \exists R.D$

(un)satisfiability general properties - exercises

Exercise

Show that $\models C \sqsubseteq D$ implies $\models \exists R.C \sqsubseteq \exists R.D$

Solution

We have to prove that for all $\mathcal{I}$, $(\exists R.C)^{\mathcal{I}} \subseteq (\exists R.C)^{\mathcal{I}}$ under the hypothesis that for all $\mathcal{I}$, $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

- *Let $x \in (\exists R.C)^{\mathcal{I}}$, we want to show that x is also in $(\exists R.D)^{\mathcal{I}}$.*
- *If $x \in (\exists R.C)^{\mathcal{I}}$, then by the interpretation of $\exists R$ there must be an y with $(x, y) \in R^{\mathcal{I}}$ such that $y \in C^{\mathcal{I}}$.*
- *By the hypothesis that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all $\mathcal{I}$, we have thaty $\in D^{\mathcal{I}}$.*
- *The fact that $(x, y) \in R^{\mathcal{I}}$ and $y \in D^{\mathcal{I}}$ implies that $x \in (\exists R.D)^{\mathcal{I}}$.*

$\mathcal{ALC}$ (un)satisfiability and validity - exercises

Exercise

For each of the following formula say if it is valid, satisfiable or unsatisfiable. If it is not valid provide a model that falsify it.

$$\forall R(A \sqcap B) \equiv \forall RA \sqcap \forall RB$$
$$\forall R(A \sqcup B) \equiv \forall RA \sqcup \forall RB$$
$$\exists R(A \sqcap B) \equiv \exists RA \sqcap \exists RB$$
$$\exists R(A \sqcup B) \equiv \exists RA \sqcup \exists RB$$

$\mathcal{ALC}$ (un)satisfiability and validity - exercises

Exercise

For each of the following formula say if it is valid, satisfiable or unsatisfiable. If it is not valid provide a model that falsify it.

$$\forall R(A \sqcap B) \equiv \forall RA \sqcap \forall RB$$
$$\forall R(A \sqcup B) \equiv \forall RA \sqcup \forall RB$$
$$\exists R(A \sqcap B) \equiv \exists RA \sqcap \exists RB$$
$$\exists R(A \sqcup B) \equiv \exists RA \sqcup \exists RB$$

Solution

$\forall R(A \sqcap B) \equiv \forall RA \sqcup \forall RB$ *is valid and we can prove that* $(\forall R(A \sqcap B))^{\mathcal{I}} = (\forall R.A \sqcap \forall R.B)^{\mathcal{I}}$ *for all interpretations* $\mathcal{I}$.

$$\begin{aligned}
(\forall R(A \sqcap B))^{\mathcal{I}} &= \{(x,y) \in R^{\mathcal{I}} \mid y \in (A \sqcap B)^{\mathcal{I}}\} \\
&= \{(x,y) \in R^{\mathcal{I}} \mid y \in A^{\mathcal{I}} \cap B^{\mathcal{I}}\} \\
&= \{(x,y) \in R^{\mathcal{I}} \mid y \in A^{\mathcal{I}}\} \cap \{(x,y) \in R^{\mathcal{I}} \mid y \in B^{\mathcal{I}}\} \\
&= (\forall R.A)^{\mathcal{I}} \cap (\forall R.B)^{\mathcal{I}} \\
&= (\forall R.A \sqcap \forall R.B)^{\mathcal{I}}
\end{aligned}$$

$\mathcal{ALC}$ (un)satisfiability and validity - exercises

Exercise

For each of the following formula say if it is valid, satisfiable or unsatisfiable. If it is not valid provide a model that falsify it.

$$\forall R(A \sqcap B) \equiv \forall RA \sqcap \forall RB$$

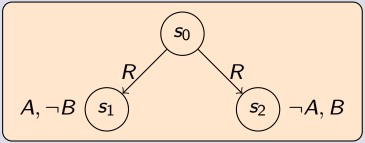
$$\forall R(A \sqcup B) \equiv \forall RA \sqcup \forall RB$$

$$\exists R(A \sqcap B) \equiv \exists RA \sqcap \exists RB$$

$$\exists R(A \sqcup B) \equiv \exists RA \sqcup \exists RB$$

Solution

$\forall R(A \sqcup B) \equiv \forall RA \sqcup \forall RB$ is not valid. The following model is such that $(\forall R(A \sqcup B))^{\mathcal{I}} \neq (\forall RA \sqcup \forall RB)^{\mathcal{I}}$



- *$s_0 \in (\forall R(A \sqcup B))^{\mathcal{I}}$ but*
- *$s_0 \notin (\forall RA)$ and*
- *$s_0 \notin (\forall RB)^{\mathcal{I}}$*

However notice that the containment: $\forall R.A \sqcup \forall R.B \sqsubseteq \forall R.(A \sqcup B)$ is valid

$\mathcal{ALC}$ (un)satisfiability and validity - exercises

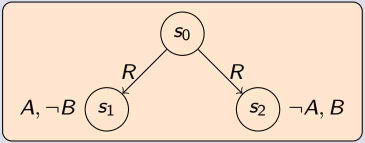
Exercise

For each of the following formula say if it is valid, satisfiable or unsatisfiable. If it is not valid provide a model that falsify it.

