— vordenker-archive —

Rudolf Kaehr

(1942-2016)

Title

Notes on the Tabularity of Polycontextural Logics

Bifunctoriality for transpositional and replicatorical Tableaux-Forest Calculi

Archive-Number / Categories

3_19 / K08

Publication Date

2012

Keywords / Topics

- MATRIX DESIGN FOR POLYCONTEXTURAL TABLEAUX LOGICS : Concept of matrix-distribution, Bifunctoriality and tabular notation, A wee 3x3-forest of tableaux_trees with bifunctoriality
- FROM A TABLEAUX TO A MATRIX PRESENTATION : Tree-Definition, Sketch of a matrix definition for forests of tableaux trees, Bivariat functors, Unary and binary indexed functors, Polyvariat functors, Matrix distribution, Forest-tableaux proofs, Bifunctoriality in action, Decision mechanisms, Polycontextural transformations, Abstractions: Matrix, super-operators, unification, tableaux, Extension mechanism for polycontextural logics, Metatheoretical laws for polycontextural logics, Trans-contextural duality principle
- REMARKS ON A MORE MATHEMATICAL FORMALIZATION : Pfalzgraf's fiber bundle approach: linear-indexed fibered bundles, Matrix formalization: tabular-indexed fibre bundles
- LOGIC PROGRAMMING IN THE fiELD : Polycontextural logical programming: The SCHELLING turn
- TABULAR DISSEMINATION OF KENOMIC CELLULAR AUTOMATA : Explicit tabular notation of distributed CAs

Disciplines

Cybernetics, Computer Sciences, Logic and Mathematics, Theory of Science

Abstract

Some more stuff towards an adequate formalization of polycontextural logics. The tabular tableaux_forest approach.

Citation Information / How to cite

Rudolf Kaehr: "Notes on the Tabularity of Polycontextural Logics", www.vordenker.de (Sommer Edition, 2017) J. Paul (Ed.), http://www.vordenker.de/rk/rk_Notes-on-Tabularity-of-Polycontextural-Logics_2012.pdf

Notes on the Tabularity of Polycontextural Logics

Bifunctoriality for transpositional and replicational Tableaux-Forest Calculi

Rudolf Kaehr Dr.phil@

Abstract

Some more stuff towards an adequate formalization of polycontextural logics. The tabular tableaux_forest approach. (Work in progress, v. 0.5, 25.July 2012)

1. Matrix Design for polycontextural Tableaux Logics

1.1. Concept of matrix-distribution

1.1.1. Bifunctoriality and tabular notation

Logic is easily connected with trees. Raymond Smullyan started the movement of "*Logic with Trees*" (Colin Howson), Melvin Fitting, the master of all trees dedicates his book *"First-order Logic and Automated Theorem Proving" "To Raymond Smullyan who brought me into the trees".*

The tree or tableaux method is highly elaborated by Melvin Fitting as the ultimate tableaux method, used today as a proof method for nearly all kinds of logical systems. There had been predecessors, as usual, like Evert Beth and Jaako Hintikka, or the Dialog Logic approaches of Paul Lorenzen and E. M. Barth.

Tree-thinking goes back to *the Porphyry of Tyre* with his *Porphyrian tree*. Tree-thinking is fundamental for Western thinking. Chinese thinking in contrast is based not on trees but on grids (Yang Hui (楊輝, c. 1238 - c. 1298)).
http://the-chinese-challenge.blogspot.co.uk/2007/03/chinese-centralism.html

The tableaux approach to logic seems to be very natural. Its emphasis is focussed on a structure with a singular beginning (root) and (mostly) binary decision procedures for the prolongations of the tree build on the base of such a root and its branching. The established hierarchy between the root and its nodes is perfectly stabilized by the success of its applications and its lucid rationality rooted in classical Western thinking of Porphyrian tree ontology and its re-invention in the Semantic Web, too.

It is believed, historically and actually, that non-rooted and non-hierarchical systems of thought and action are leading for short or long into chaos.

Postmodernist thinking believes that such arguments of and against hierarchical organizations are obsolete. Even the smallest kid experiences and knows how much we all are connected and taking part in massive networks where there is no beginning and no end and everything is nevertheless working fine. What's a correct impression for kids is not necessary the truth of the adult game.

With or without clouds, the internet connections are strictly hierarchically mathematized, programmed, organized, regulated, governed and policed.

The mass of data and "contents" are blinding the fact of the covered simple hierarchical form of organization of the deep-structure of the Web. Not just ICANN and the reduction to uni-directional communication but also the reduction of any sign system to techniques and ideologies of digitalism is determining the structural poverty of the overwhelming possibilities on an informational data-level.

For whatever reasons I never could find any enthusiasm for such an ultimate tree.

To stay in the context of the established form of rationality I prefer to live with/in forests instead of singular trees. I don't see any reason why a node might not change into a root and a root not becoming a node of a different, equally fundamental tree.

Traditional trees are not just defined by their uniqueness and hierarchy but by their definitive lack of interchangeability, chiasm or proemiality of the 'fundamental' terms, like nodes and root.

In fact, trees don't come in plural. All the singular and factual trees, say of logic, are dominated by the concept and methods of a single, unique and ultimate idea of a tree.

A first, and simple approach to surpass such limitations is proposed with the idea and some elaborations of forest-based polycontextural logics.

Hence, nothing is wrong with *"Logic with Trees"*. I opt to just disseminate such ultimate trees. This, as such and alone, wouldn't be specially interesting. What makes the forest approach interesting is the possibility of *interactions* between the plurality of such simultaneously existing ultimate trees. A forest is not the sum of singular trees but the interactivity between trees.

For the case of just one singular but ultimate tree we don't have to know much about the structure of the place it is planted. Because of its uniqueness, the knowledge of its ground(ing) can freely be omitted. For a forest, the loci of the trees becomes crucial. Disseminated trees are indexed to localize them in the grid of the ground. A ground and locus of a tree is not itself a tree. Hence, any logical characterization of the loci of the trees, that is building of a matrix and a grid, is obsolete. The matrix of the dissemination of logic-trees is defined by a a-logical or pre-logical structure. This pre-logical and pre-semiotic structure is covered by the methods of kenogrammatics. Thus, the grid of the forest is the kenomic matrix.

Again, the game starts again. There is no necessity to suppose a static hierarchy between the grid and the forests.

Trees in formal languages are reduced to the simple structure of "*append*" and "*remove*" of "*items*". Hence, disseminated trees are indexed, in this case, double-indexed to define a matrix of trees, and are defined by the similarly simple operations of "*leave*" a tree, 'horizontally' for replications (reflection) and "*leave*" a tree vertically for transpositions (transjunctions).

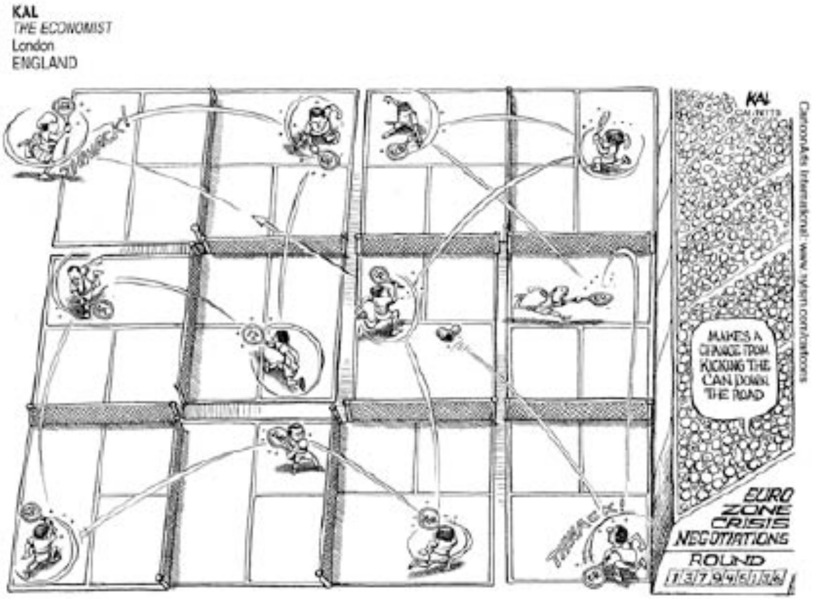
Other operations between trees, like *permutation*, *reduction* and *iteration* of trees of a forest, are easily introduced and implemented into the formal game of forest-logics. Forests are not static. They might grow or shrink and change their patterns.

From a more mathematical point of view, forests and their interplays are well ruled by the polycontextural concept of interchangeability, i.e. a generalization and subversion of the category-theoretic concept of bifunctoriality.

Without any big deviations and dangerous revolutions a move from the tree-culture to a forest-world of thinking and acting seems to be a fairly save and sane step of evolution even for the timid Western searcher of truth and computational efficiency.

In earlier papers about *tree-farming* I proposed contextural forests as forests of *colored* trees. This time, coloring has to wait for the paint.

Tabular complexity in action


KAL
THE ECONOMIST
London
ENGLAND
THWACK!
KAL
Cartoonists International www.nytsyn.com/cartoons
MAKES A CHANGE FROM KICKING THE CAN DOWN THE ROAD
THWACK!
EURO ZONE CRISIS NEGOTIATIONS
ROUND

History and sketches of the tabular approach to polycontexturality
http://www.thinkartlab.com/pkl/lola/PolyLogics.pdf
http://www.thinkartlab.com/pkl/lola/AFOSR-Place-Valued-Logic.pdf
http://www.thinkartlab.com/pkl/lola/ConTeXtures.pdf
http://www.thinkartlab.com/pkl/media/SKIZZE-0.9.5-medium.pdf

1.1.2. Why Smullyan's analytical tree method?

"The tree method *is one of several decision procedures available for classical propositionai logic and first-order functional calculus and for nonclassical logics, inducing intuitionistic propositional logic and intuitionistic first-order logic, modal logics, and multiple-valued logics The tree method provides exceptionally elegant proofs of the consistency and satisfiability of formulae.*

Falsifiability trees allow easy testing of the validity of proofs and are a canonization of proof by contradiction for natural deduction systems, while truth trees allow easy derivation of theorems in these systems. Tree proofs permit graphical-geometric representations of logical relations, and appear to be of greater intuitive accessibility than either the axiomatic method or the method of natural deduction.

The proofs of the completeness and soundness of the tree method and its variants are also straightforward, and the method combines insights and results of model theory and proof theory in a fashion that clearly identifies the most basic concepts of proof involving such model-theoretic results as Craig's Interpolation Lemma, Beth's Definability Theorem, and Robinson's Consistency Theorem." (Irving H. Anellis)
http://projecteuclid.org/DPubS?service=UI&version=1.0&verb=Display&handle=euclid.rml/1204834539

1.1.3. A wee 3x3-forest of tableaux_trees with bifunctoriality

H5 = (X laa Y) iij (X laa Y)

Tableaux – forests for " laa " and " iij "

<table>
<tr><th>iij</th><th>locus1</th><th>locus2</th><th>locus3</th></tr>
<tr><td>va1</td><td>[t1 .1 : (f1 : X || t1 : Y)]
[f1 .1 : (t1 : X && f1 : Y)]</td><td>–</td><td>–</td></tr>
<tr><td>val2</td><td>[t1 .2 : (f2 : X || t2 : Y)]
[f1 .2 : (t2 : X && f2 : Y)</td><td>–</td><td>–</td></tr>
<tr><td>val3</td><td>–</td><td>–</td><td>[t3 .3 : (f3 : X || t3 : Y)]
[f3 .3 : (t3 : X && f3 : Y)]</td></tr>
</table>

<table>
<tr><th>laa</th><th>locus1</th><th>locus2</th><th>locus3</th></tr>
<tr><td>va1</td><td>[t1 .1 : (t1 : X && t1 : Y) || (t1 : X && f1 : Y)]
[f1 .1 : (f1 : X && f1 : Y)]</td><td>[t2 .1 : (f1 : X && f1 : Y)]
[f2 .1 : (f1 : X && t1 : Y)]</td><td>[t3 .1 : (t1 : X && t1 : Y)]
[f3 .1 : (f1 : X && t1 : Y)]</td></tr>
<tr><td>val2</td><td>–</td><td>[t2 .2 : (t2 : X && t2 : Y)]
[f2 .2 : (f2 : X || f2 : Y)]</td><td>–</td></tr>
<tr><td>val3</td><td>–</td><td>–</td><td>[t3 .3 : (t3 : X || t3 : Y)]
[f3 .3 : (f3 : X && f3 : Y)]</td></tr>
</table>

$$\textbf{Syntax}(\textbf{H5}) = \begin{pmatrix} (X\,l\,Y)\,i\,(X\,l\,Y) & - & - \\ (X\,a\,Y)\,j\,(X\,a\,Y) & - & - \\ - & - & (X\,a\,Y)\,j\,(X\,a\,Y) \end{pmatrix} = \begin{pmatrix} (X\,l\,Y)\,i\,(X\,l\,Y) & (X\,l'\,Y) & - \\ & (X\,l'\,Y) & \\ (X\,a\,Y)\,i\,(X\,a\,Y) & - & - \\ - & - & (X\,a\,Y)\,j\,(X\,a\,Y) \end{pmatrix}$$

Semantic scheme for f1H5

Semantic scheme for f3H5

$$\begin{pmatrix} \boxed{\begin{array}{l} \text{f1 .1 :} \\ \text{t1 .1 :} (X \, l \, Y) \\ \text{f1 .1 :} (X \, l \, Y) \\ \text{stop}(1.1) \end{array}} & \boxed{\begin{array}{l} (1.2) \\ \text{f1 .2} \, Y \\ \text{t1 .2} \, Y \\ \text{stop}(1.2) \end{array}} & [\,] \\ \boxed{\begin{array}{l} \text{f2 .1 :} \\ \text{t2 .1 :} (X \, l \, Y) \Longrightarrow (1.2) \\ \text{f2 .1 :} (X \, l \, Y) \Longrightarrow (1.2) \\ \text{stop}(2.1) \end{array}} & [\,] & [\,] \\ [\,] & [\,] & [\,] \end{pmatrix} \begin{pmatrix} [\,] & [\,] & [\,] \\ [\,] & [\,] & [\,] \\ \boxed{\begin{array}{l} (3.1) \\ \text{f3 .1} \, Y \\ \text{t3 .1} \, Y \\ \text{stop}(3.1) \end{array}} & [\,] & \boxed{\begin{array}{l} \text{t3 .3 :} \\ \text{t3 .3 :} (X \, l \, Y) \Longrightarrow (3.1) \\ \text{f3 .3 :} (X \, l \, Y) \Longrightarrow (3.1) \\ \text{stop}(3.3) \end{array}} \end{pmatrix}$$

Closing semantic tableaux_forest for f1H5

Null

f1H5	locus1	locus2	locus3
val1	(1) t1 .1 : (X laa Y) (0) (2) f1 .1 : (X laa Y) (0) (3) t1 .1 : X ‖ t1 .1 : X (1) (4) t1 .1 : Y ‖ f1 .1 : Y (1) (4) f1 .1 : X (2) (5) f1 .1 : Y (2) (6) xx = stop(1.1)	f1 .2 : X (2.1) f1 .2 : Y (2.1) f1 .2 : X (2.2) t1 .2 : Y (2.2) xx = stop(1.2)	[]
val2	(1) t2 .1 : (X laa Y) (0) (2) f2 .1 : (X laa Y) (0) (3) t2 .1 : X (1) (4) t2 .1 : Y (1) (4) f2 .1 : X ‖ f2 .1 : Y (2) (6) xx ‖ xx = stop(2.1)	[]	[]
val3	[]	[]	[]

Closing semantic tableaux_

f3H5	locus1	locus2
val1	[]	[]
val2	[]	[]
val3	t3 .1 : X (1.3) t3 .1 : Y (1.3) f3 .1 : X (2.3) t3 .1 : Y (2.3) xx = stop(3.1)	[]

Bifunctoriality of H5

$$\begin{array}{lcl} (1) \ \text{t2 .1 :} (X \, \text{laa} \, Y) \ (0) & & \text{f1 .2 :} X \ (2.1), \ \text{f1 .2 :} Y \ (2.1) \\ \qquad \text{et} & = & \qquad \text{simul} \\ (2) \ \text{f2 .1 :} (X \, \text{laa} \, Y) (0) & & \text{f1 .2 :} X \ (2.2), \ \text{t1 .2 :} Y \ (2.2) \end{array}$$

General scheme

$$(\mathrm{sf1}\ \&\&\ \mathrm{sf2}\ \&\&\ \mathrm{sf3}\ \&\&\ \mathrm{sf4}) \mathbin{//} \langle (\mathrm{t1}\ \&\&\ \mathrm{t1}') \ldots (\mathrm{tn}\ \&\&\ \mathrm{tn}') \rangle$$

Bifunctorial Rule 1

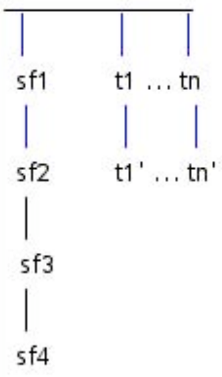
Bifunctoriality of && and //

$$(t \mathbin{//} \mathrm{ta})\ \&\&\ (t' \mathbin{//} \mathrm{ta}') \Longrightarrow (t\ \&\&\ t') \mathbin{//} (\mathrm{ta}\ \&\&\ \mathrm{ta}')$$

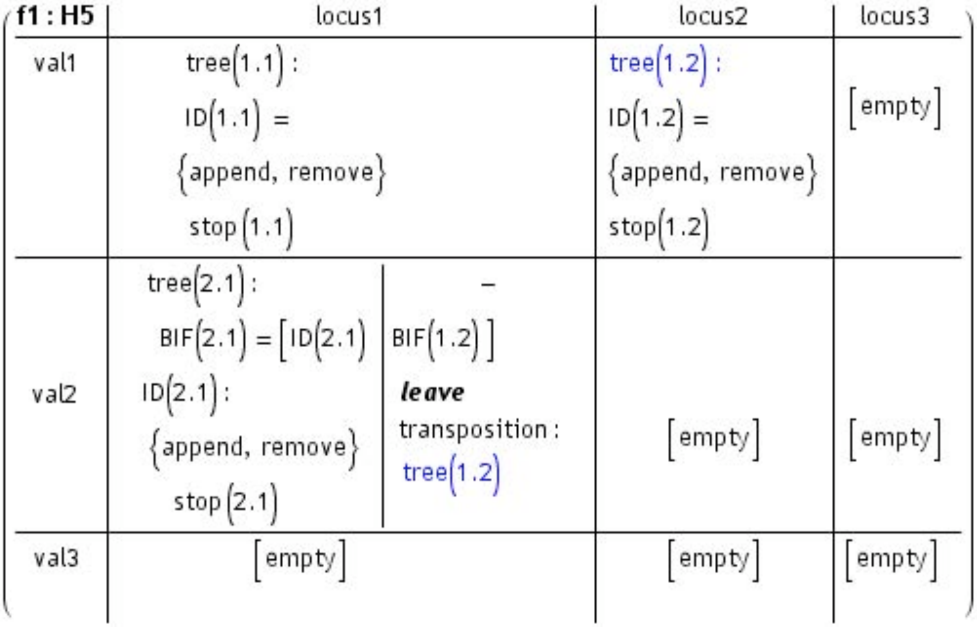
$$\frac{(\mathrm{sf1} \mathbin{//} \mathrm{t1} \ldots \mathrm{tn})\ \&\&\ (\mathrm{sf2} \mathbin{//} \mathrm{t1}' \ldots \mathrm{tn}')}{(\mathrm{sf1}\ \&\&\ \mathrm{sf2}) \mathbin{//} (\mathrm{t1} \ldots \mathrm{tn})\ \&\&\ (\mathrm{t1}' \ldots \mathrm{tn}')}$$

&& : junction,
// : transfunction

Diagram



Formal characterization for f1H5 :



f1 : H5	locus1		locus2	locus3

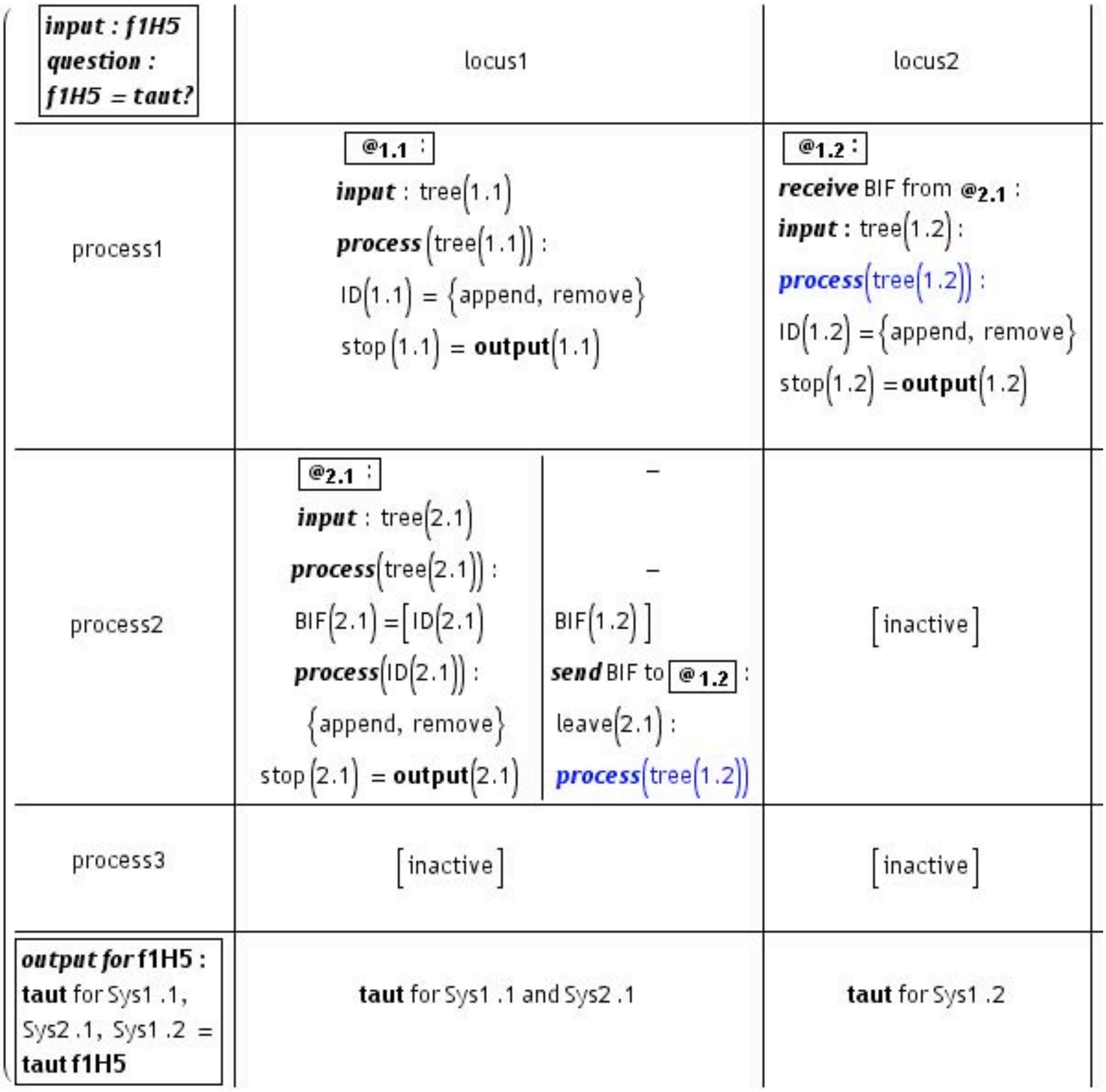
val1	tree(1.1) :
	ID(1.1) =
	{append, remove}
	stop (1.1)
			tree(1.2) :
			ID(1.2) =
			{append, remove}
			stop(1.2)
				[empty]

val2	tree(2.1) :
	BIF(2.1) = [ID(2.1)
	ID(2.1) :
	{append, remove}
	stop (2.1)
		–
		BIF(1.2)]
		leave
		transposition :
		tree(1.2)
			[empty]	[empty]

val3	[empty]		[empty]	[empty]

Multi – processor model for f1H5 :

Multi-Processor-System for matrix-distribution of tableaux_forests = (intra-process: {append, remove, leave}, inter-process: {send, receive}).



input : f1H5 *question :* *f1H5 = taut?*	locus1		locus2

process1	@1.1 :
	input : tree(1.1)
	process(tree(1.1)) :
	ID(1.1) = {append, remove}
	stop(1.1) = **output**(1.1)
			@1.2 :
			receive BIF from @2.1 :
			input : tree(1.2) :
			process(tree(1.2)) :
			ID(1.2) = {append, remove}
			stop(1.2) = **output**(1.2)

process2	@2.1 :
	input : tree(2.1)
	process(tree(2.1)) :
	BIF(2.1) = [ID(2.1)
	process(ID(2.1)) :
	{append, remove}
	stop(2.1) = **output**(2.1)
		–
		–
		BIF(1.2)]
		send BIF to @1.2 :
		leave(2.1) :
		process(tree(1.2))
			[inactive]

process3	[inactive]		[inactive]
***output* for f1H5 :** **taut** for Sys1 .1, Sys2 .1, Sys1 .2 = **taut f1H5**	**taut** for Sys1 .1 and Sys2 .1		**taut** for Sys1 .2

2. From a tableaux to a matrix presentation

2.1. TreeDefinition

2.1.1. Fitting Smullyan trees

TreeDefinition in Prolog by Fitting, p. 156.

```
/* member(Item, List) := Item occurs in List
*/
member(X, [X |  _]).
member(X, [_ | Tail]) :- member(X, Tail).

/* remove(Item, List, Newlist) :-
        Newlist is the result of removing all occurrences of Item from List.
*/
remove(X, [ ], [ ]).
remove(X, [Y | Tail], Newtail) :-
     X == Y
remove(X, [Y | Tail], [Y | Newtail]) :-
     X \= Y
remove(X, Tail, Newtail).
```

```
/*  append(ListA, ListB, Newlist) :-
        Newlist is the result of appending ListA and ListB.
*/
append([ ], List, List).
append([Head | Tail], [Head | Newlist]) :-
   append(Tail, List, Newlist).
```

Conceptual structure of Fitting's TreeDefinition
Mono-Contexturality: Prolog, Logic.
TreeDefinition = (Operators, Elements):
Elements = {Item, List, Newlist; Head, Tail, Newtail, ListA, ListB} with
operators= {member, remove, append}.

