Theoretical Informatics and Applications Will be set by the publisher
Informatique Théorique et Applications

A NOTE ON GLOBAL INDUCTION MECHANISMS IN A μ-CALCULUS WITH EXPLICIT APPROXIMATIONS *

Christoph Sprenger[1, 2] and Mads Dam[3]

Abstract. We investigate a Gentzen-style proof system for the first-order μ-calculus based on cyclic proofs, produced by unfolding fixed point formulas and detecting repeated proof goals. Our system uses explicit ordinal variables and approximations to support a simple semantic induction discharge condition which ensures the well-foundedness of inductive reasoning. As the main result of this paper we propose a new syntactic discharge condition based on traces and establish its equivalence with the semantical condition. We give an automata-theoretic reformulation of this condition which is more suitable for practical proofs. For a detailed comparison with previous work we consider two simpler syntactic conditions and show that they are stronger than our new condition.

1991 Mathematics Subject Classification. 68Q60, 03F07, 03B35.

1. Introduction

The first-order μ-calculus [8] provides a useful setting for semi-automatic program verification. It is expressive enough to encode, from the bottom up, a range of program logics (e.g. LTL, CTL, CTL*, Hoare Logic) as well as process calculi and programming languages including their data types and operational semantics. A framework based on this idea is described by Fredlund [5] and has been applied to a substantial part of a real concurrent programming language Erlang in the Erlang Verification Tool [1]. A key aspect in the design of such a framework is proof

* *The research of first author was supported by Swiss European Fellowship No. 83EU-065536. In the final phase of writing this article he was supported by an ERCIM fellowship at INRIA.*

[1] Swedish Institute of Computer Science, SICS, Box 1263, 164 29 Kista, Sweden; e-mail: sprenger@sics.se (until Aug, 2002)

[2] INRIA Sophia Antipolis, 2004, route des Lucioles, BP 93, 06902 Sophia Antipolis, France; e-mail: sprenger@sophia.inria.fr (current address)

[3] Dept. of Microelectronics and Information Technology, Royal Institute of Technology, KTH, Forum 105, 164 40 Kista, Sweden; e-mail: mfd@imit.kth.se

search, in particular the handling of fixed point formulas. The standard approach, Park's fixed point induction rule (*cf.* [6]), is not suitable for proof search in practice. An alternative is to admit cyclic proof structures (*cf.* [2,4,7,13]) and look for sound *induction discharge conditions*, external global criteria that ensure the well-foundedness of the inductive reasoning. In this setting, proof trees are completed into proof graphs by adding back edges (called repeats) from non-axiom leaves to nodes of which they are substitution instances. This type of proof is favourable to proof search as it allows one to delay decisions concerning induction strategies as long as possible.

In this paper, we study induction discharge conditions in the context of a Gentzen-style proof system for the first-order μ-calculus and present a new syntactic condition, which is weaker than previously published ones in the sense that it qualifies more proof structures as valid proofs. Our proof system is a variant of previous systems [4,5,10,11]. In particular, it shares with [4,5,10] the technique, first proposed for the modal μ-calculus by Dam and Gurov [4], of introducing explicit approximation ordinal variables and ordering constraints between them into the proof system. Discharge conditions then rely on these ordering constraints. In the presence of a Cut rule, the use of approximation ordinals considerably simplifies earlier treatments (*cf.* [3]). Dam and Gurov proposed a simple semantic discharge condition, which essentially requires that no infinite path in the proof structure can be assigned a coherent infinite sequence of valuations. This condition expresses in a natural way the requirement of well-foundedness of all inductive reasoning, but due to its semantical nature it is not suitable for the purpose of practical proof. We show that it is equivalent to our practically more useful syntactic condition, while previous syntactic conditions [4,5,10] turn out to be strictly stronger.

Our new condition relies on the notion of *traces*, which are non-increasing chains (w.r.t. the ordering constraints) of ordinal variables associated with a path of a proof structure. They can be seen as a uniform generalisation of the μ- and ν-traces described by Niwinski and Walukiewicz [7] to systems with a Cut rule and explicit approximants. We identify *progress* in a trace with positions where a strict decrease with respect to the constraints occurs. The equivalence with the semantical condition is then established by showing that a trace progressing at infinitely many positions implies well-foundedness on the semantical side and, conversely, the absence of such a trace gives rise to non-wellfoundedness. Based on the observation that every trace can be transformed into a normal trace where progress is made only at repeat nodes, we are able to give a compact automata-theoretic characterisation of our trace-based discharge condition in terms of an inclusion of the languages recognised by two Büchi automata. This formulation may serve as the basis of an implementation in a proof tool such as the Erlang Verification Tool. Being weaker than previously known conditions, the automata-based criterion might be able to detect proofs where the others fail to do so, which is an advantage for semi-automatic proof search.

For a detailed comparison of our new condition with previously published work, we then turn our attention to two simpler discharge conditions. Common to both of these is that they are based on progress and preservation properties of single ordinal variables at the repeats of the proof structure. The first condition requires that in each strongly connected subgraph of the proof structure there is a repeat progressing on some ordinal variable, while all other repeats preserve that variable. We show how this condition, which is similar to one proposed by Fredlund [5] (and [11], though in a somewhat different setting), corresponds to a simplified, but strictly stronger version of our automata-based condition. Secondly, we restrict our attention to the special case of *simple* proof structures, where repeats loop back to ancestral nodes (*i.e.* they point to a node on the path from the root to the discharged leaf) and introduce a new alternative notion of discharge where repeats are organised in a partial order, called *induction order*. Progress and preservation properties imposed on each repeat then depend on its position in this order. For simple proof structures this condition generalises the one originally proposed by Dam and Gurov [4] and is equivalent to both the previous one as well as to the condition presented by Schöpp [10]. While drawing its inspiration from the latter, it is more local in the sense that it avoids a quantification over strongly connected subgraphs.

The outline of the rest of the paper is as follows. The next section introduces the μ-calculus with explicit approximations. Section 3 presents the proof system, using the semantical induction discharge condition in the basic notion of proof. In Section 4 we first introduce our trace-based discharge condition and establish its equivalence with the semantical one. Based on the notion of normal traces we then propose an automata-theoretic characterisation of this condition. Restricted forms of syntactic discharge are discussed in Section 5 and compared to previous work. Section 6 concludes the paper with a discussion of the results and an outlook on future work.

2. μ-Calculus with Explicit Approximations

2.1. Fixed points

We first briefly recall some basic facts from fixed point theory. Suppose A is an arbitrary set. Let $\mathbf{2} = \{0, 1\}$ be the two-point lattice and let $\mathsf{Pred}(A) = \mathbf{2}^A$ be the lattice of predicates over A ordered pointwise.

Definition 2.1. Let $\Psi \colon \mathsf{Pred}(A) \to \mathsf{Pred}(A)$ be a monotone map on $\mathsf{Pred}(A)$. The *ordinal approximation* $\mu^\alpha \Psi$ of the fixed point $\mu\Psi$ is defined by

$$
\begin{array}{lcll}
\mu^0 \Psi & = & \lambda x.0 & \\
\mu^{\alpha+1} \Psi & = & \Psi(\mu^\alpha \Psi) & \\
\mu^\gamma \Psi & = & \bigvee_{\alpha<\gamma} \mu^\alpha \Psi & \text{for a limit ordinal } \gamma
\end{array}
$$

Theorem 2.2. *Let* $\Psi \colon \mathsf{Pred}(A) \to \mathsf{Pred}(A)$ *be a monotone map on* $\mathsf{Pred}(A)$. *Then*

(1) $\mu\Psi = \bigvee_\alpha \mu^\alpha \Psi$ *is the least fixed point of* Ψ *(Knaster-Tarski)*
(2) $\mu^\alpha \Psi = \bigvee_{\beta<\alpha} \Psi(\mu^\beta \Psi)$

2.2. Syntax

We assume countably infinite sets of *individual* variables $x, y, z, \ldots \in V_I$, *predicate* variables $X^n, Y^n, Z^n, \ldots \in V_P^n$ of each arity $n \geq 0$, and *ordinal* variables $\iota, \kappa, \lambda, \ldots \in V_O$. We write $v, v', \ldots$ for variables of any of the aforementioned type. Let $t, t', \ldots$ range over the terms of some first-order signature Σ. We write $\overline{t}$ for a vector $t_1, \ldots, t_n$ of terms, let $|\overline{t}|$ denote its length n and $\{\overline{t}\}$ the set $\{t_1, \ldots, t_n\}$.

Definition 2.3. The syntax of *μ-calculus formulas ϕ and n-ary abstractions Φ^n over signature* Σ is inductively defined by

$$\begin{array}{lll} \phi & ::= & t = t' \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \exists x.\phi \mid \exists\kappa.\phi \mid \exists\kappa' < \kappa.\phi \mid \Phi^n(\overline{t}) \\ \Phi^n & ::= & X^n \mid \mu X^n(\overline{x}).\phi \mid \mu^\kappa X^n(\overline{x}).\phi \end{array}$$

with the restriction that $|\overline{x}| = n$ in *fixed point abstractions* $\mu X^n(\overline{x}).\phi$ and *approximation abstractions* $\mu^\kappa X^n(\overline{x}).\phi$, and $|\overline{t}| = n$ in applications $\Phi^n(\overline{t})$. Furthermore, fixed point and approximation abstraction formation are subject to the usual formal monotonicity condition requiring that each occurrence of X^n in ϕ appears in the scope of an even number of negation symbols.