$$\forall R(A \sqcap B) \equiv \forall RA \sqcap \forall RB$$
$$\forall R(A \sqcup B) \equiv \forall RA \sqcup \forall RB$$
$$\exists R(A \sqcap B) \equiv \exists RA \sqcap \exists RB$$
$$\exists R(A \sqcup B) \equiv \exists RA \sqcup \exists RB$$

Solution

$\exists R(A \sqcap B) \equiv \exists RA \sqcap \exists RB$ is not valid. The following model is such that $(\exists R(A \sqcap B))^{\mathcal{I}} \neq (\exists RA \sqcap \forall RB)^{\mathcal{I}}$



- $s_0 \in (\exists RA)^{\mathcal{I}}$ *and*
- $s_0 \in (\exists RB)^{\mathcal{I}}$ *but*
- $s_0 \notin (\exists R(A \sqcap B))^{\mathcal{I}}$

However notice that the containment: $\exists R(A \sqcap B) \sqsubseteq \exists RA \sqcap \exists RB$ is valid

$\mathcal{ALC}$ (un)satisfiability and validity - exercises

Exercise

For each of the following formula say if it is valid, satisfiable or unsatisfiable. If it is not valid provide a model that falsify it.

$$\forall R(A \sqcap B) \equiv \forall RA \sqcap \forall RB$$
$$\forall R(A \sqcup B) \equiv \forall RA \sqcup \forall RB$$
$$\exists R(A \sqcap B) \equiv \exists RA \sqcap \exists RB$$
$$\exists R(A \sqcup B) \equiv \exists RA \sqcup \exists RB$$

Solution

$\exists R(A \sqcup B) \equiv \exists RA \sqcup \exists RB$ *is valid. We can provide a proof similar to the case of* $\forall R.(A \sqcap B) \equiv \forall R.A \sqcap \forall R.B$*, but in the following we provide an alternative proof, which is based on other equivalences:*

$$\begin{aligned} \exists R(A \sqcup B) &\equiv \neg\forall R(\neg(A \sqcup B)) \\ &\equiv \neg\forall R.(\neg A \sqcap \neg B) \\ &\equiv \neg(\forall R.(\neg A) \sqcap \forall R.(\neg B)) \\ &\equiv \neg(\forall R.(\neg A) \sqcup \neg\forall R.(\neg B) \\ &\equiv \exists R.A \sqcup \exists R.B \end{aligned}$$

$\mathcal{ALC}$ (un)satisfiability and validity - exercises

Exercise

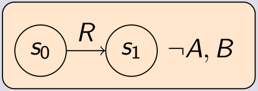
For each of the following concept say if it is valid, satisfiable or unsatisfiable. If it is valid, or unsatisfiable, provide a proof. If it is satisfiable (and not valid) then exhibit a model that interprets the concept in a non-empty set

1. $\neg(\forall R.A \sqcup \exists R.(\neg A \sqcap \neg B))$
2. $\exists R.(\forall S.C) \sqcap \forall R.(\exists S.\neg C)$
3. $(\exists S.C \sqcap \exists S.D) \sqcap \forall S.(\neg C \sqcup \neg D)$
4. $\exists S.(C \sqcap D) \sqcap (\forall S.\neg C \sqcup \exists S.\neg D)$
5. $C \sqcap \exists R.A \sqcap \exists R.B \sqcap \neg\exists R.(A \sqcap B)$

$\mathcal{ALC}$ (un)satisfiability and validity - exercises

Solution

1. $\neg(\forall R.A \sqcup \exists R.(\neg A \sqcap \neg B))$ *Satisfiable*



$$s_0 \in (\neg(\forall R.A \sqcup \exists R.(\neg A \sqcap \neg B))^{\mathcal{I}}$$
$$s_1 \notin (\neg(\forall R.A \sqcup \exists R.(\neg A \sqcap \neg B))^{\mathcal{I}}$$

2. $\exists R.(\forall S.C) \sqcap \forall R.(\exists S.\neg C)$ *unsatisfiable, since* $\exists R.\forall S.C \equiv \neg\forall R.\neg\forall S.C \equiv \neg\forall R.\exists S.\neg C$. *This implies that* $\exists R.(\forall S.C) \sqcap \forall R.(\exists S.\neg C)$ *is equivalent to* $\neg(\forall R.\exists S.\neg C) \sqcap (\forall R.\exists S.\neg C)$, *which is a concept of the form* $\neg B \sqcap B$ *which is always unsatisfiable.*
3. $(\exists S.C \sqcap \exists S.D) \sqcap \forall S.(\neg C \sqcup \neg D)$ *satisfiable*
4. $\exists S.(C \sqcap D) \sqcap (\forall S.\neg C \sqcup \exists S.\neg D)$ *unsatisfiable*
5. $C \sqcap \exists R.A \sqcap \exists R.B \sqcap \neg\exists R.(A \sqcap B)$ *satisfiable*

$\mathcal{ALC}$ Basic Inference problems with A-boxes

Consistency of an A-Box $\mathcal{A}$

The A-box $\mathcal{A}$ is consistent if and only if is there a model $\mathcal{I}$ that satisfies $\mathcal{A}$, i.e., $\mathcal{I} \models \mathcal{A}$.

