The Event-B Proof Obligation Generator

Stefan Hallerstede
ETH Zürich

Version 6

Contents

1 Introduction

This text describes a proof obligation generator for EventB. Most of the document describes the actual generated proof obligations and justification of their correctness. The algorithm for their generation is very simple.

We distinguish generated proof obligations from theoretical ones. Theoretical proof obligations are well-suited for hand-written mathematical proofs but less suited for machine-assisted proof. In particular, generated proof obligations have be obtained by decomposing theoretical proof obligations as far as possible so that they are as simple as possible; and hopefully provable by an automatic prover. Substitutions produced by the proof obligation generator are left unevaluated. These are applied in a preprocessing step of the proof manager. The reason for this is to keep the design of the proof obligation generator distinct from the actual provers. By using witnesses in models a part of the proof has been moved into modelling itself. The price to pay is that one has to think about proving while modelling. The advantage is that proofs are decomposed and almost all existential quantifiers are removed from the consequents of proof obligations.

There are three main sections on contexts, initial models, and refined models. Each of these contains three subsections: the *description* subsection introduces the notation used in the section; the *theory* subsection presents the theoretical proof obligations and derives the generated proof obligations by proof; the *generated proof obligations* subsection contains the list of proof obligations to be generated by the proof obligation generator. This last section also contains proof obligations for well-definedness. On first reading well-definedness proof obligations should be ignored. These are necessary but are actually not derived from the theoretical proof obligations.

1.1 Naming Conventions

Throughout this document we use the following conventions to name items occurring in B developments. The names are used with arbitrary subscripts and superscripts.

contexts	B, C
context names	CTX
carrier set names	s
constant names	c
property names	PRP
property predicates	P
context theorem names	THM
context theorem predicates	Q
models	M, N
model names	MDL, REF
variables	o, v, w, x, y
external variables	$\overset{\times}{\mathrm{o}}$, $\overset{\times}{\mathrm{v}}$, $\overset{\times}{\mathrm{w}}$, $\overset{\times}{\mathrm{x}}$, $\overset{\times}{\mathrm{y}}$
invariant names	INV
invariant predicates	I, J, K
model theorem names	THM
model theorem predicates	Q
variant expressions	D
events	e
event names	EVT, EVM, EVN
guard names	GRD, GRM, GRN
guard predicates	G
local variables	t
substitutions	R, S, T, Ξ
witnesses	U, V, W

1.2 Context and Model Relationships

We denote by $C_1 \sqsubseteq C_2$ that context C_1 is refined by context C_2. Similarly, $M_1 \sqsubseteq M_2$ denotes that model M_1 is refined by model M_2. We use this notation also to represent chains of refinements

$$C_1 \sqsubseteq C_2 \sqsubseteq \ldots \sqsubseteq C_m \text{ , resp.}$$
$$M_1 \sqsubseteq M_2 \sqsubseteq \ldots \sqsubseteq M_n \ .$$

We denote by $M \rightarrow C$ that model M sees context C.

Using this notation we define a set of abstract operators on the structure of models and contexts. We must also show that these operators create proper sets of hypotheses, i.e. do not create type-conflicts. We must show that they are well-defined. Each definition is accompanied by an informal proof. We add an empty C_0 at the beginning of the refinement chains in order to simplify subsequent definitions and assume the following sees relationships between models and contexts:

$$
\begin{array}{ccccccccccccccccc}
C_0 & \sqsubseteq & \ldots & \sqsubseteq & C_{k_1} & \sqsubseteq & \ldots & \sqsubseteq & C_{k_2} & \sqsubseteq & \ldots & \sqsubseteq & C_{k_{n-1}} & \sqsubseteq & \ldots & \sqsubseteq & C_{k_n} \\
\uparrow & & & & \uparrow & & & & \uparrow & & & & \uparrow & & & & \uparrow \\
M_0 & & \sqsubseteq & & M_1 & & \sqsubseteq & & M_2 & \sqsubseteq & \ldots & \sqsubseteq & M_{n-1} & & \sqsubseteq & & M_n
\end{array}
$$

Instead of saying that a model sees an empty context we usually say that it sees no context. The operator $\sqcup$ used in the definitions joins two sets of predicates. Logically it corresponds to conjunction. Operator $\mathcal{P}$ yields the properties of a context C_ℓ:

$$\mathcal{P}(C_\ell) \;\widehat{=}\; \text{(properties of context } C_\ell\text{)}$$

Property 1 *$\mathcal{P}$ is well-defined.*

Operator $\mathcal{Q}$ yields the properties and theorems of a context C_ℓ and of all its abstractions:

$$
\begin{aligned}
\mathcal{Q}(C_0) &\;\widehat{=}\; \top \\
\mathcal{Q}(C_\ell) &\;\widehat{=}\; \text{(properties and theorems of context } C_\ell\text{)} \sqcup \mathcal{Q}(C_{\ell-1})
\end{aligned}
$$

Property 2 *$\mathcal{Q}$ is well-defined.*

Operator $\mathcal{J}$ yields the invariants a model M_ℓ:

$$\mathcal{J}(M_\ell) \;\widehat{=}\; \text{(invariants of model } M_\ell\text{)}$$

