Description Logics

Franz Baader

Theoretical Computer Science

TU Dresden

Germany

1. Motivation and introduction to Description Logics

2. Tableau-based reasoning procedures

Dresden

Reasoning procedures

- The procedure should be a decision procedure for the problem:
  - soundness: positive answers are correct
  - completeness: negative answers are correct
  - termination: always gives an answer in finite time
- The procedure should be as efficient as possible:

  preferably optimal w.r.t. the (worst-case) complexity of the problem
- The procedure should be practical:

  easy to implement and optimize, and behave well in applications

Tableau algorithm

for $\mathcal{ALC}$

It is sufficient to design a decision procedure for consistency of an ABox without a TBox:

- TBoxes can be eliminated by expanding concept descriptions
- satisfiability, subsumption, and the instance problem can be reduced to consistency

The tableau-based consistency algorithm tries to generate a finite model for the input ABox $\mathcal{A}_0$:

- applies tableau rules to extend the ABox *one rule per constructor*
- checks for obvious contradictions
- an ABox that is complete (no rule applies) and open (no obvious contradictions) describes a model

Tableau algorithm

example

$\mathcal{T}$ $\quad$ GoodStudent $\equiv$ Smart $\sqcap$ Studious

Subsumption question:

$$\exists\text{attends.Smart} \sqcap \exists\text{attends.Studious} \sqsubseteq^?_{\mathcal{T}} \exists\text{attends.GoodStudent}$$

Reduction to satisfiability: is the following concept unsatisfiable w.r.t. $\mathcal{T}$?

$$\exists\text{attends.Smart} \sqcap \exists\text{attends.Studious} \sqcap \neg\exists\text{attends.GoodStudent}$$

Reduction to consistency: is the following ABox inconsistent w.r.t. $\mathcal{T}$?

$$\{\ (\exists\text{attends.Smart} \sqcap \exists\text{attends.Studious} \sqcap \neg\exists\text{attends.GoodStudent})(a)\ \}$$

Expansion: is the following ABox inconsistent?

$$\{\ (\exists\text{attends.Smart} \sqcap \exists\text{attends.Studious} \sqcap \neg\exists\text{attends.(Smart} \sqcap \text{Studious}))(a)\ \}$$

Negation normal form: is the following ABox inconsistent?

$$\{\ (\exists\text{attends.Smart} \sqcap \exists\text{attends.Studious} \sqcap \forall\text{attends.}(\neg\text{Smart} \sqcup \neg\text{Studious}))(a)\ \}$$

