Introduction to Mizar

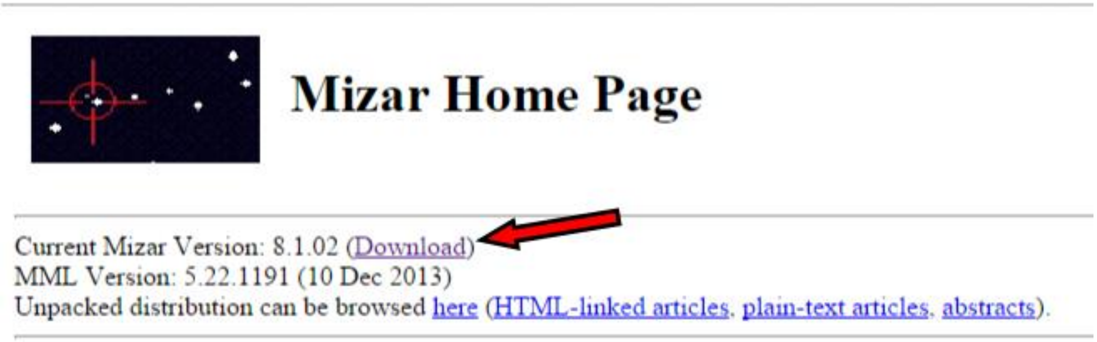
James Bilbe & Nichole Orozco

Mizar is a proof checking software program. The aim of computer proof checking software like Mizar is not to find proofs, but to verify them. Currently, checking a proof by computer is done by taking a known mathematical proof and entering it into the special language recognized by a proof verifier program, and then running the verifier to hopefully obtain no errors. The two main challenges in using a proof checker today are the time needed to learn the syntax and general usage of the system and the time needed to formalize a proof in the system even when the user is already proficient with it. We hope to provide you with a *basic* guide to getting started in Mizar. (Our goal is to familiarize the reader with the system, not to become a proficient user).

I. Getting Started with Mizar

Step 1: Go to mizar.org

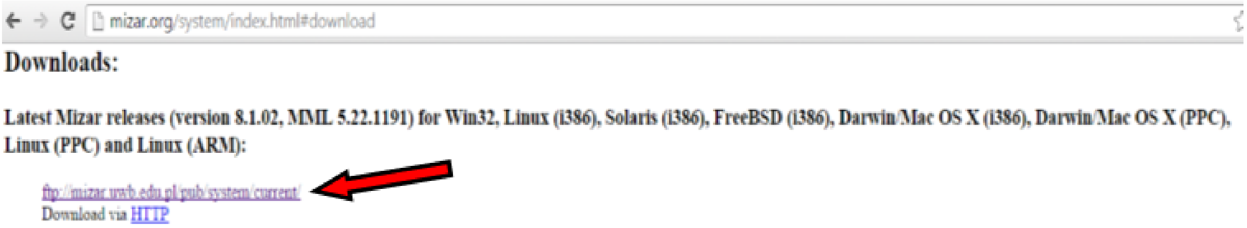
Step 2: Click on DOWNLOAD



Mizar Home Page

Current Mizar Version: 8.1.02 (Download)
MML Version: 5.22.1191 (10 Dec 2013)
Unpacked distribution can be browsed here (HTML-linked articles, plain-text articles, abstracts).

Step 3: Click link



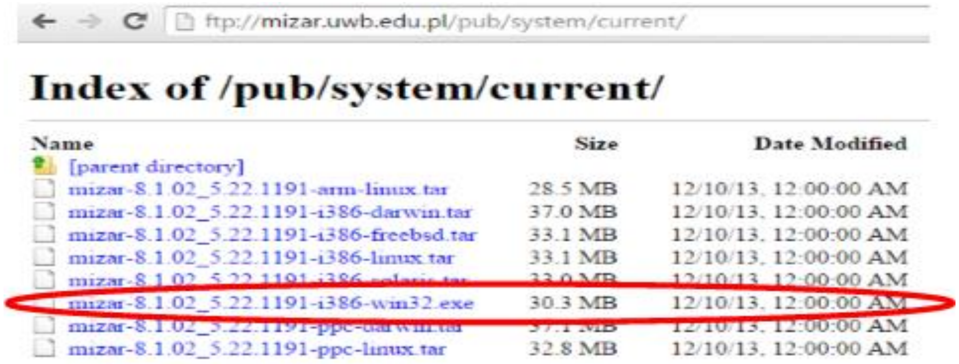
mizar.org/system/index.html#download

Downloads:

Latest Mizar releases (version 8.1.02, MML 5.22.1191) for Win32, Linux (i386), Solaris (i386), FreeBSD (i386), Darwin/Mac OS X (i386), Darwin/Mac OS X (PPC), Linux (PPC) and Linux (ARM):

ftp://mizar.uwb.edu.pl/pub/system/current/
Download via HTTP

Step 4: Click link



ftp://mizar.uwb.edu.pl/pub/system/current/

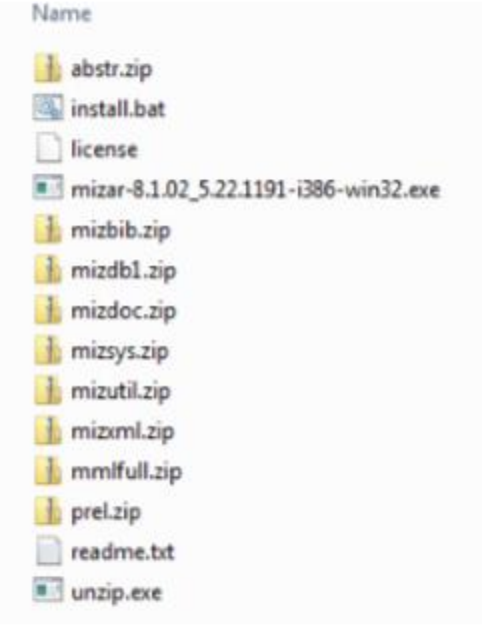
Index of /pub/system/current/

Name	Size	Date Modified
[parent directory]		
mizar-8.1.02_5.22.1191-arm-linux.tar	28.5 MB	12/10/13, 12:00:00 AM
mizar-8.1.02_5.22.1191-i386-darwin.tar	37.0 MB	12/10/13, 12:00:00 AM
mizar-8.1.02_5.22.1191-i386-freebsd.tar	33.1 MB	12/10/13, 12:00:00 AM
mizar-8.1.02_5.22.1191-i386-linux.tar	33.1 MB	12/10/13, 12:00:00 AM
mizar-8.1.02_5.22.1191-i386-solaris.tar	33.0 MB	12/10/13, 12:00:00 AM
mizar-8.1.02_5.22.1191-i386-win32.exe	30.3 MB	12/10/13, 12:00:00 AM
mizar-8.1.02_5.22.1191-ppc-darwin.tar	37.1 MB	12/10/13, 12:00:00 AM
mizar-8.1.02_5.22.1191-ppc-linux.tar	32.8 MB	12/10/13, 12:00:00 AM

Step 5: Create a folder called Temp in Local Disk(C:\ )

Step 6: From Downloads window drag Mizar to Temp folder

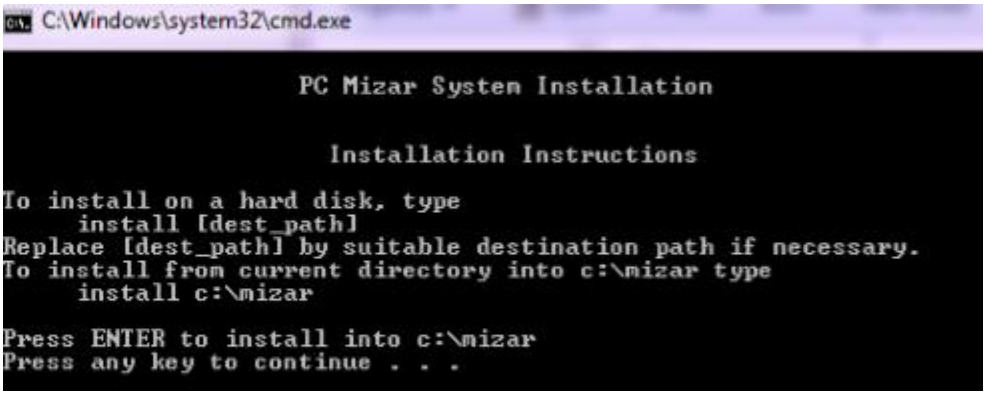
Step 7: Double click **mizar-8.1.02_5.22.1191-i386-win32.exe**



Name

- abstr.zip
- install.bat
- license
- mizar-8.1.02_5.22.1191-i386-win32.exe
- mizbib.zip
- mizdb1.zip
- mizdoc.zip
- mizsys.zip
- mizutil.zip
- mizxml.zip
- mmlfull.zip
- prel.zip
- readme.txt
- unzip.exe

