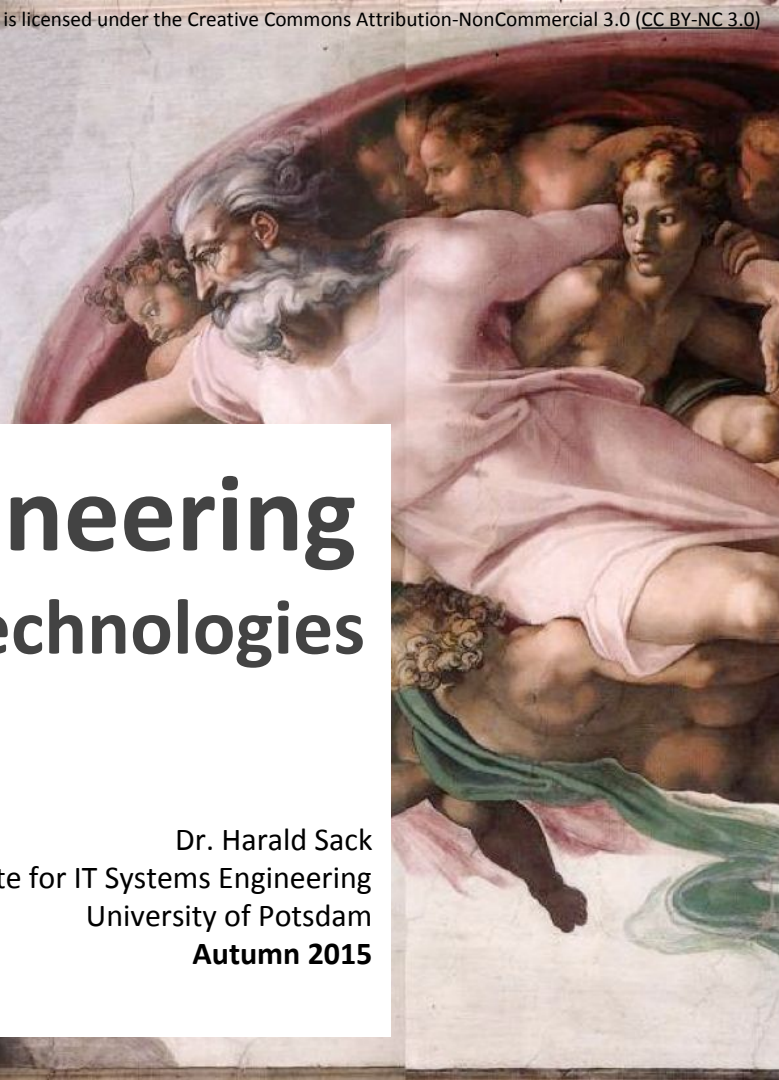
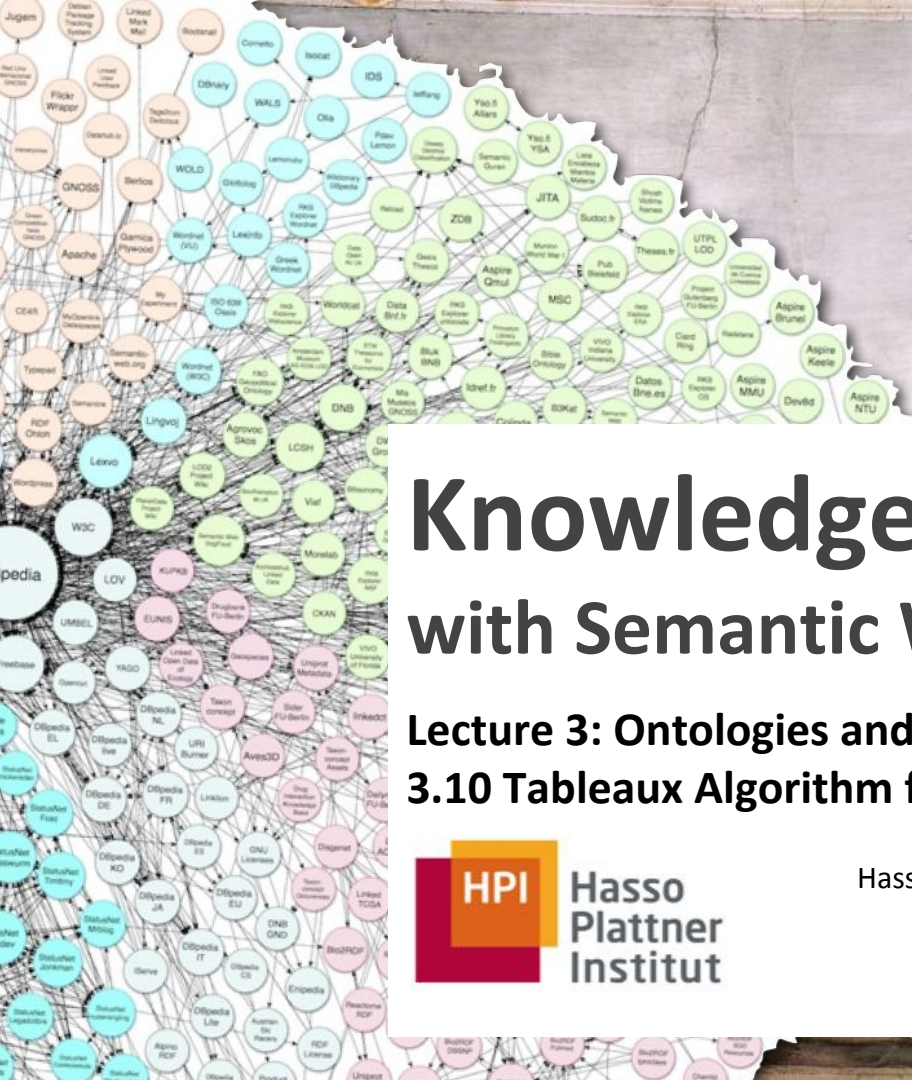




