VisualAI: A Framework for Interactive Abstract Interpretation

Jung Hyun Kim

SoftSec Lab., KAIST

IS661 Spring, 2024

Abstract Interpretation (AI)

[1] Cousot, Patrick, and Radhia Cousot. "Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints." Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages. 1977.

Abstract Interpretation (AI)

- Abstract Interpretation[1] (AI): a *sound approximation* of a program.

[1] Cousot, Patrick, and Radhia Cousot. "Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints." Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages. 1977.

Abstract Interpretation (AI)

- Abstract Interpretation[1] (AI): a *sound approximation* of a program.

```
int a;
if (cond) {
  a = 0;
} else {
  a = 1;
}
print(a);
```


[1] Cousot, Patrick, and Radhia Cousot. "Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints." Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages. 1977.

Abstract Interpretation (AI)

- Abstract Interpretation[1] (AI): a *sound approximation* of a program.

```
int a;
```

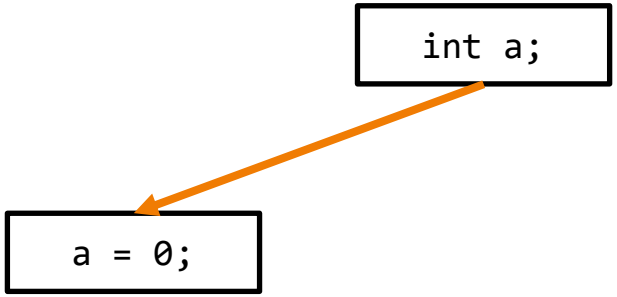
```
int a;
if (cond) {
  a = 0;
} else {
  a = 1;
}
print(a);
```


[1] Cousot, Patrick, and Radhia Cousot. "Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints." Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages. 1977.

Abstract Interpretation (AI)

- Abstract Interpretation[1] (AI): a *sound approximation* of a program.

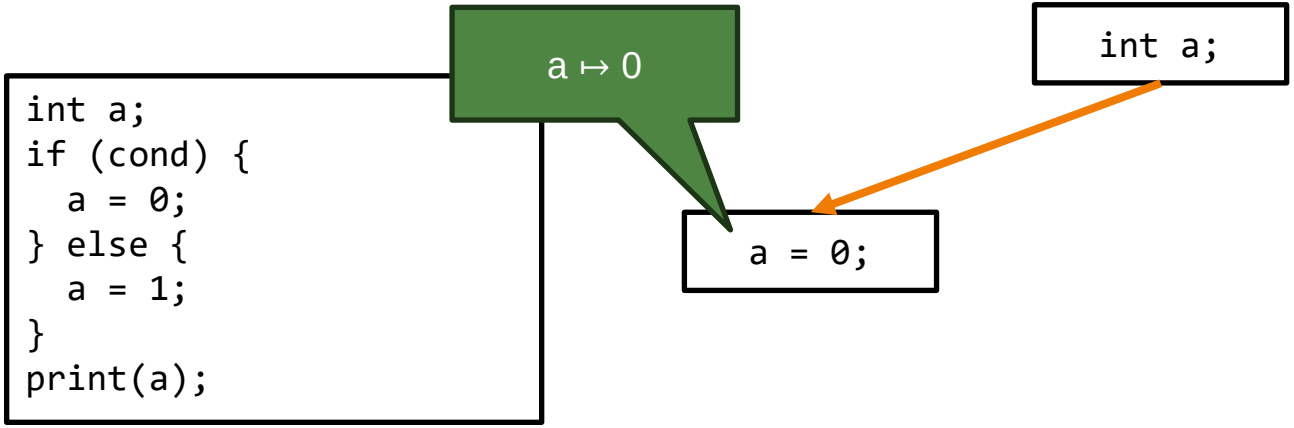
```
int a;
if (cond) {
  a = 0;
} else {
  a = 1;
}
print(a);
```



[1] Cousot, Patrick, and Radhia Cousot. "Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints." Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages. 1977.

Abstract Interpretation (AI)

