CS 511, Fall 2024, Lecture Slides 29, Part II

Analytic Tableaux for Classical First-Order Logic

Assaf Kfoury

24 October 2024

REVIEW and PRELIMINARIES

- These lecture slides continue Lecture Slides 29, Part I, which introduced tableaux for propositional logic and tableaux for first-order logic.
- These lecture slides also depend on Lecture Slides 25, which is a presentation of unification, limited to the kind we use in **first-order tableaux** (and, later, in **first-order resolution**).

second TABLEAU method: FREE VARIABLES + UNIFICATION

- We avoid some of the problems in the *first tableau method* (in Lecture Slides 29, Part I), by modifying the quantifier rules and how we use them – informally:
  - delay applications of rule $(\forall)$, the source of the problems, when possible,
  - when $(\forall)$ is applied, instantiate with a fresh variable (not a ground term),
  - the generated sub-formulas in the tableau T are thus no longer closed,
  - the new fresh variables in T are implicitly universally quantified outside T.

[1] Note the (subtle) error in the rule $(\exists)$ in the Wikipedia article, under "**First-order tableau with unification**" – click here.

second TABLEAU method: FREE VARIABLES + UNIFICATION

- We avoid some of the problems in the *first tableau method* (in Lecture Slides 29, Part I), by modifying the quantifier rules and how we use them – informally:
  - delay applications of rule $(\forall)$, the source of the problems, when possible,
  - when $(\forall)$ is applied, instantiate with a fresh variable (not a ground term),
  - the generated sub-formulas in the tableau T are thus no longer closed,
  - the new fresh variables in T are implicitly universally quantified outside T.
- Modified quantifier rules for *second tableau method*:
  - rule $(\forall)$ for WFF's that start with a universal quantifier:

    $$(\forall) \qquad \frac{\forall x\, \varphi(x)}{\varphi[x := y]}$$

    where y is a new fresh variable,
  - rule $(\exists)$ for WFF's that start with an existential quantifier:

    $$(\exists) \qquad \frac{\exists x\, \varphi(x)}{\varphi[x := f(y_1, \ldots, y_n)]}$$

    where f is a new Skolem function and $\{y_1, \ldots, y_n\} = \mathrm{FV}(\exists x\, \varphi)$.[1]

[1] Note the (subtle) error in the rule $(\exists)$ in the Wikipedia article, under "**First-order tableau with unification**" – click here.

second TABLEAU method: FREE VARIABLES + UNIFICATION

- What to do with the free variables that rule $(\forall)$ insert in a tableau?

  We need to introduce an additional rule, called the ***substitution rule***, which, every time it is applied, is relative to what is called a ***unifier***.

second TABLEAU method: FREE VARIABLES + UNIFICATION

- What to do with the free variables that rule $(\forall)$ insert in a tableau?

  We need to introduce an additional rule, called the ***substitution rule***, which, every time it is applied, is relative to what is called a ***unifier***.

- If σ is a *unifier*, then we will write "(σ)" to denote the *substitution rule* relative to σ, spelled out as follows:

  (σ) If σ is the most general unifier (MGU) of two literals A and B, where A and $\neg B$ are on the same path of tableau T, then σ is applied simultaneously to all the WFF's in T.

  where a *literal* is an atomic WFF.

second TABLEAU method: FREE VARIABLES + UNIFICATION

- For a precise formulation of (σ):
  - If T is a tableau, and π is a path from the root of T to a leaf node in T, then

    $$T \oplus_\pi \varphi$$

    is a new tableau obtained from T by appending φ below the path π.
  - *WFF's*(π) is the set of WFF's occurring along a path π in a tableau.
  - *MGU*(A, B) is the most general unifier of two literals (atomic formulas).
  - *paths*(T) is the set of paths in the tableau T.

second TABLEAU method: FREE VARIABLES + UNIFICATION

- For a precise formulation of (σ):
  - If T is a tableau, and π is a path from the root of T to a leaf node in T, then

    $$T \oplus_\pi \varphi$$

    is a new tableau obtained from T by appending φ below the path π.
  - *WFF's*(π) is the set of WFF's occurring along a path π in a tableau.
  - *MGU*(A, B) is the most general unifier of two literals (atomic formulas).
  - *paths*(T) is the set of paths in the tableau T.
- Rule (σ) for tableaux with free variables:

$$\boxed{(\sigma) \quad \frac{T}{\sigma(T) \oplus_\pi \times} \qquad \pi \in \textit{paths}(T), \{A, \neg B\} \subseteq \textit{WFF's}(\pi), \sigma = \textit{MGU}(A, B)}$$

Note that the unifier σ is applied to the entire tableau T.

second TABLEAU method: FREE VARIABLES + UNIFICATION

- For a precise formulation of (σ):
  - If T is a tableau, and π is a path from the root of T to a leaf node in T, then

