ECE570 Lecture 17: Semantic Tableaux

Jeffrey Mark Siskind

School of Electrical and Computer Engineering

Fall 2013

PURDUE UNIVERSITY

Syntax of FOL—I

Given:

- a set X of *variable symbols*,
- a sequence of sets $F_1, F_2, \ldots, F_i, \ldots$ of *function symbols* of arity i, and
- a sequence of sets $P_1, P_2, \ldots, P_i, \ldots$ of *predicate symbols* of arity i.

A *term* is:

- x, where $x \in X$, or
- $f(t_1, \ldots, t_n)$, where $f \in F_n$ and each t_i is a term.

Syntax of FOL—II

An *atomic formula* is:

- $p(t_1, \ldots, t_n)$, where $p \in P_n$ and each t_i is a term.

A *formula* is:

- an atomic formula or
- $\neg\Phi$, $\Phi \wedge \Psi$, $\Phi \vee \Psi$, $\Phi \rightarrow \Psi$, $\Phi \leftrightarrow \Psi$, $\forall x \Phi$, or $\exists x \Phi$, where $x \in X$ and Φ and Ψ are formulas.

Syntax of FOL as a Nondeterministic Scheme Program—I

```
(define (list-of n f)
 (if (= n 0) '() (cons (f) (list-of (- n 1) f))))

(define (a-term)
 (either
  (a-member-of *x*)
  (let ((n ((an-integer-between 0 (length *f*)))))
   (cons (a-member-of (list-ref *f* n))
         (list-of n (lambda () (a-term)))))))
```

Syntax of FOL as a Nondeterministic Scheme Program—II

```
(define (an-atomic-formula)
 (let ((n ((an-integer-between 0 (length *p*)))))
  (cons (a-member-of (list-ref *p* n))
        (list-of n (lambda () (a-term))))))

(define (a-formula)
 (either (an-atomic-formula)
         `(not ,(a-formula))
         `(and ,(a-formula) ,(a-formula))
         `(or ,(a-formula) ,(a-formula))
         `(implies ,(a-formula) ,(a-formula))
         `(iff ,(a-formula) ,(a-formula))
         `(every ,(a-member-of *x*) ,(a-formula))
         `(some ,(a-member-of *x*) ,(a-formula))))
```

Semantics of FOL—I

An *interpretation* (first-order structure) $M = \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P} \rangle$.
U is a set, the 'Universe of Discourse.'
$\mathcal{X}$ maps each $x \in X$ to an element of U.
$\mathcal{F}$ is a sequence of maps $\mathcal{F}_1, \mathcal{F}_2, \ldots$.
$\mathcal{F}_i$ maps each $f \in F_i$ to an element of $\underbrace{U \times \cdots \times U}_{i} \rightarrow U$.

$\mathcal{P}$ is a sequence of maps $\mathcal{P}_1, \mathcal{P}_2, \ldots$.
$\mathcal{P}_i$ maps each $p \in P_i$ to a subset of $\underbrace{U \times \cdots \times U}_{i}$.

Semantics of FOL—II

$\mathcal{X}[x \mapsto u]$ denotes the map that is identical to $\mathcal{X}$ except that it maps x to u.

$$
\begin{array}{rcl}
\mathcal{E}(x, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) & \triangleq & \mathcal{X}(x) \\
\mathcal{E}(f(t_1, \ldots, t_n), \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) & \triangleq & \mathcal{F}_n(f)(\mathcal{E}(t_1, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle), \ldots, \mathcal{E}(t_n, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle)) \\
\mathcal{V}(p(t_1, \ldots, t_n), \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) & \triangleq & \mathcal{P}_n(p)(\mathcal{E}(t_1, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle), \ldots, \mathcal{E}(t_n, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle)) \\
\mathcal{V}(\neg\Phi, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) & \triangleq & \neg\mathcal{V}(\Phi, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) \\
\mathcal{V}(\Phi \wedge \Psi, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) & \triangleq & \mathcal{V}(\Phi, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) \wedge \mathcal{V}(\Psi, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) \\
\mathcal{V}(\Phi \vee \Psi, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) & \triangleq & \mathcal{V}(\Phi, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) \vee \mathcal{V}(\Psi, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) \\
\mathcal{V}(\Phi \rightarrow \Psi, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) & \triangleq & \mathcal{V}(\Phi, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) \rightarrow \mathcal{V}(\Psi, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) \\
\mathcal{V}(\Phi \leftrightarrow \Psi, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) & \triangleq & \mathcal{V}(\Phi, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) \leftrightarrow \mathcal{V}(\Psi, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) \\
\mathcal{V}(\forall x\Phi, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) & \triangleq & (\forall u \in U)\mathcal{V}(\Phi, \langle U, \mathcal{X}[x \mapsto u], \mathcal{F}, \mathcal{P}\rangle) \\
\mathcal{V}(\exists x\Phi, \langle U, \mathcal{X}, \mathcal{F}, \mathcal{P}\rangle) & \triangleq & (\exists u \in U)\mathcal{V}(\Phi, \langle U, \mathcal{X}[x \mapsto u], \mathcal{F}, \mathcal{P}\rangle)
\end{array}
$$