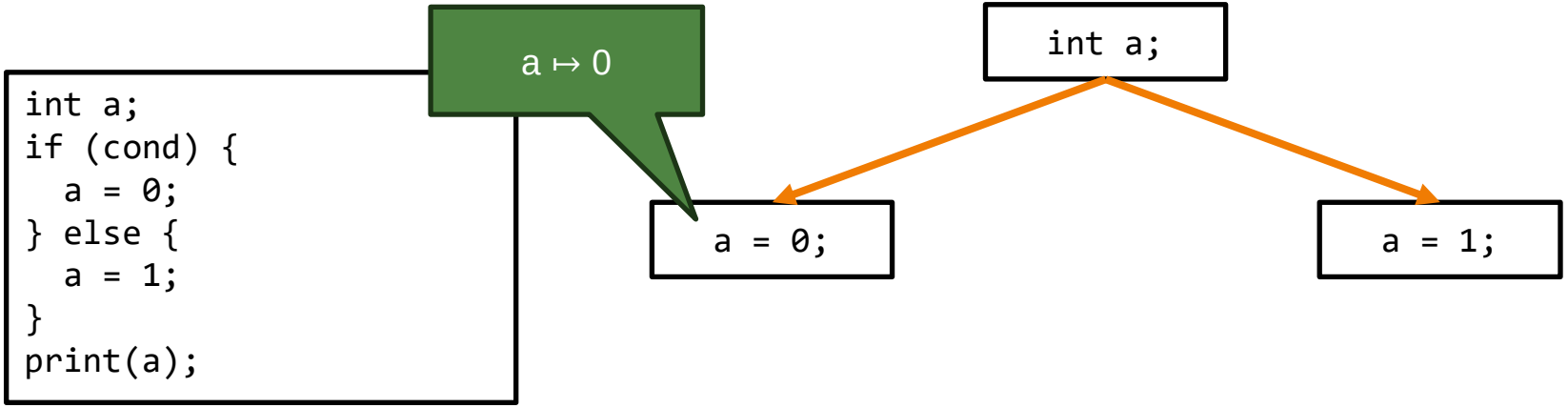
- Abstract Interpretation[1] (AI): a *sound approximation* of a program.



[1] Cousot, Patrick, and Radhia Cousot. "Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints." Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages. 1977.

Abstract Interpretation (AI)

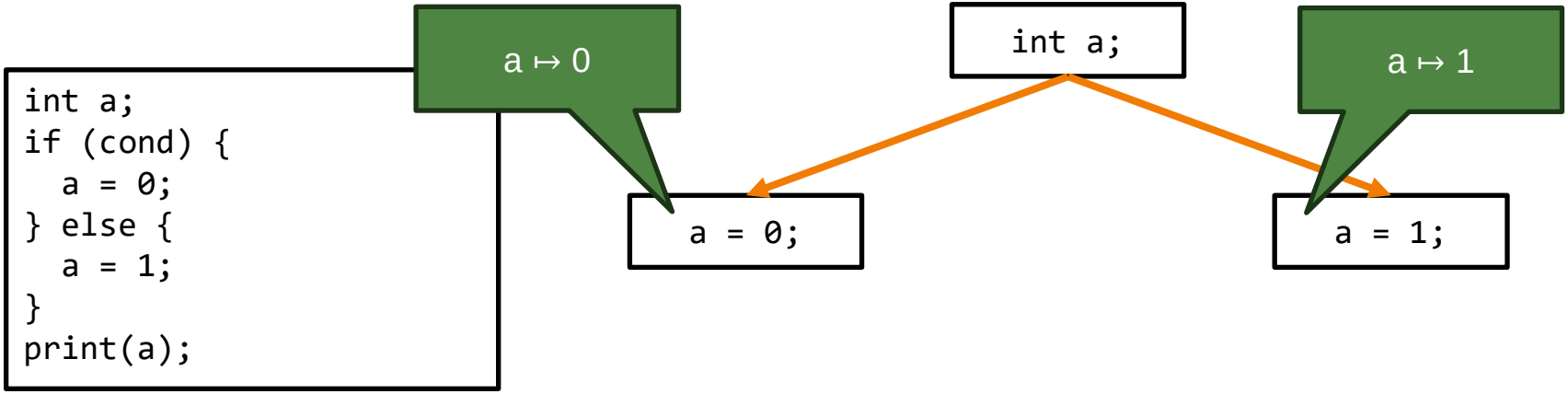
- Abstract Interpretation[1] (AI): a *sound approximation* of a program.



[1] Cousot, Patrick, and Radhia Cousot. "Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints." Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages. 1977.

Abstract Interpretation (AI)

