COMPUTING
SCIENCE

Poster Session at The Newcastle Connection 2012

Victor Khomenko (Ed.)
Abstract

The report contains 2-page summaries of the research presented at poster session at The Newcastle Connection 2012 event by PhD students of Schools of Computing Science and Electrical and Electronic Engineering, Newcastle University.
Bibliographical details

KHOMENKO, V. (ED.)
Poster Session at The Newcastle Connection 2012
[By] V. Khomenko (Ed.)
(Newcastle University, Computing Science, Technical Report Series, No. CS-TR-1348)

Added entries

NEWCASTLE UNIVERSITY

Abstract

The report contains 2-page summaries of the research presented at poster session at The Newcastle Connection 2012 event by PhD students of Schools of Computing Science and Electrical and Electronic Engineering, Newcastle University.

About the editor

Victor Khomenko obtained his MSc with distinction in Computer Science, Applied Mathematics and Teaching of Mathematics and Computer Science in 1998 from Kiev Taras Shevchenko University, and PhD in Computing Science in 2003 from Newcastle University. He was a Program Committee Chair for the International Conference on Application of Concurrency to System Design (ACSD'10). He also organised the Workshop on UnFOLDing and partial order techniques (UFO'07) and Workshop on BALSA Re-Synthesis (RESYN'09). In January 2005 Victor became a Lecturer in the School of Computing Science, Newcastle University, and in September 2005 he obtained a Royal Academy of Engineering / EPSRC Post-doctoral Research Fellowship and worked on the Design and Verification of Asynchronous Circuits (DAVAC) project. After the end of this award, in September 2010, he switched back to Lectureship. Victor’s research interests include model checking of Petri nets, Petri net unfolding techniques, verification and synthesis of self-timed (asynchronous) circuits.

Suggested keywords

THE NEWCASTLE CONNECTION 2012 EVENT
Globally Asynchronous Elastic Logic Synthesis

Johnson Fernandes
Prof. Alex Yakovlev, Dr. Danil Sokolov, Dr. Alex Bystrov

August 17, 2012

Abstract

The main aim of this project is to deliver an integrated design methodology for synthesising digital systems with mixed synchronous-asynchronous architectures. The proposed technique combines Globally Asynchronous Locally Synchronous (GALS) design with Elastic Logic principles. Elastic Logic and Asynchronous design are envisioned to share a common timing discipline which should simplify the GALS wrappers and lead to performance gains. Currently, research work is being dedicated to the development of EDA strategies for automating the partitioning of a digital system into GALS blocks.

This project is funded by the EPSRC grant EP/I038551/1.

Background

System-on-Chip (SoC) methodology is being increasingly chosen for complex digital system design owing to its scope for massive integration of functionality on a single chip. However, the ease of component reuse in SoCs is plagued by nanoscale design issues such as excessive power consumption, heat dissipation and EMI. Various techniques have been proposed to solve these issues in the traditional synchronous design framework such as clock gating, power gating, dynamic frequency/voltage scaling etc. These solutions cost long designer hours and are implemented by trading off reliability, power or performance owing to the strict timing discipline of the synchronous design flow.

Asynchronous design techniques resolve most of the design issues inherently, as they do not rely on a global clock. But they too would not pose an optimal design choice; due to the inability to reuse synchronous IP cores, lack of mature CAD tools, high overhead from communication protocol and high transitioning costs.