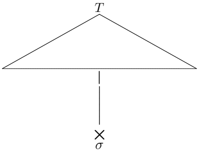
    $$T \oplus_\pi \varphi$$

    is a new tableau obtained from T by appending φ below the path π.
  - *WFF's*(π) is the set of WFF's occurring along a path π in a tableau.
  - *MGU*(A, B) is the most general unifier of two literals (atomic formulas).
  - *paths*(T) is the set of paths in the tableau T.
- Rule (σ) for tableaux with free variables:

$$\boxed{(\sigma) \quad \frac{T}{\sigma(T) \oplus_\pi \times} \qquad \pi \in \textit{paths}(T), \{A, \neg B\} \subseteq \textit{WFF's}(\pi), \sigma = \textit{MGU}(A, B)}$$

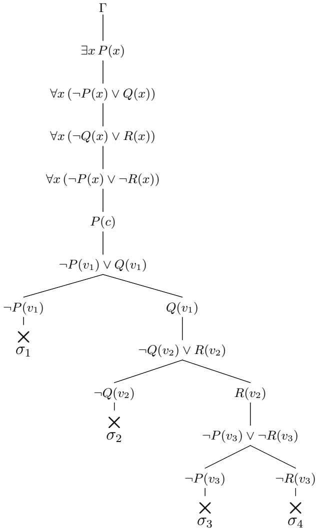
Note that the unifier σ is applied to the entire tableau T.

Schematically in the example on the next slide:



second TABLEAU method: example

$$\Gamma \stackrel{\text{def}}{=} \Big\{ \exists x\, P(x),\ \forall x\, \big(\neg P(x) \vee Q(x)\big),\ \forall x\, \big(\neg Q(x) \vee R(x)\big),\ \forall x\, \big(\neg P(x) \vee \neg R(x)\big) \Big\}$$



where $\sigma_1 \stackrel{\text{def}}{=} \{v_1 \mapsto c\}$, $\sigma_2 \stackrel{\text{def}}{=} \{v_2 \mapsto c\}$, $\sigma_3 \stackrel{\text{def}}{=} \{v_3 \mapsto c\}$, $\sigma_4 \stackrel{\text{def}}{=} \{\ \}$ (identity substitution)

second TABLEAU method: FREE VARIABLES + UNIFICATION

Soundness and completeness of the ***free-variable tableau method*** also hold:

- ***Soundness*** of rules $\{(\forall), (\exists), (\sigma)\}$ (together with the rules for propositional tableaux): *If we can generate a closed tableau from an initial set Γ of sentences (in prenex normal form), then Γ is unsatisfiable.*
- ***Completeness*** of rules $\{(\forall), (\exists), (\sigma)\}$ (together with the rules for propositional tableaux): *If a set Γ of sentences (in prenex normal form) is unsatisfiable, there exists a closed tableau generated from Γ by these rules.*

ground TABLEAUX versus free-variable TABLEAUX

- We compare the two methods on a simple example:

$$\Gamma \stackrel{\text{def}}{=} \Big\{\forall x \forall y \left(P(x,y) \to P(y,x)\right),\ P(a,b),\ P(b,c),\ \neg P(c,b)\Big\}$$

- By easy inspection, Γ is not satisfiable – which will be here confirmed by tableaux.

[2] There are different ways of defining the optimality of a tableau. For simplicity here, we identify **optimality** with **least number of applications of the expansion rules**.

ground TABLEAUX versus free-variable TABLEAUX

- We compare the two methods on a simple example:

$$\Gamma \stackrel{\text{def}}{=} \Big\{\forall x \forall y \left(P(x,y) \to P(y,x)\right),\ P(a,b),\ P(b,c),\ \neg P(c,b)\Big\}$$

- By easy inspection, Γ is not satisfiable – which will be here confirmed by tableaux.

Preliminary remarks for a first comparison:

- We first compare the two methods with no look-ahead of any kind and no heuristics of any kind (*e.g.*, apply "unary" rules before "binary" rules). The resulting tableaux are **not optimal**.[2]
- For this example, the set of ground terms is finite: $\{a, b, c\}$.
- For brevity, we merge two consecutive applications of rule $(\forall)$ into a single step, when applied to the sentence $\forall x \forall y \left(P(x,y) \to P(y,x)\right)$. Moreover, for brevity again, we merge into that single step the application of rule $(\to)$ which immediately follows it.
- We assume a fixed order in which pairs of ground terms are generated, namely: $(a,a), (a,b), (a,c), \quad (b,a), (b,b), (b,c), \quad (c,a), (c,b), (c,c)$, which is the order in which the variable pair (x,y) is instantiated to ground terms.