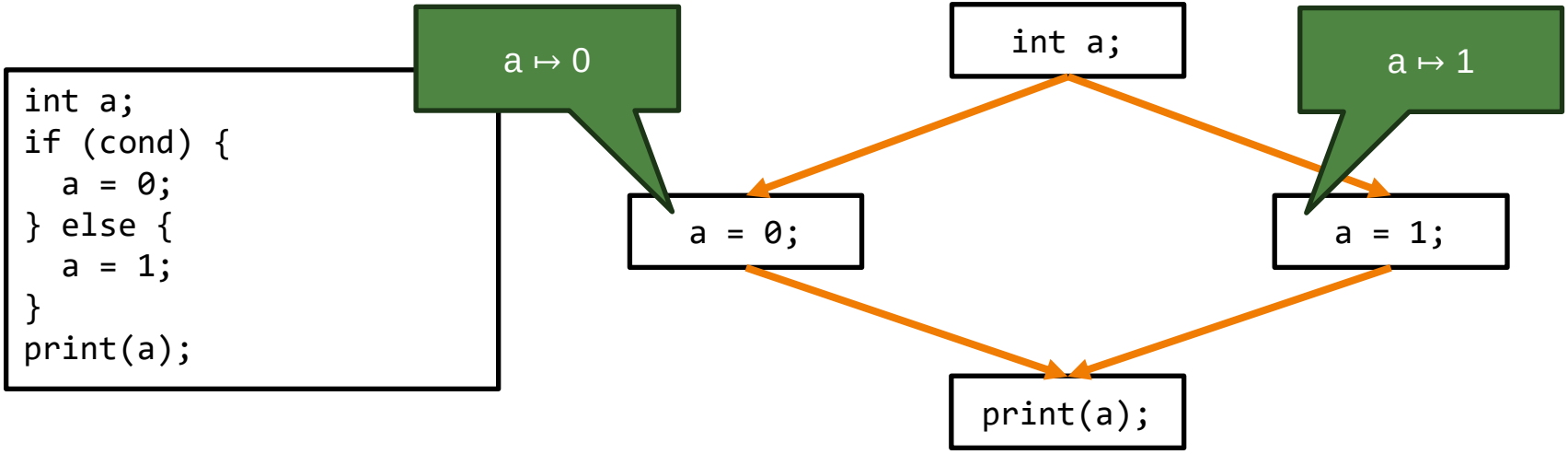
- Abstract Interpretation[1] (AI): a *sound approximation* of a program.



[1] Cousot, Patrick, and Radhia Cousot. "Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints." Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages. 1977.

Abstract Interpretation (AI)

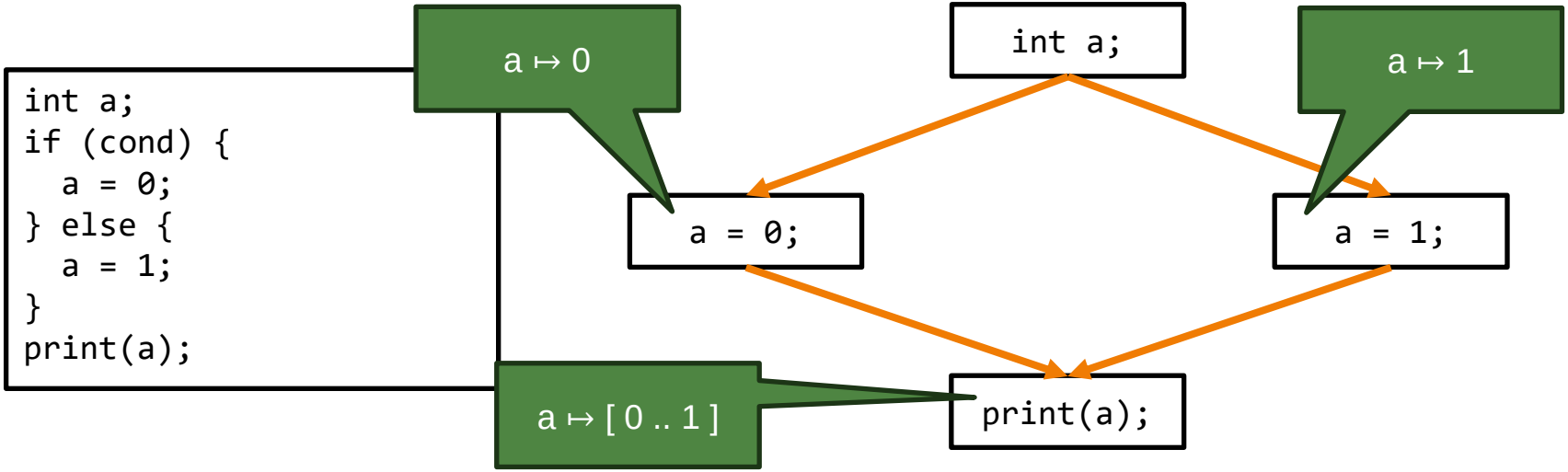
- Abstract Interpretation[1] (AI): a *sound approximation* of a program.



[1] Cousot, Patrick, and Radhia Cousot. "Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints." Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages. 1977.

Abstract Interpretation (AI)

- Abstract Interpretation[1] (AI): a *sound approximation* of a program.



[1] Cousot, Patrick, and Radhia Cousot. "Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints." Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages. 1977.