Property 3 *$\mathcal{J}$ is well-defined.*

Operator $\mathcal{I}$ yields the invariants and theorems of a model M_ℓ and of all its abstractions:

$$
\begin{aligned}
\mathcal{I}(M_0) &\;\widehat{=}\; \top \\
\mathcal{I}(M_\ell) &\;\widehat{=}\; \text{(invariants and theorems of model } M_\ell\text{)} \sqcup \mathcal{I}(M_{\ell-1})
\end{aligned}
$$

Property 4 *$\mathcal{I}$ is well-defined.*

Operator $\mathcal{U}$ yields the invariants and theorems of a model M_ℓ and of all its abstractions and the the properties and theorems of the seen context C_{k_ℓ} and of all its abstractions:

$$\mathcal{U}(M_\ell) \;\widehat{=}\; \mathcal{I}(M_\ell) \sqcup \mathcal{Q}(C_{k_\ell})$$

Property 5 *$\mathcal{U}$ is well-defined.*

1.3 Proof Obligations

Each proof obligation is described by the following structure:

Proof Obligation: REF

FOR	*obj* WHERE
	cnd
ID	"*NN*"
GPO	$\Sigma \ \vdash \ \Gamma$

where the entry *GPO* can be repeated for case distinction. **REF** is a symbolic name for the proof obligation. The structure has three entries FOR, ID, and GPO. The field FOR denotes the object (or the objects) *obj* for which the proof obligation is generated, and the condition *cnd* under which it is generated. The field *ID* contains the name *NN* of a generated proof obligation. Usually, *NN* is a compound name that contains some information about the generated proof obligation itself. Finally, the generated proof obligation in form of a sequent $\Sigma \vdash \Gamma$ is stated in field GPO. The typing environment $\mathcal{E}$ associated with each sequent is not stated explicitly in the proof obligations. It can be added to the hypothesis of the sequent: $\mathcal{E}; \Sigma \ \vdash \ \Gamma$. Note, that $\mathcal{E}$ depends on the items of the B model from which the proof obligation was generated. For instance, local variables may have different types in different events. The typing environment is provided to the proof obligation be the proof manager.

Note that the statement to be proved is the generated proof obligation GPO. By the term proof obligation we refer to the entire structure. All generated proof obligations must be uniquely identifiable by their name stated in field ID:

Property 6 (UNIQUE) *The name NN of a generated proof obligation is a unique name for that proof obligation.*

Furthermore, they must be well-defined:

Theorem 1 (WDEF) *Let*

$$\Sigma \ \vdash \ \Gamma$$

be a generated proof obligation. Then the formula Γ *and all formulas in* Σ *are well-defined.*

The operator WD used to express well-definedness of predicates and expressions is defined in Deliverable D3.2 (D7): *The Event-B Language*. The proof of Theorem 1 is split across all proof obligations. That is, we argue for its truth with each proof obligation stated. We use the property of WD that predicate $\mathsf{WD}(A)$ for some predicate A, respectively $\mathsf{WD}(E)$ for some expression E, is well-defined.

1.4 Derivation of Proof Obligations.

In order to show soundness, completeness, and necessity of the generated proof obligations we proceed as follows. We pretend to give a theoretical proof of correctness of a particular context, initial model or refined model. We rely on the static properties of Event-B models and the generated proof obligations. Static properties (e.g. well-definedness of $\mathcal{P}$, $\mathcal{Q}$, $\mathcal{J}$, $\mathcal{I}$, and $\mathcal{U}$) have been verified before the context, initial model or refined model is submitted for proof obligation generation. Hence, we can assume they hold. Conceptually, we assume we had proven all generated proof obligations as lemmas and then use them in the theoretical proofs. A proof obligation is called *necessary* if it is required by at least one theoretical proof. A collection of generated proof obligations is called *complete* if it is sufficient to discharge all theoretical proofs.

Soundness and completeness ensure that once all generated proof obligations have been discharged, the theoretical proof for context, initial model, or refined model have been achieved.

Necessity serves to verify that we do not generate too many proof obligations. This is needed for efficiency and practicality of proof obligation generator to be implement.

1.5 Differential Proof Obligation Generation

For each proof obligation there are four possibilities when comparing two sets of proof obligations of some context, initial model, or refined model:

it may be **unchanged**;
it may have been **changed**;
it may have been **added**;
it may have been **removed**.

We say a generated proof obligation depends *directly* on some item (e.g. an invariant or substitution) if the item occurs directly in its sequent (perhaps as a parameter of an abstract operator). A proof obligation depends *indirectly* on some item if the item is contained in a sequent but does not occur directly. For example, this is the case for properties contained in $\mathcal{P}(C)$. Note, however, that C itself occurs directly in the sequent and so it depends directly on C. The following algorithm is used to generate proof obligations differentially:

for all items of the context, initial model or refined model:
  generate the unique identifier *NN* of the associated proof obligation $\Sigma \;\vdash\; \Gamma$;
  if there is already a proof obligation with the same identifier,
    then
      if the proof obligation depends directly on a **changed** item,
        then
          generate new proof obligation and remove old;
          mark (new) proof obligation
        otherwise
          mark (old) proof obligation

otherwise

generate new proof obligation;

mark (new) proof obligation

finally, remove all unmarked proof obligations

This algorithm ensures that when items on which a particular proof obligation depends directly have been **changed** or **added**, the proof obligation is regenerated or generated. And if such an item has been **removed** the proof obligation is removed too. Proof obligations that do not refer, or only indirectly, to items that have **changed** are not regenerated. (They may still have to be reproved, though.)

