Multidimensional Repetitive Pattern Discovery

Linus W. Dietz

Software Technologies Research Group

January 15, 2016

Outline

Outline

Basis of this work

- Initial version of dsOli [White, 2014]
- Bachelor Thesis [Dietz, 2015], 9/14 – 3/15
- Hiwi work, 9/15 – today

Problem Statement

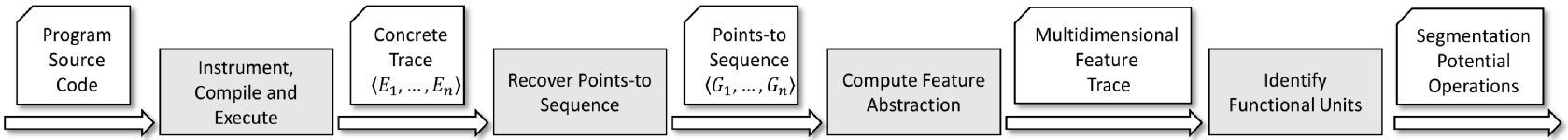

Figure 1: Preprocessing Steps [White, 2014]

- Find a suitable segmentation of a trace generated by an instrumented program.
- The trace is a multidimensional abstraction of heap memory operations from the viewpoint of each entry pointer.

Abstraction of Heap Memory

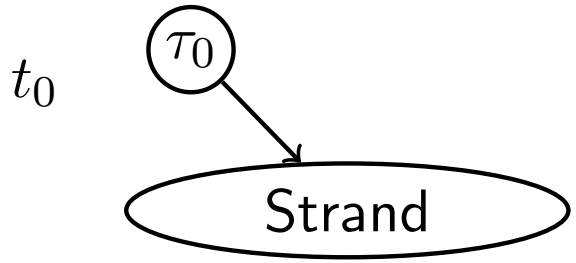

Figure 2: Abstraction of Heap Memory to the Feature Trace

Abstraction of Heap Memory

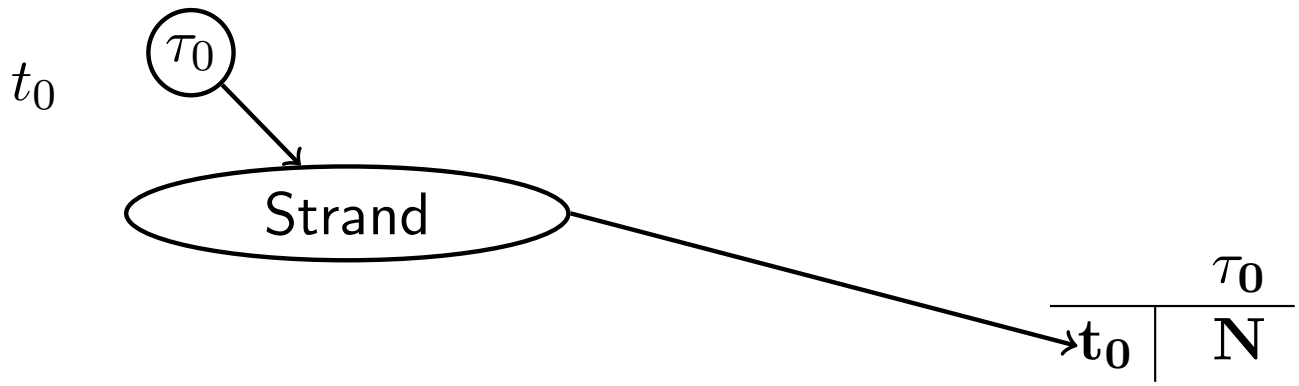

Figure 2: Abstraction of Heap Memory to the Feature Trace

Abstraction of Heap Memory

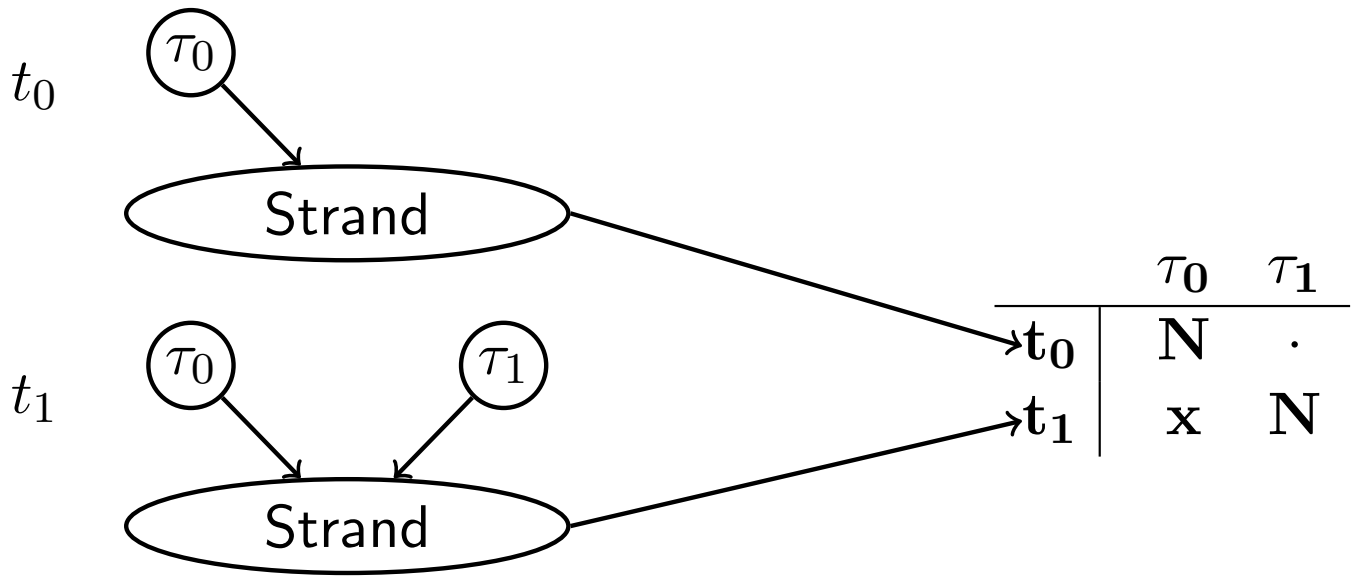

Figure 2: Abstraction of Heap Memory to the Feature Trace