Globally Asynchronous Locally Synchronous (GALS) systems is considered a better alternative to designing SoCs. GALS system design, quoted as ‘best of both worlds’, is an approach that can exploit the advantages of asynchronous design and at the same time maximally reuse the products of synchronous design flow. This design technique divides a digital system into locally synchronous islands which communicate asynchronously by handshake mechanism. Due to the presence of multiple clock domains, GALS design provides genuine scope for power savings [1] and energy efficiency compared to its synchronous counterpart by solutions such as idling the clock when data not requested, optimum operating frequency for individual modules, dynamic voltage and frequency scaling etc [2, 3]. The problem with the existing GALS methods is that they are all of an assemble-and-verify paradigm. Lack of CAD tools supporting mixed synchronous-asynchronous design and absence of a GALS design method targeting a process of synthesis with optimisation have made it difficult for GALS to be adapted by industry.
**Project Hypothesis**

The principal approach in this project – Globally Asynchronous Elastic Logic Synthesis (GAELS) presents a novel GALS design methodology that synthesises Elastic GALS blocks from a given schematic of a digital system. Optimisation targets such as energy budget per cluster, balancing power profile to reduce EMI, area etc. will determine the size and complexity of these blocks.

Elasticity refers to the property of a system to adapt itself to variations in computation and communication delays. The latency insensitive signalling protocols of Elastic Logic [4], Valid and Stop signals, would allow for a timing-resilient interface between asynchronous and synchronous domains. Performance gains is expected merely from simplification of the GALS wrappers due to the common timing discipline. A thorough investigation into the theory and application of Elastic Logic to GALS design will be conducted in the research.

The digital system would be first divided into low granularity blocks with elastic interface. These blocks can be used as building blocks to form the locally synchronous islands. The system is then partitioned into such multi-block components called ‘Localities’ [5]. Localities are formed by optimising the system partitioning to both functional and physical criteria. Accurate theoretical modelling and experimental verification is required to determine the impact of each of these parameters on the size of the Elastic GALS module. A systematic process for development of theory and methods targeting automation of partitioning and incorporation of different optimisation scenarios will be followed in this research. The objectives of the GAELS project are listed below.

- Formalise new theoretical models for dataflow representation of the Elastic GALS systems.
- Formulate theory, methods and algorithms for partitioning into locally synchronous islands.
- Identify a front-end language for describing GAELS design.
- Develop methods for system architectural analysis and automation of GAELS components.
- Integration of new methods and algorithms into an industrial CAD environment.
- Design of a case study for evaluation and comparison against previous approaches.

**References**


Collaborative Modelling and Co-simulation for Resilient Embedded Control Systems

Ken Pierce, Carl Gamble, and John Fitzgerald
School of Computing Science, Newcastle University,
Newcastle upon Tyne, NE1 7RU, United Kingdom.
{kenneth.pierce,carl.gamble,john.fitzgerald}@ncl.ac.uk

Abstract. We present a short summary of the work of the DESTECS project, which aims to help engineers and software designers collaborate in designing resilient embedded control systems. We achieve this by allowing discrete-event (DE) models of software and continuous time (CT) models of physical systems to be combined into co-models and evaluated through co-simulation.

1 Introduction

Building dependable embedded control system presents numerous challenges to engineers. Increasing market pressures require a shorter time to market and reduced costs, with increased complexity arising from the need to incorporate distributed, networked controllers and improve system resilience. In addition, these systems require collaboration across disciplines, with engineers and software designers working together. One approach to dealing with this complexity is to build models in formally defined languages and run simulations in order to validate designs at an early stage, without resorting to expensive prototypes. Traditionally these models have been single-domain: software systems are modelled with discrete-event (DE) formalisms, while physical systems are modelled in continuous-time (CT) formalisms. The distinct cultures and abstractions of these formalisms can hinder collaboration. Conventional development processes tend to separate disciplines either concurrently, with a risky integration phase, or sequentially, with one discipline’s models being used as input to the next.

The DESTECS project aims to bridge this gap by allowing DE and CT models to be combined as co-models. A co-model contains a DE model of the controller software, a CT model of the physical system, with a contract defining the communication between them, allowing state and events to be shared during simulation (co-simulation). The individual models are simulated by their existing tools, with an intermediate co-simulation engine controlling the progress of simulated time. The primary benefit of the DESTECS approach is that co-models can describe systems that would be difficult to model within the CT or DE formalism alone. Additionally, it allows engineers and software designers to

1 http://www.destecs.org/
collaborate using a single shared model. This can uncover problems at an earlier design stage than would be discovered by traditional sequential or parallel development with single-domain models.