This algorithm ensures also that we do not keep unnecessary proof obligations. It assumes that items that have **changed** have been marked as such before. This is done by a preprocessor that compares the items on which the old proof obligations are based with the items on which the new proof obligations will be based. The proof obligation generator keeps a copy of the old checked model (or context) for this purpose.

The decision whether a proof for a particular proof obligation is still valid or not lies with the proof manager. The proof obligation generator ignores this issue.

We note on the predicate set operators $\mathcal{P}$, $\mathcal{Q}$, $\mathcal{J}$, $\mathcal{I}$, and $\mathcal{U}$:

Theorem 2 *The sets $\mathcal{P}(C)$, $\mathcal{Q}(C)$, $\mathcal{J}(M)$, $\mathcal{I}(M)$, and $\mathcal{U}(M)$ do not depend on the order in which properties, context theorems, invariants, and model theorems appear in contexts and models.*

Proof: This follows directly from the way these sets are constructed. We only rely on the structure of contexts and models among each other. □

The validity of Theorem 2 is important for the efficiency of the proof obligation generator. Proof obligations refer symbolically to these sets and would have to be regenerated more often if the order was important. Assume we used parameterised versions, say, $\mathcal{P}_\ell(C)$ of operator $\mathcal{P}(C)$ containing the first ℓ properties of context C. Then $\mathcal{P}_\ell(C)$ would rely on the order in which the properties appear in C, and whenever we would make a change to that order we would have to replace $\mathcal{P}_\ell(C)$ in many proof obligations. In this case, we could put the properties contained in $\mathcal{P}_\ell(C)$ directly in the corresponding sequents. In fact, this is what we do in situations where the order is important, e.g., in well-definedness proof obligations for properties.

1.6 Operators

We use a number of terms and abstract operators to express the theoretical and the generated proof obligations. These are higher-order constructs that cannot be defined in terms of the B mathematical language.

Well-definedness Operator. The WD operator expresses a well-definedness condition for a predicate A or an expression E, written: $\mathsf{WD}(A)$ and $\mathsf{WD}(E)$, respectively.

Substitution. A *substitution* R has either of the following forms:

$\boxed{\mathsf{skip}}$ $\boxed{u := E}$ $\boxed{u :\in E}$ $\boxed{u :\mid A}$

where E is an expression that may contain occurrences of before-values v, and A is a predicate that may contain occurrences of before values v and after-values v'. A substitution of the form $u := E$ is called *simple* if u is a singleton, and *simultaneous* if u is a list with several variables.

Frame Operator. The frame $\mathsf{frame}(R)$ of a substitution R is the list of variables occurring on the left hand side of R. Each variable may only occur once in a frame. We use set-theoretic notation with frames: $\cup$ for union, $\cap$ for intersection, $\setminus$ for difference, $\varnothing$ for the empty frame.

Multiple Substitution. Lists of substitutions are written $R_1 \parallel \ldots \parallel R_n$ and are allowed to be empty. Such a list is called a *multiple substitution.* The frames of all component substitutions must be disjoint. The multiple substitution $R_1 \parallel \ldots \parallel R_n$ should be read like a parallel composition of the component substitutions R_ℓ, i.e. a simultaneous substitution. The frame $\mathsf{frame}(R)$ of a multiple substitution R is the union of the frames of the component substitutions.

Substitution Operator. For deterministic substitutions R of the form $u := E$ and multiple substitutions with deterministic components we introduce extra notation. In order to apply a multiple substitution R to a predicate A or expression E we define an operator $[R]$: we denote R applied to A by $[R]\,A$ and R applied to E by $[R]\,E$. If R is empty then $[R]$ is the identity. Substitution operators can be composed (sequentially), denoted by $[R_1]\,[R_2]\ldots[R_n]$. We refer to substitution operators as substitutions too, because it is always clear from the context (and notation) what is meant.

Guard Operator. The guard of an event e is the necessary condition under which it may occur. The guard operator yields this guard for event e. It is written $\mathsf{GD}(e)$.

Direct Before-After Operator. The BA operator returns the before-after predicate of a multiple substitution. For an empty multiple substitution R we define $\mathsf{BA}(R) = \top$. The before-after predicate of a substitution is defined by

$$
\begin{aligned}
\mathsf{BA}(\mathsf{skip}) &\;\widehat{=}\; \top \\
\mathsf{BA}(u := E) &\;\widehat{=}\; u' = E \;, \\
\mathsf{BA}(u :\in E) &\;\widehat{=}\; u' \in E \;, \\
\mathsf{BA}(u :\mid A) &\;\widehat{=}\; A \;.
\end{aligned}
$$

The before-after predicate of a non-empty multiple substitution $R_1 \parallel \ldots \parallel R_n$ is defined to be the conjunction of the before-after predicates of the components:

$$\mathsf{BA}(R_1 \parallel \ldots \parallel R_n) \;\widehat{=}\; \mathsf{BA}(R_1) \wedge \ldots \wedge \mathsf{BA}(R_n) \;.$$

Relative Before-After Operator. The BA_v operator returns the before-after predicate of a multiple substitution R relative to the variable list v. The frame of R must be contained in v. We define:

$$\mathsf{BA}_v(R) \quad \widehat{=} \quad \mathsf{BA}(R) \wedge \mathsf{BA}(\Xi) \ ,$$

where Ξ equals $u := u$ with $u = v \setminus \mathsf{frame}(R)$ which is similar to skip except that $\mathsf{frame}(\Xi) = u$.

