Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 334 (2018) 17–29

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Fuzzy Set Abstraction

Jacob Lidman and Josef Svenningsson[1]

Computer science and engineering,
Chalmers University of Technology,
Gothenburg, Sweden

Abstract

Program analysis plays a key part in improving modern software. Static (sound) analyses produce globally correct, but often pessimistic results while dynamic (complete) analyses yield highly precise results but with limited coverage. We present the Fuzzy set abstraction which generalizes previous work based on 3-valued logic. Our abstraction allows for hybrid analysis where static results are refined dynamically through the use of fuzzy control systems.

Keywords: Abstract interpretation, static program analysis, dynamic program analysis

1 Introduction

Static and dynamic analysis are complementary. Static analysis is sound because it summaries all possible executions, whereas dynamic analysis provides more precise information because it summaries the executions which actually happen in practice.

Over-approximation in static analysis is sometimes a sever problem for applications that rely on the results. Static alias analysis often produce point-to sets several times larger than dynamic alias analysis[8],[9] and in extension inhibits several opportunities for parallelization.

Being able to combine both kinds of analyzes can greatly improve results, for instance in non-functional verification (e.g. deducing worst-case benefit of compiler optimizations) when pessimistic assumptions about input state/environment is used. In this case, sound results are interesting at compile-time so that optimizations that are guaranteed to be detrimental is not applied. In contrast, complete results are interesting at run-time where the actual set of inputs are known and hence the benefit of an optimization can be accurately evaluated. The fuzzy dataflow framework[5] showed how program analyzes based on fuzzy logic can uncover

[1] Email: lidman@chalmers.se, josefs@chalmers.se

https://doi.org/10.1016/j.entcs.2018.03.003

optimization opportunities that classical frameworks would not. The generalization to many-valued fuzzy logics allow program properties to be true or false to a certain degree. The truth values are elements of the unit interval [2] and denote bias of the program property. For instance, a result of 0.1875 would indicate that the property tends to false since it is closer to 0 (false) than to 1 (true).

The increased expressiveness offered by using fuzzy logic in program analysis however motivates additional research into the properties of the analysis framework, in particular soundness and completeness.

We introduce the Fuzzy Set Abstraction that generalize the three-valued logic abstraction[10]. We present the theoretical foundation of the fuzzy set abstraction and prove soundness for the static analysis (Section 3.1). We also present a dynamic analysis (Section 3.2) where we use an adaptive fuzzy inference system from fuzzy control theory to gradually specialize the analysis results to improve accuracy.

2 Preliminaries

We briefly introduce several concepts from the fuzzy set community. Our static analyses manipulate fuzzy sets using predicate transformers, expressed using fuzzy logic (Section 2.1), and collector functions motivated by possibility theory (Section 2.2). Similarly our dynamic analyses start from the results of the static analysis and iteratively specialize it to increase the accuracy of our results. This process relies on a fuzzy classifier (Section 2.3).

2.1 Fuzzy set and logic

Fuzzy sets assign a partial membership to each element as opposed to classical sets where the membership is binary. We use the common point-wise ordering to relate two fuzzy sets: $\langle S, \mu_A \rangle \leq \langle S, \mu_B \rangle \Leftrightarrow \forall s \in S : \mu_A(s) \leq \mu_B(s)$.

Definition 2.1 Let S be a set of elements and μ_S a *membership function* that assigns a membership value from the unit interval $[0, 1]$ to each element. Then $\langle S, \mu_S \rangle$ is a fuzzy set.

A fuzzy set over a singleton set can be considered a description of partial truth. Fuzzy logic defines logical connectives to manipulate such fuzzy sets. Here, complement $\tilde{\neg}$ is often defined as negation (i.e., $1 - x$) and, in the Min-max fuzzy logic, the max operation is used for disjunction $\tilde{\vee}$ and min operator for conjunction $\tilde{\wedge}$.

Definition 2.2 Fuzzy logics $\langle \tilde{\wedge}, \tilde{\vee}, \tilde{\neg} \rangle$ satisfy the De Morgans laws. $\tilde{\wedge}$ and $\tilde{\vee}$ are two binary functions $[0, 1]^2 \to [0, 1]$ that are commutative, associative and monotonically decreasing/increasing and have identity elements ($x \,\tilde{\wedge}\, 1 = x$ and $x \,\tilde{\vee}\, 0 = x$). Similarly $\tilde{\neg}$ is a unary function $\tilde{\neg} : [0, 1] \to [0, 1]$ that is decreasing, involutory (i.e. $\tilde{\neg}(\tilde{\neg}(x)) = x$) and satisfy the boundary conditions $\tilde{\neg}(0) = 1$ and $\tilde{\neg}(1) = 0$ [3]. The

[2] To guarantee termination we use a finite congruence set of the unit interval.

[3] In the fuzzy logic literature the conjunction operator is called a Triangular norm (T-norm), the disjunction operator called Triangular conorm (S-norm) and complement the C-norm