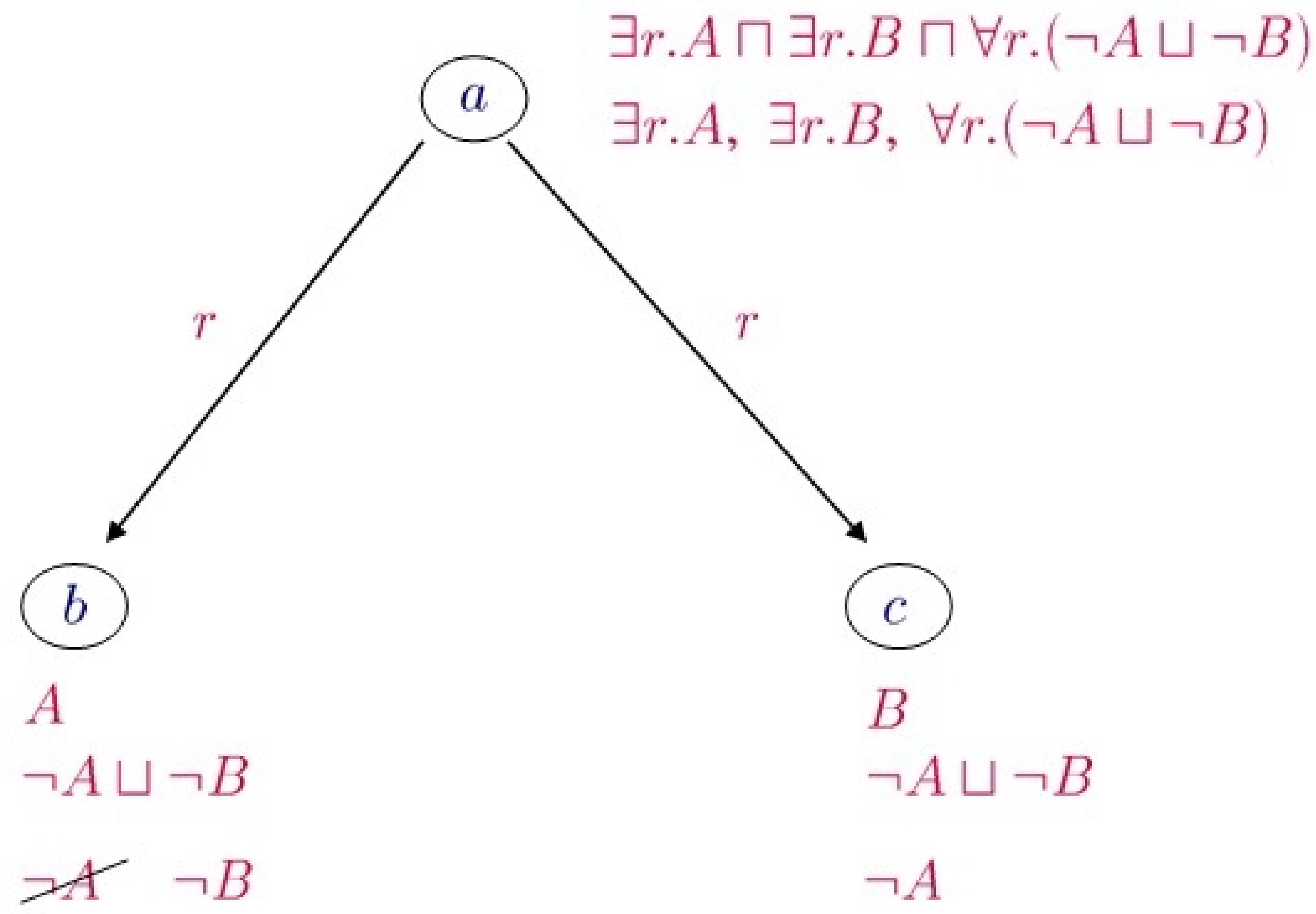
Dresden

Tableau algorithm

example continued

Is the following ABox inconsistent?

$\{ (\exists \text{attends.Smart} \sqcap \exists \text{attends.Studious} \sqcap \forall \text{attends.}(\neg\text{Smart} \sqcup \neg\text{Studious}))(a) \}$



complete and open ABox yields a model for the input ABox

and thus a counterexample to the subsumption relationship

Tableau algorithm

more formal description

Input: An $\mathcal{ALC}$-ABox $\mathcal{A}_0$

Output: "yes" if $\mathcal{A}_0$ is consistent

"no" otherwise

negation only in front of concept names

Preprocessing:

transform all concept descriptions in $\mathcal{A}_0$ into negation normal form (NNF) by applying the following rules:

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

The NNF can be computed in polynomial time, and it does not change the semantics of the concept.

Tableau algorithm

more formal description

Data structure:

finite set of ABoxes rather than a single ABox: start with $\{\mathcal{A}_0\}$ ← *in NNF*

Application of tableau rules:

the rules take one ABox from the set and replace it by finitely many new ABoxes

Termination:

if no more rules apply to any ABox in the set

complete ABox:
no rule applies to it

Answer:

"yes" if the set contains an open ABox, i.e., an ABox not containing an obvious contradiction of the form

$$A(a) \text{ and } \neg A(a) \quad \text{for some individual name } a$$

"no" if all ABoxes in the set are closed (i.e., not open)

Tableau rules

one for every constructor (except for negation)

The $\sqcap$-rule

Condition: $\mathcal{A}$ contains $(C \sqcap D)(a)$, but not both $C(a)$ and $D(a)$

Action: $\mathcal{A}' := \mathcal{A} \cup \{C(a), D(a)\}$

The $\sqcup$-rule

Condition: $\mathcal{A}$ contains $(C \sqcup D)(a)$, but neither $C(a)$ nor $D(a)$

Action: $\mathcal{A}' := \mathcal{A} \cup \{C(a)\}$ and $\mathcal{A}'' := \mathcal{A} \cup \{D(a)\}$

The $\exists$-rule

Condition: $\mathcal{A}$ contains $(\exists r.C)(a)$, but there is no c with $\{r(a,c), C(c)\} \subseteq \mathcal{A}$

Action: $\mathcal{A}' := \mathcal{A} \cup \{r(a,b), C(b)\}$ where b is a new individual name

The $\forall$-rule

Condition: $\mathcal{A}$ contains $(\forall r.C)(a)$ and $r(a,b)$, but not $C(b)$

