Labelled Tableaux for D_2

Janusz Ciuciura

Abstract
In the late forties, Stanisław Jaśkowski published his well-know papers on the *discursive sentential calculus*, D_2. He provided a definition of it by an interpretation in the language of S_5 of Lewis. However, it is the *sisyphean labour* to transform any discursive formula into its modal counterpart. The inconvenience results in the search for a new simple tool we could use trying to answer the question whether a discursive formula is valid in D_2 or it is not. We intend to introduce a more efficient method that makes the translation procedure redundant and present a direct semantics and (*Signed* and *Unsigned*) *Labelled Tableaux* for D_2.

1. Introduction

Discursive or *Discussive* Logic, introduced by Jaśkowski and further promoted by Kotas, Dubikajtis and the others, has already found its way in contemporary logic as a first formal approach to paraconsistency. Jaśkowski fully recognized the need for logic to be flexible enough to depict some of "weaknesses" of the natural language such as vagueness and imprecision. It sometimes happens in science, not to mention everyday life, that we use terms, which are more or less vague or imprecise. Even in literature we can come across a good example of how dangerous it always is to reason from the terms:

> "I had - said he [Sherlock Holms] – come to an entirely erroneous conclusion… The presence of the gypsies, and the use of the word *band*, which was used by the poor girl, no doubt, to explain the appearance which she had caught a horrid glimpse of by the light of her match, were sufficient to put me upon an entirely wrong scent."[1]

Jaśkowki realized vagueness of a term could lead to *seeming contradictions*. It can easily be noticed, in our experience, that we generally fall into them while discussing on a topic. Since we cannot safely assume that imprecise terms and seeming contradictions will never occur among participants in the discussion our formal language should allow the contradictions to be present. Unfortunately, classical and almost all of the nonclassical logics he considered in [6] are not too helpful to gain a real insight into their nature. Jaśkowski decided to apply a translation procedure instead and the system S_5 of Lewis turned out to be very useful for that purpose.

To characterize the translation, we will firstly apply a standard sentential language of S_5, $\boldsymbol{Lg}_{S5}$ for short, whose primitive vocabulary includes four classical connectives, i.e. $\sim$, $\vee$, $\wedge$, $\rightarrow$, and two modal symbols, i.e. M (possibility), L (necessity). The set of all propositional variables will be denoted by *var* and the capital letters {P, Q, R,…} stand for (schemata of) formulas. The language $\boldsymbol{Lg}_{D2}$ differs from $\boldsymbol{Lg}_{S5}$ in many respects. Not only do the modal

[1] Sir Arthur Conan Doyle, *The Speckled Band*, in: **The Best of Sherlock Holmes**, Wordsworth Classics, Hertfordshire 1998, p.119.

connectives disappear, but also the classical conjunction, implication and equivalence are absent from $\boldsymbol{Lg}_{D2}$. They are replaced by their discursive counterparts for purely technical and philosophical reason.[2]

Providing a definition of D_2, we compare the languages of D_2 and S_5

- $\boldsymbol{Lg_{D2}} = < For_{D2}, \sim, \vee, \wedge_d, \rightarrow_d >$;
- $\boldsymbol{Lg_{S5}} = < For_{S5}, \sim, \vee, \wedge, \rightarrow, M, L >$,

define a function f: $For_{D2} \Rightarrow For_{S5}$ as follows:

(i) $f(p) = p$ if $p \in var$;
(ii) $f(\sim P) = \sim f(P)$;
(iii) $f(P \vee Q) = f(P) \vee f(Q)$;
(iv) $f(P \wedge_d Q) = f(P) \wedge Mf(Q)$;
(v) $f(P \rightarrow_d Q) = Mf(P) \rightarrow f(Q)$[3]

and finally, introduce the key definition:

(vi) For every formula $P \in For_{D2}$, $P \in D_2$ iff $Mf(P) \in S_5$.