Step 8: Click on **install.bat** in Temp folder then press ENTER

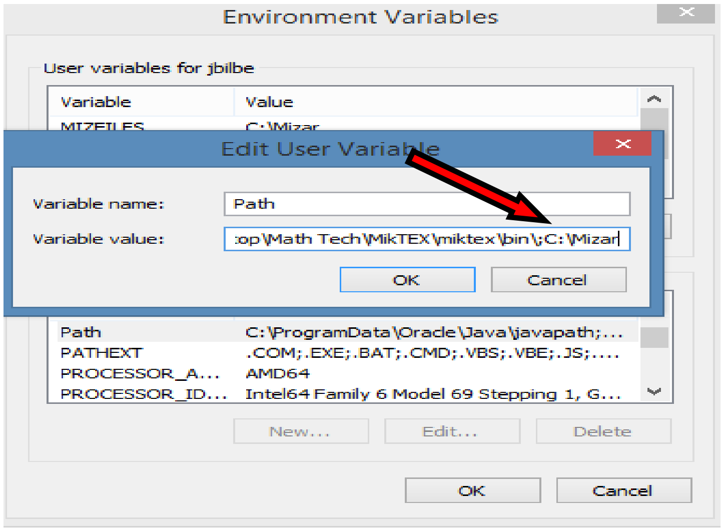


```
                     PC Mizar System Installation

                       Installation Instructions

To install on a hard disk, type
    install [dest_path]
Replace [dest_path] by suitable destination path if necessary.
To install from current directory into c:\mizar type
    install c:\mizar

Press ENTER to install into c:\mizar
Press any key to continue . . .
```

II. Creating and Running an Article

Step 1: Go to Control Panel\System\Environment Variable

Step 2: Add the **C:\Mizar** to the System PATH variable. Also add MIZFILES environment variable with value **C:\Mizar**



Environment Variables

User variables for jbilbe

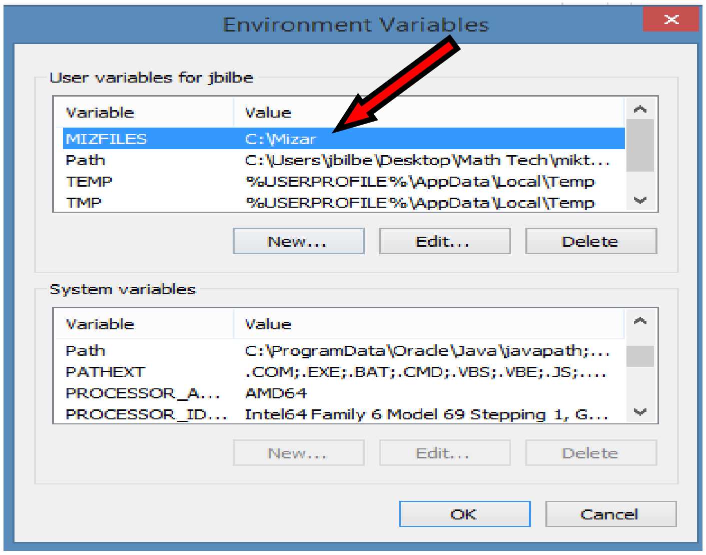
Variable	Value
MIZFILES	C:\Mizar

Edit User Variable

Variable name: Path

Variable value: :op\Math Tech\MikTEX\miktex\bin\;C:\Mizar

Variable	Value
Path	C:\ProgramData\Oracle\Java\javapath;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	AMD64
PROCESSOR_ID...	Intel64 Family 6 Model 69 Stepping 1, G...



Environment Variables

User variables for jbilbe

Variable	Value
MIZFILES	C:\Mizar
Path	C:\Users\jbilbe\Desktop\Math Tech\mikt...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

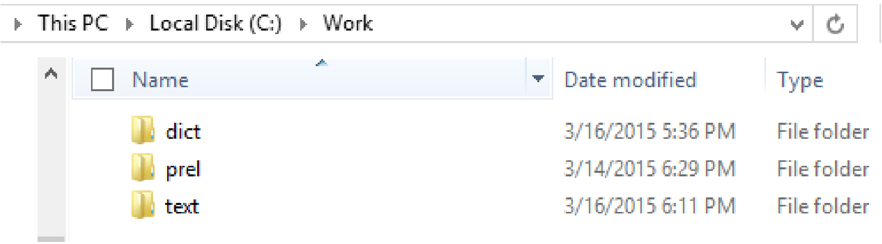
System variables

Variable	Value
Path	C:\ProgramData\Oracle\Java\javapath;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	AMD64
PROCESSOR_ID...	Intel64 Family 6 Model 69 Stepping 1, G...

