

# The Structure of Mizar Types

Grzegorz Bancerek

bancerek@mizar.org

Białystok Technical University

TYPES 2006

University of Nottingham

# 1. Types in Mizar

- *PROP* - propositions, *SCHEME* - propositions with second order variables
- *TERM* - terms
- *ADJ* - adjectives
- *TYPE* - types
- *JUST* - justification, *DED* - deductions
- *LOCI* - definition loci, *DEF* - definitions, *REG* - registrations

## 2. Type $PROP$

contradiction :  $\rightarrow PROP$  (1)

not :  $PROP \rightarrow PROP$  (2)

& :  $PROP \rightarrow PROP \rightarrow PROP$  (3)

or :  $PROP \rightarrow PROP \rightarrow PROP$  (4)

implies :  $PROP \rightarrow PROP \rightarrow PROP$  (5)

iff :  $PROP \rightarrow PROP \rightarrow PROP$  (6)

for \_ holds \_ :  $VAR \rightarrow PROP \rightarrow PROP$  (7)

ex \_ st \_ :  $VAR \rightarrow PROP \rightarrow PROP$  (8)

being :  $IDENT \rightarrow TYPE \rightarrow VAR$  (9)

## 2.1. Type $PROP$

$$\text{is} : TERM \rightarrow TYPE \rightarrow PROP \quad (10)$$

$$\text{is} : TERM \rightarrow ADJ \rightarrow PROP \quad (11)$$

$$P_i^k : \underbrace{TERM \rightarrow \dots \rightarrow TERM}_k \rightarrow PROP \quad (12)$$

### 3. Correlates

The relation  $\approx$  is the smallest congruence satisfying the following conditions for any propositions  $\varphi$ ,  $\psi$ ,  $\varphi_1$ ,  $\varphi_2$ , and  $\varphi_3$ :

$$\text{not not } \varphi \approx \varphi \tag{13}$$

$$\varphi \& \text{not contradiction} \approx \varphi \tag{14}$$

$$\text{not contradiction} \& \varphi \approx \varphi \tag{15}$$

$$\varphi_1 \& (\varphi_2 \& \varphi_3) \approx (\varphi_1 \& \varphi_2) \& \varphi_3 \tag{16}$$

$$\varphi \text{ or } \psi \approx \text{not (not } \varphi \& \text{not } \psi) \tag{17}$$

$$\varphi \text{ implies } \psi \approx \text{not } (\varphi \& \text{not } \psi) \tag{18}$$

$$\varphi \text{ iff } \psi \approx (\varphi \text{ implies } \psi) \& (\psi \text{ implies } \varphi) \tag{19}$$

$$\text{ex } x \text{ being } \theta \text{ st } \varphi \approx \text{not for } x \text{ being } \theta \text{ holds not } \varphi \tag{20}$$

## 4. Types $TYPE$ , $ADJ$ , and $TERM$

$$* : ADJ \rightarrow TYPE \rightarrow TYPE \quad (21)$$

$$M_i^k : \underbrace{TERM \rightarrow \dots \rightarrow TERM}_k \rightarrow TYPE \quad (22)$$

## 4. Types $TYPE$ , $ADJ$ , and $TERM$

$$* : ADJ \rightarrow TYPE \rightarrow TYPE \quad (21)$$

$$M_i^k : \underbrace{TERM \rightarrow \dots \rightarrow TERM}_k \rightarrow TYPE \quad (22)$$

$$\text{non} : ADJ \rightarrow ADJ \quad (23)$$

$$A_i^k : \underbrace{TERM \rightarrow \dots \rightarrow TERM}_k \rightarrow ADJ \quad (24)$$

## 4. Types $TYPE$ , $ADJ$ , and $TERM$

$$* : ADJ \rightarrow TYPE \rightarrow TYPE \quad (21)$$

$$M_i^k : \underbrace{TERM \rightarrow \dots \rightarrow TERM}_k \rightarrow TYPE \quad (22)$$

$$\text{non} : ADJ \rightarrow ADJ \quad (23)$$

$$A_i^k : \underbrace{TERM \rightarrow \dots \rightarrow TERM}_k \rightarrow ADJ \quad (24)$$

$$: IDENT \rightarrow TERM \quad (25)$$

$$\text{qua} : TERM \rightarrow TYPE \rightarrow TERM \quad (26)$$

$$F_i^k : \underbrace{TERM \rightarrow \dots \rightarrow TERM}_k \rightarrow TERM \quad (27)$$