[2] There are different ways of defining the optimality of a tableau. For simplicity here, we identify **optimality** with **least number of applications of the expansion rules**.

ground TABLEAUX versus free-variable TABLEAUX

- On slide 15 is a ground tableau (first method) for Γ (which is just too large to fit in a single slide ...).
- On slide 16 is a free-variable tableau (second method) for Γ.
- Both tableaux are organized similarly, **but not optimally**:
  - Every node is labelled with a boxed pair of integers $\boxed{i : j}$ with $i > j \geqslant 0$:

    i is the unique ID number of the node in the tableau,
    j is the ID number of the node on which node i depends.
  - Label $\boxed{i : 0}$ means the WFF at node i is from Γ.
  - Node ID's are linearly ordered in the order in which the tableau is developed:

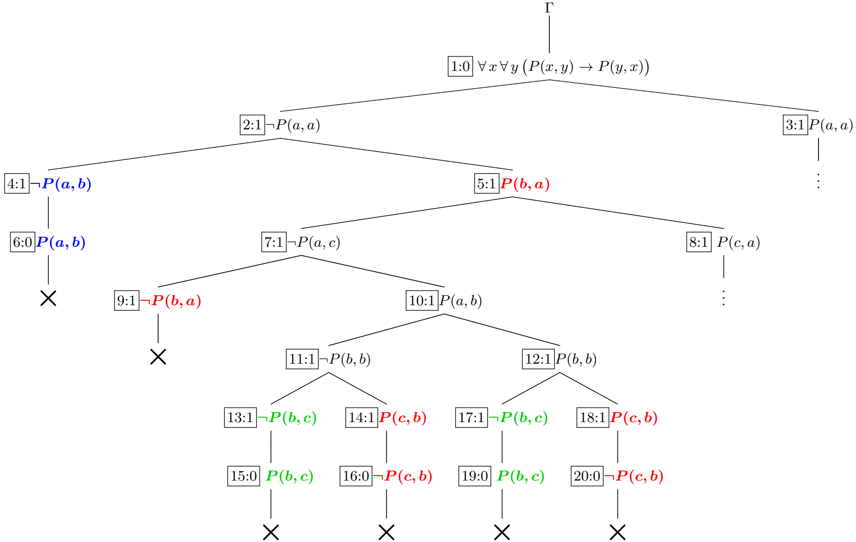
    using WFF's in Γ in their given order from left to right ,[3]

    except when a conflict between atomic WFF's is detected.

[3] So that, in particular, $\forall x \forall y \left(P(x, y) \to P(y, x)\right)$ is considered first and ahead of $P(a, b)$, $P(b, c)$, and $\neg P(c, b)$.

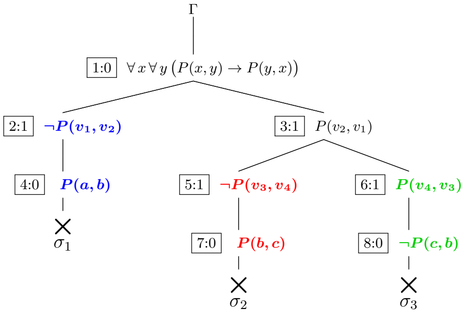
ground TABLEAUX versus free-variable TABLEAUX

a **ground tableau** for $\Gamma \stackrel{\text{def}}{=} \Big\{\forall x\, \forall y\, \big(P(x,y) \to P(y,x)\big),\ P(a,b),\ P(b,c),\ \neg P(c,b)\Big\}$



ground TABLEAUX versus free-variable TABLEAUX

a **free-variable tableau** for $\Gamma \stackrel{\text{def}}{=} \Big\{\forall x \forall y \,\big(P(x,y) \to P(y,x)\big),\ P(a,b),\ P(b,c),\ \neg P(c,b)\Big\}$



where $\sigma_1 \stackrel{\text{def}}{=} \{v_1 \mapsto a, v_2 \mapsto b\}$

$\sigma_2 \stackrel{\text{def}}{=} \{v_3 \mapsto b, v_4 \mapsto c\}$

$\sigma_3 \stackrel{\text{def}}{=} \{\ \}$ (identity substitution)

ground TABLEAUX versus free-variable TABLEAUX

Preliminary remarks for a second comparison:

- We use the same notation and conventions as those in the **first comparison**.
- We use the same ordering of the WFF's in Γ, and the same ordering of pairs of ground terms, as those in the **first comparison**.
- Where the **second comparison** is different from the **first comparison**:
  - We use the heuristic ***unary*** expansion rules before ***binary*** expansion rules.
  - We instantiate the variable pair (x, y) only to ground terms directly leading to a conflict.