Abstraction of Heap Memory

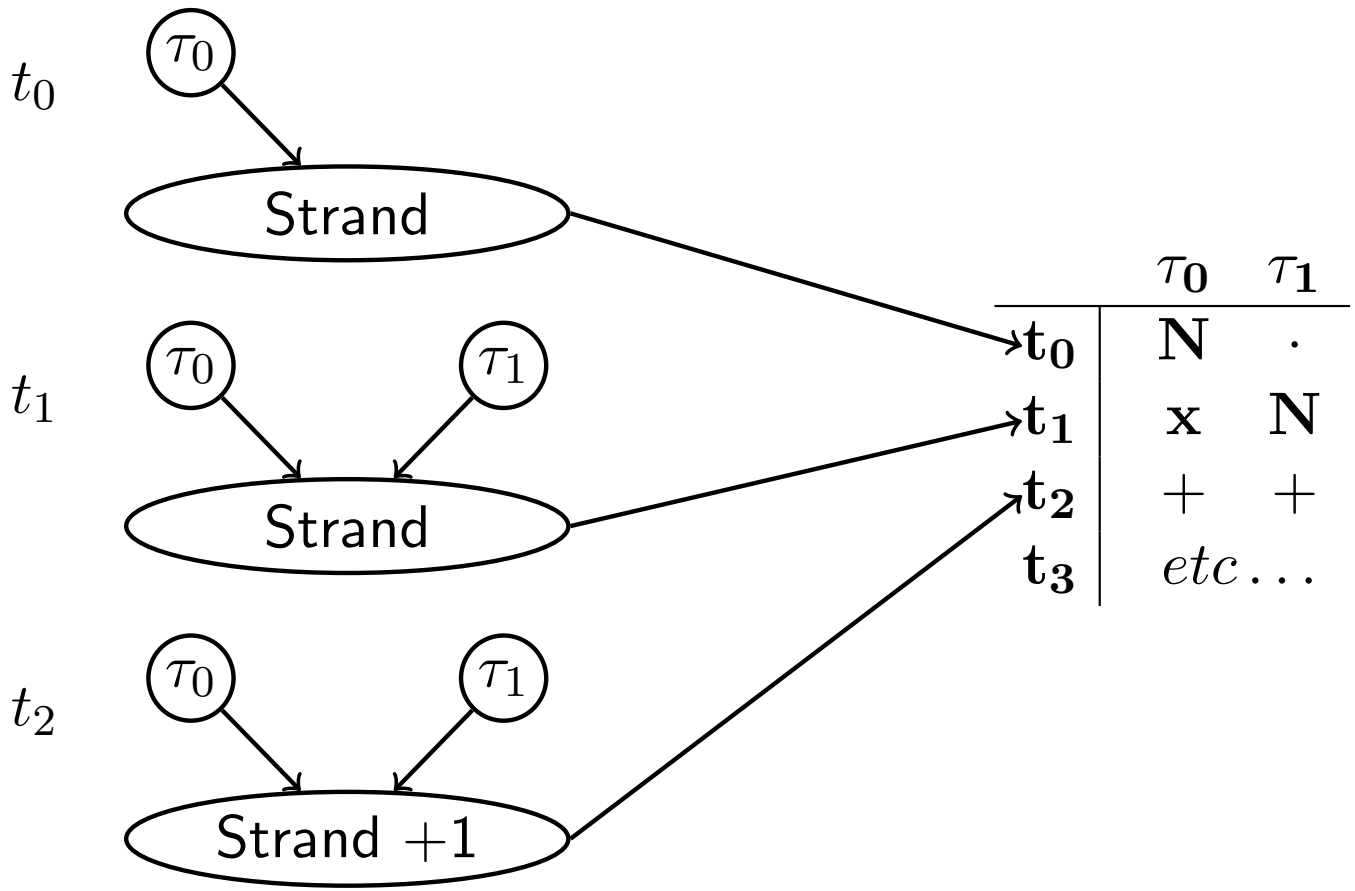

Figure 2: Abstraction of Heap Memory to the Feature Trace

Assumptions

- Each operation is invoked a suitable large number of times.
- More precise: each control path of an operation must be invoked sufficiently often.
- Each invocation of one control path of an operation must be identical (after the abstraction).
- Different operations must be distinguishable.

Outline

Related Work in Sequence Mining

- Bioinformatics:
  - Global sequence alignment [Needleman and Wunsch, 1970]
  - Local sequence alignment
    [Waterman et al., 1984, Wang et al., 1994]
- Basket analysis [Agrawal and Srikant, 1995]
- Process mining [van der Aalst, 2012]
- HCI: detecting repetitive user actions

Outline

Overview

- Genetic algorithm to search patterns.
- Fitness function uses Minimum Descriptive Length Principle (MDL) compression. [Rissanen, 1978]
- Evaluation of the approach using synthetic training data.
- Validation of segmentation using source code.

Terminology

- **Trace:** set of SubTraces
- **SubTrace:** sequence of Symbols, column of a Trace
- **Symbol:** representation of an atomic memory instruction
- **Pattern:** set of SubPatterns
- **SubPattern:** sequence of Symbols, column of a Pattern
- **Individual:** set of Patterns
- (initial) **seed:** Patterns with one 'short-running' SubTrace

Terminology

- **Trace:** set of SubTraces
- **SubTrace:** sequence of Symbols, column of a Trace
- **Symbol:** representation of an atomic memory instruction
- **Pattern:** set of SubPatterns
- **SubPattern:** sequence of Symbols, column of a Pattern
- **Individual:** set of Patterns
- (initial) **seed:** Patterns with one 'short-running' SubTrace

Goal: Find an Individual I, such that each Pattern of I corresponds to one mode of invocation of an operation in a given Trace.

