

August 5-9, CICM2024 @ Montreal, Canada

Remote Verification System for Mizar Integrated with Emwiki

Toshiki Kai, Yuta Teruya, Kazuhisa Nakasho
Yamaguchi University

Index

1. Introduction
2. Research Background
3. Mizar Server
4. VSCode for the Web Extension
5. Evaluation
6. Conclusion and Future Work

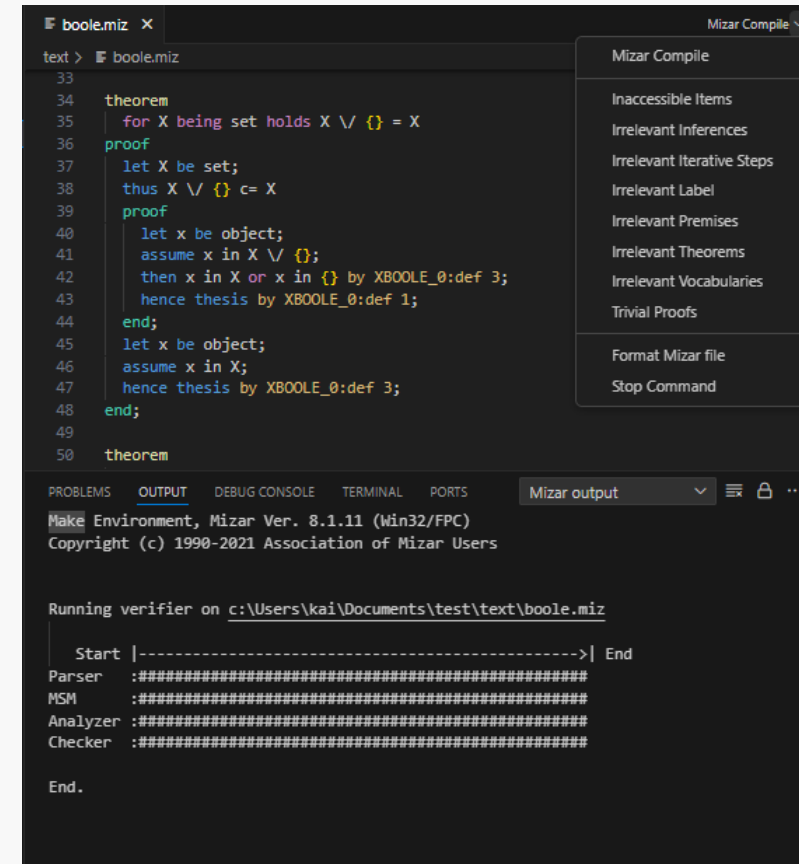
VSCoDe Mizar Extension

● History

- Development has been started around 2020.
- Provide an easy-to-use Mizar IDE for beginners.

● Features

- Syntax highlighting
- Code formatting
- Execution and halting of Mizar commands
- Definition jump and hover functions
for referencing cited theorems and definitions



The screenshot displays the VS Code editor with the Mizar extension. The main editor window shows a Mizar file named 'boole.miz' with the following code:

```
33
34 theorem
35   for X being set holds X \ \ {} = X
36 proof
37   let X be set;
38   thus X \ \ {} c= X
39 proof
40   let x be object;
41   assume x in X \ \ {};
42   then x in X or x in {} by XBOOLE_0:def 3;
43   hence thesis by XBOOLE_0:def 1;
44 end;
45 let x be object;
46 assume x in X;
47 hence thesis by XBOOLE_0:def 3;
48 end;
49
50 theorem
```

A 'Mizar Compile' menu is open, showing options: Inaccessible Items, Irrelevant Inferences, Irrelevant Iterative Steps, Irrelevant Label, Irrelevant Premises, Irrelevant Theorems, Irrelevant Vocabularies, Trivial Proofs, Format Mizar file, and Stop Command.