Action: $\mathcal{A}' := \mathcal{A} \cup \{C(b)\}$

Tableau algorithm

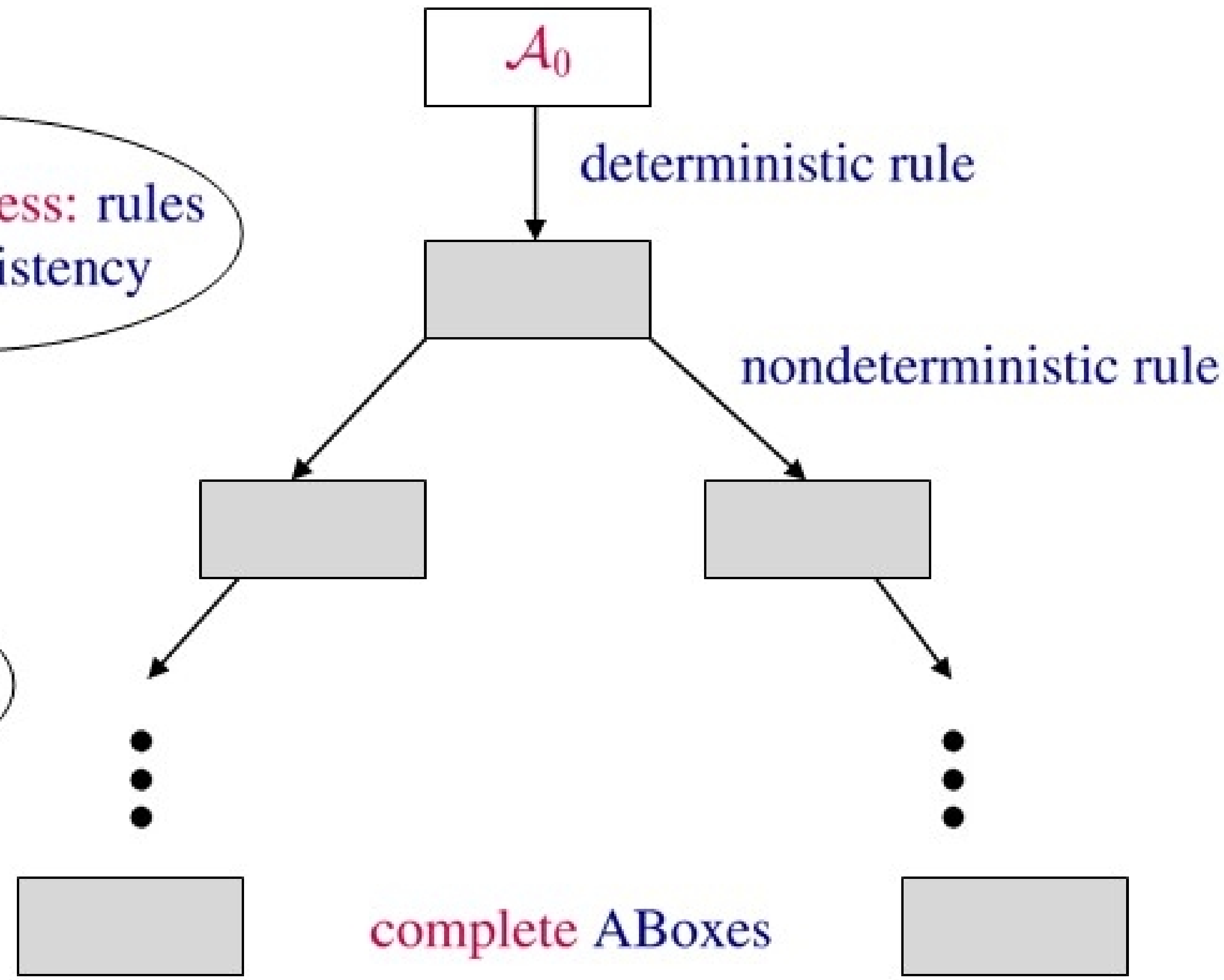
is a decision procedure for consistency

1.

local correctness: rules preserve consistency

2.

termination: no infinite paths



soundness: any complete and open ABox has a model

completeness: closed ABoxes do not have a model

3.

trivial

Dresden

Local correctness

rules preserve consisteny

The $\exists$-rule

Condition: $\mathcal{A}$ contains $(\exists r.C)(a)$, but there is no c with $\{r(a,c), C(c)\} \subseteq \mathcal{A}$

Action: $\mathcal{A}' := \mathcal{A} \cup \{r(a,b), C(b)\}$ where b is a new individual name

To show: $\mathcal{A}$ has a model iff $\mathcal{A}'$ has a model

$\Rightarrow$ Let $\mathcal{I}$ be a model of $\mathcal{A}$.

Since $(\exists r.C)(a) \in \mathcal{A}$, there is a $d \in \Delta^{\mathcal{I}}$ such that $(a^{\mathcal{I}}, d) \in r^{\mathcal{I}}$ and $d \in C^{\mathcal{I}}$.