Difficulties In Developing AI

Difficulties In Developing AI

1. It is hard to inspect *the inside of an AI process*.

Difficulties In Developing AI

1. It is hard to inspect *the inside of an AI process*.

Lack of *Visualization*

Difficulties In Developing AI

1. It is hard to inspect *the inside of an AI process*.

2. It is hard to follow *how states are propagated*.

Difficulties In Developing AI

1. It is hard to inspect *the inside of an AI process*.

2. It is hard to follow *how states are propagated*.

Lack of *Visualization*

Lack of *Interaction*

Interactive AI With Visualization

Interactive AI With Visualization

- *VisualAI*: An interactive visualizer for AI.

Interactive AI With Visualization

- *VisualAI*: An interactive visualizer for AI.
  - Visualization: the abstract state, graph, statement, etc.

Interactive AI With Visualization

- *VisualAI*: An interactive visualizer for AI.
  - Visualization: the abstract state, graph, statement, etc.
  - Interaction: real-time debugging, on-demand information, etc.

Interactive AI With Visualization

- *VisualAI*: An interactive visualizer for AI.
  - Visualization: the abstract state, graph, statement, etc.
  - Interaction: real-time debugging, on-demand information, etc.

- It helps people understand the internals of an AI.

Interactive AI With Visualization

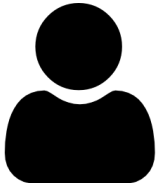
- *VisualAI*: An interactive visualizer for AI.
  - Visualization: the abstract state, graph, statement, etc.
  - Interaction: real-time debugging, on-demand information, etc.

- It helps people understand the internals of an AI.
  - Using *VisualAI*, we found several bugs in our binary analyzer.

