Synthesizing Intensional Behavior Models by Graph Transformation

A Problem
Analysis
Input
Specification
Specification Recovery

Input
Specification

Analysis

Specification Recovery

Black-Box
Dynamic Analysis
Data Abstractions
Intensional Behavior Models

Is it relevant?
BEM Inference

top(push(Stack(), "a")) = "a"
size(push(Stack(), "a")) = 1

top(push(Stack(), "b")) = "b"
size(push(Stack(), "b")) = 1

BEM Inference

top(pop(push(Stack(), "a"))) throws Error
size(pop(push(Stack(), "a"))) = 0

top(pop(push(Stack(), "b"))) throws Error
size(pop(push(Stack(), "b"))) = 0

BEM Inference

top(push(push(Stack(), "a"), "a")) = "a"
size(push(push(Stack(), "a"), "a")) = 2

top(push(push(Stack(), "b"), "a")) = "a"
size(push(push(Stack(), "b"), "a")) = 2

BEM Inference

top(pop(push(push(Stack(), "a"), "a"))) = "a"
size(pop(push(push(Stack(), "a"), "a"))) = 1

top(pop(push(push(Stack(), "b"), "a"))) = "b"
size(pop(push(push(Stack(), "b"), "a"))) = 1
BEM Inference

\[
\text{top}(\text{pop}(\text{push}(\text{push}(\text{Stack()}, \text{"a"})), \text{"a"}))) = \text{"a"}
\]

\[
\text{size}(\text{pop}(\text{push}(\text{push}(\text{Stack()}, \text{"a"})), \text{"a"}))) = \text{1}
\]

\[
\text{top}(\text{pop}(\text{push}(\text{push}(\text{Stack()}, \text{"b"})), \text{"a"}))) = \text{"b"}
\]

\[
\text{size}(\text{pop}(\text{push}(\text{push}(\text{Stack()}, \text{"b"})), \text{"a"}))) = \text{1}
\]

**Discriminating Operation**

- **Stack**
  - size(): 2
  - top(): \text{"a"}

- **Stack**
  - size(): 2
  - top(): \text{"a"}

- **Stack**
  - size(): 1
  - top(): \text{"b"}

- **Stack**
  - size(): 1
  - top(): \text{"b"}

**Inferred BEM**

- **Stack**
  - size(): 1
  - top(): \text{"a"}

- **Stack**
  - size(): 2
  - top(): \text{"b"}

- **Stack**
  - size(): 2
  - top(): \text{"a"}

- **Stack**
  - size(): 2
  - top(): \text{"b"}

- **Stack**
  - size(): 2
  - top(): \text{"a"}

- **Stack**
  - size(): 2
  - top(): \text{"b"}

**Canonical Terms**

- **Stack**
  - size(): 0
  - top() -> \text{Error}

- **Stack**
  - size(): 1
  - top(): \text{"a"}

- **Stack**
  - size(): 2
  - top(): \text{"b"}

- **Stack**
  - size(): 1
  - top(): \text{"a"}

- **Stack**
  - size(): 2
  - top(): \text{"b"}

- **Stack**
  - size(): 2
  - top(): \text{"a"}

- **Stack**
  - size(): 2
  - top(): \text{"b"}

- **Stack**
  - size(): 2
  - top(): \text{"a"}

- **Stack**
  - size(): 2
  - top(): \text{"b"}
Rule Synthesis

1:Stack
size() = x0
top() = x1
CT = x2

RHS

1:Stack
size() = x0
top() = x1
CT = x2

push(p0)

Stack
size() = y0
top() = y1
CT = y2

3

y0 = x0 + 1; ...
y1 = p0; ...
y2 = append(x2, push(p0)); ...

Invariant Detection also for Preconditions
Ad-hoc Invariants for Canonical Terms

Intensional Behavior Model

LHS

1:Stack
size() = x
top() = y
CT = z

RHS

1:null

Stack
size() = 0
(top() = Error
CT = Stack)

1:Stack
size() = x + 1
top() = p
CT = append(z, push(p))

1:Stack
size() = x + 1
top() = y
CT = z

push(p)

2:Stack
size() = x2
top() = y2
CT = z2

1:Stack
size() = x2
top() = y1
CT = z1

pop()

2:Stack
size() = x2
top() = y2
CT = z2

AC: z1 = append(z2, push(y1))

Related Work

* The pioneer: Daikon [Ernst et al]
* Recent Advances in Invariant Detection: DySy [Csallner et al]
* Inductive Logic Programming [Sankaranarayanan et al]
* Temporal API Rules [Acharya et al]
* Behavior Model Extraction [Dallmeier et al, Lorenzoli et al]
* Best Competitive: Heureka [Henkel and Diwan], infers Algebraic Specifications

Evaluation

* Evaluated against Heureka
* Tested with java.util Containers
* Tested with relevant Data Abstractions

Available at: http://home.dei.polimi.it/mocci/spy/
Evaluation Setting

- Results depend on how many ad-hoc invariants have been added
- We added invariant patterns to cope with specific behaviors of the data abstractions
- Obviously, there exist more complex data abstractions whose spec is hard to infer
- Results have no absolute value, they show a relative improvement
  i.e., who performs better with those Data Abstractions within the above limits!

Conclusions

- A method for Specification Recovery through Black-Box Dynamic Analysis
- Works for Containers and Data Abstractions
- Retrieves Precise Specifications for relevant and complex data abstractions
- Several Future Work for Improvement: Nondeterministic Components/Modeling, More General Stateful Components

Thank you :)