Tableaux for propositional logic

Chiara Ghidini – Maurizio Lenzerini

FBK-IRST, Trento, Italy – Sapienza Università di Roma

Logica e informatica – A.Y. 2021/22

Outline

- We will be looking into tableau methods for classical propositional logic.
- Analytic Tableaux are a family of mechanical proof methods, developed for a variety of different logics. Tableaux are nice, because they are both easy to grasp for *humans* and easy to implement on *machines*.

Tableaux

- Early work by Beth and Hintikka (around 1955). Later refined and popularised by Raymond Smullyan:
  - R.M. Smullyan. First-order Logic. Springer-Verlag, 1968.
- More modern expositions include:
  - M. Fitting. First-order Logic and Automated Theorem Proving. 2nd edition. Springer-Verlag, 1996.
  - M. D'Agostino, D. Gabbay, R. Hähnle, and J. Posegga (eds.). Handbook of Tableau Methods. Kluwer, 1999.
  - R. Hähnle. Tableaux and Related Methods. In: A. Robinson and A. Voronkov (eds.), Handbook of Automated Reasoning, Elsevier Science and MIT Press, 2001.
  - Proceedings of the yearly Tableaux conference: `http://i12www.ira.uka.de/TABLEAUX/`

How does it work?

The tableau method is a method for proving, in a mechanical manner, that a given finite set of formulas is not satisfiable. A tableau for a finite set of formulas Γ is a tree structure that is built in a set of steps with the goal of checking whether Γ is unsatisfiable or satisfiable.

Note that this allows us to perform automated *deduction*:

Given :	finite set of premises Γ and conclusion ϕ
Task :	prove $\Gamma \models \phi$
How?	show $\Gamma \cup \{\neg\phi\}$ is not satisfiable,
	i.e. add the complement of the conclusion to the premises
	and derive a contradiction ("refutation procedure")

Reducing logical implication to (un)satisfiability

Theorem

$\Gamma \models \phi$ *if and only if* $\Gamma \cup \{\neg\phi\}$ *is unsatisfiable*

Proof.

$\Rightarrow$ Suppose that $\Gamma \models \phi$, this means that every interpretation $\mathcal{I}$ that satisfies Γ satisfies ϕ too, and therefore $\mathcal{I} \not\models \neg\phi$. This implies that there is no interpretation that satisfies both Γ and $\neg\phi$.

$\Leftarrow$ Suppose that $\Gamma \cup \{\neg\phi\}$ is unsatisfiable. There are two cases: Γ unsatisfiable, or Γ unsatisfiable. In the first case, Γ has no model, and therefore it is obvious that $\Gamma \models \phi$. In the second case, consider any model $\mathcal{I}$ of Γ. Since $\Gamma \cup \{\neg\phi\}$ is not satisfiable, it immediately follows that $\mathcal{I} \models \phi$, and therefore we conclude that $\Gamma \models \phi$.

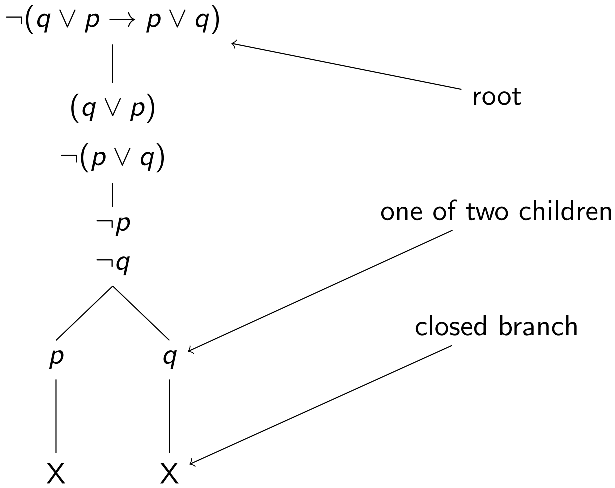
$\square$

Constructing tableau refutation proofs

We want to check $\Gamma \models \phi$ by using refutation, i.e., by trying to build a refutation proof.

- **Data structure**: a candidate refutation proof is represented as a tableau - i.e., a binary tree where each node is labelled with a set of formulas.
- **Start**: we start by putting the finite set of premises Γ and the negated conclusion $\neg\phi$ into the root of an otherwise empty tableau.
- **Expansion**: we keep applying expansion rules to the formulas on the tree as long as it is possible, thereby adding new nodes (formulas) and possibly splitting branches.
- **Closure**: we close branches that are obviously contradictory.
- **Success**: we will prove that the refutation procedure is successful (and therefore a refutation proof is found) if and only if we can close all branches.