Knowledge Engineering with Semantic Web Technologies

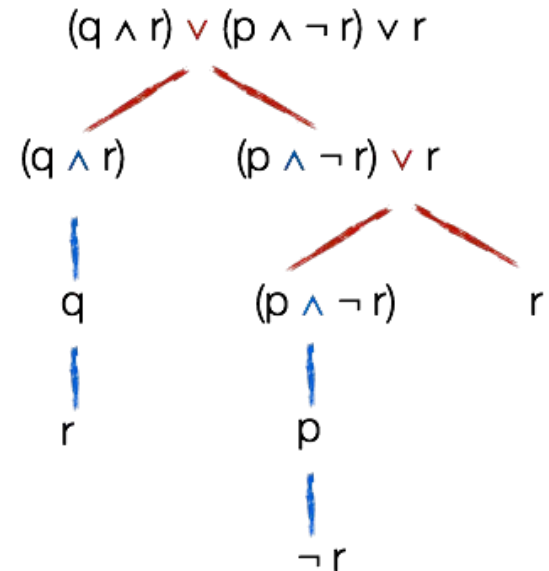
Lecture 3: Ontologies and Logic 3.10 Tableaux Algorithm for ALC



Dr. Harald Sack
Hasso Plattner Institute for IT Systems Engineering
University of Potsdam
Autumn 2015

Tableaux - A Short Recapitulation

- The tableaux algorithm is a proof algorithm to check the consistency of a logical formula by inferring that its negation is a contradiction (***proof by refutation***).
- Construct a **Tree**, where each **node** is marked with a logical formula. A **path** from the root to a leaf is the **conjunction** of all formulas represented within the nodes of the path; a **branch** of the path represents a **disjunction**.
- The tree is created by successive application of the **Tableaux Extension Rules**.



Tableaux Algorithm for $\mathcal{ALC}$

- Transformation of the DL Knowledge Base to **Negation normalform NNF** necessary
- Let W be a knowledge base,
 - Substitute $C \equiv D$ by $C \sqsubseteq D$ and $D \sqsubseteq C$
 - Substitute $C \sqsubseteq D$ by $\neg C \sqcup D$.
 - Apply the **NNF Transformations**
- Resulting knowledge base **NNF(W)**
 - Negation Normalform of W .
 - W and **NNF(W)** are logically equivalent
 - Negation is placed directly in front of atomic classes

NNF Transformations:

$\text{NNF}(C) = C$, if C is atomic
 $\text{NNF}(\neg C) = \neg C$, if C is atomic
 $\text{NNF}(\neg \neg C) = \text{NNF}(C)$
 $\text{NNF}(C \sqcup D) = \text{NNF}(C) \sqcup \text{NNF}(D)$
 $\text{NNF}(C \sqcap D) = \text{NNF}(C) \sqcap \text{NNF}(D)$
 $\text{NNF}(\neg(C \sqcup D)) = \text{NNF}(\neg C) \sqcap \text{NNF}(\neg D)$
 $\text{NNF}(\neg(C \sqcap D)) = \text{NNF}(\neg C) \sqcup \text{NNF}(\neg D)$
 $\text{NNF}(\forall R.C) = \forall R.\text{NNF}(C)$
 $\text{NNF}(\exists R.C) = \exists R.\text{NNF}(C)$
 $\text{NNF}(\neg \forall R.C) = \exists R.\text{NNF}(\neg C)$
 $\text{NNF}(\neg \exists R.C) = \forall R.\text{NNF}(\neg C)$

Transformation into NNF

Example:

$$\begin{aligned}
 & \text{NNF}(P \sqsubseteq (E \sqcap U) \sqcup \neg(\neg E \sqcup D)) \\
 = & \text{NNF}(\neg P \sqcup ((E \sqcap U) \sqcup \neg(\neg E \sqcup D))) \\
 = & \text{NNF}(\neg P) \sqcup \text{NNF}((E \sqcap U) \sqcup \neg(\neg E \sqcup D)) \\
 = & \neg P \sqcup (\text{NNF}(E \sqcap U)) \sqcup \text{NNF}(\neg(\neg E \sqcup D)) \\
 = & \neg P \sqcup (\text{NNF}(E) \sqcap \text{NNF}(U)) \sqcup (\text{NNF}(\neg \neg E) \sqcap \text{NNF}(\neg D)) \\
 = & \neg P \sqcup (E \sqcap U) \sqcup (\text{NNF}(E) \sqcap \neg D) \\
 = & \neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D)
 \end{aligned}$$

NNF Transformations:

$$\begin{aligned}
 \text{NNF}(C) &= C, \text{ if } C \text{ is atomic} \\
 \text{NNF}(\neg C) &= \neg C, \text{ if } C \text{ is atomic} \\
 \text{NNF}(\neg \neg C) &= \text{NNF}(C) \\
 \text{NNF}(C \sqcup D) &= \text{NNF}(C) \sqcup \text{NNF}(D) \\
 \text{NNF}(C \sqcap D) &= \text{NNF}(C) \sqcap \text{NNF}(D) \\
 \text{NNF}(\neg(C \sqcup D)) &= \text{NNF}(\neg C) \sqcap \text{NNF}(\neg D) \\
 \text{NNF}(\neg(C \sqcap D)) &= \text{NNF}(\neg C) \sqcup \text{NNF}(\neg D) \\
 \text{NNF}(\forall R.C) &= \forall R. \text{NNF}(C) \\
 \text{NNF}(\exists R.C) &= \exists R. \text{NNF}(C) \\
 \text{NNF}(\neg \forall R.C) &= \exists R. \text{NNF}(\neg C) \\
 \text{NNF}(\neg \exists R.C) &= \forall R. \text{NNF}(\neg C)
 \end{aligned}$$

Tableaux Expansion Rules for $\mathcal{ALC}$

- tableaux expansion rules are used to derive new ABox facts
- Let $\mathcal{KB}=(\mathcal{T}, \mathcal{A})$, then
 - ABox facts $C(a) \in \mathcal{A}$ or $R(a, b) \in \mathcal{A}$ can be added to the tableaux
 - TBox Axioms $C \in \mathcal{T}$ can be instantiated with known individuals $a \in \mathcal{A}$ and added to the tableaux: $C(a) \in \mathcal{A}$

$\sqcap$ -Rule: if $(C \sqcap D)(a) \in \mathcal{A}$ and $\{C(a), D(a)\} \not\subseteq \mathcal{A}$
 then $\mathcal{A}' = \mathcal{A} \cup \{C(a), D(a)\}$

$\sqcup$ -Rule: if $(C \sqcup D)(a) \in \mathcal{A}$ and $\{C(a), D(a)\} \cap \mathcal{A} = \emptyset$
 then $\mathcal{A}' = \mathcal{A} \cup \{C(a)\}$ and $\mathcal{A}'' = \mathcal{A} \cup \{D(a)\}$

$\exists$ -Rule: if $\exists R.C(a) \in \mathcal{A}$ and there is no b such that $C(b) \in \mathcal{A}$ and $R(a, b) \in \mathcal{A}$
 then $\mathcal{A}' = \mathcal{A} \cup \{C(z), R(a, z)\}$ for a new individual $z \notin \mathcal{A}$