Although Jaśkowski gave a general description of how to transform any discursive formula into its modal equivalent, the translation procedure was a little unhandy to operate in practice. To illustrate it, one can show, as a simple example, that the formula $(\sim P \wedge_d \sim (P \rightarrow_d Q)) \rightarrow_d (Q \rightarrow_d (P \wedge_d \sim P))$ is valid in D_2. In fact, we are made to pass through the indicated translation and show the formula $M(M(\sim P \wedge M \sim (M P \rightarrow Q)) \rightarrow (M Q \rightarrow (P \wedge M \sim P)))$ to be valid in S_5. The inconvenience results in the search for a new tool we could apply trying to avoid using the translation rules.

2. Kripke Semantics for D_2

We present here a direct semantics for D_2 which, unlike the one in [8], is suggested by Kripke-type semantics. As, in [3], we have already used it while introducing a new axiomatization for D_2 we limit ourselves to only recapitulating its main points.

By Kripke frame (D_2-frame) we mean a pair $<W, R>$, where W is a non-empty set (of points, possible worlds) and R is a binary relation on W. Furthermore, R is subject to the conditions:

(i) $x R x$, for every $x \in W$;
(ii) if $x R y$, then $y R x$, for every $x, y \in W$;
(iii) if $x R y$ and $y R z$, then $x R z$, for every $x, y, z \in W$.

The conditions define R as being the equivalence relation on W.

A Kripke model $\boldsymbol{M}$ (D_2-model) is a triple $<W, R, v>$, where v is a mapping from propositional variables to sets of worlds (more formally, v: $var \Rightarrow 2^W$). The satisfaction relation $\models_{\boldsymbol{m}}$ is defined:

---

[2] For details, see [6], [8], [1] and [2].

[3] $P \leftrightarrow_d Q = (P \rightarrow_d Q) \wedge_d (Q \rightarrow_d P)$.

(*var*)	$x \models_m p$	iff	$x \in v(p)$;
(~)	$x \models_m \sim P$	iff	$x \not\models_m P$;
(∨)	$x \models_m P \vee Q$	iff	$x \models_m P$ or $x \models_m Q$;
($\wedge_d$)	$x \models_m P \wedge_d Q$	iff	$x \models_m P$ and there exists some $y \in W$, with $x Ry$ and $y \models_m Q$;
($\rightarrow_d$)	$x \models_m P \rightarrow_d Q$	iff	if there exists some $y \in W$, with $x Ry$ and $y \models_m P$ then $x \models_m Q$.

Next we define the notion of a valid sentence:

$\models_M P$ iff for any model $\boldsymbol{M}$, any $x \in W$, there exists $y \in W$ such that: $x R y$ and $y \models_m P$.

For a convenient notation, we will henceforth write:

(Ab) $x \models^m P$ iff there exists $y \in W$ such that $x R y$ and $y \models_m P$.

Consequently, we obtain:

$\models_M P$ iff for any model $\boldsymbol{M}$ and any $x \in W$, $x \models^m P$.

The semantics we adopted is free from the translation rules. It enables us to take a step in the right, for our purpose, direction.

3. Signed Labelled Tableaux for D_2

We assume familiarity with tableau methods and confine ourselves to recalling the relevant terminology (see [4] or [5] for details). In our language, from now on, we will use *signed labelled formulas* such as $\sigma :: T\,P$ (or $\sigma :: F\,P$), where σ is a label and $T\,P$ (or $F\,P$) is a *signed formula* (i.e. a formula prefixed with a "T" or "F"). Intuitively, $\sigma :: T\,P$ is read as "P is true at the world σ" and $\sigma :: F\,P$ as "P is false at the world σ". By *label*, we understand a nonempty sequence of natural numbers separated by periods. The length of a label σ is the number of periods it contains plus one (we denote the length of σ by σ^*). A label τ is a *simple* (*immediate*) extension of a label σ if $\tau = \sigma.n$ (and $n \in \boldsymbol{N}$). A label τ is an *extension* of a label σ if $\tau = \sigma.n_1.n_2.n_3.\ \ldots\ .n_k$ where $k \geq 1$ with each $n_i \geq 1$. We call ρ *root label* and always presuppose that $\rho = 1$. It follows that the set of labels is *strongly generated*. A tableau for a labelled formula P is a downward rooted tree, where each of the nodes contains a signed labelled formula, constructed using the branch extension rules to be defined below.