Forests of Trees
Distribution of TreeDefiniton.
Method: Indexed categories (Pfalzgraf)
Junctional logical operations.
Transjunctions as semantically indexed mono-contextural mappings.

Mediated Forests of Trees
Weak polycontexturality. Category theory.
Distribution and mediation of TreeDefiniton.
Method: Single-indexed categories (Pfalzgraf) with proemial interchangeability and chiasms.
Junctional and transjunctional logical operations.
LOLA: lists of mediated tableaux trees.

Matrix of Trees
Polycontexturality, Polycontextural Logics, Morphogrammatics.
Interactionally and reflectionally distributed and mediation of TreeDefinition.
Double-indexed categories and processors.

MAT = {{term, list, append, remove} ∪ {replication, transposition}, N}.
replication = leaveV, (vertical)
transposition = leaveH, (horizontal).

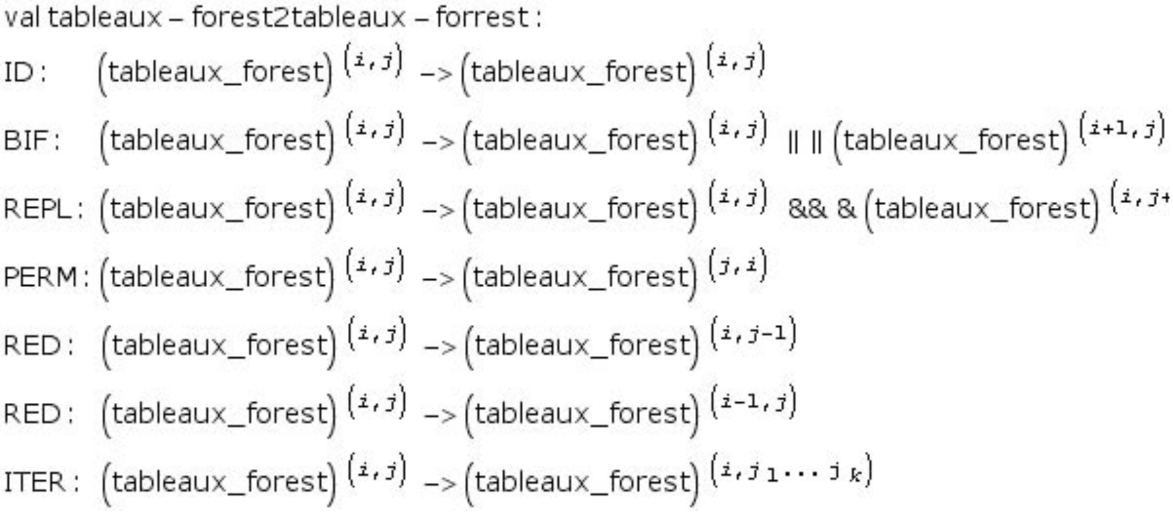
2.1.2. TreeDefinition of TabDefSig in SML by LOLA1993

```
(*$TabDefSig*)
signature TABDEF =
   sig
   type var
   type truval
   type operator
   type formula
   type tableau_tree
   exception TabDefError of string
   val fmls2fmls:  (formula list) -> (formula list)
   val defs2rules: (operator * (int * (var list) * formula)) list  -> unit
   val tabs2rules: (operator * (int * (var list) * ((truval*tableau_tree) list))) list  -> unit
   end;
```

http://www.thinkartlab.com/pkl/lola/LOLA.pdf

2.1.3. Sketch of a matrix definition for forests of tableaux trees

```
(*$MatDefSig*)
signature MATRIXDEF =
   sig
        type matrix
        type matrix_super-operators
        type tableaux_forest
  sig
      tableaux-forest
      val tab2tab: (tableaux_tree list) --> (tableaux_tree list)
```



val tableaux – forest2tableaux – forrest :

ID : $(\text{tableaux\_forest})^{(i,j)}$ -> $(\text{tableaux\_forest})^{(i,j)}$

BIF : $(\text{tableaux\_forest})^{(i,j)}$ -> $(\text{tableaux\_forest})^{(i,j)}$ || || $(\text{tableaux\_forest})^{(i+1,j)}$

REPL : $(\text{tableaux\_forest})^{(i,j)}$ -> $(\text{tableaux\_forest})^{(i,j)}$ && & $(\text{tableaux\_forest})^{(i,j+}$

PERM : $(\text{tableaux\_forest})^{(i,j)}$ -> $(\text{tableaux\_forest})^{(j,i)}$

RED : $(\text{tableaux\_forest})^{(i,j)}$ -> $(\text{tableaux\_forest})^{(i,j-1)}$

RED : $(\text{tableaux\_forest})^{(i,j)}$ -> $(\text{tableaux\_forest})^{(i-1,j)}$

ITER : $(\text{tableaux\_forest})^{(i,j)}$ -> $(\text{tableaux\_forest})^{(i,j_1 \cdots j_k)}$

```
signature TABDEF(i,j) =
   sig
        type var
   type truval
   type operator
   type formula
   type tableau_tree
        type tableaux_forrest

   exception TabDefError of string
   val fmls2fmls:  (formula list) -> (formula list)
   val defs2rules: (operator * (int  *(var list) * formula)) mat  -> unit
   val tabs2rules: (operator * (int  *(var list) * ((truval*tableau_tree) list))) list  -> unit
   end;
signature TAB2MATRDEF =
```

2.2. Bivariat functors

2.2.1. Unary and binary indexed functors

Unary indexed functors for LOLA1993

The unary indexed approach to polycontextural tableaux trees takes the distribution over different places into account. But it is not yet considering the matrix distribution necessary for an understanding of the full polycontextural approach with transpositions and replications defining a 2-dimensional grid.

Infixes

||: disjunctive
&&: conjunctive
///: transjunctive branching.

Operators

o: disjunction,
a: conjunction,
i, j: implication,
t: transjunction.

Examples

```
oao :   [t1:  t1:X || t1:Y ]
        [f1:  f1:X && f1:Y ]
        [t2:  t2:X && t2:Y ]
        [f2:  f2:X || f2:Y ]
        [t3:  t3:X && t3:Y ]
        [f3:  f3:X || f3:Y ]

ijj :  [t1:  f1:X || t1:Y]
       [f1:  t1:X && f1:Y]
       [t3:  (f2:X || t2:Y) /// (f3:X || t3:Y)]
       [f3:  (t2:X && f2:Y) /// (t3:X && f3:Y)]
       [t2:  f2:X && t2:X]
```

[f2: f2:X && t2:X]

oto : [t1: t1:X && t1:Y /// (f2:X && t2:Y)||(t2:X && f2:Y)]
[f1: f1:X && f1:Y /// (t2:X && t2:Y)]
[t2: t2:X && t2:Y]
[f2: f2:X && f2:Y]
[t3: t3:X || t3:Y /// (f2:X && t2:Y)||(t2:X && f2:Y)]
[f3: f3:X && f3:Y /// (f2:X && f2:Y)]

http://www.thinkartlab.com/pkl/lola/VERSIONT/DATA/TABLEAU.RUL

Double-indexed functors for MATRIX-LOLA

The double-indexed functors are distibuted and mediated over the kenomic matrix. Its 2-dimensionality enables, together with permutations, reductions and iterations, a proper managment of transjunctional and replicational functors.

Junctional distribution

oao : [t1.1: t1.1:X || t1.1:Y]
[f1.1: f1.1:X && f1.1:Y]
[t2.2: t2.2:X && t2.2:Y]
[f2.2: f2.2:X || f2.2:Y]
[t3.3: t3.3:X && t3.3:Y]
[f3.3: f3.3:X || f3.3:Y]

Replicational distribution

ijj : [t1.1: f1.1:X || t1.1:Y]
[f1.1: t1.1:X && f1.1:Y]
[t3.3: (f1.2:X || t1.2:Y)\\\ (f3.3:X || t3.3:Y)]
[f3.3: (t1.2:X && f1.2:Y)\\\ (t3.3:X && f3.3:Y)]
[t2.2: f2.2:X && t2.2:X]
[f2.2: f2.2:X && t2.2:X] =[]

iij: [t1.1: (f1:X || t1:Y) /// (f1.2:X || t1.2:Y)]
[f1.1: (t1:X && f1:Y) /// (t1.2:X && f1.2:Y)]
[t3.3: f1.3:X || t1.3:Y]
[f3.3: t1.3:X && f1.3:Y]
[t2: f2.2:X && t2.2:X] =[]
[f2: f2.2:X && t2.2:X]

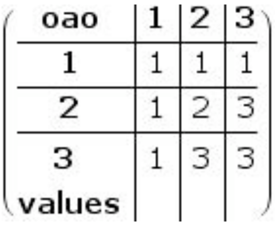
Transjunctional distribution

oto : [t1.1: t1.1:X && t1.1:Y /// (f2.1:X && t2.1:Y)||(t2.1:X && f2.1:Y)]
[f1.1: f1.1:X && f1.1:Y /// (t2.1:X && t2.1:Y)]
[t2.2: t2.2:X && t2.2:Y]
[f2.2: f2.2:X && f2.2:Y]
[t3.3: t3.3:X || t3.3:Y /// (f2.1:X && t2.1:Y)||(t2.1:X && f2.1:Y)]
[f3.3: f3.3:X && f3.3:Y /// (f2.1:X && f2.1:Y)]

2.2.2. Distribution tables

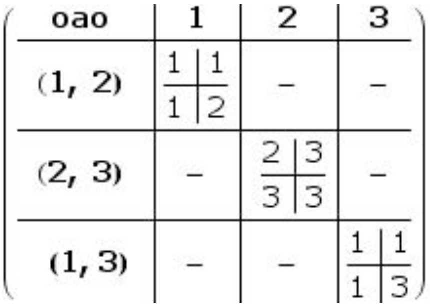
Total functions: F_{tot}: (i, j) --> (i, j)

F_{oao} :
$o_{1.1}$: (1, 2) --> (1, 2;1)
$a_{2.2}$: (2, 3) --> (2, 3;2)
$o_{3.3}$: (1, 3) --> (1, 3;3).



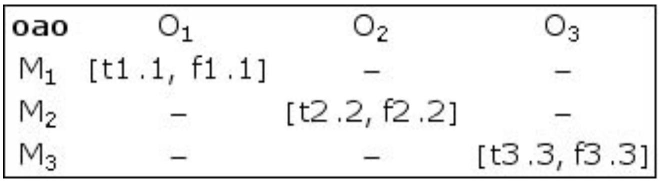
oao	1	2	3
1	1	1	1
2	1	2	3
3	1	3	3
values			

=>



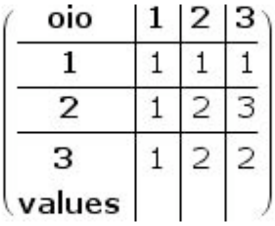
oao	1	2	3
(1, 2)	1 | 1 1 | 2	–	–
(2, 3)	–	2 | 3 3 | 3	–
(1, 3)	–	–	1 | 1 1 | 3

:



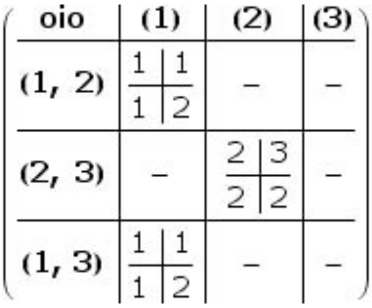
oao	O_1	O_2	O_3
M_1	[t1 .1, f1 .1]	–	–
M_2	–	[t2 .2, f2 .2]	–
M_3	–	–	[t3 .3, f3 .3]

F_{oio} :
$o_{1.1}$: (1, 2) --> (1, 2;1)
$a_{2.2}$: (2, 3) --> (2, 3;2)
$o_{3.3}$: (1, 3) --> (1, 3;1).



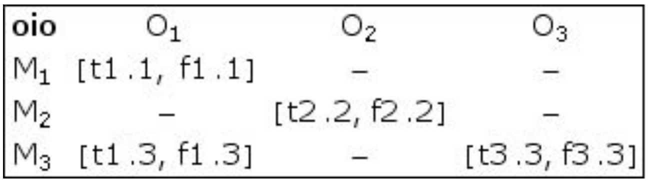
oio	1	2	3
1	1	1	1
2	1	2	3
3	1	2	2
values			

=>



oio	(1)	(2)	(3)
(1, 2)	$\begin{array}{c\|c}1&1\\\hline 1&2\end{array}$	–	–
(2, 3)	–	$\begin{array}{c\|c}2&3\\\hline 2&2\end{array}$	–
(1, 3)	$\begin{array}{c\|c}1&1\\\hline 1&2\end{array}$	–	–

:

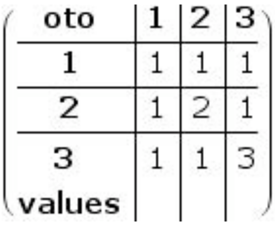


oio	O_1	O_2	O_3
M_1	[t1 .1, f1 .1]	–	–
M_2	–	[t2 .2, f2 .2]	–
M_3	[t1 .3, f1 .3]	–	[t3 .3, f3 .3]

Partial functions F_{part}

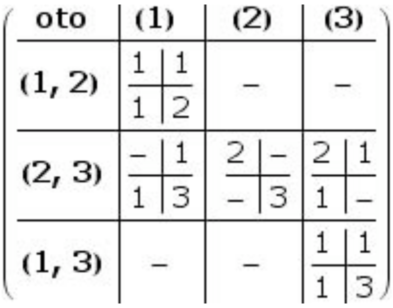
Transjunctions: F_{trans}: (i, j) --> ((i, j_1), (i,j_2),..., (i,j_n)

F_{oto}:
$o_{1.1}$: (1, 2) --> (1,2;1)
$t_{2.2}$: (2, 3) --> (2, 3;1), (2, 3;2), (2, 3;3)),
$o_{3.3}$: (1, 3) --> (1, 3;3)



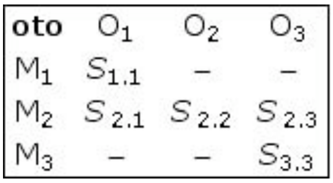
oto	1	2	3
1	1	1	1
2	1	2	1
3	1	1	3
values			

=>



oto	(1)	(2)	(3)
(1, 2)	$\begin{array}{c\|c}1&1\\\hline 1&2\end{array}$	–	–
(2, 3)	$\begin{array}{c\|c}–&1\\\hline 1&3\end{array}$	$\begin{array}{c\|c}2&–\\\hline –&3\end{array}$	$\begin{array}{c\|c}2&1\\\hline 1&–\end{array}$
(1, 3)	–	–	$\begin{array}{c\|c}1&1\\\hline 1&3\end{array}$

:



oto	O_1	O_2	O_3
M_1	$S_{1.1}$	–	–
M_2	$S_{2.1}$	$S_{2.2}$	$S_{2.3}$
M_3	–	–	$S_{3.3}$

Mediation scheme

$$\begin{tikzcd}
t3.3 \equiv t1.1 \arrow[r] \arrow[d, leftrightarrow] & f1.1 \arrow[d, leftrightarrow] \\
f3.3 \equiv f2.2 & t2.2 \arrow[l]
\end{tikzcd}$$

F_{iij} :
$i_{1.1}$: (1, 2) --> (1, 2;1)
$i_{2.2}$: (2, 3) --> (1, 3;1)
$j_{3.3}$: (1, 3) --> (1, 3;3).

$$\left(\begin{array}{c|c|c|c} \textbf{ijj} & 1 & 2 & 3 \\ \hline 1 & 1 & 2 & 3 \\ \hline 2 & 1 & 1 & 3 \\ \hline 3 & 1 & 1 & 1 \\ \text{values} & & & \end{array}\right) \Rightarrow \left(\begin{array}{c|c|c|c} \textbf{ijj} & 1 & 2 & 3 \\ \hline (1,\ 2) & \begin{array}{c|c}1&2\\\hline 1&1\end{array} & - & - \\ \hline (2,\ 3) & - & - & - \\ \hline (1,\ 3) & \begin{array}{c|c}1&3\\\hline 1&1\end{array} & - & \begin{array}{c|c}1&3\\\hline 1&1\end{array} \end{array}\right)$$

, **ijj** = $i_{1.1}\ i_{1.3}\ i_{3.3}$

PM(ijj)	O_1	O_2	O_3
M_1	$S_{1.1}$	–	–
M_2	–	–	–
M_3	$S_{1.3}$	–	$S_{3.3}$

PM(ijj)	O_1	O_2	O_3
M_1	[t1 .1, f1 .1]	–	–
M_2	–	–	–
M_3	[t1 .3, f1 .3]	–	[t3 .3, f3 .3]

PM/n1(ijj)	O_1	O_2	O_3
M_1	$S_{1.1}$	–	–
M_2	–	$S_{2.2}$	–
M_3	–	$S_{2.3}$	–

PM/n1(ijj)	O_1	O_2	O_3
M_1	[f1 .1, t1 .1]	–	–
M_2	–	[t2 .2, f2 .2]	–
M_3	–	[t2 .3, f2 .3]	–

neg1(ijj):

[t1.1, f1.1] = [t1,f1]:pos(1,1)	[f1.1, t1.1] = [t1, f1]:pos(1,1)
[t1.3, f1.3] = [t1,f1]:pos(1,3)	[t2 .2, f2 .2] = [t2, f2]:pos(2,2)
[t3.3, f3.3] = [t3,f3]:pos(3,3)	[t2 .3, f2 .3] = [t2, f2]:pos(2,3)

$$\left(\begin{array}{c|c|c|c} \textbf{iij} & 1 & 2 & 3 \\ \hline 1 & 1 & 2 & 3 \\ \hline 2 & 1 & 1 & 2 \\ \hline 3 & 1 & 1 & 1 \\ \text{values} & & & \end{array}\right) \Rightarrow \left(\begin{array}{c|c|c|c} \textbf{iij} & 1 & 2 & 3 \\ \hline (1,\ 2) & \begin{array}{c|c}1&2\\\hline 1&1\end{array} & - & - \\ \hline (2,\ 3) & \begin{array}{c|c}1&2\\\hline 1&1\end{array} & - & - \\ \hline (1,\ 3) & - & - & \begin{array}{c|c}1&3\\\hline 1&1\end{array} \end{array}\right)$$

, **iij** = $i_{1.1}\ i_{1.2}\ i_{3.3}$

iij	O_1	O_2	O_3
M_1	$S_{1.1}$	–	–
M_2	$S_{1.2}$	–	–
M_3	–	–	$S_{3.3}$

⇒

iij	O_1	O_2	O_3
M_1	[t1 .1, f1 .1]	–	–
M_2	[t1 .2, f1 .2]	–	–
M_3	–	–	[t3 .3, f3 .3]

$$\left(\begin{array}{c|c|c|c} \mathbf{ii}\,\boldsymbol{i'}\,\boldsymbol{i''}\,\boldsymbol{j} & 1 & 2 & 3 \\ \hline 1 & 1 & 2 & 3 \\ \hline 2 & 1 & 1 & 2.2.2 \\ \hline 3 & 1 & 1 & 1 \\ \text{values} & & & \end{array}\right) \Rightarrow \left(\begin{array}{c|c|c|c} \textbf{iij} & 1 & 2 & 3 \\ \hline (1,\ 2) & \begin{array}{c|c}1&2\\\hline 1&1\end{array} & - & - \\ \hline (2,\ 3) & \begin{array}{c|c}1&2\\\hline 1&1\end{array} \\ \begin{array}{c|c}1&2\\\hline 1&1\end{array} \\ \begin{array}{c|c}1&2\\\hline 1&1\end{array} & - & - \\ \hline (1,\ 3) & - & - & \begin{array}{c|c}1&3\\\hline 1&1\end{array} \end{array}\right) :$$

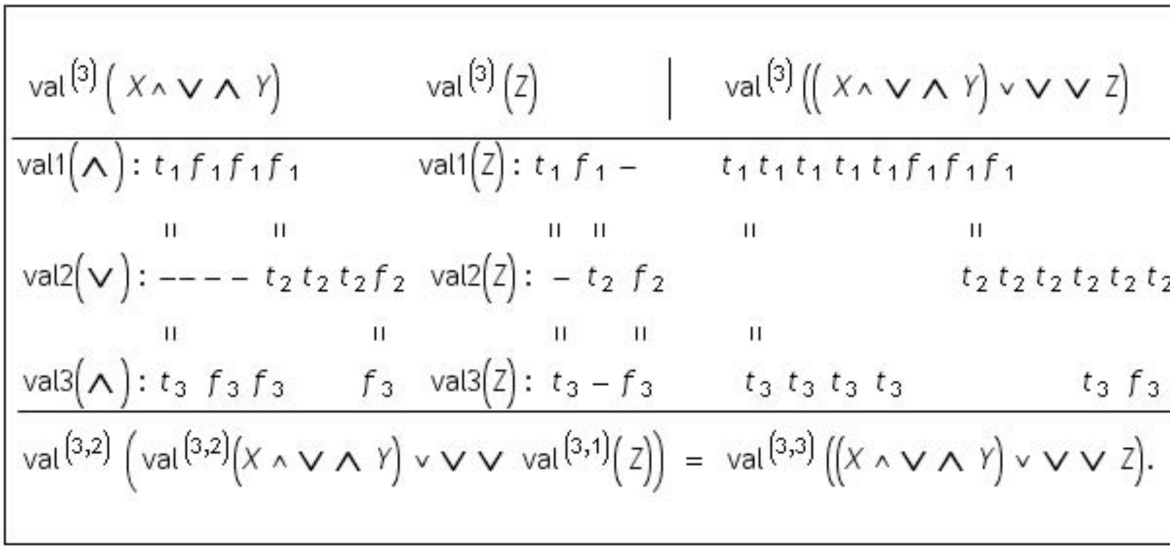
PM	O_1	O_2	O_3
M_1	$S_{1.1}$	–	–
M_2	$S_{1.2.2.2}$	–	–
M_3	–	–	$S_{3.3}$

2.2.3. Polyvariat functors

Decomposibility of m-valued n-ary functions into its two-valued parts is disturbed by the fact that an extensional m-valued n-ary function receives more different values as necessary for a decomposition into binary states. Hence, a 3-valued 3-ary function with $\{1,2,3\}^3$ gets constellations like (1,2,3) that are not as such decomposable as such into 2-valued parts.

val((X o Y) o Z) != val(val(X o Y), Z))

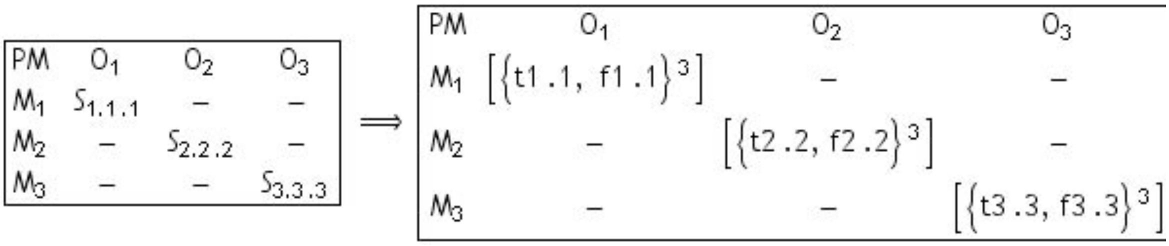
Example



$\mathrm{val}^{(3)}(X \wedge \vee \wedge Y)$	$\mathrm{val}^{(3)}(Z)$	$\mathrm{val}^{(3)}((X \wedge \vee \wedge Y) \vee \vee \vee Z)$
$\mathrm{val1}(\wedge): t_1\, f_1\, f_1\, f_1$	$\mathrm{val1}(Z): t_1\, f_1\, -$	$t_1\, t_1\, t_1\, t_1\, t_1\, f_1\, f_1\, f_1$
$\parallel \quad \parallel$	$\parallel \quad \parallel$	$\parallel \quad \parallel$
$\mathrm{val2}(\vee): ----\ t_2\, t_2\, t_2\, f_2$	$\mathrm{val2}(Z): -\ t_2\ f_2$	$t_2\, t_2\, t_2\, t_2\, t_2\, t_2$
$\parallel \quad \parallel$	$\parallel \quad \parallel$	$\parallel$
$\mathrm{val3}(\wedge): t_3\ f_3\, f_3\ \ f_3$	$\mathrm{val3}(Z): t_3 - f_3$	$t_3\, t_3\, t_3\, t_3\ \ t_3\ f_3$

$$\mathrm{val}^{(3,2)}\left(\mathrm{val}^{(3,2)}(X \wedge \vee \wedge Y) \vee \vee \vee\ \mathrm{val}^{(3,1)}(Z)\right) = \mathrm{val}^{(3,3)}((X \wedge \vee \wedge Y) \vee \vee \vee Z).$$

$$\left|\mathrm{val}^{(3,3)}(X, Y, Z)\right| = 3^3 - 3 = 24$$



PM	O_1	O_2	O_3
M_1	$S_{1.1.1}$	–	–
M_2	–	$S_{2.2.2}$	–
M_3	–	–	$S_{3.3.3}$

$\Longrightarrow$

PM	O_1	O_2	O_3
M_1	$[\{t1.1, f1.1\}^3]$	–	–
M_2	–	$[\{t2.2, f2.2\}^3]$	–
M_3	–	–	$[\{t3.3, f3.3\}^3]$

Tableaux – Rules

1. $\{t_i, f_i\} \in$ Contradiction, $i \in \binom{m}{2}$

2. $\{t_i, f_i, f_{i+1}\} \notin$ VB, $i \in \binom{m}{2}$

Example

$\{f_{2.3}, t_{1.3}, f_1 \equiv t_2\} \notin$ VB

Null

$$H = \left(X \vee \bigvee \bigvee \neg_1 Y\right) \vee \bigvee \bigvee \neg_2 Z\right)$$

