UNIVERSITY OF ABERDEEN

Knowledge-Based Systems

Description Logic Reasoning - Why and how did that happen

Jeff Z. Pan

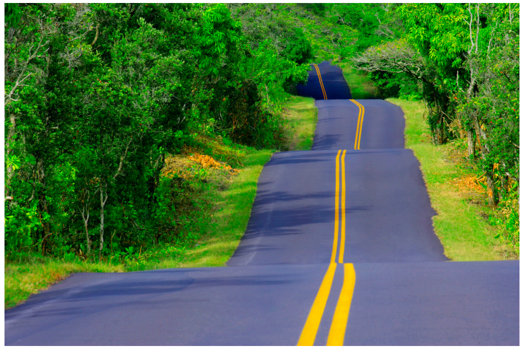
http://homepages.abdn.ac.uk/jeff.z.pan/pages/

UNIVERSITY OF ABERDEEN

Assessments

- 25% Continuous Assessment
  - Demonstrate practical skills related to ontology and rule, including building a **reasoner**
- **The TrOWL Award**
  - **for the best reasoner among the assessment submissions**
  - **to be announced at the revision lecture**

Roadmap



- Foundation
  - KR, ontology and rule; set theory
- Knowledge capture
- Knowledge representation
  - Ontology: Semantic Web standards RDF and OWL, Description Logics
  - Rule: Jess
- Knowledge reasoning
  - Ontology: formal semantics, tableaux algorithm
  - Rule: forward chaining, backward chaining
- Knowledge reuse and evaluation
- Meeting the real world
  - Jess and Java, Efficiency, Invited talk

UNIVERSITY OF ABERDEEN

Database and RDF

- An RDF statement is a data unit with global and linkable IDs for data and schema

Student ID	Name	take-course
p001	John	cs3019
p002	Tom	cs3023

  - [csd:p001 rdf:type csd:Student .]
  - [csd:p002 rdf:type csd:Student .]
  - [csd:p001 csd:name “John” .]
  - [csd:p002 csd:name “Tom” .]
  - [csd:p001 csd:take-course csd:cs3019 .]
  - [csd:p002 csd:take-course csd:cs3023 .]

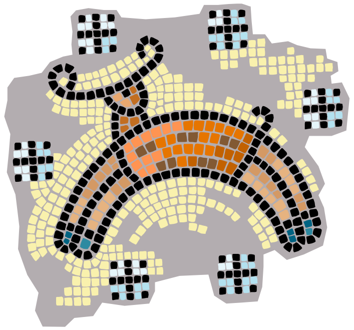
Schema and Data

- Schema:
  - [csd:UndgStudent rdfs:subClassOf csd:Student .]
  - [csd:take-course rdfs:range csd:Course .]
- Data:

Student ID	Name	take-course
p001	John	cs3019
p002	Tom	cs3023

  - [csd:p001 rdf:type csd:Student .]
  - [csd:p002 rdf:type csd:Student .]
  - [csd:p001 csd:name "John" .]
  - [csd:p002 csd:name "Tom" .]
  - [csd:p001 csd:take-course csd:cs3019 .]
  - [csd:p002 csd:take-course csd:cs3023 .]

DL Exercise



Q: Write down the following OWL axioms in DL syntax

1. ObjectProperty (eats domain (Animal)
   range (LivingThing))

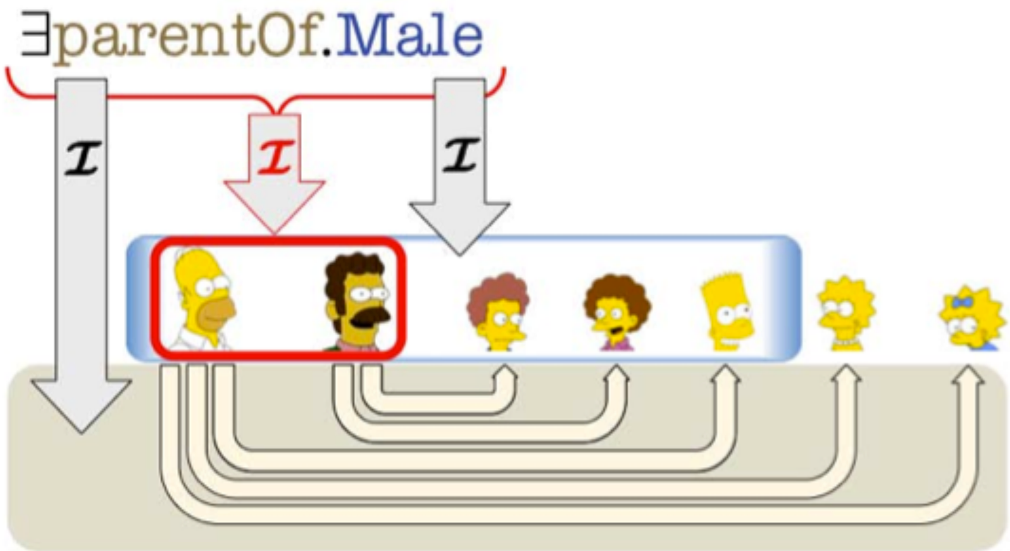
2. ObjectProperty (owns domain (Person)
   range (intersectionOf (LivingThing
   complementOf (Person))))