The bottom panel shows the 'Mizar output' window with the following text:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Mizar output
Make Environment, Mizar Ver. 8.1.11 (Win32/FPC)
Copyright (c) 1990-2021 Association of Mizar Users

Running verifier on c:\Users\kai\Documents\test\text\boole.miz

Start |----->| End
Parser :#####
MSM    :#####
Analyzer :#####
Checker :#####

End.
```

Contribution

- **Mizar Server**
 - Eliminate local setup for Mizar developers.
 - Execute Mizar commands on a remote server.
- **VSCoDe for the Web Extension**
 - Use Mizar Server with VSCoDe for the Web.
 - Implement a remote verification environment in emwiki.

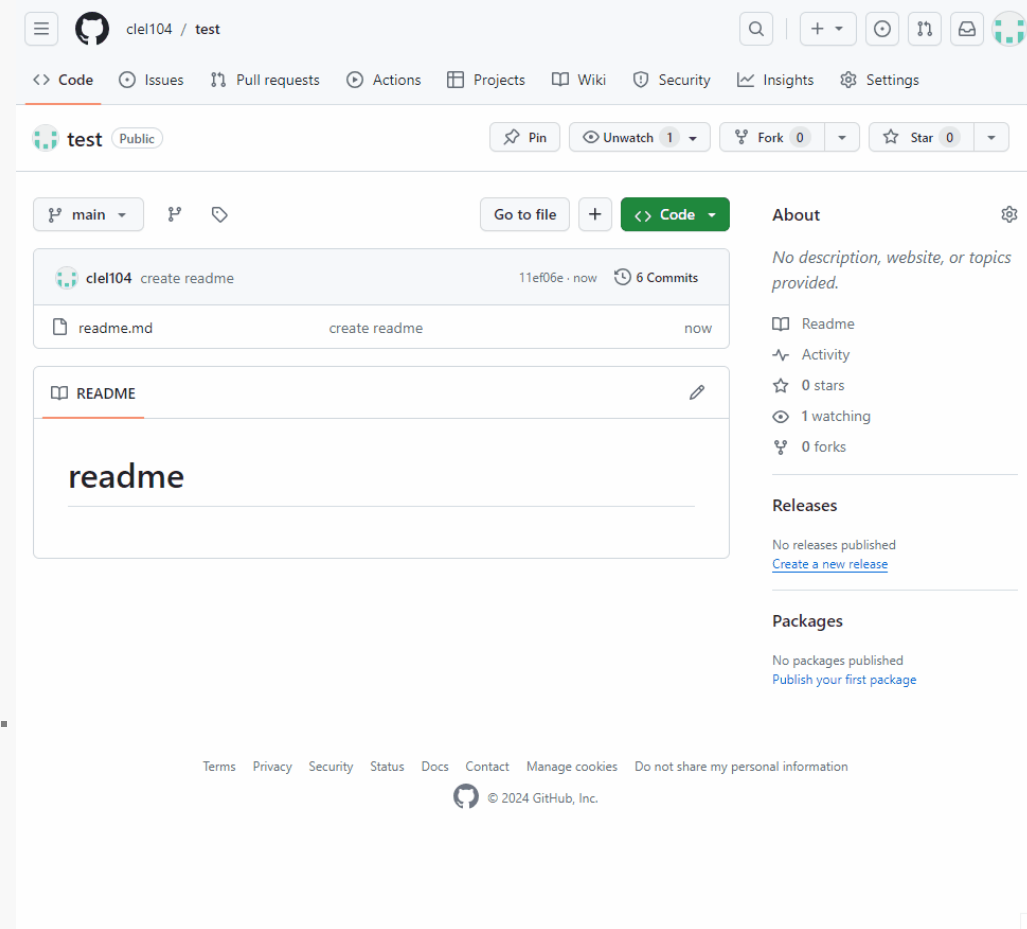
VSCoDe for the Web

● Cloud-based Code Editor

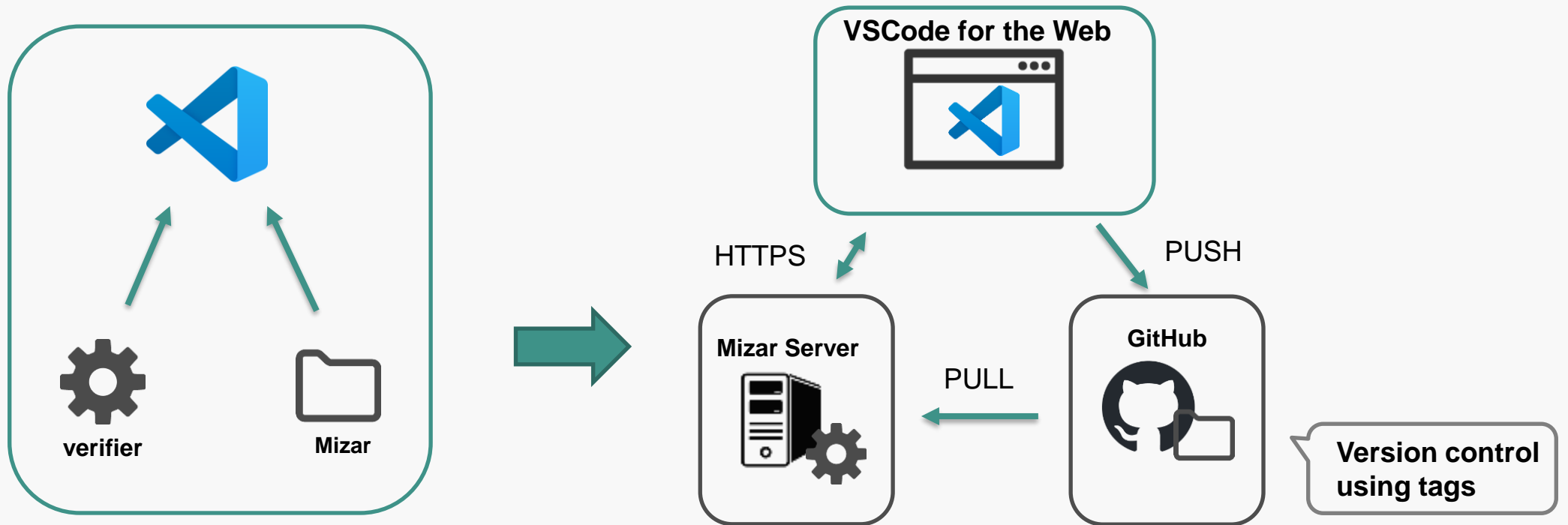
- No installation required.
- Create projects and code online instantly.
