Towards the general notion
of a synthetic tableau system∗

Dorota Leszczyńska-Jasion†

Abstract
In this note I consider a general notion of a system of synthetic tableaux. The notion generalizes definitions introduced in [2], [4], [1].

The method of synthetic tableaux
In order to describe the ST-systems for such logics as CPL, FOL and propositional modal logics in one account, we have developed the general notion of synthetic tableau system.

Definition 1. Let $\mathcal{L}$ be a formal language and let $\mathbf{L}$ be a logic expressed in this language (i.e. a set of formulas of $\mathcal{L}$). Then by Synthetic Tableau System for $\mathbf{L}$ (ST-system for $\mathbf{L}$) we understand a tuple:

$$(\mathcal{L}, \mathcal{A}, \mathcal{R}_S, \mathcal{R}_B, \mathcal{R}_T, \mathcal{R}_C)$$

where

- $\mathcal{A} \subseteq \text{FORM}_\mathcal{L}$ is a set of formulas called atoms;
- $\mathcal{R}_S$ is a set of the linear (synthesizing) rules, where each rule is a set of pairs $\langle P, C \rangle$ or a set of triples $\langle P_1, P_2, C \rangle$; formulas $P, P_1, P_2 \in \text{FORM}_\mathcal{L}$ are premises and $C \in \text{FORM}_\mathcal{L}$ is a conclusion of the rule;
- $\mathcal{R}_B$ is a set of the branching rules understood as the set of pairs $\langle a, \neg a \rangle$, where $a \in \mathcal{A}$;
- $\mathcal{R}_T$ is the set of rules for creating subtableaux; this are formalized as functions which assign a synthetic tableau to a synthetic tableau;
- $\mathcal{R}_C$ contains the rules of consistency check.

∗ This draft, prepared in November 2016, was a part of the project supported by funds of the National Science Centre, Poland, grant no. 2012/04/A/HS1/00715.
† Department of Logic and Cognitive Science, Institute of Psychology, Adam Mickiewicz University, e-mail address: Dorota.Leszczynska@amu.edu.pl
The atoms from $A$ are to be understood syntactically—they perform the role of syntactical atoms in the construction of a synthetic tableau, but they need not be atoms in the semantical sense, since they need not be logically independent. Actually, the only case of pure logical independence of atoms is that of CPL, where $A = VAR$. Atoms such as $(\forall x)P(x)$ and $(\exists x)P(x)$ are not logically independent, similarly atoms $\Box p$ and $\Diamond p$ in modal logics.

The characterization of set $R_B$ presupposes that “$\neg$” is in the language and that it behaves classically. This need not be the case. The method has been adjusted to logics such as three-valued Lukasiewicz logic $L3$ or paraconsistent logic $CLuN$ (see [3], [5], respectively). The author used truth-signs to incorporate the third logical value (in the case of $L3$) or the classical negation (in the case of $CLuN$) into the language of tableaux. This warrants that branchings “partition the logical space”. These examples show that our account may be too narrow. On the other hand, a modification of synthetic tableau systems for first-order and for modal logics introducing truth-signs seems to be straightforward.

An example of rules in $R_T$ are $UG1$ and $UG2$. It is difficult to formalize such rules in the form of pairs, etc., exactly because they are global. Moreover, the account of rules as functions is much more general and suitable to account for all the rules of the system. E.g., the branching rule $(a, \neg a)$ says that if $T$ is a synthetic tableau for formula $A$ and $a \in \text{Sub}(A)$, then $T_1$, described below, is also a synthetic tableau for $A$.

- $T_1$ is just like $T$ except that for certain branch $B$ of $T$, $T_1$ has two branches $B_1$ and $B_2$ instead of $B$, where $B_1$ is like $B$ but has one more node (leaf) labelled with $a$, and $B_2$ is like $B$ but has one more node (leaf) labelled with $\neg a$.

The reader will observe that in such an account we have to add the rule which says that a tree with only one node (the root with no label) is a synthetic tableau for every formula. This may be added as a special case of rules from any of $R_S, R_B, R_T$.

Similarly, rule $UG1$ says that if $T$ is a synthetic tableau for formula $A$ such that:

- there is a subtree $T^*$ of $T$ such that every open branch of $T^*$ ends with formula $C \rightarrow A(x)$, and
- $x$ does not occur freely in $C$, and
- no formula in $T^*$ has been synthesized with the use of a premise which is not on $T^*$, then

$T_1$ described below is also a synthetic tableau for $A$.

- $T_1$ is just like $T$ except that for every open branch $B$ of $T$, tree $T_1$ has branch $B_1$ which, again, is just like $B$ except that it has one more node (leaf) with formula $C \rightarrow (\forall x)A(x)$ on it.

Observe that this time we do not restrict $C \rightarrow (\forall x)A(x)$ to be in $\text{Sub}(A)$.

The description of this kind of rules is even more complicated in the case of rules for modal logics.
The last component of tuple (1) consists of the rules for consistency check. In the case of FOL there is only one such rule, composed into the notion of proof—it says that a branch containing $F$ and $\neg F$ is closed. Again, the situation gets more complicated in the case of modal logics.

In the case of CPL, $\mathcal{R}_T = \mathcal{R}_C = \emptyset$. This simplicity is, however, a specific feature of the logic, not of the ST-systems.

References