We will henceforth omit the arity annotations from predicate variables and assume that arities match as required by the previous definition. The sets of free variables of formulas and abstractions are defined as expected. In particular, we have

$$\begin{array}{lll} \mathrm{fv}(\exists\kappa' < \kappa.\phi) & = & (\mathrm{fv}(\phi) - \{\kappa'\}) \cup \{\kappa\} \\ \mathrm{fv}(\Phi(\overline{t})) & = & \mathrm{fv}(\Phi) \cup \mathrm{fv}(\overline{t}) \\ \mathrm{fv}(\mu X(\overline{x}).\phi) & = & \mathrm{fv}(\phi) - \{X, \overline{x}\} \\ \mathrm{fv}(\mu^\kappa X(\overline{x}).\phi) & = & (\mathrm{fv}(\phi) - \{X, \overline{x}\}) \cup \{\kappa\} \end{array}$$

We identify formulas and abstractions that differ only by a renaming of their bound variables. Dual connectives are defined from the primitive ones in the usual way. The greatest fixed point $\nu X(\overline{x}).\phi$ and the greatest fixed point approximation $\nu^\kappa X(\overline{x}).\phi$ are defined by

$$\begin{array}{lll} \nu X(\overline{x}).\phi & = & \neg\mu X(\overline{x}).\neg\phi[\neg X/X] \\ \nu^\kappa X(\overline{x}).\phi & = & \neg\mu^\kappa X(\overline{x}).\neg\phi[\neg X/X] \end{array}$$

We assume that substitutions $\sigma, \sigma', \ldots$ map term variables to terms, predicate variables to abstractions of the same arity and ordinal variables to ordinal variables. We write $\phi\sigma$ or $\sigma(\phi)$ to denote the formula obtained from ϕ by substituting each occurrence of a variable v by $\sigma(v)$, renaming bound variables as necessary to avoid capture of free variables.

2.3. Semantics

Let Σ be a first-order signature. A Σ-model $\mathcal{M} = (\mathcal{A}, \rho)$ consists of a Σ-structure $\mathcal{A}$ interpreting the symbols in Σ and an $\mathcal{A}$-environment ρ interpreting each variable in its respective domain. We write $|\mathcal{A}|$ for the support set of the structure $\mathcal{A}$. The semantics interprets a μ-calculus formula ϕ as an element $\|\phi\|_{\mathcal{M}} \in \mathbf{2}$ and a n-ary abstraction Φ as an element $\|\Phi\|_{\mathcal{M}} \in \mathsf{Pred}(|\mathcal{A}|^n)$. We usually drop $\mathcal{M}$ and write $\|\phi\|_\rho$ and $\|\Phi\|_\rho$ if the structure $\mathcal{A}$ is clear from the context. The semantics $\|t\|_\rho \in |\mathcal{A}|$ of a term t is defined as usual.

Definition 2.4. (Semantics) Given a signature Σ and a Σ-model $(\mathcal{A}, \rho)$ the semantics of μ-calculus formulas ϕ and abstractions Φ over Σ is inductively defined by

$$\begin{array}{lll}
\|t = t'\|_\rho & = & \text{if } \|t\|_\rho = \|t'\|_\rho \text{ then } 1 \text{ else } 0 \\
\|\neg\phi\|_\rho & = & 1 - \|\phi\|_\rho \\
\|\phi_1 \vee \phi_2\|_\rho & = & \max\{\|\phi_1\|_\rho, \|\phi_2\|_\rho\} \\
\|\exists x.\phi\|_\rho & = & \bigvee_{a \in |\mathcal{A}|} \|\phi\|_{\rho[a/x]} \\
\|\exists \kappa.\phi\|_\rho & = & \bigvee_{\beta} \|\phi\|_{\rho[\beta/\kappa]} \\
\|\exists \kappa' < \kappa.\phi\|_\rho & = & \bigvee_{\beta < \rho(\kappa)} \|\phi\|_{\rho[\beta/\kappa']} \\
\|\Phi(\bar{t})\|_\rho & = & \|\Phi\|_\rho(\|\bar{t}\|_\rho) \\
\|X\|_\rho & = & \rho(X) \\
\|\mu X(\overline{x}).\phi\|_\rho & = & \mu\Psi \\
\|\mu^\kappa X(\overline{x}).\phi\|_\rho & = & \mu^{\rho(\kappa)}\Psi
\end{array}$$

where $\Psi = \lambda P.\lambda\overline{a}.\|\phi\|_{\rho[P/X,\overline{a}/\overline{x}]}$ in the clauses for fixed point and approximation abstractions.

A model $\mathcal{M} = (\mathcal{A}, \rho)$ *satisfies* a formula ϕ, written $\mathcal{M} \models \phi$ if $\|\phi\|_\rho = 1$. The formula ϕ is called *valid*, written $\models \phi$, if it is satisfied in all Σ-models. Given a Σ-model $\mathcal{M} = (\mathcal{A}, \rho)$ we extend ρ a posteriori to terms t and formulas ϕ other than variables by defining $\rho(t) = \|t\|_\rho$ and $\rho(\phi) = \|\phi\|_\rho$. This allows us to compose substitutions σ and environments ρ as in $\rho \circ \sigma$.

3. The Proof System

The proof system we present in this section is an adaption to the first-order setting of the one proposed by Schöpp [10] for the modal μ-calculus. The main difference with the original proposal [4] is the addition of ordinal quantification. We would like to stress, however, that all results of this paper remain valid for systems without ordinal quantification as well as for modal variants.

3.1. Ordinal Constraints

Our proof system uses explicit ordinal approximations of fixed point formulas. Constraints between ordinal variables will be recorded in strict partial orders $\mathcal{O} = (|\mathcal{O}|, <_{\mathcal{O}})$, where $|\mathcal{O}|$ is a finite set of ordinal variables from V_O and $<_{\mathcal{O}}$ is a binary irreflexive, transitive relation on $|\mathcal{O}|$. We refer to $\mathcal{O}$ as a *set of ordinal constraints*. We write $\iota \leq_{\mathcal{O}} \kappa$ if either $\iota <_{\mathcal{O}} \kappa$ or $\iota = \kappa$ and $\kappa \in |\mathcal{O}|$. If $\mathcal{O}$ and $\mathcal{O}'$ are two strict partial orders we write $\mathcal{O} \subseteq \mathcal{O}'$ to mean that $|\mathcal{O}| \subseteq |\mathcal{O}'|$ and $<_{\mathcal{O}} \subseteq <_{\mathcal{O}'}$. In sequents, we will write $\mathcal{O}, \kappa$ for $(|\mathcal{O}| \cup \{\kappa\}, <_{\mathcal{O}})$ and $\mathcal{O}, \kappa' < \kappa$ for $(|\mathcal{O}| \cup \{\kappa', \kappa\}, <_{\mathcal{O}'})$, where $<_{\mathcal{O}'}$ is the transitive closure of $<_{O} \cup \{(\kappa', \kappa)\}$. Given a model $\mathcal{M} = (\mathcal{A}, \rho)$ we say that ρ *respects* $\mathcal{O}$, if $\rho(\iota) < \rho(\kappa)$ whenever $\iota <_{\mathcal{O}} \kappa$.

3.2. Sequents and Proof Rules

The *sequents* of the proof system are of the form $\Gamma \vdash_{\mathcal{O}} \Delta$, where Γ and Δ are finite multisets of formulas and $\mathcal{O} = (|\mathcal{O}|, <_{\mathcal{O}})$ is a set of ordinal constraints. A sequent is *well-formed* if all ordinal variables occurring free in Γ or Δ are elements of $|\mathcal{O}|$. We restrict our attention to well-formed sequents without further mentioning. The set of free variables of a sequent $\Gamma \vdash_{\mathcal{O}} \Delta$ is defined by $\mathrm{fv}(\Gamma \vdash_{\mathcal{O}} \Delta) = \mathrm{fv}(\Gamma \cup \Delta) \cup |\mathcal{O}|$. Given a Σ-model $\mathcal{M} = (\mathcal{A}, \rho)$ we say that $\mathcal{M}$ *satisfies* a sequent $\Gamma \vdash_{\mathcal{O}} \Delta$ if ρ respects $\mathcal{O}$ and $\mathcal{M} \models \phi$ for all $\phi \in \Gamma$ imply that $\mathcal{M} \models \psi$ for some $\psi \in \Delta$. A model $\mathcal{M}$ *falsifies* a sequent $\Gamma \vdash_{\mathcal{O}} \Delta$ if $\mathcal{M}$ does not satisfy $\Gamma \vdash_{\mathcal{O}} \Delta$. The sequent $\Gamma \vdash_{\mathcal{O}} \Delta$ is *valid* if it is satisfied in all models and *invalid* otherwise.

The rules of our proof system are displayed in Tables 1 and 2. They are presented in tableau style with the conclusion above the line and the premises below. Table 1 shows standard rules of first-order logic with equality. The fixed point rules of Table 2 are a direct application of Theorem 2.2. Note the asymmetry of the rules (μ_1-L) and (μ_0-R). It is not difficult to show, once the notion of proof has been introduced (see Definition 3.4), the derivability of the sequents

- $\Gamma, (\mu X(\overline{x}).\phi)(\overline{t}) \vdash_{\mathcal{O}} \phi[\mu X(\overline{x}).\phi/X, \overline{t}/\overline{x}], \Delta$, and
- $\Gamma, \exists\kappa.(\mu^{\kappa} X(\overline{x}).\phi)(\overline{t}) \vdash_{\mathcal{O}} (\mu X(\overline{x}).\phi)(\overline{t}), \Delta$

which is sufficient to derive the symmetric rules (μ_0-L) and (μ_1-R), respectively. However, taking (μ_1-R) as primitive appears not to be sufficient to derive (μ_0-R) without introducing further proof machinery, ordinal successors or existential ordinal variables. Rules ($\exists_O$-L) and ($\exists_O^<$-L) both introduce a fresh ordinal variable while the latter rule additionally generates a new ordinal constraint. Their right hand side versions, ($\exists_O$-R) and ($\exists_O^<$-R), respectively require an ordinal variable and a constraint as a witness. Rule (OrdStr) allows us to strengthen ordinal constraints in a controlled way. More precisely, we may add new ordinal variables and constraints to the order $\mathcal{O}$ of a sequent $\Gamma \vdash_{\mathcal{O}} \Delta$ as long as no new variable goes below a variable in $|\mathcal{O}|$ with respect to the extended order $<_{\mathcal{O}'}$. This rule is sometimes helpful to find repeats (see Definition 3.1 below) and seems to be required to prove some simple valid sequents such as $\Gamma, \exists\kappa.\phi \vdash_{\mathcal{O}} \exists\kappa.\exists\kappa' < \kappa.\phi[\kappa'/\kappa], \Delta$.