## 5. Type *JUST*

*LABEL-LIST* - list of labels separated by commas

by  $\_$  : *LABEL-LIST*  $\rightarrow$  *JUST* (28)

from  $\_(-)$  : *LABEL*  $\rightarrow$  *LABEL-LIST*  $\rightarrow$  *JUST* (29)

proof  $\_$  end : *DED*  $\rightarrow$  *JUST* (30)

## 6. Type *DED*

$\_ : \_ \_ ; \_ : LABEL \rightarrow PROP \rightarrow JUST \rightarrow DED$  (31)

thus  $\_ : \_ \_ ; \_ : LABEL \rightarrow PROP \rightarrow JUST \rightarrow DED$  (32)

assume  $\_ : \_ ; \_ : LABEL \rightarrow PROP \rightarrow DED$  (33)

let  $\_ ; \_ : VAR \rightarrow DED$  (34)

take  $\_ = \_ ; \_ : IDENT \rightarrow TERM \rightarrow DED$  (35)

$\_ \_ : DED \rightarrow DED \rightarrow DED$  (36)

consider  $\_$  being  $\_$  such that  $\_ : \_ \_ ; \_ : IDENT \rightarrow TYPE$   
 $\rightarrow LABEL \rightarrow PROP \rightarrow JUST \rightarrow DED$

reconsider  $\_ = \_$  as  $\_ \_ ; \_ : IDENT \rightarrow TERM \rightarrow TYPE$   
 $\rightarrow JUST \rightarrow DED$

## 7. Environment: type $VARS$ and $PROPS$

$\_ \text{ being } \_ : IDENT \rightarrow TYPE \rightarrow VAR$

$\_ = \_ : IDENT \rightarrow TERM \rightarrow ABBR$  (37)

$\_ = \_ : \_ : IDENT \rightarrow TERM \rightarrow TYPE \rightarrow ABBR$  (38)

$\_ : \_ : LABEL \rightarrow PROP \rightarrow LPROP$  (39)

labels :  $PROPS \rightarrow LABEL-LIST$  (40)

Type  $VARS$  consists of lists of typed variables ( $VAR$ ) and abbreviations ( $ABBR$ ) separated by commas.

Type  $PROPS$  consists of lists of labeled propositions ( $LPROP$ ) separated by commas.

## 8. Loci

Type  $LOCI$  consists of lists of typed variables

$$x_1 : \theta_1, x_2 : \theta_2, \dots, x_k : \theta_k \quad (41)$$

where  $\theta_i : TYPE$  and may depends on variables  $x_1, \dots, x_{i-1}$  only,  
 $\text{vars}(\theta_i) \subseteq \{x_1, \dots, x_{i-1}\}$ .

Terms  $\tau_1, \tau_2, \dots, \tau_k$  are proper realization of loci (41) if

$$\tau_1 : \theta_1, \quad \tau_2 : \theta_2[\tau_1/x_1], \quad \dots$$

$$\tau_k : \theta_k[\tau_1/x_1, \dots, \tau_{k-1}/x_{k-1}]$$

## 9. Constructors and registrations

$\text{attr} : \mathcal{LOCI} \rightarrow \mathcal{TYPE} \rightarrow \mathcal{PROP} \rightarrow \mathcal{ATTR}$  (42)

$\text{func} : \mathcal{LOCI} \rightarrow \mathcal{TYPE} \rightarrow \mathcal{PROP} \rightarrow \mathcal{FUNC}$  (43)

$\text{mode} : \mathcal{LOCI} \rightarrow \mathcal{TYPE} \rightarrow \mathcal{PROP} \rightarrow \mathcal{MODE}$  (44)

$\text{pred} : \mathcal{LOCI} \rightarrow \mathcal{PROP} \rightarrow \mathcal{PRED}$  (45)