In addition to the core concept of co-modelling and co-simulation, DESTECS also aims to support designers in modelling faults and fault tolerance mechanisms to achieve resilience and to support them in design space exploration (DSE), allowing designers to make better informed decisions by performing multiple co-simulations representing different designs and scenarios.

In DESTECS, discrete-event models are expressed using the Vienna Development Method’s (VDM) formal specification language, supported by the Overture\(^2\) tool. VDM supports object-orientation and concurrency \(^2\), and provides features to describe real-time embedded systems \(^3\). Continuous time models are built, simulated and visualised using the 20-sim\(^3\) tool \(^1\). This allows the dynamics of the plant to be modelled in several ways, including the powerful bond graph notation.

2 Results

The main results of the DESTECS project are:

– A prototype tool, based on the Eclipse platform, that can define co-models and perform co-simulations.
– Methodological guidelines that complement to the tool, covering topics including: production of initial co-models; modelling of realistic and erroneous behaviours; patterns for fault tolerance; and design of experiments for design space exploration (DSE).
– Industrial case studies demonstrating the tools and methodologies in the areas of personal transportation, excavation, and envelope filling, with smaller studies in space systems and robotics.
– A proof-of-concept demonstration the use of automated co-model analysis (ACA) to aid the exploration of the design space of a line-measuring robot.
– Production of training materials and the running of a five-day summer school, attended by 20 students representing 15 nationalities.

References


\(^2\)http://www.overturetool.org/
\(^3\)http://www.20sim.com/
1. **INTRODUCTION**

Nowadays, business processes become the backbone of organisations to automate and increase the efficiency and effectiveness of their services and products. The rapid growth of the Internet and other Web based technologies sparked competition between organisations to present faster, cheaper and smart environment for customers. In response to these requirements, organisations are examining how their business processes may be evaluated in order to improve business performance. Quantitative evaluation is a key factor of business process (BP) performance analysis to evaluate and improve the organisations processes. Performance analysis focuses on quantitative evaluation of BP such as service levels, throughput times, and resource utilisation. Performance analytical modelling techniques, such as Stochastic Petri Nets (SPN) [1] have been broadly used for performance analysis of BP. These techniques provide a powerful modelling and analysis tools to determine the effects of various parameters on performance indicators of BP. However, despite the importance of quantitative analysis of BP, hardly any studies are found in the literature that shows the use of models for analysis of the quantitative behaviour and optimisation of BP [2].

2. **GENERIC FRAMEWORK**

The proposed framework consists of three modules:

- Modelling and mapping module: A module used to define BP in a formal language and mapped it to stochastic model.
- Analytical module: Used to analysis stochastic model by standard analytic tool support used as a solver.
- Algorithmic module: A mathematical module used to solve the desirable algorithms of BP improvement and optimisation (i.e. availability, Scheduling)

3. **METHODOLOGY**

- Insert BP model in a formal modelling language.
- Mapping BP into a Stochastic Petri Nets (SPN) model according to following relation: Task → Transition, Resources → Place and Relation → Arc. The expected execution time of task \( eet_t = \lambda^{-1} \), where \( \lambda \) equal the firing delay of transition of SPN model.
- Compute the performance metrics of SPN model using stochastic Petri net analysis tools. (i.e. Stochastic Petri Net Package (SPNP) [3].
Solve the mathematical algorithms of BP improvement and optimisation.

4. CONCLUSION

This work introduces a generic framework for quantitative evaluation of business process that can be used for improve and optimise any BP. The framework consists of three modules; modelling and mapping module, analytical module and algorithmic module. A software support tools build in Java using the Eclipse platform used for design and execute framework modules. The execution of framework is presented in an automated way.

