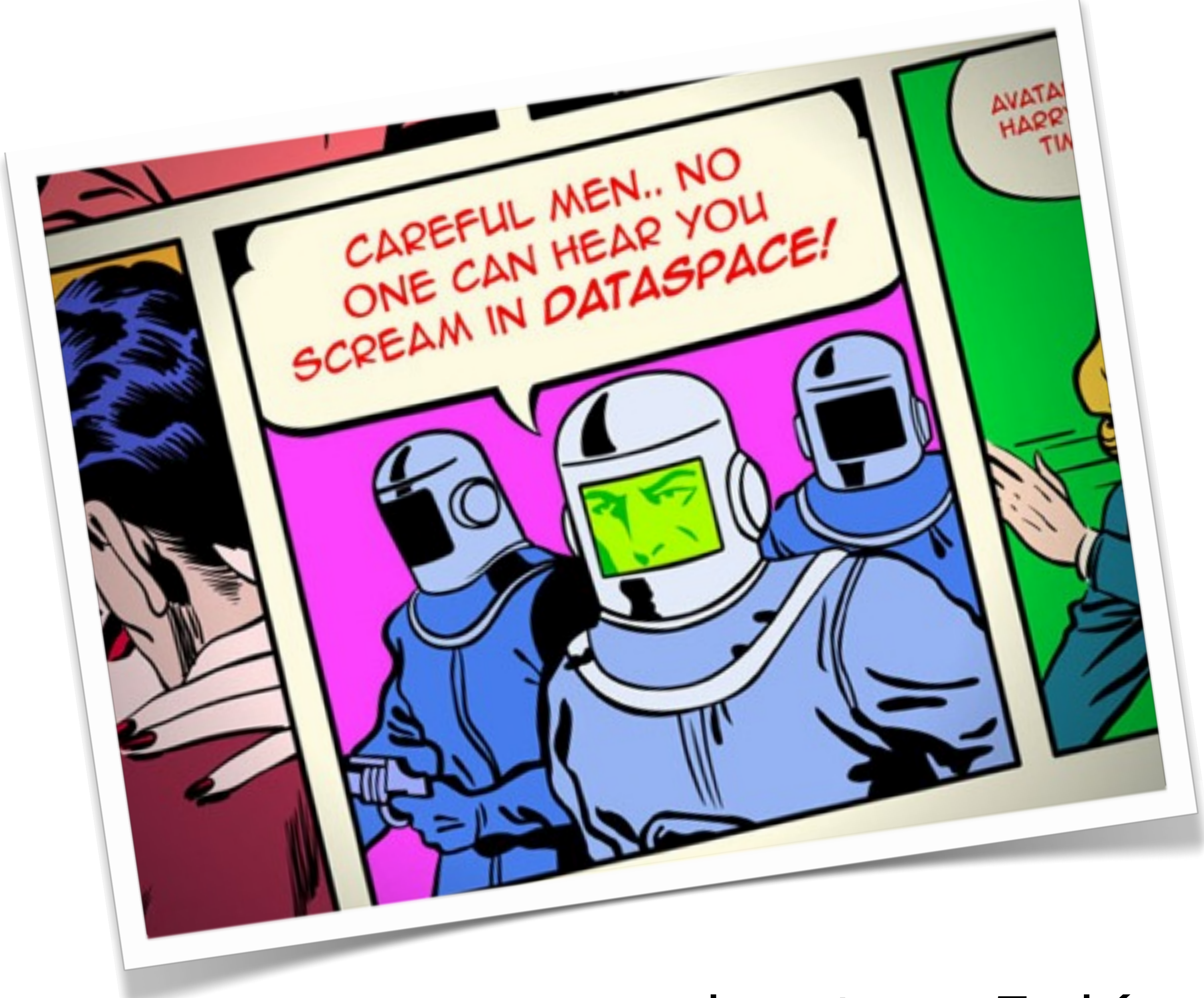


Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Semantic Web Technologies

Lecture 5: Knowledge Representations II

03: Tableaux Algorithm for ALC

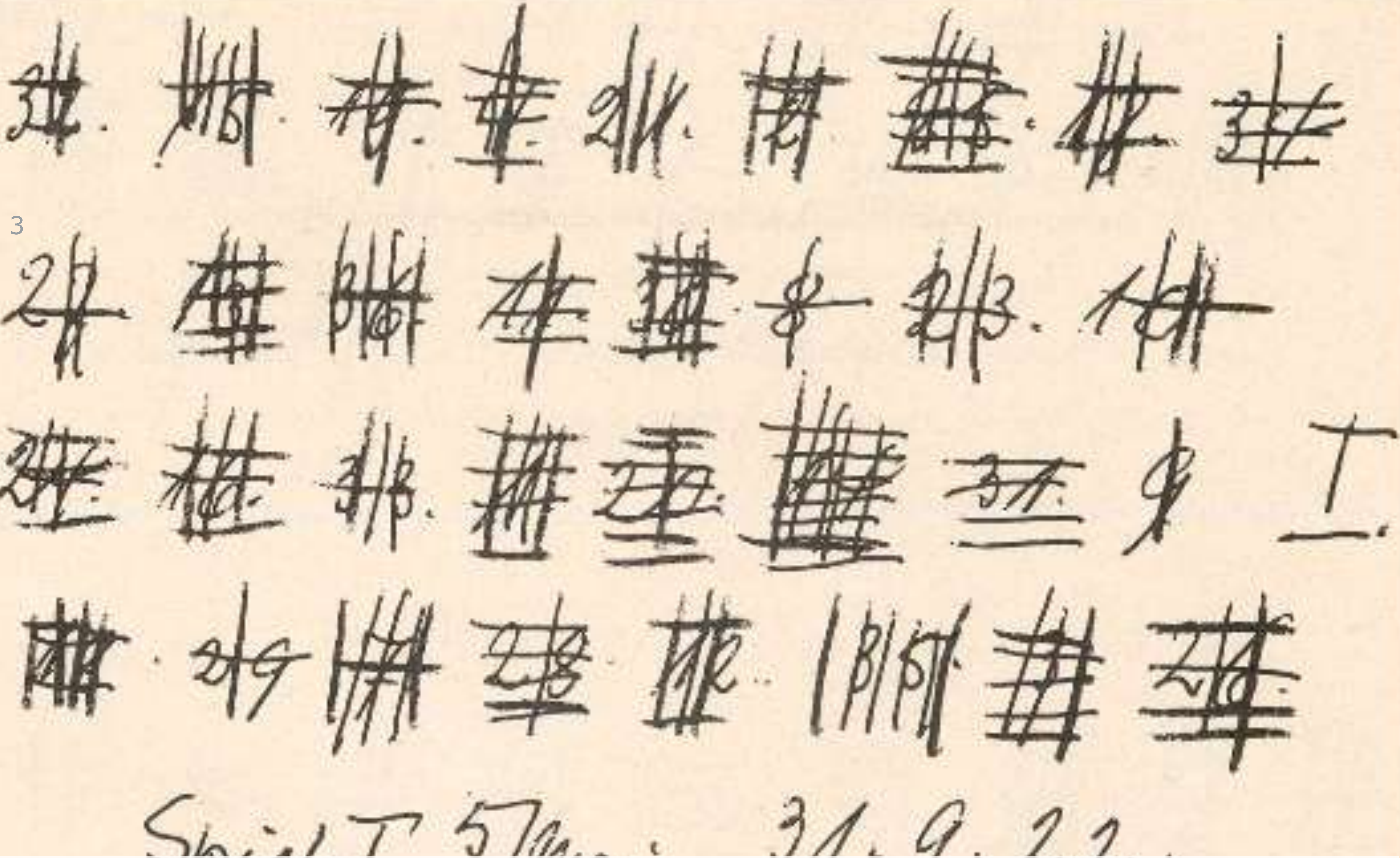
Dr. Harald Sack
Hasso Plattner Institute for IT Systems Engineering
University of Potsdam