An example of the tree data structure



Application of an expansion rule

- Expansion rules have the form:

$$\frac{\phi}{\psi}$$

- Here is what it means to apply an expansion rule of the above form: if the formula ϕ appears in a node belonging to a branch with leaf L, then we perform a suitable action on L, depending on ϕ and ψ.
- Which action has to be performed is specified in the next slides, depending on the different types of expansion rule.

Expansion rules of propositional tableau

α rules

$$\frac{\phi \wedge \psi}{\begin{array}{c}\phi\\ \psi\end{array}} \qquad \frac{\neg(\phi \vee \psi)}{\begin{array}{c}\neg\phi\\ \neg\psi\end{array}} \qquad \frac{\neg(\phi \rightarrow \psi)}{\begin{array}{c}\phi\\ \neg\psi\end{array}}$$

$\neg\neg$-Elimination

$$\frac{\neg\neg\phi}{\phi}$$

β rules

$$\frac{\phi \vee \psi}{\phi \mid \psi} \qquad \frac{\neg(\phi \wedge \psi)}{\neg\phi \mid \neg\psi} \qquad \frac{\phi \rightarrow \psi}{\neg\phi \mid \psi}$$

Branch Closure

$$\frac{\begin{array}{c}\phi\\ \neg\phi\end{array}}{X}$$

Note: These are the standard ("Smullyan-style") tableau expansion rules.

- We do not consider $\equiv$, since we can rewrite $\phi \equiv \psi$ as $(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$
- We regard X as a formula.

Smullyan's uniform notation

Two types of formulas: conjunctive (type-α) and disjunctive (type-β):

α	α_1	α_2
$\phi \wedge \psi$	ϕ	ψ
$\neg(\phi \vee \psi)$	$\neg\phi$	$\neg\psi$
$\neg(\phi \rightarrow \psi)$	ϕ	$\neg\psi$

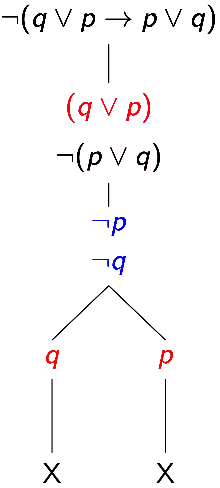
β	β_1	β_2
$\phi \vee \psi$	ϕ	ψ
$\neg(\phi \wedge \psi)$	$\neg\phi$	$\neg\psi$
$\phi \rightarrow \psi$	$\neg\phi$	ψ

We can now state α and β rules as follows:

$$\frac{\alpha}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}} \qquad \frac{\beta}{\beta_1 \mid \beta_2}$$

Note: α rules are also called deterministic rules. β rules are also called splitting rules.

An example of expansion rule application



In this expansion rule application, the branch is the one whose leaf is labeled by the blue set $\{\neg p, \neg q\}$; the formula used to apply the rule is the red formula $(q \vee p)$; the rule is a β-rule, that adds the two red children.

Some definitions

Definition (Closed branch and closed tableau)

A closed branch is a branch which contains X in the leaf, i.e., which contains a formula and its negation. A tableau is closed if all its branches are closed.

Definition (Saturated tableau)

A tableau is saturated if no further applications of expansion rules can produce any new formula on any branch which is not closed.

Note that every closed tableau is saturated.

Definition (Open branch)

In a saturated tableau every branch which is not closed is called open.

Definition (Derivation $\Gamma \vdash_t \phi$)

Let Γ and ϕ be a finite set of propositional formulae and a propositional formula, respectively. We write $\Gamma \vdash_t \phi$ to mean that there exists a closed tableau for $\Gamma \cup \{\neg\phi\}$.