$\mathbf{f1.1}\, H$	$\mathbf{t2.2}\, H$
$\text{f1.1}\left(X \vee \bigvee \bigvee \neg_1 Y\right)$	$\text{t2.2}\left(X \vee \bigvee \bigvee \neg_1 Y\right) \mid \text{t2.2}\neg_2 Z$
$\text{f1.1}\neg_2 Z \parallel \text{f3.3}\, Z$	$\text{t2.2}\, X \mid \text{t2.2}\ \neg_1 Y \mid \text{f2.2}\ Z$
$\text{f1.1}\, X$	$\parallel \text{t3.3}\, Y$
$\text{t1.1}\, Y$	

$$\left\{\text{f1.1, f1.1, f3.3}\right\}, \left\{\text{t2.2, t3.3, f2.2}\right\} \notin \text{VB.}$$

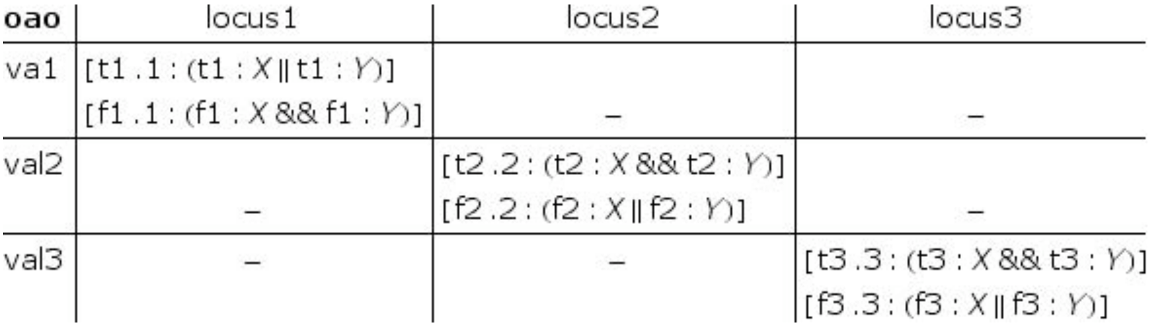
$$H = \neg_1\left(\left(\neg_1 \vee \bigvee \bigvee \neg_1 Y\right) \vee \bigvee \bigvee \neg_1 Z\right) .\text{ijj}. \left(\left(X \wedge \bigvee \bigvee Y\right) \wedge \bigvee \bigvee Z\right)$$

H	locus1	locus2	
va1	$\boldsymbol{\text{f1.1}\, H}$ $\text{t1.1}\ \neg_1\left(\neg_1 \vee \bigvee \bigvee \neg_1 Y\right) \vee \bigvee \bigvee \neg_1 Z$ $\text{f1.1}\left(X \wedge \vee \vee Y\right) \wedge \vee \vee Z$ $\text{f1.1}\left(\neg_1 X \vee \neg_1 Y\right) \vee \neg_1 Z$ $\text{f1.1}\neg_1 X \text{f1.1}\neg_1 Y$ $\text{f1.1}\neg_1 Z$ $\text{f1.1}\, X \mid \text{f1.1.}\, Y \mid \text{f1.1}\, Z$ xx | xx | xx	[]	
val2	[]	$\text{t3.2}\left(\neg_1 \vee \bigvee \bigvee \neg_1 Y\right) \vee \bigvee \bigvee \neg_1 Z$ $\text{t3.3}\neg_1 X \mid \text{t3.3}\neg_1 Y \mid \text{t3.3}\neg_1 Z$	$\text{t3.2}\left(\neg_1 \vee\right.$ $\text{t3.2}\neg_1 X \mid$
val3	[]	$\text{t2.3}\ \neg_1\left(\left(\neg_1 \vee \bigvee \bigvee \neg_1 Y\right) \vee \bigvee \bigvee \neg_1 Z\right)$ $\text{f2.3}\left(X \wedge \vee \vee Y\right) \wedge \vee \vee Z$ $\text{f1.1}\left(\left(\neg_1 X \vee \neg_1 Y\right) \vee \neg_1 Z\right)$ $\text{f2.3}\, X$ $\text{f2.3}\, Y$ $\text{f2.3}\, Z\, \text{t2.3}\, X \mid \text{t2.3}\, Y \mid \text{t2.3}\, Z$ xx | xx | xx	$(0)\, \boldsymbol{\text{f3.3}\, H:}$ $(1)\, \text{t3.3}\ \neg_1\left(\left(\right.\right.$ $(2)\, \text{f3.3}\left(X \wedge\right.$ $\text{f3.3}\, X$ $\text{f3.3}\, Y$ $\text{f3.3}\, Z\, \text{t3.3}\, X$ xx | xx

2.3. Matrix distribution

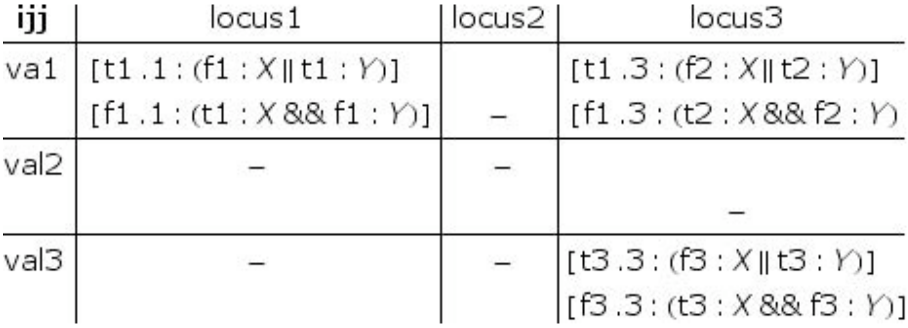
Infixes
||: disjunctive,
&&: conjunctive,
///: transjunctive,
\\\\\\: replicative branching.

oao: [||, &&]:

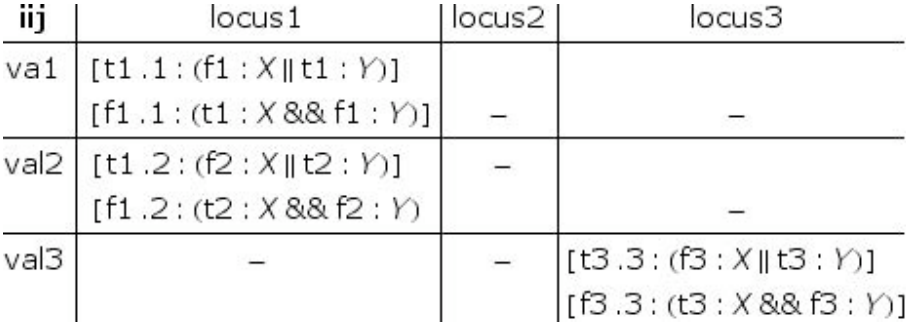


oao	locus1	locus2	locus3
va1	[t1 .1 : (t1 : *X* ‖ t1 : *Y*)] [f1 .1 : (f1 : *X* && f1 : *Y*)]	–	–
val2	–	[t2 .2 : (t2 : *X* && t2 : *Y*)] [f2 .2 : (f2 : *X* ‖ f2 : *Y*)]	–
val3	–	–	[t3 .3 : (t3 : *X* && t3 : *Y*)] [f3 .3 : (f3 : *X* ‖ f3 : *Y*)]

iij: [‖, &&, \\\]:

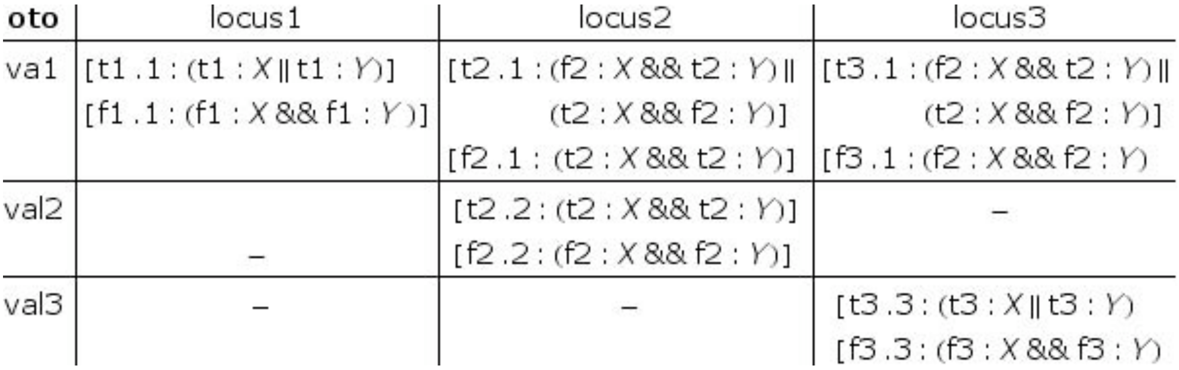


ijj	locus1	locus2	locus3
va1	[t1 .1 : (f1 : *X* ‖ t1 : *Y*)] [f1 .1 : (t1 : *X* && f1 : *Y*)]	–	[t1 .3 : (f2 : *X* ‖ t2 : *Y*)] [f1 .3 : (t2 : *X* && f2 : *Y*)
val2	–	–	–
val3	–	–	[t3 .3 : (f3 : *X* ‖ t3 : *Y*)] [f3 .3 : (t3 : *X* && f3 : *Y*)]



iij	locus1	locus2	locus3
va1	[t1 .1 : (f1 : *X* ‖ t1 : *Y*)] [f1 .1 : (t1 : *X* && f1 : *Y*)]	–	–
val2	[t1 .2 : (f2 : *X* ‖ t2 : *Y*)] [f1 .2 : (t2 : *X* && f2 : *Y*)	–	–
val3	–	–	[t3 .3 : (f3 : *X* ‖ t3 : *Y*)] [f3 .3 : (t3 : *X* && f3 : *Y*)]

oto: [‖, &&, ///]:



oto	locus1	locus2	locus3
va1	[t1 .1 : (t1 : *X* ‖ t1 : *Y*)] [f1 .1 : (f1 : *X* && f1 : *Y*)]	[t2 .1 : (f2 : *X* && t2 : *Y*) ‖ (t2 : *X* && f2 : *Y*)] [f2 .1 : (t2 : *X* && t2 : *Y*)]	[t3 .1 : (f2 : *X* && t2 : *Y*) ‖ (t2 : *X* && f2 : *Y*)] [f3 .1 : (f2 : *X* && f2 : *Y*)
val2	–	[t2 .2 : (t2 : *X* && t2 : *Y*)] [f2 .2 : (f2 : *X* && f2 : *Y*)]	–
val3	–	–	[t3 .3 : (t3 : *X* ‖ t3 : *Y*) [f3 .3 : (f3 : *X* && f3 : *Y*)

oto : [t1.1: t1.1:X ‖ t1.1:Y /// [] /// (f2.1:X && t2.1:Y) ‖ (t2.1:X && f2.1:Y) /// []]
[f1.1: f1.1:X && f1.1:Y /// [] /// t2.1:X && t2.1:Y /// []]
[t2.2: t2.2:X && t2.2:Y]
[f2.2: f2.2:X && f2.2:Y]
[t3.3: t3.3:X ‖ t3.3:Y /// [] /// (f2.1:X && t2:Y) ‖ (t2.1:X && f2.1:Y) /// []]
[f3.3: f3.3:X && f3.3:Y /// [] /// f2.1:X && f2.1:Y /// []]

Iterations

PM	O_1	O_2	O_3
M_1	$S_{1.1}$	$S_{2.1}$	–
M_2	$S_{1.2.2.2}$	$S_{2.2}$	$S_{3.2}$
M_3	–	$S_{2.3}$	$S_{3.3}$

ops $= [\circ, \mathbin{\scriptscriptstyle\blacksquare}, \blacktriangleright]$, *i.e.* composition, replication and iteration

$$\begin{pmatrix} O_1 \\ \amalg \\ O_2 \\ \amalg \\ O_3 \end{pmatrix} \begin{pmatrix} \circ -- \\ \amalg \\ -\circ- \\ \amalg \\ --\circ \end{pmatrix} \begin{pmatrix} M_1 \mathbin{\scriptscriptstyle\blacksquare} M_2 \blacktriangleright M_2 \blacktriangleright M_2 \\ \amalg \\ M_2 \mathbin{\scriptscriptstyle\blacksquare}_{2.1} M_1 \mathbin{\scriptscriptstyle\blacksquare}_{2.3} M_3 \\ \amalg \\ M_3 \mathbin{\scriptscriptstyle\blacksquare}_{3.2} M_2 \end{pmatrix} = \begin{pmatrix} (O_1 \circ M_1) \mathbin{\scriptscriptstyle\blacksquare} (O_1 \circ M_2) \blacktriangleright (O_1 \circ M_2) \blacktriangleright (O_1 \circ M_2) \\ \amalg \\ (O_2 \circ M_2) \mathbin{\scriptscriptstyle\blacksquare}_{2.1} (O_2 \circ M_1) \mathbin{\scriptscriptstyle\blacksquare}_{2.3} (O_2 \circ M_3) \\ \amalg \\ (O_3 \circ M_3) \mathbin{\scriptscriptstyle\blacksquare}_{3.2} (O_3 \circ M_2) \end{pmatrix}$$

Null

ii''' *j*	locus1	locus2	locus3
va1	[t1 .1 : (f1 : *X* ‖ t1 : *Y*)] [f1 .1 : (t1 : *X* && f1 : *Y*)]###	–	–
val2	[t1 .2 : (f2 : *X* ‖ t2 : *Y*)] [f1 .2 : (t2 : *X* && f2 : *Y*)] ▶ [t1 .2 .1 : (f2 : *X* ‖ t2 : *Y*)] [f1 .2 .1 : (t2 : *X* && f2 : *Y*)] ▶ [t1 .2 .2 : (f2 : *X* ‖ t2 : *Y*)] [f1 .2 .2 : (t2 : *X* && f2 : *Y*)]	–	–
val3	–	–	[t3 .3 : (f3 : *X* ‖ t3 : *Y*)] [f3 .3 : (t3 : *X* && f3 : *Y*)]

Example: Disambiguation of the implication "j" in "ooj"

ooj : [t1: t1: X ‖ t1:Y /// f2:X && f2:Y /// f3:X ‖ t3:Y]
[f1: f1:X && f1:Y /// t2:X ‖ t2:Y /// t3:X && f3:Y]

ooj' : [t1: t1: X ‖ t1:Y /// f2:X && f2:Y]
[f1: f1:X && f1:Y /// t2:X ‖ t2:Y]
[t3: f3:X ‖ t3:Y]
[f3: t3:X && f3:Y]

$$\left(\begin{array}{c|c|c|c} \text{ooj} & \square & \square & \square \\ \hline \square & 1 & 1 & 2 \\ \hline \square & 1 & 2 & 2 \\ \hline \square & 1 & 2 & 1 \end{array}\right) \Longrightarrow \left(\begin{array}{c|c|c|c} \text{ooj} & \square & \square & \square \\ \hline \square & t_1 & t_1 & f_1 \\ \hline \square & t_1 & f_1 & f_1 \\ \hline \square & t_1 & \text{f1} & t_1 \end{array}\right) \Longrightarrow \left(\begin{array}{c|c|c|c} \text{o}_{1.1}\,\text{o}_{1.2}\,\text{j}_{1.3} & \square & \square & \square \\ \hline \square & o_{1.1} & \square & \square \\ \hline \square & o_{1.2} & \square & \square \\ \hline \square & j_{1.3} & \square & \square \end{array}\right)$$

$$\left(\begin{array}{c|c|c|c} \text{ooj'} & \square & \square & \square \\ \hline \square & 1 & 1 & 3 \\ \hline \square & 1 & 2 & 2 \\ \hline \square & 1 & 2 & 1 \end{array}\right) \Longrightarrow \left(\begin{array}{c|c|c|c} \text{ooj} & \square & \square & \square \\ \hline \square & t_{1,3} & t_1 & f_3 \\ \hline \square & t_1 & f_1 & f_1 \\ \hline \square & t_3 & \text{f1} & t_{1,3} \end{array}\right) \Longrightarrow \left(\begin{array}{c|c|c|c} \text{o}_{1.1}\,\text{o}_{1.2}\,\text{j}_{3.3} & \square & \square & \square \\ \hline \square & o_{1.1} & \square & \square \\ \hline \square & o_{1.2} & \square & \square \\ \hline \square & - & \square & j_{3.3} \end{array}\right)$$

Result:

The specially distinguished implication " j' " is nothing more than the general implication " j "
which is different to the implication " j " at the place (1, 3). The abstract implication " j " is rep

2.3.1. Distributivity and Bifunctoriality of terms

Among the main rules to organize logical proof are the principle of *distributivity* for classical logic and *bifunctoriality* for polycontextural logics.

Distributivity

a1 && (a2 || a3) ⟹ (a1 && a2) || (a1 && a3)
(a1 || a2) && a3 ⟹ (a1 || a3) && (a2 || a3)

Bifunctoriality

Bifunctoriality of transposition

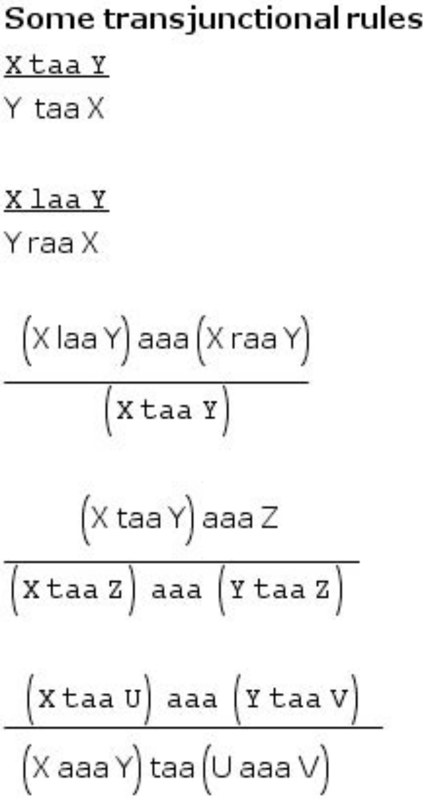
(t// /ta) && (t’ // /ta') ⇒ (t && t') // /(ta && ta')

Bifunctoriality of replication

(t \\\ ta) && (t’ \\\ ta') ⇒ (t && t') \\\ (ta && ta')

Interactivity of transposition and replication

(t \\\ ta) // /(t’ \\\ ta') ⇒ (t// /t') \\\ (ta // /' ta')



Some transjunctional rules

$$\frac{X\ \mathrm{taa}\ Y}{Y\ \mathrm{taa}\ X}$$

$$\frac{X\ \mathrm{laa}\ Y}{Y\ \mathrm{raa}\ X}$$

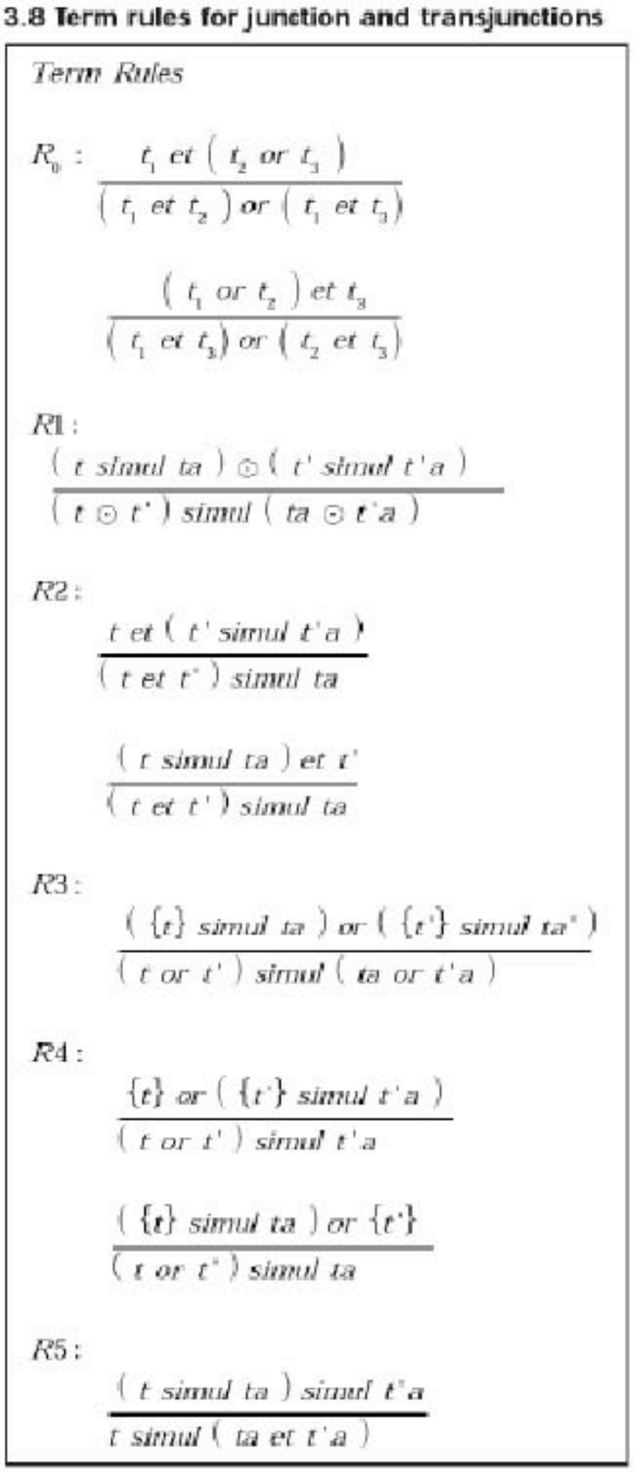
$$\frac{\left(X\ \mathrm{laa}\ Y\right)\ \mathrm{aaa}\ \left(X\ \mathrm{raa}\ Y\right)}{\left(X\ \mathrm{taa}\ Y\right)}$$

$$\frac{\left(X\ \mathrm{taa}\ Y\right)\ \mathrm{aaa}\ Z}{\left(X\ \mathrm{taa}\ Z\right)\ \mathrm{aaa}\ \left(Y\ \mathrm{taa}\ Z\right)}$$

$$\frac{\left(X\ \mathrm{taa}\ U\right)\ \mathrm{aaa}\ \left(Y\ \mathrm{taa}\ V\right)}{\left(X\ \mathrm{aaa}\ Y\right)\ \mathrm{taa}\ \left(U\ \mathrm{aaa}\ V\right)}$$

An extension in classical logic of Distr(X, Y, Z; a, o) to Distr(X, Y, U, V; a, o) is trivially achieved by substitution. In other words, Distr(X, Y, U, V, a, o) in classical logic is trivially reduced to Distr(X, Y, Z; a, o). Therefore, there is no genuine ‘bifunctiorial’ principle for classical logic. For polycontextural logics with their transjunctions ‘bifunctoriality’ is crucial.

BIF(X, Y, U, V; a, t) is not reducible to Distr(X, Y, U, V; a, o). Transjunctional operators are not definable in terms of junctional operators. Therefore, the rules for transjunctions and their interplay with junctions has to be established separately.