Let $\mathcal{I}'$ be the interpretation that coinicides with $\mathcal{I}$,
with the exception that $b^{\mathcal{I}'} = d$.

Since b does not occur in $\mathcal{A}$, $\mathcal{I}'$ is a model of $\mathcal{A}$.

By definition of $b^{\mathcal{I}'}$, it is also a model of $\{r(a,b), C(b)\}$.

$\Leftarrow$ trivial since $\mathcal{A} \subseteq \mathcal{A}'$.

Dresden

Local correctness

rules preserve consisteny

The $\sqcup$-rule

Condition: $\mathcal{A}$ contains $(C \sqcup D)(a)$, but neither $C(a)$ nor $D(a)$

Action: $\mathcal{A}' := \mathcal{A} \cup \{C(a)\}$ and $\mathcal{A}'' := \mathcal{A} \cup \{D(a)\}$

To show: $\mathcal{A}$ has a model iff $\mathcal{A}'$ has a model or $\mathcal{A}''$ has a model

$\Rightarrow$ Let $\mathcal{I}$ be a model of $\mathcal{A}$.

Since $(C \sqcup D)(a) \in \mathcal{A}$, we have $a^{\mathcal{I}} \in (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$.

If $a^{\mathcal{I}} \in C^{\mathcal{I}}$, then $\mathcal{I}$ is a model of $\mathcal{A}'$.

If $a^{\mathcal{I}} \in D^{\mathcal{I}}$, then $\mathcal{I}$ is a model of $\mathcal{A}''$.

$\Leftarrow$ trivial since $\mathcal{A} \subseteq \mathcal{A}'$ and $\mathcal{A} \subseteq \mathcal{A}''$.

Termination

is an easy consequence of the following facts:

The label $\mathcal{L}(a)$ of an individual name consists of the concepts in concept assertions for a.

1. rule application is monotonic: every application of a rule extends the label of an individual, and does not remove anything;

2. concepts in labels are subdescriptions of concepts occurring in the input ABox $\mathcal{A}_0$;

$\Longrightarrow$ *finite number of rule applications per individual*

3. the number of new individuals that are r-successors of an individual is bounded by the number of existential restrictions in $\mathcal{A}_0$;

4. the length of successor chains of new individuals is bounded by the maximal size of the concepts in $\mathcal{A}_0$:
   - if x is a new individual, then it has a unique predecessor y
   - the maximal size of concepts in $\mathcal{L}(x)$ is strictly smaller than in $\mathcal{L}(y)$

$\Longrightarrow$ *finitely many new individuals*

Soundness

any complete and open ABox has a model

Let $\mathcal{A}$ be a complete and open ABox.

The canonical interpretation $\mathcal{I}_\mathcal{A}$ induced by $\mathcal{A}$ is defined as follows:

- $\Delta^{\mathcal{I}_\mathcal{A}} := \{x \mid x$ is an individual name occurring in $\mathcal{A}\}$
- $x^{\mathcal{I}_\mathcal{A}} := x$ for all indidual names occurring in $\mathcal{A}$
- $A^{\mathcal{I}_\mathcal{A}} := \{x \mid A(x) \in \mathcal{A}\}$ for all $A \in N_C$
- $r^{\mathcal{I}_\mathcal{A}} := \{(x, y) \mid r(x, y) \in \mathcal{A}\}$ for all $r \in N_R$

Claim $\mathcal{I}_\mathcal{A}$ is a model of $\mathcal{A}$.

Soundness

$\mathcal{I}_\mathcal{A}$ is a model of $\mathcal{A}$.

- if $r(x, y) \in \mathcal{A}$, then $(x^{\mathcal{I}_\mathcal{A}}, y^{\mathcal{I}_\mathcal{A}}) = (x, y) \in r^{\mathcal{I}_\mathcal{A}}$ by definition of $r^{\mathcal{I}_\mathcal{A}}$
- for $C(x) \in \mathcal{A}$, we show $x^{\mathcal{I}_\mathcal{A}} = x \in C^{\mathcal{I}_\mathcal{A}}$ by induction on the size of C:
  - $C = A$ for $A \in N_C$: trivial by definition of $A^{\mathcal{I}_\mathcal{A}}$
  - $C = \neg A$ for $A \in N_C$:
    since $\mathcal{A}$ is open, $A(x) \notin \mathcal{A}$, and thus $x \notin A^{\mathcal{I}_\mathcal{A}}$ by definition of $A^{\mathcal{I}_\mathcal{A}}$
  - $C = C_1 \sqcap C_2$:
    since $\mathcal{A}$ is complete, $(C_1 \sqcap C_2)(x) \in \mathcal{A}$ implies that
    $C_1(x) \in \mathcal{A}$ and $C_2(x) \in \mathcal{A}$;
    by induction, this yields $x \in C_1^{\mathcal{I}_\mathcal{A}}$ and $x \in C_2^{\mathcal{I}_\mathcal{A}}$,
    and thus $x \in (C_1 \sqcap C_2)^{\mathcal{I}_\mathcal{A}}$
  - the other constructors can be treated similarly