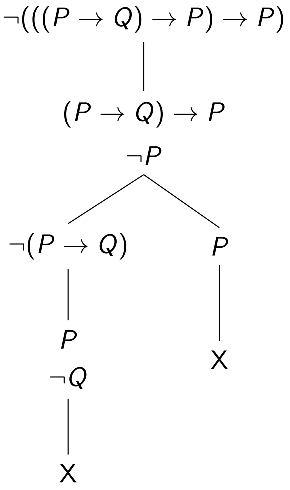
Exercises

Exercise

Show that:

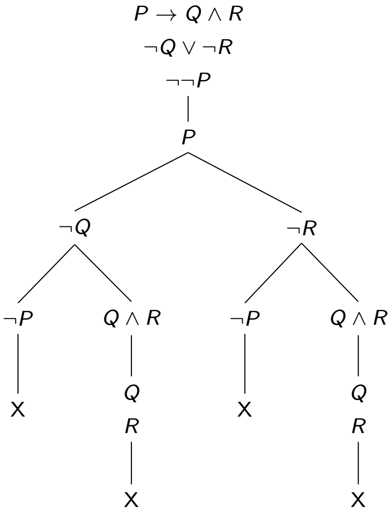
- $\vdash_t ((P \rightarrow Q) \rightarrow P) \rightarrow P$
- $P \rightarrow (Q \wedge R), \neg Q \vee \neg R \vdash_t \neg P$

We show that $\vdash_t ((P \to Q) \to P) \to P$ by exhibiting a closed tableau for $\neg(((P \to Q) \to P) \to P)$



Solutions

We show that $P \to (Q \land R), \neg Q \lor \neg R \vdash_t \neg P$ by exhibiting this tableau:



Note: different orderings of expansion rules are possible!

Tableau algorithm

Is there a systematic way to check whether $\Gamma \vdash_t \phi$? Yes, here is the tableau algorithm:

1. start with a tree constituted only by the root labeled by the set of formulas $\Gamma \cup \{\neg\phi\}$ in the root
2. if all branches are closed, then return yes
3. if not all branches are closed, and the tableau is saturated, then return no
4. if not all branches are closed and the tableau is not saturated, then
   - choose a non-closed branch B and a formula ϕ in B such that an expansion rule application ρ on ϕ produces at least one node in B with a formula not appearing in B (note that such pair $\langle B, \phi \rangle$ exists, because the tableau is not saturated)
   - perform such expansion rule application ρ and go to 2)

Termination

Theorem (Termination)

For any propositional tableau, after a finite number of steps of the tableau algorithm, no more expansion rule will be applicable.

Hint for proof: Since each expansion rule application results in ever shorter formulas, we always come up with a saturated tableau.

Note: Importantly, termination may *not* hold in more powerful logics than propositional logic.

Soundness and completeness

To actually conclude that the tableau method is a correct automated deduction procedure we have to prove two facts.

Theorem (Soundness)

If $\Gamma \vdash_t \phi$ *then* $\Gamma \models \phi$

Theorem (Completeness)

If $\Gamma \models \phi$ *then* $\Gamma \vdash_t \phi$

Remember: We write $\Gamma \vdash_t \phi$ to say that there exists a closed tableau for $\Gamma \cup \{\neg\phi\}$.

Proof of soundness - preliminary definitions

Definition (Satisfiable branch)

A branch B of a tableaux τ is satisfiable if the set of formulas that occurs in B is satisfiable. I.e., if there is an interpretation $\mathcal{I}$, such that $\mathcal{I} \models \phi$ for all ϕ appearing in the label of some node in B.

Proof of soundness - preliminary lemma

First prove the following lemma.

Lemma (Satisfiable branches)

- *If a non-splitting rule is applied to a satisfiable branch, the result is another satisfiable branch.*
- *If a splitting rule is applied to a satisfiable branch, at least one of the resulting branches is also satisfiable.*

Hint for proof: prove for all the expansion rules that they extend a satisfiable branch sb to (at least) a branch sb' which is satisfiable.

Proof of soundness - proof of preliminary lemma

Propositional α-rules: the example of $\wedge$

$$\frac{\phi \wedge \psi}{\begin{array}{c}\phi \\ \psi\end{array}}$$

- let $\mathcal{I}$ be such that $\mathcal{I} \models sb$
- since $\phi \wedge \psi \in sb$ then $\mathcal{I} \models \phi \wedge \psi$
- which implies that $\mathcal{I} \models \phi$ and $\mathcal{I} \models \psi$
- which implies that $\mathcal{I} \models sb'$ with $sb' = sb \cup \{\phi, \psi\}$.