Non-discursive rules:

As D_2 contains the set of all disjunctio-negational theses of classical propositional logic the rules for disjunction and negation are identical to the ones used in classical case.

$$(\boldsymbol{T}\vee)\ \frac{\sigma :: T\,P \vee Q}{\sigma :: T\,P \mid \sigma :: T\,Q} \qquad (\boldsymbol{F}\vee)\ \frac{\sigma :: F\,P \vee Q}{\begin{array}{c}\sigma :: F\,P\\ \sigma :: F\,Q\end{array}}$$

$$(\boldsymbol{T}\sim)\ \frac{\sigma :: T \sim P}{\sigma :: F\,P} \qquad (\boldsymbol{F}\sim)\ \frac{\sigma :: F \sim P}{\sigma :: T\,P}$$

The rules $(F \vee)$, $(F \sim)$ and $(T \sim)$ are linear, but $(T \vee)$ is branching.

Discursive rules:

The rules in question are also divided into two types: linear and branching, but they do not imitate the classical rules in a simple way. The rules for discursive conjunction are the following:

$$\textbf{(}\boldsymbol{T}\wedge_{\mathbf{d}}\textbf{)}\quad \frac{\sigma :: T\ \mathrm{P} \wedge_d \mathrm{Q}}{\begin{array}{l}\sigma :: T\ \mathrm{P}\\ \sigma.\,n :: T\ \mathrm{Q}\end{array}} \qquad \textbf{(}\boldsymbol{F}\wedge_{\mathbf{d}}\textbf{)}\quad \frac{\sigma :: F\,\mathrm{P} \wedge_d \mathrm{Q}}{\sigma :: F\,\mathrm{P} \mid \sigma.\,n :: F\,\mathrm{Q}}$$

where for $(T\wedge_d)$ $\sigma.\,n$ is a label that is *new* to the branch and for $(F\wedge_d)$ $\sigma.\,n$ is a label that has been *already used* in the branch.

The rules for discursive implication are as follows:

$$\textbf{(}\boldsymbol{T}\rightarrow_{\mathbf{d}}\textbf{)}\quad \frac{\sigma :: \mathrm{P} \rightarrow_d \mathrm{Q}}{\sigma.\,n :: F\,\mathrm{P} \mid \sigma :: T\,\mathrm{Q}} \qquad \textbf{(}\boldsymbol{F}\rightarrow_{\mathbf{d}}\textbf{)}\quad \frac{\sigma :: F\,\mathrm{P} \rightarrow_d \mathrm{Q}}{\begin{array}{l}\sigma.\,n :: T\ \mathrm{P}\\ \sigma :: F\,\mathrm{Q}\end{array}}$$

where for $(T\rightarrow_d)$ $\sigma.\,n$ has been *already used* in the branch and for $(F\rightarrow_d)$ $\sigma.\,n$ is a label that is *new* to the branch.

Closure rule:

A branch of a tableau is closed if we can apply the rule:

$$\textbf{(C)}\quad \frac{\begin{array}{l}\sigma :: T\,\mathrm{P}\\ \sigma :: F\,\mathrm{P}\end{array}}{\text{closed}}$$

Otherwise the branch is open. A tableau is closed if all of its branches are closed, otherwise the tableau is open.

We return, for a while, to the initial Jaśkowski's ideas before introducing the last rule. Jaśkowski suggested treating a discussion as a set of opinions given by participants. There follows an idea to precede each opinion by the provision: *for a certain admissible meaning of the terms used*. It found expression in the definition (vi) and (Ab) – see above, and should also be present in our system. The idea is reflected in an additional rule.

Special rule:

$$\textbf{(S)}\quad \frac{\rho :: F\,\mathrm{P}}{\sigma.\,n :: F\ \mathrm{P}}$$

where ρ is a root label and $\sigma.\,n$ is a label that has been *already used* in the branch. The application of the rule is always limited to root labels.

Let P be a formula. By a *tableau proof* of P we mean a closed tableau with $1 :: F\,\mathrm{P}$.

Now, we give a few examples to illustrate how the rules we defined work.

Example 1. Closed tableau for the law of the excluded middle.