3.3. Pre-Proofs, Runs and Proofs

A *derivation tree* $\mathcal{D} = (\mathcal{N}, \mathcal{E}, \mathcal{L})$ is a tree-shaped graph $(\mathcal{N}, \mathcal{E})$ with nodes $\mathcal{N}$ and edges $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ together with a function $\mathcal{L}$ labeling each node of the tree with a sequent in a way that is consistent with the application of the proof rules. We will often write $N(\Gamma \vdash_{\mathcal{O}} \Delta)$ for $\mathcal{L}(N) = \Gamma \vdash_{\mathcal{O}} \Delta$. The proof structures of our system are essentially finite graphs, which are generated from a derivation tree by

Structural Rules

$$\text{(Id)}\ \frac{\Gamma, \phi \vdash_{\mathcal{O}} \phi, \Delta}{\cdot} \qquad \text{(Cut)}\ \frac{\Gamma \vdash_{\mathcal{O}} \Delta}{\Gamma, \phi \vdash_{\mathcal{O}} \Delta \quad \Gamma \vdash_{\mathcal{O}} \phi, \Delta}$$

$$\text{(W-L)}\ \frac{\Gamma, \phi \vdash_{\mathcal{O}} \Delta}{\Gamma \vdash_{\mathcal{O}} \Delta} \qquad \text{(W-R)}\ \frac{\Gamma \vdash_{\mathcal{O}} \phi, \Delta}{\Gamma \vdash_{\mathcal{O}} \Delta}$$

Logical and Equality Rules

$$(\neg\text{-L})\ \frac{\Gamma, \neg\phi \vdash_{\mathcal{O}} \Delta}{\Gamma \vdash_{\mathcal{O}} \phi, \Delta} \qquad (\neg\text{-R})\ \frac{\Gamma \vdash_{\mathcal{O}} \neg\phi, \Delta}{\Gamma, \phi \vdash_{\mathcal{O}} \Delta}$$

$$(\vee\text{-L})\ \frac{\Gamma, \phi_1 \vee \phi_2 \vdash_{\mathcal{O}} \Delta}{\Gamma, \phi_1 \vdash_{\mathcal{O}} \Delta \quad \Gamma, \phi_2 \vdash_{\mathcal{O}} \Delta} \qquad (\vee\text{-R})\ \frac{\Gamma \vdash_{\mathcal{O}} \phi_1 \vee \phi_2, \Delta}{\Gamma \vdash_{\mathcal{O}} \phi_1, \phi_2, \Delta}$$

$$(\exists_I\text{-L})\ \frac{\Gamma, \exists x.\phi \vdash_{\mathcal{O}} \Delta}{\Gamma, \phi \vdash_{\mathcal{O}} \Delta}\ x \notin \mathrm{fv}(\Gamma \cup \Delta) \qquad (\exists_I\text{-R})\ \frac{\Gamma \vdash_{\mathcal{O}} \exists x.\phi, \Delta}{\Gamma \vdash_{\mathcal{O}} \phi[t/x], \Delta}$$

$$(=\text{-L})\ \frac{\Gamma[t_2/x], t_1 = t_2 \vdash_{\mathcal{O}} \Delta[t_2/x]}{\Gamma[t_1/x] \vdash_{\mathcal{O}} \Delta[t_1/x]} \qquad (=\text{-R})\ \frac{\Gamma \vdash_{\mathcal{O}} t = t, \Delta}{\cdot}$$

Table 1. Proof Rules of First-Order Logic with Equality

adding a back edge from each non-axiom leaf to a node of which it is a repetition (up to some substitution). Let us fix an arbitrary derivation tree $\mathcal{D} = (\mathcal{N}, \mathcal{E}, \mathcal{L})$.

Definition 3.1. (Repeat) Let $M(\Gamma \vdash_{\mathcal{O}} \Delta)$ and $N(\Gamma' \vdash_{\mathcal{O}'} \Delta')$ be nodes of $\mathcal{D}$. Then the triple (M, N, σ) is called a *repeat* of $\mathcal{D}$, if N is a leaf of $\mathcal{D}$ and σ is a substitution such that

(1) $\phi \in \Gamma$ implies $\sigma(\phi) \in \Gamma'$, and
(2) $\psi \in \Delta$ implies $\sigma(\psi) \in \Delta'$, and
(3) $\kappa \in |\mathcal{O}|$ implies $\sigma(\kappa) \in |\mathcal{O}'|$, and
(4) $\iota <_{\mathcal{O}} \kappa$ implies $\sigma(\iota) <_{\mathcal{O}'} \sigma(\kappa)$.

The node N is called *repeat node* and M its *companion node*.

It is worth remarking that we do not require companions to be ancestors of their corresponding repeat nodes in $\mathcal{D}$.

Definition 3.2. (Pre-Proof) A *pre-proof* $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ for a sequent $\Gamma \vdash_{\mathcal{O}} \Delta$ consists of a derivation tree $\mathcal{D} = (\mathcal{N}, \mathcal{E}, \mathcal{L})$ with root node labeled by $\Gamma \vdash_{\mathcal{O}} \Delta$ and a set of repeats $\mathcal{R}$ for $\mathcal{D}$ such that each non-axiom leaf appears in exactly one

Fixed Point Rules

$$(\mu_1\text{-L})\ \frac{\Gamma, (\mu X(\overline{x}).\phi)(\overline{t}) \vdash_{\mathcal{O}} \Delta}{\Gamma, \exists\kappa.(\mu^{\kappa} X(\overline{x}).\phi)(\overline{t}) \vdash_{\mathcal{O}} \Delta} \qquad (\mu_0\text{-R})\ \frac{\Gamma \vdash_{\mathcal{O}} (\mu X(\overline{x}).\phi)(\overline{t}), \Delta}{\Gamma \vdash_{\mathcal{O}} \phi[\mu X(\overline{x}).\phi/X, \overline{t}/\overline{x}], \Delta}$$

$$(\mu^{\kappa}\text{-L})\ \frac{\Gamma, (\mu^{\kappa} X(\overline{x}).\phi)(\overline{t}) \vdash_{\mathcal{O}} \Delta}{\Gamma, \exists\kappa' < \kappa.\phi[\mu^{\kappa'} X(\overline{x}).\phi/X, \overline{t}/\overline{x}] \vdash_{\mathcal{O}} \Delta}$$

$$(\mu^{\kappa}\text{-R})\ \frac{\Gamma \vdash_{\mathcal{O}} (\mu^{\kappa} X(\overline{x}).\phi)(\overline{t}), \Delta}{\Gamma \vdash_{\mathcal{O}} \exists\kappa' < \kappa.\phi[\mu^{\kappa'} X(\overline{x}).\phi/X, \overline{t}/\overline{x}], \Delta}$$

Ordinal Rules

$$(\exists_O\text{-L})\ \frac{\Gamma, \exists\kappa.\phi \vdash_{\mathcal{O}} \Delta}{\Gamma, \phi \vdash_{\mathcal{O},\kappa} \Delta}\ \kappa \notin |\mathcal{O}| \qquad (\exists_O\text{-R})\ \frac{\Gamma \vdash_{\mathcal{O}} \exists\kappa.\phi, \Delta}{\Gamma \vdash_{\mathcal{O}} \phi[\iota/\kappa], \Delta}\ \iota \in |\mathcal{O}|$$

$$(\exists_O^<\text{-L})\ \frac{\Gamma, \exists\kappa' < \kappa.\phi \vdash_{\mathcal{O}} \Delta}{\Gamma, \phi \vdash_{\mathcal{O},\kappa'<\kappa} \Delta}\ \kappa' \notin |\mathcal{O}| \qquad (\exists_O^<\text{-R})\ \frac{\Gamma \vdash_{\mathcal{O}} \exists\kappa' < \kappa.\phi, \Delta}{\Gamma \vdash_{\mathcal{O}} \phi[\iota/\kappa'], \Delta}\ \iota <_{\mathcal{O}} \kappa$$

$$(\text{OrdStr})\ \frac{\Gamma \vdash_{\mathcal{O}} \Delta}{\Gamma \vdash_{\mathcal{O}'} \Delta}\ \mathcal{O} \subseteq \mathcal{O}' \text{ and } \iota <_{\mathcal{O}'} \kappa,\ \kappa \in |\mathcal{O}| \Rightarrow \iota \in |\mathcal{O}|$$

TABLE 2. Fixed Point and Ordinal Proof Rules

repeat of $\mathcal{R}$. The *pre-proof graph* of $\mathcal{P}$ is defined by $\mathcal{G}(\mathcal{P}) = (\mathcal{N}, \mathcal{E}', \mathcal{L})$, where $\mathcal{E}' = \mathcal{E} \cup \{(N, M) \mid \exists\sigma.(M, N, \sigma) \in \mathcal{R}\}$.

By a *path of* $\mathcal{P}$ we mean a path in $\mathcal{G}(\mathcal{P})$. A path π is called *rooted* if its first node $\pi(0)$ is the root of $\mathcal{D}$. We say that π *traverses a repeat* $R = (M, N, \sigma)$ if $\pi(i) = N$ and $\pi(i+1) = M$ for some position i.

Not every pre-proof is a proof of the validity of its root sequent. We now give a simple semantical *induction discharge condition* ensuring that all inductive reasoning embodied in a pre-proof is well-founded.

Definition 3.3. (Run) Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a pre-proof, $\mathcal{A}$ a Σ-structure and suppose $\Pi = (N_0, \rho_0) \cdots (N_i, \rho_i) \cdots$ is a (finite or infinite) sequence of pairs of nodes of $\mathcal{D}$ and $\mathcal{A}$-environments. Suppose that $N_i(\Gamma_i \vdash_{\mathcal{O}_i} \Delta_i)$. Then Π is called a *run of the pre-proof* $\mathcal{P}$ if

(1) N_0 is the root of $\mathcal{P}$, and
(2) ρ_i respects $\mathcal{O}_i$ for all i, and
(3) for all successive pairs (N_i, N_{i+1}) of nodes, either

(a) $(N_i, N_{i+1}) \in \mathcal{E}$ and ρ_{i+1} agrees with ρ_i on all free variables common to N_i and N_{i+1}, or
(b) $(N_{i+1}, N_i, \sigma) \in \mathcal{R}$ and $\rho_{i+1} = \rho_i \circ \sigma$.

Note that $\pi = N_0 \cdots N_i \cdots$ is a rooted path of $\mathcal{P}$. We say that the *run* Π *follows the path* π.

Definition 3.4. (Proof) A pre-proof $\mathcal{P}$ for $\Gamma \vdash_{\mathcal{O}} \Delta$ satisfies *discharge condition (rDC)* if all runs of $\mathcal{P}$ are finite, in which case $\mathcal{P}$ is called a *proof for* $\Gamma \vdash_{\mathcal{O}} \Delta$.