3.8 Term rules for junction and transjunctions

Term Rules

R_0 :
$$\frac{t_1 \text{ et } (t_2 \text{ or } t_3)}{(t_1 \text{ et } t_2) \text{ or } (t_1 \text{ et } t_3)}$$

$$\frac{(t_1 \text{ or } t_2) \text{ et } t_3}{(t_1 \text{ et } t_3) \text{ or } (t_2 \text{ et } t_3)}$$

R1 :
$$\frac{(t \text{ simul } ta) \odot (t' \text{ simul } t'a)}{(t \odot t') \text{ simul } (ta \odot t'a)}$$

R2 :
$$\frac{t \text{ et } (t' \text{ simul } t'a)}{(t \text{ et } t') \text{ simul } ta}$$

$$\frac{(t \text{ simul } ta) \text{ et } t'}{(t \text{ et } t') \text{ simul } ta}$$

R3 :
$$\frac{(\{t\} \text{ simul } ta) \text{ or } (\{t'\} \text{ simul } ta')}{(t \text{ or } t') \text{ simul } (ta \text{ or } t'a)}$$

R4 :
$$\frac{\{t\} \text{ or } (\{t'\} \text{ simul } t'a)}{(t \text{ or } t') \text{ simul } t'a}$$

$$\frac{(\{t\} \text{ simul } ta) \text{ or } \{t'\}}{(t \text{ or } t') \text{ simul } ta}$$

R5 :
$$\frac{(t \text{ simul } ta) \text{ simul } t'a}{t \text{ simul } (ta \text{ et } t'a)}$$

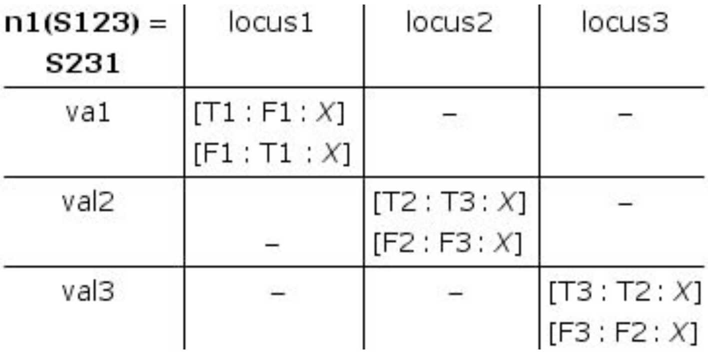
http://www.thinkartlab.com/pkl/lola/Transjunctions/Tale%20of%20Transjunctions.pdf

2.3.2. Negation and permutation

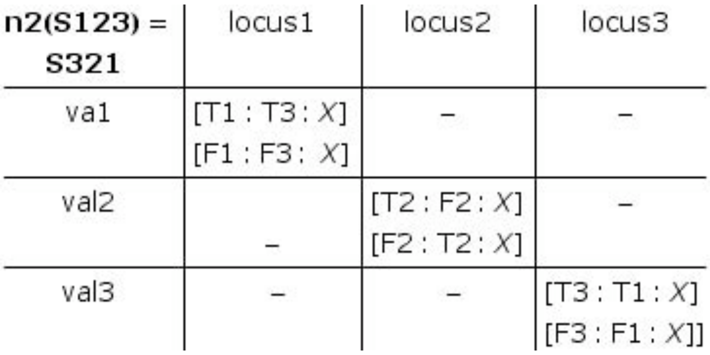
LOLA-definitions
n1 3 (X):=
[T1: F1: X] [F1: T1:X]
[T2: T3: X] [F2: F3:X]
[T3: T2: X] [F3: F2:X]

n2 3 (X) :=
[T1: T3: X] [F1: F3: X]
[T2: F2: X] [F2: T2:X]
[T3: T1: X] [F3: F1:X]

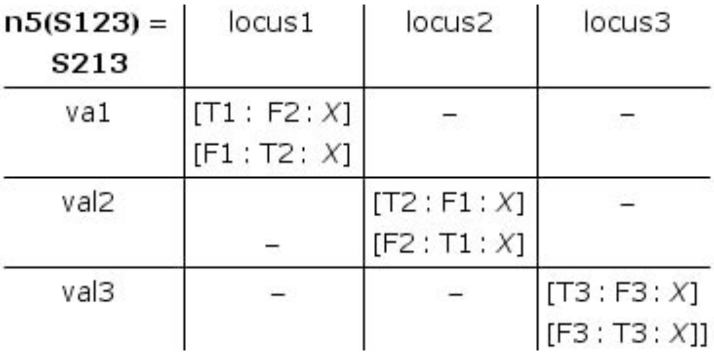
Matrix definitions of negations



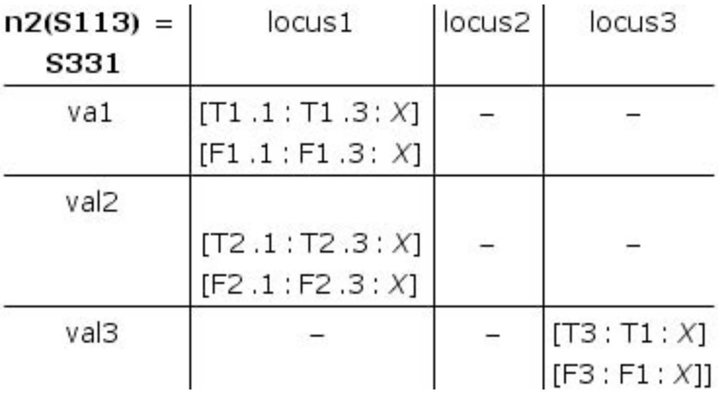
n1(S123) = S231	locus1	locus2	locus3
va1	[T1 : F1 : *X*] [F1 : T1 : *X*]	–	–
val2	–	[T2 : T3 : *X*] [F2 : F3 : *X*]	–
val3	–	–	[T3 : T2 : *X*] [F3 : F2 : *X*]



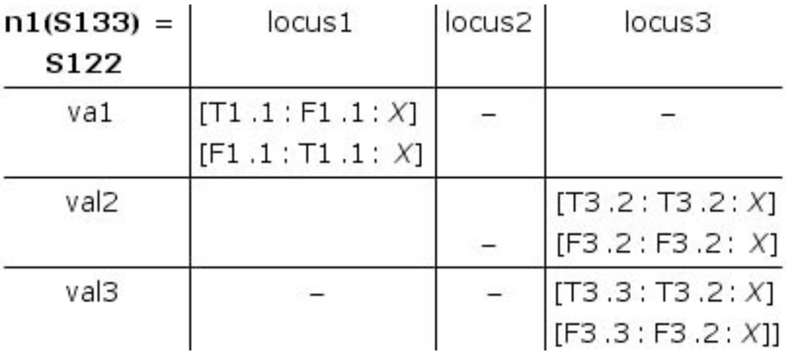
n2(S123) = S321	locus1	locus2	locus3
va1	[T1 : T3 : *X*] [F1 : F3 : *X*]	–	–
val2	–	[T2 : F2 : *X*] [F2 : T2 : *X*]	–
val3	–	–	[T3 : T1 : *X*] [F3 : F1 : *X*]]



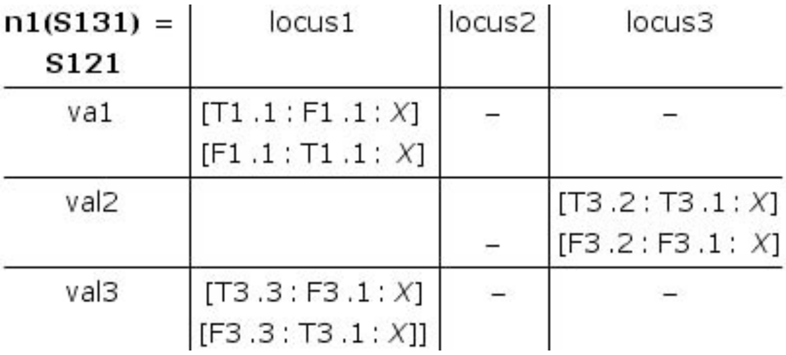
n5(S123) = S213	locus1	locus2	locus3
va1	[T1 : F2 : *X*] [F1 : T2 : *X*]	–	–
val2	–	[T2 : F1 : *X*] [F2 : T1 : *X*]	–
val3	–	–	[T3 : F3 : *X*] [F3 : T3 : *X*]]



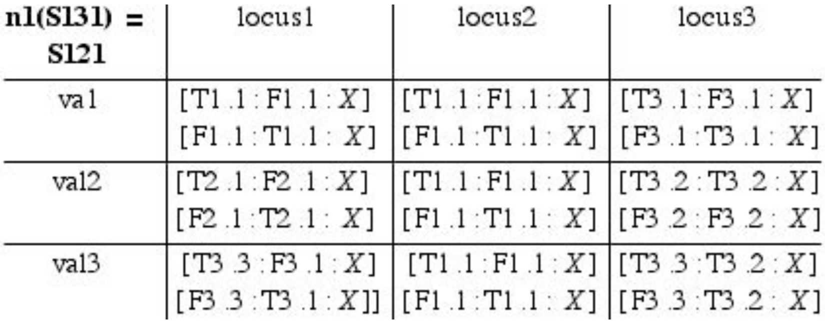
n2(S113) = S331	locus1	locus2	locus3
va1	[T1 .1 : T1 .3 : *X*] [F1 .1 : F1 .3 : *X*]	–	–
val2	[T2 .1 : T2 .3 : *X*] [F2 .1 : F2 .3 : *X*]	–	–
val3	–	–	[T3 : T1 : *X*] [F3 : F1 : *X*]]



n1(S133) = S122	locus1	locus2	locus3
va1	[T1 .1 : F1 .1 : *X*] [F1 .1 : T1 .1 : *X*]	–	–
val2		–	[T3 .2 : T3 .2 : *X*] [F3 .2 : F3 .2 : *X*]
val3	–	–	[T3 .3 : T3 .2 : *X*] [F3 .3 : F3 .2 : *X*]]



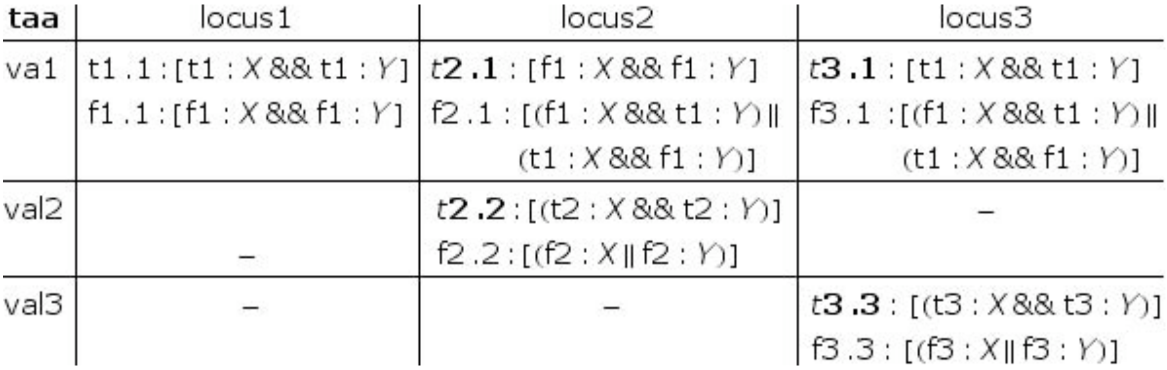
n1(S131) = **S121**	locus1	locus2	locus3
va1	[T1 .1 : F1 .1 : *X*] [F1 .1 : T1 .1 : *X*]	–	–
val2		–	[T3 .2 : T3 .1 : *X*] [F3 .2 : F3 .1 : *X*]
val3	[T3 .3 : F3 .1 : *X*] [F3 .3 : T3 .1 : *X*]]	–	–



n1(S131) = **S121**	locus1	locus2	locus3
va l	[T1 .1 : F1 .1 : *X*] [F1 .1 : T1 .1 : *X*]	[T1 .1 : F1 .1 : *X*] [F1 .1 : T1 .1 : *X*]	[T3 .1 : F3 .1 : *X*] [F3 .1 : T3 .1 : *X*]
val2	[T2 .1 : F2 .1 : *X*] [F2 .1 : T2 .1 : *X*]	[T1 .1 : F1 .1 : *X*] [F1 .1 : T1 .1 : *X*]	[T3 .2 : T3 .2 : *X*] [F3 .2 : F3 .2 : *X*]
val3	[T3 .3 : F3 .1 : *X*] [F3 .3 : T3 .1 : *X*]]	[T1 .1 : F1 .1 : *X*] [F1 .1 : T1 .1 : *X*]	[T3 .3 : T3 .2 : *X*] [F3 .3 : T3 .2 : *X*]

Matrix indices, again

taa : [t1: t1:X && t1:Y]
[f1: f1:X && f1:Y]
[t2: t2:X && t2:Y /// f1:X && f1:Y /// [] /// []]
[f2: f2:X || f2:Y /// (f1:X && t1:Y) || (t1:X && f1:Y) /// [] /// []]
[t3: t3:X && t3:Y /// t1:X && t1:Y /// [] /// []]
[f3: f3:X || f3:Y /// (f1:X && t1:Y) || (t1:X && f1:Y) /// [] /// []]



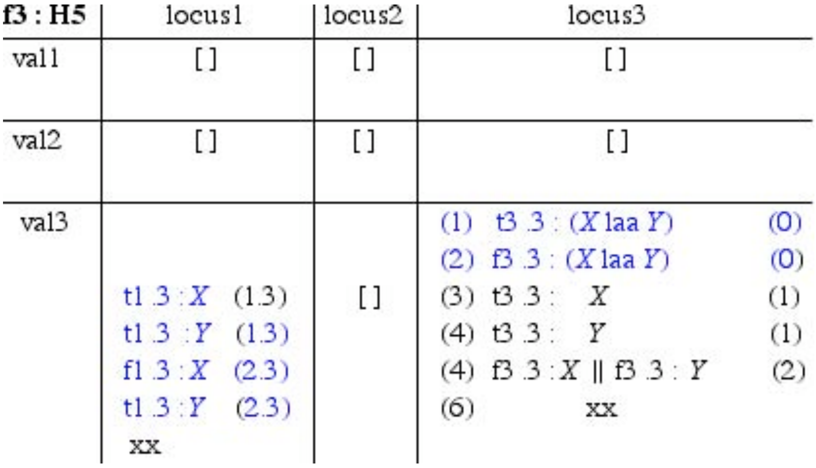
taa	locus1	locus2	locus3
va1	t1 .1 : [t1 : *X* && t1 : *Y*] f1 .1 : [f1 : *X* && f1 : *Y*]	**t2 .1** : [f1 : *X* && f1 : *Y*] f2 .1 : [(f1 : *X* && t1 : *Y*) || (t1 : *X* && f1 : *Y*)]	**t3 .1** : [t1 : *X* && t1 : *Y*] f3 .1 :[(f1 : *X* && t1 : *Y*) || (t1 : *X* && f1 : *Y*)]
val2	–	**t2 .2** : [(t2 : *X* && t2 : *Y*)] f2 .2 : [(f2 : *X* || f2 : *Y*)]	–
val3	–	–	**t3 .3** : [(t3 : *X* && t3 : *Y*)] f3 .3 : [(f3 : *X* || f3 : *Y*)]

πi.j : i=place, j=value, hence t2.1: value t1 at place locus2.

2.4. Forest-tableaux proofs

Another example; without negation

H5 = ((X laa Y) iij (X laa Y))

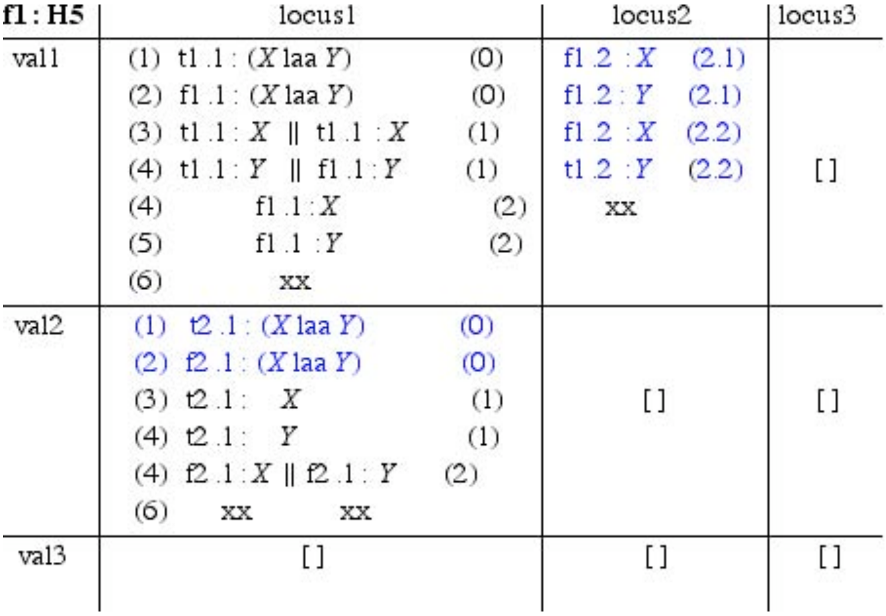


f3 : H5	locus1	locus2	locus3
val1	[]	[]	[]
val2	[]	[]	[]
val3	t1 .3 : X (1.3) t1 .3 : Y (1.3) f1 .3 : X (2.3) t1 .3 : Y (2.3) xx	[]	(1) t3 .3 : (X laa Y) (0) (2) f3 .3 : (X laa Y) (0) (3) t3 .3 : X (1) (4) t3 .3 : Y (1) (4) f3 .3 : X || f3 .3 : Y (2) (6) xx

Bifunctoriality
(A simul C) et (B simul D) = (A et B) simul (B et C):

$$t3\ .3 : (X \text{ laa } Y) \text{ et } f3\ .3 : (X \text{ laa } Y) = \begin{array}{ccc} t3\ .3 : X\ (1.) & & 11.3 : X\ (2.3) \\ \text{et} & \text{simul} & \text{et} \\ t1\ .3 : Y\ (1.3) & & t1\ .3 : Y\ (2.3) \end{array}$$

H5 = ((X laa Y) iij (X laaY))



f1 : H5	locus1	locus2	locus3
val1	(1) t1 .1 : (X laa Y) (0) (2) f1 .1 : (X laa Y) (0) (3) t1 .1 : X || t1 .1 : X (1) (4) t1 .1 : Y || f1 .1 : Y (1) (4) f1 .1 : X (2) (5) f1 .1 : Y (2) (6) xx	f1 .2 : X (2.1) f1 .2 : Y (2.1) f1 .2 : X (2.2) t1 .2 : Y (2.2) xx	[]
val2	(1) t2 .1 : (X laa Y) (0) (2) f2 .1 : (X laa Y) (0) (3) t2 .1 : X (1) (4) t2 .1 : Y (1) (4) f2 .1 : X || f2 .1 : Y (2) (6) xx xx	[]	[]
val3	[]	[]	[]

[1]LOLA-1993-proof of f1H5

```
f1: ((X laa Y) iij (X laa Y))
                                 |
                +---------------  ///  ---------------+  : this branching is
replicational \\\ in Matrix.
              |                                     |
              |                      +------------------------------+
      +------ && ------+             |              |               |
      |                |             |      +------ && ------+      |
t1: (X laa Y)    f1: (X laa Y)       |      |                |      |
                                     |t2: (X laa Y)    f2: (X laa Y)|
                                     +------------------------------+
                                                     |
                                   +----------------  ///  ---------------+  :
this branching is transjunctional.
```

```
                                     |                                          |
                                     |                         +-------------------------
                                                             +-----------------     &&
------+                 |               |                 |
                  |                               |                 |      +------- &&
------+        |
                 |                             |                |       |
        +--------  ||  ------+                    +--  &&  --+              |t2:  (X  laa
Y)     f2: (X laa Y)|
       |                   |               |         |         +-------------------------
            |                                 |                      f1:  X        f1:
Y
       +--      &&      --+                                                +--      &&
--+
   |         |         |         |
t1: X    t1: Y    t1: X    f1: Y
```

The proof ends with:

```
|
+-  ///   ---------+          : replicational
|                  |
+-+        +-----------------+
|$|        |       |         | : transjunctional.
+-+        | +-   ///   --+  |
           | |            |  |
           |+-+          +---+|
           ||$|          |+-+||
           |+-+          ||$|||
           |             |+-+||
           |             +---+|
           +-----------------+
```

2.4.1. Bifunctoriality in action

f1 H1 = f1 ((X taa Y) .iij. n5(n5 X oto n5 Y)

The clean separation of the transjunctional and the junctional parts of the tableaux is ruled by the bifunctorial term rule R1 based on the matrix representation.

Again, without a strict tabular matrix-scheme of the distribution of tableaux_trees, an application of the bifunctorial term rule wouldn't be possible.
On the other hand, an intuitive approach as shown in the colored term development would be helplessly lost in higher complexity. The example shows at least a first step of a distribution of different loci, S1 and S2.

In fact, the concept of a tabular dissemination was clearly stateted but the bifunctorial term rules had still been missing for h conceptual and manual approach. Nevertheless, bifunctoriality was at the base of the implementation of the theorem prover LOLA (1993).

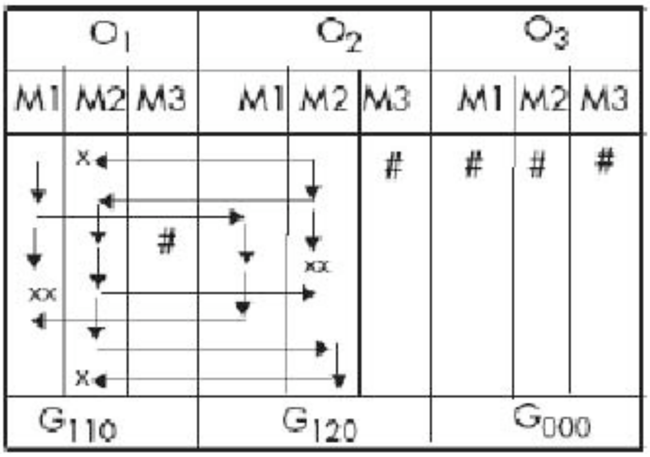
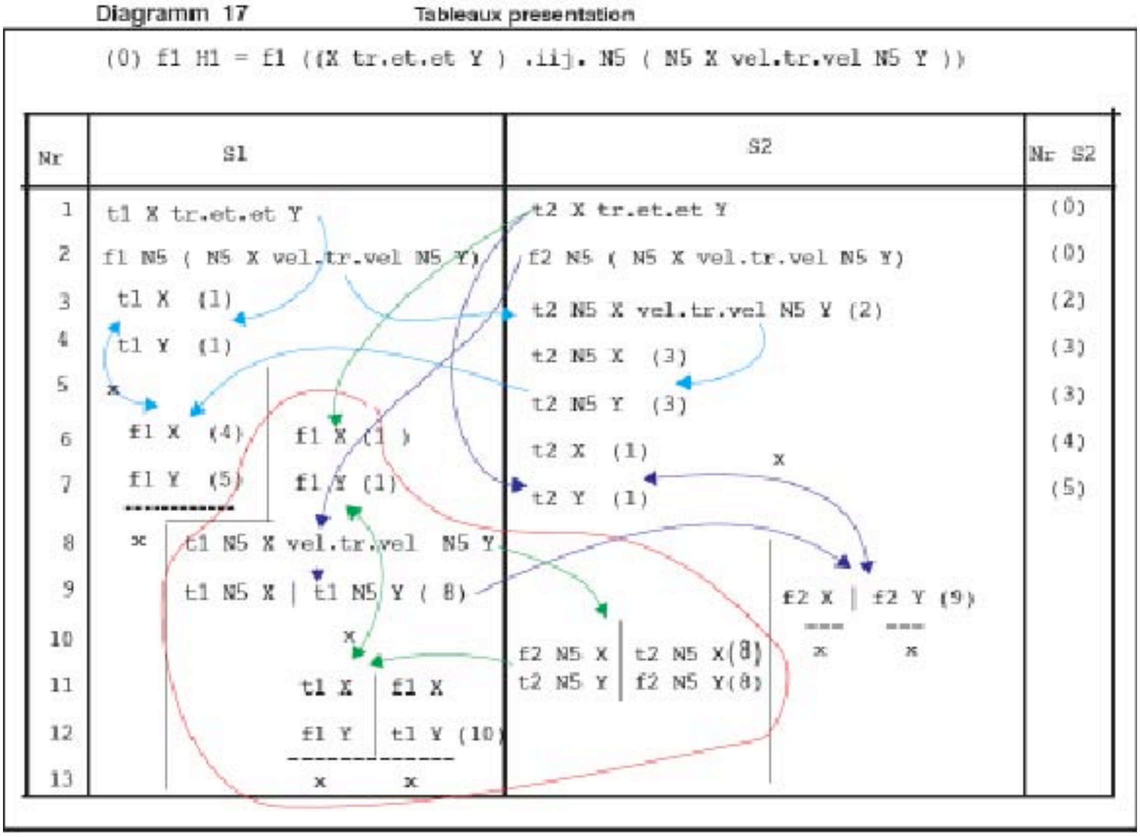


Diagramm 17 Tableaux presentation



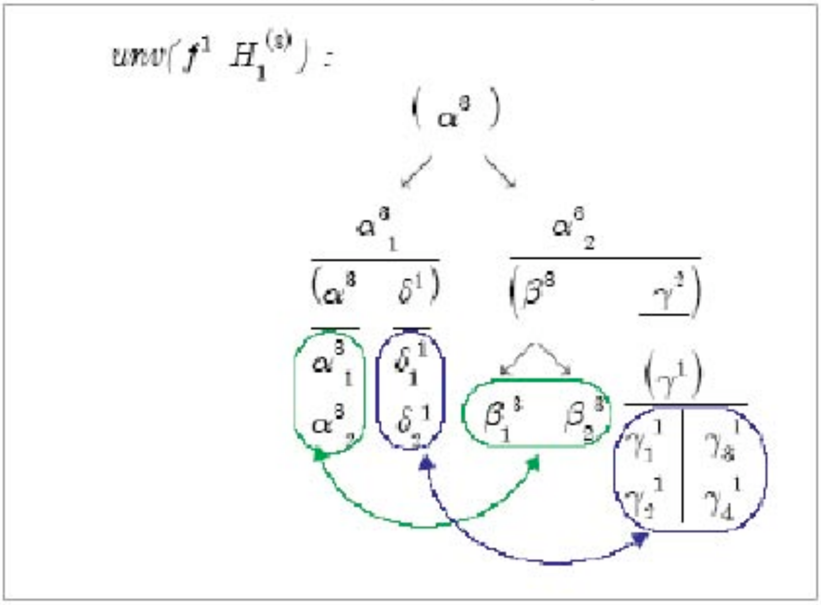
(0) f1 H1 = f1 ((X tr.et.et Y) .iij. N5 (N5 X vel.tr.vel N5 Y))

Nr	S1	S2	Nr S2
1	t1 X tr.et.et Y	t2 X tr.et.et Y	(0)
2	f1 N5 (N5 X vel.tr.vel N5 Y)	f2 N5 (N5 X vel.tr.vel N5 Y)	(0)
3	t1 X (1)	t2 N5 X vel.tr.vel N5 Y (2)	(2)
4	t1 Y (1)	t2 N5 X (3)	(3)
5	x	t2 N5 Y (3)	(3)
6	f1 X (4) | f1 X (1)	t2 X (1)	(4)
7	f1 Y (5) | f1 Y (1)	t2 Y (1) x	(5)
8	x | t1 N5 X vel.tr.vel N5 Y		
9	t1 N5 X | t1 N5 Y (8)	f2 X | f2 Y (9)	
10	x	f2 N5 X | t2 N5 X(8) x x	
11	t1 X | f1 X	t2 N5 Y | f2 N5 Y(8)	
12	f1 Y | t1 Y (10)		
13	x x		

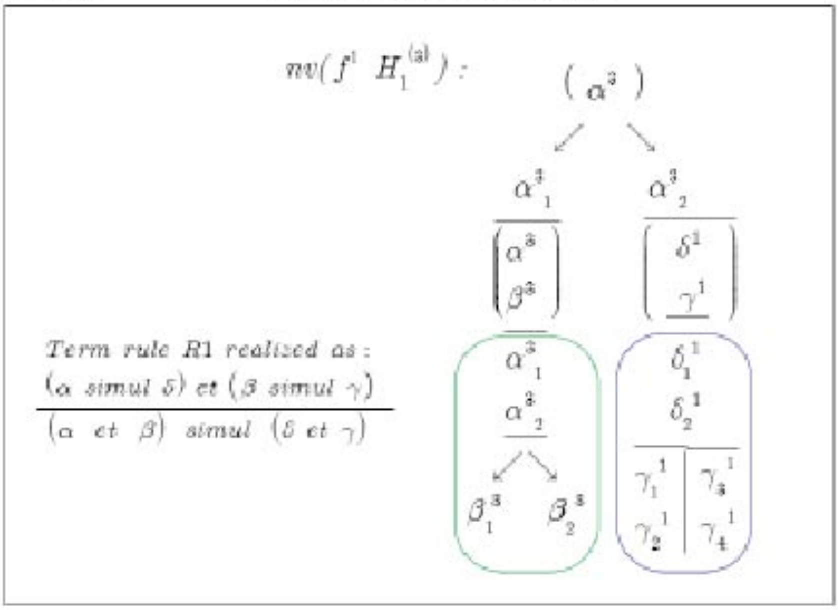
Sub-systems S1 and S2 are closing directly, S1 at step 8 and S2 at step 9. This would be enough to close the tableaux for S1 and S2. But there is an additional part of the formula which is closing separately in S1, closing at step 13, encircled in red.