Patterns

$\Pi_1 =$

	π_0	π_1	π_2	π_3
t_0	x	N	·	·
t_1	v	v	·	·
t_2	x	x	·	·
t_3	x	x	N	N
t_4	+	+	x	x
t_5	v	v	v	v
t_6	x	x	x	·
t_7	x	x	·	·
t_8	x	·	·	·

Figure 3: Insert Pattern

$\Pi_2 =$

	π_0	π_1	π_2	π_3
t_0	x	x	N	N
t_1	x	+	x	x
t_2	v	v	v	v
t_3	x	x	x	·
t_4	x	x	·	·

Figure 4: Push Pattern

Minimum Descriptive Length Principle

The Fitness Function of our Approach

- Minimize $L(H) + L(D|H)$,
  where $L(H)$ is the length of hypothesis H and $L(D|H)$ the length of the compressed data D given hypothesis H.
  [Rissanen, 1978]
- In our case:
  - $L(H) =$ number of Symbols of all Patterns in an Individual.
  - $L(D|H) =$ number of unmatched Symbols of the compressed Trace given an Individual $+$ one cost for each applied Pattern and SubPattern.

Genetic Algorithm Operators

- Horizontal expansion
- Horizontal contraction
- Vertical expansion
- Vertical contraction
- Pattern add
- Pattern remove

	π_0	π_1	π_2	π_3
t_0	x	N	·	·
t_1	v	v	·	·
t_2	x	x	·	·
t_3	x	x	N	N
t_4	+	+	x	x
t_5	v	v	v	v
t_6	x	x	x	·
t_7	x	x	·	·
t_8	x	·	·	·

Figure 5: A Pattern

Genetic Algorithm Operators

- **Horizontal expansion**
- Horizontal contraction
- Vertical expansion
- Vertical contraction
- Pattern add
- Pattern remove

	π_0	π_1	π_2	π_3
t_0	x	N	·	·
t_1	v	v	·	·
t_2	x	x	·	·
t_3	x	x	N	N
t_4	+	+	x	x
t_5	v	v	v	v
t_6	x	x	x	·
t_7	x	x	·	·
t_8	x	·	·	·

Figure 5: A Pattern

Genetic Algorithm Operators

- Horizontal expansion
- **Horizontal contraction**
- Vertical expansion
- Vertical contraction
- Pattern add
- Pattern remove

	π_0	π_1	π_2	π_3
t_0	x	N	·	·
t_1	v	v	·	·
t_2	x	x	·	·
t_3	x	x	N	N
t_4	+	+	x	x
t_5	v	v	v	v
t_6	x	x	x	·
t_7	x	x	·	·
t_8	x	·	·	·

Figure 5: A Pattern

Genetic Algorithm Operators

- Horizontal expansion
- Horizontal contraction
- **Vertical expansion**
- Vertical contraction
- Pattern add
- Pattern remove

	π_0	π_1	π_2	π_3
t_0	x	N	·	·
t_1	v	v	·	·
t_2	x	x	·	·
t_3	x	x	N	N
t_4	+	+	x	x
t_5	v	v	v	v
t_6	x	x	x	·
t_7	x	x	·	·
t_8	x	·	·	·

Figure 5: A Pattern

Genetic Algorithm Operators

- Horizontal expansion
- Horizontal contraction
- Vertical expansion
- **Vertical contraction**
- Pattern add
- Pattern remove

	π_0	π_1	π_2	π_3
t_0	x	N	·	·
t_1	v	v	·	·
t_2	x	x	·	·
t_3	x	x	N	N
t_4	+	+	x	x
t_5	v	v	v	v
t_6	x	x	x	·
t_7	x	x	·	·
t_8	x	·	·	·

