Logics and Statistics for Language Modeling

Carlos Areces

areces@loria.fr
http://www.loria.fr/~areces/ls
INRIA Nancy Grand Est
Nancy, France

2009/2010

Today's Program

Today's Program

- Description Logics
  - Syntax and Semantics
  - The Tableaux Method
  - Inference Tasks

Description Logics: Syntax

Description Logics: Syntax

- Atomic Concepts: $\mathsf{CON} = \{C_1, C_2, \ldots, C_n, \ldots\}$
- Roles: $\mathsf{ROL} = \{R_1, R_2, \ldots, R_n, \ldots\}$
- Individuals: $\mathsf{IND} = \{i_1, i_2, \ldots, i_n, \ldots\}$

Description Logics: Syntax

- Atomic Concepts: $\text{CON} = \{C_1, C_2, \ldots, C_n, \ldots\}$
- Roles: $\text{ROL} = \{R_1, R_2, \ldots, R_n, \ldots\}$
- Individuals: $\text{IND} = \{i_1, i_2, \ldots, i_n, \ldots\}$
- (Complex) Concepts:

$$\top \mid \neg C \mid C_1 \vee C_2 \mid \exists R.C$$

for C, C_1, C_2 concepts and R a role.

Description Logics: Syntax

- Atomic Concepts: $\text{CON} = \{C_1, C_2, \ldots, C_n, \ldots\}$
- Roles: $\text{ROL} = \{R_1, R_2, \ldots, R_n, \ldots\}$
- Individuals: $\text{IND} = \{i_1, i_2, \ldots, i_n, \ldots\}$
- (Complex) Concepts:

  $$\top \mid \neg C \mid C_1 \vee C_2 \mid \exists R.C$$

  for C, C_1, C_2 concepts and R a role.
- (Partial) Definitions:

  $$C_1 \sqsubseteq C_2$$

  for C_1, C_2 concepts.

Description Logics: Syntax

- Atomic Concepts: $\text{CON} = \{C_1, C_2, \ldots, C_n, \ldots\}$
- Roles: $\text{ROL} = \{R_1, R_2, \ldots, R_n, \ldots\}$
- Individuals: $\text{IND} = \{i_1, i_2, \ldots, i_n, \ldots\}$
- (Complex) Concepts:
$$\top \mid \neg C \mid C_1 \vee C_2 \mid \exists R.C$$
for C, C_1, C_2 concepts and R a role.
- (Partial) Definitions:
$$C_1 \sqsubseteq C_2$$
for C_1, C_2 concepts.
- Assertions:
$$i : C \mid (i_1, i_2) : R$$
for C a concept, i, i_1, i_2 individuals and R a role.

Description Logics: Semantics

Description Logics: Semantics

- Define the following translation T_x from DL to FOL:

Description Logics: Semantics

- Define the following translation T_x from DL to FOL:

$$T_x(C) \quad = \quad C(x) \text{ for } C \text{ atomic}$$

Description Logics: Semantics

- Define the following translation T_x from DL to FOL:

$$\begin{array}{rcl} T_x(C) & = & C(x) \text{ for } C \text{ atomic} \\ T_x(\neg C) & = & \neg T_x(C) \end{array}$$

Description Logics: Semantics

- Define the following translation T_x from DL to FOL:

$$\begin{array}{rcl} T_x(C) & = & C(x) \text{ for } C \text{ atomic} \\ T_x(\neg C) & = & \neg T_x(C) \\ T_x(C_1 \vee C_2) & = & T_x(C_1) \vee T_x(C_2) \end{array}$$

Description Logics: Semantics

- Define the following translation T_x from DL to FOL:

$$
\begin{array}{rcl}
T_x(C) & = & C(x) \text{ for } C \text{ atomic} \\
T_x(\neg C) & = & \neg T_x(C) \\
T_x(C_1 \vee C_2) & = & T_x(C_1) \vee T_x(C_2) \\
T_x(\exists R.C) & = & \exists y.(R(x,y) \wedge T_y(C)) \\
 & & \quad \text{for } y \text{ a new variable}
\end{array}
$$

Description Logics: Semantics

- Define the following translation T_x from DL to FOL:

$$
\begin{array}{rcl}
T_x(C) & = & C(x) \text{ for } C \text{ atomic} \\
T_x(\neg C) & = & \neg T_x(C) \\
T_x(C_1 \vee C_2) & = & T_x(C_1) \vee T_x(C_2) \\
T_x(\exists R.C) & = & \exists y.(R(x,y) \wedge T_y(C)) \\
 & & \quad \text{for } y \text{ a new variable} \\
T_x(C_1 \sqsubseteq C_2) & = & \forall x.(T_x(C_1) \rightarrow T_x(C_2))
\end{array}
$$