A Demo Video

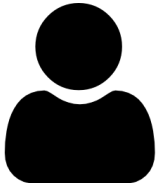
Technical Details

Technical Details



User

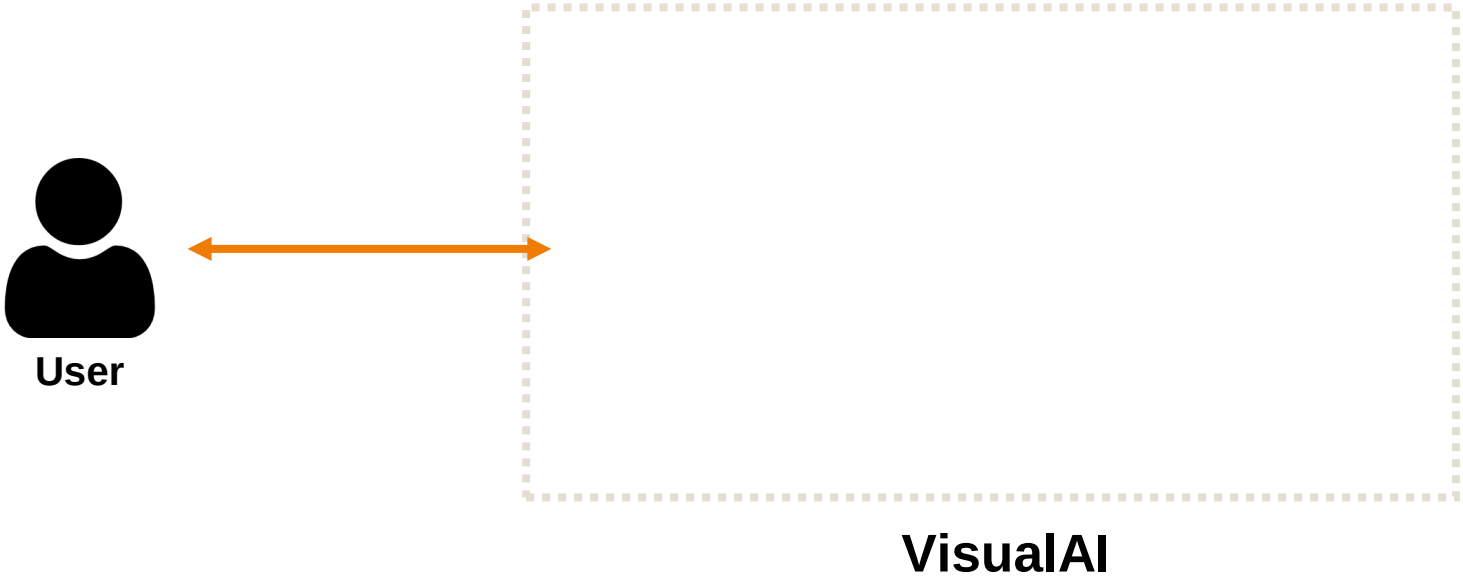
Technical Details



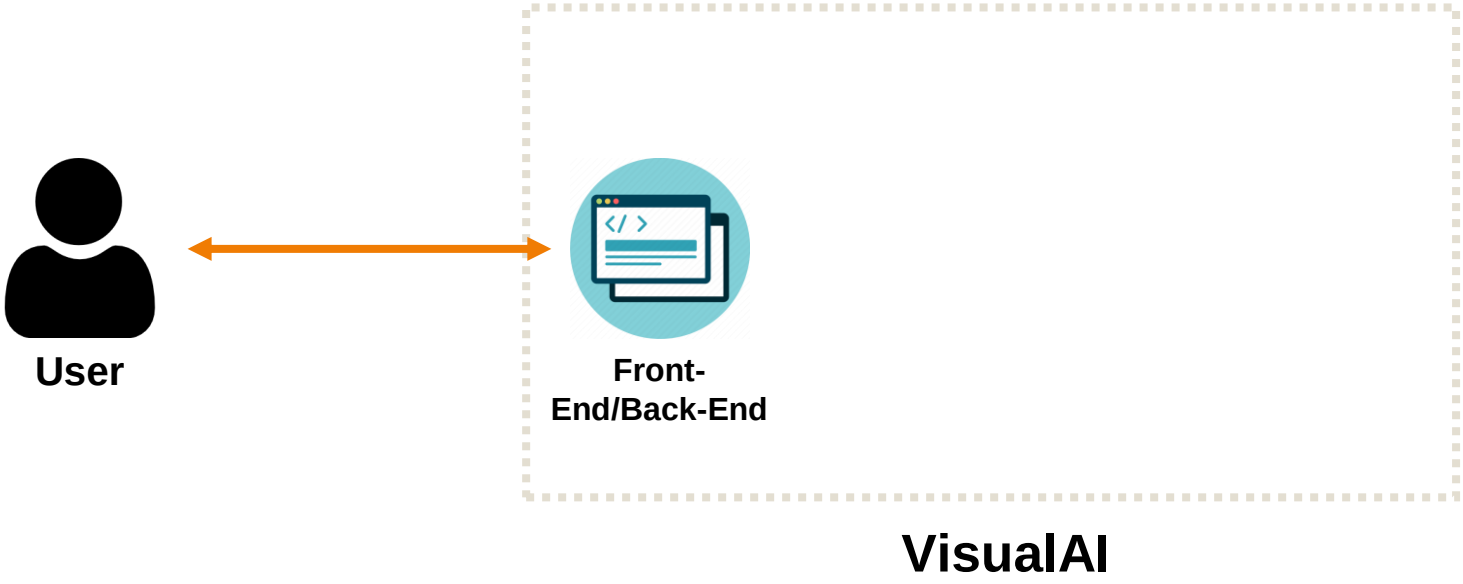
User

VisualAI

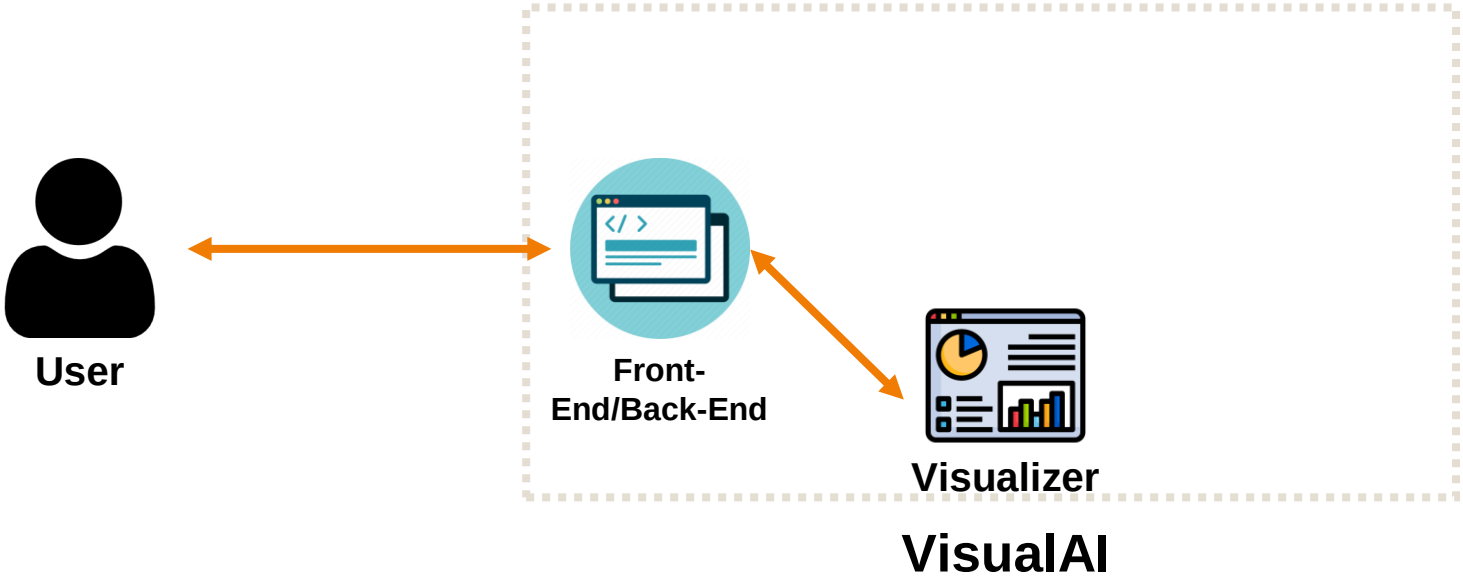
Technical Details



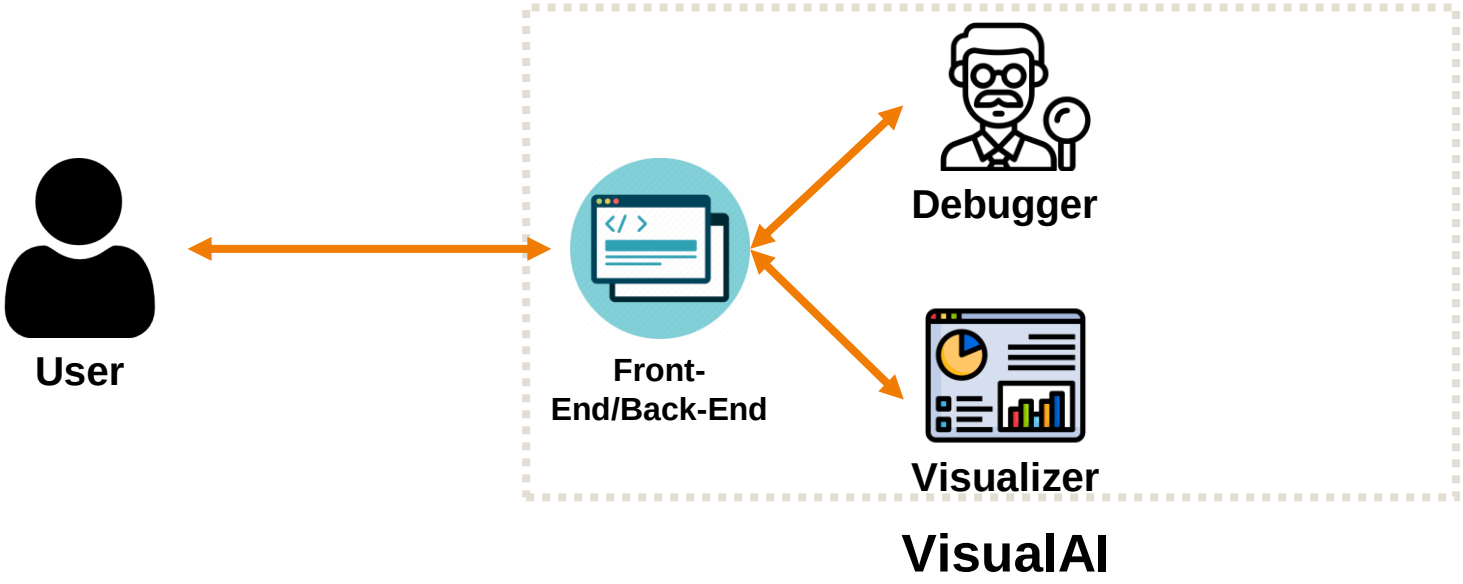
Technical Details



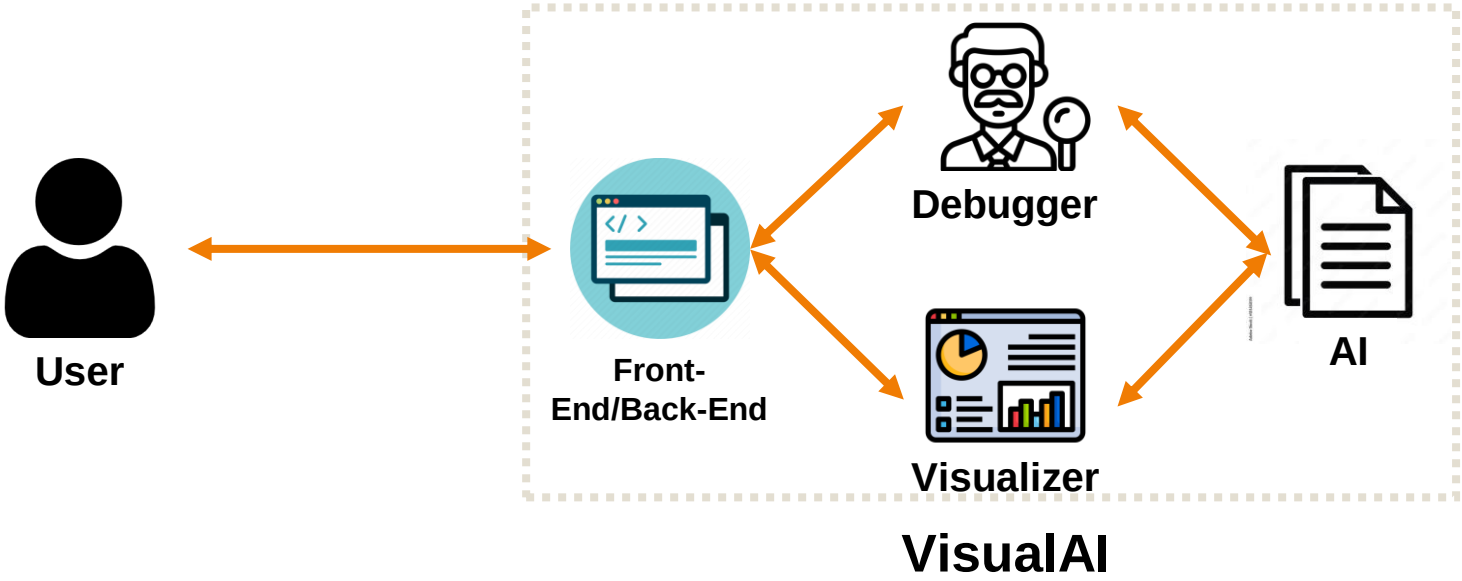
Technical Details



Technical Details



Technical Details



Technical Details – Front-/Back-End

[2] Garrett, Jesse James. "Ajax: A new approach to web applications." (2005).
[3] https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

Technical Details – Front-/Back-End

- A simple HTML Server using Socket.

[2] Garrett, Jesse James. "Ajax: A new approach to web applications." (2005).
[3] https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

Technical Details – Front-/Back-End

- A simple HTML Server using Socket.
- Provides both *ajax*[2] handling and its main page rendering.

[2] Garrett, Jesse James. "Ajax: A new approach to web applications." (2005).
[3] https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

Technical Details – Front-/Back-End

- A simple HTML Server using Socket.
- Provides both *ajax*[2] handling and its main page rendering.
  - *ajax*: used for communication between the server and a webpage.

[2] Garrett, Jesse James. "Ajax: A new approach to web applications." (2005).
[3] https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