Feasibility Operator. By $\mathsf{FIS}(R)$ we denote the *feasibility condition* of a substitution R. It is defined by:

$$\begin{array}{lcl} \mathsf{FIS}(\mathsf{skip}) & \widehat{=} & \top \\ \mathsf{FIS}(u := E) & \widehat{=} & \top \ , \\ \mathsf{FIS}(u :\in E) & \widehat{=} & E \neq \varnothing \ , \\ \mathsf{FIS}(u :\mid A) & \widehat{=} & \exists\, u' \cdot A \ . \end{array}$$

The operator $\mathsf{FIS}(R)$ is undefined for multiple substitutions.

Aside. An event is called *feasible* if all substitutions of its action are feasible. Because all events are required to be feasible in an event model, the term $\mathsf{GD}(e)$ corresponds to the formula $(\exists\, t \cdot G_1 \wedge .. \wedge G_g)$ where t are the local variables of e and $G_1, .., G_g$ are the explicitly stated guards of event e. We often use directly the formula $(\exists\, t \cdot G_1 \wedge .. \wedge G_g)$ instead of $\mathsf{GD}(e)$ for the guard of event e.

Freeness Operator. The free operator yields the list of free variables of a predicate A or an expression E, written: $\mathsf{free}(A)$ and $\mathsf{free}(E)$, respectively. Given a multiple substitution R the term $\mathsf{free}(R)$ denotes the variables occurring free in the right hand sides of the substitutions in R. If R is the empty multiple substitution, then $\mathsf{free}(R)$ is empty.

Primed Free Variables. We define the operator $\mathsf{primed}(X)$ where X is an expression E, a predicate A, or a substitution S, by: $u \in \mathsf{primed}(X) \Leftrightarrow u' \in \mathsf{free}(X)$.

Not-free-in Operator. The not-free-in operator nfin describes a relation between identifier lists z and predicates A or expressions E. We write z nfin A, respectively z nfin E, to say that z does not occur free in A, respectively E.

Local variables. In an event of the form **any** z **where** $\ldots$ **then** $\ldots$ **end**, z are called its *local variables.*

Property 7 (LOCAL) *Let z be local variables of some event e of some model M. Then*

$$z \ \mathsf{nfin}\ \mathcal{U}(M) \ .$$

2 Proof Obligations of Contexts

We first describe the structure of contexts, in the followings section we present the theoretical proof obligations. These are proven assuming that the generated proof obligations have already been proved. I.e. the generated proof obligations (plus the static properties) imply the theoretical proof obligations. The last section lists the generated proof obligations.

2.1 Description

This section presents the definitions required for formulating the theory and the proof obligations for contexts.

Let C be a context with name CTX with carrier sets s and constants c, and containing the following sequence of property and theorem declarations:

property PRP_1 P_1 $\vdots$ **property** PRP_m P_m	**theorem** THM_1 Q_1 $\vdots$ **theorem** THM_n Q_n

Let B be an abstraction of C, i.e. $B \sqsubseteq C$.

2.2 Theory

There is no relevant difference between initial contexts and refined contexts. Hence, they are treated uniformly in the theory and the proof obligations.

2.2.1 Context Theorems

We must prove that each theorem Q_ℓ is implied by properties of C and properties and theorems of its abstractions.

Theorem 3

$$\mathcal{Q}(B);\ \mathcal{P}(C);\ Q_1;\ \ldots;\ Q_{\ell-1} \vdash Q_\ell$$

Proof: This is trivially implied by CTX_THM. □

2.3 Generated Proof Obligations

2.3.1 Well-definedness of Properties

Proof Obligation: CTX_PRP_WD

FOR	**property** P_ℓ of C WHERE
	$\ell \in 1 \mathrel{..} m$
ID	"$CTX/PRP_\ell/$**WD**"
GPO	$\mathcal{Q}(B);\ P_1;\ \ldots;\ P_{\ell-1} \vdash \mathsf{WD}(P_\ell)$

Proof of WDEF: (See Theorem 1) The sequent is well-defined because context abstraction is an acyclic directed graph, and we can assume that we have shown well-definedness of $\mathcal{Q}(B)$, and $P_1 \ldots P_{\ell-1}$ before by CTX_PRP_WD. $\square$