$\mathcal{ALC}$ Basic Inference problems with A-boxes

Consistency of an A-Box $\mathcal{A}$

The A-box $\mathcal{A}$ is consistent if and only if is there a model $\mathcal{I}$ that satisfies $\mathcal{A}$, i.e., $\mathcal{I} \models \mathcal{A}$.

Consistency of a knowledge base $\mathcal{K}$

The A-box $\mathcal{A}$ is consistent w.r.t., the T-box $\mathcal{T}$ if and only if is there a model $\mathcal{I}$ of $\mathcal{T}$ that satisfies $\mathcal{A}$, i.e., there is a $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$.

Consistency of a knowledge base $\mathcal{K}$

The assertion $C(a)$ (resp, $R(a, b)$) is a logical consequence of the knowledge base $\mathcal{K}$, in symbols $\mathcal{K} \models C(a)$ if for all interpretations $\mathcal{I}$ that satisfies $\mathcal{K}$, then $\mathcal{I} \models C(a)$ (resp, $\mathcal{I} \models R(a, b)$)

$\mathcal{ALC}$ Complex Inference Tasks

Concept hierarchy

The subsumption hierarchy of $\mathcal{T}$, is a partial order on the set of primitive concepts defined as follows:

$$\{A \prec B | A, B \in \Sigma_C \text{ and } A \sqsubseteq_{\mathcal{T}} B\}$$

Individual classification

For all individual $o \in \Sigma_I$ determine all the primitive concepts $A \in \Sigma_C$, such that $\mathcal{T} \models A(o)$.

Negation Normal Form

Definition

A concept C is in negation normal form (NNF) if the $\neg$ operator is applied only to atomic concepts

Lemma

Every concept C can be reduced in an equivalent concept in NNF.

proof

A concept C can be reduced in NNF by the following rewriting rules that push inside the $\neg$ operator:

$$
\begin{aligned}
\neg(C \sqcap D) &\equiv \neg C \sqcup \neg D \\
\neg(C \sqcup D) &\equiv \neg C \sqcap \neg D \\
\neg(\neg C) &\equiv C \\
\neg\forall R.C &\equiv \exists R.\neg C \\
\neg\exists R.C &\equiv \forall R.\neg C
\end{aligned}
$$

Checking satisfiability of a concept in $\mathcal{ALC}$

Tableaux

Let C_0 be an $\mathcal{ALC}$-concept in NNF. In order to test satisfiability of C_0, the algorithm starts with $\mathcal{A}_0 := \{C_0(x_0)\}$, and applies the following rules:

Rule	Condition	$\longrightarrow$	Effect
$\rightarrow_\sqcap$	$C_1 \sqcap C_2(x) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{C_1(x), C_2(x)\}$
$\rightarrow_\sqcup$	$C_1 \sqcup C_2(x) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{C_1(x)\}$ or $\mathcal{A} \cup \{C_2(x)\}$
$\rightarrow_\exists$	$\exists R.C(x) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{R(x,y), C(y)\}$
$\rightarrow_\forall$	$\forall R.C(x), R(x,y) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{C(y)\}$

Every rule is applicable only if it has an effect on $\mathcal{A}$, i.e., if it adds some new assertion; otherwise it's not applicable.

Checking satisfiability of a concept in $\mathcal{ALC}$

Definition

An ABox $\mathcal{A}$

- is complete iff none of the transformation rules applies to it.
- has a clash iff $\{C(x), \neg C(x)\} \subseteq \mathcal{A}$
- is closed if it contains a clash
- is open if it is not closed

Checking satisfiability of a concept in $\mathcal{ALC}$

Lemma

- *There cannot be an infinite sequence of rule applications*

$$\{C_0(x_0)\} \rightarrow \mathcal{A}_1 \rightarrow \mathcal{A}_2 \rightarrow \dots$$

- *If $\mathcal{A}'$ is obtained by applying a deterministic rule to $\mathcal{A}$, then $\mathcal{A}$ is consistent iff $\mathcal{A}'$ is consistent*
- *If $\mathcal{A}'$ and $\mathcal{A}''$ can be obtained by applying a non-deterministic rule to $\mathcal{A}$, then $\mathcal{A}$ is consistent iff either $\mathcal{A}'$ or $\mathcal{A}''$ are consistent*
- *Any closed ABox $\mathcal{A}$ is inconsistent.*
- *Any complete and open ABox $\mathcal{A}$ is consistent.*

Canonical model

Satisfiability of complete and open A-box

To show item 5 of previous lemma, we describe a method for generating an interpretation $\mathcal{I}_\mathcal{A}$ starting from a complete and closed A-box $\mathcal{A}$. This model is called Canonical interpretation

Canonical interpretation $\mathcal{I}_\mathcal{A}$

1. $\Delta^{\mathcal{I}_\mathcal{A}} = \{x | \text{either } C(x),\ r(x, y), \text{ or } r(y, x) \in \mathcal{A}\}$
2. $A^{\mathcal{I}_\mathcal{A}} = \{x | A(x) \in \mathcal{A}\}$
3. $R^{\mathcal{I}_\mathcal{A}} = \{(x, y) | R(x, y) \in \mathcal{A}\}$.