$\forall$ -Rule: if $\forall R.C(a) \in \mathcal{A}$ and $R(a, b) \in \mathcal{A}$, but $C(b) \notin \mathcal{A}$
 then $\mathcal{A}' = \mathcal{A} \cup \{C(b)\}$

Tableaux Algorithm for $\mathcal{ALC}$

- Tableaux expansion rules are applicable in any order
- The tableaux algorithm is non-deterministic (due to disjunction)
- If the resulting tableaux is closed, the original knowledge base is **unsatisfiable**.
- Only select elements that lead to new elements within the tableaux.
If this is not possible, then the algorithm terminates and the original knowledge base is **satisfiable**.

Tableaux algorithm for checking satisfiability of $\mathcal{ALC}$ concepts:

Input: an $\mathcal{ALC}$ concept in NNF

Output: **true** if there is a clash-free tableaux where no more rules can be applied
false otherwise (tableaux closed)

Tableaux Algorithm ($\mathcal{ALC}$) - Example

$$\mathcal{T} = \{ \text{Professor} \sqsubseteq (\text{Person} \sqcap \text{UniversityEmployee}) \sqcup (\text{Person} \sqcap \neg \text{Student}) \}$$

$$\models \text{Professor} \sqsubseteq \text{Person}$$

?

- proof that $\text{Professor} \sqsubseteq \text{Person}$ is a logical consequence of $\mathcal{KB}$
- proof that $\mathcal{KB} \cup \neg(\text{Professor} \sqsubseteq \text{Person})$ is unsatisfiable

$$\begin{aligned} \text{NNF}(\neg(\text{Professor} \sqsubseteq \text{Person})) &= \text{NNF}(\neg(\neg \text{Professor} \sqcup \text{Person})) \\ &= \text{NNF}(\neg \neg \text{Professor}) \sqcap \text{NNF}(\neg \text{Person}) \\ &= \text{Professor} \sqcap \neg \text{Person} \end{aligned}$$

$$\mathcal{T}' = \{ \neg \text{Professor} \sqcup (\text{Person} \sqcap \text{UniversityEmployee}) \sqcup (\text{Person} \sqcap \neg \text{Student}), \\ \text{Professor} \sqcap \neg \text{Person} \}$$

Tableaux Algorithm ($\mathcal{ALC}$) - Example

$\mathcal{T} = \{ \neg \text{Professor} \sqcup (\text{Person} \sqcap \text{UniversityEmployee}) \sqcup (\text{Person} \sqcap \neg \text{Student}), \\ \text{Professor} \sqcap \neg \text{Person} \}$

1 | from KB $\text{Professor} \sqcap \neg \text{Person}(a)$ (*instantiate Class with new individual a*)

2 | $\sqcap$ from 1 $\text{Professor}(a)$

3 | $\sqcap$ from 1 $\neg \text{Person}(a)$

4 | from KB $(\neg \text{Professor} \sqcup (\text{Person} \sqcap \text{UniversityEmployee}) \sqcup (\text{Person} \sqcap \neg \text{Student}))(a)$

5 | $\sqcup$ from 4 $\neg \text{Professor}(a)$ | 6 | $\sqcup$ from 4 $(\text{Person} \sqcap \text{UniversityEmployee}) \sqcup (\text{Person} \sqcap \neg \text{Student}))(a)$

7 | $\sqcup$ from 6 $(\text{Person} \sqcap \text{UniversityEmployee})(a)$ | 8 | $\sqcup$ from 6 $(\text{Person} \sqcap \neg \text{Student}))(a)$