2.3.2 Well-definedness of Theorems

Proof Obligation: CTX_THM_WD

FOR	**theorem** Q_ℓ of C WHERE
	$\ell \in 1 \,..\, n$
ID	"$CTX\,/\,THM_\ell/$**WD**"
GPO	$\mathcal{Q}(B);\ \mathcal{P}(C);\ Q_1;\ \ldots;\ Q_{\ell-1} \vdash \mathsf{WD}(Q_\ell)$

Proof of WDEF: The sequent is well-defined because context abstraction is an acyclic directed graph, and we can assume that we have shown well-definedness of $\mathcal{Q}(B)$ and $\mathcal{P}(C)$, and $Q_1 \ldots Q_{\ell-1}$ before by CTX_THM_WD. $\square$

2.3.3 Context Theorems

Proof Obligation: CTX_THM

FOR	**theorem** Q_ℓ of C WHERE
	$\ell \in 1 \,..\, n$
ID	"$CTX\,/\,THM_\ell/$**THM**"
GPO	$\mathcal{Q}(B);\ \mathcal{P}(C);\ Q_1;\ \ldots;\ Q_{\ell-1} \vdash Q_\ell$

Proof of WDEF: The sequent is well-defined because context abstraction is an acyclic directed graph, and we can assume that we have shown well-definedness of $\mathcal{Q}(B)$ and $\mathcal{P}(C)$, and $Q_1 \ldots Q_\ell$ before by CTX_THM_WD. $\square$

3 Proof Obligations of Initial Models

3.1 Description

Let M be an initial model with name MDL. Assume M sees context C with name CTX (or no context at all). Let v be the variables of M. Let M contain the following sequences of invariants and theorems:

invariant $INV_1\ I_1$ $\vdots$ **invariant** $INV_m\ I_m$	**theorem** $THM_1\ Q_1$ $\vdots$ **theorem** $THM_n\ Q_n$

Initialisation of M is partitioned into two parts corresponding to internal and external initialisation. The initialisations of M have the form:

$$\boxed{\begin{array}{l} R_1 \\ \vdots \\ R_r \end{array}}$$

for some $r \geq 1$, i.e. they have the form of an unguarded action $R_1 \parallel \ldots \parallel R_r$. All other events e (with name EVT) have the form

$$\boxed{\begin{array}{l} \textbf{any} \\ \quad t_1, \ldots, t_j \\ \textbf{where} \\ \quad GRD_1 \;\; G_1 \\ \quad \vdots \\ \quad GRD_g \;\; G_g \\ \textbf{then} \\ \quad R_1 \\ \quad \vdots \\ \quad R_r \\ \textbf{end} \end{array}}$$

for some $r \geq 1$ where $t_1, \ldots, t_j$ are the local variables (possibly none), $G_1, \ldots, G_g$ the guards (possibly none), and $R_1 \parallel \ldots \parallel R_r$ is the action of event e.

Remark. The various definitions should rather be read to specify patterns. Reusing place holder names and indices allows us to treat modelling items in a uniform way, thus, simplifies subsequent definitions. Still, the names and indices have been chosen such that we do not need to rename when using them in the theory (Section 3.2) and the proof obligations (Section 3.3).

3.1.1 Internal and External

Variables. We refer to external variables u of M by $\overset{\times}{u}$.

Initialisation. Internal and external initialisation assign only to internal or external variables respectively. The *combined initialisation* $R_1 \parallel \ldots \parallel R_k$ is defined by the list combining the internal and the external initialisation of M, i.e. it equals $R_1^\epsilon \parallel \ldots \parallel R_{r_\epsilon}^\epsilon \parallel R_1^\iota \parallel \ldots \parallel R_{r_\iota}^\iota$ where we use superscript ϵ to indicate external and superscript ι to indicate internal initialisation. This means the combined initialisation is the parallel composition of internal and external initialisation.

Events. External events only assign to external variables, and internal events to either kind of variable. We do not use special notation to distinguish internal and external events.

Remark. In initial models the distinction between internal and external has no significance with the exception of deadlock-freedom.

3.1.2 Actions

Whenever convenient we abbreviate an action $R_1 \parallel \ldots \parallel R_r$ by R.

Components. Let $R_1 \| \ldots \| R_r$ be an action. Each component R_ℓ ($\ell \in 1..r$) is a substitution of either form:

$$\boxed{\mathsf{skip}} \qquad \boxed{u_\ell := E_\ell} \qquad \boxed{u_\ell :\in E_\ell} \qquad \boxed{u_\ell :\mid A_\ell}$$

where for $\ell \in 1 \,..\, r$ the u_ℓ are all distinct. No variable occurs in more than one u_ℓ. A substitution $u_\ell(F) := E_\ell$ is to be rewritten into

$$u_\ell := u_\ell \mathbin{\lhd\!\!\!+} \{F \mapsto E_\ell\}$$

before it is subjected to proof obligation generation. We use the notation $R \sim X$ to say that R resembles substitution X, where X is one of the substitutions skip, $u_\ell := E_\ell$, $u_\ell :\in E_\ell$, or $u_\ell :\mid A_\ell$.