Theorem

It is decidable whether or not an $\mathcal{ALC}$-concept is satisfiable

Complexity of reasoning in $\mathcal{ALC}$

Exercise

Consider the concept C_n inductively defined as follows;

$$\begin{aligned} C_1 &= \exists R.A \sqcup \exists R.\neg A \\ C_{n+1} &= \exists R.A \sqcup \exists R.\neg A \sqcap \forall R.C_n \end{aligned}$$

Check the form of the canonical interpretation of the A-box generated starting form $\{C_n(x_0)\}$.

Solution

Given the input description C_n the satisfiability algorithm generates a complete and open ABox whose canonical interpretation is a binary tree of depth n, and thus consists of $2^{n+1} - 1$ individuals.

So in principle the complexity of checking sat in $\mathcal{ALC}$ is exponential in space

Complexity of reasoning in $\mathcal{ALC}$

Theorem

Satisfiability of $\mathcal{ALC}$ concepts is PSPACE*-complete.*

Proof sketch of membership in PSPACE.

We show that if an $\mathcal{ALC}$-concept is satisfiable, we can construct a model using only polynomial space.

- Since PSSPACE $=$ NPSPACE, we consider a non-deterministic algorithm that for each application of the $\rightarrow_{\sqcup}$-rule, chooses the "correct" direction
- Then, the tree model property of $\mathcal{ALC}$ implies that the different branches of the tree model to be constructed by the algorithm can be explored separately as follows:
  1. Apply the $\rightarrow_{\sqcap}$ and $\rightarrow_{\sqcup}$ rules exhaustively, and check for clashes.
  2. Choose a node x and exhaustively apply the $\rightarrow_{\exists}$-rule to generate all necessary direct successors of x.
  3. Exhaustively apply the $\rightarrow_{\forall}$ rule to propagate concepts to the newly

Exercises: Satisfiability in $\mathcal{ALC}$

Exercise

Check the satisfiability of the following concepts:

1. $\neg(\forall R.A \sqcup \exists R.(\neg A \sqcap \neg B))$
2. $\exists R.(\forall S.C) \sqcap \forall R.(\exists S.\neg C)$
3. $(\exists S.C \sqcap \exists S.D) \sqcap \forall S.(\neg C \sqcup \neg D)$
4. $\exists S.(C \sqcap D) \sqcap (\forall S.\neg C \sqcup \exists S.\neg D)$
5. $C \sqcap \exists R.A \sqcap \exists R.B \sqcap \neg\exists R.(A \sqcap B)$

$\mathcal{ALC}$ Tableaux - exercise

Exercise

Check by means of tableaux, if the following subsumption is valid

$$\neg\forall R.A \sqcap \forall R((\forall R.B) \sqcup A) \sqsubseteq \forall R.\neg(\exists R.A) \sqcup \exists R.(\exists R.B)$$

Solution

- *to check subsumption of $C \sqsubseteq D$, we check inconsistency of $C \sqcup \neg D$, i.e., inconsistency of*

$$\neg\forall R.A \sqcap \forall R((\forall R.B) \sqcup A) \sqcap \neg(\forall R.\neg(\exists R.A) \sqcup \exists R.(\exists R.B)) \quad (4)$$

- *First we transform* (4) *in NNF, as follows:*

$$\exists R.\neg A \sqcap \forall R(\forall R.B \sqcup A) \sqcap (\exists R.\exists R.A \sqcap \forall R.\forall R.\neg B)$$

$\mathcal{ALC}$ Tableaux - exercise

Solution

	$\exists R.\neg A \sqcap \forall R(\forall R.B \sqcup A) \sqcap (\exists R.\exists R.A \sqcap \forall R.\forall R.\neg B)(x_0)$	(5)
(5) $\rightarrow_\sqcap$	$\exists R.\neg A(x_0)$	(6)
	$\forall R(\forall R.B \sqcup A)(x_0)$	(7)
	$\exists R.\exists R.A(x_0)$	(8)
	$\forall R.\forall R.\neg B(x_0)$	(9)
(6) $\rightarrow_\exists$	$R(x_0, x_1)$	(10)
	$\neg A(x_1)$	(11)
(10), (7) $\rightarrow_\forall$	$(\forall R.B \sqcup A)(x_1)$	(12)
(10), (9) $\rightarrow_\forall$	$\forall R\neg B(x_1)$	(13)
(12) $\rightarrow_\sqcup$	$\forall R.B(x_1)$	(14)
(8) $\rightarrow_\exists$	$R(x_0, x_2)$	(15)
	$\exists R.A(x_2)$	(16)
(15), (7) $\rightarrow_\forall$	$(\forall R.B \sqcup A)(x_2)$	(17)
(17) $\rightarrow_\sqcup$	$\forall R.B(x_2)$	(18)
(15), (9) $\rightarrow_\forall$	$\forall R\neg B(x_2)$	(19)
(16) $\rightarrow_\exists$	$R(x_2, x_3)$	(20)
	$A(x_3)$	(21)
(20), (18) $\rightarrow_\forall$	$B(x_3)$	(22)
(20), (19) $\rightarrow_\forall$	$\neg B(x_3)$	(23)
(22), (23)	*CLASH*	(24)