Description Logics: Semantics

- Define the following translation T_x from DL to FOL:

$$
\begin{array}{rcl}
T_x(C) & = & C(x) \text{ for } C \text{ atomic} \\
T_x(\neg C) & = & \neg T_x(C) \\
T_x(C_1 \vee C_2) & = & T_x(C_1) \vee T_x(C_2) \\
T_x(\exists R.C) & = & \exists y.(R(x,y) \wedge T_y(C)) \\
 & & \quad \text{for } y \text{ a new variable} \\
T_x(C_1 \sqsubseteq C_2) & = & \forall x.(T_x(C_1) \rightarrow T_x(C_2)) \\
T_x(i : C) & = & T_i(C)
\end{array}
$$

Description Logics: Semantics

- Define the following translation T_x from DL to FOL:

$$
\begin{array}{rcl}
T_x(C) & = & C(x) \text{ for } C \text{ atomic} \\
T_x(\neg C) & = & \neg T_x(C) \\
T_x(C_1 \vee C_2) & = & T_x(C_1) \vee T_x(C_2) \\
T_x(\exists R.C) & = & \exists y.(R(x,y) \wedge T_y(C)) \\
 & & \quad \text{for } y \text{ a new variable} \\
T_x(C_1 \sqsubseteq C_2) & = & \forall x.(T_x(C_1) \rightarrow T_x(C_2)) \\
T_x(i : C) & = & T_i(C) \\
T_x((i_1, i_2) : R) & = & R(i_1, i_2)
\end{array}
$$

Description Logics: Semantics

- Define the following translation T_x from DL to FOL:

$$\begin{array}{rcl} T_x(C) & = & C(x) \text{ for } C \text{ atomic} \\ T_x(\neg C) & = & \neg T_x(C) \\ T_x(C_1 \vee C_2) & = & T_x(C_1) \vee T_x(C_2) \\ T_x(\exists R.C) & = & \exists y.(R(x,y) \wedge T_y(C)) \\ & & \quad \text{for } y \text{ a new variable} \\ T_x(C_1 \sqsubseteq C_2) & = & \forall x.(T_x(C_1) \rightarrow T_x(C_2)) \\ T_x(i : C) & = & T_i(C) \\ T_x((i_1, i_2) : R) & = & R(i_1, i_2) \end{array}$$

- Let φ be a formula in DL. Then φ is satisfiable iff

$$M, g \models T_x(\varphi)$$

for M a first-order model for the signature $\langle \text{VARS}, \text{IND}, \{\}, \text{CON} \cup \text{ROL} \rangle$ and g an arbitrary assignment.

The Tableaux Method

The Tableaux Method

- What is a tableaux? It's a method to search for models

The Tableaux Method

- What is a tableaux? It's a method to search for models
  - It's a collection of formulas (assertions) organized as a tree.

The Tableaux Method

- What is a tableaux? It's a method to search for models
  - It's a collection of formulas (assertions) organized as a tree.
  - Each branch of the tree represent a (partial) model of the root formula.

The Tableaux Method

- What is a tableaux? It's a method to search for models
  - It's a collection of formulas (assertions) organized as a tree.
  - Each branch of the tree represent a (partial) model of the root formula.
  - Branches are expanded via tableaux rules.

The Tableaux Method

- What is a tableaux? It's a method to search for models
  - It's a collection of formulas (assertions) organized as a tree.
  - Each branch of the tree represent a (partial) model of the root formula.
  - Branches are expanded via tableaux rules.
  - If a branch contains a contradition it is closed.

The Tableaux Method

- What is a tableaux? It's a method to search for models
  - It's a collection of formulas (assertions) organized as a tree.
  - Each branch of the tree represent a (partial) model of the root formula.
  - Branches are expanded via tableaux rules.
  - If a branch contains a contradition it is closed.
  - If no further rule can be applied and there is at least a branch which is not closed, then we have found a model for the root.

The Tableaux Method

- What is a tableaux? It's a method to search for models
  - It's a collection of formulas (assertions) organized as a tree.
  - Each branch of the tree represent a (partial) model of the root formula.
  - Branches are expanded via tableaux rules.
  - If a branch contains a contradition it is closed.
  - If no further rule can be applied and there is at least a branch which is not closed, then we have found a model for the root.
- A branch is closed if for some C and some i, both $i : C$ and $i : \neg C$ are in the branch; or if $i : \neg\top$ is in the branch.