The analysis on the base of the bifunctorial term rule R1 gives a conceptually and technically clear result and eliminates the adhocism of the “red” results.

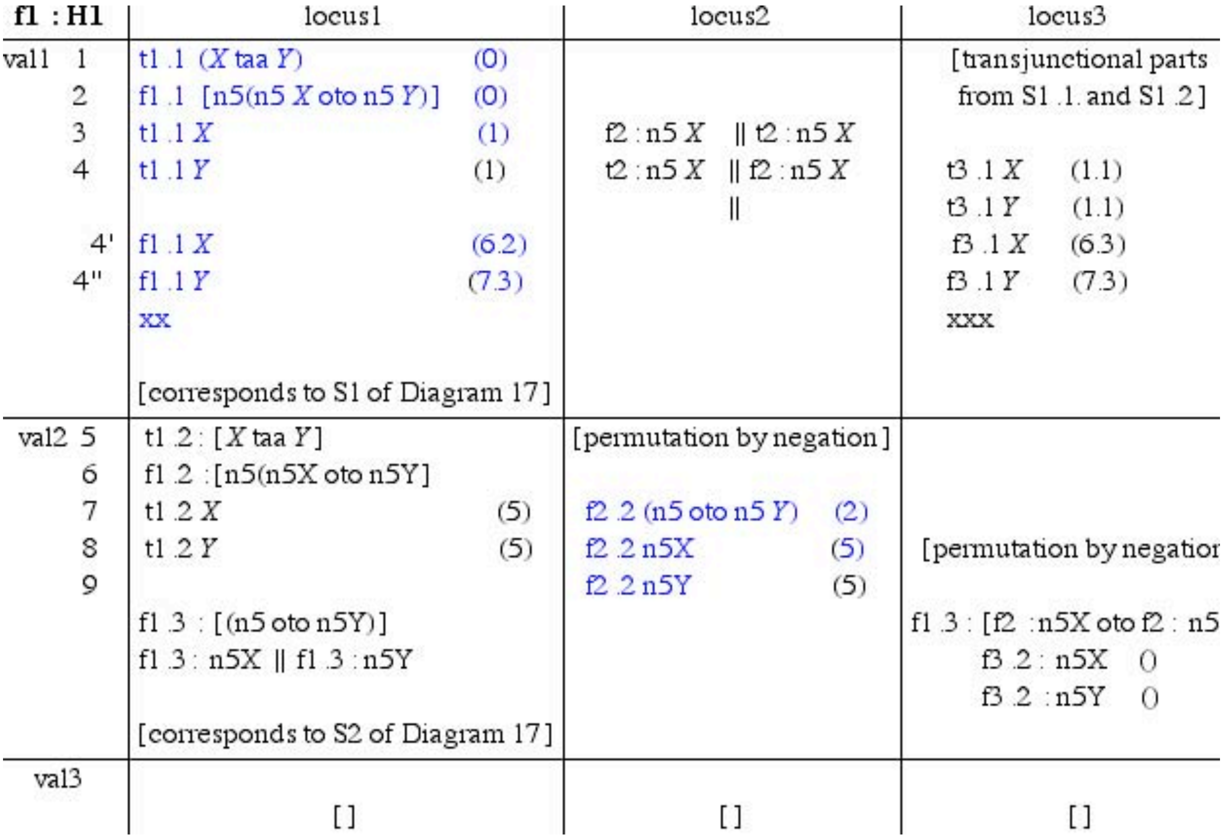
18 Unification tableaux tree development of f1H1



amm 19 Unification of f1H1 with meta-rules



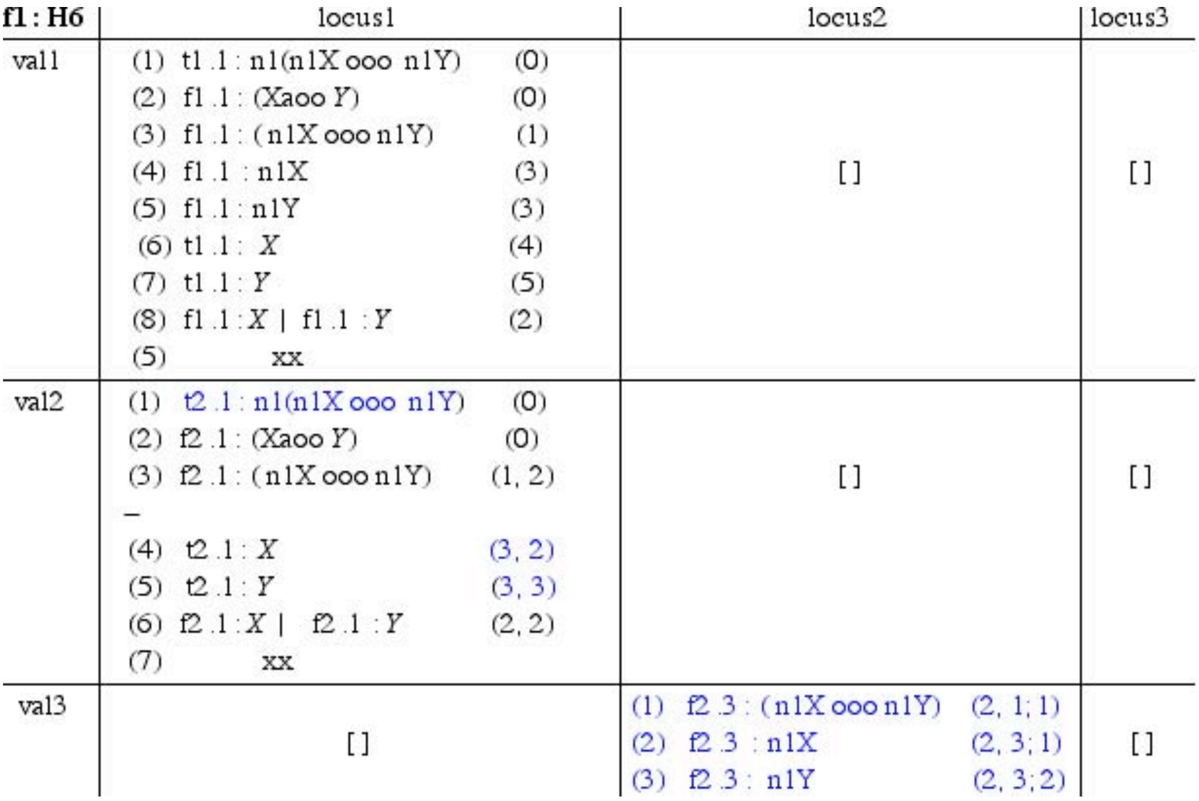
f1 H1 = f1 ((X taa Y) .iij. n5(n5 X oto n5Y)



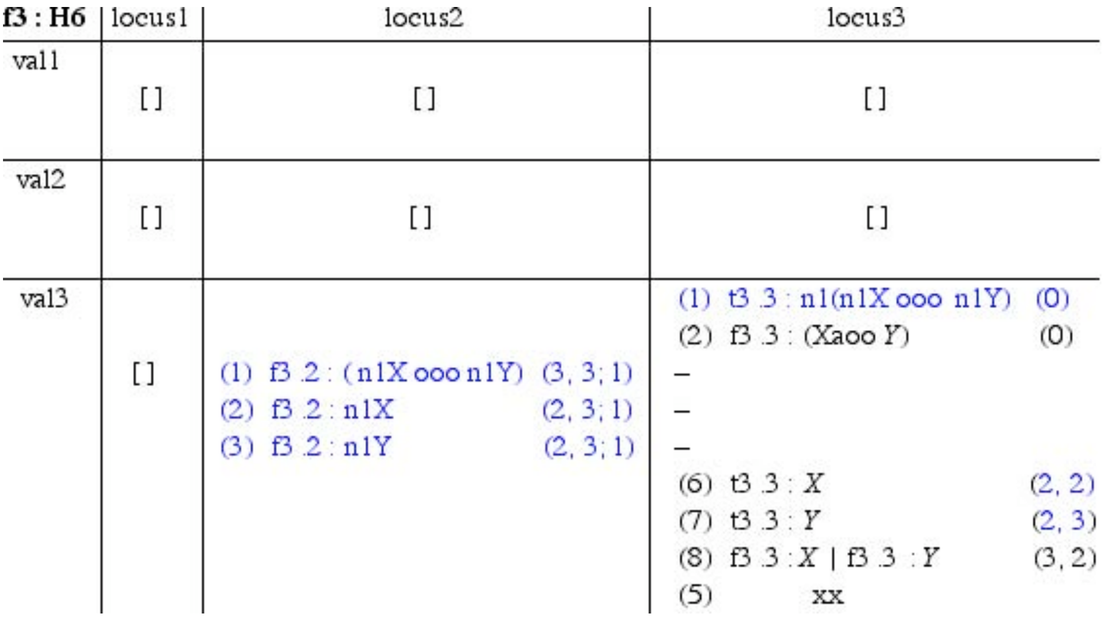
f1 : H1	locus1		locus2	locus3
val1 1	t1 .1 (*X* taa *Y*)	(0)		[transjunctional parts
2	f1 .1 [n5(n5 *X* oto n5 *Y*)]	(0)		from S1 .1. and S1 .2]
3	t1 .1 *X*	(1)	f2 : n5 *X* ‖ t2 : n5 *X*	
4	t1 .1 *Y*	(1)	t2 : n5 *X* ‖ f2 : n5 *X*	t3 .1 *X* (1.1)
			‖	t3 .1 *Y* (1.1)
4'	f1 .1 *X*	(6.2)		f3 .1 *X* (6.3)
4"	f1 .1 *Y*	(7.3)		f3 .1 *Y* (7.3)
	xx			xxx
	[corresponds to S1 of Diagram 17]			
val2 5	t1 .2 : [*X* taa *Y*]		[permutation by negation]	
6	f1 .2 :[n5(n5X oto n5Y]			
7	t1 .2 *X*	(5)	f2 .2 (n5 oto n5 *Y*) (2)	
8	t1 .2 *Y*	(5)	f2 .2 n5X (5)	[permutation by negatior
9			f2 .2 n5Y (5)	
	f1 .3 : [(n5 oto n5Y)]			f1 .3 : [f2 :n5X oto f2 : n5
	f1 .3 : n5X ‖ f1 .3 : n5Y			f3 .2 : n5X ()
				f3 .2 :n5Y ()
	[corresponds to S2 of Diagram 17]			
val3	[]		[]	[]

2.4.2. Negation and permutation in proofs

H6 = (n1(n1X ooo n1Y) iij (X aoo Y)



f1 : H6	locus1	locus2	locus3
val1	(1) t1 .1 : n1(n1X ooo n1Y) (0) (2) f1 .1 : (Xaoo *Y*) (0) (3) f1 .1 : (n1X ooo n1Y) (1) (4) f1 .1 : n1X (3) (5) f1 .1 : n1Y (3) (6) t1 .1 : *X* (4) (7) t1 .1 : *Y* (5) (8) f1 .1 : *X* | f1 .1 : *Y* (2) (5) xx	[]	[]
val2	(1) t2 .1 : n1(n1X ooo n1Y) (0) (2) f2 .1 : (Xaoo *Y*) (0) (3) f2 .1 : (n1X ooo n1Y) (1, 2) – (4) t2 .1 : *X* (3, 2) (5) t2 .1 : *Y* (3, 3) (6) f2 .1 : *X* | f2 .1 : *Y* (2, 2) (7) xx	[]	[]
val3	[]	(1) f2 .3 : (n1X ooo n1Y) (2, 1; 1) (2) f2 .3 : n1X (2, 3; 1) (3) f2 .3 : n1Y (2, 3;2)	[]



f3 : H6	locus1	locus2	locus3
val1	[]	[]	[]
val2	[]	[]	[]
val3	[]	(1) f3 .2 : (n1X ooo n1Y) (3, 3; 1) (2) f3 .2 : n1X (2, 3; 1) (3) f3 .2 : n1Y (2, 3; 1)	(1) t3 .3 : n1(n1X ooo n1Y) (0) (2) f3 .3 : (Xaoo *Y*) (0) – – – (6) t3 .3 : *X* (2, 2) (7) t3 .3 : *Y* (2, 3) (8) f3 .3 : *X* | f3 .3 : *Y* (3, 2) (5) xx

2.4.3. Negations and permutations

$$\textbf{H7} = \text{n1}\left(\text{n2}\left(\text{n1}\left(\text{n2}\left(\text{n1}\left(\text{n2}\left(X\right)\right)\right)\right)\right)\right) \text{ iij } \left(X\right)$$

f1 . 1 : H7	negation cycle
t1 . 1 f1 . 1	n1(n2(n1(n2(n1(n2(X)))))) (X)
f1 . 1	n2(n1(n2(n1(n2(X)))))
f1 . 3	n1(n2(n1(n2(X))))
f1 . 2	n2(n1(n2(X)))
t1 . 2	n1(n2(X))
t1 . 3	n2(X)
t1 . 1	(X)
–	xxx

f2 . 1 : H7	negation cycle
t2 . 1 f2 . 1	n1(n2(n1(n2(n1(n2(X)))))) (X)
f2 . 1	n2(n1(n2(n1(n2(X)))))
f2 . 3	n1(n2(n1(n2(X))))
f2 . 2	n2(n1(n2(X)))
t2 . 2	n1(n2(X))
t2 . 3	n2(X)
t2 . 1	X
–	xxx

f3 . 3 : H7	
t3 . 3 f3 . 3	n
t3 . 2	
f3 . 2	
f3 . 3	
f3 . 1	
t3 . 1	
t3 . 3	
–	

f3 . 3 : H7	sub1	sub2	sub3
t3 . 3 f3 . 3	–	–	n1(n2(n1(n2(n1(n2(X)))))) (X)
f2 . 2	–	n2(n1(n2(n1(n2(X)))))	–
t2 . 2	–	n1(n2(n1(n2(X))))	–
t3 . 3	–	–	n2(n1(n2(X)))
t1 . 1	n1(n2(X))	–	–
f1 . 1	n2(X)	–	–
f3 . 3	–	–	(X)
–	–	–	xxx

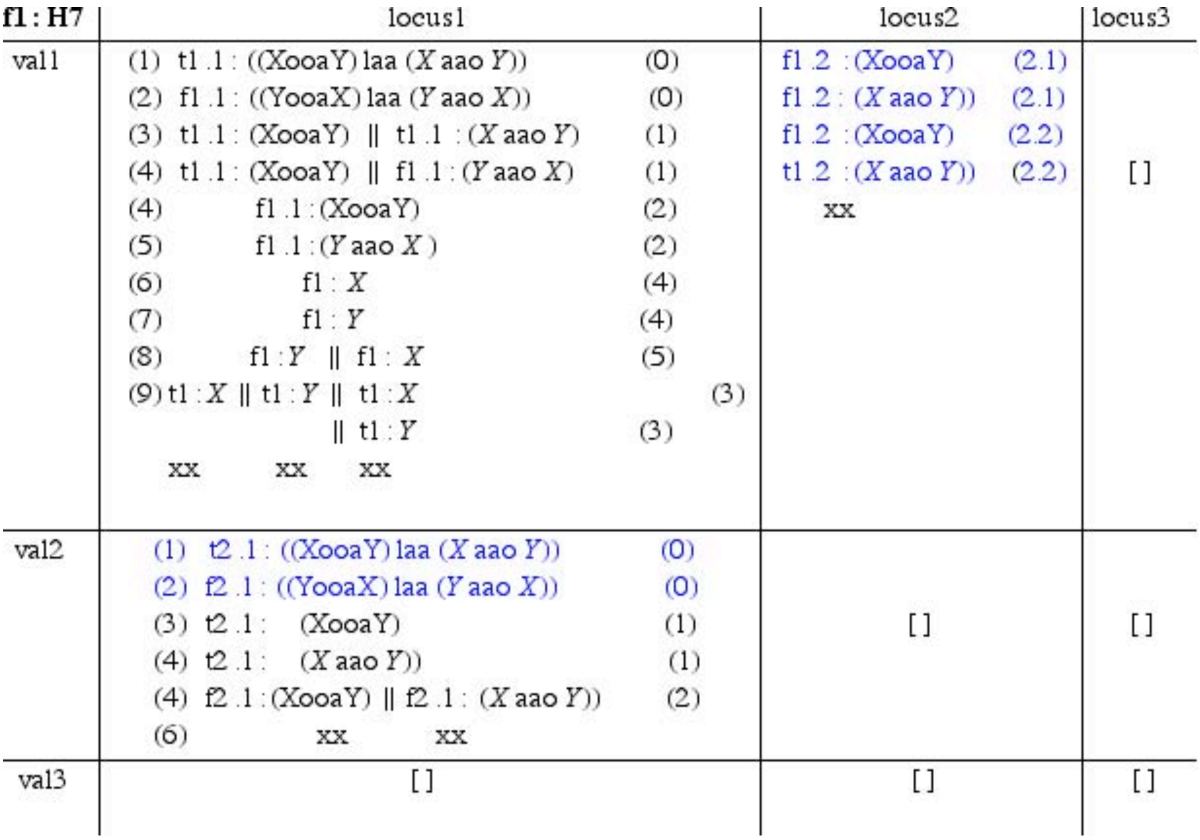
neg1(ijj) :

$[t1.1,\ f1.1] = [t1,\ f1] : pos(1,\ 1) \Longrightarrow [f1.1,\ t1.1] = [t1,\ f1] : pos(1,\ 1)$

$[t1.3,\ f1.3] = [t1,\ f1] : pos(1,\ 3) \Longrightarrow [t2.2,\ f2.2] = [t2,\ f2] : pos(2,\ 2)$

$[t3.3,\ f3.3] = [t3,\ f3] : pos(3,\ 3) \Longrightarrow [t2.3,\ f2.3] = [t2,\ f2] : pos(2,\ 3)$

H7 = ((XooaY) laa (X aao Y)) iij ((YooaX) laa (Y aao X))



f1 : H7	locus1	locus2	locus3
val1	(1) t1 .1 : ((XooaY) laa (X aao Y)) (0) (2) f1 .1 : ((YooaX) laa (Y aao X)) (0) (3) t1 .1 : (XooaY) || t1 .1 : (X aao Y) (1) (4) t1 .1 : (XooaY) || f1 .1 : (Y aao X) (1) (4) f1 .1 : (XooaY) (2) (5) f1 .1 : (Y aao X) (2) (6) f1 : X (4) (7) f1 : Y (4) (8) f1 : Y || f1 : X (5) (9) t1 : X || t1 : Y || t1 : X (3) || t1 : Y (3) xx xx xx	f1 .2 : (XooaY) (2.1) f1 .2 : (X aao Y)) (2.1) f1 .2 : (XooaY) (2.2) t1 .2 : (X aao Y)) (2.2) xx	[]
val2	(1) t2 .1 : ((XooaY) laa (X aao Y)) (0) (2) f2 .1 : ((YooaX) laa (Y aao X)) (0) (3) t2 .1 : (XooaY) (1) (4) t2 .1 : (X aao Y)) (1) (4) f2 .1 : (XooaY) || f2 .1 : (X aao Y)) (2) (6) xx xx	[]	[]
val3	[]	[]	[]

2.5. Decision mechanisms

2.5.1. Intra-contextural logical proofs

From an environment of a complex system *Syst*$^{(3,3)}$ a request might be addressed, asking to check if the logical formula *H5* is valid for all its subsystems. In short: Does the complex tableaux prover LOLA$^{\text{plus}}$proof the validity of *H5*? In other words: Are all subsystem tableaux closing in all branches of the tree-development of the formula H5 for f1 and f3?

Firstly, it is supposed that the formula H5 is syntactically well-formed, then questions arises about its semantic and its proof-theoretic properties.

input question for Syst$^{(3,3)}$: H5 ∈ tautology? :
H5 ∈ tautology iff f1H5 and f3H5 ∈ taut.

calculation : f1H5	locus1	locus2	locus3
val1	Sys1 .1 xx	Sys1 .2 xx	[]
val2	Sys2 .1 xx	[]	[]
val3	[]	[]	[]

=

answer	locus1	lo
val1	semantic provability @ 1.1	seman provab
val2	semantic provability @2.1	

calculation : f3H5	locus1	locus2	locus3
val1	[]	[]	[]
val2	[]	[]	[]
val3	Sys3 .1 xx	[]	Sys3 .3 xx

=

answer	locus1	locus3
val3	semantic provability @3.1	semantic provability

calculation : f1 .3 H5	locus1	locus2	locus3
val1	Sys1 .1 xx	Sys1 .2 xx	[]
val2	Sys2 .1 xx	[]	[]
val3	Sys3 .1 xx	[]	Sys3 .3 xx

Result : f1 H5 and **f3H5** ∈ **tau***t*, hence **H5** ∈ **taut.**

2.5.2. Polycontextural transformations

How to define or represent matrix-constellations of logical operators out of other existing constellations?
In classical logic this reduces to the question, how to represent, say, conjunction by disjunction? There is no need to ask for more. Classical junctions are ‘one-place’ operations in the matrix. That is, classical junctions are not distributed but appear at singular not marked place.
De Morgan is at its place. Completeness of definability by say the Sheffer stroke is secured.

Definability for polycontextural logics (pcl) is more intricate. How to define, a pcl-constellation by means of other pcl-constellations? Is there a Sheffer analogon? Are transjunctional functions definable by junctional functions?
It is well known that transjunctional logical functions are not definable by junctional functions and negations.

This has a structural equivalent: Bifunctoriality is not definable by functorial distributivity, and other junctional rules.

Furthermore, a new abstraction is achieved with the consideration of the matrix-constellations of functions only. Neither decidablity and definability of distributed contextural logical formulas is here in the focus but just the different *places* in the matrix where the parts of the complex formulas are taking place.

Hence, how to define constellations?

One answer is given by the introduction of the super-operators *sops* = {id, perm, red, repl, iter, bif}.
Hence, the question transforms to the question of the logical definability of the super-operator transformation in the domain of logic..

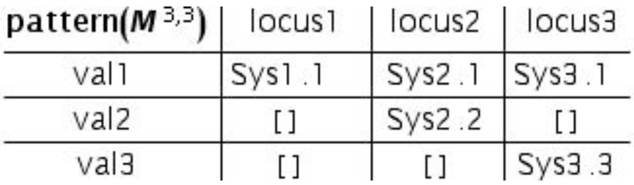
From the point of view of matrix-distribution, constellations like "*laa*" and "*raa*" are equivalent. But also "*taa*" as "*laa.*" "*aaa*". "*raa*" are equivalent too. And more: the conjunctions "*aa*" might be replaced by disjunctions "*oo*", or even by a combination of "*a*" and "*o*", "*ao*" and "*oa*".

A chain of abstractions
Matrix pattern($M^{(m,n)}$) --> [α,β,γ,δ]-distribution --> π-value distribution.

A further abstraction leads directly to the value independent constellations of *morphogrammatics*. On a morphogrammatic level, combinatorial questions had been elaborated in extenso.

Example

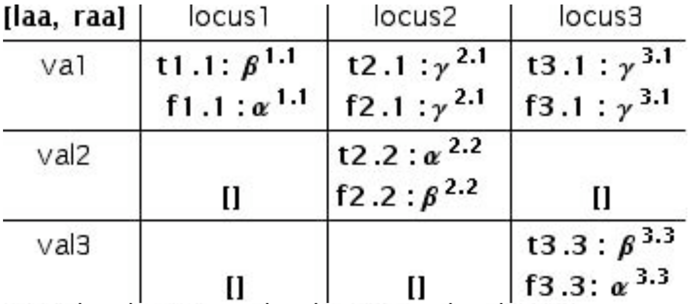
pattern($M^{3,3}$):



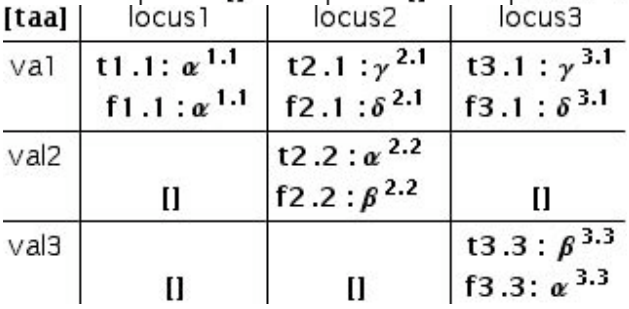
pattern($M^{3,3}$)	locus1	locus2	locus3
val1	Sys1 .1	Sys2 .1	Sys3 .1
val2	[]	Sys2 .2	[]
val3	[]	[]	Sys3 .3

The constellation *pattern(M)* has at least three unified main realizations: *unif(raa), unfi(raa)* and *unif(taa),* and additionally the secondary junctional combinations, like "tao", "toa", "too", etc.