- Collaborate seamlessly with GitHub-synced environments.

● GitHub Integration

- Launch directly from GitHub repositories with one click.
- Reflect immediately on GitHub with a commit.



Port the Mizar Extension to VSCode for the Web Extension



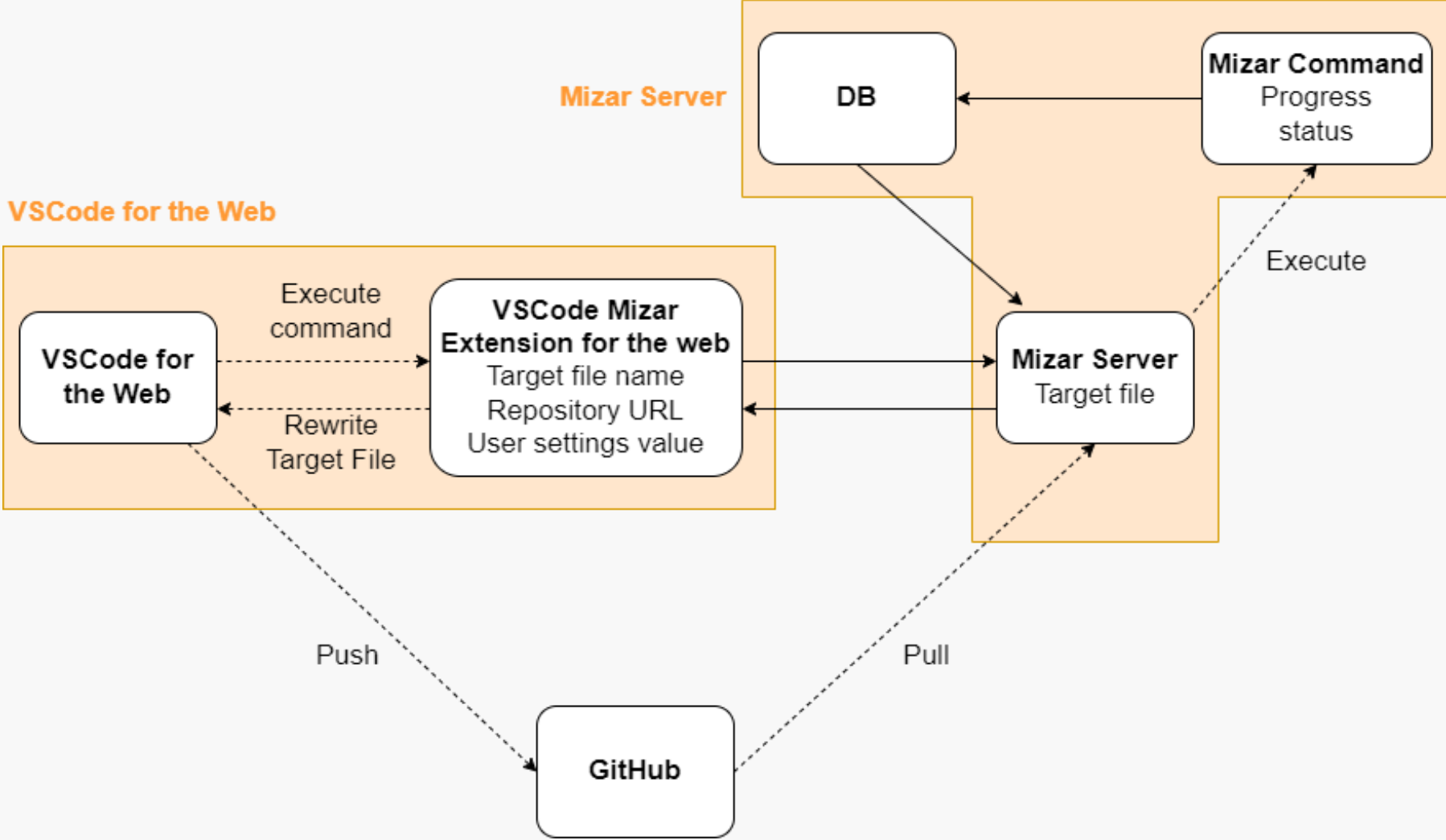
Mizar Extension

Access files placed locally

VSCode for the Web Extension

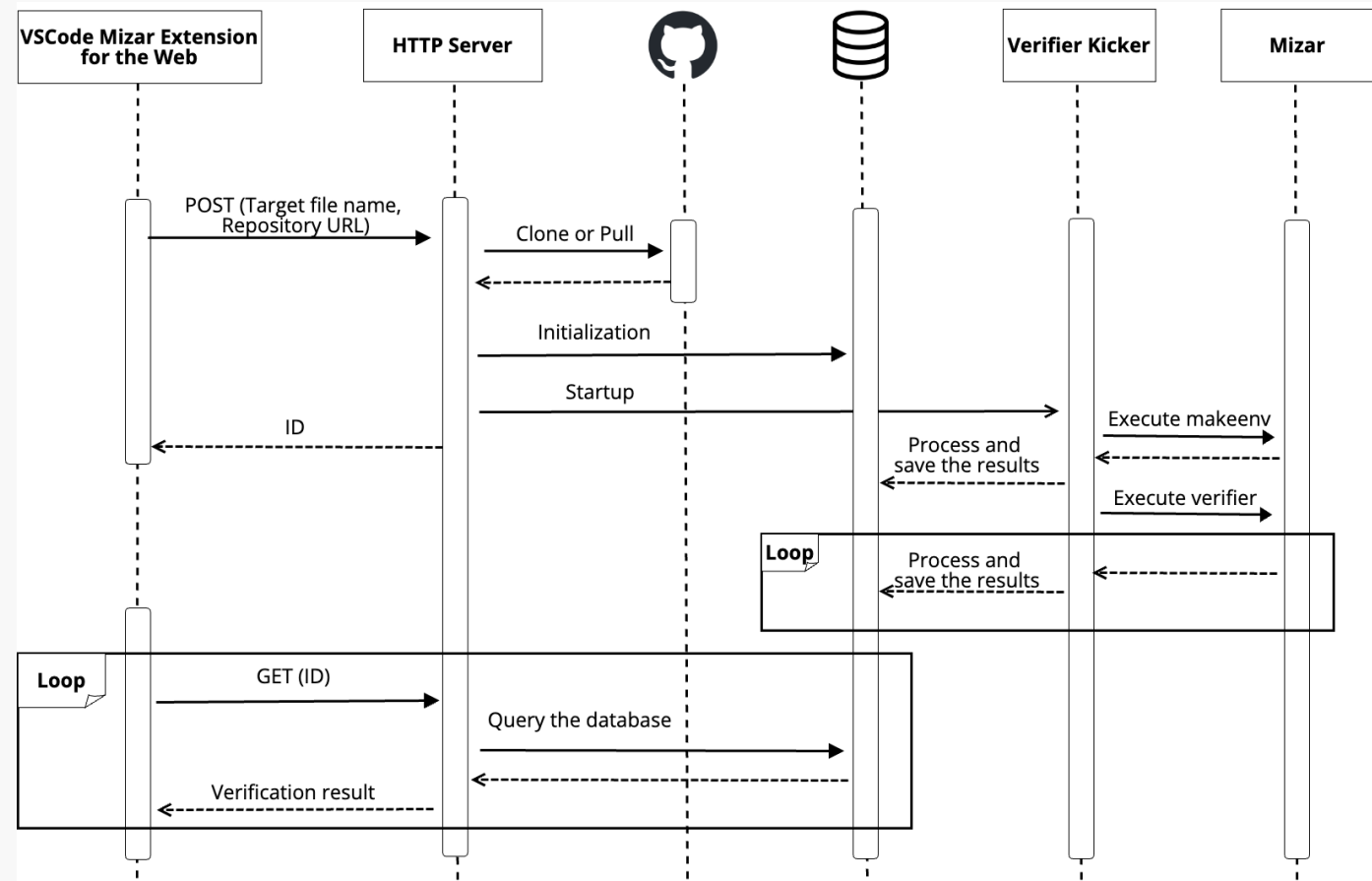
Retrieve information from Mizar Server and GitHub

Architecture Diagram



Mizar Server

- **Cloud-based processing**
 - Access via HTTP API
 - No local setup required
 - Cross-platform compatibility
- **Asynchronous execution**
 - Execute Mizar commands in background
- **Secure and integrated**
 - TLS/SSL communication
 - Integrate GitHub repository



VSCode for the Web Extension

The image shows a screenshot of the VSCode editor interface with the Mizar extension. The editor displays Mizar code with syntax highlighting. A yellow box highlights the word "theorem" on line 84. Another yellow box highlights the code block from line 85 to 105, with an arrow pointing to it from a label "Syntax highlighting". A third yellow box highlights the "definition" block from line 94 to 105, with an arrow pointing to it from a label "Definition jumps and hover information". The right sidebar shows the "Mizar output" terminal with a progress bar and status information. A yellow box highlights the progress bar area, with an arrow pointing to it from a label "Real-time progress monitoring". A fourth yellow box highlights the "Mizar Compile" button in the top toolbar, with an arrow pointing to it from a label "Execute and halt Mizar commands via Mizar Server".