Tableaux Rules

Tableaux Rules

For Conjunction:

$$\frac{a : C_1 \wedge C_2}{\begin{array}{c} a : C_2 \\ a : C_1 \end{array}} \ (\wedge)$$

Tableaux Rules

For Conjunction:

$$\frac{a : C_1 \wedge C_2}{\begin{array}{c} a : C_2 \\ a : C_1 \end{array}} \ (\wedge) \qquad \frac{a : \neg(C_1 \wedge C_2)}{a : \neg C_1 \quad | \quad a : \neg C_2} \ (\neg\wedge)$$

Tableaux Rules

For Conjunction:

$$\frac{a : C_1 \wedge C_2}{\begin{array}{c} a : C_2 \\ a : C_1 \end{array}} \ (\wedge) \qquad \frac{a : \neg(C_1 \wedge C_2)}{a : \neg C_1 \quad | \quad a : \neg C_2} \ (\neg\wedge)$$

For Disjunction

$$\frac{a : C_1 \vee C_2}{a : C_1 \quad | \quad a : C_2} \ (\vee)$$

Tableaux Rules

For Conjunction:

$$\frac{a : C_1 \wedge C_2}{\begin{array}{c} a : C_2 \\ a : C_1 \end{array}} \ (\wedge) \qquad \frac{a : \neg(C_1 \wedge C_2)}{a : \neg C_1 \quad | \quad a : \neg C_2} \ (\neg\wedge)$$

For Disjunction

$$\frac{a : C_1 \vee C_2}{a : C_1 \quad | \quad a : C_2} \ (\vee) \qquad \frac{a : \neg(C_1 \vee C_2)}{\begin{array}{c} a : \neg C_2 \\ a : \neg C_1 \end{array}} \ (\neg\vee)$$

Tableaux Rules

For Conjunction:

$$\frac{a : C_1 \wedge C_2}{\begin{array}{c} a : C_2 \\ a : C_1 \end{array}} \ (\wedge) \qquad \frac{a : \neg(C_1 \wedge C_2)}{a : \neg C_1 \quad | \quad a : \neg C_2} \ (\neg\wedge)$$

For Disjunction

$$\frac{a : C_1 \vee C_2}{a : C_1 \quad | \quad a : C_2} \ (\vee) \qquad \frac{a : \neg(C_1 \vee C_2)}{\begin{array}{c} a : \neg C_2 \\ a : \neg C_1 \end{array}} \ (\neg\vee)$$

For Negation: $\frac{a : \neg\neg C}{a : C} \ (\neg)$

Tableaux Rules

Tableaux Rules

For Existential:

$$\frac{a : \exists R.C}{\begin{array}{c} b : C \\ (a, b) : R \end{array}} \ (\exists)$$

for b a new individual

Tableaux Rules

For Existential:

$$\frac{a : \exists R.C}{\begin{array}{c} b : C \\ (a,b) : R \end{array}} \ (\exists)$$

for b a new individual

$$\frac{\begin{array}{c} a : \neg(\exists R.C) \\ (a,b) : R \end{array}}{b : \neg C} \ (\neg\exists)$$

Tableaux Rules

For Existential:

$$\frac{a : \exists R.C}{\begin{array}{c} b : C \\ (a,b) : R \end{array}} \ (\exists) \qquad \frac{\begin{array}{c} a : \neg(\exists R.C) \\ (a,b) : R \end{array}}{b : \neg C} \ (\neg\exists)$$

for b a new individual

For Universal:

$$\frac{\begin{array}{c} a : \forall R.C \\ (a,b) : R \end{array}}{b : C} \ (\forall)$$

Tableaux Rules

For Existential:

$$\frac{a : \exists R.C}{\begin{array}{c} b : C \\ (a, b) : R \end{array}} \ (\exists) \qquad \frac{\begin{array}{c} a : \neg(\exists R.C) \\ (a, b) : R \end{array}}{b : \neg C} \ (\neg\exists)$$

for b a new individual

For Universal:

$$\frac{\begin{array}{c} a : \forall R.C \\ (a, b) : R \end{array}}{b : C} \ (\forall) \qquad \frac{a : \neg(\forall R.C)}{\begin{array}{c} b : \neg C \\ (a, b) : R \end{array}} \ (\neg\forall)$$

for b a new individual

Reasoning with Description Logics

Reasoning with Description Logics

- We defined before what it meant for a formula of DL to be sat/unsat (using the translation to FOL).