[α,β,γ,δ]-distribution:



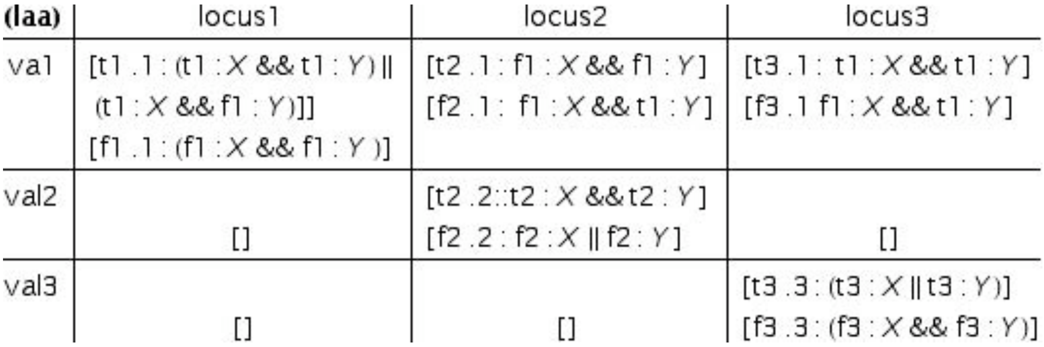
[laa, raa]	locus1	locus2	locus3
val1	t1 .1: $\beta^{1.1}$ f1 .1 : $\alpha^{1.1}$	t2 .1 :$\gamma^{2.1}$ f2 .1 :$\gamma^{2.1}$	t3 .1 : $\gamma^{3.1}$ f3 .1 : $\gamma^{3.1}$
val2	[]	t2 .2 : $\alpha^{2.2}$ f2 .2 : $\beta^{2.2}$	[]
val3	[]	[]	t3 .3 : $\beta^{3.3}$ f3 .3: $\alpha^{3.3}$



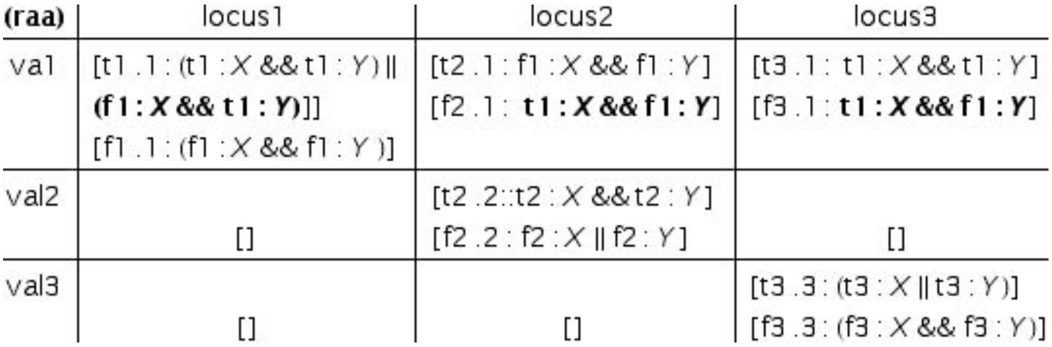
[taa]	locus1	locus2	locus3
val1	t1 .1: $\alpha^{1.1}$ f1 .1 :$\alpha^{1.1}$	t2 .1 :$\gamma^{2.1}$ f2 .1 :$\delta^{2.1}$	t3 .1 : $\gamma^{3.1}$ f3 .1 : $\delta^{3.1}$
val2	[]	t2 .2 : $\alpha^{2.2}$ f2 .2 : $\beta^{2.2}$	[]
val3	[]	[]	t3 .3 : $\beta^{3.3}$ f3 .3: $\alpha^{3.3}$

Again, α-β-constellations have different value realisations π (α, β). Nevertheles, their combinations, say conjunctions with implications, have to consider their mediation rules.

π-value distribution:



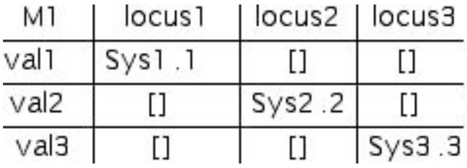
(laa)	locus1	locus2	locus3
val1	[t1 .1 : (t1 : X && t1 : Y) || (t1 : X && f1 : Y)]] [f1 .1 : (f1 : X && f1 : Y)]	[t2 .1 : f1 : X && f1 : Y] [f2 .1 : f1 : X && t1 : Y]	[t3 .1 : t1 : X && t1 : Y] [f3 .1 f1 : X && t1 : Y]
val2	[]	[t2 .2::t2 : X &&t2 : Y] [f2 .2 : f2 : X || f2 : Y]	[]
val3	[]	[]	[t3 .3 : (t3 : X || t3 : Y)] [f3 .3 : (f3 : X && f3 : Y)]



(raa)	locus1	locus2	locus3
val1	[t1 .1 : (t1 : X && t1 : Y) || **(f1 : X && t1 : Y)**]] [f1 .1 : (f1 : X && f1 : Y)]	[t2 .1 : f1 : X && f1 : Y] [f2 .1 : **t1 : X && f1 : Y**]	[t3 .1 : t1 : X && t1 : Y] [f3 .1 : **t1 : X && f1 : Y**]
val2	[]	[t2 .2::t2 : X &&t2 : Y] [f2 .2 : f2 : X || f2 : Y]	[]
val3	[]	[]	[t3 .3 : (t3 : X || t3 : Y)] [f3 .3 : (f3 : X && f3 : Y)]

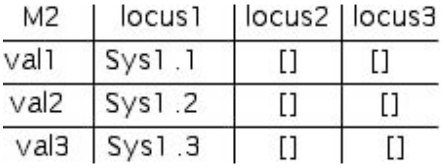
The constellations [raa] and [laa] are equivalent on the unificational level but different on the value level of the concrete formulas. Hence,
[laa]: [t1 .1 : t1 : X && f1 : Y] ≠ [raa]: **[t1 .1 : f1 : X && t1 : Y]**, and [laa]: [f2 .1 : f1 : X && t1 : Y] != [f2 .1 : **t1 : X && f1 : Y**].

Transformations
How to transform constellation M1 into constellation M2?:
Formaly, it is a *reduction*: $\text{red}_{2.3}$: M1 --> M2.
How is this reduction realized on the logicl level?



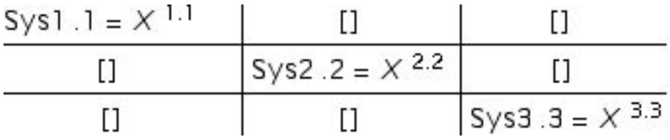
M1	locus1	locus2	locus3
val1	Sys1 .1	[]	[]
val2	[]	Sys2 .2	[]
val3	[]	[]	Sys3 .3

=>



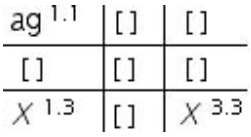
M2	locus1	locus2	locus3
val1	Sys1 .1	[]	[]
val2	Sys1 .2	[]	[]
val3	Sys1 .3	[]	[]

Take $X^{(3)}$ =



Sys1 .1 = $X^{1.1}$	[]	[]
[]	Sys2 .2 = $X^{2.2}$	[]
[]	[]	Sys3 .3 = $X^{3.3}$

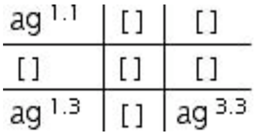
with ($X^{(3)}$ ooo neg1($X^{3)}$)) we get [$ag^{1.1}$, $X^{3.1}$, $X^{3.3}$]



$ag^{1.1}$	[]	[]
[]	[]	[]
$X^{1.3}$	[]	$X^{3.3}$

,

and with (($X^{(3)}$ ooo neg1($X^{3)}$)) ooo neg3(X^{3})) we get[$ag^{1.1}$, $ag^{1.3}$, $ag^{3.3}$]



$ag^{1.1}$	[]	[]
[]	[]	[]
$ag^{1.3}$	[]	$ag^{3.3}$

.

2.6. Abstractions: Matrix, super-operators, unification, tableaux

2.6.1. Matrix abstractions

$$H = (X \text{ laa } Y) \text{ iij } (X \text{ laaY})$$

f1 : H5	locus1	locus2	locus3
val1	Sys1 .1 xx	Sys1 .2 xx	[]
val2	Sys2 .1 xx	[]	[]
val3	[]	[]	[]

t1 : H5	locus1	locus2	locus3
val1	Sys 1.1 --	Sys 1.2 --	[]
val2	[]	[]	[]
val3	[]	[]	[]

f3 : H5	locus1	locus2	locus3
val1	[]	[]	[]
val2	[]	[]	[]
val3	Sys 1.3 xx	[]	Sys 3.3 xx

t3 : H5	locus1	locus2	locus3
val1	[]	[]	[]
val2	[]	[]	[]
val3	Sys 1.3 xx	[]	Sys 3.3 xx

2.6.2. Activity patterns

$$\begin{tikzcd} \begin{pmatrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{pmatrix} \arrow[r, Rightarrow, "f1:H5"] \arrow[d, Rightarrow, "f3:H5"] & \begin{pmatrix} \blacksquare & \blacksquare & \square \\ \blacksquare & \square & \square \\ \square & \square & \square \end{pmatrix} \\ \begin{pmatrix} \square & \square & \square \\ \square & \square & \square \\ \blacksquare & \square & \blacksquare \end{pmatrix} & \end{tikzcd}$$

: ▮ = processor active (@)

: □ = processor inactive (♀)

$$\text{f1H5}: \begin{array}{|cccc|} \hline \text{PM} & O_1 & O_2 & O_3 \\ M_1 & ♀_{1.1} & ♀_{1.2} & ♀_{1.2} \\ M_2 & ♀_{2.1} & ♀_{2.2} & ♀_{2.3} \\ M_3 & ♀_{3.1} & ♀_{3.2} & ♀_{3.3} \\ \hline \end{array} \longrightarrow \begin{array}{|cccc|} \hline \text{f1(laa)} & O_1 & O_2 & O_3 \\ M_1 & @_{1.1} & @_{1.2} & ♀_{1.3} \\ M_2 & @_{2.1} & ♀_{2.2} & ♀_{2.3} \\ M_3 & ♀_{3.1} & ♀_{3.2} & ♀_{3.3} \\ \hline \end{array}$$

$$\text{SOPS}^{(3,1)}[\text{laa}] = \left(\begin{array}{l} ♀_{1.1} \longrightarrow @_{1.1} \;//\; @_{1.2} \\ ♀_{2.2} \longrightarrow ♀_{2.2} \;\backslash\backslash\backslash\; @_{2.1} \\ ♀_{3.3} \longrightarrow ♀_{3.2} \end{array} \right)$$

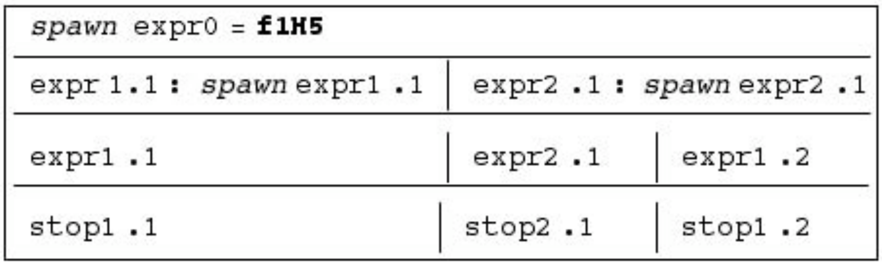
Processor model for f1H5 :
Multi – Processor – System for matrix – distribution of tableaux_forests =
(intra – process : {append, remove, leave}, inter – process : {send, receive}).

input : **f1H5** **question** : **f1H5 = taut ?**	locus1		locus2
process1	$@_{1.1}$: ***input*** : tree(1.1) ***process*** (tree(1.1)) : ID(1.1) = {append, remove} stop (1.1) = **output**(1.1)		$@_{1.2}$: ***receive*** BIF from $@_{2.1}$ ***input*** : tree(1.2) : ***process***(tree(1.2)) : ID(1.2) = {append, re stop(1.2) = **output**(1.

process2	$@_{2.1}$:
	input : tree(2.1)
	process(tree(2.1)) :
	BIF(2.1) = [ID(2.1)
	process(ID(2.1)) :
	{append, remove}
	stop (2.1) = **output**(2.1)
		–
		–
		BIF(1.2)]
		send BIF to $@_{1.2}$:
		leave(2.1) :
		process(tree(1.2))
			[inactive]

process3	[inactive]		[inactive]
output for **f1H5 :** **taut for Sys1 . 1,** **Sys2 . 1, Sys1 . 2 :** **taut f1H5**	**taut** for Sys1 . 1 and Sys2 . 1		**taut** for Syss1 .2

The matrix development as a concurrent procedure with *spawn*

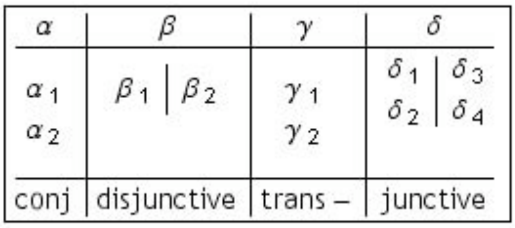


spawn expr0 = **f1H5**		
expr 1.1 : *spawn* expr1 .1	expr2 .1 : *spawn* expr2 .1	
expr1 .1	expr2 .1	expr1 .2
stop1 .1	stop2 .1	stop1 .2

Rudolf Kaehr. FIBONACCI in ConTeXtures, An application 2005
Available at : http : // works.bepress.com / thinkartlab / 21

2.6.3. Extended Smullyan unification

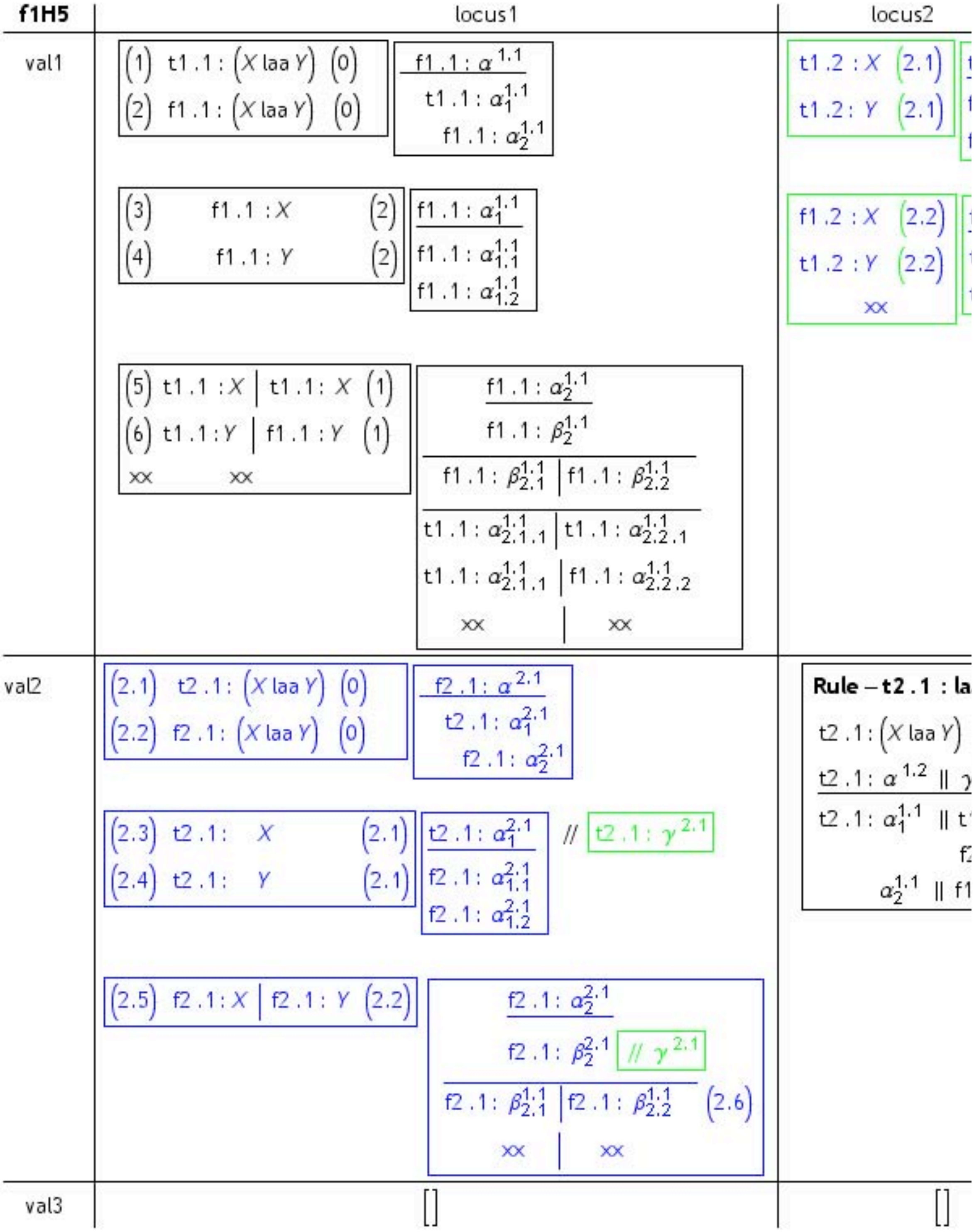
Unification



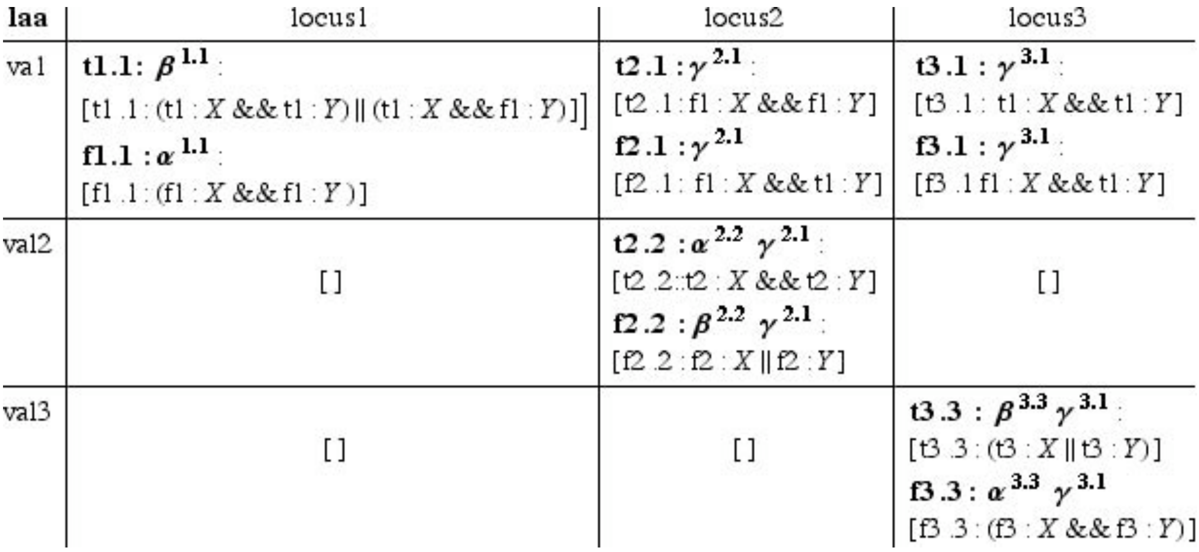
α	β		γ	δ	
α_1 α_2	β_1	β_2	γ_1 γ_2	δ_1 δ_2	δ_3 δ_4
conj	disjunctive		trans –	junctive	

Null

f1H5 = $\left(X \text{ laa } Y\right)$ iij $\left(X \text{ laa} Y\right)$ $\quad \left(0\right)$



f1H5	locus1	locus2
val1	(1) t1.1: (X laa Y) (0) (2) f1.1: (X laa Y) (0) $\underline{f1.1:\alpha^{1.1}}$ $t1.1:\alpha_1^{1.1}$ $f1.1:\alpha_2^{1.1}$ (3) f1.1: X (2) (4) f1.1: Y (2) $\underline{f1.1:\alpha_1^{1.1}}$ $f1.1:\alpha_{1.1}^{1.1}$ $f1.1:\alpha_{1.2}^{1.1}$ (5) t1.1: X | t1.1: X (1) (6) t1.1: Y | f1.1: Y (1) xx xx $\underline{f1.1:\alpha_2^{1.1}}$ $f1.1:\beta_2^{1.1}$ $f1.1:\beta_{2.1}^{1.1}$ | $f1.1:\beta_{2.2}^{1.1}$ $t1.1:\alpha_{2.1.1}^{1.1}$ | $t1.1:\alpha_{2.2.1}^{1.1}$ $t1.1:\alpha_{2.1.1}^{1.1}$ | $f1.1:\alpha_{2.2.2}^{1.1}$ xx | xx	t1.2: X (2.1) t1.2: Y (2.1) t t t f1.2: X (2.2) t1.2: Y (2.2) xx
val2	(2.1) t2.1: (X laa Y) (0) (2.2) f2.1: (X laa Y) (0) $\underline{f2.1:\alpha^{2.1}}$ $t2.1:\alpha_1^{2.1}$ $f2.1:\alpha_2^{2.1}$ (2.3) t2.1: X (2.1) (2.4) t2.1: Y (2.1) $\underline{t2.1:\alpha_1^{2.1}}$ $f2.1:\alpha_{1.1}^{2.1}$ $f2.1:\alpha_{1.2}^{2.1}$ // $t2.1:\gamma^{2.1}$ (2.5) f2.1: X | f2.1: Y (2.2) $\underline{f2.1:\alpha_2^{2.1}}$ $f2.1:\beta_2^{2.1}$ // $\gamma^{2.1}$ $f2.1:\beta_{2.1}^{1.1}$ | $f2.1:\beta_{2.2}^{1.1}$ (2.6) xx | xx	**Rule – t2.1 : la** t2.1: (X laa Y) $\underline{t2.1:\alpha^{1.2}}$ || γ $t2.1:\alpha_1^{1.1}$ || t f $\alpha_2^{1.1}$ || f1
val3	[]	[]



laa	locus1	locus2	locus3
val	**t1.1:** $\beta^{1.1}$: [t1 .1 : (t1 : X && t1 : Y) || (t1 : X && f1 : Y)]] **f1.1 :** $\alpha^{1.1}$: [f1 .1 : (f1 : X && f1 : Y)]	**t2 .1 :** $\gamma^{2.1}$: [t2 .1 : f1 : X && f1 : Y] **f2 .1 :** $\gamma^{2.1}$ [f2 .1 : f1 : X && t1 : Y]	**t3 .1 :** $\gamma^{3.1}$: [t3 .1 : t1 : X && t1 : Y] **f3 .1 :** $\gamma^{3.1}$: [f3 .1 f1 : X && t1 : Y]
val2	[]	**t2 .2 :** $\alpha^{2.2}\ \gamma^{2.1}$: [t2 .2::t2 : X && t2 : Y] **f2 .2 :** $\beta^{2.2}\ \gamma^{2.1}$: [f2 .2 : f2 : X || f2 : Y]	[]
val3	[]	[]	**t3 .3 :** $\beta^{3.3}\gamma^{3.1}$: [t3 .3 : (t3 : X || t3 : Y)] **f3 .3 :** $\alpha^{3.3}\ \gamma^{3.1}$ [f3 .3 : (f3 : X && f3 : Y)]

2.7. Extension mechanism for polycontextural logics

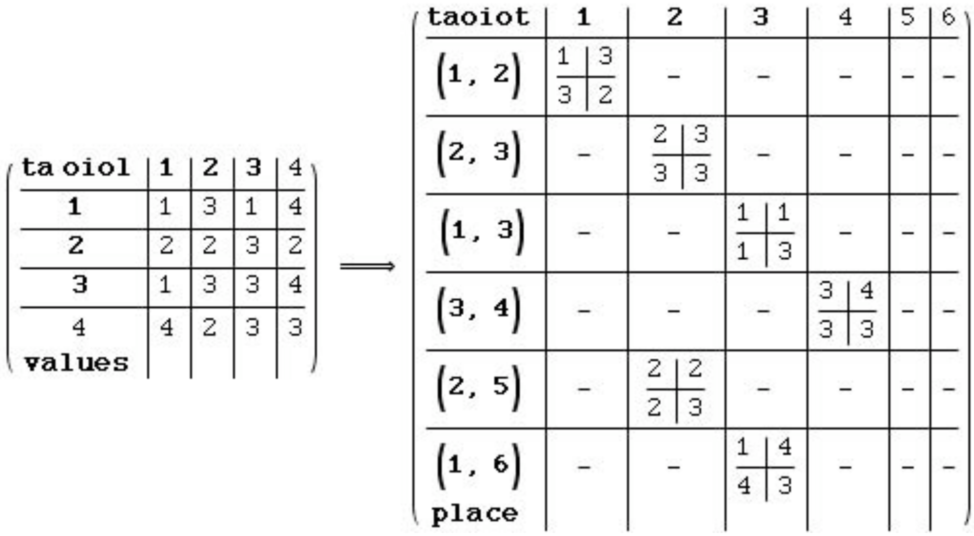
2.7.1. Modular compositions

The inversion of decomposition of logical functions is the action of composition. Both are based on the principle of polycontextural modularity.