$\text{exreg} : \mathcal{LOCI} \rightarrow \mathcal{TYPE} \rightarrow \mathcal{REG}$  (46)

$\text{condreg} : \mathcal{LOCI} \rightarrow \mathcal{TYPE} \rightarrow \mathcal{ADJ} \rightarrow \mathcal{REG}$  (47)

$\text{funcreg} : \mathcal{LOCI} \rightarrow \mathcal{TERM} \rightarrow \mathcal{TYPE} \rightarrow \mathcal{ADJ} \rightarrow \mathcal{REG}$  (48)

## 10. Environment: type $DEFS$

Type  $DEFS$  consists of lists of definitions like

$$P_i^k = \text{pred}(x_1 : \theta_1, \dots, x_k : \theta_k, \varphi) \quad (49)$$

$$M_i^k = \text{mode}(x_1 : \theta_1, \dots, x_k : \theta_k, \theta, \varphi) \quad (50)$$

$$F_i^k = \text{func}(x_1 : \theta_1, \dots, x_k : \theta_k, \theta, \varphi) \quad (51)$$

$$A_i^k = \text{attr}(x_1 : \theta_1, \dots, x_k : \theta_k, \theta, \varphi) \quad (52)$$

$$\text{vars}(\theta) \subseteq \{x_1, \dots, x_k\}, \text{vars}(\varphi) \subseteq \{x_1, \dots, x_k, x\}$$

$x$  is the special variable standing for defined object.

## 11. Environment: type $REGS$

Type  $REGS$  consists of lists of registrations like

$$\text{exreg}(x_1 : \theta_1, \dots, x_k : \theta_k, \theta) \quad (53)$$

$$\text{condreg}(x_1 : \theta_1, \dots, x_k : \theta_k, \theta, \alpha) \quad (54)$$

$$\text{funcreg}(x_1 : \theta_1, \dots, x_k : \theta_k, \tau, \theta, \alpha) \quad (55)$$

$$\text{vars}(\theta), \text{vars}(\alpha), \text{vars}(\tau) \subseteq \{x_1, \dots, x_k\}$$

## 12. Assertions

$\text{::} : DED \rightarrow PROP \rightarrow PROOF\text{-ASSERTION}$  (56)

$\text{::} : TERM \rightarrow TYPE \rightarrow TYPE\text{-ASSERTION}$  (57)

$\text{Wid} : TYPE \rightarrow TYPE \rightarrow TYPE\text{-WIDENING}$  (58)

$\text{Wid} : TYPE \rightarrow TYPE \rightarrow TYPE\text{-WIDENING}$  (59)

$\text{Def} : PROP \rightarrow PROP \rightarrow DEF\text{-EXP}$  (60)

$\text{Roundup} : TYPE \rightarrow ADJ \rightarrow ROUNDUP$  (61)

$\text{Roundup} : TERM \rightarrow ADJ \rightarrow ROUNDUP$  (62)

$\text{Der} : VARS \rightarrow PROPS \rightarrow DEFS \rightarrow REGS$   
 $\quad \quad \quad \rightarrow ASSERTION \rightarrow SEQUENT$  (63)

## 12.1. Assertions

The proof assertion  $\pi : \varphi$  means that  $\pi$  is a deduction (proof) of  $\varphi$ .

The type assertion  $\tau : \theta$  means that  $\theta$  is a syntactic type of  $\tau$  (it implies but differs from “ $\tau$  is  $\theta$ ”).

The type widening  $\theta_1 \preceq \theta_2$  means that  $\theta_1$  widens to (or is subtype of)  $\theta_2$ .

The definitional expansion  $\varphi \Rightarrow \psi$  means that  $\varphi$  is expanded to  $\psi$  according to definitia of used predicates, modes, and attributes.

The adjective possession  $\theta \Leftarrow \alpha$  means that  $\theta$  possesses  $\alpha$ . The adjective possession  $\tau \Leftarrow \alpha$  means that the type of  $\tau$  possesses  $\alpha$ .

