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

- Propositional Logics (Again)
  - Syntax and Semantics
  - Satisfiability, Validity, Contingency, etc.
  - The Tableau Method
  - Decision Methods
  - Exercises

Propositional Logic

Propositional Logic

- The language of propositional logic (PL) is very easy:

Some propositional symbols:	$p_1, p_2, p_3, \ldots$
Two logical symbols:	$\neg, \vee$
Two syntactic symbols:	(,)

Propositional Logic

- The language of propositional logic (PL) is very easy:

Some propositional symbols:	$p_1, p_2, p_3, \ldots$
Two logical symbols:	$\neg, \vee$
Two syntactic symbols:	(,)

- What is a formula?
  - Every propositional symbol p_i is a formula in PL
  - If φ is a formula in PL then $\neg(\varphi)$ is a formula in PL
  - If φ and ψ are formulas in PL then $(\varphi \vee \psi)$ is a formula in PL
  - Nothing esle is a formula

Propositional Logic

Propositional Logic

- Given an assignment V of true or false to the propositional symbols we can determine for any formula whether it is true or false under V.

Propositional Logic

- Given an assignment V of true or false to the propositional symbols we can determine for any formula whether it is true or false under V.
- Let's write $V \models \varphi$ for V makes φ true.

Propositional Logic

- Given an assignment V of true or false to the propositional symbols we can determine for any formula whether it is true or false under V.
- Let's write $V \models \varphi$ for V makes φ true.
- If $V \models \varphi$ we say that V is a model of φ.
- The symbol $\models$ links models (V) and formulas (φ), and says that the formula φ is true in the model V.

Propositional Logic

- Given an assignment V of true or false to the propositional symbols we can determine for any formula whether it is true or false under V.
- Let's write $V \models \varphi$ for V makes φ true.
- If $V \models \varphi$ we say that V is a model of φ.
- The symbol $\models$ links models (V) and formulas (φ), and says that the formula φ is true in the model V.

$$\begin{array}{rcl} V \models \neg(\phi) & \text{iff} & V \not\models \phi \\ V \models (\phi \vee \psi) & \text{iff} & \text{either } V \models \phi \text{ or } V \models \psi \end{array}$$

Satisfiable/Unsatisfiable and Tautology/Contingeny/Contradiction

Satisfiable/Unsatisfiable and Tautology/Contingeny/Contradiction

- We say that a PL formula φ is satisfiable iff for some V, $V \models \varphi$.
- We say that a PL formula φ is unsatisfiable iff it is not satisfiable.

Satisfiable/Unsatisfiable and Tautology/Contingeny/Contradiction

- We say that a PL formula φ is satisfiable iff for some V, $V \models \varphi$.
- We say that a PL formula φ is unsatisfiable iff it is not satisfiable.
- We say that a PL formula φ is a tautology iff for all V, $V \models \varphi$.
- We say that a PL formula φ is a contradiction iff for no V, $V \models \varphi$.
- We say that a PL formula φ is a contingency iff it is neither a tautology nor a contradiction.

Relations between these notions

Relations between these notions

- If a formula is a tautology, is it satisfiable?

Relations between these notions

- If a formula is a tautology, is it satisfiable?
  YES.

Relations between these notions

- If a formula is a tautology, is it satisfiable?
  YES.
- If a formula is a contingency, is it satisfiable?

Relations between these notions

- If a formula is a tautology, is it satisfiable?
  YES.
- If a formula is a contingency, is it satisfiable?
  YES.

Relations between these notions

- If a formula is a tautology, is it satisfiable?
  YES.
- If a formula is a contingency, is it satisfiable?
  YES.
- If a formula is a contingency, is it unsatisfiable?

Relations between these notions

- If a formula is a tautology, is it satisfiable?
  YES.
- If a formula is a contingency, is it satisfiable?
  YES.
- If a formula is a contingency, is it unsatisfiable?
  NO

Relations between these notions

- If a formula is a tautology, is it satisfiable?
  YES.
- If a formula is a contingency, is it satisfiable?
  YES.
- If a formula is a contingency, is it unsatisfiable?
  NO
- If a formula is satisfiable, can it be a tautology?