9 | $\sqcap$ from 7 $\text{Person}(a)$

11 | $\sqcap$ from 8 $\text{Person}(a)$

10 | $\sqcap$ from 7 $\text{UniversityEmployee}(a)$

12 | $\sqcap$ from 8 $\neg \text{Student}(a)$

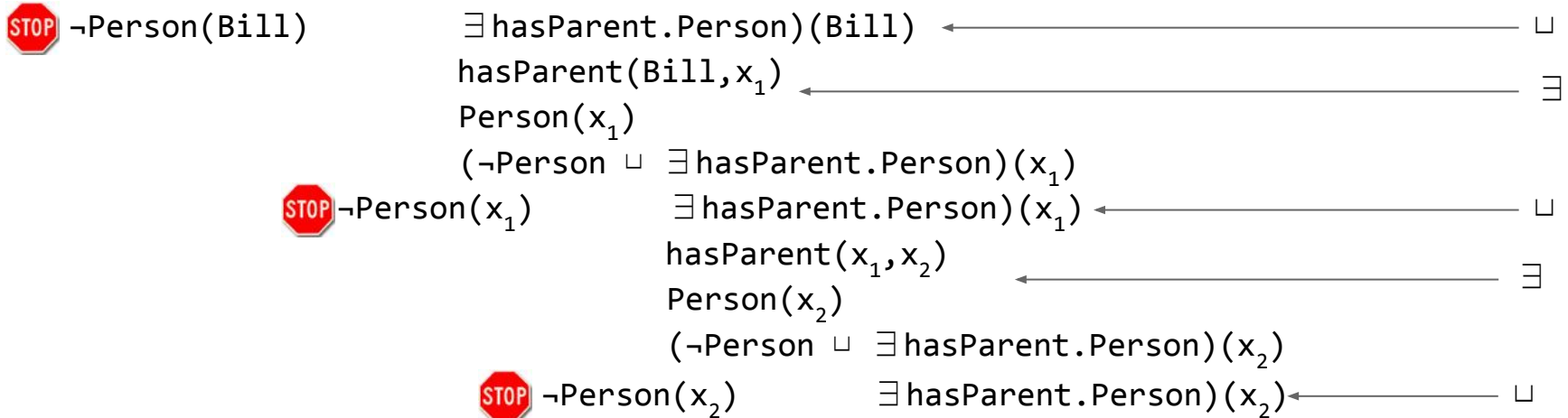
Tableaux Algorithm ($\mathcal{ALC}$) - Blocking

$\mathcal{T} = \{\neg \text{Person} \sqcup \exists \text{hasParent. Person}\}$

$\models \neg \text{Person}(\text{Bill})$? $\mathcal{T} = \{\neg \text{Person} \sqcup \exists \text{hasParent. Person}\} \mathcal{A} = \{\text{Person}(\text{Bill})\}$

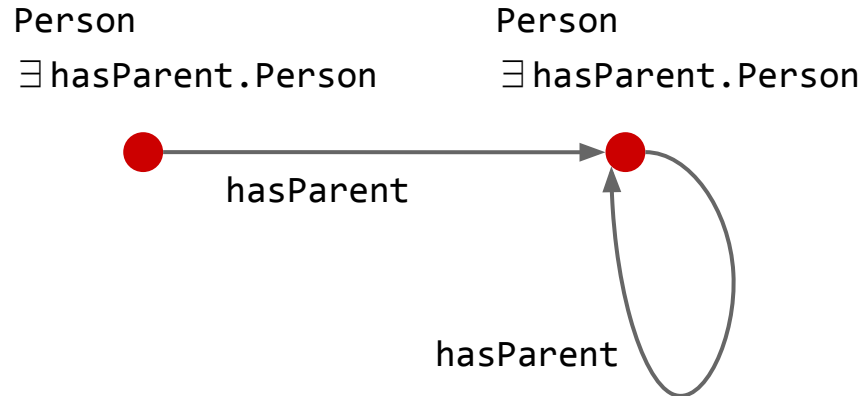
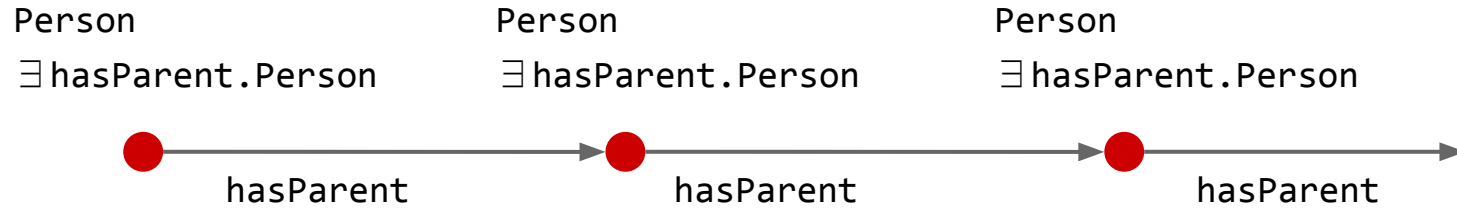
Person(Bill)