Reasoning with Description Logics

- We defined before what it meant for a formula of DL to be sat/unsat (using the translation to FOL).
  The tablaux rules that we saw up to know can be used to prove that an assertion is sat/unsat.

Reasoning with Description Logics

- We defined before what it meant for a formula of DL to be sat/unsat (using the translation to FOL).
  The tablaux rules that we saw up to know can be used to prove that an assertion is sat/unsat.
- There are different Reasoning Tasks that we might be interested in, when using DL.

Reasoning with Description Logics

- We defined before what it meant for a formula of DL to be sat/unsat (using the translation to FOL).
  The tablaux rules that we saw up to know can be used to prove that an assertion is sat/unsat.
- There are different Reasoning Tasks that we might be interested in, when using DL.
  - Concept Inconsistency: Given a concept C, is C always empty in every model?

Reasoning with Description Logics

- We defined before what it meant for a formula of DL to be sat/unsat (using the translation to FOL).
  The tablaux rules that we saw up to know can be used to prove that an assertion is sat/unsat.
- There are different Reasoning Tasks that we might be interested in, when using DL.
  - Concept Inconsistency: Given a concept C, is C always empty in every model?
    Equivalently, can we find a model where C is not empty?

Reasoning with Description Logics

- We defined before what it meant for a formula of DL to be sat/unsat (using the translation to FOL).
  The tablaux rules that we saw up to know can be used to prove that an assertion is sat/unsat.
- There are different Reasoning Tasks that we might be interested in, when using DL.
  - Concept Inconsistency: Given a concept C, is C always empty in every model?
    Equivalently, can we find a model where C is not empty?
  - Concept Membership: Given some definitions T, some assertions A, a concept C and an individual a, does the information in $\langle T, A \rangle$ makes a a C?

Reasoning with Description Logics

- We defined before what it meant for a formula of DL to be sat/unsat (using the translation to FOL).
  The tablaux rules that we saw up to know can be used to prove that an assertion is sat/unsat.
- There are different Reasoning Tasks that we might be interested in, when using DL.
  - Concept Inconsistency: Given a concept C, is C always empty in every model?
    Equivalently, can we find a model where C is not empty?
  - Concept Membership: Given some definitions T, some assertions A, a concept C and an individual a, does the information in $\langle T, A \rangle$ makes a a C?
    Equivalently, does every model where $\langle T, A \rangle$ is true, also makes $a : C$ true?

Reasoning with Description Logics

- We defined before what it meant for a formula of DL to be sat/unsat (using the translation to FOL).
  The tablaux rules that we saw up to know can be used to prove that an assertion is sat/unsat.
- There are different Reasoning Tasks that we might be interested in, when using DL.
  - Concept Inconsistency: Given a concept C, is C always empty in every model?
    Equivalently, can we find a model where C is not empty?
  - Concept Membership: Given some definitions T, some assertions A, a concept C and an individual a, does the information in $\langle T, A \rangle$ makes a a C?
    Equivalently, does every model where $\langle T, A \rangle$ is true, also makes $a : C$ true?
  - Concept Equivalence: Given some definitions T, some assertions A, and two concepts C_1 and C_2, does the information in $\langle T, A \rangle$ makes the concept $C_1 and C_2$ equivalent?

Reasoning with Description Logics

- We defined before what it meant for a formula of DL to be sat/unsat (using the translation to FOL).
  The tablaux rules that we saw up to know can be used to prove that an assertion is sat/unsat.
- There are different Reasoning Tasks that we might be interested in, when using DL.
  - Concept Inconsistency: Given a concept C, is C always empty in every model?
    Equivalently, can we find a model where C is not empty?
  - Concept Membership: Given some definitions T, some assertions A, a concept C and an individual a, does the information in $\langle T, A \rangle$ makes a a C?
    Equivalently, does every model where $\langle T, A \rangle$ is true, also makes $a : C$ true?
  - Concept Equivalence: Given some definitions T, some assertions A, and two concepts C_1 and C_2, does the information in $\langle T, A \rangle$ makes the concept $C_1 and C_2$ equivalent?
    Equivalently, does every model where $\langle T, A \rangle$ is true, also makes $C_1 \equiv C_2$ true?