REFERENCES


1. Introduction

WeDRisk [1] is an approach designed to manage web and distributed (WD) software development risks. The WeDRisk approach aims to address the weaknesses of existing approaches to Risk Management (RM), which are less able to deal with the specific challenges of WD development [1]. A key part of the approach is flexibility, to deal with the rapid evolution which is typical of such developments. This flexibility is achieved by customization of the management type (i.e. Plain RM is an essential RM when the resources are limited or the time is critical, and Deep RM is a full RM when the resources and time are enough or the experience is enough), and by providing a method for coping with atypical risks. A number of evaluation methods were used to evaluate the WeDRisk approach including peer reviews, two controlled experiments [2], case study and expert evaluation. This paper summarizes a controlled experiment [3,4] that has been used to evaluate some central parts of the approach (estimation, customization and atypical risk modules), in terms of their usefulness, ease of understanding and usability.

2. Methods

This experiment aims to evaluate some novel aspects of the WeDRisk approach (see Figure 1), mainly, the consideration and estimation of WD factors, risk management customization and atypical risks concept and absorbing mechanism. We recruited by email about 35 subjects to participate in this experiment. Of these, we selected 24 subjects to participate in this experiment as they met the conditions (experience and knowledge in software development) [3]. The subjects were researchers, PhD students and visiting researchers at the School of Computing Science/Newcastle University. The subjects were either involved in software projects or had attended software engineering courses. In order to improve the design of the experiment a pilot study was carried out before starting the real experiment sessions. The pilot study was conducted using another group of participants. As a result of the pilot study a number of issues were addressed and improved for the real experiment including: the estimated time required to complete tasks; the sequence of tasks; instructions to subjects; data collection procedure and risky situation design [3].
nature of the test the subjects are divided into two groups (control and experimental). The experimental group subjects use the WeDRisk modules, but the control group subjects do not use these modules and they depend on their knowledge and experience.

3. Results

**Risks Estimation:**

An improved equation to estimate WD development risks called Total Risk Estimation Value TREV equation (2) was evaluated. The TREV equation is intended to consider WD factors (WDF) (i.e. Sites Dependency, Sites distributions and Communication). Compared with the existing RE equation (1) the evaluation result illustrated that the TREV equation is more suitable to estimate and consider WD development risks.

\[
RE = \text{RiskProb} \times \text{RiskMag} \quad \text{(1)}
\]

\[
\text{TREV} = RE \times \sum \text{WDF} \quad \text{(2)}
\]

WD estimation matrix is a matrix used to estimate the total of WDF values. The experiment result demonstrated how the WD estimation matrix is useful, understandable and helpful to estimate and consider WDF and there are consensuses in the estimated values. However, some subjects suggested including other factors in the WDF estimation matrix such as sites reliability.

**RM Customization:**

The customization matrix has helped the experimental group subjects to decide what type of RM (plain or deep) is suitable for each injected risky situation. The control group subjects have used more time than the experimental group subjects and there were inconsistencies in their decisions compared with the experimental group decisions. Moreover, in some situations the control group subjects were not able to decide and they have asked for the support many times. The experiment result showed how the matrix was easy to use, simple and how it gives quick results. Generally, the subject were agreed with the idea that the RM customization is useful, saves time and effort and it helps when there is a limitation in time or resources.

**Atypical Risks:**

The experiment result showed that the experimental group subjects who used the WeDRisk atypical module were able to act systematically and absorbed the injected atypical risk, whereas, the control group subjects act unsystematically and they got confused.