$\mathcal{ALC}$ Tableaux - exercise

Solution

$$
\begin{array}{lll}
 & \exists R.\neg A \sqcap \forall R(\forall R.B \sqcup A) \sqcap & \\
 & (\exists R.\exists R.A \sqcap \forall R.\forall R.\neg B)(x_0) & (5) \\
(5) \rightarrow_{\sqcap} & \exists R.\neg A(x_0) & (6) \\
 & \forall R(\forall R.B \sqcup A)(x_0) & (7) \\
 & \exists R.\exists R.A(x_0) & (8) \\
 & \forall R.\forall R.\neg B(x_0) & (9) \\
(6) \rightarrow_{\exists} & R(x_0, x_1) & (10) \\
 & \neg A(x_1) & (11) \\
(10),(7) \rightarrow_{\forall} & (\forall R.B \sqcup A)(x_1) & (12) \\
(10),(9) \rightarrow_{\forall} & \forall R \neg B(x_1) & (13) \\
\hline
(12) \rightarrow_{\sqcup} & \forall R.B(x_1) & (14) \\
(8) \rightarrow_{\exists} & R(x_0, x_2) & (15) \\
 & \exists R.A(x_2) & (16)
\end{array}
$$

$$
\begin{array}{lll}
(15),(7) \rightarrow_{\forall} & (\forall R.B \sqcup A)(x_2) & (17) \\
\hline
(17) \rightarrow_{\sqcup} & A(x_2) & (18)
\end{array}
$$

$\mathcal{ALC}$ Tableaux - exercise

Solution

$$
\begin{array}{rll}
 & \exists R.\neg A \sqcap \forall R(\forall R.B \sqcup A) \sqcap & \\
 & (\exists R.\exists R.A \sqcap \forall R.\forall R.\neg B)(x_0) & (5) \\
(5) \rightarrow_{\sqcap} & \exists R.\neg A(x_0) & (6) \\
 & \forall R(\forall R.B \sqcup A)(x_0) & (7) \\
 & \exists R.\exists R.A(x_0) & (8) \\
 & \forall R.\forall R.\neg B(x_0) & (9) \\
(6) \rightarrow_{\exists} & R(x_0, x_1) & (10) \\
 & \neg A(x_1) & (11) \\
(10), (7) \rightarrow_{\forall} & (\forall R.B \sqcup A)(x_1) & (12) \\
(10), (9) \rightarrow_{\forall} & \forall R \neg B(x_1) & (13) \\
\hline
(12) \rightarrow_{\sqcup} & \forall R.B(x_1) & (14) \\
(8) \rightarrow_{\exists} & R(x_0, x_2) & (15) \\
 & \exists R.A(x_2) & (16)
\end{array}
$$

$$
\begin{array}{rll}
(15), (7) \rightarrow_{\forall} & (\forall R.B \sqcup A)(x_2) & (17) \\
\hline
(17) \rightarrow_{\sqcup} & A(x_2) & (18) \\
(15), (9) \rightarrow_{\forall} & \forall R \neg B(x_2) & (19) \\
(16) \rightarrow_{\exists} & R(x_2, x_3) & (20) \\
 & A(x_3) & (21) \\
(20), (18) \rightarrow_{\forall} & B(x_3) & (22) \\
(20), (19) \rightarrow_{\forall} & \neg B(x_3) & (23)
\end{array}
$$

Consistency of $\mathcal{ALC}$ A-boxes

Consistency of $\mathcal{ALC}$-ABoxe

Let $\mathcal{A}_0$ be an $\mathcal{ALC}$-ABox in NNF. To test $\mathcal{A}_0$ for consistency, we simply apply the rules given above to $\mathcal{A}_0$.

Theorem

Consistency of ALC ABoxes is PSPACE-COMPLETE.

Exercise

Which of the following statements are true? Explain your answer.

1. $\forall R.(A \sqcap B) \sqsubseteq \forall R.A \sqcap \forall R.B$
2. $\forall R.A \sqcap \forall R.B \sqsubseteq \forall R.(A \sqcap B)$
3. $\forall R.A \sqcup \forall R.B \sqsubseteq \forall R.(A \sqcup B)$
4. $\forall R.(A \sqcup B) \sqsubseteq \forall R.A \sqcup \forall R.B$

5. $\exists R.(A \sqcap B) \sqsubseteq \exists R.A \sqcap \exists R.B$
6. $\exists R.(A \sqcup B) \sqsubseteq \exists R.A \sqcup \exists R.B$
7. $\exists R.A \sqcup \exists R.B \sqsubseteq \exists R.(A \sqcup B)$
8. $\exists R.A \sqcap \exists R.B \sqsubseteq \exists R.(A \sqcap B)$

Exercise

Which of the following statements are true? Explain your answer.