Tableaux Rules

Tableaux Rules

For a set T of Definitions

$$\frac{C_1 \sqsubseteq C_2 \in T}{a : \neg C_1 \vee C_2} \ (\sqsubseteq) \qquad \frac{C_1 \equiv C_2 \in T}{\begin{array}{c} a : \neg C_2 \vee C_1 \\ a : \neg C_1 \vee C_2 \end{array}} \ (\equiv)$$

for a any individual in the tableaux

Tableaux Rules

For a set T of Definitions

$$\frac{C_1 \sqsubseteq C_2 \in T}{a : \neg C_1 \vee C_2}\ (\sqsubseteq) \qquad \frac{C_1 \equiv C_2 \in T}{\begin{array}{c} a : \neg C_2 \vee C_1 \\ a : \neg C_1 \vee C_2 \end{array}}\ (\equiv)$$

for a any individual in the tableaux

For a set A of Assertions

$$\frac{a : C \in A}{a : C}\ (a :) \qquad \frac{(a, b) : R \in A}{(a, b) : R}\ ((a, b) :)$$

Using the Tableaux Rules

Using the Tableaux Rules

- Concept Inconsistency: Given a concept C, is C always empty in every model?

Using the Tableaux Rules

- Concept Inconsistency: Given a concept C, is C always empty in every model?
  Run the tableaux rules on $a : C$ for an arbitrary a. If all the branches are closed, then C is always empty in every model.

Using the Tableaux Rules

- Concept Inconsistency: Given a concept C, is C always empty in every model?
  Run the tableaux rules on $a : C$ for an arbitrary a. If all the branches are closed, then C is always empty in every model.
- We prove that $C \wedge \neg(D \vee C)$ is inconsistent.

$$a : C \wedge \neg(D \vee C)$$
$$a : C$$
$$a : \neg(D \vee C)$$
$$a : \neg D$$
$$a : \neg C$$
$$\otimes$$

Using the Tableaux Rules

Using the Tableaux Rules

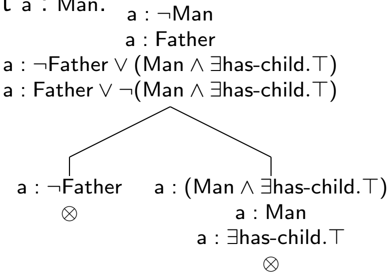
- Concept Membership: Given some definitions T, some assertions A, a concept C and an individual a, does the information in $\langle T, A \rangle$ makes a a C?

Using the Tableaux Rules

- Concept Membership: Given some definitions T, some assertions A, a concept C and an individual a, does the information in $\langle T, A \rangle$ makes a a C?
  Run the tableaux rules on $a : \neg C$. If all the branches are closed, then in every model $a : C$.

Using the Tableaux Rules

- Concept Membership: Given some definitions T, some assertions A, a concept C and an individual a, does the information in $\langle T, A \rangle$ makes a a C?
  Run the tableaux rules on $a : \neg C$. If all the branches are closed, then in every model $a : C$.
- We prove that given $T = \{\text{Father} \equiv \text{Man} \wedge \exists\text{has-child}.\top\}$ and $A = \{\text{a} : \text{Father}\}$ it follows that a : Man.



Using the Tableaux Rules

Using the Tableaux Rules

- Concept Equivalence: Given some definitions T, some assertions A, and two concepts C_1 and C_2, does the information in $\langle T, A \rangle$ makes the concept $C_1 and C_2$ equivalent?

Using the Tableaux Rules

- Concept Equivalence: Given some definitions T, some assertions A, and two concepts C_1 and C_2, does the information in $\langle T, A \rangle$ makes the concept $C_1 and C_2$ equivalent?
  Run the tableaux rules on $a : C_1 \wedge \neg C_2$. If all the branches are closed, then in every model $C_1 \sqsubseteq C_2$.
  Do the same for $a : C_1 \wedge \neg C_2$.

Exercises

Prove that, with respect to the following definitions,

$$
\begin{aligned}
\text{Man} &\equiv \text{Male} \wedge \text{Human} \\
\text{Parent} &\equiv \exists \text{children}.\top \\
\text{Father} &\equiv \text{Man} \wedge \text{Parent} \\
\text{Father-with-only-male-children} &\equiv \text{Father} \wedge \text{Human} \wedge (\forall \text{children}.\text{Male}) \\
\text{Father-with-only-sons} &\equiv \text{Man} \wedge (\exists \text{children}.\top) \wedge (\forall \text{children}.\text{Man})
\end{aligned}
$$

the concept Father-with-only-sons and Father-with-only-male-children are **not** equivalent.