Chapter 4

Reasoning with tableaux algorithms

We start with an algoritm for deciding consistency of an ABox without a TBox since this covers most of the inference problems introduced in Chapter 2:

- acyclic TBoxes can be eliminated by expansion
- satisfiability, subsumption, and the instance problem can be reduced to ABox 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

Dresden

Tableau algorithm

example

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

Subsumption question:

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

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

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

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

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

Expansion: is the following ABox inconsistent?

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

Negation normal form: is the following ABox inconsistent?

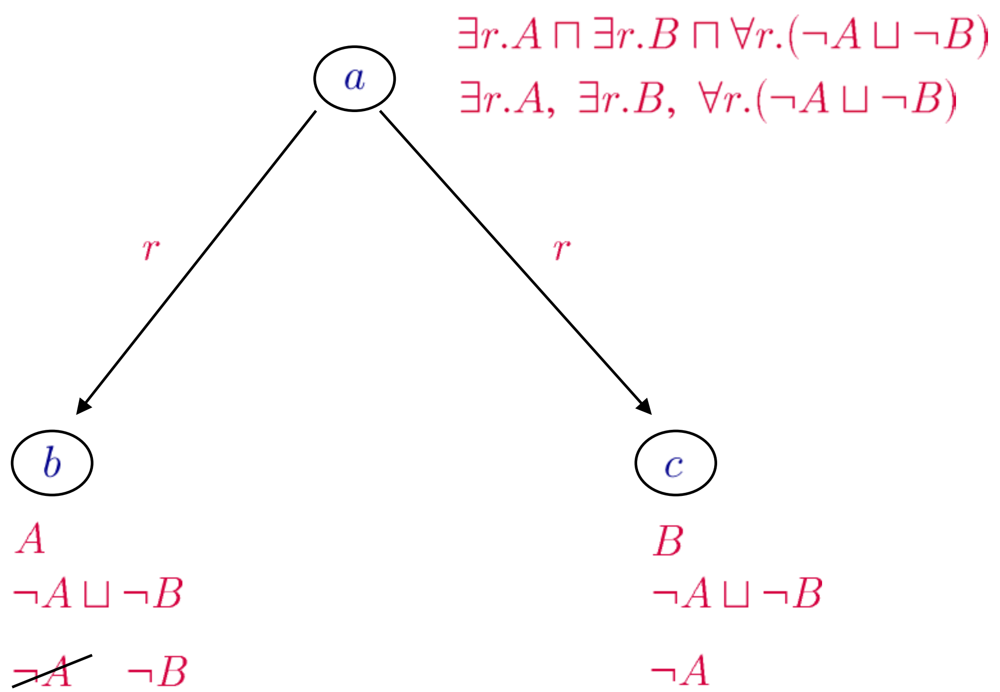
$$\{\ (\exists \text{attended.Smart} \sqcap \exists \text{attended.Studious} \sqcap \forall \text{attended.}(\neg\text{Smart} \sqcup \neg\text{Studious}))(a)\ \}$$

Tableau algorithm

example continued

Is the following ABox inconsistent?

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



complete and open ABox

yields a model for the input ABox

and thus a counterexample

to the subsumption relationship

Dresden

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:

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

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

Tableau algorithm

more formal description

in NNF

Data structure:

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

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:

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

$A(a)$ and $\neg A(a)$ for some individual name a

"inconsistent" 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 $C(b)$

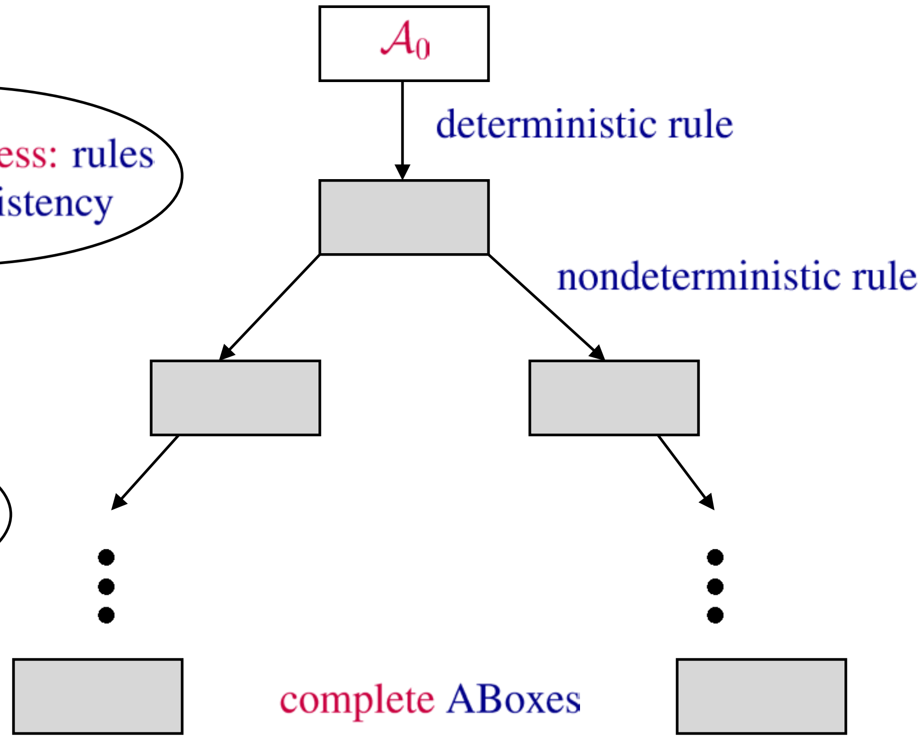
Action: $\mathcal{A}' := \mathcal{A} \cup \{C(b)\}$

Tableau algorithm

is a decision procedure for consistency

Lemma 4.1

local correctness: rules preserve consistency



Lemma 4.8

termination: no infinite paths

soundness: any complete and open ABox has a model

completeness: closed ABoxes do not have a model

Lemma 4.2

Dresden