2.7.2. Example of a 4-contextural constellation

$$H = X <> \wedge \vee \supset \vee <> Y : (X\ \mathbf{taoiot}\ Y)$$

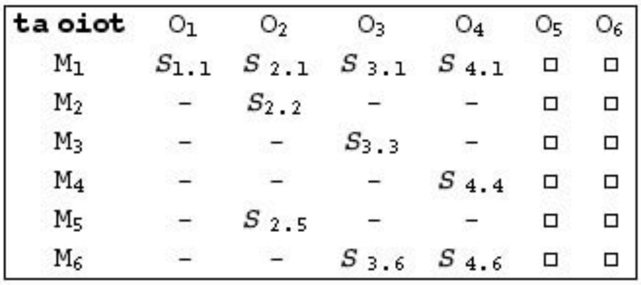
$$H = X^{(3)} \left(\begin{array}{c|c|c} <> & \wedge & \supset \\ \vee & \vee & \\ <> & - & - \end{array}\right) Y^{(3)} = \left(\begin{array}{c|c|c|c} X <>_{1.1} Y & X <>_{2.2} Y & X <>_{3.1} Y & X <>_{4.1} Y \\ \hline - & X \wedge_{2.2} Y & - & - \\ \hline - & - & X \vee_{3.3} Y & - \\ \hline - & - & - & X \supset_{4.4} Y \\ \hline - & X \vee_{2.5} Y & - & - \\ \hline - & - & X <>_{3.6} Y & X <>_{4.6} Y \end{array}\right)$$



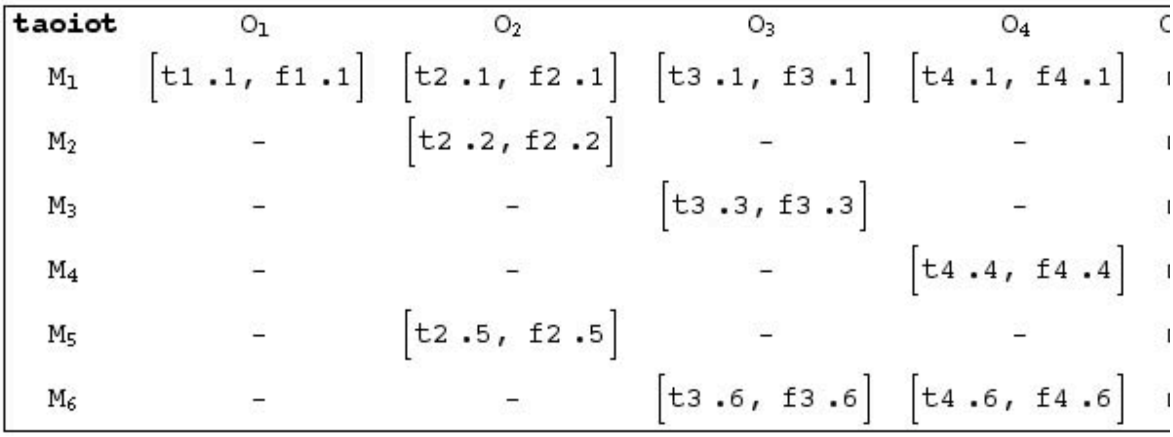
ta oiol	1	2	3	4
1	1	3	1	4
2	2	2	3	2
3	1	3	3	4
4 values	4	2	3	3

⟹

taoiot	1	2	3	4	5	6
(1, 2)	1 | 3 3 | 2	–	–	–	–	–
(2, 3)	–	2 | 3 3 | 3	–	–	–	–
(1, 3)	–	–	1 | 1 1 | 3	–	–	–
(3, 4)	–	–	–	3 | 4 3 | 3	–	–
(2, 5)	–	2 | 2 2 | 3	–	–	–	–
(1, 6) place	–	–	1 | 4 4 | 3	–	–	–

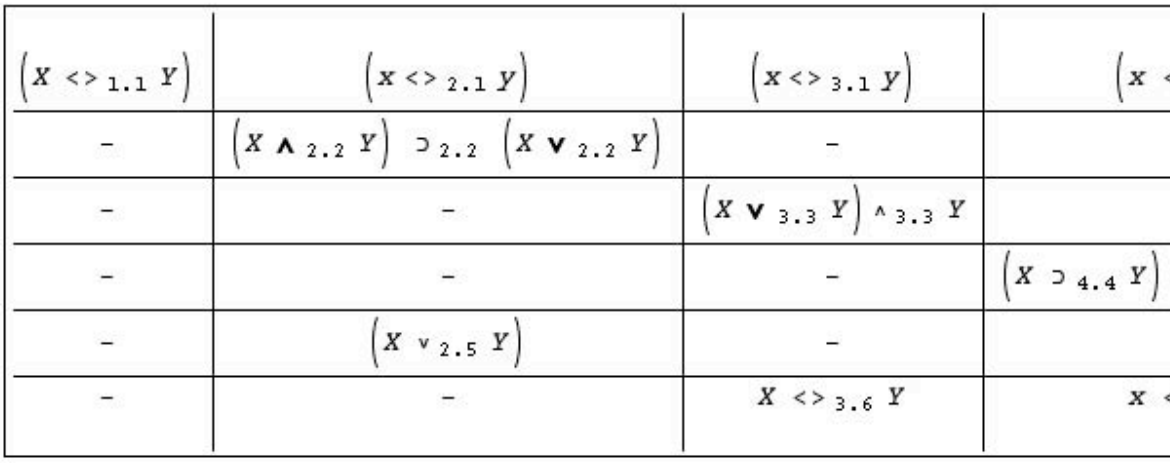


ta oiot	O_1	O_2	O_3	O_4	O_5	O_6
M_1	$S_{1.1}$	$S_{2.1}$	$S_{3.1}$	$S_{4.1}$	□	□
M_2	–	$S_{2.2}$	–	–	□	□
M_3	–	–	$S_{3.3}$	–	□	□
M_4	–	–	–	$S_{4.4}$	□	□
M_5	–	$S_{2.5}$	–	–	□	□
M_6	–	–	$S_{3.6}$	$S_{4.6}$	□	□



taoiot	O_1	O_2	O_3	O_4	O
M_1	[t1 .1, f1 .1]	[t2 .1, f2 .1]	[t3 .1, f3 .1]	[t4 .1, f4 .1]	
M_2	–	[t2 .2, f2 .2]	–	–	
M_3	–	–	[t3 .3, f3 .3]	–	
M_4	–	–	–	[t4 .4, f4 .4]	
M_5	–	[t2 .5, f2 .5]	–	–	
M_6	–	–	[t3 .6, f3 .6]	[t4 .6, f4 .6]	

Distributed formulas :



$(X <>_{1.1} Y)$	$(x <>_{2.1} y)$	$(x <>_{3.1} y)$	$(x <$
–	$(X \wedge_{2.2} Y) \supset_{2.2} (X \vee_{2.2} Y)$	–	
–	–	$(X \vee_{3.3} Y) \wedge_{3.3} Y$	
–	–	–	$(X \supset_{4.4} Y)$
–	$(X \vee_{2.5} Y)$	–	
–	–	$X <>_{3.6} Y$	$x <$

2.8. Metatheoretical laws for polycontextural logics

2.8.1. Distributed duality laws

Metatheory of polycontextural logics is just in its beginnings. Nevertheless, there are many insights, elaborations and results been published in earlier work, beginning with my *Materialien* 1976.

An important metatheoretical principle in classical formal theories and logics is the *"duality principle"*.

"The **Duality Principle for Categories** states

Whenever a property P holds for all categories, then the property P^{op} holds for all categories." (Herrlich, 2004, p. 27)
http://katmat.math.uni-bremen.de/acc/acc.pdf

Mario José Caccamo, *A Formal Calculus for Categories*, gives a classical definition of Herrlich's category-theoretical principle of *"get two for the price of one"*:

"Duality
A category *C* gives rise to another category by just reversing all the arrows. This new category called the dual or opposite is denoted by C^{op}. The composition of the arrows in C^{op} can be expressed in terms of the arrows in *C* by reading the composition
in the reverse order: *g o f* in C^{op} uniquely corresponds to *f o g* in *C*. In the same spirit, a functor *F : C -> D* can be *dualised* to obtain a functor $F^{op}: C^{op} \to D^{op}$.
More generally a statement *S* involving a category *C* automatically gives a dual statement S^{op} obtained by reversing all the arrows. This is known as the *duality principle*."
http://www.brics.dk/DS/03/7/BRICS-DS-03-7.pdf

A simple application of the duality principle is known in classical logic as the *De Morgan* laws. As much as logical functions are disseminated, *De Morgan* too is not escaping its matrix-dissemination (Kaehr, *Materialien,* 1976).

2.8.2. Hierarchic and heterarchic understanding of duality

The hierarchy of duality
Also duality means that the dual-parts are of equal value (meaning, relevance) there is nebvertheless a clear hierarchy set between the first and the second of a duality.
"Buy one, get one for free": this implies a succession and therefore a herarchy. And equally by dual 'inversion':
"Get one for free, buy one".

$C = (C^{op})^{op}$:
$C \dashrightarrow C^{op} \dashrightarrow (C^{op})^{op} \dashrightarrow C$, equally,
$C^{op} \dashrightarrow (C^{op})^{op} \dashrightarrow C \dashrightarrow C^{op}$.

The heterarchy of duality

$$C \amalg C^{op} \dashrightarrow \begin{pmatrix} \begin{pmatrix} C \\ \amalg \\ C^{op} \end{pmatrix} \\ \amalg \\ [C, C^{op}] \end{pmatrix}.$$

The categories or processes of "*buy*" and "*get*" are interacting simultaneously together. Its simultaneity is reflected or conceived from a third position as a triadicity of "neither-nor" or of "both-and" as $[C, C^{op}]$.

Quadralectics of the duality
A further concretization of the duality of C and C^{op} is achieved with the full '*quadralectics*' of $Q = (C, C^{op}, [C, C^{op}], \{C, C^{op}\})$, with
C : position,
C^{op} : opposition,
$[C, C^{op}]$: neither-nor, rejection
$\{C, C^{op}\}$: both-and, acception.
http://www.thinkartlab.com/pkl/lola/Quadralectic%20Diamonds

/Quadralectic%20Diamonds.pdf

2.8.3. Trans-contextural duality principle

But a much more interesting question, based on the insights of categorial interchangeability, arises: *Is there a duality between composition and dissemination?* That is, is there a kind of a duality between the "operators" "o" and "∐"? A positive answer would establish a new kind of duality:the *trans-contextural duality* between composition and dissemination. A further transclassical duality principle is established between *transpositional* and *replicational* combinations of polycontextural complexions.

As far as there is systematically no "*position*", *C*, without its (dual) "*opposition*", C^{op}, and *vice versa*, there are no compositions without disseminations, and vice versa, there are no disseminations without compositions. Hence, there is a systematic, i.e. a meta-theoretical duality established between intra-contextural compositions (combinations) and inter-contextural disseminations of a polycontextural complexion.

This kind of trans-contextural duality is staged by the interplay of *proemiality*.

Trans-duality of combination and dissemination:

$$\text{D (o, ∐)} = \begin{pmatrix} \begin{pmatrix} \circ \\ \amalg \\ \circ \end{pmatrix} \\ \amalg \\ \circ \end{pmatrix} <==> \begin{pmatrix} \begin{pmatrix} \amalg \\ \circ \\ \amalg \end{pmatrix} \\ \circ \\ \amalg \end{pmatrix} .$$

Trans-duality is not less self-referential, i.e. proemial, than *De Morgan* duality but well-reflected in its super-additivity.

Duality in the mono-contextural sense as the *duality principle* is achieved by a dissemination into itself, i.e. a option or *positioning* (Setzung, Fichte) of the epistemé to uniqueness.

$$\mathcal{D}\left(\circ,\ \amalg = 1\right) = d\left(\circ\right) = \left(\circ\right):$$
$$d\left(\circ\right) \equiv d\left(X\right) \circ d\left(Y\right) = d\left(X \circ Y\right) : \text{mono} - \text{contextural duality principle.}$$

3. Remarks on a more mathematical formalization

3.1. Pfalzgraf's fiber bundle approach: linear-indexed fibered bundles

3.1.1. Free and Derived Logical Fiberings

Some citations:

"*Logical Fibering can be seen as a methodology in symbolic computation. We have thus a direct link between symbolic computation and AI.*"
http://www.tmrfindia.org/ijcsa/v9i24.pdf

A Logical Fibering is a triple (E, π, B). E is called total space, B base space, and π : E --> B is called projection map. For b ∈ B the preimage set of all x ∈ E such that π (x) = b, is called Fiber over b (Pfalzgraf, 1991).

"The global set of truth values, hence, is Ω = {a, b, c}. This global logic shall be decomposed to the three classical two valued logics L_i, i=1,2,3 (with local values {T_i,

F_{i}})The total universe of available local values is $\Omega^3 = \{T_1, T_2, T_3, F_1, F_2, F_3\}$. The decomposition is done by finding suitable equivalence relations on Ω^3 such that the set of residue classes $\Omega := \Omega^3/\equiv$ yields the global truth value set $\Omega = \{a, b, c\}$. The respective logical connectives (the "AND" operation in the current case) shall be implemented by local logical connectives in the subsystems.

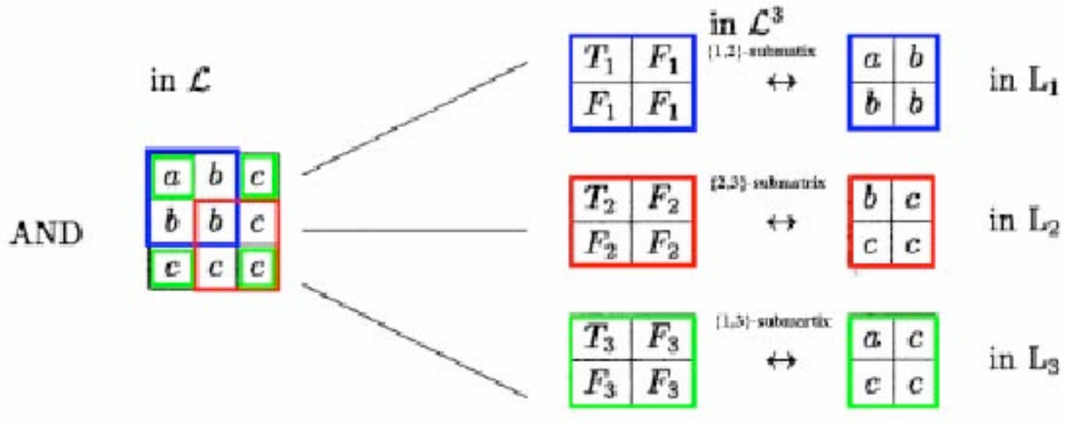


Figure 2-13: Decomposition method ("AND" operation on 3-valued logic)

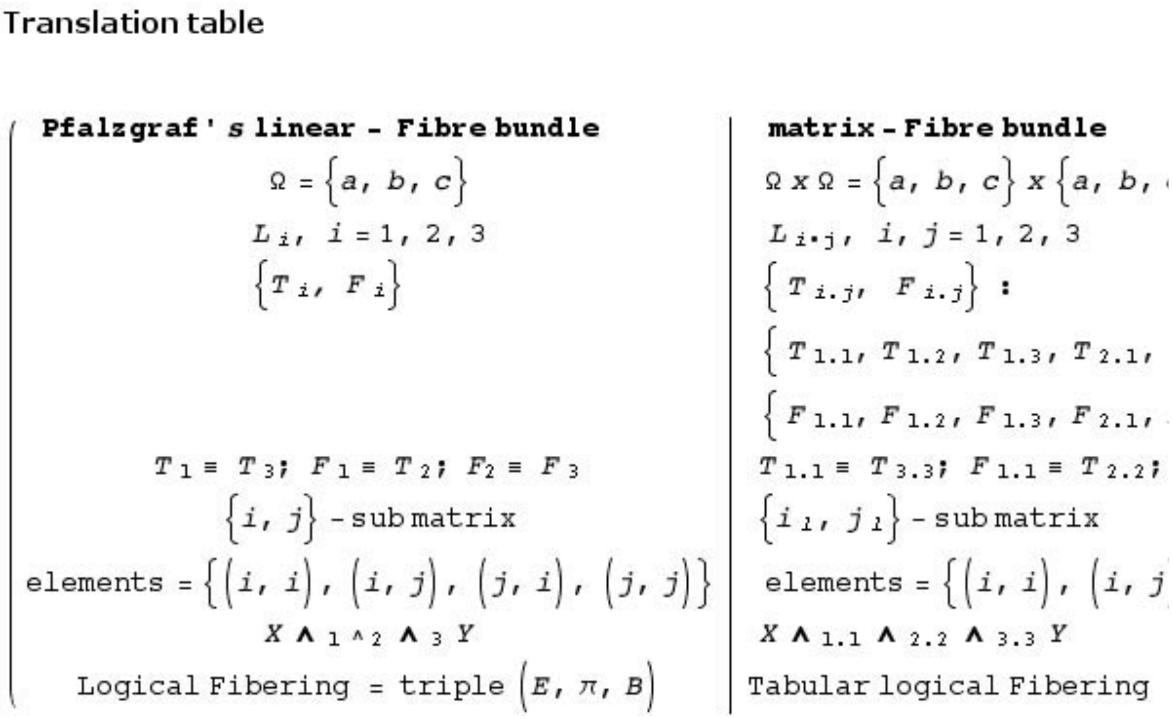
"The decomposition method is illustrated by Figure 2-13 (adapted from [PFA95]). Firstly, from thetruth table in *L* we derive three sub matrixes (defined by their limits): the {1,2}-sub matrix, the {2, 3}-sub matrix and the {1, 3}-sub matrix, considering that the {i, j}-sub matrix consists of the elements {(i, i), (i, j), (j, i), (j, j)}.

In the above case it may be observed that each sub matrix gives the values of a classical conjunction truth table, which may be expressed by $x_{\mathrm{i}} \wedge y_i$ respectively. The present relation may furthermore be modeled by the mediation scheme in Figure 2-11 as the encountered relation follows $T_1 \equiv T_3$; $F_1 \equiv T_2$; $F_2 \equiv F_3$ (see above). Finally, it may be expressed by the bivariate operation $X_{\wedge\wedge\wedge} Y$ [PFA95]."
http://www.iks.kit.edu/fileadmin/User/Calmet/dissertationen/diss_Schneider.pdf

"**Remark**: The compatibility condition with respect to the three suboperations (submatrices) can be expressed as follows (cf. [20]): The three 2x2-matrices have to be merged to a 3x3-matrix scheme along the diagonal of the 3x3-matrix such that the corresponding diagonal elements match (i.e. the 2x2-matrices are the suitable submatrices). In this sense our decomposition method is the reverse process to this merge."
http://www.rac.es/ficheros/doc/00158.pdf
[20]: J. Pfalzgraf. *Logical fiberings and polycontextural systems*. In Fundamentals of Artificial Intelligence Research,
Ph.Jorrand, J.Kelemen (eds.). Lecture Notes in Computer Science 535, Subseries in AI, Springer Verlag, 1991.

3.2. Matrix formalization: tabular-indexed fibre bundles

3.2.1. Notes on a natural extension of Pfalzgraf's approach



Translation table

Pfalzgraf ' s linear - Fibre bundle	matrix - Fibre bundle
$\Omega = \{a, b, c\}$	$\Omega \, x \, \Omega = \{a, b, c\} \, x \, \{a, b,$
$L_i, \ i = 1, 2, 3$	$L_{i \cdot j}, \ i, j = 1, 2, 3$
$\{T_i, \ F_i\}$	$\{T_{i.j}, \ F_{i.j}\}$:
	$\{T_{1.1}, T_{1.2}, T_{1.3}, T_{2.1},$
	$\{F_{1.1}, F_{1.2}, F_{1.3}, F_{2.1},$
$T_1 \equiv T_3; \ F_1 \equiv T_2; \ F_2 \equiv F_3$	$T_{1.1} \equiv T_{3.3}; \ F_{1.1} \equiv T_{2.2};$
$\{i, j\}$ - sub matrix	$\{i_1, j_1\}$ - sub matrix
elements $= \{(i, i), (i, j), (j, i), (j, j)\}$	elements $= \{(i, i), (i, j$
$X \wedge_1 \wedge_2 \wedge_3 Y$	$X \wedge_{1.1} \wedge_{2.2} \wedge_{3.3} Y$
Logical Fibering = triple (E, π, B)	Tabular logical Fibering

"A Logical Fibering is a triple (E, п, B). E is called *total* space, B *base* space, and п : E --> B is called *projection* map. For b ∊ B the preimage set of all x ∊ E such that п (x) = b, is called *Fiber* over b (Pfalzgraf, 1991)."

3.2.2. Tabular examples

Total functions: $\mathbf{F_{tot}}$: (i, j) --> (i, j)

F_{oao}: $o_{1.1}: (1, 2) \longrightarrow (1, 2; 1)$

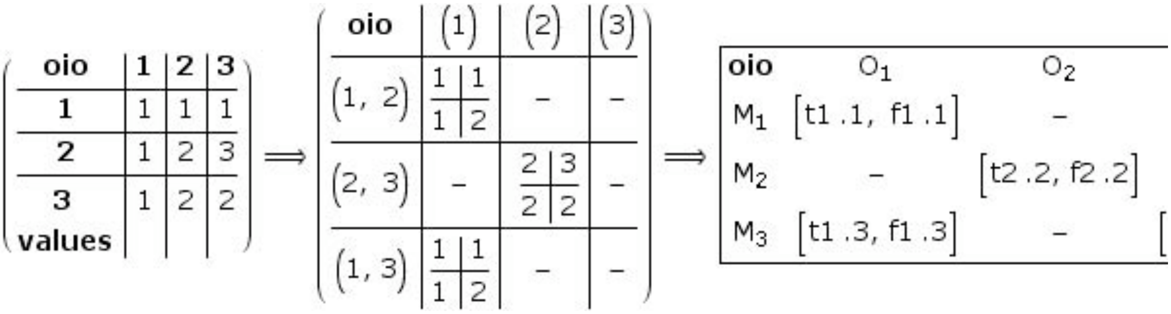
$a_{2.2}: (2, 3) \longrightarrow (2, 3; 2)$

$o_{3.3}: (1, 3) \longrightarrow (1, 3; 3).$

$$\left(\begin{array}{c|c|c|c} \text{oao} & 1 & 2 & 3 \\ \hline 1 & 1 & 1 & 1 \\ \hline 2 & 1 & 2 & 3 \\ \hline 3 & 1 & 3 & 3 \\ \text{values} & & & \end{array}\right) \Longrightarrow \left(\begin{array}{c|c|c|c} \text{oao} & 1 & 2 & 3 \\ \hline (1, \ 2) & \begin{array}{c|c} 1 & 1 \\ \hline 1 & 2 \end{array} & - & - \\ \hline (2, \ 3) & - & \begin{array}{c|c} 2 & 3 \\ \hline 3 & 3 \end{array} & - \\ \hline (1, \ 3) & - & - & \begin{array}{c|c} 1 & 1 \\ \hline 1 & 3 \end{array} \end{array}\right) \Longrightarrow$$

oao	O_1	O_2	O_3
M_1	[t1 .1, f1 .1]	-	-
M_2	-	[t2 .2, f2 .2]	-
M_3	-	-	[t3 .3, f3 .3]

$\mathbf{F_{oio}}$:
$o_{1.1}$: (1, 2) --> (1, 2;1)
$a_{2.2}$: (2, 3) --> (2, 3;2)
$o_{3.3}$: (1, 3) --> (1, 3;1).



$$\left(\begin{array}{c|c|c|c} \textbf{oio} & \textbf{1} & \textbf{2} & \textbf{3} \\ \hline \textbf{1} & 1 & 1 & 1 \\ \hline \textbf{2} & 1 & 2 & 3 \\ \hline \textbf{3} & 1 & 2 & 2 \\ \textbf{values} & & & \end{array}\right) \Longrightarrow \left(\begin{array}{c|c|c|c} \textbf{oio} & (1) & (2) & (3) \\ \hline (1,\ 2) & \begin{array}{c|c} 1 & 1 \\ \hline 1 & 2 \end{array} & - & - \\ \hline (2,\ 3) & - & \begin{array}{c|c} 2 & 3 \\ \hline 2 & 2 \end{array} & - \\ \hline (1,\ 3) & \begin{array}{c|c} 1 & 1 \\ \hline 1 & 2 \end{array} & - & - \end{array}\right) \Longrightarrow \boxed{\begin{array}{llll} \textbf{oio} & O_1 & O_2 & \\ M_1 & [t1\ .1,\ f1\ .1] & - & \\ M_2 & - & [t2\ .2,\ f2\ .2] & \\ M_3 & [t1\ .3,\ f1\ .3] & - & [\end{array}}$$

Partial functions F_{part}

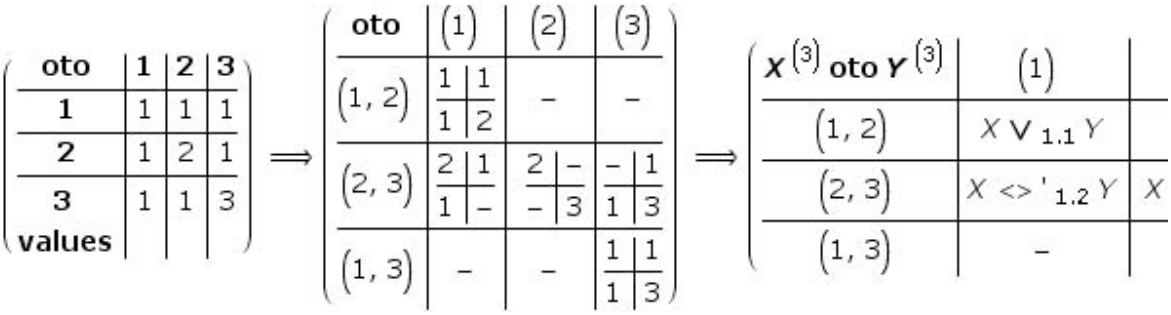
Transjunctions: F_{trans}: (i, j) --> ((i, j_1), (i,j_2),..., (i,j_n)

F_{oto}:

$o_{1.1}$: (1, 2) --> (1,2;1)

$t_{2.2}$: (2, 3) --> (2, 3;1), (2, 3;2), (2, 3;3)),

$o_{3.3}$: (1, 3) --> (1, 3;3)



$$\left(\begin{array}{c|c|c|c} \textbf{oto} & \textbf{1} & \textbf{2} & \textbf{3} \\ \hline \textbf{1} & 1 & 1 & 1 \\ \hline \textbf{2} & 1 & 2 & 1 \\ \hline \textbf{3} & 1 & 1 & 3 \\ \textbf{values} & & & \end{array}\right) \Longrightarrow \left(\begin{array}{c|c|c|c} \textbf{oto} & (1) & (2) & (3) \\ \hline (1,\ 2) & \begin{array}{c|c} 1 & 1 \\ \hline 1 & 2 \end{array} & - & - \\ \hline (2,\ 3) & \begin{array}{c|c} 2 & 1 \\ \hline 1 & - \end{array} & \begin{array}{c|c} 2 & - \\ \hline - & 3 \end{array} & \begin{array}{c|c} - & 1 \\ \hline 1 & 3 \end{array} \\ \hline (1,\ 3) & - & - & \begin{array}{c|c} 1 & 1 \\ \hline 1 & 3 \end{array} \end{array}\right) \Longrightarrow \left(\begin{array}{c|c|c} \boldsymbol{X}^{(3)}\ \textbf{oto}\ \boldsymbol{Y}^{(3)} & (1) & \\ \hline (1,\ 2) & X \vee_{1.1} Y & \\ \hline (2,\ 3) & X <>'_{1.2} Y & X \\ \hline (1,\ 3) & - & \end{array}\right.$$

4. Logic programming in the field

4.1. Logic and programming

"*Many AI practitioners have shied away from using standard logic programming languages as the knowledge representation language in AI systems, partly due to the diffculty associated with representing uncertain, incomplete, or conflicting information in such languages. The root of these diffculties is the inherent limitations of first-order logic as the basis of the standard logic programming systems. One such limitation is the monotoniciy of first-order logic which makes it unsuitable as a mechanism for revisable reasoning. Another important limitation stems from the all-or-nothing nature of classical first-order logic: statements can be evaluated to be completely true or completely false. Intelligent agents, however, must often deal with information which is uncertain, or incomplete.*

"*The above brief discussion suggests that such systems must have two common characteristics: they must rely on the expressive power of an underlying* multi-valued *logic which can deal with contradictory as well as incomplete or uncertain information, and secondly, such systems should be able to interpret statements not only based on their truth or falsity, but also based on some* measure of the knowledge *or information contained within those statements.*"
Bamshad Mobasher et al, Algebraic Semantics for Knowledge-based Logic Programs
http://maya.cs.depaul.edu/mobasher/papers/wudds94.pdf

The statement in the citation "*interpret statements not only based on their truth or falsity, but also based on some measure of the knowledge or information contained within those statements.*" hints to another chance of modeling, especially if connected with its first part "*they must rely on the expressive power of an underlying* multi-valued *logic which can deal with contradictory as well as incomplete or uncertain information*".