4. Conclusions

WeDRisk is an approach designed to manage web and distributed software development risks. The WeDRisk approach is intended to be flexible and able to deal with the rapid evolution nature of the WD developments. This paper has described a controlled experiment that was used to evaluate some novel aspects and modules of WeDRisk approach (i.e. estimation, customization and atypical risk modules), using some injected risky situations. From the experiment result it can be concluded that the evaluated WeDRisk modules have successfully and effectively dealt with the injected risky situations. The experiment result illustrated how these modules cover and considers the WD factors and they are useful, understandable and easy to use. The result also exhibited that there are some improvement could be done on the modules, which came in terms of suggestions, observations or from result analysis (e.g. TREV equation can be generalized and used to estimate risks in other types of software developments).

References


A Generic Logging Architecture Template to Mitigate the Risks Associated with Spam Activities in Infrastructure as a Service Cloud

Winai Wongthai*, Francisco Liberal Rocha*, Aad van Moorsel*
*School of Computing Science
Newcastle University, Newcastle upon Tyne, Tyne and Wear, UK
Email: winai.wongthai, f.e.liberal-rocha, and aad.vanmoorsel@ncl.ac.uk

Abstract—Infrastructure as a Service (IaaS) offers cloud-based computer infrastructure with raw storage and networking, typically on a pay-per-use basis. The Cloud Security Alliance (CSA) defines seven main threats to the security of cloud services. Logging systems provide evidence to support provider accountability and in this paper we research the extend to which such logging system help mitigate risks and/or assist in defending against the threats identified by CSA. A generic architecture 'template' of logging systems is proposed that encompasses all possible instantiations of logging solutions for IaaS cloud. We identify a logging solution to mitigate the risks associated with CSA threat 1 (related to spam activities). We argue that the architecture template can be used to facilitate systematic analysis of logging systems before deploying them in the real world.

I. INTRODUCTION

Many people argue that the cloud has the potential to transform the IT industry in a wide variety of application areas [1]. Trust of consumers in a cloud offering is extremely important for its continued proliferation. The CSA (https://cloudsecurityalliance.org) has extensively published on this topic, such as Top Threats to Cloud Computing report [2]. This document is used in this research, providing the basic threats for which we identify logging-based accountability solutions.

Although by itself not sufficient to mitigate the risks associated with the threats, a logging system is an important aspect of any accountability solution in the cloud, as argued by many researchers [3], [4]. Recent cloud accountability/monitoring-related solutions and proposals include as [5], [6]. Ideally, a logging system can be analyzed with respect to its features and achievable goals before deployment. Summary of Contributions: We propose a generic logging architecture template for IaaS and discuss how it can be used to mitigate the risks associated with the CSA threats. We then design and implement a logging system to mitigate CSA threat 1, which refers to the use of cloud computing for e-mail spamming and similar activities.

II. BACKGROUND

Figure 1 shows the IaaS architecture. A hw is a machine that works as a host of all domains (dom0 and domUs), and is managed and owned by the provider. A hypervisor is a layer of software that runs directly on the hw replacing the operating system (OS). Dom0 is a privileged domain guest OS that is launched by the hypervisor. It directly accesses the hw and manages domUs. A domU is an unprivileged domain guest OS that runs on top of the hypervisor, but has no direct access to the hw. It is virtually owned by the customer, and operates independently in the system. However, domU is actually launched and controlled by dom0. We study all CSA threats in [2]. However, in our case study, we focus only threat 1. For example, people with bad intentions can register legitimately to use domUs with details of a credit card, and later they engage in malicious activities (e.g., spamming).

Accountability is necessary in order to mitigate the risks associated with all CSA threats, and thus enable customers to securely and confidently adopt IaaS. One important mechanism for accountability is monitoring, which deploys a logging system as its core. Accountability in IaaS roots on similar concept with Accountable Cloud (AC) that is proposed by Haeberlen [4]. AC is the cloud that can answer questions asked by consumers and can provide reliable evidence about how their data is processed and handled.

