KE/Tableaux

The term *Tableaux* refers to a family of deduction methods for different logics. We start by introducing one of them:

"non-free-variable KE for classical FOL"

What is it for?

Given: set of *premises* Δ and *conclusion* φ (all FOL sentences)

Task: prove $\Delta \models \varphi$

How? show $\Delta \cup \{\neg\varphi\}$ is not satisfiable (which is equivalent), i.e. add the complement of the conclusion to the premises and derive a contradiction (*"refutation procedure"*)

Constructing KE Proofs

Data structure. A KE proof is represented as a *tableau*: a binary tree whose nodes are labelled with formulas.

Start. We start by writing the premises and the negated conclusion into the root of an otherwise empty tableau.

Expansion. We apply *expansion rules* to the formulas on the tree; this results in new formulas being added and branches being split.

Closure. A branch that is obviously contradictory can be *closed*.

Success. A proof is *successful* iff we can close all branches.

Propositional KE Rules

Alpha Rules

$$\frac{A \land B}{\begin{array}{c}A\\ B\end{array}} \qquad \frac{\neg(A \lor B)}{\begin{array}{c}\neg A\\ \neg B\end{array}} \qquad \frac{\neg(A \to B)}{\begin{array}{c}A\\ \neg B\end{array}}$$

¬¬-Elimination

$$\frac{\neg\neg A}{A}$$

Beta Rules

$$\frac{\begin{array}{c}A \lor B\\ \neg A\end{array}}{B} \qquad \frac{\begin{array}{c}A \lor B\\ \neg B\end{array}}{A} \qquad \frac{\begin{array}{c}A \to B\\ A\end{array}}{B} \qquad \frac{\begin{array}{c}A \to B\\ \neg B\end{array}}{\neg A} \qquad \frac{\begin{array}{c}\neg(A \land B)\\ A\end{array}}{\neg B} \qquad \frac{\begin{array}{c}\neg(A \land B)\\ B\end{array}}{\neg A}$$

Propositional KE Rules (2)

PB

$$\frac{}{A \;\big|\; \neg A}$$

Closing branches

$$\frac{\begin{array}{c}A\\ \neg A\end{array}}{\times}$$

Eta Rules

$$\frac{\begin{array}{c}A \leftrightarrow B\\ A\end{array}}{B} \qquad \frac{\begin{array}{c}A \leftrightarrow B\\ B\end{array}}{A} \qquad \frac{\begin{array}{c}\neg(A \leftrightarrow B)\\ A\end{array}}{\neg B} \qquad \frac{\begin{array}{c}\neg(A \leftrightarrow B)\\ B\end{array}}{\neg A}$$

$$\frac{\begin{array}{c}A \leftrightarrow B\\ \neg A\end{array}}{\neg B} \qquad \frac{\begin{array}{c}A \leftrightarrow B\\ \neg B\end{array}}{\neg A} \qquad \frac{\begin{array}{c}\neg(A \leftrightarrow B)\\ \neg A\end{array}}{B} \qquad \frac{\begin{array}{c}\neg(A \leftrightarrow B)\\ \neg B\end{array}}{A}$$

Quantifier Rules

Gamma Rules

$$\frac{(\forall x)A}{A[x \leftarrow t]} \qquad \frac{\neg(\exists x)A}{\neg A[x \leftarrow t]}$$

Delta Rules

$$\frac{(\exists x)A}{A[x \leftarrow c]} \qquad \frac{\neg(\forall x)A}{\neg A[x \leftarrow c]}$$

- t is an arbitrary ground term
- c is a constant symbol new to the branch

Observations

- **Rule application order.** The order in which rules are applied can change the size of proof trees significantly. Example: applying α before PB is generally a good idea, etc.
- **Beta simplification.** If we have a β-formula (like $A \rightarrow B$) and the complement of a matching minor premise (like $\neg A$) on the same branch, then we don't need to apply the β-rule.
- **Analytic PB.** The choice of PB-formulas can be restricted to subformulas appearing on the branch (more precisely: to subformulas of non-analysed β- and η-formulas).
- **Gamma rule.** The γ-rule is the only rule we may have to apply more than once to the same formula. (This is precisely what makes things so difficult: we *don't know* how often we have to apply it.)