Spring 2013

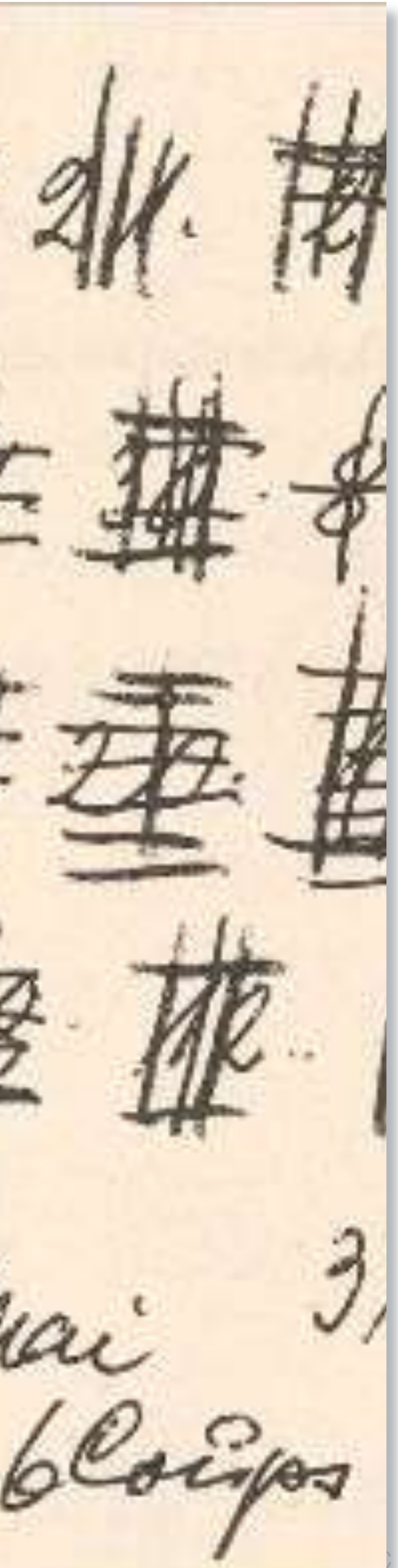
Hasso Plattner Institut

Lecture 5: Knowledge Representations II

Open HPI - Course: Semantic Web Technologies

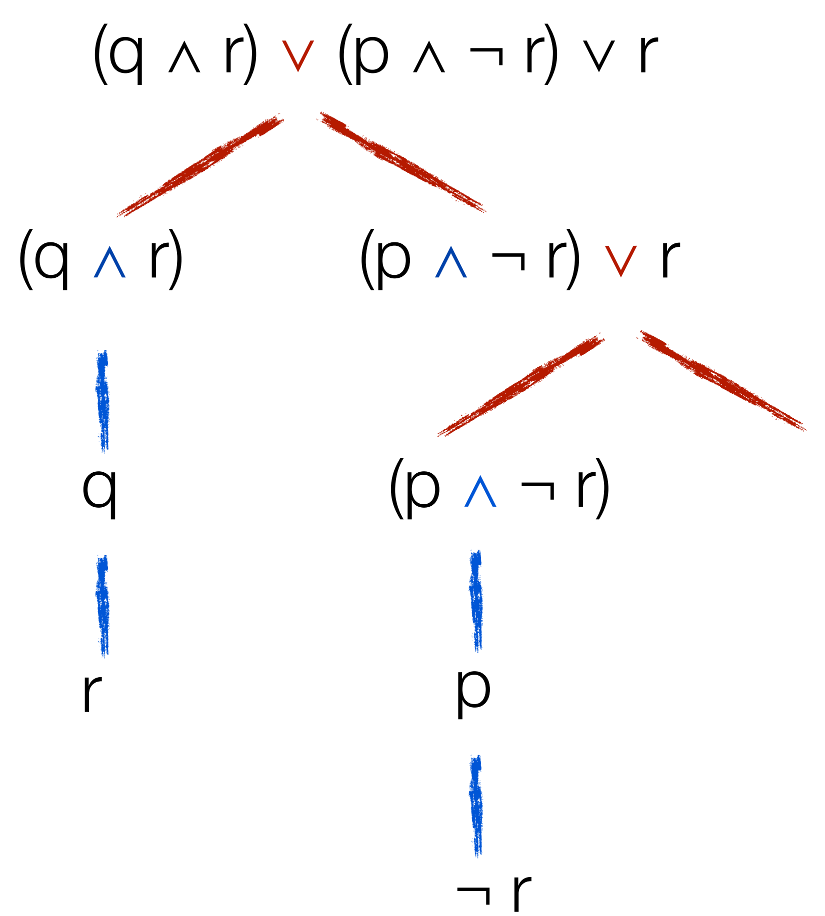


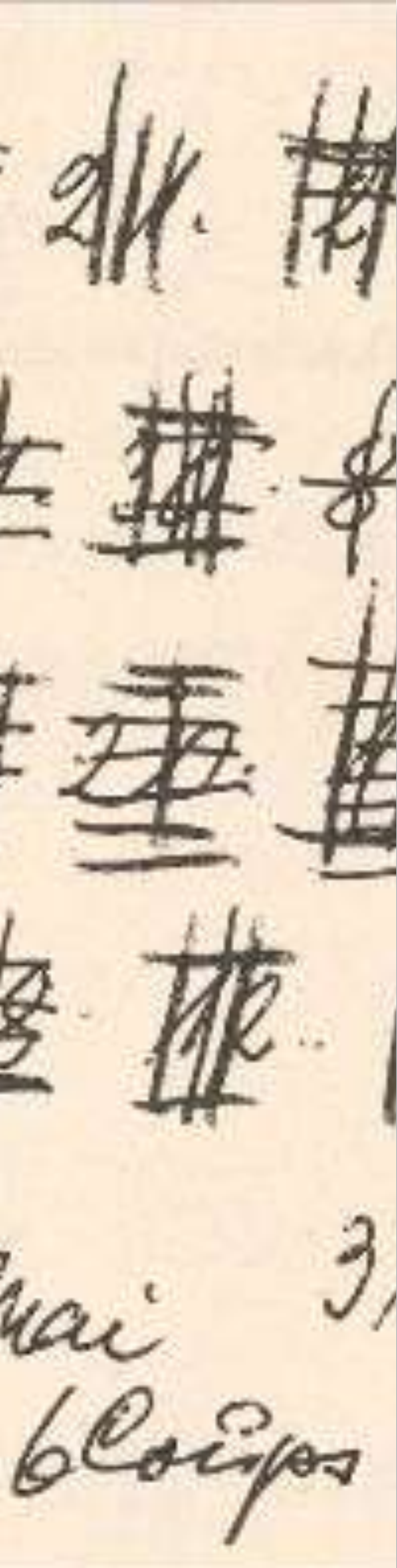
03 Tableaux Algorithm for ALC



Short Recapitulation:

- Proof algorithm to check the consistency of a logical formula by inferring that its negation is a contradiction (*proof by refutation*).
- Construct 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 Description Logics

- Transformation to *Negation normalform* necessary

- Let $\mathtt{W}$ 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 from the next page

- Resulting knowledge base $\mathtt{NNF(W)}$
  - Negation normalform of W.
  - Negation is placed directly in front of atomic classes.

Tableaux Transformation in Negation Normalform

- NNF Transformations

```
NNF(C) = C, if C is atomic
NNF(¬C) = ¬C, if C is atomic
NNF(¬¬C) = NNF(C)
NNF(C⊔D) = NNF(C) ⊔ NNF(D)
NNF(C⊓D) = NNF(C) ⊓ NNF(D)
NNF(¬(C⊔D)) = NNF(¬C) ⊓ NNF(¬D)
NNF(¬(C⊓D)) = NNF(¬C) ⊔ NNF(¬D)
NNF(∀R.C) = ∀R.NNF(C)
NNF(∃R.C) = ∃R.NNF(C)
NNF(¬∀R.C) = ∃R.NNF(¬C)
NNF(¬∃R.C) = ∀R.NNF(¬C)
```

- `W` and `NNF(W)` are logically equivalent.