Tableau algorithm

is a decision procedure for consistency

1. Started with a finite ABox $\mathcal{A}_0$ in NNF the algorithm always terminates with a finite set of complete ABoxes $\mathcal{A}_1, \ldots \mathcal{A}_n$

2. Local correctness: $\mathcal{A}_0$ consistent iff one of $\mathcal{A}_1, \ldots, \mathcal{A}_n$ consistent

3. Answer "no":

   none of $\mathcal{A}_1, \ldots, \mathcal{A}_n$ open

   $\mathcal{A}_1, \ldots, \mathcal{A}_n$ inconsistent

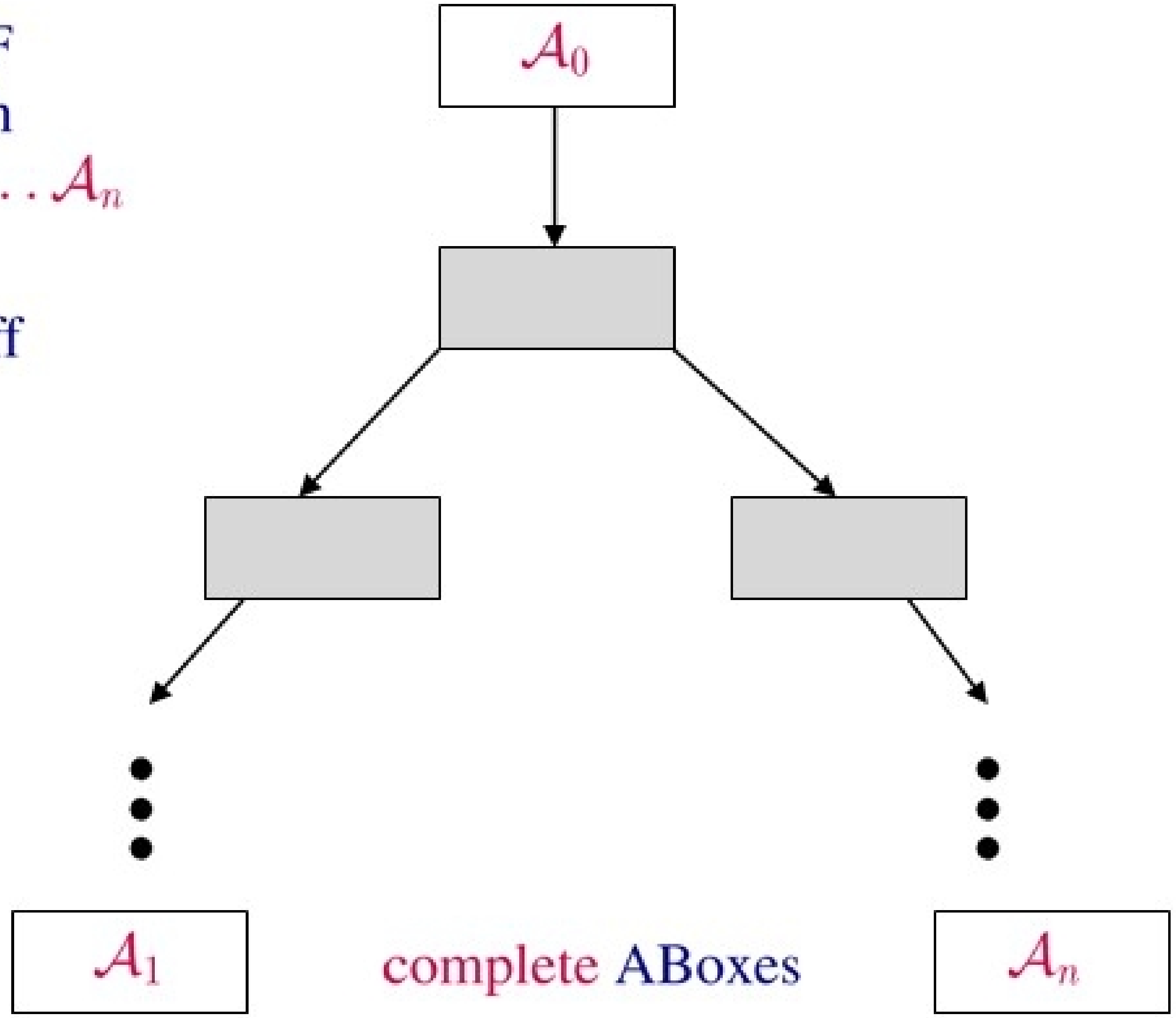
   $\mathcal{A}_0$ inconsistent

4. Answer "yes":

   one of $\mathcal{A}_1, \ldots, \mathcal{A}_n$ open

   one of $\mathcal{A}_1, \ldots, \mathcal{A}_n$ consistent

   $\mathcal{A}_0$ consistent



Dresden

Adding number restrictions

Number restrictions: $(\geq n\, r.C)$, $(\leq n\, r.C)$ with semantics

$$(\geq n\, r.C)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid \mathrm{card}(\{e \mid (d,e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}) \geq n\}$$

$$(\leq n\, r.C)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid \mathrm{card}(\{e \mid (d,e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}) \leq n\}$$

Negation normal form:

$$\neg(\geq n+1\, r.C) \rightsquigarrow (\leq n\, r.C)$$

$$\neg(\geq 0\, r.C) \rightsquigarrow \bot$$

$$\neg(\leq n\, r.C) \rightsquigarrow (\geq n+1\, r.C)$$

Extension of algorithm:

- new rules: $\geq$-rule and $\leq$-rule
- new assertions: inequality assertions of the form $x \neq y$ with obvious semantics $x^{\mathcal{I}} \neq y^{\mathcal{I}}$
- new obvious contradictions

inequality assertions viewed as symmetric

Dresden

Adding number restrictions

the tableau rules

The $\geq$-rule