$(\neg \text{Person} \sqcup \exists \text{hasParent. Person})(\text{Bill})$



no termination possible

Tableaux Algorithm ($\mathcal{ALC}$) - Blocking



Blocking

Tableaux Algorithm ($\mathcal{ALC}$) - Blocking

$\mathcal{T} = \{\neg \text{Person} \sqcup \exists \text{hasParent}.\text{Person}\}$

$\models \neg \text{Person}(\text{Bill})$? $\mathcal{T} = \{\neg \text{Person} \sqcup \exists \text{hasParent}.\text{Person}\}$ $\mathcal{A} = \{\text{Person}(\text{Bill})\}$

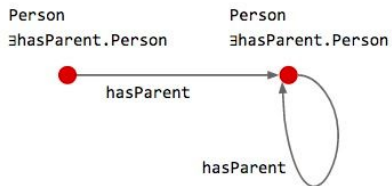
Person(Bill)

$(\neg \text{Person} \sqcup \exists \text{hasParent}.\text{Person})(\text{Bill})$

	$\neg \text{Person}(\text{Bill})$	$\exists \text{hasParent}.\text{Person})(\text{Bill})$	$\leftarrow$	$\sqcup$
		hasParent(Bill, x_1)	$\leftarrow$	$\exists$
		Person(x_1)		
		$(\neg \text{Person} \sqcup \exists \text{hasParent}.\text{Person})(x_1)$		
	$\neg \text{Person}(x_1)$	$\exists \text{hasParent}.\text{Person})(x_1)$	$\leftarrow$	$\sqcup$



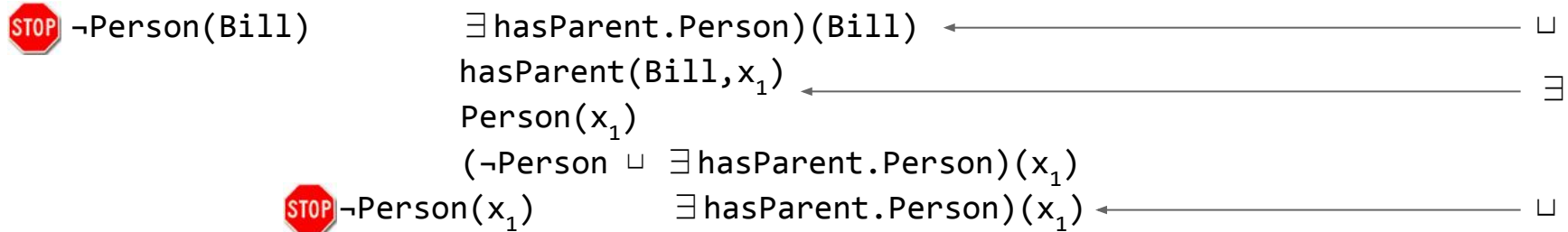
termination



Tableaux Algorithm ($\mathcal{ALC}$) - Blocking

Person(Bill)

$(\neg \text{Person} \sqcup \exists \text{hasParent}.\text{Person})(\text{Bill})$



$\sigma(\text{Individual}) = \{\text{Classes}\}$

σ is **Class Extension** of an Individual

$\sigma(\text{Bill}) = \{\text{Person},$
 $\neg \text{Person} \sqcup \exists \text{hasParent}.\text{Person},$
 $\exists \text{hasParent}.\text{Person}\}$

$\sigma(x_1) = \{\text{Person},$
 $\neg \text{Person} \sqcup \exists \text{hasParent}.\text{Person},$
 $\exists \text{hasParent}.\text{Person}\}$

$\sigma(x_1) \subseteq \sigma(\text{Bill})$, therefore Bill blocks x_1



termination

Tableaux Algorithm ($\mathcal{ALC}$) - Blocking

- The Selection of $(\exists R.C)(a)$ in the tableaux path A is **blocked**, if there is already an individual b with $\{C \mid C(a) \in A\} \subseteq \{C \mid C(b) \in A\}$.
- Two possibilities of termination:
 1. Closing the tableaux.
Knowledge Base is **unsatisfiable**.
 2. All non blocked selections from the tableaux don't lead to an extension.
Knowledge Base is **satisfiable**.



11: EXTRA - FOL - Canonical Forms

OpenHPI - Course Knowledge Engineering with Semantic Web Technologies

Lecture 3: Ontologies and Logic