MASARYK UNIVERSITY
FACULTY OF INFORMATICS

Parameter Synthesis from Hypotheses Formulable in CTL Logic

BACHELOR THESIS

Samuel Pastva

Brno, Spring 2015

Declaration

Hereby I declare, that this paper is my original authorial work, which I have worked out by my own. All sources, references and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Samuel Pastva

Advisor: John Foo, Ph.D.

Acknowledgement

I would like to thank my supervisor …

Abstract

The aim of the bachelor work is to provide . . .

Keywords

keyword1, keyword2 . . .

Contents

1 Introduction

Lorem Ipsum [?]

2 Algorithm

In this chapter, we describe the distributed algorithm that computes the assumption function $\mathcal{A}$.

2.1 Distributed Environment

In this section, we briefly describe the distributed environment assumed by our algorithm, in order to prevent any possible confusion.

We assume a distributed environment with fixed number of reliable processes connected by reliable, order-preserving channels (The order preservation can be relaxed to some extent). We also assume that each process has a fixed identifier and the set of all process identifiers is equal to the result set of the partition function. Each process can communicate directly (using the function SEND) with any other process (assuming it knows other process's identifier) and all messages that can't be processed directly are stored in a queue until they can be processed.

Several parts of the algorithm do not have explicit termination (they terminate by reaching deadlock - no messages are exchanged between processes). In such cases, suitable termination detection algorithm is employed to detect this deadlock and terminate computation properly. Our implementation uses Safra's algorithm [CITATION] for this purpose, but the related code has been skipped for easier readability.

We also divide algorithm description into three main parts: Process variables, Initialization and Message handler. First section describes data structures stored in process memory available during whole computation. Initialization section is executed exactly once and no messages can be received until it's finished. Message handler defines what should happen when message is received.

2.2 Algorithm outline

The main idea of the algorithm is described in REFERENCE and resembles other CTL model checking algorithms.

procedure CHECKCTL($\phi, \mathcal{K} = (id, f, \mathcal{P}, S, I, \xrightarrow{p}, L)$)
    $\mathcal{A} \leftarrow \{(p, s, \alpha, \mathtt{tt}) \mid p \in \mathcal{P} \wedge \alpha \in L(s)\}$
    for all $i < |\phi|$ **do**
        for all ψ **in** $cl(\phi)$ **where** $|\psi| = i$ **do**
            $\mathcal{A} \leftarrow$ CHECKFORMULA($\psi, \mathcal{K}, \mathcal{A}$)
        end for
    end for
end procedure

The algorithm starts by initializing the assumption function using the labeling function defined in kripke fragment. After that, it traverses the structure of formula, starting from smallest formulas and uses computed results to process more complex formulas. Function CHECKFORMULA computes all states and colors where formula ψ holds and returns assumption function updated accordingly. This is done using the local information contained in given kripke fragment, assumptions previously computed for smaller formulas and also by communicating with other processes. Note that only assumptions relevant for particular process are computed and returned (each process has information only about it's own state space).

2.3 Common operations

In this section, we define functions used to simplify the algorithm description. Let us fix a formula ϕ and a parametrised kripke fragment $\mathcal{K} = (id, f, \mathcal{P}, S, I, \xrightarrow{p}, L)$ as input of the algorithm.

Intuitively, function *validStates* : $cl(\phi) \times AS_{\mathcal{K}}^{\phi} \rightarrow S \times 2^{\mathcal{P}}$ computes a set of states and parameters where truth of given formula is assumed. It is also responsible for handling of boolean operators, since these can be computed without any inter-process communication.

$$validStates(\phi_1 \wedge \phi_2, \mathcal{A}) = \{(s, p) \mid \mathcal{A}(s, p, \phi_2) = \mathtt{tt} \wedge \mathcal{A}(s, p, \phi_2) = \mathtt{tt}\}$$

$$validStates(\phi_1 \vee \phi_2, \mathcal{A}) = \{(s, p) \mid \mathcal{A}(s, p, \phi_2) = \mathtt{tt} \vee \mathcal{A}(s, p, \phi_2) = \mathtt{tt}\}$$

$$validStates(\neg\phi, \mathcal{A}) = \{(s,p) \mid \mathcal{A}(s,p,\phi_2) = \texttt{ff}\}$$

$$validStates(\phi, \mathcal{A}) = \{(s,p) \mid \mathcal{A}(s,p,\phi) = \texttt{tt}\}$$