1. $\forall R.(A \sqcap B) \sqsubseteq \forall R.A \sqcap \forall R.B$
2. $\forall R.A \sqcap \forall R.B \sqsubseteq \forall R.(A \sqcap B)$
3. $\forall R.A \sqcup \forall R.B \sqsubseteq \forall R.(A \sqcup B)$
4. $\forall R.(A \sqcup B) \sqsubseteq \forall R.A \sqcup \forall R.B$
   $R^{\mathcal{I}} = \{(x, y), (x, z)\},\ A^{I} = \{y\},\ B^{\mathcal{I}} = \{z\}$
5. $\exists R.(A \sqcap B) \sqsubseteq \exists R.A \sqcap \exists R.B$
6. $\exists R.(A \sqcup B) \sqsubseteq \exists R.A \sqcup \exists R.B$
7. $\exists R.A \sqcup \exists R.B \sqsubseteq \exists R.(A \sqcup B)$
8. $\exists R.A \sqcap \exists R.B \sqsubseteq \exists R.(A \sqcap B)$
   $R^{\mathcal{I}} = \{(x, y), (x, z)\},\ A^{I} = \{y\},\ B^{\mathcal{I}} = \{z\}$

Reasoning in $\mathcal{ALC}$ with T-box

Subsumption w.r.t. TBoxes A subsumption $C \sqsubseteq D$ follows from a TBox $\mathcal{T}$, in symbols $\mathcal{T} \models C \sqsubseteq D$, if for every interpretation $\mathcal{I}$, if $\mathcal{I} \models \mathcal{T}$ then $\mathcal{I} \models C \sqsubseteq D$

Concept satisfiability w.r.t. TBoxes A concept C is satisfiable w.r.t. a TBox $\mathcal{T}$ if there exists an interpretation $\mathcal{I} \models \mathcal{T}$ and such that $C^{\mathcal{I}} \neq \emptyset$.

TBox satisfiability A TBox $\mathcal{T}$ is satisfiable if, there is a model of $\mathcal{T}$.

We have the following reductions to concept satisfiability w.r.t. T-Boxes:

- $\mathcal{T} \models C \sqsubseteq D$ if and only if $C \sqcap \neg D$ is not consistent w.r.t. $\mathcal{T}$.
- $\mathcal{T}$ is satisfiable if $\top$ is consistent w.r.t. $\mathcal{T}$.

$\mathcal{ALC}$ concept satisfiability w.r.t. Acyclic T-box

Definition (Acyclic T-box)

A TBox is acyclic if it is a set of concept definitions that neither contains multiple definitions nor cyclic definitions.

Multiple definitions are of the form $A \doteq C$ and $A \doteq D$ for distinct concept descriptions C and D

cyclic definitions are of the form

$$A_1 \doteq C_1[A_2],\ A_2 \doteq C_2[A_3],\ \dots,\ A_n \doteq C_n[A_1]$$

where $C[A]$ means that the atomic concept A occurs in the complex concept description C.

Unfolding w.r.t. an acyclic T-Box

Naive reduction to $\mathcal{ALC}$ satisfiability

Satisfiability w.r.t. acyclic T-box can be reduced to $\mathcal{ALC}$ satisfiability without T-Boxes by unfolding the definitions

Unfolding: recursively replacing defined names by their defining concepts until no more defined names occur.

Definition (unfolding C w.r.t. $\mathcal{T}$)

If $\mathcal{T}$ is an acyclic T-box that does not contain multiple definitions, then the unfolding of C w.r.t. $\mathcal{T}$, is a concept denoted $unfold_{\mathcal{T}}(C)$ recursively defined as follows:

- $unfold_{\mathcal{T}}(A) = A$ if $\mathcal{T}$ does not contain any definition for A
- $unfold_{\mathcal{T}}(A) = unfold(C)$ if $\mathcal{T}$ contains the definition $A \equiv C$
- $unfold_{\mathcal{T}}(C \circ D) = unfold_{\mathcal{T}}(C) \circ unfold_{\mathcal{T}}(D)$ for $\circ = \sqcap, \sqcup$
- $unfold_{\mathcal{T}}(\circ C) = \circ unfold_{\mathcal{T}}(C)$ for $\circ = \neg, \exists R, \forall R$.

Theorem

C is satisfiable w.r.t. $\mathcal{T}$ iff $unfold_{\mathcal{T}}(C)$ is satisfiable.