(a)	$1 :: F\, \mathrm{P} \vee \sim \mathrm{P}$	(start)
(b)	$1 :: F\, \mathrm{P}$	$(F \vee)$,(a)
(c)	$1 :: F \sim \mathrm{P}$	$(F \vee)$,(a)
(d)	$1 :: T\, \mathrm{P}$	$(F \sim)$,(c)
	closed	(C),(b),(d)

The notation $(F \vee)$,(a) indicates that the rule $(F \vee)$ was applied to the line (a).

Example 2. Closed tableau for the principle of contradiction.

(a)	$1 :: F \sim (\mathrm{P} \wedge_{\mathrm{d}} \sim \mathrm{P})$	(start)
(b)	$1 :: T\, \mathrm{P} \wedge_{\mathrm{d}} \sim \mathrm{P}$	$(F \sim)$,(a)
(c)	$1 :: T\, \mathrm{P}$	$(T \wedge_{\mathrm{d}})$,(b)
(d)	$1.1 :: T \sim \mathrm{P}$	$(T \wedge_{\mathrm{d}})$,(b)
(e)	$1.1 :: F\, \mathrm{P}$	$(T \sim)$,(d)
(f)	$1.1 :: F \sim (\mathrm{P} \wedge_{\mathrm{d}} \sim \mathrm{P})$	(S),(a)
(g)	$1.1 :: T\, \mathrm{P} \wedge_{\mathrm{d}} \sim \mathrm{P}$	$(F \sim)$,(f)
(h)	$1.1 :: T\, \mathrm{P}$	$(T \wedge_{\mathrm{d}})$,(g)
(i)	$1.1.1 :: T \sim \mathrm{P}$	$(T \wedge_{\mathrm{d}})$,(g)
	closed	(C),(e),(h)

Let us pay attention to the line (f) where we used the rule (S) to close the tableau.

Example 3. Closed tableau for the Clavius' law.

(a)	$1 :: F\, (\mathrm{P} \rightarrow_{\mathrm{d}} \sim \mathrm{P}) \rightarrow_{\mathrm{d}} \sim \mathrm{P}$	(start)
(b)	$1.1 :: T\ \mathrm{P} \rightarrow_{\mathrm{d}} \sim \mathrm{P}$	$(F \rightarrow_{\mathrm{d}})$,(a)
(c)	$1 :: F \sim \mathrm{P}$	$(F \rightarrow_{\mathrm{d}})$,(a)
(d)	$1 :: T\, \mathrm{P}$	$(F \sim)$,(c)
1st branch		
(e)	$1.1 :: F\, \mathrm{P}$	$(T \rightarrow_{\mathrm{d}})$,(b)
(f)	$1.1 :: F\, (\mathrm{P} \rightarrow_{\mathrm{d}} \sim \mathrm{P}) \rightarrow_{\mathrm{d}} \sim \mathrm{P}$	(S),(a)
(g)	$1.1.1 :: T\ \mathrm{P} \rightarrow_{\mathrm{d}} \sim \mathrm{P}$	$(F \rightarrow_{\mathrm{d}})$,(f)
(h)	$1.1\ F \sim \mathrm{P}$	$(F \rightarrow_{\mathrm{d}})$,(f)
(i)	$1.1 :: T\, \mathrm{P}$	$(F \sim)$,(h)
	Closed	(C),(e),(i)
2nd branch		
(e)'	$1.1 :: T \sim \mathrm{P}$	$(T \rightarrow_{\mathrm{d}})$,(b)
(f)'	$1.1 :: F\, \mathrm{P}$	(F ~),(e)'
(g)'	$1.1 :: F\, (\mathrm{P} \rightarrow_{\mathrm{d}} \sim \mathrm{P}) \rightarrow_{\mathrm{d}} \sim \mathrm{P}$	(S),(a)
(h)'	$1.1.2 :: T\, \mathrm{P} \rightarrow_{\mathrm{d}} \sim \mathrm{P}$	$(F \rightarrow_{\mathrm{d}})$,(g)'
(i)'	$1.1 :: F \sim \mathrm{P}$	$(F \rightarrow_{\mathrm{d}})$,(g)'
(j)'	$1.1 :: T\, \mathrm{P}$	(F ~),(i)'
	Closed	(C),(f)',(j)'

In our example, we applied one of the branching rules, i.e. $(T \rightarrow_{\mathrm{d}})$, to the line (b) and used the notions *1st branch* and *2nd branch* to indicate that the (*new*) branches were opened.

