Semantics and Pragmatics of NLP
Propositional Tablaux

Alex Lascarides & Ewan Klein

School of Informatics
University of Edinburgh

28 January 2008

1 Drawing Inferences

2 Propositional Tableaux

3 Summary

Taking Stock

We have:

- Introduced syntax and semantics for FOL plus lambdas.
- Represented FOL formulae and their models in NLTK.
- Shown how to build LFs in a feature-based grammar.

We've tackled constructing logical form
What about interpreting it?

Approach

- How can we automate the process of drawing inferences from LFs?
- Start with quantifier-free fragment of FOL, i.e., propositional logic.
- Tableaux method.

Propositional Logic

- FOL inference is undecidable and practical techniques are complex.
- Only scratch the surface
- So, we'll examine inferences involving $\neg$, $\rightarrow$, $\wedge$, $\vee$. This is propositional logic.
- Instead of writing:
  (((boxer vincent) $\wedge$ (happy mia))$\vee$
  (($\neg$(boxer vincent)) $\wedge$ (happy marsellus)))
  we write: $(p \wedge q) \vee (\neg p \wedge r)$
- Internal structure of atomic FOL formulae isn't important in propositional logic.

Which Inference Tool?

Theorem Provers:

Input: formula

Output: formula is valid or formula is not valid.

Model Builders:

Input: formula

Output: a (usually finite) model that satisfies the formula, or no model if formula is inconsistent.

E.g., Prover9 + Mace4

The Tableaux Method

- Syntactic, but based on clear semantic intuitions.
  - Instructions on what you can write down next, given what you've written down so far.
  - Instructions preserve truth
    and they tend to break down complex formulae into simpler ones.
- Finding a tableaux proof does not depend on human insight.
- Tableaux systems can in fact be regarded as model building tools.

The Basic Idea

Proof by Refutation:

1. To test whether ϕ is valid (written $\models \phi$);
   - Assume it's false; and
   - attempt to generate a contradiction, by using the instructions on what you can write next.
   - If you can't find a contradiction, then you've constructed a model for $\neg\phi$.
   - So $\neg\phi$ is consistent.
   - So ϕ is not valid, since it's negation is true in at least one model.
2. Method: break down ϕ into simpler statements, and look for combination of:
   - *p* is true
   - *p* is false

   for some atomic sentence *p*.

From Validity to Entailment

To test entailment: $\phi_1, \ldots, \phi_n \models \psi$

- Use tableau method to test whether there is some M such that $M \models \neg((\phi_1 \wedge \ldots \wedge \phi_n) \rightarrow \psi)$.
- I.e, whether $\models (\phi_1 \wedge \ldots \wedge \phi_n) \rightarrow \psi$.
- This is OK because propositional logic has a Deduction Theorem:

$$\phi \models \psi \text{ iff } \models \phi \rightarrow \psi$$

  This doesn't hold of all logics.

Example: $p \vee \neg p$

$$F(p \vee \neg p)$$

- This is our first tableau!
- F means we want to falsify $p \vee \neg p$
- Line numbers useful for book-keeping.

$$1.\ F(p \vee \neg p)$$

Continuing with this Example

1.	$F(p \vee \neg p)$	$\surd$
2.	Fp	$1, F_{\vee}$
3.	$F\neg p$	$1, F_{\vee}$

- Our second tableaux!
- Uses the tableaux expansion rule called $F_{\vee}$ (falsify a disjunction) to break down the disjunction in line 1. into pieces.
- $\surd$ shows you have applied the appropriate rule to this line.
- Never need to apply a rule to the same line twice, which is nice.

And Carrying On

1.	$F(p \vee \neg p)$	$\surd$
2.	Fp	$1, F_{\vee}$
3.	$F\neg p$	$1,\ F_{\vee},\ \surd$
4.	Tp	$3, F_{\neg}$.

- $F_{\neg}$: falsify a negation.
- We're finished!
  - The tableau is rule saturated. You can't apply any more rules.
- Tableau is also closed.
  - Conflict in lines 2. and 4.
- So we have proved that $p \vee \neg p$ is valid!

Another Example

$$1. \quad F\neg(q \wedge r) \rightarrow (\neg q \vee \neg r)$$

$F_{\rightarrow}$ tells us how to falsify an implication:

$$\begin{array}{lll} 1. & F\neg(q \wedge r) \rightarrow (\neg q \vee \neg r) & \surd \\ 2. & T\neg(q \wedge r) & 1, F_{\rightarrow} \\ 3. & F(\neg q \vee \neg r) & 2, F_{\rightarrow} \end{array}$$

Line 3. calls for $F_{\vee}$ (falsify a disjunction)
Can do it now! Don't have to do line 2. first...