Partitioning. We can partition the action $R_1 \parallel \ldots \parallel R_r$ into S and T such that $S = R_{k_1} \parallel \ldots \parallel R_{k_p}$ is a multiple substitution with components of R of the form $w_{k_\ell} := E_{k_\ell}$ for $\ell \in 1 \,..\, p$; and $T = R_{i_1} \parallel \ldots \parallel R_{i_q}$ is a multiple substitution with components of R of the form $w_{i_\ell} :\in E_{i_\ell}$ or $w_{i_\ell} :\mid A_{i_\ell}$ for $\ell \in 1 \,..\, q$. Let v_X be the variables occurring on the left hand side of X, where X is one of R, S, or T. Note, that S or T, or both, can be empty. Note also, that R is the identity substitution on all variables that occur neither in v_S nor in v_T.

Restriction. For a substitution R and a list of variables z we define the restriction $R_{|z}$ of R to z by

$$R_{|z} \quad = \quad \text{all substitutions } R_\ell \text{ where a member of } z \text{ appears on the left hand side of } R_\ell$$

Note, that $R_{|z}$ can be the empty multiple substitution.

Primed Substitutions. For substitution (or witness) S of the form $u := E$ the primed variant S' is defined by $u' := E$. This generalises component-wise to multiple substitutions (and combined witnesses). Witnesses are defined in Section 4.1.3.

3.2 Theory

The theory of initial models is considerably simpler than the theory of refined models that is presented in Section 4. The simple reason is that initial models do not have refinement related proof obligations.

We must prove that the initial model M is consistent.

3.2.1 Model Theorems

We must prove that each theorem Q_ℓ is implied by properties of C and properties and theorems of its abstractions and the invariants of M.

Theorem 4

$$\mathcal{Q}(C);\ \mathcal{J}(M);\ Q_1;\ \ldots;\ Q_{\ell-1} \vdash Q_\ell$$

Proof: This is trivially implied by MDL_THM. $\square$

3.2.2 Feasibility of Initialisation

We must show that the combined initialisation of M is feasible assuming that only properties (and theorems) of the context C hold. Let R be the combined initialisation of M.

Theorem 5

$$\mathcal{Q}(C) \vdash \exists\, v' \cdot \mathsf{BA}_v(R)$$

Proof: Because v_R equals v in the combined initialisation we can replace BA_v by BA: $\mathcal{Q}(C) \vdash \exists\, v' \cdot \mathsf{BA}(R)$. Each after-value u' only appears on one conjunct of $\mathsf{BA}(R)$. This allows us to move the existential quantifiers into each conjunct: $\mathcal{Q}(C) \vdash \mathsf{FIS}(R_1) \wedge \ldots \wedge \mathsf{FIS}(R_r)$. We decompose this sequent into r sequents of the form $\mathcal{Q}(C) \vdash \mathsf{FIS}(R_\ell)$ where $\ell \in 1 \,..\, r$. Applying the definition of FIS this means we have nothing to prove in case $R_\ell \sim \mathsf{skip}$ or $R_\ell \sim u_\ell := E_\ell$. In the remaining two cases we have to prove $\mathcal{Q}(C) \vdash E_\ell \neq \varnothing$ if $R_\ell \sim u_\ell :\in E_\ell$, and $\mathcal{Q}(C) \vdash \exists\, u'_\ell \cdot A_\ell$ if $R_\ell \sim u_\ell :\mid A_\ell$. This corresponds to proving MDL_INI_FIS for all ℓ. $\square$

3.2.3 Invariant Establishment

We have to show that after initialisation of M the invariant holds assuming only properties (and theorems) of the context C. Let R be the combined initialisation of M.

Theorem 6

$$\mathcal{Q}(C);\ \mathsf{BA}_v(R) \vdash [v := v']\,(I_1 \wedge \ldots \wedge I_m)$$

Proof: Note that v_R equals v in the combined initialisation, hence, we can rewrite the sequent replacing BA_v by BA: $\mathcal{Q}(C);\ \mathsf{BA}(R) \vdash [v_R := v'_R]\,(I_1 \wedge \ldots \wedge I_m)$. First we decompose the sequent into m sequents: $\mathcal{Q}(C) \vdash \mathsf{BA}(R) \Rightarrow [v_R := v'_R]\, I_\ell$. We partition R into a deterministic part S and a non-deterministic part T: $\mathcal{Q}(C) \vdash \mathsf{BA}(T) \wedge \mathsf{BA}(S) \Rightarrow [v_R := v'_R]\, I_\ell$. The predicate $\mathsf{BA}(S)$ consists of a set of equations of the form $v'_S = \ldots$, hence, we can apply the equalities to the conclusion, $\mathcal{Q}(C) \vdash \mathsf{BA}(T) \Rightarrow [S']\,[v_R := v'_R]\, I_\ell$. Now we know that S and T do have disjoint left hand sides, thus, we can rewrite the conclusion once more to yield: $\mathcal{Q}(C) \vdash \mathsf{BA}(T) \Rightarrow [S]\,[v_T := v'_T]\, I_\ell$. Finally, we can restrict the substitutions S and T to the variables z occurring free in I_ℓ. This gives: $\mathcal{Q}(C) \vdash \mathsf{BA}(T_{|z}) \Rightarrow [S_{|z}]\,[v_{T_{|z}} := v'_{T_{|z}}]\, I_\ell$, i.e. MDL_INI_INV. $\square$