Tableaux Transformation in Negation Normalform

- Example: $P \sqsubseteq (E \sqcap U) \sqcup \neg(\neg E \sqcup D)$
- In NNF: $\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D)$

$C \sqsubseteq D = \neg C \sqcup D$

$\neg(C \sqcup D) = \neg C \sqcap \neg D$

Tableaux Extension Rules for DL

Selection	Action
$C(a) \in W$ (ABox)	Add $C(a)$
$R(a,b) \in W$ (ABox)	Add $R(a,b)$
$C \in W$ (TBox)	Add $C(a)$ for a known Individual a
$(C \sqcap D)(a) \in A$	Add $C(a)$ and $D(a)$
$(C \sqcup D)(a) \in A$	Split the path. Add (1) $C(a)$ and (2) $D(a)$
$(\exists R.C)(a) \in A$	Add $R(a,b)$ and $C(b)$ for a new Individual b
$(\forall R.C)(a) \in A$	if $R(a,b) \in A$, then add $C(b)$

- 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 (DL) Example(4):

- P … Professor
- E … Person
- U … University Employee
- D … Student

- Knowledge Base: $P \sqsubseteq (E \sqcap U) \sqcup (E \sqcap \neg D)$
- Is $P \sqsubseteq E$ a logical consequence?

- Knowledge Base (with [negated] query) in NNF:
  $\{\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D),\ (P \sqcap \neg E)(a)\}$

Tableaux Algorithm (DL) Example(4):

- Knowledge Base W:{¬P⊔(E⊓U)⊔(E⊓¬D),(P⊓¬E)(a)}

Tableaux Algorithm (DL) Example(4):

- Knowledge Base $W:\{\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D), (P \sqcap \neg E)(a)\}$
- Tableaux A:

Tableaux Algorithm (DL) Example(4):

- Knowledge Base $\mathtt{W}:\{\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D), (P \sqcap \neg E)(a)\}$
- Tableaux $\mathtt{A}$:

  (1) $(P \sqcap \neg E)(a)$ (from knowledge base)

Tableaux Algorithm (DL) Example(4):

- Knowledge Base $W: \{\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D), (P \sqcap \neg E)(a)\}$
- Tableaux A:

  (1) $(P \sqcap \neg E)(a)$ (from knowledge base)

  (2|α from 1) $P(a)$

Tableaux Algorithm (DL) Example(4):

- Knowledge Base $\mathtt{W}$: $\{\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D), (P \sqcap \neg E)(a)\}$
- Tableaux $\mathtt{A}$:

  (1) $(P \sqcap \neg E)(a)$ (from knowledge base)

  (2|α from 1) $P(a)$

  (3|α from 1) $\neg E(a)$

Tableaux Algorithm (DL) Example(4):

- Knowledge Base $W: \{\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D), (P \sqcap \neg E)(a)\}$
- Tableaux A:

(1) $(P \sqcap \neg E)(a)$ (from knowledge base)

(2|α from 1) $P(a)$

(3|α from 1) $\neg E(a)$

(4) $(\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D))(a)$ (from knowledge base)

Tableaux Algorithm (DL) Example(4):

- Knowledge Base $\mathtt{W}$: $\{\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D), (P \sqcap \neg E)(a)\}$
- Tableaux $\mathtt{A}$:

(1) $(P \sqcap \neg E)(a)$ (from knowledge base)

(2|α from 1) $P(a)$

(3|α from 1) $\neg E(a)$

(4) $(\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D))(a)$ (from knowledge base)

(5|β from 4)) $\underline{\neg P(a)}$ | (6) $((E \sqcap U) \sqcup (E \sqcap \neg D))(a)$

Tableaux Algorithm (DL) Example(4):

- Knowledge Base W: $\{\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D), (P \sqcap \neg E)(a)\}$
- Tableaux A:

(1) $(P \sqcap \neg E)(a)$ (from knowledge base)

(2|α from 1) $P(a)$

(3|α from 1) $\neg E(a)$

(4) $(\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D))(a)$ (from knowledge base)

(5|β from 4)) $\underline{\neg P(a)}$ | (6) $((E \sqcap U) \sqcup (E \sqcap \neg D))(a)$

(7|β from 6) $(E \sqcap U)(a)$ | (8) $(E \sqcap \neg D)(a)$

Tableaux Algorithm (DL) Example(4):

- Knowledge Base W: $\{\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D), (P \sqcap \neg E)(a)\}$
- Tableaux A:

(1) $(P \sqcap \neg E)(a)$ (from knowledge base)

(2|α from 1) $P(a)$

(3|α from 1) $\neg E(a)$

(4) $(\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D))(a)$ (from knowledge base)

(5|β from 4)) $\underline{\neg P(a)}$ | (6) $((E \sqcap U) \sqcup (E \sqcap \neg D))(a)$

(7|β from 6) $(E \sqcap U)(a)$ | (8) $(E \sqcap \neg D)(a)$

(9|α from 7) $\underline{E(a)}$

Tableaux Algorithm (DL) Example(4):

- Knowledge Base $W:\{\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D), (P \sqcap \neg E)(a)\}$
- Tableaux A:

(1) $(P \sqcap \neg E)(a)$ (from knowledge base)

(2|α from 1) $P(a)$

(3|α from 1) $\neg E(a)$

(4) $(\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D))(a)$ (from knowledge base)

(5|β from 4)) $\underline{\neg P(a)}$ | (6) $((E \sqcap U) \sqcup (E \sqcap \neg D))(a)$

(7|β from 6) $(E \sqcap U)(a)$ | (8) $(E \sqcap \neg D)(a)$

(9|α from 7) $\underline{E(a)}$

(11|α from 7) $U(a)$

Tableaux Algorithm (DL) Example(4):

- Knowledge Base $W: \{\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D), (P \sqcap \neg E)(a)\}$
- Tableaux A:

(1) $(P \sqcap \neg E)(a)$ (from knowledge base)

(2|α from 1) $P(a)$

(3|α from 1) $\neg E(a)$

(4) $(\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D))(a)$ (from knowledge base)

(5|β from 4)) $\underline{\neg P(a)}$ | (6) $((E \sqcap U) \sqcup (E \sqcap \neg D))(a)$

(7|β from 6) $(E \sqcap U)(a)$ | (8) $(E \sqcap \neg D)(a)$

(9|α from 7) $\underline{E(a)}$ (10|α from 8) $\underline{E(a)}$

(11|α from 7) $U(a)$

Tableaux Algorithm (DL) Example(4):

- Knowledge Base W: $\{\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D), (P \sqcap \neg E)(a)\}$
- Tableaux A:

(1) $(P \sqcap \neg E)(a)$ (from knowledge base)

(2|α from 1) $P(a)$

(3|α from 1) $\neg E(a)$

(4) $(\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D))(a)$ (from knowledge base)

(5|β from 4)) $\underline{\neg P(a)}$ | (6) $((E \sqcap U) \sqcup (E \sqcap \neg D))(a)$

(7|β from 6) $(E \sqcap U)(a)$ | (8) $(E \sqcap \neg D)(a)$

(9|α from 7) $\underline{E(a)}$ (10|α from 8) $\underline{E(a)}$

(11|α from 7) $U(a)$ (12|α from 8) $\neg D(a)$

Tableaux Algorithm (DL) Example(5):

- Knowledge Base: $\neg\text{Person} \sqcup \exists\text{hasParent.Person}$
- infer: $\neg\text{Person(Bill)}$

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

Tableaux Algorithm (DL) Example(5):

- Knowledge Base: $\neg\text{Person} \sqcup \exists\text{hasParent.Person}$
- infer: $\neg\text{Person(Bill)}$

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

Person(Bill)

Tableaux Algorithm (DL) Example(5):

- Knowledge Base: $\neg$Person $\sqcup$ $\exists$hasParent.Person
- infer: $\neg$Person(Bill)

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

Person(Bill)
($\neg$Person $\sqcup$ $\exists$hasParent.Person)(Bill)

Tableaux Algorithm (DL) Example(5):

- Knowledge Base: ¬Person ⊔ ∃hasParent.Person
- infer: ¬Person(Bill)

{ ¬Person ⊔ ∃hasParent.Person, Person(Bill) }

Person(Bill)

(¬Person ⊔ ∃hasParent.Person)(Bill)

¬Person(Bill)

∃hasParent.Person(Bill) ← ⊔

Tableaux Algorithm (DL) Example(5):

- Knowledge Base: ¬Person ⊔ ∃hasParent.Person
- infer: ¬Person(Bill)

{ ¬Person ⊔ ∃hasParent.Person, Person(Bill) }

Person(Bill)

(¬Person ⊔ ∃hasParent.Person)(Bill)

¬Person(Bill)   ∃hasParent.Person(Bill) ⊔

STOP

Tableaux Algorithm (DL) Example(5):

- Knowledge Base: $\neg$Person $\sqcup$ $\exists$hasParent.Person
- infer: $\neg$Person(Bill)

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

Person(Bill)

($\neg$Person $\sqcup$ $\exists$hasParent.Person)(Bill)

$\neg$Person(Bill)

$\exists$hasParent.Person(Bill) ← $\sqcup$

hasParent(Bill,x_1)

Person(x_1) ← $\exists$

Tableaux Algorithm (DL) Example(5):

- Knowledge Base: ¬Person ⊔ ∃hasParent.Person
- infer: ¬Person(Bill)

{ ¬Person ⊔ ∃hasParent.Person, Person(Bill) }

Person(Bill)

(¬Person ⊔ ∃hasParent.Person)(Bill)

¬Person(Bill)

∃hasParent.Person(Bill) ← ⊔

hasParent(Bill,x_1)

Person(x_1) ← ∃

(¬Person ⊔ ∃hasParent.Person)(x_1)

Tableaux Algorithm (DL) Example(5):

- Knowledge Base: $\neg$Person $\sqcup$ $\exists$hasParent.Person
- infer: $\neg$Person(Bill)

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

Person(Bill)

($\neg$Person $\sqcup$ $\exists$hasParent.Person)(Bill)

$\neg$Person(Bill) | $\exists$hasParent.Person(Bill) $\leftarrow$ $\sqcup$

hasParent(Bill,x_1)

Person(x_1) $\leftarrow$ $\exists$

($\neg$Person $\sqcup$ $\exists$hasParent.Person)(x_1)