Figure 5: A Pattern

Genetic Algorithm Operators

- Horizontal expansion
- Horizontal contraction
- Vertical expansion
- Vertical contraction
- **Pattern add**
- **Pattern remove**

	π_0	π_1	π_2	π_3
t_0	x	N	·	·
t_1	v	v	·	·
t_2	x	x	·	·
t_3	x	x	N	N
t_4	+	+	x	x
t_5	v	v	v	v
t_6	x	x	x	·
t_7	x	x	·	·
t_8	x	·	·	·

Figure 5: A Pattern

Genetic Algorithm Operators

- Horizontal expansion
- Horizontal contraction
- Vertical expansion
- Vertical contraction
- Pattern add
- Pattern remove

	π_0	π_1	π_2	π_3
t_0	x	N	·	·
t_1	v	v	·	·
t_2	x	x	·	·
t_3	x	x	N	N
t_4	+	+	x	x
t_5	v	v	v	v
t_6	x	x	x	·
t_7	x	x	·	·
t_8	x	·	·	·

Figure 5: A Pattern

Outline

Status Quo

- Maturing implementation and an automated tool chain around the genetic algorithm.
- Very stable results with perfect or near perfect segmentations for synthetic Traces with 2–3 different operations and 90–100 occurrences.
- Slight decrease in reliability for traces with 4 operations, yet usually still very good segmentations.
- ‘ls’-traces: several functional units are impossible to detect with the current approach, the selection of the initial seed becomes important.

Issues

- What is a good segmentation?
- Which data is realistic?
- Parameter estimation (≈ 10 params) is costly and would overfit training data.
- Which abstraction level do we want to achieve?

Limits of Approach

- Noisy data
- Threshold of repetitiveness
- Unmatchable functional units
- Practical limits regarding the number of different operations

Future Steps

- Visualization.
- Abstraction of Traces.
- Test applicability for obfuscated code and object code.

References I

[Agrawal and Srikant, 1995] Agrawal, R. and Srikant, R. (1995).
Mining sequential patterns.
In *Proceedings of the Eleventh International Conference on Data Engineering*, ICDE '95, pages 3–14, Washington, DC, USA. IEEE Computer Society.

[Dietz, 2015] Dietz, L. W. (2015).
Multidimensional Repetitive Pattern Discovery for Locating Data Structure Operations.
B. Sc. Thesis, Bamberg.

[Needleman and Wunsch, 1970] Needleman, S. B. and Wunsch, C. D. (1970).
A general method applicable to the search for similarities in the amino acid sequence of two proteins.
Journal of Molecular Biology, 48(3):443 – 453.

[Rissanen, 1978] Rissanen, J. (1978).
Modeling by shortest data description.
Automatica, 14(5):465 – 471.

References II

[van der Aalst, 2012] van der Aalst, W. (2012).
Process mining: Overview and opportunities.
ACM Trans. Manage. Inf. Syst., 3(2):7:1–7:17.

[Wang et al., 1994] Wang, J. T.-L., Chirn, G.-W., Marr, T. G., Shapiro, B., Shasha, D., and Zhang, K. (1994).
Combinatorial pattern discovery for scientific data: Some preliminary results.
In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, SIGMOD '94, pages 115–125, New York, NY, USA. ACM.

[Waterman et al., 1984] Waterman, M., Arratia, R., and Galas, D. (1984).
Pattern recognition in several sequences: Consensus and alignment.
Bulletin of Mathematical Biology, 46(4):515–527.

[White, 2014] White, D. H. (2014).
dsOli: Data structure operation location and identification.
In *Proceedings of the 22Nd International Conference on Program Comprehension*, ICPC 2014, pages 48–52, New York, NY, USA. ACM.

Questions ?

Linus Dietz
linus.dietz@uni-bamberg.de