4. Simplified Tableaux for D_2

The accessibility relation we should define to capture the conditions on D_2-frames is reflexive, symmetric and transitive. It implies that any label is accessible from any other and we might well regard a label as a natural number instead of treating it as a (nonempty) sequence of positive integers separated by periods. It simplifies some of the technical details. The main merit of the modification consists in making tableau proofs less complicated. The modified rules take the form:

Discursive conjunction:

$$(\boldsymbol{T} \wedge_{\mathrm{d}})\quad \begin{array}{c} \sigma :: T\ \mathrm{P} \wedge_{\mathrm{d}} \mathrm{Q} \\ \hline \sigma :: T\ \mathrm{P} \\ \tau :: T\ \mathrm{Q} \\ (\text{for } \tau\ \textit{new}) \end{array} \qquad (\boldsymbol{F} \wedge_{\mathrm{d}})\quad \begin{array}{c} \sigma :: F\,\mathrm{P} \wedge_{\mathrm{d}} \mathrm{Q} \\ \hline \sigma :: F\,\mathrm{P} \mid \sigma' :: F\,\mathrm{Q} \\ (\text{for } \sigma'\ \textit{used}) \end{array}$$

Discursive implication:

$$(\boldsymbol{T} \rightarrow_{\mathrm{d}})\quad \begin{array}{c} \sigma :: \mathrm{P} \rightarrow_{\mathrm{d}} \mathrm{Q} \\ \hline \sigma' :: F\,\mathrm{P} \mid \sigma :: T\,\mathrm{Q} \\ (\text{for } \sigma'\ \textit{used}) \end{array} \qquad (\boldsymbol{F} \rightarrow_{\mathrm{d}})\quad \begin{array}{c} \sigma :: F\,\mathrm{P} \rightarrow_{\mathrm{d}} \mathrm{Q} \\ \hline \tau :: T\ \mathrm{P} \\ \sigma :: F\,\mathrm{Q} \\ (\text{for } \tau\ \textit{new}) \end{array}$$

Special rule:

$$(\mathbf{S})\quad \begin{array}{c} \rho :: F\,\mathrm{P} \\ \hline \sigma' :: F\ \mathrm{P} \\ (\text{for } \sigma'\ \textit{used}) \end{array}$$

Notice that we impose the same condition on the (S) rule as described in Section 3.

Here is an example of a tableau proof, in the modified system, of $(\sim\mathrm{P} \rightarrow_{\mathrm{d}} \mathrm{P}) \rightarrow_{\mathrm{d}} \mathrm{P}$.

Example 4. Closed tableau for the second Clavius' law.

(a)	$1 :: F\,(\sim\mathrm{P} \rightarrow_{\mathrm{d}} \mathrm{P}) \rightarrow_{\mathrm{d}} \mathrm{P}$	(start)
(b)	$2 :: T\ \sim\mathrm{P} \rightarrow_{\mathrm{d}} \mathrm{P}$	$(F \rightarrow_{\mathrm{d}})$,(a)
(c)	$1 :: F\ \mathrm{P}$	$(F \rightarrow_{\mathrm{d}})$,(a)
1st branch		
(d)	$1 :: F \sim\mathrm{P}$	$(T \rightarrow_{\mathrm{d}})$,(b)
(e)	$1 :: T\ \mathrm{P}$	(F ~),(d)
	Closed	(C),(c),(e)
2nd branch		
(d)'	$2 :: T\ \mathrm{P}$	$(T \rightarrow_{\mathrm{d}})$,(b)
(e)'	$2 :: F\,(\sim\mathrm{P} \rightarrow_{\mathrm{d}} \mathrm{P}) \rightarrow_{\mathrm{d}} \mathrm{P}$	(S),(a)
(f)'	$3 :: T \sim\mathrm{P} \rightarrow_{\mathrm{d}} \mathrm{P}$	$(F \rightarrow_{\mathrm{d}})$,(e)'
(g)'	$2 :: F\,\mathrm{P}$	$(F \rightarrow_{\mathrm{d}})$,(e)'
	Closed	(C),(d)',(g)'

In the next example, we will generate an infinite tableau for a "notorious" law of *CPC*.