$\neg$Person(x_1) | $\exists$hasParent.Person(x_1) $\leftarrow$ $\sqcup$

Tableaux Algorithm (DL) Example(5):

- Knowledge Base: ¬Person ⊔ ∃hasParent.Person
- infer: ¬Person(Bill)

{ ¬Person ⊔ ∃hasParent.Person, Person(Bill) }

Person(Bill)

(¬Person ⊔ ∃hasParent.Person)(Bill)

¬Person(Bill) | ∃hasParent.Person(Bill) ← ⊔

hasParent(Bill,x_1)

Person(x_1) ← ∃

(¬Person ⊔ ∃hasParent.Person)(x_1)

¬Person(x_1) | ∃hasParent.Person(x_1) ← ⊔

Tableaux Algorithm (DL) Example(5):

- Knowledge Base: $\neg\text{Person} \sqcup \exists\text{hasParent.Person}$
- infer: $\neg$Person(Bill)

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

Person(Bill)

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

$\neg$Person(Bill)

$\exists$hasParent.Person(Bill) $\sqcup$

hasParent(Bill,x_1)

Person(x_1) $\exists$

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

$\neg$Person(x_1)

$\exists$hasParent.Person(x_1) $\sqcup$

hasParent(x_1,x_2)

Person(x_2) $\exists$

Tableaux Algorithm (DL) Example(5):

- Knowledge Base: ¬Person ⊔ ∃hasParent.Person
- infer: ¬Person(Bill)

{ ¬Person ⊔ ∃hasParent.Person, Person(Bill) }

Person(Bill)

(¬Person ⊔ ∃hasParent.Person)(Bill)

¬Person(Bill)

∃hasParent.Person(Bill) ⊔

hasParent(Bill,x_1)

Person(x_1) ∃

(¬Person ⊔ ∃hasParent.Person)(x_1)

¬Person(x_1)

∃hasParent.Person(x_1) ⊔

hasParent(x_1,x_2)

Person(x_2) ∃

(¬Person ⊔ ∃hasParent.Person)(x_2)

Tableaux Algorithm (DL) Example(5):

- Knowledge Base: $\neg Person \sqcup \exists hasParent.Person$
- infer: $\neg Person(Bill)$ $\{ \neg Person \sqcup \exists hasParent.Person, Person(Bill) \}$

$Person(Bill)$

$(\neg Person \sqcup \exists hasParent.Person)(Bill)$

$\neg Person(Bill)$ | $\exists hasParent.Person(Bill)$ ← $\sqcup$

$hasParent(Bill,x_1)$

$Person(x_1)$ ← $\exists$

$(\neg Person \sqcup \exists hasParent.Person)(x_1)$

$\neg Person(x_1)$ | $\exists hasParent.Person(x_1)$ ← $\sqcup$

$hasParent(x_1,x_2)$

$Person(x_2)$ ← $\exists$

$(\neg Person \sqcup \exists hasParent.Person)(x_2)$

$\neg Person(x_2)$ | $\exists hasParent.Person(x_2)$ ← $\sqcup$

Tableaux Algorithm (DL) Example(5):

- Knowledge Base: $\neg Person \sqcup \exists hasParent.Person$
- infer: $\neg Person(Bill)$

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

$Person(Bill)$

$(\neg Person \sqcup \exists hasParent.Person)(Bill)$

$\neg Person(Bill)$ | $\exists hasParent.Person(Bill)$ ← $\sqcup$

$hasParent(Bill,x_1)$

$Person(x_1)$ ← $\exists$

$(\neg Person \sqcup \exists hasParent.Person)(x_1)$

$\neg Person(x_1)$ | $\exists hasParent.Person(x_1)$ ← $\sqcup$

$hasParent(x_1,x_2)$

$Person(x_2)$ ← $\exists$

$(\neg Person \sqcup \exists hasParent.Person)(x_2)$

$\neg Person(x_2)$ | $\exists hasParent.Person(x_2)$ ← $\sqcup$

Tableaux Algorithm (DL) Example(5):

- Knowledge Base: $\neg\text{Person} \sqcup \exists\text{hasParent.Person}$
- infer: $\neg\text{Person(Bill)}$

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

Person(Bill)

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

$\neg\text{Person(Bill)}$

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

$\text{hasParent}(\text{Bill}, x_1)$

$\text{Person}(x_1)$ $\leftarrow \exists$

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

$\neg\text{Person}(x_1)$

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

$\text{hasParent}(x_1, x_2)$

$\text{Person}(x_2)$ $\leftarrow \exists$

$(\neg\text{Person} \sqcup \exists\text{hasParent.Person})(x_2)$

$\neg\text{Person}(x_2)$

$\exists\text{hasParent.Person}(x_2)$ $\leftarrow \sqcup$

Tableaux Algorithm (DL) Example(5):

- Knowledge Base: ¬Person ⊔ ∃hasParent.Person
- infer: ¬Person(Bill)

{ ¬Person ⊔ ∃hasParent.Person, Person(Bill) }

Person(Bill)

(¬Person ⊔ ∃hasParent.Person)(Bill)

¬Person(Bill) | ∃hasParent.Person(Bill) ← ⊔

hasParent(Bill,x_1)

Person(x_1) ← ∃

(¬Person ⊔ ∃hasParent.Person)(x_1)