Intuitively, the finiteness of runs in a proof rests on the well-foundedness of the underlying interpretation of the ordinal variables and thus prevents unsound circular reasoning. This intuition will be made explicit in Section 4.

3.4. Soundness

Lemma 3.5. (Local soundness) *The proof rules of Tables 1 and 2 are sound. In particular, if there is a Σ-model $(\mathcal{A}, \rho)$ falsifying the conclusion C of a rule then there is an $\mathcal{A}$-environment ρ' such that $(\mathcal{A}, \rho')$ falsifies some premise P of that rule. Moreover, ρ and ρ' agree on all free variables common to C and P.*

Proof. By inspection of the proof rules. For the fixed point rules the claim follows immediately from Theorem 2.2. The cases of the ordinal rules are straightforward except for rule (OrdStr), which we discuss now. Suppose some model $(\mathcal{A}, \rho)$ falsifies the sequent $\Gamma \vdash_{\mathcal{O}} \Delta$ in the conclusion. The sequent in the premise is $\Gamma \vdash_{\mathcal{O}'} \Delta$ with $\mathcal{O} \subseteq \mathcal{O}'$. We define the environment ρ' inductively by

$$\rho'(v) = \begin{cases} 1 + \max\{\rho'(\iota) \mid \iota <_{\mathcal{O}'} \kappa\} & \text{if } v = \kappa \text{ and } \kappa \in |\mathcal{O}'| - |\mathcal{O}| \\ \rho(v) & \text{otherwise} \end{cases}$$

It is sufficient to check that ρ' respects $\mathcal{O}'$ and thus falsifies the premise sequent $\Gamma \vdash_{\mathcal{O}'} \Delta$. Consider some constraint $\iota <_{\mathcal{O}'} \kappa$. If $\kappa \in |\mathcal{O}|$ then also $\iota \in |\mathcal{O}|$ by the side condition of the rule. Hence, $\rho'(\iota) < \rho'(\kappa)$ is inherited from $\mathcal{O}$. If $\kappa \in |\mathcal{O}'| - |\mathcal{O}|$ then we have $\rho'(\iota) < 1 + \rho'(\iota) \leq \rho'(\kappa)$ by the definition of ρ'. □

Lemma 3.6. *Let (M, N, σ) be a repeat and let $(\mathcal{A}, \rho)$ be a model falsifying $\mathcal{L}(N)$. Then $(\mathcal{A}, \rho \circ \sigma)$ falsifies $\mathcal{L}(M)$.*

Proof. By the definition of a repeat. □

Theorem 3.7. (Soundness) *If there is a proof for $\Gamma \vdash_{\mathcal{O}} \Delta$, then $\Gamma \vdash_{\mathcal{O}} \Delta$ is valid.*

Proof. Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a proof for $\Gamma \vdash_{\mathcal{O}} \Delta$ and suppose for a contradiction that some model $(\mathcal{A}, \rho_0)$ falsifies $\Gamma \vdash_{\mathcal{O}} \Delta$. We iteratively construct an infinite run Π. The construction starts with $\Pi_0 = (N_0, \rho_0)$, where N_0 is the root node of $\mathcal{P}$. Clearly, ρ_0 respects $\mathcal{O}_0$ since $(\mathcal{A}, \rho_0)$ falsifies $\mathcal{L}(N_0)$. Hence, Π_0 is a finite run.

Assume that we have already constructed the finite run $\Pi_i = (N_0, \rho_0) \cdots (N_i, \rho_i)$ for some $i \geq 0$ such that $(\mathcal{A}, \rho_i)$ is a model falsifying $\mathcal{L}(N_i)$. Note that $\mathcal{L}(N_i)$ can not be an axiom. We show now that Π_i can be extended to a run $\Pi_{i+1} =$

$\Pi_i(N_{i+1}, \rho_{i+1})$ such that $(\mathcal{A}, \rho_{i+1})$ is a model falsifying $\mathcal{L}(N_{i+1}) = \Gamma_{i+1} \vdash_{\mathcal{O}_{i+1}} \Delta_{i+1}$ (and hence ρ_{i+1} respects $\mathcal{O}_{i+1}$). We distinguish two cases. If N_i is a non-leaf node then there is by Lemma 3.5 a successor node M of N_i and an $\mathcal{A}$-environment ρ such that $(\mathcal{A}, \rho)$ falsifies $\mathcal{L}(M)$ and ρ agrees with ρ_i on all free variables common to $\mathcal{L}(N_i)$ and $\mathcal{L}(M)$. Then the sequence Π_{i+1} obtained by setting $N_{i+1} = M$ and $\rho_{i+1} = \rho$ is again a run. If N_i is a repeat node then there is a repeat $(M, N_i, \sigma) \in \mathcal{R}$ and we set $N_{i+1} = M$ and $\rho_{i+1} = \rho_i \circ \sigma$. By Lemma 3.6 $(\mathcal{A}, \rho_{i+1})$ falsifies $L(N_{i+1})$, so Π_{i+1} is a run. The limit Π of the sequence $\Pi_0, \Pi_1, \ldots$ is an infinite run of $\mathcal{P}$, contradicting the assumption that $\mathcal{P}$ is a proof. $\square$

4. Syntactic Discharge Conditions

As condition (rDC) captures the well-foundedness of the reasoning in a pre-proof in a very natural way, it serves as our reference discharge condition. Due to its semantical nature it is, however, hardly usable in practical proofs and we therefore introduce two alternative, purely syntactical, discharge conditions and show that they characterise condition (rDC). For the remainder of this section we fix an arbitrary pre-proof $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ with $\mathcal{D} = (\mathcal{N}, \mathcal{E}, \mathcal{L})$.

4.1. Traces and Progress

A trace of $\mathcal{P}$ is a path of $\mathcal{P}$ labeled by ordinal constraints that are linked to yield a non-increasing chain of ordinal dependencies.

Definition 4.1. (Trace) Let $\tau = (N_0, (\kappa_0, \kappa_0')) \cdots (N_i, (\kappa_i, \kappa_i')) \cdots$ be a (finite or infinite) sequence of pairs consisting of a node of $\mathcal{D}$ and a pair of ordinal variables of V_O. Suppose that $N_i(\Gamma_i \vdash_{\mathcal{O}_i} \Delta_i)$. Then τ is a *trace* of $\mathcal{P}$ if

1. $\kappa_i' \leq_{\mathcal{O}_i} \kappa_i$ for all i, and
2. for all successive pairs (N_i, N_{i+1}) of nodes, either
   - (a) $(N_i, N_{i+1}) \in \mathcal{E}$ and $\kappa_i' = \kappa_{i+1}$, or
   - (b) $(N_{i+1}, N_i, \sigma) \in \mathcal{R}$ and $\kappa_i' = \sigma(\kappa_{i+1})$.

We say that the trace $\tau = (N_0, (\kappa_0, \kappa_0')) \cdots (N_i, (\kappa_i, \kappa_i')) \cdots$ follows the path $\pi = N_0 \cdots N_i \cdots$.

Definition 4.2. (Progress) Let $\tau = (N_0, (\kappa_0, \kappa_0')) \cdots (N_i, (\kappa_i, \kappa_i')) \cdots$ with $\mathcal{L}(N_i) = \Gamma_i \vdash_{\mathcal{O}_i} \Delta_i$. Then

- τ *progresses at position* i if $\kappa_i' <_{\mathcal{O}_i} \kappa_i$, and
- τ is *progressive* if there are infinitely many positions where τ progresses, and
- a path π in $\mathcal{G}(\mathcal{P})$ is *progressive* if there is a progressive trace τ following a suffix π^i of π.

Example 4.3. Figure 1 represents the trace

$$\tau = (N_0, (\delta, \varepsilon))(N_1, (\alpha, \beta))(N_2, (\beta, \gamma))(N_3, (\gamma, \gamma))(N_4, (\kappa, \kappa)),$$

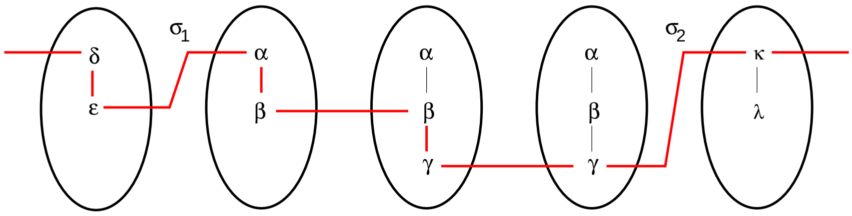

FIGURE 1. Example of a trace

where $R_1 = (N_0, N_1, \sigma_1)$ and $R_2 = (N_3, N_4, \sigma_2)$ are repeats with $\sigma_1(\alpha) = \epsilon$ and $\sigma_2(\kappa) = \gamma$. The trace follows the path $N_0 N_1 N_2 N_3 N_4$ and progresses at positions 0, 1 and 2.

4.2. The Trace-Based Discharge Condition

Definition 4.4. (tDC) A pre-proof $\mathcal{P}$ satisfies *discharge condition (tDC)* if all infinite paths of $\mathcal{P}$ are progressive.

Theorem 4.5. *A pre-proof $\mathcal{P}$ is a proof if and only if it satisfies condition (tDC).*

Proof. It is sufficient to show for any infinite rooted path π in $\mathcal{P}$ that there is no infinite run following π if and only if there is a progressive trace following some suffix of π. Let $\pi = N_0 \cdots N_i \cdots$ be an infinite path in $\mathcal{P}$ and let $\mathcal{O}_j$ denote the partial order appearing in sequent $\mathcal{L}(N_j)$.