Example 5. Infinite tableau for the Duns Scotus thesis

(a)	$1 :: F\, \mathrm{P} \rightarrow_d (\sim \mathrm{P} \rightarrow_d \mathrm{Q})$	(start)
(b)	$2 :: T\, \mathrm{P}$	$(F \rightarrow_d)$,(a)
(c)	$1 :: F \sim \mathrm{P} \rightarrow_d \mathrm{Q}$	$(F \rightarrow_d)$,(a)
(d)	$3 :: T \sim \mathrm{P}$	$(F \rightarrow_d)$,(c)
(e)	$1 :: F\, \mathrm{Q}$	$(F \rightarrow_d)$,(c)
(f)	$3 :: F\, \mathrm{P}$	$(T \sim)$,(d)
(g)	$2 :: F\, \mathrm{P} \rightarrow_d (\sim \mathrm{P} \rightarrow_d \mathrm{Q})$	(S),(a)
(h)	$4 :: T\, \mathrm{P}$	$(F \rightarrow_d)$,(g)
(i)	$2 :: F \sim \mathrm{P} \rightarrow_d \mathrm{Q}$	$(F \rightarrow_d)$,(g)
(j)	$5 :: T \sim \mathrm{P}$	$(F \rightarrow_d)$,(i)
(k)	$2 :: F\, \mathrm{Q}$	$(F \rightarrow_d)$,(i)
(l)	$5 :: F\, \mathrm{P}$	$(T \sim)$,(j)
(m)	$3 :: F\, \mathrm{P} \rightarrow_d (\sim \mathrm{P} \rightarrow_d \mathrm{Q})$	(S),(a)
(n)	$6 :: T\, \mathrm{P}$	$(F \rightarrow_d)$,(m)
(o)	$3 :: F \sim \mathrm{P} \rightarrow_d \mathrm{Q}$	$(F \rightarrow_d)$,(m)
(p)	$7 :: T \sim \mathrm{P}$	$(F \rightarrow_d)$,(o)
(r)	$3 :: F\, \mathrm{Q}$	$(F \rightarrow_d)$,(o)
(s)	$7 :: F\, \mathrm{P}$	$(T \sim)$,(p)
(t)	$4 :: F\, \mathrm{P} \rightarrow_d (\sim \mathrm{P} \rightarrow_d \mathrm{Q})$	(S),(a)
	$\vdots$	

The procedure goes on ad infinitum.

5. Unsigned Labelled Tableaux for D_2

In Sections 3 and 4, we introduced *Signed Labelled Tableaux for D_2*. We currently focus on a slightly different attitude to the problem in question and give a new set of tableau rules for the discursive logic. The main idea to get rid of the translation procedure is still playing a leading part, but we express it in a different way making some major changes in the formal language. We will work with *labelled formulas* such as σ :: P, where σ is a label (being viewed as a natural number) and P is a formula. The notation σ:: P intuitively means "P holds in world σ".

D_2-tableau is a tree of labelled formulas with root label ρ (we always assume that $\rho = 1$) and all the nodes of a tree are obtained by the rules schematically described in Table 1. A branch of D_2-tableau is *closed* if it contains $\perp$, otherwise it is open. A D_2-tableau is *closed* if all of the branches it contains are closed, otherwise it is open. By a *tableau proof* of P we mean a closed tableau with 1:: ~ P.

Observe that the rules $(\sim \vee)$, $(\sim \wedge_d)$ and $(\rightarrow_d)$ are branching just like $(T \vee)$, $(F \wedge_d)$ and $(T\rightarrow_d)$ given in Sections 3 and 4. The others are linear. We restrict the usage of the (S) rule to the root label ρ.