Relations between these notions

- If a formula is a tautology, is it satisfiable?
  YES.
- If a formula is a contingency, is it satisfiable?
  YES.
- If a formula is a contingency, is it unsatisfiable?
  NO
- If a formula is satisfiable, can it be a tautology?
  YES

Relations between these notions

- If a formula is a tautology, is it satisfiable?
  YES.
- If a formula is a contingency, is it satisfiable?
  YES.
- If a formula is a contingency, is it unsatisfiable?
  NO
- If a formula is satisfiable, can it be a tautology?
  YES
- If a formula is satisfiable, can it be a contingency?

Relations between these notions

- If a formula is a tautology, is it satisfiable?
  YES.
- If a formula is a contingency, is it satisfiable?
  YES.
- If a formula is a contingency, is it unsatisfiable?
  NO
- If a formula is satisfiable, can it be a tautology?
  YES
- If a formula is satisfiable, can it be a contingency?
  YES

Relations between these notions

- If a formula is a tautology, is it satisfiable?
  YES.
- If a formula is a contingency, is it satisfiable?
  YES.
- If a formula is a contingency, is it unsatisfiable?
  NO
- If a formula is satisfiable, can it be a tautology?
  YES
- If a formula is satisfiable, can it be a contingency?
  YES
- If a formula is satisfiable, can it be a contradiction?

Relations between these notions

- If a formula is a tautology, is it satisfiable?
  YES.
- If a formula is a contingency, is it satisfiable?
  YES.
- If a formula is a contingency, is it unsatisfiable?
  NO
- If a formula is satisfiable, can it be a tautology?
  YES
- If a formula is satisfiable, can it be a contingency?
  YES
- If a formula is satisfiable, can it be a contradiction?
  NO

So let's turn to satisfiability checking ...

- We'll use tableaux to perform this task.
- A tableaux is essentially a tree-like data structure that records attempts to build a model.
- Tableaux are built by applying rules to an input formula. These rules systematically tear the formula to detect all possible ways of building a model.
- Each branch of a tableaux records one way of trying to build a model. Some branches ("closed branches") don't lead to models. Others branching ("open branches") do.
- The best way to learn is via an example...

Tableaux for PL

Tableaux for PL

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Tableaux for PL

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

Tableaux for PL

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}} \ (\wedge)$$

Tableaux for PL

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}} \; (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad\qquad \neg\psi} \; (\neg\wedge)$$

Tableaux for PL

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}} \ (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad\qquad \neg\psi} \ (\neg\wedge)$$

$$\frac{\neg\neg\varphi}{\varphi} \ (\neg\neg)$$

Tableaux for PL

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}} \; (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad\qquad \neg\psi} \; (\neg\wedge)$$

$$\frac{\neg\neg\varphi}{\varphi} \; (\neg\neg)$$

$$(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$$

Tableaux for PL

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}} \ (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad\qquad \neg\psi} \ (\neg\wedge)$$

$$\frac{\neg\neg\varphi}{\varphi} \ (\neg\neg)$$

$$(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$$
$$\neg(p \wedge q) \wedge \neg\neg r$$
$$p$$

Tableaux for PL

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}} \; (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad\qquad \neg\psi} \; (\neg\wedge)$$

$$\frac{\neg\neg\varphi}{\varphi} \; (\neg\neg)$$

$$(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$$
$$\neg(p \wedge q) \wedge \neg\neg r$$
$$p$$
$$\neg(p \wedge q)$$
$$\neg\neg r$$

Tableaux for PL

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}} \ (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad \neg\psi} \ (\neg\wedge)$$

$$\frac{\neg\neg\varphi}{\varphi} \ (\neg\neg)$$

$$\begin{array}{c}
(\neg(p \wedge q) \wedge \neg\neg r) \wedge p \\
\neg(p \wedge q) \wedge \neg\neg r \\
p \\
\neg(p \wedge q) \\
\neg\neg r \\
r
\end{array}$$