Interpretations of Class Descriptions

Given an interpretation, we can compute the semantic counterparts of class descriptions

$$\exists r.C = \{\ x \mid \exists y.\ (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\ \}$$

$$\forall r.C = \{\ x \mid \forall y.\ (x,y) \in r^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\ \}$$



Standards DL Reasoning Services

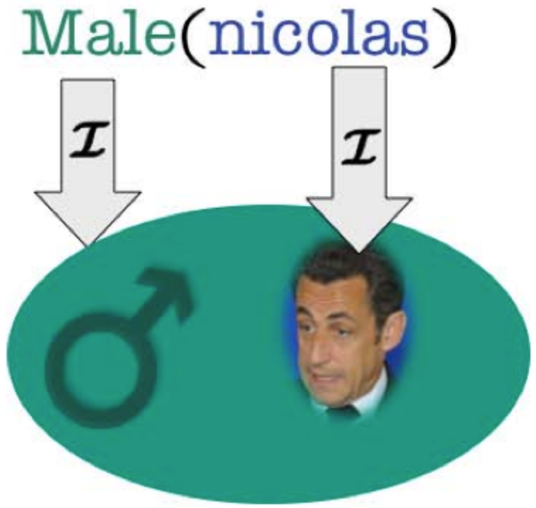
- "Easier" reasoning services
  - whether O is consistent
  - whether a given class is satisfiable
- "Harder" reasoning services
  - whether O entails a class inclusion axiom
  - whether O entails an individual axiom

Class Instance Checking

- Reducing Class Instance Checking to Ontology Consistency Checking
  - If O entails C(x), then in every interpretation I of O, we have x^I is in C^I
  - It means $O \cup \{\neg C(x)\}$ is inconsistent

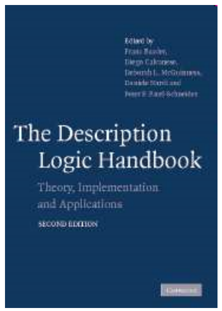
Example

- If an ontology O entails Male (nicolas)
  - then in every interpretation I of O
  - we have $nicolas^I \in Male^I$
- Now if we extend O to O' with a new axiom
  - ¬Male(nicolas) (*)
- How to construct an interpretation I' for O'?
  - as all interpretations of O' should satisfy O
  - we could start from interpretations of O
- It is easy to see that I' does not exist
  - If I' does not satisfy O, then it does not satisfy O' either
  - If I' satisfies O, then it does not satisfy ¬Male(nicolas)



Lecture Outline

- Motivation
- Introduction to tableaux algorithms
- Some detailed discussions on tableaux algorithms
- Practical



[Section: 9.3.2.1]

Motivations: Reasoning

- How to perform DL reasoning based on formal semantics
  - Tableaux algorithm
- Most existing OWL DL reasoners implement tableaux algorithm

Lecture Outline

- Motivation
- Introduction to tableaux algorithms
  - The big picture
- Some detailed discussions on tableaux algorithms
- Practical

Tableaux Algorithm

- The first sound and complete algorithm for expressive DLs
  - Ontology Consistency Checking
  - Class Satisfiability Checking
- Basic idea: Build an interpretation
  - A **tableau** is a representative of an interpretation
  - We can construct an interpretation based on a tableau

Key Steps

1. Initialise the tableau with individual axioms
   – the initial tableau might not satisfy all the axioms
2. Repair the initial tableau by applying expansion rules
   – so as to add new information into the tableau
   – this might require backtracking
3. If the tableau satisfy all the axioms, returns Consistent
4. If every possible attempt repair results in some contradiction, returns Inconsistent
   – Contradiction: $\{A, \neg A\} \subseteq L(x)$ or $\{\bot\} \subseteq L(x)$ ($\bot$ is bottom, interpreted as **empty set**)

Tableau Initialisation for Ontology Consistency Checking

- For checking if an given ontology O is consistent
  - for every individual axiom o:C, construct a root x-o of a tableau, and set L(x-o)={C}
  - for every individual axiom <o1,o2>:R, construct an edge (x-o1,x-o2) between two roots x-o1 and x-o2 and set L(x-o1,x-o2)={R}.