## 13. Proof roles

$\mathcal{V}: VARS, \mathcal{A}: PROPS, \mathcal{D}: DEFS, \mathcal{R}: REGS$

$$\frac{}{\mathcal{V}; \mathcal{A}; \mathcal{D}; \mathcal{R} \vdash \varepsilon : \text{not contradiction}} \quad (64)$$

$$\frac{\mathcal{V}; \mathcal{A}, \ell : \varphi; \mathcal{D}; \mathcal{R} \vdash \pi : \psi \quad J \subseteq \text{labels}(\mathcal{A})}{\mathcal{V}; \mathcal{A}; \mathcal{D}; \mathcal{R} \vdash (\ell : \varphi J; \pi) : \psi} \quad (65)$$

$$\frac{\mathcal{V}; \mathcal{A}, \ell : \varphi; \mathcal{D}; \mathcal{R} \vdash \pi : \psi \quad J \subseteq \text{labels}(\mathcal{A})}{\mathcal{V}; \mathcal{A}; \mathcal{D}; \mathcal{R} \vdash (\text{thus } \ell : \varphi J; \pi) : \varphi \ \& \ \psi} \quad (66)$$

$$\frac{\mathcal{V}; \mathcal{A}, \ell : \varphi; \mathcal{D}; \mathcal{R} \vdash \pi : \psi}{\mathcal{V}; \mathcal{A}; \mathcal{D}; \mathcal{R} \vdash (\text{assume } \ell : \varphi; \pi) : \varphi \text{ implies } \psi} \quad (67)$$

### 13.1. Proof roles

$$\frac{\mathcal{V}, x : \theta; \mathcal{A}; \mathcal{D}; \mathcal{R} \vdash \pi : \varphi}{\mathcal{V}; \mathcal{A}; \mathcal{D}; \mathcal{R} \vdash (\text{let } x \text{ be } \theta; \pi) : \text{for } x \text{ being } \theta \text{ holds } \varphi} \quad (68)$$

$$\frac{\mathcal{V}, x = \tau; \mathcal{A}; \mathcal{D}; \mathcal{R} \vdash \pi : \varphi \quad \mathcal{V}; \mathcal{A}; \mathcal{D}; \mathcal{R} \vdash \tau : \theta}{\mathcal{V}; \mathcal{A}; \mathcal{D}; \mathcal{R} \vdash (\text{take } x = \tau; \pi) : \text{ex } x \text{ being } \theta \text{ st } \varphi} \quad (69)$$

$$\frac{\mathcal{V}; \mathcal{A}; \mathcal{D}; \mathcal{R} \vdash \pi : \varphi \quad \mathcal{V}; \mathcal{A}; \mathcal{D}; \mathcal{R} \vdash \psi \Rightarrow \varphi}{\mathcal{V}; \mathcal{A}; \mathcal{D}; \mathcal{R} \vdash \pi : \psi} \quad (70)$$

$$\frac{\mathcal{V}; \mathcal{A}; \mathcal{D}; \mathcal{R} \vdash \pi : \varphi \quad \varphi \approx \psi}{\mathcal{V}; \mathcal{A}; \mathcal{D}; \mathcal{R} \vdash \pi : \psi} \quad (71)$$

## 14. Type assertions

$$\frac{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau_1 : \theta_1 \quad \dots \quad \mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau_k : \theta_k[\tau_1/x_1, \dots, \tau_{k-1}/x_{k-1}]}{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash F_i^k(\tau_1, \dots, \tau_k) : \theta[\tau_1/x_1, \dots, \tau_k/x_k]} \quad (72)$$

if  $F_i^k = \text{func}(x_1 : \theta_1, \dots, x_k : \theta_k, \theta, \varphi) \in \mathcal{D}$

$$\frac{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau : \theta \leq \theta'}{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau : \theta'} \quad (73)$$

$$\frac{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau \Leftarrow \alpha}{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau : \alpha * \theta'} \quad (74)$$