¬Person(x_1) | ∃hasParent.Person(x_1) ← ⊔

hasParent(x_1,x_2)

Person(x_2) ← ∃

(¬Person ⊔ ∃hasParent.Person)(x_2)

¬Person(x_2) | ∃hasParent.Person(x_2) ← ⊔

Problem with existential quantification
also for OWL:minCardinality

Tableaux Algorithm (DL) Example(5):

- Knowledge Base: $\neg Person \sqcup \exists hasParent.Person$
- infer: $\neg Person(Bill)$

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

$Person(Bill)$

$(\neg Person \sqcup \exists hasParent.Person)(Bill)$

$\neg Person(Bill)$ | $\exists hasParent.Person(Bill)$ ← $\sqcup$

STOP

$hasParent(Bill,x_1)$

$Person(x_1)$ ← $\exists$

$(\neg Person \sqcup \exists hasParent.Person)(x_1)$

$\neg Person(x_1)$ | $\exists hasParent.Person(x_1)$ ← $\sqcup$

STOP

$hasParent(x_1,x_2)$

$Person(x_2)$ ← $\exists$

$(\neg Person \sqcup \exists hasParent.Person)(x_2)$

$\neg Person(x_2)$ | $\exists hasParent.Person(x_2)$ ← $\sqcup$

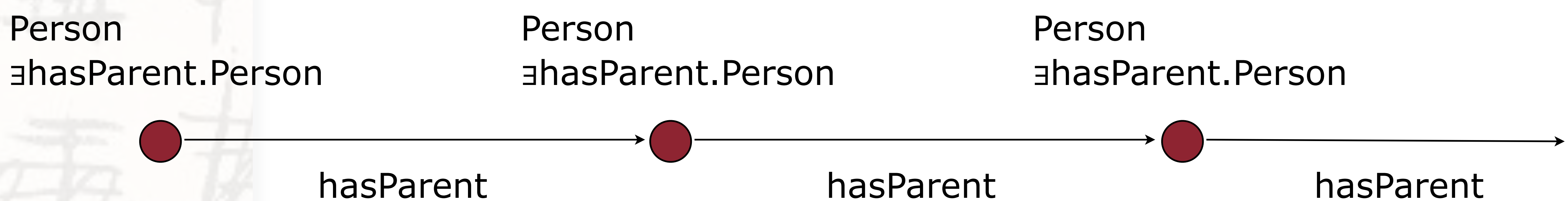
STOP

no termination possible

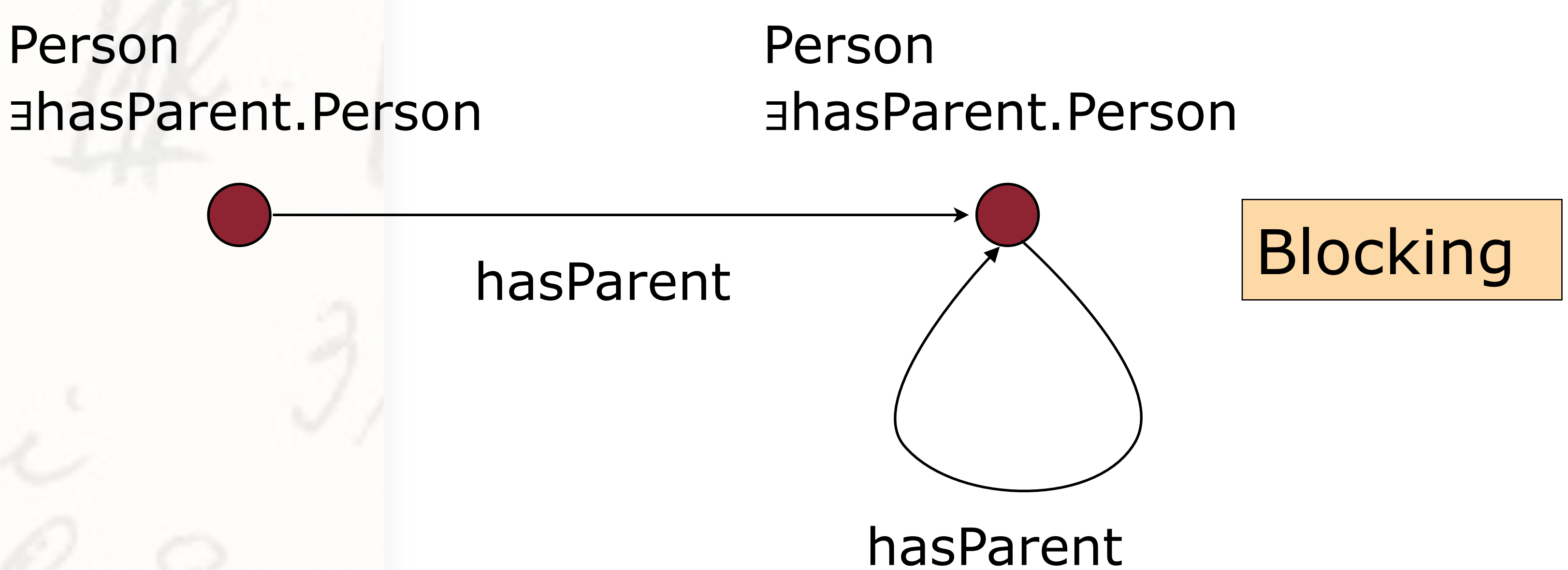
Problem with existential quantification
also for OWL:minCardinality

Idea of Blocking

- the following had been constructed in the Tableaux:



- Idea: reuse old nodes



Correctness of this short cut must be proven!