Technical Details – Front-/Back-End

- A simple HTML Server using Socket.
- Provides both *ajax*[2] handling and its main page rendering.
  - *ajax*: used for communication between the server and a webpage.
  - To follow *CORS*[3] policy, we must use the *same* origin.

[2] Garrett, Jesse James. "Ajax: A new approach to web applications." (2005).
[3] https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

Technical Details – Front-/Back-End

- A simple HTML Server using Socket.
- Provides both *ajax*[2] handling and its main page rendering.
  - *ajax*: used for communication between the server and a webpage.
  - To follow *CORS*[3] policy, we must use the *same* origin.

- An *ajax* example:

[2] Garrett, Jesse James. "Ajax: A new approach to web applications." (2005).
[3] https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

Technical Details – Front-/Back-End

- A simple HTML Server using Socket.
- Provides both *ajax*[2] handling and its main page rendering.
  - *ajax*: used for communication between the server and a webpage.
  - To follow *CORS*[3] policy, we must use the *same* origin.

- An *ajax* example:
  - Requesting a function information from the server.

[2] Garrett, Jesse James. "Ajax: A new approach to web applications." (2005).
[3] https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

Technical Details – Front-/Back-End

- A simple HTML Server using Socket.
- Provides both *ajax*[2] handling and its main page rendering.
  - *ajax*: used for communication between the server and a webpage.
  - To follow *CORS*[3] policy, we must use the *same* origin.

- An *ajax* example:
  - Requesting a function information from the server.
  - Running a single step in an AI.

[2] Garrett, Jesse James. "Ajax: A new approach to web applications." (2005).
[3] https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

Technical Details – Front-/Back-End

- A simple HTML Server using Socket.
- Provides both *ajax*[2] handling and its main page rendering.
  - *ajax*: used for communication between the server and a webpage.
  - To follow *CORS*[3] policy, we must use the *same* origin.

- An *ajax* example:
  - Requesting a function information from the server.
  - Running a single step in an AI.
  - Fetching the current state of a specific vertex.

[2] Garrett, Jesse James. "Ajax: A new approach to web applications." (2005).
[3] https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

Technical Details – Visualizer

[4] Zhu, Nick Qi. Data visualization with D3. js cookbook. Packt Publishing Ltd, 2013.

Technical Details – Visualizer

- *d3.js*[4]: a visualization library for web applications.

[4] Zhu, Nick Qi. Data visualization with D3. js cookbook. Packt Publishing Ltd, 2013.

Technical Details – Visualizer

- *d3.js*[4]: a visualization library for web applications.
- Implemented in a single HTML file with 500 LOC.