Condition: $\mathcal{A}$ contains $(\geq n\, r.C)(a)$, but there are no $c_1, \ldots, c_n$ with
$\{r(a, c_1), C(c_1), \ldots, r(a, c_n), C(c_n)\} \cup \{c_i \not\doteq c_j \mid 1 \leq i, j \leq n\} \subseteq \mathcal{A}$

Action: $\mathcal{A}' := \mathcal{A} \cup \{r(a, b_1), C(c_1), \ldots, r(a, b_n), C(b_n)\} \cup \{b_i \not\doteq b_j \mid 1 \leq i, j \leq n\}$
where $b_1, \ldots, b_n$ are new individual names

The $\leq$-rule

Condition: $\mathcal{A}$ contains $(\leq n\, r.C)(a)$, and there are $b_1, \ldots, b_{n+1}$ with
$\{r(a, b_1), C(b_1), \ldots, r(a, b_{n+1}), C(b_{n+1})\} \subseteq \mathcal{A}$,
but $\{b_i \not\doteq b_j \mid 1 \leq i, j \leq n+1\} \not\subseteq \mathcal{A}$

Action: for all $i < j$ with $b_i \not\doteq b_j \notin \mathcal{A}$
$\mathcal{A}_{i,j} := \mathcal{A}[b_i \leftarrow b_j]$ *b_i replaced by b_j*

Dresden

Adding number restrictions

the new obvious contradictions

- $\mathcal{A}$ contains $(\leq n\, r.C)(a)$, and there are $b_1, \ldots, b_{n+1}$ with $\{r(a, b_1), C(b_1), \ldots, r(a, b_{n+1}), C(b_{n+1})\} \subseteq \mathcal{A}$ and $\{b_i \dot{\neq} b_j \mid 1 \leq i, j \leq n+1\} \subseteq \mathcal{A}$
- $\mathcal{A}$ contains $a \dot{\neq} a$ for some individual name a

Adding number restrictions

does this yield a decision procedure?

To show that the algorithm obtained this way is a decision procedure for ABox consistency, we must show

1. local correctness: rules preserve consistency

   easy to show

2. completeness: a closed ABox does not have a model

   trivial

3. soundness: a complete and open ABox has a model

   wrong!

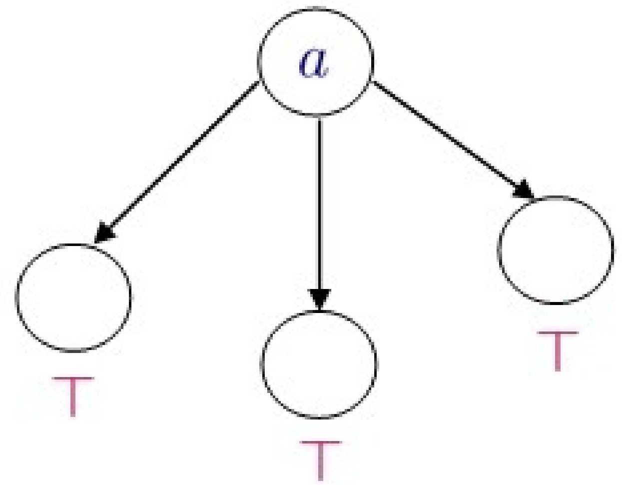
4. termination: there is no infinite chain of rule applications

   wrong!