Proof of soundness - proof of preliminary lemma

Propositional β-rules: the example of $\vee$

$$\frac{\phi \vee \psi}{\phi \mid \psi}$$

- let $\mathcal{I}$ be such that $\mathcal{I} \models sb$
- since $\phi \vee \psi \in sb$ then $\mathcal{I} \models \phi \vee \psi$
- which implies that $\mathcal{I} \models \phi$ or $\mathcal{I} \models \psi$
- which implies that $\mathcal{I} \models sb'$ with $sb' = sb \cup \{\phi\}$ or $\mathcal{I} \models sb''$ with $sb'' = sb \cup \{\psi\}$.

Proof of soundness (II)

We have to show that $\Gamma \vdash_t \phi$ implies $\Gamma \models \phi$. We prove it by contradiction, that is, assume $\Gamma \vdash_t \phi$ but $\Gamma \not\models \phi$ and try to derive a contradiction.

- If $\Gamma \not\models \phi$ then $\Gamma \cup \{\neg\phi\}$ is satisfiable (see theorem on relation between logical consequence and (un) satisfiability)
- therefore the initial branch of the tableau (the root $\Gamma \cup \{\neg\phi\}$) is satisfiable
- therefore the tableau for this formula will always have a satisfiable closed branch (easily provable by using previous Lemma on satisfiable branches)
- This contradicts our assumption that at one point all branches will be closed ($\Gamma \vdash_t \phi$), because a closed branch clearly is not satisfiable.
- Therefore we can conclude that $\Gamma \not\models \phi$ cannot be and therefore that $\Gamma \models \phi$ holds.

Proof of completeness - the Hintikka's lemma

Definition (Hintikka set)

A set of propositional formulas Γ is called a Hintikka set provided the following hold:

1. not both $p \in H$ and $\neg p \in H$ for all propositional atoms p;
2. if $\neg\neg\phi \in H$ then $\phi \in H$ for all formulas ϕ ;
3. if $\phi \in H$ and ϕ is a type-α formula then $\alpha_1 \in H$ and $\alpha_2 \in H$;
4. if $\phi \in H$ and ϕ is a type-β formula then either $\beta_1 \in H$ or $\beta_2 \in H$.

Remember:

- type-α formulae are of the form $\phi \wedge \psi$, $\neg(\phi \vee \psi)$, or $\neg(\phi \to \psi)$
- type-β formulae are of the form $\phi \vee \psi$, $\neg(\phi \wedge \psi)$, or $\phi \to \psi$

Proof of completeness - Hintikka's lemma (c'nd)

Lemma (Hintikka Lemma)

Every Hintikka set is satisfiable

Proof:

- We construct a model $\mathcal{I} : \mathcal{P} \rightarrow \{\text{True}, \text{False}\}$ from a given Hintikka set H as follows:
  Let $\mathcal{P}$ be the set of propositional variables occurring in literals of H,

$$\mathcal{I}(p) = \begin{cases} \text{True} & \text{if } p \in H, \\ \text{False} & \text{if } p \notin H. \end{cases}$$

- We now prove by induction that $\mathcal{I}$ is a propositional model that satisfies all the formulae in H. That is, if $\phi \in H$ then $\mathcal{I} \models \phi$.

  Base case We investigate literal formulae.
  Let p be an atomic formula in H. Then $\mathcal{I}(p) = \mathit{True}$ by definition of $\mathcal{I}$. Thus, $\mathcal{I} \models p$
  Let $\neg p$ be a negation of an atomic formula in H. From the property (1) of Hintikka set, the fact that $\neg p$ belongs to H implies that $p \notin H$. Therefore from the definition of $\mathcal{I}$ we have that $\mathcal{I}(p) = \mathit{False}$, and therefore $\mathcal{I} \models \neg p$

Proof of completeness - Hintikka's lemma (c'nd)

Inductive step We prove the theorem for all non-literal formulae.

- Let θ be of the form $\neg\neg\phi$.
  Then because of the property (2) of Hintikka sets $\phi \in H$.
  Therefore $\mathcal{I} \models \phi$ because of the inductive hypothesis.
  Then $\mathcal{I} \not\models \neg\phi$ and $\mathcal{I} \models \neg\neg\phi$ because of the definition of propositonal satisfiability of $\neg$.
- Let θ be a type-α formula. Then, its components α_1 and α_2 belong to H because of property (3) of the Hintikka set.
  We can apply the inductive hypothesis to α_1 and α_2 and derive that $\mathcal{I} \models \alpha_1$ and $\mathcal{I} \models \alpha_2$
  It is now easy to prove that $\mathcal{I} \models \theta$
- Let θ be a type-β formula. Then, at least one of its components β_1 or β_2 belong to H because of property (4) of the Hintikka set.
  We can apply the inductive hypothesis to β_1 or β_2 and derive that $\mathcal{I} \models \beta_1$ or $\mathcal{I} \models \beta_2$
  It is now easy to prove that $\mathcal{I} \models \theta$

Proof of completeness

Completeness proof (sketch).