## 15. Type widening

$$\frac{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau_1 : \theta_1 \quad \dots \quad \mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau_k : \theta_k[\tau_1/x_1, \dots, \tau_{k-1}/x_{k-1}]}{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash M_i^k(\tau_1, \dots, \tau_k) \prec \theta[\tau_1/x_1, \dots, \tau_k/x_k]} \quad (75)$$

if  $M_i^k = \text{mode}(x_1 : \theta_1, \dots, x_k : \theta_k, \theta, \varphi) \in \mathcal{D}$

$$\frac{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \theta_1 \preceq \theta_2 \preceq \theta_3}{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \theta_1 \preceq \theta_3} \quad \frac{}{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \theta \preceq \top} \quad (76)$$

$$\frac{}{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \alpha * \theta \preceq \theta} \quad \frac{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \theta \Leftarrow \alpha}{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \theta \preceq \alpha * \theta} \quad (77)$$

## 16. Adjective possession

Let  $\sigma$  be the substitution  $\tau_1/x_1, \dots, \tau_k/x_k$ .

$$\frac{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \theta' \preceq \theta[\sigma]}{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \theta' \Leftarrow \alpha[\sigma]} \quad (78)$$

if  $\text{condreg}(x_1 : \theta_1, \dots, x_k : \theta_k, \theta, \alpha) \in \mathcal{R}$

$$\frac{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau[\sigma] : \theta[\sigma]}{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau \Leftarrow \alpha[\sigma]} \quad (79)$$

if  $\text{funcreg}(x_1 : \theta_1, \dots, x_k : \theta_k, \tau, \theta, \alpha) \in \mathcal{R}$

## 17. Definitional expansion

$$\frac{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau_1 : \theta_1 \quad \dots \quad \mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau_k : \theta_k[\tau_1/x_1, \dots]}{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash P_i^k(\tau_1, \dots, \tau_k) \& \psi \Rightarrow \varphi[\sigma] \& \psi} \quad (80)$$

if  $P_i^k = \text{pred}(x_1 : \theta_1, \dots, x_k : \theta_k, \varphi) \in \mathcal{D}$

$$\frac{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau : \theta' \leq \theta[\sigma] \quad \mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau_1 : \theta_1 \quad \dots}{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash (\tau \text{ is } M_i^k(\tau_1, \dots, \tau_k)) \& \psi \Rightarrow \varphi[\sigma] \& \psi} \quad (81)$$

if  $M_i^k = \text{mode}(x_1 : \theta_1, \dots, x_k : \theta_k, \theta, \varphi) \in D$ .

$$\frac{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau : \theta' \leq \theta[\sigma] \quad \mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau_1 : \theta_1 \quad \dots}{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash (\tau \text{ is } A_i^k(\tau_1, \dots, \tau_k)) \& \psi \Rightarrow \varphi[\sigma] \& \psi} \quad (82)$$

if  $A_i^k = \text{attr}(x_1 : \theta_1, \dots, x_k : \theta_k, \theta, \varphi) \in D$ .

## 17.1. Definitional expansion

$$\frac{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau_1 : \theta_1 \quad \dots \quad \mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau_k : \theta_k[\tau_1/x_1, \dots]}{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \text{not } P_i^k(\tau_1, \dots, \tau_k) \& \psi \Rightarrow \text{not } \varphi[\sigma] \& \psi} \quad (83)$$

$$\frac{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau_1 : \theta_1 \quad \dots \quad \mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \tau_k : \theta_k[\tau_1/x_1, \dots]}{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \text{not } (P_i^k(\tau_1, \dots, \tau_k) \& \psi) \Rightarrow \text{not } (\varphi[\sigma] \& \psi)} \quad (84)$$

if  $P_i^k = \text{pred}(x_1 : \theta_1, \dots, x_k : \theta_k, \varphi) \in \mathcal{D}$

$$\frac{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \varphi_1 \Rightarrow \varphi_2 \Rightarrow \varphi_3}{\mathcal{V}; \mathcal{D}; \mathcal{R} \vdash \varphi_1 \Rightarrow \varphi_3} \quad (85)$$

## 18. Semilattice of Mizar types

Mizar types (objectes of type  $TYPE$ ) form an upper-bounded sup-semilattice with the order  $\preceq$  and the top  $\top$ .