[4] Zhu, Nick Qi. Data visualization with D3. js cookbook. Packt Publishing Ltd, 2013.

Technical Details – Visualizer

- *d3.js*[4]: a visualization library for web applications.

- Implemented in a single HTML file with 500 LOC.
  - Most features were implemented in *javascript*.

[4] Zhu, Nick Qi. Data visualization with D3. js cookbook. Packt Publishing Ltd, 2013.

Technical Details – Visualizer

- *d3.js*[4]: a visualization library for web applications.
- Implemented in a single HTML file with 500 LOC.
  - Most features were implemented in *javascript*.
- Uses event callbacks for interactive visualization:

[4] Zhu, Nick Qi. Data visualization with D3. js cookbook. Packt Publishing Ltd, 2013.

Technical Details – Visualizer

- *d3.js*[4]: a visualization library for web applications.
- Implemented in a single HTML file with 500 LOC.
  - Most features were implemented in *javascript*.
- Uses event callbacks for interactive visualization:
  - *Click* for showing the clicked vertex's information.

[4] Zhu, Nick Qi. Data visualization with D3. js cookbook. Packt Publishing Ltd, 2013.

Technical Details – Visualizer

- *d3.js*[4]: a visualization library for web applications.
- Implemented in a single HTML file with 500 LOC.
  - Most features were implemented in *javascript*.
- Uses event callbacks for interactive visualization:
  - *Click* for showing the clicked vertex's information.
  - *Mouseover* for on-demand information rendering.

[4] Zhu, Nick Qi. Data visualization with D3. js cookbook. Packt Publishing Ltd, 2013.

Technical Details – Debugger

[5] F Sharp (F#), https://learn.microsoft.com/en-us/dotnet/fsharp/what-is-fsharp.

Technical Details – Debugger

- Implemented in *F#*[5].

[5] F Sharp (F#), https://learn.microsoft.com/en-us/dotnet/fsharp/what-is-fsharp.

Technical Details – Debugger

- Implemented in *F#*[5].
- Modified the previous AI engine to interact with the outside.

[5] F Sharp (F#), https://learn.microsoft.com/en-us/dotnet/fsharp/what-is-fsharp.

Technical Details – Debugger

- Implemented in *F#*[5].
- Modified the previous AI engine to interact with the outside.
  - Do not calculate its fixpoint in a row; just stop at every step.

[5] F Sharp (F#), https://learn.microsoft.com/en-us/dotnet/fsharp/what-is-fsharp.

Technical Details – Debugger

- Implemented in *F#*[5].
- Modified the previous AI engine to interact with the outside.
  - Do not calculate its fixpoint in a row; just stop at every step.
- Send the information to the Front-End.

[5] F Sharp (F#), https://learn.microsoft.com/en-us/dotnet/fsharp/what-is-fsharp.

Technical Details – Debugger

- Implemented in *F#*[5].
- Modified the previous AI engine to interact with the outside.
  - Do not calculate its fixpoint in a row; just stop at every step.
- Send the information to the Front-End.
  - State information.

[5] F Sharp (F#), https://learn.microsoft.com/en-us/dotnet/fsharp/what-is-fsharp.

Technical Details – Debugger

- Implemented in *F#*[5].
- Modified the previous AI engine to interact with the outside.
  - Do not calculate its fixpoint in a row; just stop at every step.
- Send the information to the Front-End.
  - State information.
  - Statement information.

[5] F Sharp (F#), https://learn.microsoft.com/en-us/dotnet/fsharp/what-is-fsharp.

Impacts

Impacts

- Found two critical bugs in an AI implementation.

Impacts

- Found two critical bugs in an AI implementation.
- Wrong Top Calculation Bug

Impacts

- Found two critical bugs in an AI implementation.

- Wrong Top Calculation Bug
  - A variable was calculated as Top even though it should not be so.

Impacts

- Found two critical bugs in an AI implementation.

- Wrong Top Calculation Bug
  - A variable was calculated as Top even though it should not be so.
- Insufficient Sensitivity Bug

Impacts

- Found two critical bugs in an AI implementation.

- Wrong Top Calculation Bug
  - A variable was calculated as Top even though it should not be so.
- Insufficient Sensitivity Bug
  - Different call contexts were overly merged.

Impacts

- Found two critical bugs in an AI implementation.

- Wrong Top Calculation Bug
  - A variable was calculated as Top even though it should not be so.
- Insufficient Sensitivity Bug
  - Different call contexts were overly merged.

- With VisualAI, it was easy to track causes of the bugs.

Future Work

Future Work

- Visualize def-use (use-def) chains.

Future Work

- Visualize def-use (use-def) chains.
- Support breakpoints with conditions.

Future Work

- Visualize def-use (use-def) chains.
- Support breakpoints with conditions.
- Support an instruction-level debugging.

Question?