Semantics of FOL in Scheme—I

```
(define (e t u x f p)
 (if (symbol? t)
     (cdr (assq t x))
     (apply (cdr (assq (first t) f))
            (map (lambda (t) (e t u x f p))
                 (rest t)))))

(define (implies p q) (or (not p) q))

(define (iff p q)
 (and (implies p q) (implies q p)))
```

Semantics of FOL in Scheme—II

```
(define (v phi u x f p)
 (case (first phi)
  ((not) (not (v (second phi) u x f p)))
  ((and) (and (v (second phi) u x f p) (v (third phi) u x f p)))
  ((or) (or (v (second phi) u x f p) (v (third phi) u x f p)))
  ((implies)
   (implies (v (second phi) u x f p) (v (third phi) u x f p)))
  ((iff) (iff (v (second phi) u x f p) (v (third phi) u x f p)))
  ((every)
   (every (lambda (w)
           (v (second phi) u (cons (cons (first phi) w) x) f p))
          u))
  ((some)
   (some (lambda (w)
          (v (second phi) u (cons (cons (first phi) w) x) f p))
         u))
  (else (apply (cdr (assq (first phi) p))
               (map (lambda (t) (e t u x f p)) (rest phi))))))
```

Entailment

- $M \models \Phi$ iff $\mathcal{V}(\Phi, M)$ yields **true**.
  M is a model of Φ.
  M models Φ.
- $\models \Phi$ iff Φ is true in all interpretations.
  Φ is (universally) valid.
- $\Sigma \models \Phi$, where Σ is a set of formulas,
  iff any interpretation in which all elements of Σ are true is also a model of Φ.
  Σ entails Φ.
- Φ is satisfiable/consistent iff it has a model.
- Φ is unsatisfiable/inconsistent iff it has no model.

Refutation

$\Sigma \models \Phi$ iff $\Sigma \cup \{\neg\Phi\}$ is inconsistent.

Semantic Tableaux—I

Attempt to prove that $\Sigma \models \Phi$
by trying to find a model for $\Sigma \cup \{\neg\Phi\}$.
If we fail to find a model then $\Sigma \cup \{\neg\Phi\}$ is inconsistent
and thus $\Sigma \models \Phi$.

Γ is a queue of formulas.

- $\textsc{Push}(\Theta, \Gamma)$
- $\Theta := \textsc{Pop}(\Gamma)$
- $\textsc{Empty}(\Gamma)$
- $\textsc{Ground}(\Gamma)$

Semantic Tableaux—III

M is a set of formulas.

- $M := M \cup \{\Theta\}$
- $\Theta \in M$
- $\textsc{Ground}(M)$

Semantic Tableaux—IV

$\Gamma := \Sigma \cup \{\neg\Phi\};$
$M := \{\};$
while $\neg\textsc{Empty}(\Gamma)$
do $\Theta := \textsc{Pop}(\Gamma)$
    case Θ
    in $\neg\Phi : \ldots$
        $\vdots$
            end
    end

Semantic Tableaux—V

$\Phi \wedge \Psi$	Both $\textsc{Push}(\Phi, \Gamma)$ and $\textsc{Push}(\Psi, \Gamma)$
$\Phi \vee \Psi$	Either $\textsc{Push}(\Phi, \Gamma)$ or $\textsc{Push}(\Psi, \Gamma)$
$\Phi \rightarrow \Psi$	$\textsc{Push}(\neg\Phi \vee \Psi, \Gamma)$
$\Phi \leftrightarrow \Psi$	$\textsc{Push}((\Phi \rightarrow \Psi) \wedge (\Psi \rightarrow \Phi), \Gamma)$
$\neg\neg\Phi$	$\textsc{Push}(\Phi, \Gamma)$
$\neg(\Phi \wedge \Psi)$	$\textsc{Push}(\neg\Phi \vee \neg\Psi, \Gamma)$
$\neg(\Phi \vee \Psi)$	$\textsc{Push}(\neg\Phi \wedge \neg\Psi, \Gamma)$
$\neg(\Phi \rightarrow \Psi)$	$\textsc{Push}(\Phi \wedge \neg\Psi, \Gamma)$
$\neg(\Phi \leftrightarrow \Psi)$	$\textsc{Push}(\Phi \leftrightarrow \neg\Psi, \Gamma)$
$\neg\forall x\Phi$	$\textsc{Push}(\exists x\neg\Phi, \Gamma)$
$\neg\exists x\Phi$	$\textsc{Push}(\forall x\neg\Phi, \Gamma)$

Semantic Tableaux—VI

Φ atomic	If $\neg\Phi \in M$ then fail else $M := M \cup \{\Phi\}$.
$\neg\Phi$ atomic	If $\Phi \in M$ then fail else $M := M \cup \{\neg\Phi\}$.
$\exists x\Phi$	$\textsc{Push}(\Phi[x \mapsto \textbf{gensym}], \Gamma)$
$\forall x\Phi$	Let $T = \textsc{Ground}(\Gamma) \cup \textsc{Ground}(M)$. If $T = \{\}$ then $T := \{\textbf{gensym}\}$. For each $t \in T$ do $\textsc{Push}(\Phi[x \mapsto t], \Gamma)$. $\textsc{Push}(\forall x\Phi, \Gamma)$