Tableaux for PL

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}} \; (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad\qquad \neg\psi} \; (\neg\wedge)$$

$$\frac{\neg\neg\varphi}{\varphi} \; (\neg\neg)$$

$$(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$$

$$\neg(p \wedge q) \wedge \neg\neg r$$

$$p$$

$$\neg(p \wedge q)$$

$$\neg\neg r$$

$$r$$

$$\neg p \qquad\qquad\qquad\qquad \neg q$$

Tableaux for PL

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi\\ \psi\end{array}} \; (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad \neg\psi} \; (\neg\wedge)$$

$$\frac{\neg\neg\varphi}{\varphi} \; (\neg\neg)$$

$$(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$$

$$\neg(p \wedge q) \wedge \neg\neg r$$

$$p$$

$$\neg(p \wedge q)$$

$$\neg\neg r$$

$$r$$

$$\neg p \qquad\qquad \neg q$$

Contradiction!!!

Tableaux for PL

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi\\ \psi\end{array}}\ (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad\qquad \neg\psi}\ (\neg\wedge)$$

$$\frac{\neg\neg\varphi}{\varphi}\ (\neg\neg)$$

$$(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$$

$$\neg(p \wedge q) \wedge \neg\neg r$$

$$p$$

$$\neg(p \wedge q)$$

$$\neg\neg r$$

$$r$$

$\neg p$	$\neg q$
Contradiction!!!	*Model*

Formalizing the Diplomatic Problem

Formalizing the Diplomatic Problem

- Three propositional symbols

$$\begin{array}{lcllcl} P & \equiv & \text{invite Peru} & \neg P & \equiv & \text{exclude Peru} \\ Q & \equiv & \text{invite Qatar} & \neg Q & \equiv & \text{exclude Qatar} \\ R & \equiv & \text{invite Romania} & \neg R & \equiv & \text{exclude Romania} \end{array}$$

Formalizing the Diplomatic Problem

- Three propositional symbols

P	$\equiv$	invite Peru	$\neg P$	$\equiv$	exclude Peru
Q	$\equiv$	invite Qatar	$\neg Q$	$\equiv$	exclude Qatar
R	$\equiv$	invite Romania	$\neg R$	$\equiv$	exclude Romania

- The problem can be formalized as

prince: $P \vee \neg Q$	$\equiv$	invite Peru or exclude Qatar
queen: $Q \vee R$	$\equiv$	invite Qatar or Romania or both
king: $\neg R \vee \neg P$	$\equiv$	snub Romania or Peru or both

Formalizing the Diplomatic Problem

- Three propositional symbols

P	$\equiv$	invite Peru	$\neg P$	$\equiv$	exclude Peru
Q	$\equiv$	invite Qatar	$\neg Q$	$\equiv$	exclude Qatar
R	$\equiv$	invite Romania	$\neg R$	$\equiv$	exclude Romania

- The problem can be formalized as

prince: $P \vee \neg Q$	$\equiv$	invite Peru or exclude Qatar
queen: $Q \vee R$	$\equiv$	invite Qatar or Romania or both
king: $\neg R \vee \neg P$	$\equiv$	snub Romania or Peru or both

- $\Sigma = (P \vee \neg Q) \; \& \; (Q \vee R) \; \& \; (\neg R \vee \neg P)$

Validity via Tableaux

Validity via Tableaux

Let's show that $(p \wedge q) \rightarrow p$ is valid

Validity via Tableaux

Let's show that $(p \wedge q) \rightarrow p$ is valid

Rules for $\rightarrow$

Validity via Tableaux

Let's show that $(p \wedge q) \rightarrow p$ is valid

Rules for $\rightarrow$

$$\frac{\varphi \rightarrow \psi}{\neg\varphi \qquad \psi}\ (\rightarrow)$$

Validity via Tableaux

Let's show that $(p \wedge q) \rightarrow p$ is valid

Rules for $\rightarrow$

$$\frac{\varphi \rightarrow \psi}{\neg\varphi \qquad\qquad \psi} \; (\rightarrow)$$

$$\frac{\neg(\varphi \rightarrow \psi)}{\begin{array}{c}\varphi \\ \neg\psi\end{array}} \; (\neg \rightarrow)$$