Tableaux Algorithm (DL) with Blocking

- Knowledge Base: ¬Person ⊔ ∃hasParent.Person
- infer: ¬Person(Bill)

Person(Bill)

(¬Person ⊔ ∃hasParent.Person)(Bill)

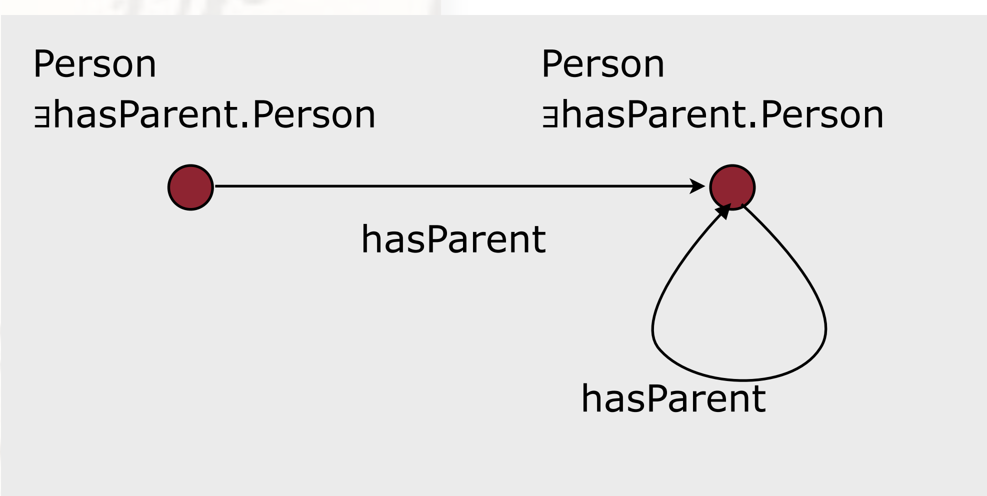
¬Person(Bill) | ∃hasParent.Person(Bill) ← ⊔

hasParent(Bill,x_1)

Person(x_1) ← ∃

(¬Person ⊔ ∃hasParent.Person)(x_1)

¬Person(x_1) | ∃hasParent.Person(x_1) ← ⊔



σ(Bill) = {Person,
¬Person ⊔ ∃hasParent.Person,
∃hasParent.Person}

σ(x_1) = { Person,
¬Person ⊔ ∃hasParent.Person,
∃hasParent.Person}

σ(x_1) ⊆ σ(Bill), so Bill blocks x_1

Tableaux Algorithm (DL) with Blocking

- Knowledge Base: ¬Person ⊔ ∃hasParent.Person
- infer: ¬Person(Bill)

Person(Bill)

(¬Person ⊔ ∃hasParent.Person)(Bill)

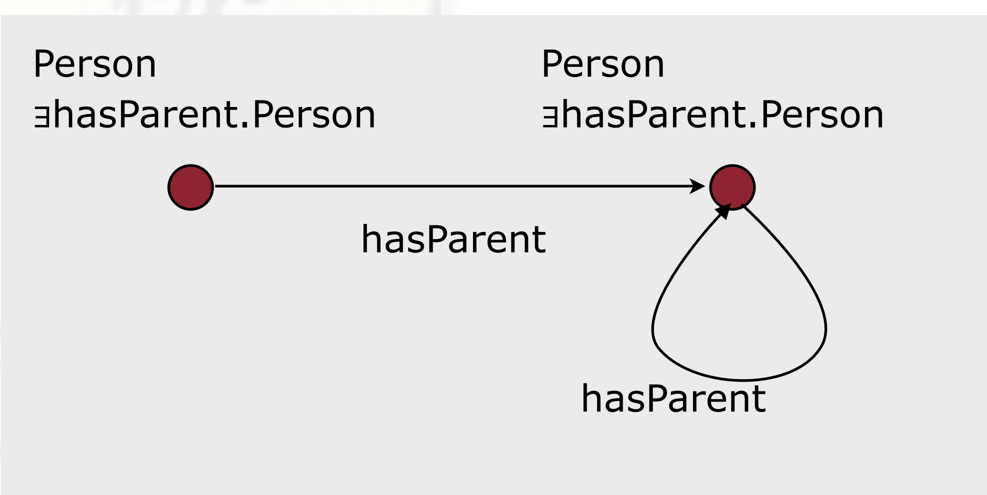
¬Person(Bill) STOP | ∃hasParent.Person(Bill) ← ⊔

hasParent(Bill,x_1)

Person(x_1) ← ∃

(¬Person ⊔ ∃hasParent.Person)(x_1)

¬Person(x_1) STOP | ∃hasParent.Person(x_1) ← ⊔



σ(Bill) = {Person,
¬Person ⊔ ∃hasParent.Person,
∃hasParent.Person}

σ(x_1) = { Person,
¬Person ⊔ ∃hasParent.Person,
∃hasParent.Person}

σ(x_1) ⊆ σ(Bill), so Bill blocks x_1 STOP termination

Tableaux Algorithm (DL) with 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**.

15

04 Web Ontology Language - OWL