Step 3: Create a folder in Local Disk(C:\ ) with any name

Step 4: Create 3 folders within this folder with the following names

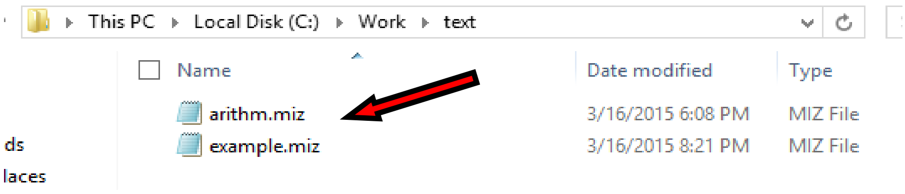
·dict ·prel · text



Name	Date modified	Type
dict	3/16/2015 5:36 PM	File folder
prel	3/14/2015 6:29 PM	File folder
text	3/16/2015 6:11 PM	File folder

Step 5: Create a Mizar article in NotePad and save it in the text folder with a **.miz** extention.

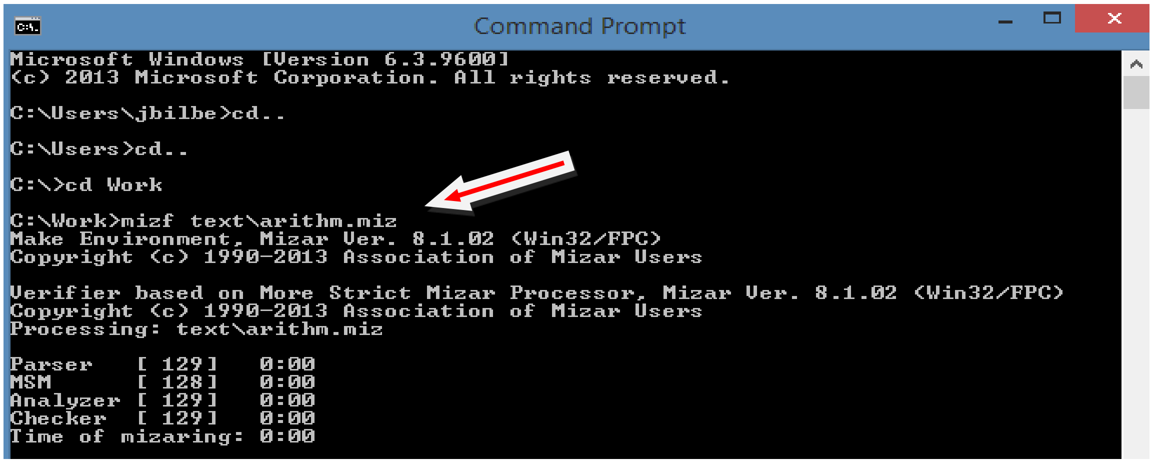
(For example an article containing the basic subset proofs might be saved as **subsets.miz)**



Name	Date modified	Type
arithm.miz	3/16/2015 6:08 PM	MIZ File
example.miz	3/16/2015 8:21 PM	MIZ File

Step 6: Start an MS-DOS prompt and make C:\Work the current directory.

Step 7: Execute command: **C:\Work>mizf text\example.miz**



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\jbilbe>cd..

C:\Users>cd..

C:\>cd Work

C:\Work>mizf text\arithm.miz
Make Environment, Mizar Ver. 8.1.02 (Win32/FPC)
Copyright (c) 1990-2013 Association of Mizar Users

Verifier based on More Strict Mizar Processor, Mizar Ver. 8.1.02 (Win32/FPC)
Copyright (c) 1990-2013 Association of Mizar Users
Processing: text\arithm.miz