UNIVERSITY OF ABERDEEN

Expansion Rule for Simple Axioms

- Simple axioms
  - $A \sqsubseteq C$ where A is a named class
  - No cycles involve A
    - such as $A \sqsubseteq \exists R.A$
- **Expansion rule for simple axioms**
  - If A is in L(x) and $A \sqsubseteq C$ is in O
  - Then add C into L(x)

Example: Ontology Consistency Checking

- Check if the following ontology is consistent
  1. English $\sqsubseteq \neg$Chinese
  2. Confucian $\sqsubseteq$Chinese
  3. Confucian $\sqsubseteq$English
  4. Bill : Confucian

x-Bill L(x-Bill)={**Confucian**}

5. Initialise the tableau: L(x-Bill)={**Confucian**} (from 4)

Example: Ontology Consistency Checking

- Check if the following ontology is consistent
  1. English $\sqsubseteq \neg$Chinese
  2. Confucian $\sqsubseteq$Chinese
  3. Confucian $\sqsubseteq$English
  4. Bill : Confucian
  5. Initialise the tableau: L(x-Bill)={Confucian}

x-Bill L(x-Bill)={Confucian, **Chinese, English**}

6. Expand L(x-Bill): L(x-Bill)={Confucian**, Chinese, English**} (from 2,3)

Example: Ontology Consistency Checking

- Check if the following ontology is consistent

1. English $\sqsubseteq \neg$Chinese
2. Confucian $\sqsubseteq$ Chinese
3. Confucian $\sqsubseteq$ English
4. Bill : Confucian
5. Initialise the tableau: L(x-Bill)={Confucian}
6. Expend L(x-Bill): L(x-Bill)={Confucian, Chinese, English}

x-Bill L(x-Bill)={Confucian, Chinese, English, ¬**Chinese**}

7. Expand L(x-Bill): L(x-Bill)={Confucian**, Chinese,** English**, ¬Chinese**} (from 1)

Example: Ontology Consistency Checking

- Check if the following ontology is consistent

1. English $\sqsubseteq$ ¬Chinese
2. Confucian $\sqsubseteq$ Chinese
3. Confucian $\sqsubseteq$ English
4. Bill : Confucian
5. Initialise the tableau: L(x-Bill)={Confucian}
6. Expand L(x-Bill): L(x-Bill)={Confucian, Chinese, English}

x-Bill L(x-Bill)={Confucian, Chinese, English, ¬Chinese}

7. Expand L(x-Bill): L(x-Bill)={Confucian, **Chinese**, English, **¬Chinese**} x-Bill can be Chinese and ¬Chinese at the same time
8. hence, the initial assumption that there exist an interpretation for O is incorrect; i.e., O is inconsistent

Lecture Outline

- Motivation
- Introduction to tableaux algorithms
- Some detailed discussions on tableaux algorithms
- Practical

Tableaux Algorithm

- The first sound and complete algorithm for expressive DLs
  - Ontology Consistency Checking
  - **Class Satisfiability Checking**
- Basic idea: Build an interpretation
  - A **tableau** is a representative of an interpretation
  - We can construct an interpretation based on a tableau

Tableau Initialisation for Class Satisfiability Checking

- A **tableau** is a representative of an interpretation

- For checking if C is satisfiable
  - Construct the root x0 of a tableau, and
  - set L(x0)={C}
  - C is the **input** class description
  - x0 is an instance of all classes in L(x0)

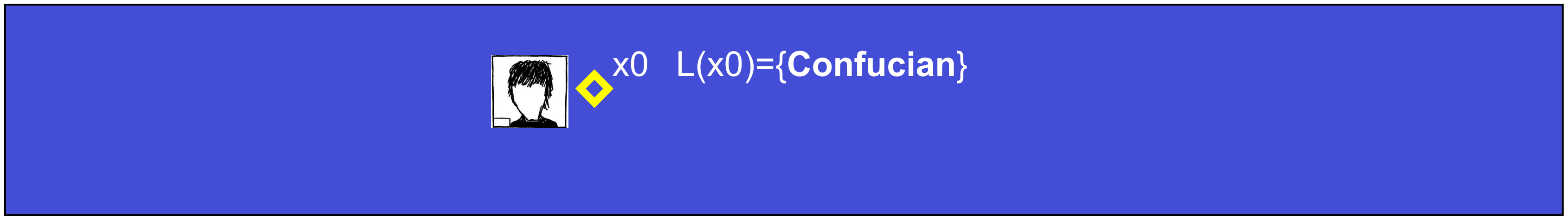
Tableau Initialisation for Class Satisfiability Checking