"$\Rightarrow$" By contraposition. Suppose that there is no progressive trace following a suffix of π. We will construct an infinite run Π following π. Define the height $h_i(\kappa)$ of ordinal variable κ in the order $\mathcal{O}_i$ inductively as follows:

$$h_i(\kappa) = \begin{cases} 0 & \text{if } \kappa \text{ is minimal in } \mathcal{O}_i \\ 1 + \max\{h_i(\iota) \mid \iota <_{\mathcal{O}_i} \kappa\} & \text{otherwise} \end{cases}$$

We obviously have $h_i(\iota) < h_i(\kappa)$ whenever $\iota <_{\mathcal{O}_i} \kappa$. The height $h(\tau)$ of a non-progressive trace $\tau = (M_0, (\kappa_0, \kappa_0')) \cdots (M_i, (\kappa_i, \kappa_i')) \cdots$ is then defined by

$$h(\tau) = \sum_j (h_j(\kappa_j) - h_j(\kappa_j'))$$

Clearly, $h(\tau)$ is finite. Now we define $d(i, \kappa) = \max(H(i, \kappa))$, where

$$H(i, \kappa) = \{h(\tau) \mid \tau \text{ is a trace following } \pi^i \text{ with } \tau(0) = (N_i, (\kappa_i, \kappa_i')) \text{ and } \kappa_i = \kappa\}$$

In order to see that d is well-defined note that no non-progressive trace following π^i can have more than $m = n \cdot k^2$ progressing positions, where $n = |\mathcal{N}|$ is the number of nodes of $\mathcal{P}$ and k is the number of distinct ordinal variables in $\mathcal{P}$. Any trace τ with more than m progressing positions must repeat some pair $\tau(j) = \tau(k) = (N, (\kappa, \kappa'))$ with $j < k$, $N(\Gamma \vdash_{\mathcal{O}} \Delta)$ and $\kappa' <_{\mathcal{O}} \kappa$. This implies that

$\tau' = (\tau(j)\tau(j+1)\cdots\tau(k-1))^\omega$ is a progressive trace following π^{i+j}, contradicting our assumption that no such trace exists. Hence, $H(i,\kappa)$ is bounded by $m \cdot l$, where $l = \max\{h_j(\kappa) \mid j \geq i \text{ and } \kappa \in \mathcal{O}_j\}$, showing that d is well-defined.

Next, we show that d satisfies the following properties:

(i) $d(i,\kappa) < d(i,\lambda)$ whenever $\kappa <_{\mathcal{O}_i} \lambda$,
(ii) $d(i+1,\kappa) \leq d(i,\kappa)$ whenever $(N_i, N_{i+1}) \in \mathcal{E}$ and $\kappa \in |\mathcal{O}_i|$, and
(iii) $d(i+1,\kappa) \leq d(i,\sigma(\kappa))$ whenever $(N_{i+1}, N_i, \sigma) \in \mathcal{R}$ and $\kappa \in |\mathcal{O}_{i+1}|$.

To see (i) assume that $\tau = (N_i,(\kappa,\kappa'))\tau'$ is a trace following π^i such that $h(\tau) = d(i,\kappa)$. Then $\tau'' = (N_i,(\lambda,\kappa'))\tau'$ is a trace following π^i with $h(\tau'') > h(\tau)$, hence $d(i,\lambda) \geq h(\tau'') > d(i,\kappa)$. For (ii) suppose $(N_i, N_{i+1}) \in \mathcal{E}$ is an edge of $\mathcal{D}$, $\kappa \in |\mathcal{O}_i|$ and $\tau = (N_{i+1},(\kappa,\kappa'))\tau'$ is a trace following π^{i+1} such that $h(\tau) = d(i+1,\kappa)$. Then $\tau'' = (N_i,(\kappa,\kappa))\tau$ is a trace following π^i with $h(\tau'') = h(\tau)$, thus $d(i,\kappa) \geq h(\tau'') = d(i+1,\kappa)$. For (iii) suppose (N_{i+1}, N_i, σ) is a repeat in $\mathcal{R}$ and that $\tau = (N_{i+1},(\kappa,\kappa'))\tau'$ is a trace following π^{i+1} such that $h(\tau) = d(i+1,\kappa)$. Then $\tau'' = (N_i,(\sigma(\kappa),\sigma(\kappa)))\tau$ is a trace following π^i with $h(\tau'') = h(\tau)$, thus $d(i,\sigma(\kappa)) \geq h(\tau'') = d(i+1,\kappa)$.

We are now in a position to construct an infinite run $\Pi = (N_0,\rho_0)\cdots(N_i,\rho_i)\cdots$ following π. The construction will satisfy the invariants

(J1) $\rho_i(\kappa) \geq d(i,\kappa)$ for all $\kappa \in |\mathcal{O}_i|$
(J2) ρ_i respects $\mathcal{O}_i$

at each position $i \in \mathbb{N}$. We start by setting $\rho_0(\kappa) = d(0,\kappa)$ for each $\kappa \in |\mathcal{O}_0|$, which trivially satisfies (J1). ρ_0 also satisfies (J2) by (i) above.

Now suppose we have already constructed $(N_0,\rho_0)\cdots(N_i,\rho_i)$ such that (J1) and (J2) hold for i. We define ρ_{i+1} and show that it satisfies invariants (J1) and (J2). We distinguish two cases:

Case 1. (N_i, N_{i+1}) is an edge of $\mathcal{D}$. We proceed by a case analysis on the on the rule applied at N_i. Common to all cases is that we define $\rho_{i+1}(v) = \rho_i(v)$ for each $v \in \mathrm{fv}(\mathcal{L}(N_i))$ (and, in particular, for $\kappa \in |\mathcal{O}_i|$). This implies

(a) $\rho_{i+1}(\kappa) \geq d(i+1,\kappa)$ for all $\kappa \in |\mathcal{O}_i|$, and
(b) $\rho_{i+1}(\iota) < \rho_{i+1}(\kappa)$ whenever $\iota <_{\mathcal{O}_i} \kappa$

by the induction hypothesis and (ii). Since we have $\mathcal{O}_i \subseteq \mathcal{O}_{i+1}$, this establishes (J1) and (J2) for all rules except ($\exists_O$-L), ($\exists_O^<$-L) and (OrdStr). For the latter rules it remains to define ρ_{i+1} on any freshly introduced ordinal variables and to check invariants (J1) and (J2) for the additional cases concerning the fresh variables:

($\exists_O$-L): We set $\rho_{i+1}(\iota) = d(i+1,\iota)$, where ι is the fresh ordinal variable introduced by the rule. (J1) is satisfied by construction and (J2) is satisfied vacuously, since there are no cases involving ι.

($\exists_O^<$-L): We set $\rho_{i+1}(\iota) = d(i+1,\iota)$, where ι is the fresh ordinal variable introduced by the rule. (J1) is satisfied by construction. For (J2) let $\kappa \in |\mathcal{O}_i|$ such that $\iota <_{\mathcal{O}_{i+1}} \kappa$. By (i) we have $d(i+1,\iota) < d(i+1,\kappa)$ and by (a) $d(i+1,\kappa) \leq \rho_{i+1}(\kappa)$, so $\rho_{i+1}(\iota) < \rho_{i+1}(\kappa)$ as required.

(OrdStr): Let $m = \max\{\rho_i(\lambda) \mid \lambda \in |\mathcal{O}_i|\}$ and define $\rho_{i+1}(\kappa) = d(i+1,\kappa) + m + 1$ for $\kappa \in |\mathcal{O}_{i+1}| - |\mathcal{O}_i|$. Invariant (J1) is clearly satisfied by construction. It remains to verify (J2). Suppose that $\iota <_{\mathcal{O}_{i+1}} \kappa$. Since the side condition of the rule guarantees that $\iota \in |\mathcal{O}_i|$ whenever $\iota <_{\mathcal{O}_{i+1}} \kappa$ and $\kappa \in |\mathcal{O}_i|$, there are two cases not already covered by (a) and (b). In the first case, where $\iota, \kappa \in |\mathcal{O}_{i+1}| - |\mathcal{O}_i|$, the result follows from (i). For the second case, where $\iota \in |\mathcal{O}_i|$ and $\kappa \in |\mathcal{O}_{i+1}| - |\mathcal{O}_i|$, we have $\rho_{i+1}(\iota) \leq m < \rho_{i+1}(\kappa)$ by the definitions of m and ρ_{i+1}.

Case 2. (N_{i+1}, N_i, σ) is a repeat in $\mathcal{R}$. We set $\rho_{i+1} = \rho_i \circ \sigma$. By induction hypothesis we have $\rho_i(\sigma(\kappa)) \geq d(i, \sigma(\kappa))$ and by (iii) $d(i, \sigma(\kappa)) \geq d(i+1, \kappa)$, hence (J1) holds. For (J2) suppose that $\iota <_{\mathcal{O}_{i+1}} \kappa$. Then $\sigma(\iota) <_{\mathcal{O}_i} \sigma(\kappa)$ by the definition of a repeat, thus we get $\rho_{i+1}(\iota) = \rho_i(\sigma(\iota)) < \rho_i(\sigma(\kappa)) = \rho_{i+1}(\kappa)$ from the induction hypothesis.

Continuing this construction ad infinitum yields an infinite run Π following π.

"$\Leftarrow$" For the opposite direction suppose there is a progressive trace

$$\tau = (N_i, (\kappa_i, \kappa_i'))(N_{i+1}, (\kappa_{i+1}, \kappa_{i+1}')) \cdots$$

following the suffix π^i of π. For a contradiction suppose further that there is an infinite run $\Pi = (N_0, \rho_0) \cdots (N_i, \rho_i) \cdots$ following π. Let $j \geq i$. Then we have $\rho_j(\kappa_j) \geq \rho_j(\kappa_j')$, since ρ_j respects $\mathcal{O}_j$. We also have $\rho_j(\kappa_j') = \rho_{j+1}(\kappa_{j+1})$ by the definition of a run, since $\kappa_j' = \kappa_{j+1}$ whenever $(N_j, N_{j+1}) \in \mathcal{E}$ is a tree edge and $\kappa_j' = \sigma(\kappa_{j+1})$ whenever $(N_{j+1}, N_j, \sigma) \in \mathcal{R}$ is a repeat of $\mathcal{P}$. Thus, there is an infinite chain $\rho_i(\kappa_i) \geq \rho_{i+1}(\kappa_{i+1}) \geq \cdots$ of ordinals, which strictly decreases at infinitely many positions, since τ is progressive. This contradicts the well-foundedness of the ordinals. $\square$

4.3. Automata-Theoretic Characterisation

We now turn to an automata-theoretic reformulation of the previous discharge condition, which is more suitable for the practical application in a proof tool. Essentially, the automata-theoretic characterisation looks for the existence of specific traces, which progress only at repeat nodes.