Validity via Tableaux

Let's show that $(p \wedge q) \rightarrow p$ is valid

$$\neg((p \wedge q) \rightarrow p)$$

Rules for $\rightarrow$

$$\frac{\varphi \rightarrow \psi}{\neg\varphi \qquad\qquad \psi}\ (\rightarrow)$$

$$\frac{\neg(\varphi \rightarrow \psi)}{\begin{array}{c}\varphi \\ \neg\psi\end{array}}\ (\neg \rightarrow)$$

Validity via Tableaux

Let's show that $(p \wedge q) \rightarrow p$ is valid

$$\neg((p \wedge q) \rightarrow p)$$
$$p \wedge q$$
$$\neg p$$

Rules for $\rightarrow$

$$\frac{\varphi \rightarrow \psi}{\neg\varphi \qquad \psi}\ (\rightarrow)$$

$$\frac{\neg(\varphi \rightarrow \psi)}{\begin{array}{c}\varphi \\ \neg\psi\end{array}}\ (\neg \rightarrow)$$

Validity via Tableaux

Let's show that $(p \wedge q) \rightarrow p$ is valid

$$\neg((p \wedge q) \rightarrow p)$$
$$p \wedge q$$
$$\neg p$$
$$p$$
$$q$$

Rules for $\rightarrow$

$$\frac{\varphi \rightarrow \psi}{\neg\varphi \qquad \psi}\ (\rightarrow)$$

$$\frac{\neg(\varphi \rightarrow \psi)}{\begin{array}{c}\varphi \\ \neg\psi\end{array}}\ (\neg \rightarrow)$$

Validity via Tableaux

Let's show that $(p \wedge q) \rightarrow p$ is valid

Rules for $\rightarrow$

$$\frac{\varphi \rightarrow \psi}{\neg\varphi \qquad \psi}\ (\rightarrow)$$

$$\frac{\neg(\varphi \rightarrow \psi)}{\begin{matrix}\varphi \\ \neg\psi\end{matrix}}\ (\neg \rightarrow)$$

$$\neg((p \wedge q) \rightarrow p)$$

$$p \wedge q$$

$$\neg p$$

$$p$$

$$q$$

Contradiction!!!

Validity via Tableaux

Let's show that $(p \wedge q) \rightarrow p$ is valid

Rules for $\rightarrow$

$$\frac{\varphi \rightarrow \psi}{\neg\varphi \qquad\qquad \psi}\ (\rightarrow)$$

$$\frac{\neg(\varphi \rightarrow \psi)}{\begin{array}{c}\varphi \\ \neg\psi\end{array}}\ (\neg \rightarrow)$$

$$\neg((p \wedge q) \rightarrow p)$$

$$p \wedge q$$

$$\neg p$$

$$p$$

$$q$$

Contradiction!!!

It is impossible to apply any more rules, and there are no open branches. Hence no model exists for the input $\neg\varphi$. Hence φ is valid.

Some of the Techniques for SAT Solving

Some of the Techniques for SAT Solving

- Complete methods (they answer SAT if and only if the formula is satisfiable).
  - truth tables
  - tableaux
  - Davis-Putman
  - resolution
  - map into linear equations

Some of the Techniques for SAT Solving

- Complete methods (they answer SAT if and only if the formula is satisfiable).
  - truth tables
  - tableaux
  - Davis-Putman
  - resolution
  - map into linear equations
- Approximation procedures (they answer SAT or UNKOWN)
  - flip the value of a variable in an unsatisfied clause
  - genetic algorithms
  - hill-climbing

Decision Methods for PL

Decision Methods for PL

- The most traditional methods for solving the SAT problem for propositional logics (PL-SAT) behave as follows. For any input formula φ

Decision Methods for PL

- The most traditional methods for solving the SAT problem for propositional logics (PL-SAT) behave as follows. For any input formula φ
  - They always answer SATISFIABLE or UNSATISFIABLE.

Decision Methods for PL

- The most traditional methods for solving the SAT problem for propositional logics (PL-SAT) behave as follows. For any input formula φ
  - They always answer SATISFIABLE or UNSATISFIABLE.
  - After a finite time.

Decision Methods for PL

- The most traditional methods for solving the SAT problem for propositional logics (PL-SAT) behave as follows. For any input formula φ
  - They always answer SATISFIABLE or UNSATISFIABLE.
  - After a finite time.
  - They always answer correctly.