Example Continued

1.	$F\neg(q \wedge r) \rightarrow (\neg q \vee \neg r)$	$\surd$
2.	$T\neg(q \wedge r)$	$1, F_{\rightarrow}$
3.	$F(\neg q \vee \neg r)$	$2, F_{\rightarrow}, \surd$
4.	$F\neg q$	$3, F_{\vee}, \surd$
5.	$F\neg r$	$3, F_{\vee}, \surd$
6.	Tq	$4, F_{\neg}$
7.	Tr	$5, F_{\neg}$

Now deal with line 2

Example Continued

1.	$F\neg(q \wedge r) \rightarrow (\neg q \vee \neg r)$	$\surd$
2.	$T\neg(q \wedge r)$	1, $F_{\rightarrow}$, $\surd$
3.	$F(\neg q \vee \neg r)$	2, $F_{\rightarrow}$, $\surd$
4.	$F\neg q$	3, $F_{\vee}$, $\surd$
5.	$F\neg r$	3, $F_{\vee}$, $\surd$
6.	Tq	4, $F_{\neg}$
7.	Tr	5, $F_{\neg}$
8.	$F(q \wedge r)$	2, $T_{\neg}$

But there are two ways of falsifying $q \wedge r$:

- q is false or r is false.

Example Continued

1.	$F\neg(q \wedge r) \rightarrow (\neg q \vee \neg r)$	$\surd$
2.	$T\neg(q \wedge r)$	1, $F_{\rightarrow}$, $\surd$
3.	$F(\neg q \vee \neg r)$	2, $F_{\rightarrow}$, $\surd$
4.	$F\neg q$	3, $F_{\vee}$, $\surd$
5.	$F\neg r$	3, $F_{\vee}$, $\surd$
6.	Tq	4, $F_{\neg}$
7.	Tr	5, $F_{\neg}$
8.	$F(q \wedge r)$	2, $T_{\neg}$

9. Fq 8, $F_{\wedge}$	Fr 8, $F_{\wedge}$

- Finished! Tableau is rule saturated.
- $\neg(q \wedge r) \rightarrow (\neg q \wedge \neg r)$ is valid! Tableau is closed.

Tableau as a Model Builder

$$
\begin{array}{lll}
1. & F(p \wedge q) \rightarrow (r \vee s) & \surd \\
2. & T(p \wedge q) & 1, F_{\rightarrow}, \surd \\
3. & F(r \vee s) & 1, F_{\rightarrow}, \surd \\
4. & Tp & 2, T_{\wedge} \\
5. & Tq & 2, T_{\wedge} \\
6. & Fr & 3, F_{\vee} \\
7. & Fs & 3, F_{\vee}
\end{array}
$$

- Tableau is rule saturated but not closed.
- So $(p \wedge q) \rightarrow (r \vee s)$ is not valid.
- In fact, tableau tells us how to make it false!
  - p is true; q is true; r is false; s is false.

The Instructions

$$T_\wedge: \quad \frac{T(\phi \wedge \psi)}{\begin{array}{c} T\phi \\ T\psi \end{array}} \qquad F_\wedge: \quad \frac{F(\phi \wedge \psi)}{F\phi \mid F\psi}$$

$$T_\neg: \quad \frac{T\neg\phi}{F\phi} \qquad F_\neg: \quad \frac{F\neg\phi}{T\phi}$$

$$F_\vee: \quad \frac{F(\phi \vee \psi)}{\begin{array}{c} F\phi \\ F\psi \end{array}} \qquad T_\vee: \quad \frac{T(\phi \vee \psi)}{T\phi \mid T\psi}$$

$$F_\rightarrow: \quad \frac{F(\phi \rightarrow \psi)}{\begin{array}{c} T\phi \\ F\psi \end{array}} \qquad T_\rightarrow: \quad \frac{T(\phi \rightarrow \psi)}{F\phi \mid T\psi}$$

- Keep applying rules until tableau is rule saturated.

Tableaux are Trees

- A (propositional) tableau is a tree;
  each node is a signed (propositional) formula.
- A branch of a tableau is a branch of the tree.
- Tableaux expansion:
  1. Find a node that:
     1. isn't a signed atomic formula (not Fp or Tp)
     2. hasn't had an expansion rule applied to it
  2. Expand it according to the rules!
  3. Keep going until tree is rule saturated.

Closed and Open Tableaux

- A branch of a tableau is closed if it contains $T\phi$ and $F\phi$.
- A tableau is closed if all its branches are closed.
  It is open if at least one of its branches is open (i.e., not closed).

Provability:

- A formula ϕ is provable (written $\vdash \phi$) iff
  it is possible to expand the initial tableau $F\phi$ to a closed tableau.

Testing Entailment (or Uninformativity)

Does $\phi_1, \ldots, \phi_n \models \psi$?

- Start with:

$$\begin{array}{c} T\phi_1 \\ \vdots \\ T\phi_n \\ F\psi \end{array}$$

- If this expands to a closed tableau, then the argument is valid.
- Or to put it another way:
  ψ is uninformative with respect to $\phi_1, \ldots, \phi_n$

Soundness and Completeness

- The tableaux system is sound:

  $$\text{If} \vdash \phi \text{ then} \models \phi$$

  That is, you can't prove something that's not valid.
- The tableaux system is complete:

  $$\text{If} \models \phi \text{ then} \vdash \phi$$

  That is, every valid formula has a proof.

Conclusion

- You can prove validities and entailments in propositional logic using the tableaux method.
- It's sound and complete (and decidable).
- It's very easy to implement, because:
  - Creating tableaux doesn't require human insight
  - It doesn't matter what choices you make at what time
  - eventually you'll get an answer.
- But propositional logic isn't powerful enough for NL semantics.
  - E.g., doesn't handle quantification
- So more powerful methods required for FOL theorem proving.

Reading: B&B Chapter 4.