Definition 4.6. (Normal trace) A trace $\tau = (N_0, (\kappa_0, \kappa_0')) \cdots (N_i, (\kappa_i, \kappa_i')) \cdots$ is called *normal* if, for all i, node N_i is a repeat node whenever τ progresses at position i. $\diamondsuit$

Example 4.7. Figure 2 shows the the trace

$$\tau = (N_0, (\delta, \varepsilon))(N_1, (\alpha, \beta))(N_2, (\beta, \gamma))(N_3, (\gamma, \gamma))(N_4, (\kappa, \kappa)),$$

of the previous Example 4.3 using continuous lines. Replacing the continuous line with the dashed lines between positions 1 and 3 yields the normal variant

$$\tau' = (N_0, (\delta, \varepsilon))(N_1, (\alpha, \alpha))(N_2, (\alpha, \alpha))(N_3, (\alpha, \gamma))(N_4, (\kappa, \kappa)),$$

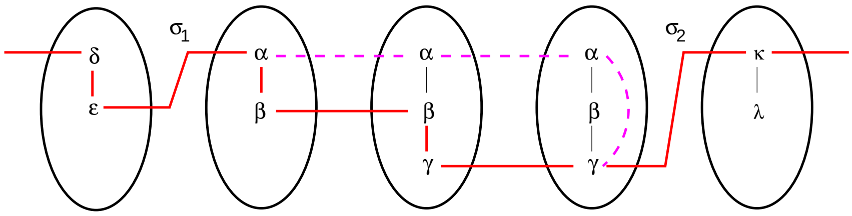

FIGURE 2. Example of a normal trace

of τ, where progress of τ at positions 1 and 2 is deferred to the repeat node at position 3 in τ'.

Lemma 4.8. *Every trace τ can be transformed into a normal trace τ' such that τ is progressive if and only if τ' is progressive.*

Proof. (Sketch) Since $\mathcal{O}_i \subseteq \mathcal{O}_{i+1}$ whenever $(N_i, N_{i+1}) \in \mathcal{E}$, progress can be deferred to repeat nodes in the manner suggested by Example 4.7. As progress is only deferred to the next repeat but never lost, progressiveness is preserved by this transformation. $\square$

By Lemma 4.8 we may without loss of generality for condition (tDC) restrict our attention to the *normal* traces of $\mathcal{P}$. Based on this observation we construct two Büchi automata, $\mathcal{B}_1$ and $\mathcal{B}_2$, over the alphabet $\mathcal{R}$ of repeats.

- **The path automaton $\mathcal{B}_1$:** recognises sequences of repeats that are traversed on paths of $\mathcal{P}$.
- **The progress automaton $\mathcal{B}_2$:** recognises sequences of repeats that are *potentially* connected through a normal trace; potentially, because this automaton tracks ordinal variable dependencies as in a normal trace, but completely ignores whether the sequence of repeats it accepts may be traversed by some path of $\mathcal{P}$.

The language inclusion $\mathcal{L}(\mathcal{B}_1) \subseteq \mathcal{L}(\mathcal{B}_2)$ then holds precisely if there is a normal trace along each infinite path of $\mathcal{P}$.

As a preparation for the formal definition of these two automata, let the relation $\rightarrow \subseteq \mathcal{R} \times \mathcal{R}$ on the repeats of $\mathcal{P}$ be defined by $R \rightarrow R'$ if there is a path in the derivation tree $\mathcal{D}$ of $\mathcal{P}$ from the companion node of R to the repeat node of R'. Moreover, let $V_o = \bigcup\{|\mathcal{O}| \mid N(\Gamma \vdash_{\mathcal{O}} \Delta) \in \mathcal{N}\}$ be the set of ordinal variables occurring free in $\mathcal{P}$.

Definition 4.9. The *path automaton* of $\mathcal{P}$ is the Büchi automaton

$$\mathcal{B}_1 = (\mathcal{R}, Q_1, Q_1^0, \delta_1, F_1)$$

where $Q_1 = Q_1^0 = F_1 = \mathcal{R}$ and the transition relation $\delta_1 \subseteq Q_1 \times \mathcal{R}_1 \times Q_1$ is defined by $\delta_1 = \{(R, R, R') \mid R \rightarrow R'\}$.

Definition 4.10. The *progress automaton* of $\mathcal{P}$ is the Büchi automaton

$$\mathcal{B}_2 = (\mathcal{R}, Q_2, Q_2^0, \delta_2, F_2)$$

where $Q_2 = Q_2^0 = (V_o \times \mathbf{2} \times V_o) \cup \{\Diamond\}$, $F_2 = V_o \times \{1\} \times V_o$ and $\delta_2 \subseteq Q_2 \times \mathcal{R} \times Q_2$ is defined by $\delta_2 = \delta_2' \cup (\{\Diamond\} \times \mathcal{R} \times Q_2)$ with

$$\delta_2' = \{((\iota, j, \kappa), (M, N, \sigma), (\kappa, k, \lambda)) \mid \sigma(\kappa) \leq_{\mathcal{O}_N} \iota \text{ and } j = 0 \Leftrightarrow \sigma(\kappa) = \iota\}$$

Note the presence of the state $\Diamond$ and the transitions from this state to any other state (including itself). Its role is to ensure that the language accepted by $\mathcal{B}_2$ is closed under prefixing with finite words over $\mathcal{R}$, reflecting the requirement in condition (tDC) that each infinite rooted path π has a trace following some *suffix* of π. Let us now illustrate these definitions with an example.

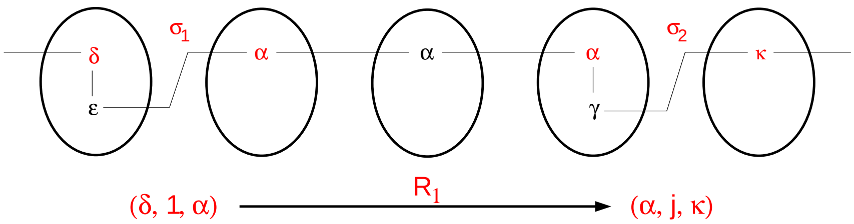

FIGURE 3. Example transition of automaton $\mathcal{B}_2$

Example 4.11. The upper part of Figure 3 shows a simplified version of the normal trace from Example 4.7. Since $\sigma_1(\alpha) <_{\mathcal{O}_0} \delta$ this trace gives rise to a transition $(\delta, 1, \alpha) \xrightarrow{R_1} (\alpha, j, \kappa)$ of automaton $\mathcal{B}_2$ for each $j \in \mathbf{2}$.

Definition 4.12. (aDC) A pre-proof $\mathcal{P}$ satisfies condition (aDC) if $L(\mathcal{B}_1) \subseteq L(\mathcal{B}_2)$.

Theorem 4.13. *A pre-proof $\mathcal{P}$ satisfies condition (tDC) if and only if $\mathcal{P}$ satisfies condition (aDC). The latter condition can be checked in time $2^{\mathcal{O}(n^2 \log n)}$.*

Proof. Let r_π be the sequence of repeats traversed by a path π. By Lemma 4.8 and since

$$L(\mathcal{B}_1) = \{r_\pi \mid \pi \text{ an infinite path of } \mathcal{P}\}$$

it is sufficient to show for all infinite paths π that there is a progressive normal trace τ following a suffix of π if and only if there is an accepting run r of $\mathcal{B}_2$ on r_π. Let therefore $\pi = N_0 N_1 \cdots$ be an infinite path of $\mathcal{P}$ with $N_i(\Gamma_i \vdash_{\mathcal{O}_i} \Delta_i)$.

"$\Rightarrow$" Consider a progressive normal trace

$$\tau = (N_k, (\kappa_k, \kappa_k'))(N_{k+1}, (\kappa_{k+1}, \kappa_{k+1}') \cdots \quad (1)$$

following π^k for some $k \geq 0$. Let $i_0, i_1, \ldots$ be the positions where a repeat node appears on τ and let $R_j = (N_{i_j+1}, N_{i_j}, \sigma_j)$ for $j \geq 0$ be the corresponding repeats. We construct the infinite sequence

$$r = \Diamond^p(\lambda_0, k_0, \lambda_1)(\lambda_1, k_1, \lambda_2)\cdots \tag{2}$$

where p is the number of repeat nodes appearing before position k on π and $\lambda_j = \kappa_{i_j}$ for $j \geq 0$. Since τ is a trace, we have $\lambda_j = \kappa_{i_j} \geq_{\mathcal{O}_{i_j}} \sigma_j(\kappa_{i_j+1})$. We also have $\kappa_{i_j+1} = \kappa_{i_{j+1}} = \lambda_{j+1}$, because τ is normal. We set $k_j = 0$ if $\sigma(\lambda_{j+1}) = \lambda_j$ and $k_j = 1$ otherwise. It is then not difficult to see that r is a run of $\mathcal{B}_2$ on r_π, which is accepting since τ is progressive.

"$\Leftarrow$" Suppose r is an accepting run of $\mathcal{B}_2$ on r_π of the form (2) above, let $r_\pi^p = R_0 R_1 \cdots$ and let i_j be the position of repeat $R_j = (N_{i_j+1}, N_{i_j}, \sigma_j)$ on π for each $j \geq 0$. We construct an infinite sequence τ of the shape 1 above by setting $k = i_0$ and

$$\begin{array}{lcll} (\kappa_{i_j}, \kappa'_{i_j}) & = & (\lambda_j, \sigma_j(\lambda_{j+1})) & \\ (\kappa_l, \kappa'_l) & = & (\lambda_{j+1}, \lambda_{j+1}) & \quad \text{for } i_j + 1 \leq l \leq i_{j+1} - 1 \end{array}$$

for $j \geq 0$. By the definition of $\mathcal{B}_2$ we know that $\sigma_j(\lambda_{j+1}) \leq_{\mathcal{O}_{i_j}} \lambda_j$ for all $j \geq 0$. Because $\mathcal{O}_l \subseteq \mathcal{O}_{l+1}$ whenever (N_l, N_{l+1}) is a tree edge, it is then easy to see that τ is a normal trace following π^k, which is progressive since r is accepting.