    Specifically, (x, y) is instantiated to the first pair in $\{(a, a), (a, b), \ldots, (c, c)\}$ that makes one (or both) of the branches of the expansion of $\forall x \forall y \left(P(x, y) \to P(y, x)\right)$ contradicts an earlier WFF on the same path from the root.

ground TABLEAUX versus free-variable TABLEAUX

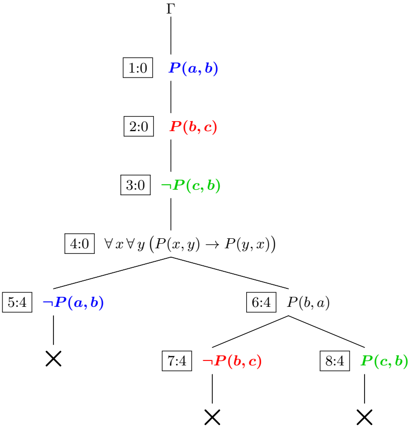
Preliminary remarks for a second comparison:

- We use the same notation and conventions as those in the **first comparison**.
- We use the same ordering of the WFF's in Γ, and the same ordering of pairs of ground terms, as those in the **first comparison**.
- Where the **second comparison** is different from the **first comparison**:
  - We use the heuristic ***unary*** expansion rules before ***binary*** expansion rules.
  - We instantiate the variable pair (x, y) only to ground terms directly leading to a conflict.

    Specifically, (x, y) is instantiated to the first pair in $\{(a, a), (a, b), \ldots, (c, c)\}$ that makes one (or both) of the branches of the expansion of $\forall x\,\forall y\,\big(P(x, y) \to P(y, x)\big)$ contradicts an earlier WFF on the same path from the root.
- With these added heuristics, the two methods appear equally efficient – at least for Γ in this example.
- On slide 19 is a ground tableau (first method) for Γ (now small enough to fit in a single slide).
- On slide 20 is a free-variable tableau (second method) for Γ.
- Can we do better? One more free-variable tableau (second method) for Γ is on slide 21, which is better (shorter) than all the preceding tableaux.

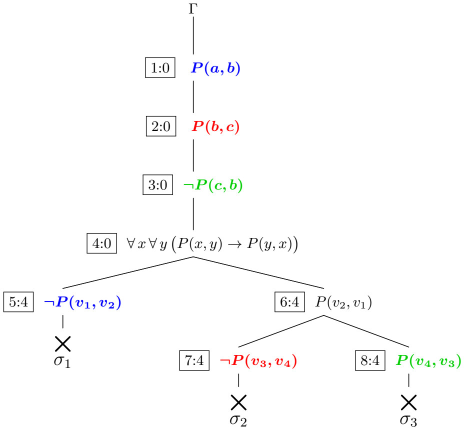
ground TABLEAUX versus free-variable TABLEAUX

another **ground tableau** for $\Gamma \stackrel{\text{def}}{=} \{\forall x \forall y\, (P(x,y) \to P(y,x)),\ P(a,b),\ P(b,c),\ \neg P(c,b)\}$



ground TABLEAUX versus free-variable TABLEAUX

another **free-variable tableau** for $\Gamma \stackrel{\text{def}}{=} \Big\{\forall x \forall y\big(P(x,y) \to P(y,x)\big), P(a,b), P(b,c), \neg P(c,b)\Big\}$



where $\sigma_1 \stackrel{\text{def}}{=} \{v_1 \mapsto a, v_2 \mapsto b\}$

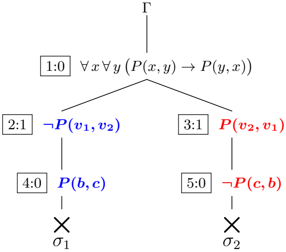
$\sigma_2 \stackrel{\text{def}}{=} \{v_3 \mapsto b, v_4 \mapsto c\}$

$\sigma_2 \stackrel{\text{def}}{=} \{\ \}$ (identity substitution)

ground TABLEAUX versus free-variable TABLEAUX

one more **free-variable tableau** for

$$\Gamma \stackrel{\text{def}}{=} \Big\{\forall x \forall y \big(P(x,y) \to P(y,x)\big), P(a,b), P(b,c), \neg P(c,b)\Big\}$$



where $\sigma_1 \stackrel{\text{def}}{=} \{v_1 \mapsto b, v_2 \mapsto c\}$

$\sigma_2 \stackrel{\text{def}}{=} \{\ \}$ (identity substitution)

second TABLEAU method: exercises

1. **Exercise**. Redo Exercise 1 on the last slide of Lecture Slides 29, Part I, now using free-variable tableaux. Spell out a strategy that will minimize the size of the tableau you produce.
2. **Exercise**. Redo Exercise 2 on the last slide of Lecture Slides 29, Part I, now using free-variable tableaux. Spell out a strategy that will minimize the size of the tableau you produce.

(THIS PAGE INTENTIONALLY LEFT BLANK)