3.2.4 Feasibility of Event Actions

We must show that all events of M are feasible assuming that all of $\mathcal{U}(M)$ hold. For each event we must prove:

Theorem 7

$$\mathcal{U}(M) \vdash \forall\, t \cdot G_1 \wedge \ldots \wedge G_g \Rightarrow \exists\, v' \cdot \mathsf{BA}_v(R)$$

Proof: We eliminate all after-values v'_Ξ of variables outside the frame of R by applying the one-point rule: $\mathcal{U}(M) \vdash \forall\, t \cdot G_1 \wedge \ldots \wedge G_g \Rightarrow \exists\, v'_R \cdot \mathsf{BA}(R)$, and move the existential quantifiers into the conjuncts: $\mathcal{U}(M) \vdash \forall\, t \cdot G_1 \wedge \ldots \wedge G_g \Rightarrow \mathsf{FIS}(R_1) \wedge \ldots \wedge \mathsf{FIS}(R_r)$. Using Theorem 7 (Section 1.6) rewriting yields: $\mathcal{U}(M);\ G_1;\ \ldots;\ G_g \vdash \mathsf{FIS}(R_1) \wedge \ldots \wedge \mathsf{FIS}(R_r)$. We decompose this sequent into r sequents of the form $\mathcal{U}(M);\ G_1;\ \ldots;\ G_g \vdash \mathsf{FIS}(R_\ell)$ where $\ell \in 1\,..\,r$. Applying the definition of FIS this means we have nothing to prove in case $R_\ell \sim \mathsf{skip}$ or $R_\ell \sim u_\ell := E_\ell$. In the remaining two cases we have to prove $\mathcal{U}(M);\ G_1;\ \ldots;\ G_g \vdash E_\ell \neq \varnothing$ if $R_\ell \sim u_\ell :\in E_\ell$, and $\mathcal{U}(M);\ G_1;\ \ldots;\ G_g \vdash \exists\, u'_\ell \cdot A_\ell$ if $R_\ell \sim u_\ell :\mid A_\ell$. This corresponds to proving MDL_EVT_FIS for all ℓ. $\square$

3.2.5 Invariant Preservation

We must show that all events of M preserve the combined invariant. We must prove for each event:

Theorem 8

$$\mathcal{U}(M);\ (\exists\, t \cdot G_1 \wedge \ldots \wedge G_g);\ (\forall\, t \cdot G_1 \wedge \ldots \wedge G_g \Rightarrow \mathsf{BA}_v(R)) \vdash [v := v']\,(I_1 \wedge \ldots \wedge I_m)$$

Proof: Using Theorem 7 rewriting yields:

$$\mathcal{U}(M);\ G_1;\ \ldots;\ G_g;\ (\forall\, t \cdot G_1 \wedge \ldots \wedge G_g \Rightarrow \mathsf{BA}_v(R)) \vdash [v := v']\,(I_1 \wedge \ldots \wedge I_m)\ .$$

We instantiate t and apply modus ponens to produce the simpler sequent:

$$\mathcal{U}(M);\ G_1;\ \ldots;\ G_g;\ \mathsf{BA}_v(R) \vdash [v := v']\,(I_1 \wedge \ldots \wedge I_m)\ .$$

Using the one-point rule on Ξ (where $\mathsf{BA}_v(R) \Leftrightarrow \mathsf{BA}(R) \wedge \mathsf{BA}(\Xi)$) we can replace BA_v by BA, yielding: $\mathcal{U}(M);\ G_1;\ \ldots;\ G_g;\ \mathsf{BA}(R) \vdash [v_R := v'_R]\,(I_1 \wedge \ldots \wedge I_m)$. We decompose this sequent into m sequents: $\mathcal{U}(M);\ G_1;\ \ldots;\ G_g \vdash \mathsf{BA}(R) \Rightarrow [v_R := v'_R]\,I_\ell$. We partition R into a deterministic part S and a non-deterministic part T, and rewrite the claim: $\mathcal{U}(M);\ G_1;\ \ldots;\ G_g \vdash \mathsf{BA}(T) \wedge \mathsf{BA}(S) \Rightarrow [v_R := v'_R]\,I_\ell$. The predicate $\mathsf{BA}(S)$ consists of a set of equations of the form $v'_S = \ldots$, hence, we can apply the equalities to the conclusion, $\mathcal{U}(M);\ G_1;\ \ldots;\ G_g \vdash \mathsf{BA}(T) \Rightarrow [S']\,[v_R := v'_R]\,I_\ell$. Now we know that S and T do have disjoint left hand sides, thus, we can rewrite the conclusion once more to yield:

$$\mathcal{U}(M);\ G_1;\ \ldots;\ G_g \vdash \mathsf{BA}(T) \Rightarrow [S]\,[v_T := v'_T]\,I_\ell\ .$$

Finally, we can restrict the substitutions S and T to the variables z occurring free in I_ℓ. This gives: $\mathcal{U}(M);\ G_1;\ \ldots;\ G_g \vdash \mathsf{BA}(T_{|z}) \Rightarrow [S_{|z}]\,[v_{T_{|z}} := v'_{T_{|z}}]\,I_\ell$, i.e. MDL_EVT_INV. $\square$

3.2.6 Deadlock Freedom (Optional)

To show deadlock-freedom we must show that the disjunction of the guards of all internal events $e_1, \ldots, e_k$ of M is true.