It remains to justify the complexity claim. The standard way to check the inclusion $L(\mathcal{B}_1) \subseteq L(\mathcal{B}_2)$ is to complement $\mathcal{B}_2$ into $\overline{\mathcal{B}_2}$ and check the product $\mathcal{B}_1 \times \overline{\mathcal{B}_2}$ for emptiness. For a pre-proof with $n = |\mathcal{N}|$ nodes we can have at most $n/2$ repeats and n different ordinal variables. This gives us $\mathcal{O}(n)$ states for $\mathcal{B}_1$ and $\mathcal{O}(n^2)$ states for $\mathcal{B}_2$. Complementing a Büchi automaton with m states can be done in time $2^{\mathcal{O}(m \log m)}$ [9]. Hence, the complementation of $\mathcal{B}_2$ takes time $2^{\mathcal{O}(n^2 \log n)}$, which does not increase by computing the product with $\mathcal{B}_1$ and the subsequent linear time emptiness check. $\square$

5. Restricted Forms of Syntactic Discharge

In this section we present two more restrictive syntactic discharge conditions and put them in relation with our new conditions as well as with those proposed in the literature. Let us consider an arbitrary but fixed pre-proof $\mathcal{P} = (\mathcal{D}, \mathcal{R})$.

5.1. Discharge Based on Strongly Connected Sets of Repeats

Definition 5.1. Let $R = (M, N, \sigma)$ be a repeat such that $M(\Gamma' \vdash_{\mathcal{O}'} \Delta')$ and $N(\Gamma \vdash_{\mathcal{O}} \Delta)$, and let $\kappa \in |\mathcal{O}'|$ be an ordinal variable. Then we say

(1) *R preserves κ* if $\sigma(\kappa) \leq_{\mathcal{O}} \kappa$, and
(2) *R progresses on κ* if $\sigma(\kappa) <_{\mathcal{O}} \kappa$.

We call a set of repeats $S \subseteq \mathcal{R}$ a *strongly connected subset (SCS)* of $\mathcal{R}$ if the subgraph $(S, \to \cap S \times S)$ of $(\mathcal{R}, \to)$ induced by S is strongly connected. Note that this is equivalent to saying that there is path π which traverses exactly the repeats in S infinitely often.

Definition 5.2. (scDC) A pre-proof $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ satisfies *condition (scDC)* if for each strongly connected subset $S \subseteq \mathcal{R}$ there an ordinal variable κ such that

(1) some repeat $R \in S$ progresses on κ, and
(2) each repeat $R' \in S$ preserves κ.

This condition is similar to the one described by Fredlund [5]. Simpson and Schöpp [11] use essentially the same condition as well, although their proof system is based on a different notion of approximation without ordinal variables. For a comparison with our previous conditions, we define $\mathcal{B}_2^- = (\mathcal{R}, Q_2^-, Q_2^0 \cap Q_2^-, \delta_2 \cap Q_2^- \times \mathcal{R} \times Q_2^-, F_2 \cap Q_2^-)$ to be the automaton obtained from $\mathcal{B}_2 = (\mathcal{R}, Q_2, Q_2^0, \delta_2, F_2)$ by restricting the sets of states and transitions to the set $Q_2^- = \{(\iota, j, \kappa) \in \mathcal{Q}_2 \mid \iota = \kappa\}$. *Condition (aDC-)* then requires that $\mathcal{L}(\mathcal{B}_1) \subseteq \mathcal{L}(\mathcal{B}_2^-)$.

Proposition 5.3. *A pre-proof $\mathcal{P}$ satisfies condition (scDC) if and only if it satisfies condition (aDC-). The latter condition can be checked in time $2^{\mathcal{O}(n \log n)}$, where $n = |\mathcal{N}|$ is the number of nodes of $\mathcal{P}$.*

Proof. "$\Rightarrow$" Suppose $\mathcal{P}$ satisfies (scDC) and let $r_1 = R_0 \cdots R_i \cdots$ be an accepting run of $\mathcal{B}_1$ on r_1. We show that r_1 is accepted by $\mathcal{B}_2$. Let S be the set of repeats occurring infinitely often in r_2. Since S is strongly connected, there is an ordinal variable κ such that some $R \in S$ progresses on κ and all $R' \in S$ preserve κ. Since $\mathcal{G}(\mathcal{P})$ is finite there is a position k such that all R_j with $j \geq k$ belong to S. Define $r_2 = \Diamond^k(\kappa, i_k, \kappa)(\kappa, i_{k+1}, \kappa) \cdots$, where, for each $j \geq k$, we set $i_j = 1$ if R_j progresses on κ and $i_j = 0$ otherwise. Then r_2 is a run of $\mathcal{B}_2^-$ on r_1, which is accepting, since there are infinitely many $j \geq k$ such that $R_j = R$. Thus, $\mathcal{P}$ satisfies condition (aDC-). "$\Leftarrow$" Suppose $\mathcal{P}$ satisfies (aDC-) and let $S \subseteq \mathcal{R}$ be strongly connected. Then there is an accepting run $r_1 = R_0 R_1 \cdots$ of $\mathcal{B}_1$ on r_1 such that $\{R_i \mid i \geq 0\} = S$. By (aDC-) there is an accepting run $r_2 = (\kappa, i_0, \kappa)(\kappa, i_1, \kappa) \cdots$ of $\mathcal{B}_2^-$ on r_1, implying that condition (scDC) holds for S and ordinal variable κ.

Automaton $\mathcal{B}_2'$ has $\mathcal{O}(n)$ states, where $n = |\mathcal{N}|$ is the number of nodes of $\mathcal{P}$. Thus, the time complexity of checking $\mathcal{L}(\mathcal{B}_1) \subseteq \mathcal{L}(\mathcal{B}_2^-)$ is $2^{\mathcal{O}(n \log n)}$ by an argument similar to the one in the proof of Theorem 4.13. □

Corollary 5.4. *If a pre-proof $\mathcal{P}$ satisfies (scDC) then it satisfies condition (tDC).*

Proof. By Theorem 4.13, since $\mathcal{L}(\mathcal{B}_2^-) \subseteq \mathcal{L}(\mathcal{B}_2)$. □

The following example shows that the converse of Corollary 5.4 does not hold in general.

Example 5.5. Let $\phi = \mu X(x).\exists z.X(z)$. The derivation in Figure 4 shows a pre-proof $\mathcal{P}$ for the sequent $\phi(x), \phi(y) \vdash$. We write ϕ^κ for $\mu^\kappa X(x).\exists z.X(z)$. We have named the nodes for reference and omitted some intermediate nodes for a more compact presentation.

$$\dfrac{\dfrac{\dfrac{\dfrac{N_0(\phi(x), \phi(y) \vdash)}{N_1(\phi^{\iota}(x), \phi^{\kappa}(y) \vdash_{\iota,\kappa})}\,(\mu_1\text{-L}, \exists_O\text{-L})}{N_2(\exists x'.\phi^{\iota'}(x'), \phi^{\kappa}(y) \vdash_{\iota'<\iota,\kappa})}\,(\mu^{\kappa}\text{-L}, \exists_O^{<}\text{-L})}{N_3(\exists x'.\phi^{\iota'}(x'), \exists y'.\phi^{\kappa'}(y') \vdash_{\iota'<\iota,\kappa'<\kappa})}\,(\mu^{\kappa}\text{-L}, \exists_O^{<}\text{-L})}{N_4(\phi^{\iota'}(x'), \exists y'.\phi^{\kappa'}(y') \vdash_{\iota'<\iota,\kappa'<\kappa})}\,(\exists_I\text{-L})$$

FIGURE 4. Pre-proof distinguishing (tDC) from (scDC)

This pre-proof has one repeat $R = (N_2, N_4, \sigma)$ with $\sigma = [x'/y, \kappa'/\iota', \kappa/\iota, \iota'/\kappa]$. Recall that we identify formulas up to renaming of bound variables. There is a infinite normal trace τ^{ω}, where

$$\tau = (N_2, (\iota, \iota))(N_3, (\iota, \iota))(N_4, (\iota, \iota'))(N_2, (\kappa, \kappa))(N_3, (\kappa, \kappa))(N_4, (\kappa, \kappa)),$$

following the the suffix $(N_2 N_3 N_4)^{\omega}$ of the only infinite path of $\mathcal{P}$. The trace τ is progressive, since $\iota' <_{\mathcal{O}_4} \iota$ at node N_4. Hence, $\mathcal{P}$ satisfies condition (tDC) and is thus a proof for $\phi(x), \phi(y) \vdash$. On the other hand, repeat R does not preserve any ordinal variable according to Definition 5.1. Hence, $\mathcal{P}$ fails to satisfy condition (scDC).

5.2. Discharge Using Induction Orders

We introduce an alternative discharge condition based on ordering the repeats of a pre-proof. Here, we restrict our attention to *simple* pre-proofs $\mathcal{P} = (\mathcal{D}, \mathcal{R})$, where for each repeat $(M, N, \sigma) \in \mathcal{R}$ there is a path from M to N in $\mathcal{D}$.

Definition 5.6. Let $R = (M, N, \sigma)$ and $R' = (M', N', \sigma')$ be two repeats in $\mathcal{R}$. The *structural dependency order* $\leq_{\mathcal{P}}$ on repeats is defined by $R \leq_{\mathcal{P}} R'$ if the companion M of R lies on the path $\pi_{R'} = M' \cdots N'$ from the companion M' to the repeat node N' of R'.

An induction order partially orders the repeats of a pre-proof. Repeats are required to be comparable under certain conditions.

Definition 5.7. (Induction Orders) A partial order $(\mathcal{R}, \preceq)$ on the set of repeats is called an *induction order* for $\mathcal{P}$, if $R \preceq R'$ or $R' \preceq R$ whenever

(1) $R'' \preceq R$ and $R'' \preceq R'$ for some R'' ($\preceq$ is *forest-like*), or
(2) $R \leq_{\mathcal{P}} R'$ or $R' \leq_{\mathcal{P}} R$ (*$\preceq$ respects $\leq_{\mathcal{P}}$*)

A *labeled induction order* $(\mathcal{R}, \preceq, \delta)$ is an induction order $(\mathcal{R}, \preceq)$ together with a map δ assigning an ordinal variable δ_R to each repeat $R \in \mathcal{R}$.

Definition 5.8. (ioDC) Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a simple pre-proof. We say that a labeled induction order $(\mathcal{R}, \preceq, \delta)$ *discharges* $\mathcal{P}$ if for all $R \in \mathcal{R}$

(1) R progresses on δ_R, and
(2) R preserves $\delta_{R'}$ whenever $R \preceq R'$.

Pre-proof $\mathcal{P}$ satisfies *condition (ioDC)* if there is a labeled induction order discharging $\mathcal{P}$.