III. THE PROPOSED ARCHITECTURE TEMPLATE

The template is shown in Figure 2. All defined components of the template can be divided into three sets. The first one is the IaaS set of components (all white boxes) which are derived from the IaaS architecture (Figure 1). The next ones are the logging processes set and the log files set of components (shaded boxes) which can be created and then located into the first set. They are the keys of the proposed architecture template. The logging processes set composes of logging processes (Px, x=1,...,4) and lib0. The log files set composes of Fy (y=1,...,4). We also provide description, examples, functions, and, locations in the IaaS environment.
of each component in all sets.

IV. A CASE STUDY

This case study is an identification of the appropriate logging system, Figure 3, based on the proposed template to mitigate the risks associated with CSA threat 1. It demonstrates how to use the template to design the logging system architecture, and how to analyze the security of this new built architecture. It simulates spam activities by assuming that an owner of domU sends a spam email to a victim (Figure 4 as an example). The mail command in Figure 4 has three arguments. They are ag1 (-s), ag2 (spamSubj), and ag3 (winai.wongthai@ncl.ac.uk). The goal of this case study is to capture c1’s ag2-3 (Figure 5, the second and third lines from the bottom), and then write them to a database as the log files.

With the proposed template, we can have an appropriate logging system which can be deployed in the IaaS real world with its security concerns. By combining arguments of both HP [6] and CSA [2], the use of a monitoring approach is possible to mitigate the risks associated with all CSA threats. Thus, we believe that this template can be used to instantiate logging system to mitigate the risks associated with threat 2-7.

VI. CONCLUSION

We propose a generic logging architecture template, and describe components for the purposes of mitigating risks associated with all CSA threats. We provide a case study of how to use the proposed architecture template for security analysis of logging systems. The proposed template and the case studies can be a starting point to build and deploy systematic logging systems in the IaaS with security concerns to truly mitigate the risks associated with all CSA threats, furthermore, the lack of trust issue for IaaS which will benefit both the customers and the providers.

REFERENCES


Basic CCS\textsuperscript{dp}: an Algebra for Dynamic Process Reconfiguration

Anirban Bhattacharyya
School of Computing Science, Newcastle University
Email: A.Bhattacharyya@ncl.ac.uk
4th/5th September 2012

1. Contribution and Focus
The contribution of this research is a novel way of formally modelling and analyzing dynamic process reconfiguration in dependable systems. Existing research on software reconfiguration has focused on link mobility and planned reconfiguration. The focus of this research is on unplanned process reconfiguration.

2. Process Syntax
Let \( P \) be the set of processes in basic CCS\textsuperscript{dp}. Let \( N \) be the countable set of names (e.g. \( a, b, c \)) that represent both input ports and input actions of the processes in \( P \); and let \( \bar{N} \) be the countable set of complementary names (e.g. \( \bar{a}, \bar{b}, \bar{c} \)) that represent both output ports and output actions of the processes in \( P \). The interaction between complementary actions (such as \( a \) and \( \bar{a} \)) is represented by the special action \( \tau \), which is internal to a process. By convention, \( \forall l \in N(\bar{l} \neq l) \). Let \( PN \) be the countable set of names (e.g. \( A, B, C \)) of the processes in \( P \). The sets \( N, \bar{N}, \{\tau\} \) and \( PN \) are assumed to be pairwise disjoint. Let \( L \) be the set of names that represent both ports and actions of the processes in \( P \), where \( L \equiv N \cup \bar{N} \). Let \( I \) be the set of input and output ports/actions of the processes in \( P \), and their internal action (\( \tau \)), where \( I \equiv L \cup \{\tau\} \). The syntax of a process \( P \) in \( P \) is defined as follows:

\[
P ::= PN<\beta> \ | \ M \ | \ P|P \ | \ \overline{P}
M ::= 0 \ | \ \alpha.P \ | \ M+M
\]

where \( PN \in PN, \tilde{\beta} \) is a tuple of \( \beta \)-values and \( \beta \in N \cup \bar{N} \), and \( \alpha \in N \cup \bar{N} \cup \{\tau\} \).