- What does it mean
  - It means we **assume** that there exists some object x0 as an instance of C
  - If applying expansion rules (see later slides) always leads to some contradiction
    - Contradiction: {A, :A} ⊆ L(x) or {⊥} ⊆ L(x) (⊥ is bottom, interpreted as **empty set**)
  - Then our assumption is incorrect --- so C cannot have any instances, i.e. C is unsatisfiable
  - Otherwise, C is satisfiable

Example: Constructing Tableaux

- Given the following ontology, check if Confucian is satisfiable

1. Chinese $\sqsubseteq$ Person
2. English $\sqsubseteq$ Person
3. Confucian $\sqsubseteq$ Chinese
4. Confucian $\sqsubseteq$ English

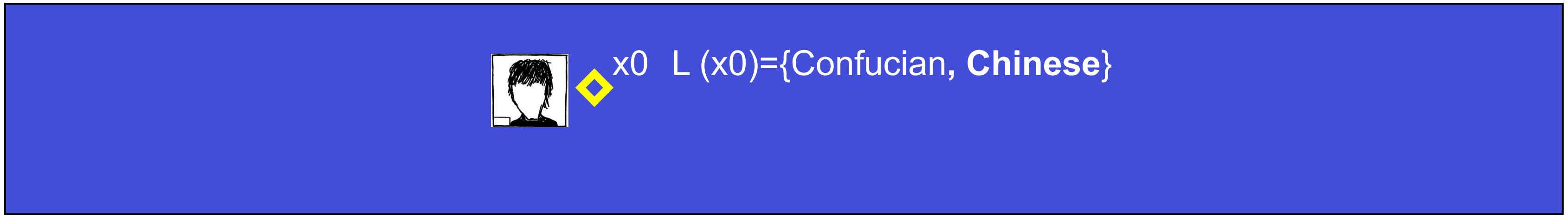


5. Initialise the tableau: L(x0)={**Confucian**}

Example: Constructing Tableaux

- Given the following ontology, check if Confucian is satisfiable

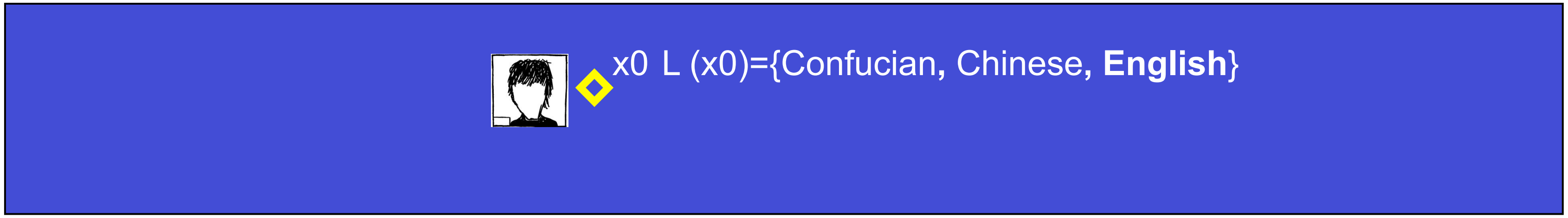
1. Chinese $\sqsubseteq$ Person
2. English $\sqsubseteq$ Person
3. Confucian $\sqsubseteq$ Chinese
4. Confucian $\sqsubseteq$ English
5. Initialise the tableau: L(x0)={Confucian}



6. Expand L(x0): L(x0)={Confucian**, Chinese**} (from 3)

Example: Constructing Tableaux

- Given the following ontology, check if Confucian is satisfiable
  1. Chinese $\sqsubseteq$ Person
  2. English $\sqsubseteq$ Person
  3. Confucian $\sqsubseteq$ Chinese
  4. Confucian $\sqsubseteq$ English
  5. Initialise the tableau: L(x0)={Confucian}
  6. Expend L(x0): L(x0)={Confucian**, Chinese**}