```
text > boole.miz
84 theorem
85   for X being set holds X \+ \ {} = X
86 proof
87   let x be set;
88   thus X \+ \ {} c= X
89 proof
90   let x be object;
91   assume x in X \+ \ {};
92   then
93 A1: x in X \ {} or x in {} \ X by XBOOLE_0:1-3;
94   per cases by A1, X;
95   suppose
96     x in X & not x in {};
97     hence thesis;
98   end;
99   suppose
100     x in {} & not x in X;
101     hence thesis by XBOOLE_0:def 1;
102   end;
103 end;
104 let x be object;
105 A2: not x in {} by XBOOLE_0:def 1;
```

Mizar output

```
Make Environment, Mizar Ver. 8.1.14 (Linux/FPC)
Copyright (c) 1990-2023 Association of Mizar Users
Running verifier on \cIel104\fork_test\text\boole.miz

Start |----->| End
Parser :#####
MSM    :#####
Analyzer :#####
Checker :#####

End.
```

Syntax highlighting

Real-time progress monitoring

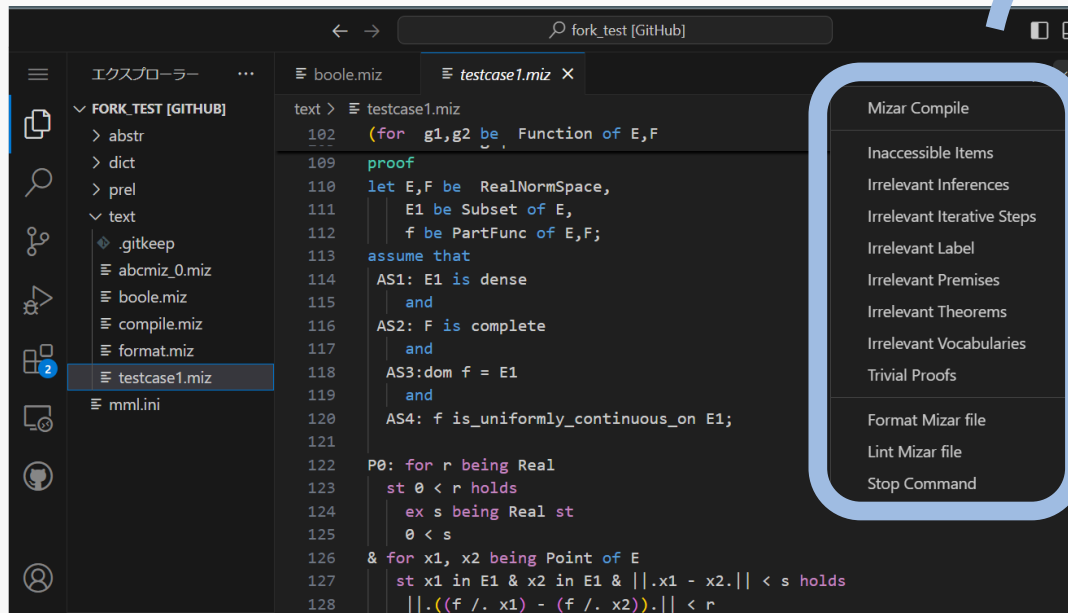
Execute and halt Mizar commands via Mizar Server

Definition jumps and hover information