Parser   [ 129]   0:00
MSM      [ 128]   0:00
Analyzer [ 129]   0:00
Checker  [ 129]   0:00
Time of mizaring: 0:00
```

III. Verifying a Proof

Basic Properties of Subsets Theorem: ***In normal notation***

1. Inclusion of Intersection: For all sets A and B,

(a) A ∩ B ⊆ A and (b) A ∩ B ⊆ B.

2. Inclusion in Union: For all sets A and B,

(a) A ⊆ A ∪ B and (b) B ⊆ A ∪ B.

3. Transitive Property of Subsets: For all sets A, B, and C,

if A ⊆ B and B ⊆ C, then A ⊆ C.

Proof:
1(a). Suppose A and B are any sets and suppose x is any element of A ∩ B. Then x ∈ A and x ∈ B by definition of intersection. So, x ∈ A.

1(b). Suppose A and B are any sets and suppose x is any element of A ∩ B. Then x ∈ A and x ∈ B by definition of intersection. So, x ∈ B.

2(a). Suppose A and B are any sets and suppose x is any element of A U B. Then x ∈ A and x ∈ B
by definition of union. So, x ∈ A.

2(b). Suppose A and B are any sets and suppose x is any element of A U B. Then x ∈ A and x ∈ B
by definition of union. So, x ∈ B.

3. Suppose A, B, and C are any sets and suppose A ⊆ B and B ⊆ C. To show that A ⊆ C, we must show that every element in A is in C. To this end we note that if x ∈ A, then x ∈ B (because A ⊆ B) and therefore x ∈ C (because B ⊆ C). Hence A ⊆ C. ■

Basic Properties of Subsets Theorem: *In Mizar notation*

```
environ

 vocabularies SUBSET_1, XBOOLE_0, TARSKI, ZFMISC_1;
 notations TARSKI, XBOOLE_0, SUBSET_1, ZFMISC_1;
 constructors TARSKI, SUBSET_1;
 registrations SUBSET_1;
 requirements BOOLE;
 theorems TARSKI, SUBSET_1, ZFMISC_1;

begin

:: This file contains statements which are obvious for Mizar checker if
:: "requirements SUBSET" is included in the environment description
:: of an article. They are published for testing purposes only.
:: Users should use appropriate requirements instead of referencing
:: to these theorems.
:: Some of these items need also "requirements BOOLE" for proper work.

theorem
  for a, b being set st a in b holds a is Element of b by SUBSET_1:def 1;

theorem
  for a, b being set st a is Element of b & b is non empty holds a in b
  by SUBSET_1:def 1;

theorem Th3:
  for a, b being set holds a is Subset of b iff a c= b
proof
  let a, b be set;
  hereby
    assume a is Subset of b;
    then a in bool b by SUBSET_1:def 1;
    hence a c= b by ZFMISC_1:def 1;
  end;
  assume a c= b;
  then a in bool b by ZFMISC_1:def 1;
  hence thesis by SUBSET_1:def 1;
end;

theorem
  for a, b, c being set st a in b & b is Subset of c holds a is Element of c
proof
```

```
  let a, b, c be set;
  assume that
A1: a in b and
A2: b is Subset of c;
  b c= c by A2,Th3;
  then a in c by A1,TARSKI:def 3;
  hence thesis by SUBSET_1:def 1;
end;

theorem
  for a, b, c being set st a in b & b is Subset of c holds c is non empty;
```

IV. Pros and Cons of Mizar?

Pros: One benefit of proof checking systems like Mizar is that, in some instances, it is required to verify a proof as Kam (2007) cited:

> The Flyspeck Project by Dr. Thomas Hales was one of the first required applications of the proof checker idea. Hales' proof of Kepler's conjecture in 1998, the statement that the "grocery store" stacking of spheres is the optimal way to conserve volume, required computer verification in parts of the proof. The referee committee checked much of it by hand, but due to the nature of the proof, it was time-consuming. After five years (not of actual checking, but overall time), the leader, G. Fejes Tóth, would only say that he was "99% certain" that the proof was correct" (Hales, 2005, p. 1). In this instance, having a computer verify the entire proof instead would have produced quicker, more satisfactory results.

The syntax of Mizar is less ambiguous compared to the natural language. There are no variations on the meaning of words. For example, when writing a proof in natural language, one writer might write the word 'therefore' and another author might use 'hence'. While in natural language these words have the same meaning, in the Mizar syntax, these words can only be used in specific situations.

Cons: Time required to simply learn the language, syntax, and formalization of a proof takes a considerable amount of time. Kam (2007) logged his time spent at approximately 89 hours.

Mizar system does not tell you what the errors are when found, it simply provides a general error code that may or may not be specific to your proof.

A user interface does not exist (not user friendly). Even though Mizar has been around since 1975.

References:

Kam,R.(2007). Master's Projects. Case Studies in Proof Checking.