Abstract Interpretation of High-Level Transformations

Ahmad Salim Al-Sibahi, IT University of Copenhagen

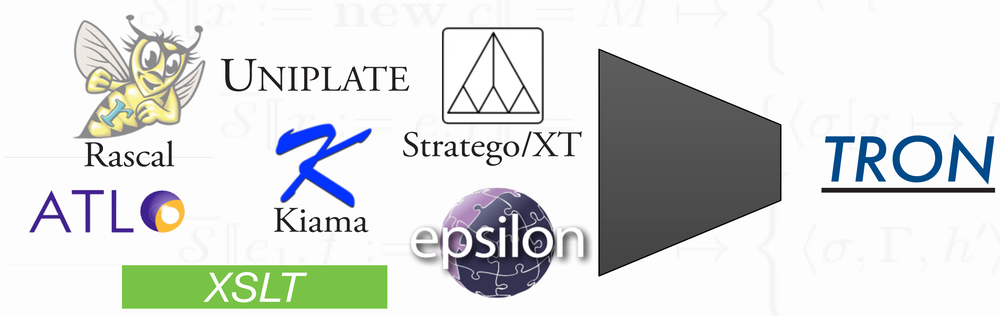
Motivation

Verifying transformations is challenging because they manipulate rich structures like programs and domain models. Increasing trustworthiness is key to wider use in critical systems.

"77% of the participants consider that refactoring comes with a risk of introducing subtle bugs and functionality regression" (Kim, Zimmermann & Nagappan – FSE '12)

High-Level Transformations

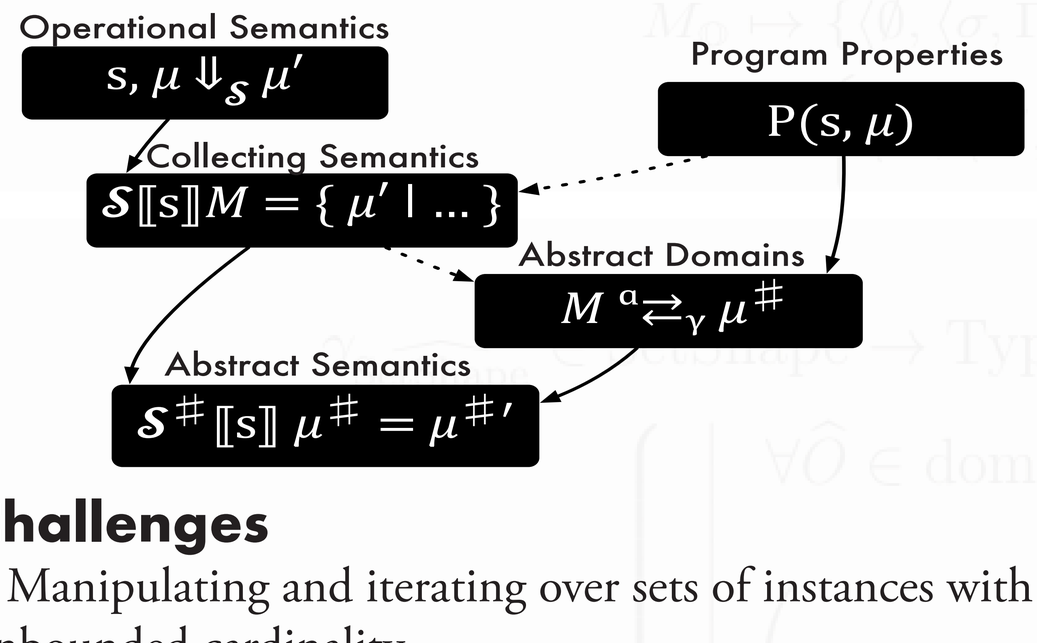
TRON (Al-Sibahi et al. – SLE '16) is a formal IMP-like language that captures key aspects of high-level transformation languages.