$\mathcal{ALC}$ concept satisfiability w.r.t. Acyclic T-box

Exponential blow up

Unfolding may lead to an exponential blow-up,

$$\begin{array}{rcl} A_0 & \doteq & \forall R.A_1 \sqcap \forall S.A_1 \\ A_1 & \doteq & \forall R.A_2 \sqcap \forall S.A_2 \\ & \vdots & \\ A_{n-1} & \doteq & \forall R.A_n \sqcap \forall S.A_n \end{array}$$

One can easily check that the unfold of A_0 generats a concept of length 2^n, and therefore the unfolding of a concept can grow exponentially

$\mathcal{ALC}$ concept satisfiability w.r.t. Acyclic T-box

Smarter strategy - Unfolding on demand

We adopt a smarter strategy: unfold only when a concept effectively appear in the tree, and apply only one unfold step. Do not unfold completely.

Rule	Condition	$\longrightarrow$	Effect
$\rightarrow_\sqcap$	$C_1 \sqcap C_2(x) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{C_1(x), C_2(x)\}$
$\rightarrow_\sqcup$	$C_1 \sqcup C_2(x) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{C_1(x)\}$ or $\mathcal{A} \cup \{C_2(x)\}$
$\rightarrow_\exists$	$\exists R.C(x) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{R(x, y), C(y)\}$
$\rightarrow_\forall$	$\forall R.C(x), R(x, y) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{C(y)\}$
$\rightarrow_\mathcal{T}$	$A(x) \in \mathcal{A}$ and $A \doteq C \in \mathcal{T}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup NNF(C)(x)$

Theorem

Satisfiability w.r.t. acyclic terminologies is PSPACE-COMPLETE *in* $\mathcal{ALC}$.

$\mathcal{ALC}$ concept satisfiability w.r.t. generic T-box

Intuition

1. $C \sqsubseteq D$ is equivalent to $\top \sqsubseteq \neg C \sqcup D$
2. The set of axioms $\{\top \sqsubseteq \neg C_1 \sqcup D_1, \ldots, \top \sqsubseteq \neg C_n \sqcup D_n\}$ can be compressed in one single axiom $\top \sqsubseteq C_{\mathcal{T}}$, where

$$C_{\mathcal{T}} = (\neg C_1 \sqcup D_1) \sqcap \cdots \sqcap (\neg C_N \sqcup D_n)$$

3. For every individual x that is generated in the A-box $\mathcal{A}$, we have to add also the fact that it is of type $C_{\mathcal{T}}$.
4. We extend the set of rules as follows:

Rule	**Condition**	$\longrightarrow$	**Effect**
$\rightarrow_\sqcap$	$C_1 \sqcap C_2(x) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{C_1(x), C_2(x)\}$
$\rightarrow_\sqcup$	$C_1 \sqcup C_2(x) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{C_1(x)\}$ or $\mathcal{A} \cup \{C_2(x)\}$
$\rightarrow_\exists$	$\exists R.C(x) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{R(x,y), C(y)\}$
$\rightarrow_\forall$	$\forall R.C(x), R(x,y) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{C(y)\}$
$\rightarrow_{\mathcal{T}}$	x occurs in $\mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup NNF(C_{\mathcal{T}})(x)$

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$\{C(x_0)\}$$

termination is no longaer guaranteed

Due to the application of the $\rightarrow_{\mathcal{T}}$-rule, the nesting of the concepts does not decrease with each rule-application step.

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$\{C(x_0)\} \quad \rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\}$$

termination is no longaer guaranteed

Due to the application of the $\rightarrow_{\mathcal{T}}$-rule, the nesting of the concepts does not decrease with each rule-application step.

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$\begin{aligned} \{C(x_0)\} \quad & \rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\} \\ & \rightarrow_{\sqcup} \{C(x_0), \exists R.C(x_0)\} \end{aligned}$$

termination is no longaer guaranteed

Due to the application of the $\rightarrow_{\mathcal{T}}$-rule, the nesting of the concepts does not decrease with each rule-application step.

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$
\begin{aligned}
\{C(x_0)\} \quad & \rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\} \\
& \rightarrow_{\sqcup} \{C(x_0), \exists R.C(x_0)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1)\}
\end{aligned}
$$

termination is no longaer guaranteed

Due to the application of the $\rightarrow_{\mathcal{T}}$-rule, the nesting of the concepts does not decrease with each rule-application step.

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$
\begin{aligned}
\{C(x_0)\} \quad & \rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\} \\
& \rightarrow_{\sqcup} \{C(x_0), \exists R.C(x_0)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1)\} \\
& \rightarrow_{\mathcal{T}} \{C(x_0), R(x_0, x_1), C(x_1), \neg C \sqcup \exists R.C(x_1)\}
\end{aligned}
$$

termination is no longaer guaranteed

Due to the application of the $\rightarrow_{\mathcal{T}}$-rule, the nesting of the concepts does not decrease with each rule-application step.

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$
\begin{aligned}
\{C(x_0)\} \quad & \rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\} \\
& \rightarrow_{\sqcup} \{C(x_0), \exists R.C(x_0)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1)\} \\
& \rightarrow_{\mathcal{T}} \{C(x_0), R(x_0, x_1), C(x_1), \neg C \sqcup \exists R.C(x_1)\} \\
& \rightarrow_{\sqcup} \{C(x_0), R(x_0, x_1), C(x_1), \exists R.C(x_1)\}
\end{aligned}
$$

termination is no longaer guaranteed

Due to the application of the $\rightarrow_{\mathcal{T}}$-rule, the nesting of the concepts does not decrease with each rule-application step.

