Abstract Interpretation of Pandas

Jan Hrubý
Charles University, Faculty of Mathematics and Physics

Motivation

Pandas is a widely used data manipulation library for Python. The dynamic nature of Pandas and Python can be a source of runtime errors. Consider the following script:

```
import pandas as pd
df = pd.read_csv("data.csv")
df_copy = df
df_copy.drop("column1", inplace=True)
grouped = df.groupby("column1")
# Error - column1 does not exist already
final_score = df["score_a"] + df["score_b_note"]
# Error - summing Series of ints with strings
print(df["colunm2"])
# Error - misspelled column name colunm2
```

Our goal is to design an analysis tool capable of recognizing similar errors.

Abstract Interpretation

Program analysis method formally defined using Lattices and Galois Connection

1. Model program values using an abstract domain (e.g., model numbers as {+, -, any})
2. Define behaviour of operations over the abstract domain
3. Evaluate (interpret) the user-written program over the abstract domain

Framework

The Pandas library provides the user with two main data structures: one-dimensional Series and two-dimensional DataFrame. To interpret the program with Pandas over the abstract domain, we define the abstract lattice. The Figures 1 and 2 show how the abstract lattice is defined.

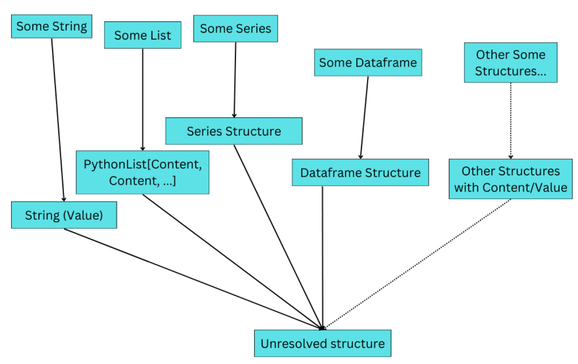

Fig. 1: Lower part of the Lattice

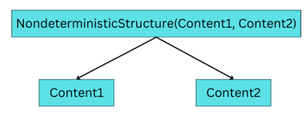

Fig. 2: Upper part of the Lattice

There is the UnresolvedStructure at the bottom of the hierarchy, representing a value that we were not able to derive (due to error in execution). And there is a NondeterministicStructure as a supremum for each pair representing uncertainty between two options. The uncertainty usually occurs in the uncertain if-statement.

Example Case Study

We show one of the case studies from the thesis. The listing 1 shows an example of a Pandas code. The listing 2 contains the information about the input CSV files which needs to be provided via a config file. The listing 3 shows the output of the Pandalyzer after analyzing the code in listing 1.

```
1  import pandas as pd
2
3  attendees_df = pd.read_csv("attendees.csv")
4  matches_df = pd.read_csv("matches.csv") \
5      .rename(columns={"name": "match_name"})
6  scores_df = pd.read_csv("scores.csv")
7
8  attendees_df["name_surname"] = \
9      attendees_df["name"] + "_" + attendees_df["surname"]
10 attendees_df = attendees_df.drop(columns=["name", "surname"])
11
12 scores_with_match_name_df = scores_df \
13     .merge(matches_df, left_on="match_id", right_on="id") \
14     .drop(columns="id")
15
16 scores_with_age_df = pd.merge(
17     scores_with_match_name_df, attendees_df, on="name_surname"
18 )
19
20 top_two_per_age_df = scores_with_age_df \
21     .sort_values("age") \
22     .groupby(["age", "match_name"]) \
23     .head(2) \
24     .drop(columns=["match_id"])
25
26 top_two_per_age_df.to_csv("top_two_per_age.csv")
```

Listing 1: Pandas code in Python

```
[attendees.csv]
name = "string"
surname = "string"
age = "int"

[matches.csv]
id = "int"
name = "string"

[scores.csv]
name_surname = "string"
match_id = "int"
score = "int"
```

Listing 2: Configuration file

```
Summary of analysis: OK
Global data structures (7):
/* snip */
Warnings (0):

Errors (0):

Output files (1):
File top_two_per_age.csv:
    name_surname : StringType
    score : IntType
    match_name : StringType
    age : IntType
```

Listing 3: Output of Pandalyzer

Implementation of Pandalyzer

- Implemented in Kotlin
- Command-line interface
- Configuration file with input structure and regex filenames support
- Supports subset of Python syntax (will be extended in the future works)
- Supports subset of Pandas functions (will be extended in the future works)
- Gives information about the errors in the code, output file structure, global variables and useful warnings
- Supports both human-readable and JSON formats
- Uses Python ast library for the parsing and AST creation
- Supports unknown structures as a result of user input
- Interprets multiple branches of the program in case where unable to choose the right branch
- Continues with analysis even when an error occurs

Capabilities

- Detecting access to non-existent column, operation on incompatible types, incorrect function arguments or operations leading to incorrect state
- CSV output reporting
- CSV input hint handling
- Handling uncertainty from the user input
- Supports pandas functions including merge, groupby, drop, rename, read_csv, to_csv, concat, Dataframe and Series creation, subscript in get and set contexts, vectorized sum and product, aggregation functions (mean, sum, first, last, count, head)
- High extensibility (w.r.t. other pandas function)

Conclusion

The first result of our work is the proposition of the framework for abstract interpretation of data-manipulation programs. The idea could be used with other data-manipulation libraries in other languages like Tibble in R. The second result is the implementation of the Pandalyzer - Pandas analyzer based on the proposed framework. The Pandalyzer is still in the early stage of the development process, and it serves as a proof of concept of the proposed framework. However, it shows that the idea is implementable and can be very useful in practice. The future works include adding support for other Pandas functions, Python language constructs, or Pandas Indexes. The Pandalyzer could also be extended to support other well-known related Python libraries such as Numpy or Matplotlib and integrate the Pandalyzer to the IDEs using the Language Server Protocol.

Acknowledgements

There are people I would not be able to finish the thesis without and I think they deserve to be mentioned. The first person is my supervisor, Professor Tomáš Petříček. I thank him for his guidance, great insight and courage he provided me with. I would also like to thank my fiancée Lana. She was there for me whenever I needed her, and I am grateful for that. I would also want to thank the tearoom Jedna Báseň and tearoom Dharmasala for a comfortable spot for writing my thesis and a steady supply of great tea.

Further info

Pandalyzer source code repository

https://github.com/Hrubian/Pandalyzer

Thesis source code repository

https://github.com/Hrubian/bachelor-thesis