The example below presents a simplified field renaming refactoring in TRON. Type-directed matching allows it to be concisely written, since all relevant field access expressions can be retrieved using a single line of code (7).

```
1 input:  target_class: Class, old_field: Field, new_field: Field
2 precondition: old_field ∈ target_class.fields
3                ∧ new_field ∉ target_class.fields
4
5 target_class.fields :=
6       (target_class.fields \ old_field) ∪ new_field
7 foreach faexpr ∈ target_class match* FieldAccessExpr do
8   if faexpr.field = old_field ∧
9       faexpr.target.type = target_class then
10        faexpr.field := new_field
11  else skip
```

Method Overview

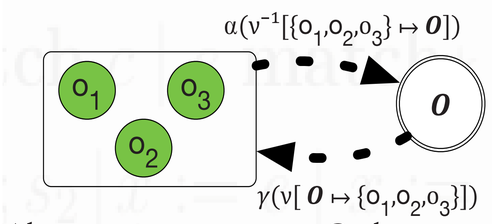


Challenges

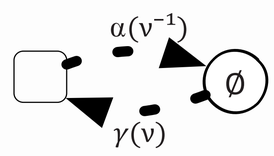
- Manipulating and iterating over sets of instances with unbounded cardinality
- Performing type-directed matching without considering all intermediate shapes of the instance graph
- Efficiently handling of aliasing and subtyping

Select Heap Abstractions

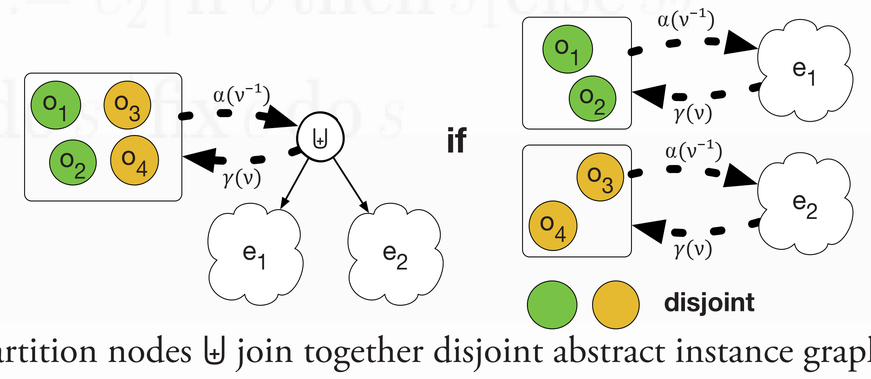
The abstractions are parametrized by a valuation (Laviron et al. - ESOP '10), which maps between concrete and abstract instances.



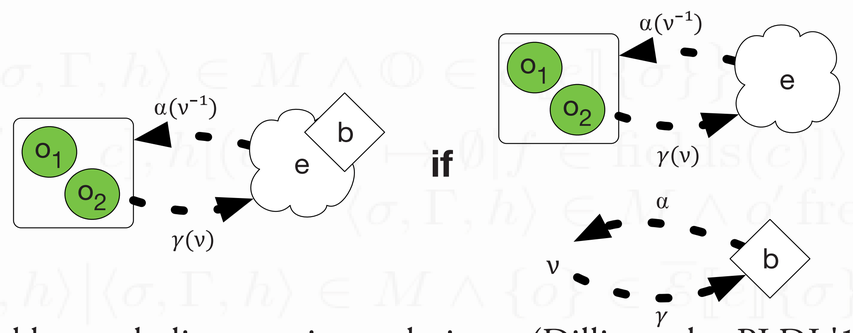
Abstract instance sets $\boldsymbol{O}$ abstract over sets of instances with unbounded size.