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$
\begin{aligned}
\{C(x_0)\} \quad & \rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\} \\
& \rightarrow_{\sqcup} \{C(x_0), \exists R.C(x_0)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1)\} \\
& \rightarrow_{\mathcal{T}} \{C(x_0), R(x_0, x_1), C(x_1), \neg C \sqcup \exists R.C(x_1)\} \\
& \rightarrow_{\sqcup} \{C(x_0), R(x_0, x_1), C(x_1), \exists R.C(x_1)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1), R(x_1, x_2), C(x_2)\}
\end{aligned}
$$

termination is no longaer guaranteed

Due to the application of the $\rightarrow_{\mathcal{T}}$-rule, the nesting of the concepts does not decrease with each rule-application step.

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$
\begin{array}{ll}
\{C(x_0)\} & \rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\} \\
& \rightarrow_{\sqcup} \{C(x_0), \exists R.C(x_0)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1)\} \\
& \rightarrow_{\mathcal{T}} \{C(x_0), R(x_0, x_1), C(x_1), \neg C \sqcup \exists R.C(x_1)\} \\
& \rightarrow_{\sqcup} \{C(x_0), R(x_0, x_1), C(x_1), \exists R.C(x_1)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1), R(x_1, x_2), C(x_2)\} \\
& \rightarrow_{\mathcal{T}} \dots
\end{array}
$$

termination is no longaer guaranteed

Due to the application of the $\rightarrow_{\mathcal{T}}$-rule, the nesting of the concepts does not decrease with each rule-application step.

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Blocking

- y is an ancestor of y in an A-box $\mathcal{A}$, if $\mathcal{A}$ contains

$$R_0(y, x_1), R_1(x_1, x_2), \ldots, R_n(x_n, x)$$

- $Ł(x) = \{C | C(x) \in \mathcal{A}\}$
- x is directly blocked in $\mathcal{A}$ if it has an ancestor y with $Ł(x) \subseteq Ł(y)$
- if y is the closest such node to x, we say that x is blocked by y
- A node is blocked if it is directly blocked or one of its ancestors is blocked

Restriction

Restrict the application of all rules to nodes which are not blocked

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$
\begin{aligned}
\{C(x_0)\} \quad & \rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\} \\
& \rightarrow_{\sqcup} \{C(x_0), \exists R.C(x_0)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1)\}
\end{aligned}
$$

Termination

With blocking strategy the algorithm is guaranteed to terminate

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$
\begin{aligned}
\{C(x_0)\} \quad & \rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\} \\
& \rightarrow_{\sqcup} \{C(x_0), \exists R.C(x_0)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1)\}
\end{aligned}
$$

x_1 *is blocked by* x_0 *since*

$$
Ł(x_1) = \{C\} \subseteq Ł(x_0) = \{C, \exists R.C\}
$$

Termination

With blocking strategy the algorithm is guaranteed to terminate

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Cyclic interpretations

The interpretation $\mathcal{I}_{\mathcal{A}}$ generated from an A-box $\mathcal{A}$ obtained by the tableaux algorithm with blocking strategy is defined as follows:

- $\Delta^{\mathcal{I}_{\mathcal{A}}} = \{x \mid C(x) \in \mathcal{A} \text{ and } x \text{ is not blocked}\}$
- $A^{\mathcal{I}_{\mathcal{A}}} = \{x \in \Delta^{\mathcal{I}_{\mathcal{A}}} \mid A(x) \in \mathcal{A}\}$
- $R^{\mathcal{I}_{\mathcal{A}}} = \{(x, y) \in \Delta^{\mathcal{I}_{\mathcal{A}}} \times \Delta^{\mathcal{I}_{\mathcal{A}}} \mid R(x, y) \in \mathcal{A}\} \cup$
  $\{(x', x) \mid x' \in \Delta^{\mathcal{I}_{\mathcal{A}}},\ R(x', x) \in \mathcal{A}, \text{and } x \text{ is blocked by } y\}$

Complexity

The algorithm is no longer in PSPACE since it may generate role paths of exponential length before blocking occurs. S

Theorem

Satisfiability of an $\mathcal{ALC}$ concept w.r.t. general T-box is EXPTIME-COMPLETE

Finite model property

Theorem

A consistent T-box in $\mathcal{ALC}$ has a finite model

proof

The model constructed via tableaux is finite. Completeness of the tableaux procedure implies that if a T-box is consistent, then the algorithm will find a model, which is indeed finite

Exercise

Transform $\neg(A \cup (\neg B \cap E) \cup (\exists R.(C \cup \forall P.(\neg D \cup (\exists P.\neg D)))))$ in negation normal form. Show that $\mathcal{K} \models A(a)$

Exercise

Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ with $\mathcal{T} = \{\top \sqsubseteq \forall R.C,\ C \sqcap A \sqsubseteq \bot$, and $\mathcal{A} = \{\exists R.A\}$