7. Expand L(x0): L(x0)={Confucian, Chinese**, English**} (from 4)

UNIVERSITY OF ABERDEEN

Example: Constructing Tableaux

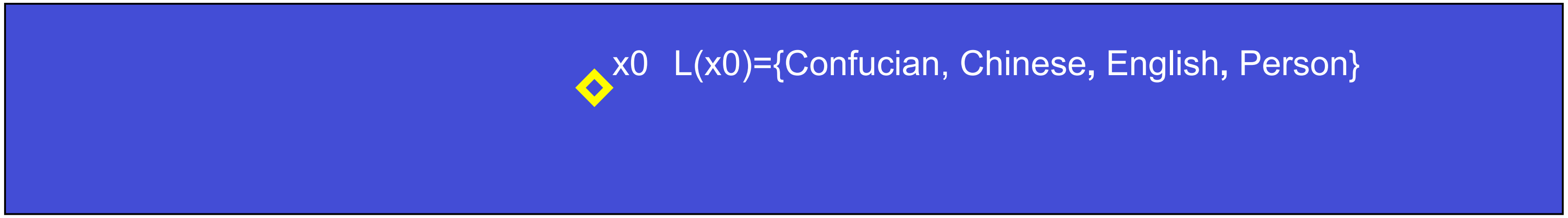
- Given the following ontology, check if Confucian is satisfiable

1. Chinese $\sqsubseteq$ Person
2. English $\sqsubseteq$ Person
3. Confucian $\sqsubseteq$ Chinese
4. Confucian $\sqsubseteq$ English
5. Initialise the tableau: L(x0)={Confucian}
6. Expand L(x0): L(x0)={Confucian**, Chinese**}
7. Expand L(x0): L(x0)={Confucian, Chinese**, English**}

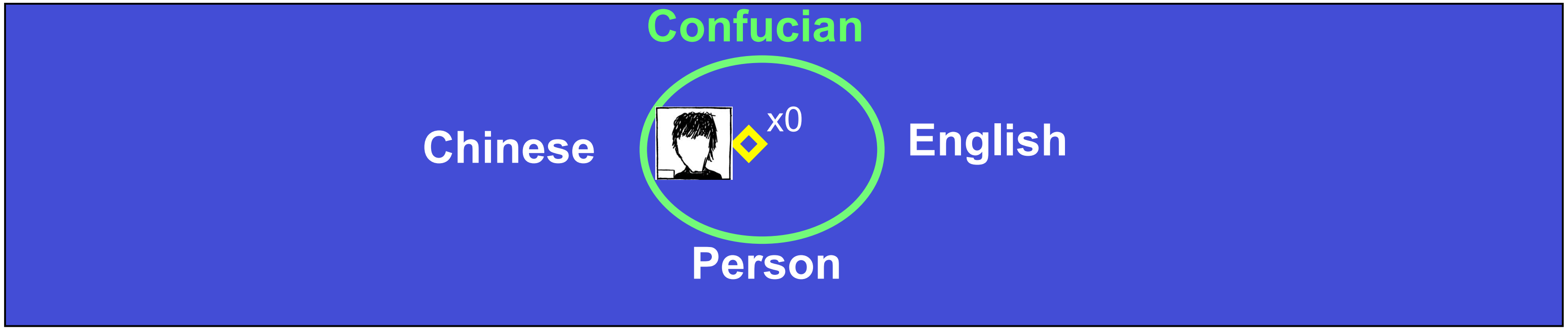
x0 L (x0)={Confucian, Chinese, English**, Person**}

8. Expand L(x0): L(x0)={Confucian**,** Chinese**,** English**, Person**} (from 1,2)

Example: From Tableaux to Interpretations



- According to the above tableau, we can construct the following interpretation
  - ✓ Δ^{I} = {x0}
  - ✓ Confucian^{I}={x0}, Chinese^{I}={x0}, English^{I}={x0}, Person^{I}={x0}
- ✓ So Confucian is satisfiable (because there is an interpretation, the above one, in which it is not an empty set)



Class Instance Checking

Question: Is the following ontology O consistent?

- Class (OldLady partial

    restriction (hasPet allValuesFrom (Cat)))

- Individual (Minnie type (OldLady)

    value (hasPet Tom))

  Individual (Tom type (complementOf (Cat))

UNIVERSITY OF ABERDEEN

Summary

- Tableaux algorithm
  - constructing a representative of an interpretation
- Class satisfiability checking and ontology consistency checking are fundamental reasoning problems