It seems not to be a too wild decision to opt for a polycontextural interpretation of the proposed situation.
The *"the measure of knowledge"* shall correspond to the measure of the *dissemiantion* of contextures. And the *"power of multi-valued logic"* gets a modeling by the polycontextural logic of the disseminated contextures. But that's not working without some subversion. Contextures are corresponding to "*workspaces*" in the sense of Fitting's theories of logic and computation. And, obviously, there is just one and only one 'workspace' accepted by Fitting's approach.

Hence, each contexture contains 'knowledge', and each contexture is ruled by its logic. The dissemination, i.e. distribution and mediation, of logics is defined by the pylycontextural 'multi-valuedness' of the complexion of contextural logics. The minimum logic of a contexture is a 2-valued logic. The 'multi-valuedness' of the complexion is build by the mediation of the distributed 2-valued logics. Negation in polycontextural logics is intra-contextural negation and inter-contextural permutation.
M Fitting, Negation As Refutation
http://comet.lehman.cuny.edu/fitting/bookspapers/pdf/papers/NegAsRef.pdf

4.2. Polycontextural logical programming: The SCHELLING turn

"Seine [Schelling, rk] These, es gäbe weder die ´eine Wahrheit´ noch die ´eine Wirklichkeit', sondern das Universum sei vielmehr als ein ´bewegliches Gewebe´ aufeinander nicht zurückführbarer Einzelwelten zu denken, formulierte die entscheidende Aufgabe der
Philosophie der Zukunft: eine Theorie bereitzustellen, die es gestattet, die Strukturgesetze des organischen Zusammenwirkens der je für sich organisierten Teilwelten aufzudecken." Gotthard Günther, Nachlass „GG", 15. Juni 1980

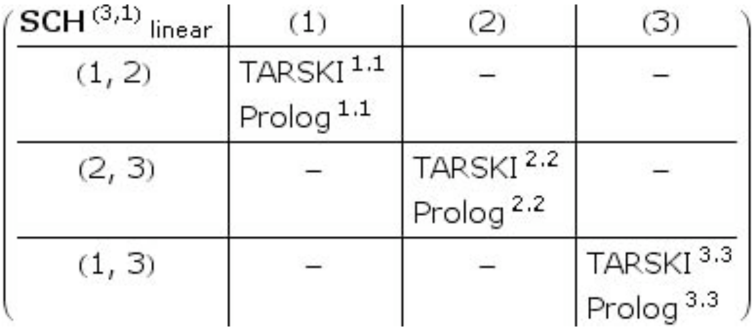
The universe of polycontextural programming is a poly-verse. A poly-verse is an interactional complexion of universes (general domains) of logic and logical programming.

In contrast to a TARSKI-world, such a polyverse of interactivity is called a SCHELLING-world. Tarski-worlds goes back to Leibniz, and Schelling-worlds had been discovered for logic by Gotthard Gunther.

A TARSKI-world is defined by classical logic and programmed by logical programming languages like Prolog.

A simple, and first-step implementation of the dynamic SCHELLING-worlds and their polycontextural logics has been achieved with the tabular matrix-approach of the dissemination, i.e. distribution and mediation, of formal languages, logics, semiotics and arithmetics.
http://memristors.memristics.com/Mereotopology/Mereotopology%20and%20Polycontexturality.pdf
http://memristors.memristics.com/Polyverses/Polyverses.pdf

SCHELLING$^{(3, 3)}$ =



$\textbf{SCH}^{(3,1)}_{\text{linear}}$	(1)	(2)	(3)
(1, 2)	TARSKI$^{1.1}$ Prolog$^{1.1}$	–	–
(2, 3)	–	TARSKI$^{2.2}$ Prolog$^{2.2}$	–
(1, 3)	–	–	TARSKI$^{3.3}$ Prolog$^{3.3}$

5. Tabular dissemination of kenomic cellular automata

5.1. Explicit tabular notation of distributed CAs

As shown in earlier papers, composed cellular automata might be constructed as mediations of classical elementary CAs.
http://memristors.memristics.com/CA-Compositions
/Memristive%20Cellular%20Automata%20Compositions.pdf

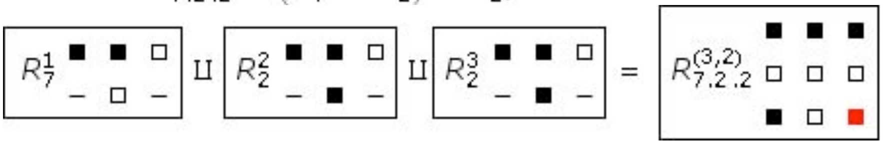
5.1.1. Homogeneous compositions

General composition scheme for mediation

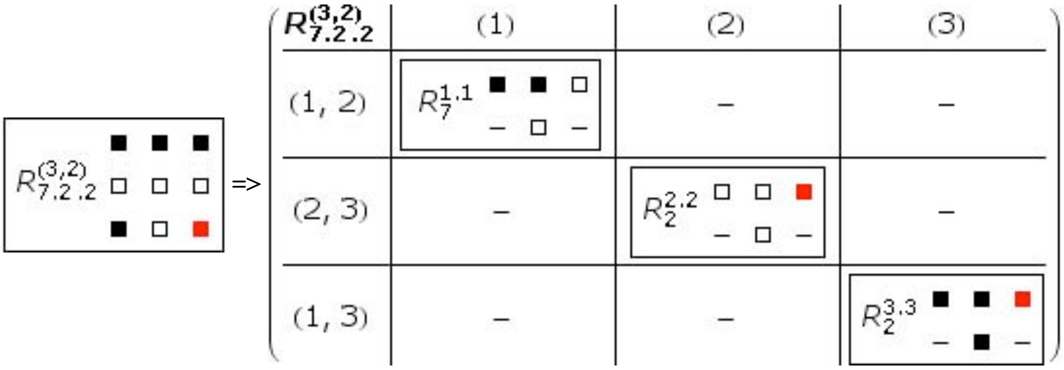
$$\boxed{S^{1.1}\begin{array}{ccc} c2 & c1 & c3 \\ - & c4 & - \end{array}}, \boxed{S^{2.2}\begin{array}{ccc} d2 & d1 & d3 \\ - & d4 & - \end{array}}, \boxed{S^{3.3}\begin{array}{ccc} e2 & e1 & e3 \\ - & e4 & - \end{array}} \Rightarrow \boxed{\begin{array}{ccc} c2 = e2 & c1 & e1 \\ c4 & c3 = d2 & d1 \\ e4 & d4 & d3 = e3 \end{array}}$$

Homogeneous compositions

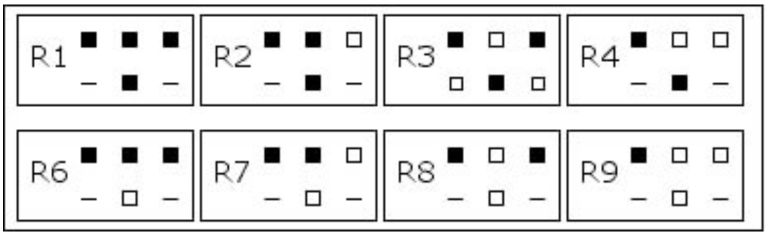
$CA^{(3,1)}: R^{1.1.3}_{7.2.2} = (R^1_7 \amalg R^2_2) \amalg R^3_2$, "∐" : mediation



$R^{(3,2)}_{7.2.2}$	■c2; e2	■c1	■e1
	□c4	□c3; d2	□d1
	■e4	□d4	🟥d3; e3



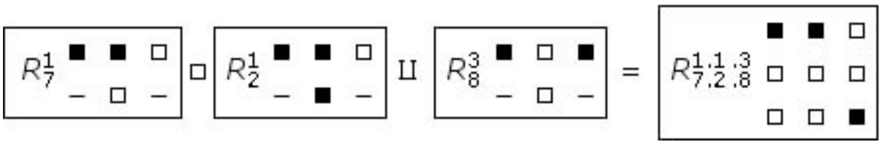
5.1.2. Replicational distributions



Replicational example

$CA^{(3,1)}: R^{1.1.3}_{7.2.8}$	S^1_1	S^1_2	S^3_3
1	rule7	rule2	rule8

$CA^{(3,1)}: R^{1.1.3}_{7.2.8} = (R^1_7 \,\square\, R^1_2) \amalg R^3_8$, "□" : replication



Tabular notation:

$CA^{(3,1)}: R^{1.1.3}_{7.2.8} = (R^1_7 \,\square\, R^1_2) \amalg R^3_8$

$R^{1.1.3}_{7.2.8}$			
	■	■	□
	□	□	□
	□	□	■

=>

$R^{1.1.3}_{7.2.8}$	(1)	(2)	(3)
(1, 2)	$R^{1.1}_{7}$ ■ ■ □ – □ –	–	–
(2, 3)	$R^{1.2}_{2}$ □ □ ■ – □ –	–	–
(1, 3)	–	–	$R^{1.3}_{8}$ ■ □ ■ – □ –

5.1.3. Transpositional distributions

Some transpositional rules

R11	R12	R13	R14	R15
■ ■ □ – 🟥 –	■ □ ■ – 🟥 –	■ □ □ – 🟥 –	■ □ 🟥 – 🟥 –	■ □ 🟥 – 🟦 –

General scheme

c_3	c_1	c_2
–	c_4	–

⟹

R12			
	c_3 🟥	c_1 ■	c_2 🟥
	–	c_4 □	–

$=_{MG}$

R12			
	■	□	■
	–	🟥	–

.

Subsystems : S_1^1 = {■, □}, S_1^2 = {🟥, □}, S_1^3 = {■, 🟥}

R12	S_1^1	S_1^2	S_1^3
1	■	–	■
2	–	🟥	🟥
3	–	🟥	🟥
4	□	□	–

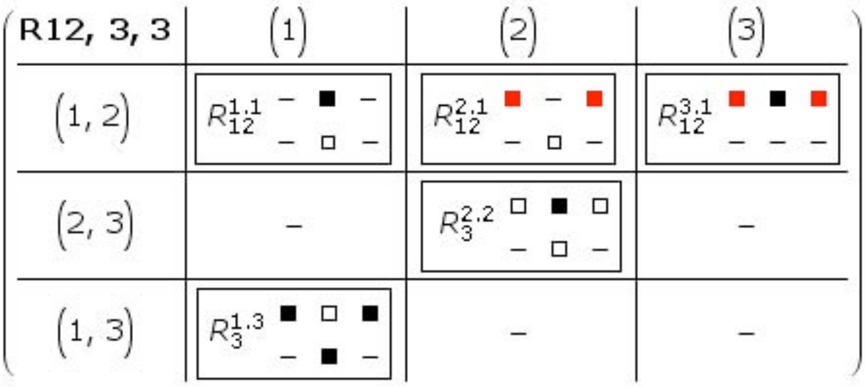
⟺

R12	$S_1^{1.2.3}$
1	■ **1.3**
2	🟥 **2.3**
3	🟥 **2.3**
4	□ **1.2**

⟺

R12			
	🟥	■	🟥
	–	□	–

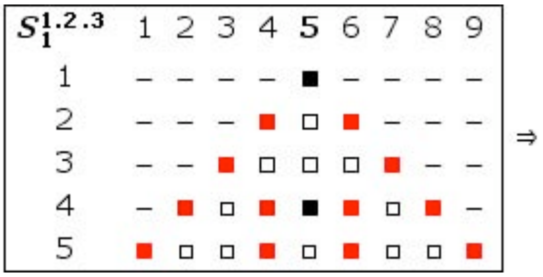
R12				=	$R^{1.1}_{12}$				$R^{2.1}_{12}$				$R^{3.1}_{12}$			
	🟥	■	🟥			–	■	–		🟥	–	🟥		🟥	■	🟥
	–	□	–			–	□	–		–	□	–		–	–	–



R12, 3, 3	(1)	(2)	(3)
(1, 2)	$R_{12}^{1.1}$ – ■ – – □ –	$R_{12}^{2.1}$ 🟥 – 🟥 – □ –	$R_{12}^{3.1}$ 🟥 ■ 🟥 – – –
(2, 3)	–	$R_3^{2.2}$ □ ■ □ – □ –	–
(1, 3)	$R_3^{1.3}$ ■ □ ■ – ■ –	–	–

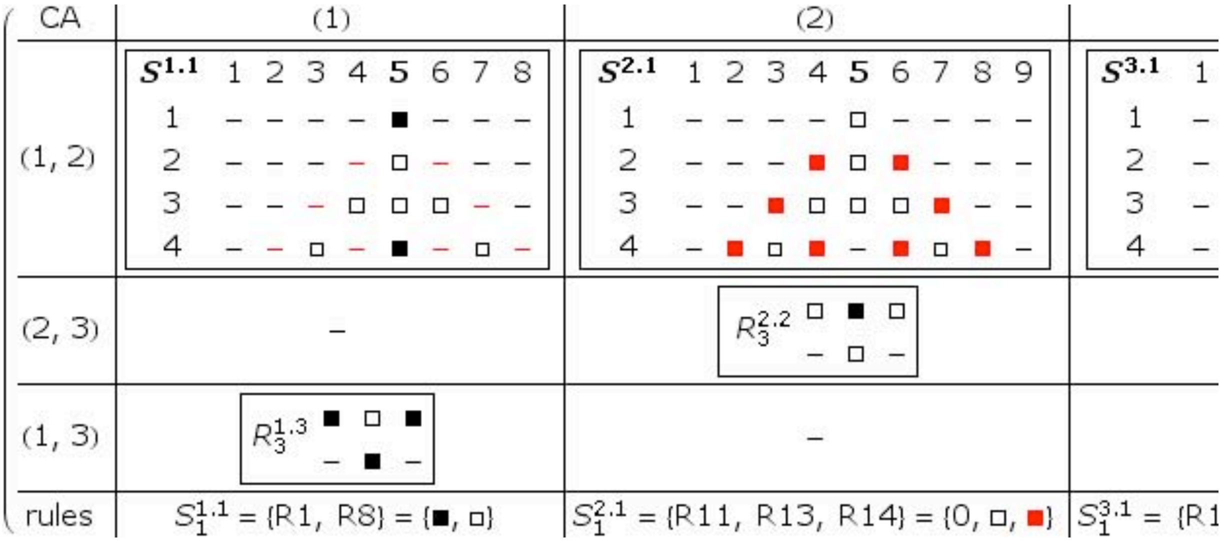
Example

Tabular notation of kenoCA=1.8.11.13.14 = [10222]



$S_1^{1.2.3}$	1	2	3	4	5	6	7	8	9
1	–	–	–	–	■	–	–	–	–
2	–	–	–	🟥	□	🟥	–	–	–
3	–	–	🟥	□	□	□	🟥	–	–
4	–	🟥	□	🟥	■	🟥	□	🟥	–
5	🟥	□	□	🟥	□	🟥	□	□	🟥

⇒



CA	(1)	(2)	
(1, 2)	$S^{1.1}$ 1 2 3 4 5 6 7 8 1 – – – – ■ – – – 2 – – – – □ – – – 3 – – – □ □ □ – – 4 – – □ – ■ – □ –	$S^{2.1}$ 1 2 3 4 5 6 7 8 9 1 – – – – □ – – – – 2 – – – 🟥 □ 🟥 – – – 3 – – 🟥 □ □ □ 🟥 – – 4 – 🟥 □ 🟥 – 🟥 □ 🟥 –	$S^{3.1}$ 1 1 – 2 – 3 – 4 –
(2, 3)	–	$R_3^{2.2}$ □ ■ □ – □ –	
(1, 3)	$R_3^{1.3}$ ■ □ ■ – ■ –	–	
rules	$S_1^{1.1}$ = {R1, R8} = {■, □}	$S_1^{2.1}$ = {R11, R13, R14} = {0, □, 🟥}	$S_1^{3.1}$ = {R1

Notes

1 **LOLA-proof**

f1: ((X laa Y) iij (X laa Y))

```
                            |
        +---------------   ///  ---------------+
        |                                      |
        |                       +------------------------------+
+------ && ------+              |              |               |
|                |              |      +------ && ------+      |
t1: (X laa Y)    f1: (X laa Y)       |      |                |      |
                                     |t2: (X laa Y)    f2: (X laa Y)|
                                     +------------------------------+
                                                    |
                                 +----------------  ///  ----------------+
                                 |                                       |
                                 |                       +----------------------------
                +---------------- &&
------+               |              |               |
                |                         |              |      +------ &&
------+       |
                |                         |              |      |                |
         +------- || ------+               +-- && --+          |t2: (X laa Y)    f2: (X laa
Y)|
         |                 |             |        |          +----------------------------
         |                 |           f1: X    f1:
Y
    +-- && --+         +-- &&
--+
    |        |        |        |

t1: X    t1: Y    t1: X    f1:
Y
                                                                              |
                                             +----------------------  ///  -----------
                                             |
                                             |                                +----------
                      +---------------------- &&
---------+                  |              |               |
                      |                                   |                 |      +---
&& ------+      |
                      |                                   |                 |      |
         +---------- || ---------+                +---- && ---+           |t2: (X
laa Y)    f2: (X laa Y)|
          |                       |                 |          |          +----------
          |                       |              +------+    +------+
    +---- && ---+           +---- &&
---+        |[f1:X]|    |[f1:Y]|
    |           |           |           |        +------+    +------+
+------+    +------+    +------+    +------+
|[t1:X]|    |[t1:Y]|    |[t1:X]|    |[f1:Y]|
+------++------+   +------+    +------+

                                                    |
                                 +----------------  ///  ----------------+
                                 |                                       |
                                 |                       +----------------------------
                +---------------- &&
------+               |              |               |
                |                         |              |      +------ &&
------+       |
                |                    +------------+      |      |                |
         +------- || ------+         |[f1:X, f1:Y]|      |t2: (X laa Y)    f2: (X laa
Y)|
         |                 |         +------------+      +----------------------------
+------------+    +------------+
|[t1:X, t1:Y]|    |[t1:X, f1:Y]|
+------------+    +------------+
                                  |
                +----------------  ///  ----------------+
                |                                       |
                |                   +------------------------------+
         +------- && ------+        |                   |          |
         |                 |        |      +------- && -------+      |
+------------+    +------------+    |      |                |      |
|[t1:X, f1:Y]|    |[f1:X, f1:Y]|       |t2: (X laa Y)    f2: (X laa Y)|
|[t1:X, t1:Y]|    +------------+       +------------------------------+
+------------+
      |
+-  ///  ---------------+
|                       |
+-+       +------------------------------+
|$|       |              |               |
+-+       |      +------ && ------+      |
          |      |                |      |
          |t2: (X laa Y)    f2: (X laa Y)|
          +------------------------------+

      |
+-  ///  ---------------------------------------+
|                                               |
+-+       +--------------------------------------------------------------------------
|$|       |                                      |
+-+       |                 +------------------- &&
-----------------+                   |
          |                 |                                        |
          |                 |                                        |
          |      +-------  ///  -------+                  +-------  ///  -------+
          |      |                     |                  |                     |
          |      |              +--------------+          |              +-----------
          |  +-- && --+         |      |       |      +-- ||
--+         |      |       ||
          |  |        |         |  +-- && --+  |      |        |         |  +-- &&
--+  ||
          |t2: X    t2: Y       |  |        |  |    f2: X    f2:
Y       |  |        |  ||
          |                     |f1: X    f1: Y|                         |f1:
X    t1: Y||
          |                     +--------------+                         +-----------
          +--------------------------------------------------------------------------
      |
+-  ///  ---------------------------------------------+
|                                                     |
+-+       +--------------------------------------------------------------------------
|$|       |                                                                      |
+-+       |                     +---------------------------------------------  ///  -
          |                     |
          |                     |                                                    +
          |         +---------- &&
-----------------------+                          |      |       ||
          |         |                                     |                          |
&& --+  ||
          |         |                                     |                          |
          |   +---- &&
---+                 +----------  ///  -------+               |f1: X    f1: Y||
          |   |           |                 |                        |               +
          |+------+    +------+             |                 +--------------+
          ||[t2:X]|    |[t2:Y]|       +---- ||
---+           |      |       |                       |
          |+------+    +------+       |           |           |  +-- &&
--+  |                       |
          |                        +------+    +------+       |  |        |  |
          |                        |[f2:X]|    |[f2:Y]|       |f1: X    t1:
Y|                       |
          |                        +------+    +------+       +--------------+
          +--------------------------------------------------------------------------

       |
+-  ///  ------------------------------------+
|                                            |
+-+       +--------------------------------------------------------------------------+
|$|       |                                                     |                    |
+-+       |                             +--------------------  ///  --------+        |
          |                             |                                   |        |
          |                             |                            +--------------+|
          |               +----------  ///  --------+                |      |       ||
          |               |                         |                |  +-- && --+  ||
          |               |                  +--------------+        |  |        |  ||
          |      +------- && ---+            |      |       |        |f1: X    f1: Y||
          |      |              |            |  +-- && --+  |        +--------------+|
          |+------------+    +------+        |  |        |  |                        |
          ||[t2:X, t2:Y]|    |[f2:Y]|        |f1: X    t1: Y|                        |
          |+------------+    |[f2:X]|        +--------------+                        |
          |                  +------+                                                |
          +--------------------------------------------------------------------------+

      |
+-  ///  -----------------------+
|                               |
+-+       +--------------------------------------------+
|$|       |      |                                     |
+-+       | +-  ///  ----------------+                 |
          | |                        |                 |
          |+-+       +--------------------------------+|
          ||$|       |               |                ||
          |+-+       |      +------- && ------+       ||
          |          |      |                 |       ||
          |          |      |                 |       ||
          |          |  +-- && --+        +-- && --+  ||
          |          |  |        |        |        |  ||
          |          |f1: X    f1: Y    f1: X    t1: Y||
          |          +--------------------------------+|
          +--------------------------------------------+
      |
+-  ///  -----------------------+
|                               |
+-+       +--------------------------------------------+
|$|       |      |                                     |
+-+       | +-  ///  ----------------+                 |
          | |                        |                 |
          |+-+       +--------------------------------+|
          ||$|       |               |                ||
          |+-+       |      +------- && ------+       ||
          |          |      |                 |       ||
          |          |      |                 |       ||
          |          |  +-- && --+        +-- && --+  ||
          |          |  |        |        |        |  ||
          |          |f1: X    f1: Y    f1: X    t1: Y||
          |          +--------------------------------+|
          +--------------------------------------------+

      |
+-  ///  -----------------------------+
|                                     |
+-+       +---------------------------------------------------------+
|$|       |      |                                                  |
+-+       | +-  ///  -----------------------+                       |
          | |                               |                       |
          |+-+       +---------------------------------------------+|
          ||$|       |                      |                      ||
          |+-+       |          +---------- && ---------+          ||
          |          |          |                       |          ||
          |          |          |                       |          ||
          |          |   +---- && ---+            +---- && ---+    ||
          |          |   |           |            |           |    ||
          |          |+------+    +------+    +------+    +------+||
          |          ||[f1:X]|    |[f1:Y]|    |[f1:X]|    |[t1:Y]|||
          |          |+------+    +------+    +------+    +------+||
          |          +--------------------------------------------+|
          +--------------------------------------------------------+
      |
+-  ///  ----------------------+
|                              |
+-+       +--------------------------------------------+
|$|       |      |                                     |
+-+       | +-  ///  ----------------+                 |
          | |                        |                 |
          |+-+       +--------------------------------+|
          ||$|       |               |                ||
          |+-+       |      +------- && ------+       ||
          |          |      |                 |       ||
          |          |+------------+    +------------+||
          |          ||[f1:X, f1:Y]|    |[f1:X, t1:Y]|||
          |          |+------------+    +------------+||
          |          +--------------------------------+|
          +--------------------------------------------+

      |
+-  ///  --------+
|                |
+-+       +---------------+
|$|       |      |        |
+-+       | +-  ///  --+  |
          | |          |  |
          |+-+       +---+|
          ||$|       |+-+||
          |+-+       ||$|||
          |          |+-+||
          |          +---+|
          +---------------+
```

eof