Since any partial order that linearly orders the repeats of each strongly connected component of $\mathcal{P}$ is an induction order, condition (ioDC) subsumes the original discharge condition (DC) proposed by Dam and Gurov [4]. For forest-like induction orders condition (ioDC) is equivalent to the condition given by Schöpp [10] as the following lemma will show. However, by relying on the structural dependency order our new definition of induction order is more local in the sense that it avoids the quantification over all SCS of a pre-proof.

Lemma 5.9. *A forest-like partial order $(\mathcal{R}, \preceq)$ respects $\leq_{\mathcal{P}}$ if and only if each SCS $S \subseteq \mathcal{R}$ has a $\preceq$-greatest element.*

Proof. (Sketch) "$\Rightarrow$" Suppose that $(\mathcal{R}, \preceq)$ respects $\leq_{\mathcal{P}}$. Let $S \subseteq \mathcal{R}$ be a SCS. In order to see that S has a $\preceq$-greatest element, it is sufficient to show that any two $R, R' \in S$ have an upper bound in S, that is, $R \preceq \hat{R}$ and $R' \preceq \hat{R}$ for some $\hat{R} \in S$. Suppose $R, R' \in S$. Using the assumption that $\mathcal{P}$ is simple and S is strongly connected, one can prove that there is a sequence $R_0 R_1 \cdots R_n$ of repeats in S such that $R = R_0$, $R' = R_n$ and $R_i \leq_{\mathcal{P}} R_{i+1}$ or $R_{i+1} \leq_{\mathcal{P}} R_i$ for $0 \leq i < n$. As $(\mathcal{R}, \preceq)$ respects $\leq_{\mathcal{P}}$ we also have $R_i \preceq R_{i+1}$ or $R_{i+1} \preceq R_i$ for $0 \leq i < n$. Using the fact that $(\mathcal{R}, \preceq)$ is forest-like a routine induction on n shows that there is a least upper bound $\hat{R}$ of R and R' in S.

"$\Leftarrow$" Suppose that each SCS $S \subseteq \mathcal{R}$ has a $\preceq$-greatest element and let R and R' be two repeat such that $R \leq_{\mathcal{P}} R'$ holds. It follows immediately that $R \preceq R'$ or $R' \preceq R$, since $S = \{R, R'\}$ is an SCS. $\square$

Theorem 5.10. *Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a* simple *pre-proof. Then $\mathcal{P}$ satisfies condition (ioDC) if and only if it satisfies condition (scDC).*

Proof. "$\Rightarrow$" Suppose $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ satisfies (ioDC) witnessing the labeled induction order $(\mathcal{R}, \preceq, \delta)$ and let $S \subseteq \mathcal{R}$ be a SCS of $\mathcal{P}$. Then S has a $\preceq$-greatest element R by Lemma 5.9. By the definition of discharge R progresses on δ_R and all $R' \in S$ preserve δ_R. Hence, $\mathcal{P}$ satisfies condition (scDC).

"$\Leftarrow$" Suppose $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ satisfies (scDC). We iteratively construct a labeled induction order $(\mathcal{R}, \preceq, \delta)$ as follows. We start with the set $\mathcal{S}_0$ partitioning $\mathcal{R}$ into its strongly connected components. At step i we pick $S_i \in \mathcal{S}_i$ and then an $R_i \in S_i$ such that R_i progresses on some ordinal variable κ_i and all $R \in S_i$ preserve κ_i. Since $\mathcal{P}$ satisfies (scDC), such R_i and κ_i exist. Then we set $\mathcal{S}_{i+1} = (\mathcal{S}_i - \{S_i\}) \cup D$, where D is obtained as the partitioning of $S_i - \{R_i\}$ into its strongly connected components. This process terminates after $n = |\mathcal{R}|$ iterations, since at each step i repeat R_i is removed from $\bigcup \mathcal{S}_i$.

We define δ by $\delta(R_i) = \kappa_i$ and $R_i \preceq R_j$ if $S_i \subseteq S_j$. By the hierarchical nature of the construction $(\mathcal{R}, \preceq)$ is certainly a forest-like partial order and, moreover, for each SCS S of $\mathcal{R}$ there is a unique S_k such that $R_k \in S \subseteq S_k$. This implies that $S_i \subseteq S_k$ for any $R_i \in S$ and thus R_k is $\preceq$-greatest in S. Hence, $(\mathcal{R}, \preceq)$ is an induction order for $\mathcal{P}$ by Lemma 5.9. Note that any repeat R progresses on δ_R by construction. Also, for any R_i with $R_i \preceq R_j$, we have $R_i \in S_i \subseteq S_j$, so R_i preserves δ_{R_j}. Thus, $(\mathcal{R}, \preceq, \delta)$ discharges $\mathcal{P}$. □

The constructive proof of the previous theorem suggests an alternative algorithm to check for discharge on simple pre-proofs: it succeeds precisely if the algorithm is able to construct a labeled induction order (which discharges the pre-proof by construction). The decomposition of a graph into its strongly connected components being linear in the number of its nodes, the recursive decomposition performed by this algorithm has complexity $2^{\mathcal{O}(n \log n)}$, where $n = |\mathcal{N}|$, which matches the complexity of checking condition (aDC-).

6. Conclusions

We have studied a Gentzen-style proof system for the μ-calculus which is based on circular proofs. In particular, we have investigated several discharge conditions which externally justify the well-foundedness of inductive reasoning embodied in these proofs. Starting from the natural semantical condition proposed by Dam and Gurov [4], we have, based on the notion of traces, given a purely syntactical condition which characterises the semantical one for any given pre-proof. The syntactical condition is more suitable for practical proof than the semantical one, especially in its automata-theoretical formulation.

Next, for a detailed comparison with previously known discharge conditions, we have focused our attention on two simpler discharge criteria. In particular, we have considered two levels of restrictions with respect to our general condition:

(1) Restrict to normal traces that track the behaviour of a single ordinal variables, disallowing the renaming of ordinal variables at repeats; this leads to condition (scDC), similar to those in [5, 11], requiring that we find, for each strongly connected subgraph, an ordinal variable and a repeat that progresses on this variable while the other repeats preserve it.

(2) Additionally restrict the form of pre-proofs to simple ones, where each repeat node is reachable in the proof tree from its companion node; this allowed us to organise the repeats of a pre-proof into a partial order, called induction order, and formulate a new condition, called (ioDC), which imposes progress and preservation conditions on each repeat according to its position in the induction order. This condition is close to those in [4, 10], but avoids a quantification over strongly connected subsets of repeats.

Our comparison showed that condition (scDC) and (ioDC) are equivalent for simple pre-proofs and condition (scDC) is in general strictly stronger than our trace-based condition on a *fixed* pre-proof. However, an important open question concerns the proof-theoretical strength of the different conditions. More precisely, it is currently unclear to us whether there are sequents that can be proved using the trace-based condition, but for which no proof exists if we restrict ourselves to using a simpler discharge criterion. We are inclined to think that this is not the case.

But, while it might not be too difficult to show that any pre-proof can be unfolded into a simple one (with an exponential blow-up due to the loss of any sharing), the proof that we can dispense with renaming of ordinal variables at repeats seems more involved.

We conclude with some remarks on the practical application of our results. First, the exponential complexity of checking a discharge condition might look rather discouraging. We think, however, that by performing the checks for each individual strongly connected component in isolation this problem is often considerably alleviated in practice. Second, although the general trace-based condition is weaker and thus qualifies more pre-proofs as proofs, it is unclear whether this difference frequently shows up in practice. Some experimentation is required to determine whether the gain in generality is worth the additional cost. Third, in a tool implementation, the ability to check discharge conditions incrementally seems desirable in order to detect failure to discharge as soon as possible. Although the automata-based conditions can be used to this effect on the partially constructed proof structure, the need to complement the second automaton each time a new repeat is added does not support incremental checking very well. The algorithm suggested by the proof of equivalence of conditions (scDC) and (ioDC) on simple pre-proofs (Theorem 5.10) might be easier to adapt for this purpose.

Finally, we would like to remark that this work is part of a larger programme aiming at clarifying the relation between the global, lazy induction mechanism studied here and an eager induction discipline as implemented in a local proof rule for well-founded induction on the ordinals [12].

We would like to thank Dilian Gurov, Lars-åke Fredlund and Alex Simpson for interesting and fruitful discussions on the subject of this paper.

References

[1] T. Arts, M. Dam, L.-å. Fredlund, and D. Gurov. System description: Verification of distributed Erlang programs. In *Proc. CADE'98*, volume 1421 of *Lecture Notes in Artificial Intelligence*, pages 38–41, 1998.

[2] Julian Bradfield and Colin Stirling. Local model checking for infinite state spaces. *Theoretical Computer Science*, 96:157–174, 1992.

[3] M. Dam. Proving properties of dynamic process networks. *Information and Computation*, 140:95–114, 1998.

[4] Mads Dam and Dilian Gurov. μ-calculus with explicit points and approximations. *Journal of Logic and Computation*, 12(2):43–57, 2002. Previously appeared in Fixed Points in Computer Science, FICS '02.

[5] Lars-åke Fredlund. *A Framework for Reasoning about Erlang Code*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2001.

[6] D. Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[7] Damian Niwiński and Igor Walukiewicz. Games for the μ-calculus. *Theoretical Computer Science*, 163:99–116, 1997.

[8] David Park. Finiteness is mu-ineffable. *Theoretical Computer Science*, 3(2):173–181, 1976.

[9] S. Safra. On the complexity of ω-automata. In *29th IEEE Symposium on Fuondations of Computer Science*, pages 319–327, 1988.

[10] Ulrich Schöpp. Formal verification of processes. Master's thesis, University of Edinburgh, 2001.

[11] Ulrich Schöpp and Alex Simpson. Verifying temporal properties using explicit approximants: Completeness for context-free processes. In *FOSSACS '02, Grenoble, France*, volume 2303 of *LNCS*, pages 372–386. Springer-Verlag, 2002.

[12] Christoph Sprenger and Mads Dam. On the geometry of inductive reasoning: Circular and tree-shaped proofs in the μ-calculus. Submitted to FOSSACS '03, October 2002.

[13] Colin Stirling and David Walker. Local model checking in the modal μ-calculus. *Theoretical Computer Science*, 89:161–177, 1991.

Communicated by (The editor will be set by the publisher).
Fri Nov 8 18:38:45 MET 2002 .