Can a proof ever fail?

General answer: *No!* The algorithm can only return "yes" or "don't know" (but not "no"). It is a so-called *semi-decision procedure* (there can be no decision procedure for FOL).

Special cases. In special cases, however, we may be able to "see" that a proof could never succeed (i.e. we can declare it a failure).

This is, for example, the case when we have enough information to construct a *countermodel* . . .

Saturated Branches

An open branch is called *saturated* iff every (complex) formulas has been analysed at least once and, additionally, every γ-formula has been instantiated with every term we can construct using the function symbols on the branch.

Failing proofs. A tableau with an open saturated branch can never be closed, i.e. we can stop an declare the proof a failure.

The solution? This only helps us in special cases though. (A single 1-ary function symbol together with a constant is already enough to construct infinitely many terms . . .)

Propositional logic. In propositional logic (where we have no γ-formulas), after a limited number of steps, every branch will be either closed or saturated. This gives us a decision procedure.

Countermodels

If a KE proof fails with a *saturated open branch*, you can use it to help you define a model $\mathcal{M}$ for all the formulas on that branch:

- domain: set of all terms we can construct using the function symbols appearing on the branch (so-called *Herbrand universe*)
- terms are interpreted as themselves (*sic!*)
- interpretation of predicate symbols: see literals on branch

In particular, $\mathcal{M}$ will be a model for the premises Δ and the negated conclusion $\neg\varphi$, thereby constituting a *counterexample* for the attempted proof of $\Delta \models \varphi$.

Careful: There's a bug in WinKE: sometimes, what is presented as a countermodel is in fact only *part* of a countermodel (but it can always be extended to an actual model).

Soundness and Completeness

Again, let φ be a sentence and let Δ be a set of sentences. We write $\Delta \vdash_{KE} \varphi$ to say that there exists a closed KE tableau for $\Delta \cup \{\neg\varphi\}$.

Before you can believe in KE you need to prove the following:

Theorem 1 (Soundness) *If* $\Delta \vdash_{KE} \varphi$ *then* $\Delta \models \varphi$.

Theorem 2 (Completeness) *If* $\Delta \models \varphi$ *then* $\Delta \vdash_{KE} \varphi$.

Important note: The mere *existence* of a closed tableau does *not* entail that we have an effective method of finding it! Concretely: we don't know how often we need to apply the γ-rule and what terms to use in the substitutions.

From now on, to simplify things, we shall not consider $\rightarrow$, $\leftrightarrow$, $\top$, and $\bot$ (which can be regarded as abbreviations).

Smullyan's Uniform Notation

Formulas of conjunctive (α) and disjunctive (β) type:

α	α_1	α_2
$A \wedge B$	A	B
$\neg(A \vee B)$	$\neg A$	$\neg B$
$\neg(A \rightarrow B)$	A	$\neg B$

β	β_1	β_2
$A \vee B$	A	B
$A \rightarrow B$	$\neg A$	B
$\neg(A \wedge B)$	$\neg A$	$\neg B$

We can now state alpha and beta rules as follows:

$$\frac{\alpha}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}} \qquad \frac{\begin{array}{c}\beta\\ \beta_1^{\mathsf{C}}\end{array}}{\beta_2} \qquad \frac{\begin{array}{c}\beta\\ \beta_2^{\mathsf{C}}\end{array}}{\beta_1} \qquad \text{where } A^{\mathsf{C}} = \begin{cases} A' & \text{for } A = \neg A', \\ \neg A & \text{otherwise} \end{cases}$$

Smullyan's Uniform Notation (2)

Formulas of universal (γ) and existential (δ) type:

γ	$\gamma_1(u)$
$(\forall x)A$	$A[x \leftarrow u]$
$\neg(\exists x)A$	$\neg A[x \leftarrow u]$

δ	$\delta_1(u)$
$(\exists x)A$	$A[x \leftarrow u]$
$\neg(\forall x)A$	$\neg A[x \leftarrow u]$

We can now state gamma and delta rules as follows:

$$\frac{\gamma}{\gamma_1(t)} \qquad \frac{\delta}{\delta_1(c)}$$

where:

- t is an arbitrary ground term
- c is a constant symbol new to the branch

Soundness Proof

Satisfiable branches. We say that a *branch* is *satisfiable* iff the set of sentences on that branch is satisfiable.

Proof sketch. First prove the following lemma:

Lemma 1 (Satisfiable branches) *If a non-branching KE rule is applied to a satisfiable branch, the result is another satisfiable branch. If PB is applied to a satisfiable branch, at least one of the resulting branches is also satisfiable.*

Now we can prove soundness by contradiction: assume $\Delta \vdash_{KE} \varphi$ but $\Delta \not\models \varphi$ and try to derive a contradiction.

$\Delta \not\models \varphi \quad \Rightarrow \quad \Delta \cup \{\neg\varphi\}$ satisfiable $\quad \Rightarrow \quad$ initial branch satisfiable
$\Rightarrow \quad$ always at least one branch satisfiable (by above lemma)

This contradicts our assumption that at one point all branches will be closed ($\Delta \vdash_{KE} \varphi$), because a closed branch is not satisfiable.

Hintikka's Lemma

Definition 1 (Hintikka set) *A set of sentences H is called a Hintikka set provided the following hold:*

(i) *not both $P \in H$ and $\neg P \in H$ for propositional atoms P;*

(ii) *if $\neg\neg\varphi \in H$ then $\varphi \in H$ for all formulas φ;*

(iii) *if $\alpha \in H$ then $\alpha_1 \in H$ and $\alpha_2 \in H$ for alpha formulas α;*

(iv) *if $\beta \in H$ then $\beta_1 \in H$ or $\beta_2 \in H$ for beta formulas β;*

(v) *for all terms t constructible from function symbols in H (at least one constant symbol): if $\gamma \in H$ then $\gamma_1(t)$ for gamma formulas γ;*

(vi) *if $\delta \in H$ then $\delta_1(t) \in H$ for some term t, for delta formulas δ.*

Lemma 2 (Hintikka) *Every Hintikka set is satisfiable.*

Completeness Proof

Fairness. We call a KE proof *fair* iff every (complex) formula gets *eventually* analysed on every branch and, additionally, every γ-formula gets *eventually* instantiated with every term constructible from the function symbols appearing on a branch.

Proof sketch. We will show the contrapositive: assume $\Delta \nvdash_{KE}$ and try to conclude $\Delta \not\models \varphi$.

If there is no KE proof for $\Delta \cup \{\neg\varphi\}$ (assumption), then there can also be no *fair* KE proof. Show that a fairly constructed non-closable branch enumerates the elements of a Hintikka set H.

H is satisfiable (Hintikka's Lemma) and we have $\Delta \subset H$ and $\neg\varphi \in H$.

So there is a model for $\Delta \cup \{\neg\varphi\}$, i.e. we get $\Delta \not\models \varphi$.

Smullyan Tableaux

The "standard" Tableaux rules (introduced by R. Smullyan in 1968) differ from the KE rules as follows:

- There is *no PB* rule.
- Beta rules are branching rules:

$$\frac{A \vee B}{A \mid B} \qquad \frac{A \rightarrow B}{\neg A \mid B} \qquad \frac{\neg(A \wedge B)}{\neg A \mid \neg B} \qquad \text{in short:} \quad \frac{\beta}{\beta_1 \mid \beta_2}$$

- Similarly for eta rules.

The rest is the same as with the KE system.