Commands in the VSCode for the Web Extension

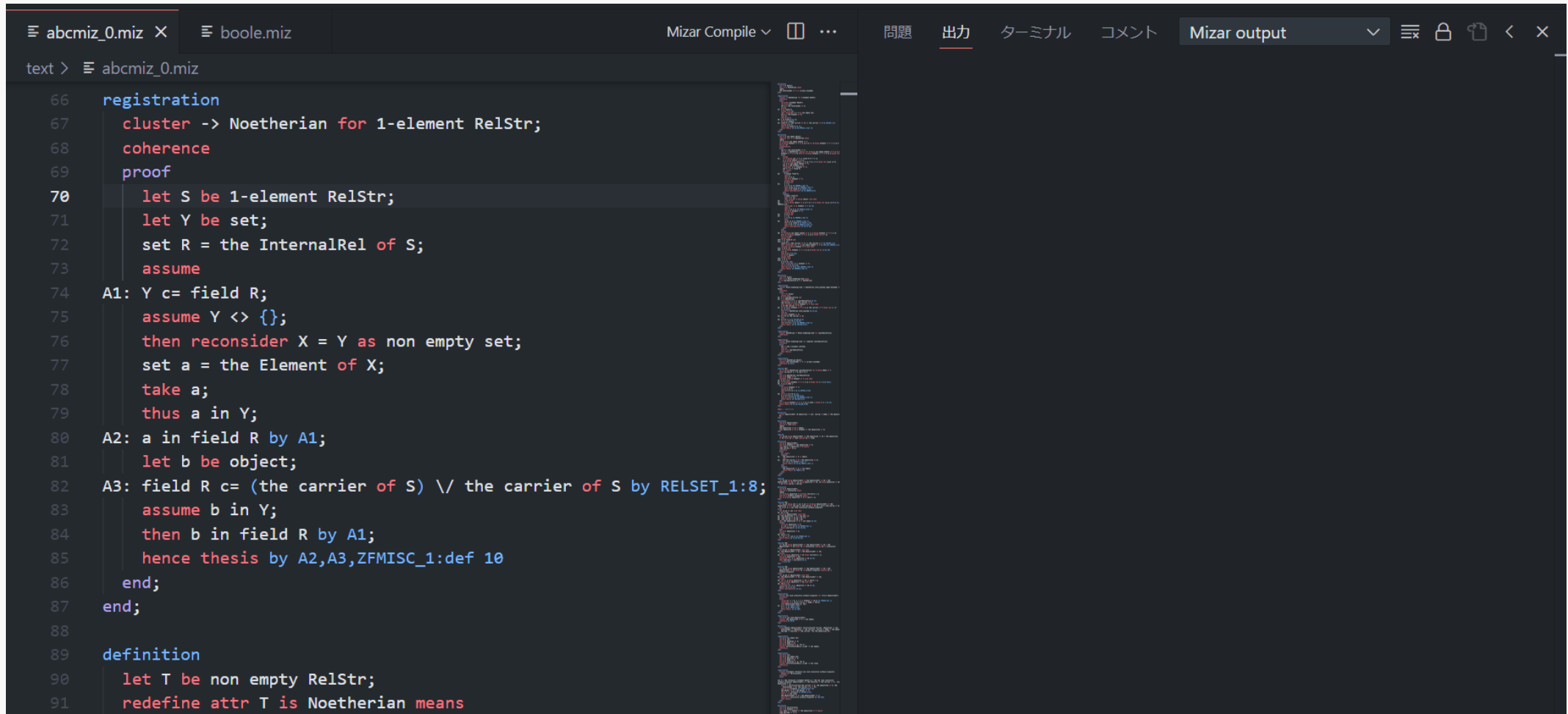
● Available commands

- Verification
- Brush-up hints
- Formatter and linter



Category	Command	Features
Verify	Mizar Compile	Perform verification of the proof.
Brush-up hints	Inaccessible Items	Detect unreachable items.
	Irrelevant Inferences	Detect irrelevant inferences.
	Irrelevant Iterative Steps	Detect redundant iteration steps.
	Irrelevant Label	Detect redundant labels.
	Irrelevant Premises	Detect irrelevant assumptions.
	Irrelevant Theorems	Detect redundant theorems.
	Irrelevant Vocabularies	Detect irrelevant words in the proof.
	Trivial Proofs	Detect trivial proofs.
Other	Format Mizar file	Format proofs.
	Lint Mizar file	Detect deeply nested blocks.
	Stop Command	Terminate commands.

Demo Video



The screenshot shows the VS Code editor interface with two tabs: 'abcmiz_0.miz' and 'boole.miz'. The active tab is 'abcmiz_0.miz', which contains a Mizar proof script. The script is as follows:

```
66 registration
67   cluster -> Noetherian for 1-element RelStr;
68   coherence
69   proof
70     let S be 1-element RelStr;
71     let Y be set;
72     set R = the InternalRel of S;
73     assume
74   A1: Y c= field R;
75     assume Y <> {};
76     then reconsider X = Y as non empty set;
77     set a = the Element of X;
78     take a;
79     thus a in Y;
80   A2: a in field R by A1;
81     let b be object;
82   A3: field R c= (the carrier of S) \ / the carrier of S by RELSET_1:8;
83     assume b in Y;
84     then b in field R by A1;
85     hence thesis by A2,A3,ZFMISC_1:def 10
86   end;
87 end;
88
89 definition
90   let T be non empty RelStr;
91   redefine attr T is Noetherian means
```

The right-hand side of the editor shows the 'Mizar output' panel, which displays the compiled output of the script, including various system messages and the final result of the proof.

Integration of Remote Verification on emwiki

● Enhancement

- Integrate MML browsing, development, and documentation in one platform
- Realize streamline Mizar workflow through seamless remote verification
- Enhance accessibility and collaboration for the Mizar community

VSCode for the Web

emwiki

Related Documents

tarski

HOME DOCUMENT ARTICLE SYMBOL THEOREM GRAPH ACCOUNT

HOME DOCUMENT DEVELOP ARTICLE SYMBOL THEOREM GRAPH SETTINGS TERUYAYUTA

The Subspace Topology

§16 The Subspace Topology

Definition. Let (X, \mathcal{T}) be a topological space, let $Y \subseteq X$. Then we define the 'Subspace Topology' of Y consists of the collection $\mathcal{T}_Y = \{Y \cap U \mid U \in \mathcal{T}\}$ and it forms a topology on Y . We refer to the topological space (Y, \mathcal{T}_Y) as a 'Subspace' of X .

Mizar defines Subspace of E in file TOPIC_01.miz.

Lemma 16.1. If B is a basis for the topology of X , then $\mathcal{B}_Y = \{B \cap Y \mid B \in \mathcal{B}\}$ is a basis for the subspace topology of Y . *Missing from Mizar*

Lemma 16.2. Let (Y, \mathcal{T}_Y) be a subspace of (X, \mathcal{T}) . If U is open in Y and Y is open in X , then U is open in X . (TOPIC_16 proves more than this)

theorem TOPIC_16.1

for X being TopSpace
for X1 being open Subspace of X
for X2 being open Subspace of X1 holds X2 is open Subspace of X

proof end;

Theorem 16.3. If A is a subspace of X and B is a subspace of Y , then the product topology on $A \times B$ is the same as the topology $A \times B$ inherits as a subspace of $X \times Y$. (TOPIC_16.3)

theorem TOPIC_16.3

for X, Y being TopSpace
for Z being Subset of [X]X
for W being Subset of X
for W being Subset of Y & Z = [W]W holds
TopSpace the carrier of [(Y \ W)X \ W] #1 = TopSpace the carrier of [(X \ W) \ W] #2, the topology of [(X \ W) \ W] #4

proof end.

testcase1.miz