- We show that $\Gamma \not\vdash_t \phi$ implies $\Gamma \not\models \phi$.
- Suppose that there is no proof for $\Gamma \cup \{\neg\phi\}$
- Let τ a saturated tableaux that start with $\Gamma \cup \{\neg\phi\}$,
- The fact that $\Gamma \not\vdash_t \phi$ implies that there is at least one open branch *ob*.
- The saturation condition implies that the set of formulas in *ob* constitute an Hintikka set H_{ob}
- From Hintikka lemma we have that there is an interpretation $\mathcal{I}_{ob}$ that satisfies *ob*.
- Since every branch of τ contains its root we have that $\Gamma \cup \{\neg\phi\} \subseteq ob$ and therefore $\mathcal{I}_{ob} \models \Gamma \cup \{\neg\phi\}$.
- We therefore conclude that $\Gamma \not\models \phi$.

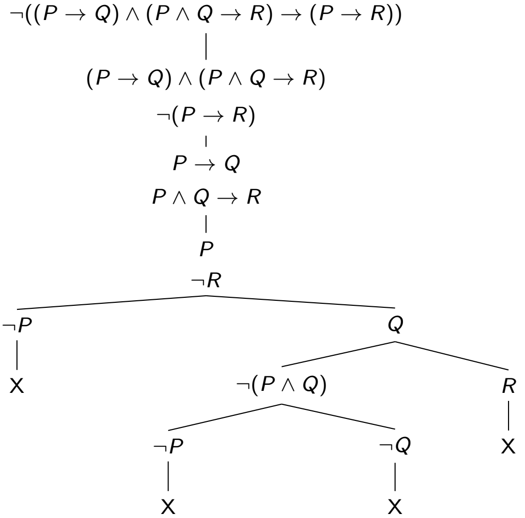
$\square$

Tableaux and satisfiability

- Obviously, we can use tableaux to check if a formula ϕ is satisfiable: apply the tableau algorithm to the tableau constituted by the root labeled by ϕ; if you got a closed tableau, ϕ is not satisfiable, otherwise it is satisfiable.

Exercise

Check whether the formula $\neg((P \to Q) \wedge (P \wedge Q \to R) \to (P \to R))$ is satisfiable.

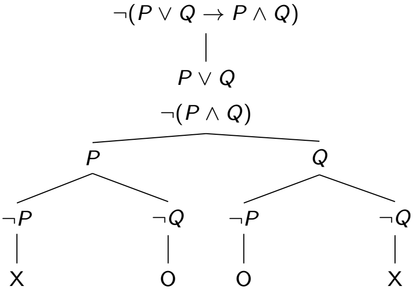


The tableau is closed and the formula is not satisfiable.

Tableaux and satisfiability: another exercise

Exercise

Check whether the formula $\neg(P \vee Q \rightarrow P \wedge Q)$ is satisfiable



Two open branches. The formula is satisfiable.
The tableau shows us all the possible interpretations $(\{P\}, \{Q\})$ that satisfy the formula.

Using the tableau to build interpretations

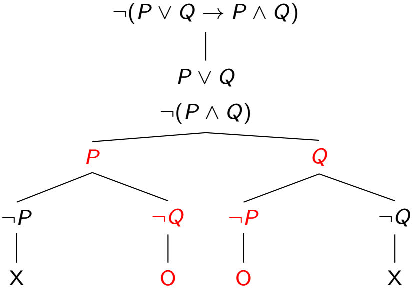
We can use a tableau in order to build an interpretation for the formula.

For each open branch in the tableau, and for each propositional atom p in the formula we define

$$\mathcal{I}(p) = \begin{cases} \text{True} & \text{if } p \text{ belongs to the branch,} \\ \text{False} & \text{if } \neg p \text{ belongs to the branch.} \end{cases}$$

If neither p nor $\neg p$ belong to the branch we can define $\mathcal{I}(p)$ in an arbitrary way.