3. Positive Processes
We define the set positive processes of \( P \) in order to retain the identity property of 0 in parallel compositions. Let \( P^+ \) be the set of positive processes of \( P \), where \( P^+ \) is defined to be the smallest subset of \( P \) that satisfies the following conditions:

1. \( \forall \alpha \in I \forall p \in P \ (\alpha.p \in P^+) \)
2. \( \forall p, q \in P \ (p+q \in P \land (p \in P^+ \lor q \in P^+) \implies p+q \in P^+) \)
3. \( \forall p, q \in P \ (p \in P^+ \lor q \in P^+ \implies p|q \in P^+) \)
4. \( \forall p \in P \forall q \in P^+ (\overline{\beta}.q \in P^+) \)
5. \( \forall \beta \in I \forall X \in PN (\beta.X \in P^+) \)

4. Labelled Transition System (LTS) Semantics
Let \( R \) be the countable set of reconfiguration actions of the processes in \( P \) (e.g. \( \tau_{rX}, \tau_{rY}, \tau_{rZ} \)) that create a process in \( P \), and let \( \bar{R} \) be the countable set of complementary reconfiguration actions of the processes in \( P \) (e.g. \( \bar{\tau}_{rX}, \bar{\tau}_{rY}, \bar{\tau}_{rZ} \)) that delete a process in \( P \) (see the Creat and Delet rules below). Each action in \( R \) is represented by \( \tau_{rX} \), with \( X \in P \), and \( \forall \tau_{rX} \in R \ (\bar{\tau}_{rX} \neq \tau_{rX}) \). The sets \( N, \bar{N}, \{\tau\}, \bar{R}, R, \bar{R} \) and \( PN \) are assumed to be pairwise disjoint. Let \( C \) be the set of reconfiguration actions of the processes in \( P \), where \( C \equiv R \cup \bar{R} \). Let \( A \) be the set of actions of the processes in \( P \), where \( A \equiv I \cup C \). The LTS semantics of basic CCS\textsuperscript{dp} is defined in Table 1.
5. Strong of-Bisimulation

Strong of-bisimulation (\(\sim_{of}\)) is the largest symmetric binary relation on \(\mathcal{P}\) such that the following two conditions hold \(\forall (p, q) \in \sim_{of}\):

**Observation:** \(\forall \alpha \in \mathcal{I}_p \ \forall p' \in \mathcal{P} \ (p \xrightarrow{\alpha} p' \implies \alpha \in \mathcal{I}_{p'} \land \exists q' \in \mathcal{P} \ (q \xrightarrow{\alpha} q' \land (p', q') \in \sim_{of}))\)

**Fraction:** \(\forall \tau_{rx} \in \mathcal{R}_p \ \forall p'' \in \mathcal{P} \ (p \xrightarrow{\tau_{rx}} p'' \implies \tau_{rx} \in \mathcal{R}_{p'} \land \exists q'' \in \mathcal{P} \ (q \xrightarrow{\tau_{rx}} q'' \land (p'', q'') \in \sim_{of}))\)

6. Inductive Dependency between \(\xrightarrow{\mu}\) and \(\sim_{of}\)

Let \(sfdrdepth : \mathcal{P} \rightarrow \mathbb{N}\) such that \(sfdrdepth(p) \doteq \max\{sfdrdepth(s) \mid s \in \text{successors}(p)\}\) with

\[
\text{sfdrdepth} : \mathcal{P} \rightarrow \mathbb{N} \text{ such that } sfdrdepth(s) \doteq \left\{
\begin{array}{ll}
0 & \text{if } \mathcal{R}_s = \emptyset \\
1 + \max\{sfdrdepth(X) \mid \tau_{rx} \in \mathcal{R}_s\} & \text{else}
\end{array}
\right.
\]

We restrict \(\mathcal{P}\) to the domain of \(sfdrdepth\) in order to create an inductive dependency between transition relations and strong of-bisimulation, and thereby avoid a circular dependency.