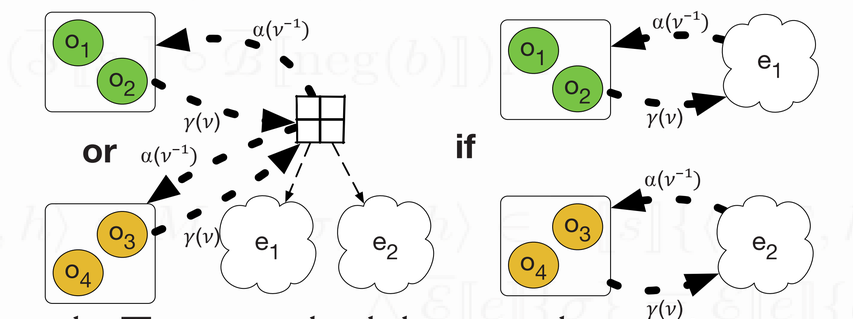
Empty set nodes $\emptyset$ abstract over empty sets of instances.



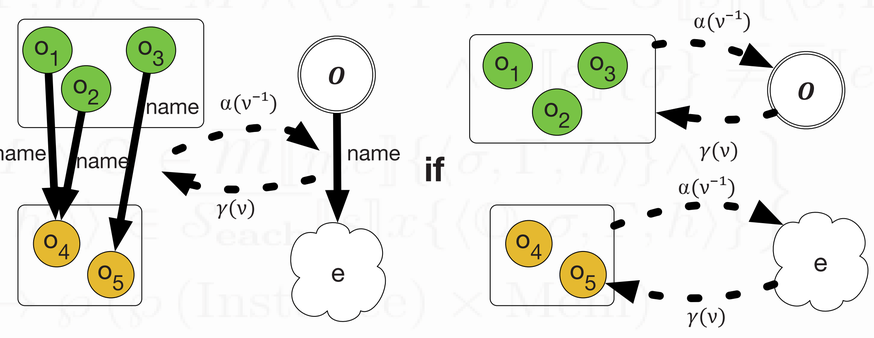
Partition nodes $\uplus$ join together disjoint abstract instance graphs.



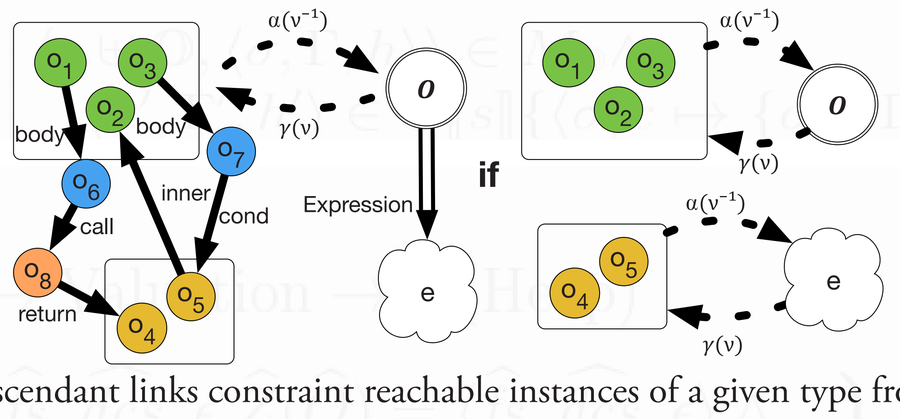
Inspired by symbolic execution techniques (Dillig et al. - PLDI '11), guard nodes $\diamond$ represent instance graphs that depend on a condition *b*.



Choice nodes $\boxplus$ represent local alternatives between instance graphs.



Field links can go from abstract instances (instance sets) to instance graphs and constraint the shape of the heap.



Descendant links constraint reachable instances of a given type from abstract instances (instance sets), abbreviating intermediate structure.

Abstract Semantics

Over-approximative execution of first example statement

Acknowledgements

Parts of this work were carried out at INRIA Rennes, and I thank my host Thomas Jensen for his continuous feedback. I thank Lars Birkedal and his team (Aarhus University), and Mihaela Sighireanu and Constantin Enea (Paris Diderot University) for the short visits I had in early stages of this work. I thank Jeff Pelz for suggesting the name TRON. I thank Andrzej Wąsowski and Aleksandar Dimovski for supervising my Ph.D.

IT UNIVERSITY OF COPENHAGEN