Function *predecessors* : $S \rightarrow S \times 2^{\mathcal{P}}$ computes set of direct predecessors of given node including the color sets labeling the appropriate transitions.

$$predecessors(to) = \{(from, P) \mid P = \{p \mid from \xrightarrow{p} to\}\}$$

Symmetrically, function *successors* : $S \rightarrow S \times 2^{\mathcal{P}}$ computes set of direct successors of given node including the color sets labeling the appropriate transitions.

$$predecessors(from) = \{(to, P) \mid P = \{p \mid from \xrightarrow{p} to\}\}$$

2.4 Exist Next Operator

Process variables:
$\mathcal{K} = (id, f, \mathcal{P}, S, I, \xrightarrow{p}, L)$ ▷ Kripke fragment
$\phi = \mathrm{EX}\phi_1$ ▷ CTL formula
$\mathcal{A}$ ▷ Initial assumption function
procedure INIT
  for all *state* **in** $validStates(\phi_1, \mathcal{A}, \mathcal{K})$ **do**
    for all $(pred, tranCol)$ **in** $predecessors(state, \mathcal{K})$ **do**
      SEND$(f(state), (pred, \mathcal{P} \cup tranCol))$
    end for
  end for
end procedure
procedure RECEIVE$(colSet, to)$
  $\mathcal{A} \leftarrow \mathcal{A} \cup \{(p, to, \phi) \mid p \in colSet\}$
end procedure

2.5 Exist Until Operator

Process variables:
$\mathcal{K} = (id, f, \mathcal{P}, S, I, \xrightarrow{p}, L)$ ▷ Kripke fragment

$\phi = \mathrm{E}\phi_1 \mathrm{U} \phi_2$ ▷ CTL formula
$\mathcal{A}$ ▷ Initial assumption function
procedure INIT
    for all *state* **in** $validStates(\phi_2, \mathcal{A}, \mathcal{K})$ **do**
        $\mathcal{A} \leftarrow \mathcal{A} \cup (\mathcal{P}, state, \phi)$
        for all $(pred, tranCol)$ **in** $predecessors(state, \mathcal{K})$ **do**
            SEND$(f(state), (state, pred, \mathcal{P} \cup tranCol))$
        end for
    end for
end procedure
procedure RECEIVE$(colSet, from, to)$
    $colSet \leftarrow colSet \cap valid(\phi_1, to, \mathcal{A})$
    if $colSet \neq \emptyset$ **and** $colSet \setminus valid(\phi, to, \mathcal{A}) \neq \emptyset$ **then**
        $\mathcal{A} \leftarrow \mathcal{A} \cup (colSat, to, \phi)$
        for all $(pred, tranCol)$ **in** $predecessors(to, \mathcal{K})$ **do**
            SEND$(f(pred), (to, pred, colSat \cup tranCol))$
        end for
    end if
end procedure

2.6 All Until Operator

Process variables:
$\mathcal{K} = (id, f, \mathcal{P}, S, I, \xrightarrow{p}, L)$ ▷ Kripke fragment
$\phi = \mathrm{A}\phi_1 \mathrm{U} \phi_2$ ▷ CTL formula
$S = \emptyset$
$\mathcal{A}$ ▷ Initial assumption function
procedure INIT
    for all *state* **in** $validStates(\phi_2, \mathcal{K})$ **do**
        for all $(pred, tranCol)$ **in** $predecessors(state, \mathcal{K})$ **do**
            SEND$(f(state), (state, pred, \mathcal{P} \cup tranCol))$
        end for
    end for
end procedure
procedure RECEIVE$(colSet, from, to)$
    $S \leftarrow S \cup (to, from, colSet)$
    $colSet \leftarrow colSet \cap valid(\phi_1, to, \mathcal{K})$

if $colSet \neq \emptyset$ **and** $colSet \setminus valid(\phi, to, \mathcal{K}) \neq \emptyset$ **then**
  $\mathcal{A} \leftarrow \mathcal{A} \cup (colSat, to, \phi)$
  for all $(pred, tranCol)$ **in** $predecessors(to, \mathcal{K})$ **do**
    $\textsc{Send}(f(pred), (to, pred, colSat \cup tranCol))$
  end for
end if
end procedure

A First appendix

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

B Another appendix

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.