Adding number restrictions

the problem with soundness

$\{\ (\geq 3\,\text{child}.\top)(a),\ \ (\leq 1\,\text{child}.\text{Female})(a),\ \ (\leq 1\,\text{child}.\neg\text{Female})(a)\ \}$



$\leq$-rule not applicable

no obvious contradiction

open, complete, but inconsistent

The choose-rule

$\sim C$: NNF of $\neg C$

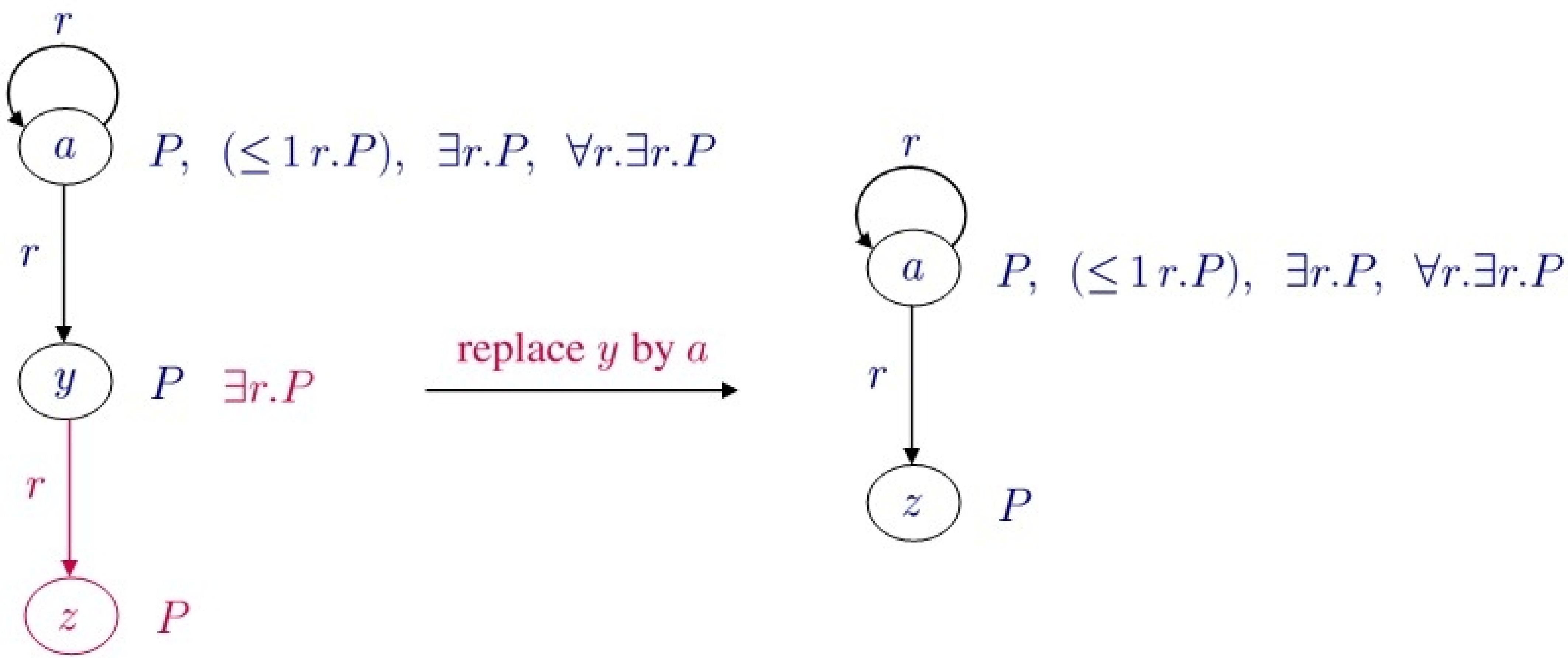
Condition:	$\mathcal{A}$ contains $(\leq n\,r.C)(a)$ and $r(a,b)$, but neither $C(b)$ nor $\sim C(b)$
Action:	$\mathcal{A}' := \mathcal{A} \cup \{C(b)\}$ and $\mathcal{A}'' := \mathcal{A} \cup \{\sim C(b)\}$

In the presence of the choose-rule, soundness can easily be shown.

Dresden

Adding number restrictions

the problem with termination



Solution:

use a strategy that applies generating rules ($\geq$-rule, $\exists$-rule) with lower priority.

