

Mizar MSE (2)

Formułując wstęp do pracy podajemy aksjomaty i twierdzenia, na które będziemy się powoływać. Jak je napiszemy w Mizarze, pokazaliśmy w poprzednim numerze.

Często występuje potrzeba (choćby dla wygody) wprowadzenia stałych obiektów dla dziedzin, którymi się zajmujemy. Na przykład w geometrii, jeśli już mamy pojęcie kąta, taką stałą może być kąt prosty. W Mizarze do wprowadzenia takich stałych służy konstrukcja zaczynająca się od *given*, na przykład

```
GIVEN PUSTY BEING ZBIOR
```

A oto początek wstępu pewnej pracy

```
ENVIRON
```

```
GIVEN 0,1,2 BEING LICZBA
AX1: FOR X BEING LICZBA ST PEXJ HOLDS GECX,0J
AX2: FOR X,Y,Z BEING LICZBA ST LTEX,YJ & LTY,ZJ HOLDS LTEX,ZJ
AX3: PEXJ & PCZJ
****
```

Używane predykaty intuicyjnie znaczący mają:

```
P [X] — X jest dodatnie,
GE [X, Y] — X jest większe lub równe Y,
LT [X, Y] — X < Y.
```

Kto się domyśli, jakiej dziedziny dotyczyła ta praca? W dalszym ciągu tego odcinka będziemy używali pojęć, które wystąpiły w tym wstępie.

Jako się rzekło, Mizar ma sprawdzać poprawność wnioskowań. Opisałiśmy już jak budować poprawne (w tym języku) zdania. Nadszedł czas, by wyjaśnić, jak w Mizarze przeprowadza się rozumowania.

Dzisiaj przedstawimy najprostszyszy sposób uzasadniania stwierdzeń. Ogólnie rozumowania składają się z ciągu kolejnych przejść — kroków. Każdy poszczególny krok powinien być uzasadniony. W wyniku takiego przejścia uzyskujemy pewne stwierdzenie (wyrażone zdaniem) — aby komputer mógł je zaakceptować, musimy przedstawić dlaczego, naszym zdaniem, jest ono prawdziwe. W najprostszym przypadku wystarczy podać przesłanki, z jakich się w danym kroku rozumowania korzysta.

Reguł logiki używanych przy przejściu od przesłanek do wniosku nie trzeba wyjaśniać maszynie (a w Mizarze nawet nie ma tego jak zrobić). Popatrzmy na przykłady

```
A: PEXJ IMPLIES GEC1,0J BY AX1
...
```

```
B: PEXJ BY AX3
...
```

```
C: GEC1,0J BY A,B
```

Słowo *by* („z uwagi na...”) oznacza właśnie powołanie się przy uzasadnianiu zdania *C* na przesłanki oznaczone *A* i *B*. Innym przykładem może być następujące rozumowanie

```
D: PEXJ OR LTEX,0J
E: NOT LTEX,1J & NOT LTEX,0J
F: NOT PEXJ OR GECX,2J
.....
```

```
G: GECX,2J BY D,E,F
```

Przy takich samych przesłankach *D*, *E*, *F* możemy też nasz wniosek uzyskać w dwu krokach.

```
H: PEXJ BY D,E
I: GECX,2J BY H,F
```

W tym przypadku oba rozumowania zostaną zaakceptowane przez system Mizara, dokładniej przez ten jego moduł (część),

który sprawdza takie właśnie uzasadnianie (w naszym żargonie nazywamy go checkerem — „sprawdzaaczem”).

Istnieje pewna klasa zdań, które dla checkera są oczywiste, to znaczy nie wymagają żadnego uzasadnienia.

Na przykład

```
(PEXJ IMPLIES GECX,0J) & PEXJ IMPLIES GECX,0J
```

Stwierdzenie powyższe nie potrzebuje uzasadnienia (nie ma *by*, średnik umieszczony bezpośrednio po tym zdaniu oddziela je od następnego stwierdzenia). Zdanie oznaczone etykietą *A* uzyskano z pewnego ogólnego prawa jako konkretny przypadek. Checker akceptuje takie uzasadnianie. Co więcej, akceptuje również następujący krok rozumowania

```
J: GEC1,0J BY AX1,AX3
```

Mając udowodniony szczególnie przypadek możemy stwierdzić ogólniej

```
L: EX W BEING LICZBA ST GECW,0J BY J
```

i checker z tym się zgodzi.

W przypadku, gdy checker nie akceptuje jakiegoś rozumowania, zaznacza to na wydruku sprawdzanego tekstu, który oddaje po sprawdzeniu i daje następujący komentarz:

„Twoje rozumowanie nie zostało zaakceptowane. Jest możliwe, że logicznie to rozumowanie jest poprawne. Niestety checker nie umiał znaleźć sprzeczności między negacją wniosku i przesłankami. Jeżeli nie jest jasne dlaczego checker odmówił akceptacji, proszę przejrzeć „opis checkera” lub skontaktować się z autorami”.

Pewien opis checkera będzie zawarty w kolejnych odcinkach.

Wicie już, jak w Mizarze dokonuje się najprostszyszy uzasadnień. Przyszła pora, byście wypróbowali samodzielnie swoje siły. Spróbujcie uzupełnić poniższe rozumowania — z jakich przesłanek trzeba korzystać, aby otrzymać podane wnioski. Wstęp do pracy sformułujemy od razu. Dziedziną badaną jest zbiór ułamków, predykat większości dla dwu ułamków *U1*, *U2* będziemy zapisywać *NW [U1, U2]* (co ma intuicyjnie znaczyć, że *U1* jest nie większe od *U2*).

Żeby sformułować zadania, założmy, że wyróżniliśmy pewne stałe i przyjmijmy o nich pewne założenia (we wstępie)

```
ENVIRON
```

```
ZMROTNOSC: FOR X BEING ULAMEK HOLDS NWCX,XJ
PRZECODNIOSC: FOR X,Y,Z BEING ULAMEK ST NWCX,YJ & NWCY,ZJ
HOLDS NWCX,ZJ
ANTYSYMETRIA: FOR X,Y BEING ULAMEK ST NWCX,YJ & NWCY,XJ
HOLDS X=Y
SPOJNOSC: FOR X,Y BEING ULAMEK HOLDS NWCX,YJ OR NWCY,XJ
GIVEN A,B,C,D,E BEING ULAMEK
Z1: A<D
Z2: NWEA,CJ
Z3: NWEB,CJ
Z4: NWC,DJ
Z5: NOT NWEA,EJ
```

```
BEGIN
```

A oto dowodzone tezy

```
T1: NWEA,DJ BY ?
T2: NWEB,DJ BY ?
T3: NWEA,AJ BY ?
T4: NOT NWEA,AJ BY ?
T5: NOT (A=C & C=D) BY ?
T6: NOT C=E BY ?
T7: EX X BEING ULAMEK ST NWCX,XJ BY ?
T7': NWEA,DJ BY ?
```

Proszę uzupełnić miejsca wskazane przez znak zapytania tak, by wnioskowanie było akceptowane. Oczywiście można dopisać dodatkowe kroki pomocnicze.

Przypominamy

Każdy, kto nadesłże pod adresem redakcji rozwiązanie (wraz z zaadresowaną do siebie kopertą — lepiej większą — z naklejonym znacznikiem), otrzyma wydruk z komputera z komentarzem do tego rozwiązania.