Component-based Synthesis of Table Consolidation and Transformation Tasks from Examples

Abstract

MORPHEUS is a novel example-driven synthesis tool for automating a large class of data preparation tasks that arise in data science. Given a set of input tables and an output table, MORPHEUS synthesizes a table transformation program that performs the desired task. We have evaluated MORPHEUS on dozens of data preparation tasks obtained from on-line forums, and we show that our approach can automatically solve a large class of problems encountered by R users.

1. Introduction

This report describes how to reproduce the main results of the PLDI'17 paper "Component-based Synthesis of Table Consolidation and Transformation Tasks from Examples". We consider the main results of the paper to be the "Summary of experimental results" presented in Figure 16 and the "Cumulative running time of MORPHEUS" presented in Figure 17.

To facilitate the reviewing process we provide a virtual machine image available at `https://www.google.com/drive/`. We have installed MORPHEUS on this virtual machine and created bash scripts to run MORPHEUS and collect the results.

After downloading and importing the virtual machine, the reviewer can find all files related to MORPHEUS in the directory `/home/morpheus/ae-pldi17-morpheus`. In the remainder of this report, we will refer to subdirectories of this directory. For instance, in the subdirectory `artifacts` the reviewer can find the files "*figure16.csv*" and "*figure17.pdf*" that correspond to the main results presented in the paper. The subdirectory `artifacts/solutions` provides a description of each benchmark and shows how MORPHEUS can be used to solve them. In particular, the reviewer can find: (i) Stackoverflow post, (ii) input-output tables, (iii) synthesized solution by MORPHEUS, and (iv) statistics on the running time and impact of deduction.

2. Reproducing the main results of the paper

All experiments in the paper were conducted using an Intel Xeon(R) computer with an E5-2640 v3 CPU and 32G of memory running Ubuntu 16.04. Running times may vary on different machines but the overall relative performance of the different specs should remain the same. For instance, in the paper "Spec 2" solved 78 out of 80 benchmarks and "No deduction" solved 54 out of 80 benchmarks with a 5 minute timeout. We run MORPHEUS inside of a virtual machine on a laptop using an Intel(R) with an i7-5600U and 8G of memory and it was able to solve 74 out of 80 benchmarks with "Spec 2" and 42 out of 80 benchmarks with "No deduction".

2.1 Reproducing "*Figure16.csv*"

The following steps should be done to reproduce the results in the subdirectory `artifacts/figure16.csv` which corresponds to Figure 16 in the paper.

1. Run the script:
   `$./run-morpheus-all.sh`
   Morpheus will solve the 80 benchmarks from the paper using the 5 variants ("spec2", "spec2-no-pe", "spec1", "spec1-no-pe", "no-deduction"). Solving all variants can be very time-consuming and it may take up to **12 hours** for this script to finish.
2. After completion, the log files from this experiment are available in subdirectory `output`.
3. The `.csv` file can be generated by executing the following script:
   `$./table-morpheus-all.sh output figure16.csv`

Since this experiment may take a very long time, we provide the log files from our experiments for the PLDI submission. These files are available in the subdirectory `logs`. The "*figure16.csv*" can be generated using the original logs by executing the following command:

```
$./table-morpheus-all.sh logs figure16.csv
```

Instead of running all variants of MORPHEUS, the reviewer can opt to run just a single variant. For instance, the reviewer can run MORPHEUS with "Spec 2" (our most precise specification) with the following command:

```
$./run-morpheus-variant.sh 1
```

This variant should terminate within an hour and the reviewer can then generate a `.csv` file that contains partial information presented in Figure 16:

```
./table-morpheus-variant.sh output/spec2 figure16-spec2.csv
```

2.2 Reproducing "*Figure17.pdf*"

Category	Description	#	No deduction		Spec 1		Spec 2	
			#Solved	Time	#Solved	Time	#Solved	Time
C1	*Reshaping* dataframes from either "long" to "wide" or "wide" to "long"	4	2	198.14	4	15.48	4	6.70
C2	*Arithmetic computations* that produce values not present in the input tables	7	6	5.32	7	1.95	7	0.59
C3	Combination of *reshaping* and *string manipulation* of cell contents	34	28	51.01	31	6.53	34	1.63
C4	*Reshaping* and *arithmetic computations*	14	9	162.02	10	90.33	12	15.35
C5	Combination of *arithmetic computations* and *consolidation* of information from multiple tables into a single table	11	7	8.72	10	3.16	11	3.17
C6	*Arithmetic computations* and *string manipulation* tasks	2	1	280.61	2	49.33	2	3.03
C7	*Reshaping* and *consolidation* tasks	1	0	✗	1	135.32	1	130.92
C8	Combination of *reshaping*, *arithmetic computations* and *string manipulation*	6	1	✗	3	198.42	6	38.42
C9	Combination of *reshaping*, *arithmetic computations* and *consolidation*	1	0	✗	0	✗	1	97.3
Total		80	54 (67.5%)	95.53	68 (85.0%)	8.57	78 (97.5%)	3.59

Figure 16. Summary of experimental results. All times are median in seconds and ✗ indicates a timeout ($>$ 5 minutes).