Models for $\neg(P \vee Q \rightarrow P \wedge Q)$



Two models:

- $\mathcal{I}(P) = \text{True}, \mathcal{I}(Q) = \text{False}$
- $\mathcal{I}(P) = \text{False}, \mathcal{I}(Q) = \text{True}$

Double-check with the truth tables!

P	Q	$P \vee Q$	$P \wedge Q$	$P \vee Q \rightarrow P \wedge Q$	$\neg(P \vee Q \rightarrow P \wedge Q)$
T	T	T	T	T	F
F	F	F	F	T	F
T	F	T	F	F	T
F	T	T	F	F	T

Homeworks!

Exercise

Show *unsatisfiability* of each of the following formulae using tableaux:

- $(p \equiv q) \equiv (\neg q \equiv p)$;
- $\neg((\neg q \to \neg p) \to ((\neg q \to p) \to q))$.

Show *satisfiability* of each of the following formulae using tableaux:

- $(p \equiv q) \to (\neg q \equiv p)$;
- $\neg(p \vee q \to ((\neg p \wedge q) \vee p \vee \neg q))$.

Show *validity* of each of the following formulae using tableaux:

- $(p \to q) \to ((p \to \neg q) \to \neg p)$;
- $(p \to r) \to (p \vee q \to r \vee q)$.

For each of the following formulae, *describe all models* of this formula using tableaux:

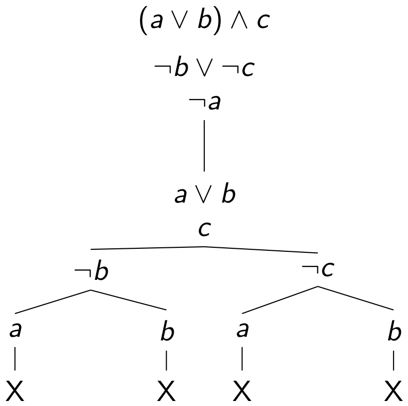
- $(q \to (p \wedge r)) \wedge \neg(p \vee r \to q)$;
- $\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))$.

Establish the *equivalences* between the following pairs of formulae using tableaux:

- $(p \to \neg p), \neg p$;
- $(p \to q), (\neg q \to \neg p)$;
- $(p \vee q) \wedge (p \vee \neg q), p$.

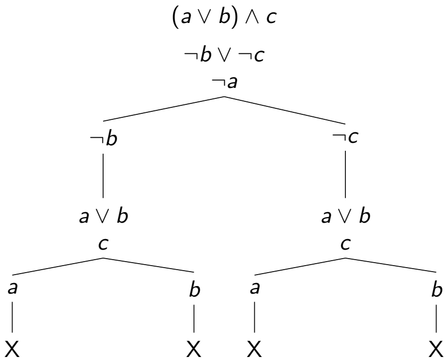
Exercise

Build a tableau for $\{(a \vee b) \wedge c, \neg b \vee \neg c, \neg a\}$



Another solution

What happens if we first expand the disjunction and then the conjunction?



Expanding β rules creates new branches. Then α rules may need to be expanded in all of them.

Strategies of expansion

- Using the "wrong" policy (e.g., expanding disjunctions first) leads to an increase of *size* of the tableau, which leads to an increase of *time* in the execution of the algorithm;
- yet, unsatisfiability is still proved if the set is unsatisfiable;
- this is not the case for other logics, where applying the wrong policy may inhibit proving unsatisfiability of some unsatisfiable sets.

Finding short proofs

- It is an open problem to find an efficient algorithm to decide in all cases which rule to use next in order to derive the shortest possible proof.
- However, as a rough guideline always apply any applicable *non-branching rules* first. In some cases, these may turn out to be redundant, but they will often prevent an exponential blow-up of the tableaux.

Efficiency

- Are analytic tableaux an efficient method of checking whether a formula is a tautology?
- Remember: using the truth-tables to check a formula involving n propositional atoms requires filling in 2^n rows (exponential $=$ very bad).
- Are tableaux any better?
- In the worst case no, but if we are lucky we may skip some of the 2^n rows !!!

Exercise

Exercise

Give proofs for the unsatisfiability of the following formula using (1) truth-tables, and (2) Smullyan-style tableaux.

$$(P \vee Q) \wedge (P \vee \neg Q) \wedge (\neg P \vee Q) \wedge (\neg P \vee \neg Q)$$