```

102 (for g1,g2 be Function of E,F
109 proof
110 let E,F be RealNormSpace,
111     E1 be Subset of E,
112     f be PartFunc of E,F;
113 assume that
114 AS1: E1 is dense
115 and
116 AS2: F is complete
117 and
118 AS3: dom f = E1
119 and
120 AS4: f is_uniformly_continuous_on E1;
121
122 P0: for r being Real
123     st 0 < r holds
124         ex s being Real st
125             0 < s
126 & for x1, x2 being Point of E
127     st x1 in E1 & x2 in E1 & ||.x1 - x2.|| < s holds
128         ||.(f /. x1) - (f /. x2).|| < r

```

Mizar Compile

- Inaccessible Items
- Irrelevant Inferences
- Irrelevant Iterative Steps
- Irrelevant Label
- Irrelevant Premises
- Irrelevant Theorems
- Irrelevant Vocabularies
- Trivial Proofs
- Format Mizar file
- Lint Mizar file
- Stop Command

● Create explanatory articles

● Browse the library

● Develop MML

Comparative evaluation: Remote vs Local Environment

Aspect	Remote System (VSCode for the Web)	Local System (Mizar Extension)
Setup	Simple GitHub repository setup	Manual environment configuration
MML Updates	Automatic	Manual
Core Features	Syntax highlighting, Command execution, Definition jumps	Syntax highlighting, Command execution, Definition jumps
Unique Features	New Linter/Formatter functionality	-
Performance	12.27 seconds	10.02 seconds
Accessibility	Access any device with web browser	Requires local installation
Network Connection	Required	Not required

Conclusion

- **Successfully implemented a remote verification environment**
 - Eliminates need for local setup
 - Integrates with emwiki platform
- **VSCoDe for the Web Extension is now available on emwiki**
 - Combines library browsing, documentation, and development in one-platform
 - Enhances accessibility for Mizar users

Future Work

- **Performance optimization**

- Explore parallelization of verification processes
- Investigate high-performance server deployment

- **Improve user experience**

- Simplify account management and GitHub integration
- Enhance accessibility across different devices and platforms

- **Potential feature expansions**

- Adding collaborative editing capabilities
- Integration with other mathematical tools and libraries