Classical rules:

$(\vee)$ $$\frac{\sigma :: P \vee Q}{\sigma :: P \;\mid\; \sigma :: Q}$$

$(\sim \vee)$ $$\frac{\sigma :: \sim(P \vee Q)}{\begin{array}{l}\sigma :: \sim P \\ \sigma :: \sim Q\end{array}}$$

$(\sim\sim)$ $$\frac{\sigma :: \sim\sim P}{\sigma :: P}$$

Discursive rules:

$(\wedge_d)$ $$\frac{\sigma :: P \wedge_d Q}{\begin{array}{l}\sigma :: P \\ \tau :: Q \quad \text{(for } \tau \textit{ new})\end{array}}$$

$(\sim \wedge_d)$ $$\frac{\sigma :: \sim(P \wedge_d Q)}{\sigma :: \sim P \;\mid\; \begin{array}{l}\sigma' :: \sim Q \\ \text{(for } \sigma' \textit{used})\end{array}}$$

$(\rightarrow_d)$ $$\frac{\sigma :: P \rightarrow_d Q}{\begin{array}{l}\sigma' :: \sim P \\ \text{(for } \sigma' \textit{used})\end{array} \;\mid\; \sigma :: Q}$$

$(\sim \rightarrow_d)$ $$\frac{\sigma :: \sim(P \rightarrow_d Q)}{\begin{array}{l}\tau :: P \\ \sigma :: \sim Q\end{array}} \quad \text{(for } \tau \textit{ new})$$

Closing rule:

(C) $$\frac{\begin{array}{l}\sigma :: P \\ \sigma :: \sim P\end{array}}{\perp}$$

Special rule:

(S) $$\frac{\rho :: \sim P}{\sigma' :: \sim P} \quad \begin{array}{l}\text{(for } \textit{root label } \rho) \\ \text{(for } \sigma' \textit{used})\end{array}$$

Table 1. Unsigned Labelled Tableaux for D_2

Here is an example of a tableau proof of $\sim\sim P \rightarrow_d P$.

Example 6. Closed tableau for the law of double negation.

(a)	$1 :: \sim(\sim\sim P \rightarrow_d \sim P)$	(start)
(b)	$2 :: \sim\sim P$	$(\sim \rightarrow_d)$,(a)
(c)	$1 :: \sim P$	$(\sim \rightarrow_d)$,(a)
(d)	$2 :: P$	$(\sim\sim)$,(b)
(e)	$2 :: \sim(\sim\sim P \rightarrow_d \sim P)$	(S),(a)
(f)	$3 :: \sim\sim P$	$(\sim \rightarrow_d)$,(e)
(g)	$2 :: \sim P$	$(\sim \rightarrow_d)$,(e)
(i)	$\perp$	(d),(g)

6. Conclusions

We have provided a method that allows to eliminate the translation procedure from D_2. The method we depicted is *friendly* to use. Since we concentrated our attention on the practical use of the system rather than its metalogical properties the fundamental theorems were not included in the present text. Instead, we gave some examples to illustrate how the method works in practice.

7. References

[1] J. Ciuciura, *History and Development of the Discursive Logic*, **Logica Trianguli** 3 (1999), pp. 3 - 31.

[2] J. Ciuciura, *Logika dyskusyjna*, **Principia** 35-36 (2003), (to appear).

[3] J. Ciuciura, *A New Axiomatization of D_2* (submitted to *BSL*)

[4] M. C. Fitting, *First-Order Logic and Automated Theorem Proving*, Springer, 1996 (first edition, 1990).

[5] R. Goré, *Tableau Methods for Modal and Temporal Logics*, in: M. D'Agostino, D. Gabbay, R. Haenle and J. Possegga (editors), **Handbook of Tableau Methods**, Kluwer Academic Publishers, Dordrecht-Boston-London, 1999, pp. 297- 396.

[6] S. Jaśkowski, *Propositional Calculus for Contradictory Deductive Systems*, **Studia Logica** 24 (1969), pp. 143 – 160.

[7] S. Jaśkowski, *O koniunkcji dyskusyjnej w rachunku zdań dla systemów dedukcyjnych sprzecznych*, **Studia Societatis Scientiarum Torunesis** 1, 8 (1949), pp. 171 – 172.

[8] J. Kotas, *Discussive Sentential Calculus of Jaśkowski*, **Studia Logica** 34/2 (1975), pp. 149 - 168.