Dresden

Adding GCIs

$C \sqsubseteq D$ with semantics $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$

A finite set of GCIs can be encoded into one GCI of the form $\top \sqsubseteq C$:

$$\{C_1 \sqsubseteq D_1, \ldots, C_n \sqsubseteq D_n\} \longrightarrow \{\top \sqsubseteq (\neg C_1 \sqcup D_1) \sqcap \ldots \sqcap (\neg C_n \sqcup D_n)\}$$

Consider a GCI $\top \sqsubseteq C$ where C is in NNF.

The GCI-rule for $\top \sqsubseteq C$

Condition: $\mathcal{A}$ contains the individual name a, but not $C(a)$

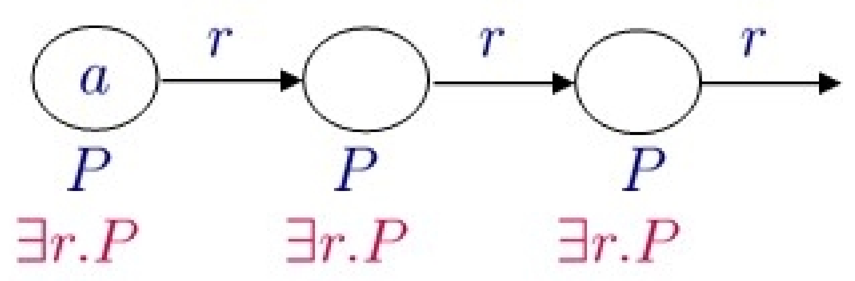
Action: $\mathcal{A}' := \mathcal{A} \cup \{C(a)\}$

Adding GCIs

does this yield a decision procedure?

- local correctness, completeness, and soundness are easy to show
- termination does not hold:

  Test consistency of $\{P(a)\}$ w.r.t. the GCI $\top \sqsubseteq \exists r.P$



Solution: blocking

- y is blocked by x iff $\mathcal{L}(y) \subseteq \mathcal{L}(x)$
- to avoid cyclic blocking we fix an enumeration of the individual names, and add to the blocking condition that y comes after x in the enumeration
- generating rules are not applied to blocked individuals

Dresden

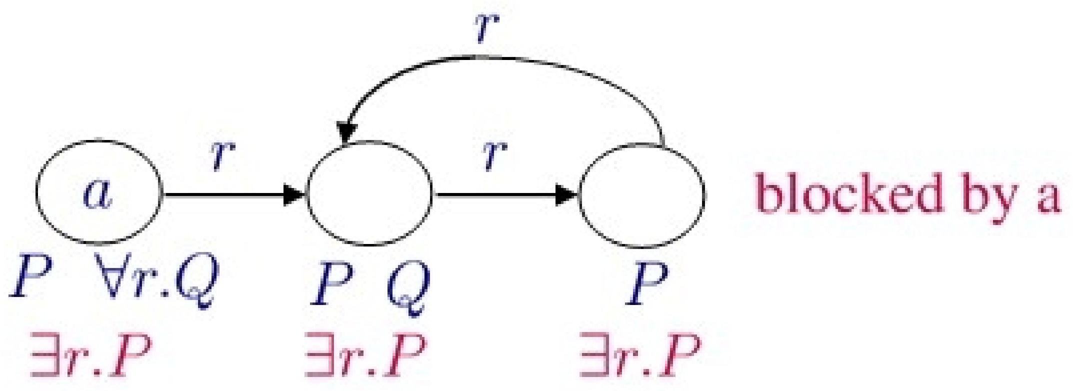
Adding GCIs

does this yield a decision procedure?

- local correctness, completeness, and termination are now easy to show
- soundness must be reconsidered:
  - because of blocking, an ABox can be complete although a generating rule applies
  - requires modification in the definition of the canonical interpretation:

    the r-successors of a blocked individual are the r-successors of the least individual (in the enumeration) blocking it

consistency of $\{(\forall r.Q)(a),\ P(a)\}$
w.r.t. the GCI $\top \sqsubseteq \exists r.P$



Assumptions (1)

- Open World Assumption
  - Given Abox A = { R(i , j), C(j) }
  - Is i an instance of $\forall$R.C
- No, cannot be proven for A:
  - UNSAT(A $\cup$ { ($\exists$R. $\neg$C)(i) }) does not hold
  - Applying the tableau rules yields an open Abox
- Could be proved if we added ($\leq$ 1 r. T)(i) to A

Assumptions (2)

- Unique name assumption
  - Different individual names denote different domain objects
  - Usually NOT adopted in DL and first-order settings in general
- Domain closure assumption
  - The set of individuals is finite
  - NOT adopted in general
    - Reduces first-order to propositional case