Decision Methods for PL

- The most traditional methods for solving the SAT problem for propositional logics (PL-SAT) behave as follows. For any input formula φ
  - They always answer SATISFIABLE or UNSATISFIABLE.
  - After a finite time.
  - They always answer correctly.
- The best known decision methods probably are
  - truth tables
  - tableaux
  - Davis-Putnam
  - resolution

Graph Coloring: The Nitty-Gritty Details

Graph Coloring: The Nitty-Gritty Details

- We will use $n \times k$ propositional symbols that we write p_{ij} (n is the number of nodes in N, k the number of colors)

Graph Coloring: The Nitty-Gritty Details

- We will use $n \times k$ propositional symbols that we write p_{ij} (n is the number of nodes in N, k the number of colors)
- We will read p_{ij} as node i has color j

Graph Coloring: The Nitty-Gritty Details

- We will use $n \times k$ propositional symbols that we write p_{ij} (n is the number of nodes in N, k the number of colors)
- We will read p_{ij} as node i has color j
- We have to say that
  1. Each node has (at least) one color.
  2. Each node has no more than one color.
  3. Related nodes have different colors.

Graph Coloring: The Nitty-Gritty Details

- We will use $n \times k$ propositional symbols that we write p_{ij} (n is the number of nodes in N, k the number of colors)
- We will read p_{ij} as node i has color j
- We have to say that
  1. Each node has (at least) one color.
  2. Each node has no more than one color.
  3. Related nodes have different colors.

1. Each node has one color: $p_{i1} \vee \ldots \vee p_{ik}$, for $1 \leq i \leq n$

Graph Coloring: The Nitty-Gritty Details

- We will use $n \times k$ propositional symbols that we write p_{ij} (n is the number of nodes in N, k the number of colors)
- We will read p_{ij} as node i has color j
- We have to say that
  1. Each node has (at least) one color.
  2. Each node has no more than one color.
  3. Related nodes have different colors.

1. Each node has one color: $p_{i1} \vee \ldots \vee p_{ik}$,
   for $1 \leq i \leq n$
2. Each node has no more than one color: $\neg p_{il} \vee \neg p_{im}$,
   for $1 \leq i \leq n$, and $1 \leq l < m \leq k$

Graph Coloring: The Nitty-Gritty Details

- We will use $n \times k$ propositional symbols that we write p_{ij} (n is the number of nodes in N, k the number of colors)
- We will read p_{ij} as node i has color j
- We have to say that
  1. Each node has (at least) one color.
  2. Each node has no more than one color.
  3. Related nodes have different colors.

1. Each node has one color: $p_{i1} \vee \ldots \vee p_{ik}$,
   for $1 \leq i \leq n$
2. Each node has no more than one color: $\neg p_{il} \vee \neg p_{im}$,
   for $1 \leq i \leq n$, and $1 \leq l < m \leq k$
3. Neighboring nodes have different colors. $\neg p_{il} \vee \neg p_{jl}$,
   for i and j neighboring nodes, and $1 \leq l \leq k$

Applications: Graph Coloring 3

Applications: Graph Coloring 3

- Results:
  - the GSAT and WalkSAT algorithms seem to be competitive with specialized graph coloring algorithms

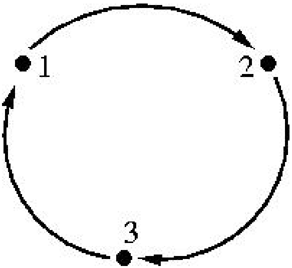
Applications: Graph Coloring 3

- Results:
  - the GSAT and WalkSAT algorithms seem to be competitive with specialized graph coloring algorithms
- An application to algebra:
  - problems dealing with quasigroups can be viewed as specialized coloring problems
  - some open problems on quasigroups have been encoded in this way and solved automatically by propositional provers. E.g. is there a quasigroup satisfying the equations

$$\forall a.(a \cdot a) = a$$
$$\forall a, b.((b \cdot a) \cdot b) \cdot b = a$$

Exercises

Use the graph coloring coding in the following graph, using 2 colors (R and B):



Show the first three lines of a truth table for the formula obtained in the codification.