Theorem 9

$$\mathcal{U}(M) \vdash \mathsf{GD}(e_1) \vee \ldots \vee \mathsf{GD}(e_k)$$

Proof: By MDL_DLK. □

3.2.7 (Internal) Anticipated Events

In an initial model anticipated events do not cause any different or additional proof obligations. The differences only appear in refinements (where new events are introduced).

3.2.8 Internal and External Events

All proof obligations must be proven for all events, internal and external. In a refinement external events can only be refined in a more constrained way. In an initial model there are no extra constraints on external events.

3.3 Generated Proof Obligations

3.3.1 Well-definedness of Invariants

Proof Obligation: MDL_INV_WD

FOR	**invariant** I_ℓ of M WHERE
	$\ell \in 1 \,..\, m$
ID	"*MDL/INV$_\ell$/***WD**"
GPO	$\mathcal{Q}(C);\ I_1;\ \ldots;\ I_{\ell-1} \vdash \mathsf{WD}(I_\ell)$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{Q}(C)$, and $I_1 \ldots I_{\ell-1}$ before by MDL_INV_WD. □

3.3.2 Well-definedness of Theorems

Proof Obligation: MDL_THM_WD

FOR	**theorem** Q_ℓ of M WHERE
	$\ell \in 1 \mathinner{..} n$
ID	"$\mathit{MDL}/\mathit{THM}_\ell/$**WD**"
GPO	$\mathcal{Q}(C);\ \mathcal{J}(M);\ Q_1;\ \ldots;\ Q_{\ell-1} \vdash \mathsf{WD}(Q_\ell)$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{J}(M)$, and $Q_1 \ldots Q_{\ell-1}$ before by MDL_THM_WD. $\square$

3.3.3 Model Theorems

Proof Obligation: MDL_THM

FOR	**theorem** Q_ℓ of M WHERE
	$\ell \in 1 \mathinner{..} n$
ID	"$\mathit{MDL}/\mathit{THM}_\ell/$**THM**"
GPO	$\mathcal{Q}(C);\ \mathcal{J}(M);\ Q_1;\ \ldots;\ Q_{\ell-1} \vdash Q_\ell$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{J}(M)$, and $Q_1 \ldots Q_\ell$ before by MDL_THM_WD. $\square$

3.3.4 Well-definedness of Initialisation

Proof Obligation: MDL_INI_WD

FOR	**substitution** R_ℓ of **combined initialisation** of M WHERE	
	$\ell \in 1 \mathinner{..} r$ AND $u_\ell = \mathsf{frame}(R_\ell)$	
ID	"$\mathit{MDL}/$**INIT**$/u_\ell/$**WD**"	
GPO	$\top$	(if $R_\ell \sim \mathsf{skip}$)
GPO	$\mathcal{Q}(C) \vdash \mathsf{WD}(E_\ell)$	(if $R_\ell \sim u_\ell := E_\ell$)
GPO	$\mathcal{Q}(C) \vdash \mathsf{WD}(E_\ell)$	(if $R_\ell \sim u_\ell :\in E_\ell$)
GPO	$\mathcal{Q}(C) \vdash \mathsf{WD}(A_\ell)$	(if $R_\ell \sim u_\ell :\mid A_\ell$)

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{Q}(C)$. $\square$

3.3.5 Feasibility of Initialisation

Proof Obligation: MDL_INI_FIS

FOR	**substitution** R_ℓ of **combined initialisation** of M WHERE
	$\ell \in 1\,..\,r$ AND $u_\ell = \mathsf{frame}(R_\ell)$
ID	"MDL/**INIT**/u_ℓ/**FIS**"
GPO	$\top$ (if $R_\ell \sim \mathsf{skip}$)
GPO	$\top$ (if $R_\ell \sim u_\ell := E_\ell$)
GPO	$\mathcal{Q}(C) \vdash E_\ell \neq \varnothing$ (if $R_\ell \sim u_\ell :\in E_\ell$)
GPO	$\mathcal{Q}(C) \vdash \exists\, u'_\ell \cdot A_\ell$ (if $R_\ell \sim u_\ell :\mid A_\ell$)

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{Q}(C)$, and of E_ℓ, respectively A_ℓ, by MDL_INI_WD. $\square$

3.3.6 Invariant Establishment

Proof Obligation: MDL_INI_INV

FOR	**combined initialisation** of M and **invariant** I_ℓ of M WHERE
	$\ell \in 1\,..\,m$ AND $z = \mathsf{free}(I_\ell)$
ID	"MDL/**INIT**/INV_ℓ/**INV**"
GPO	$\mathcal{Q}(C) \vdash \mathsf{BA}(T_{\mid z}) \Rightarrow [S_{\mid z}]\,[v_{T_{\mid z}} := v'_{T_{\mid z}}]\, I_\ell$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{Q}(C)$, and T and S by MDL_INI_WD, and I_ℓ by MDL_INV_WD. $\square$

3.3.7 Well-definedness of Guards

Proof Obligation: MDL_GRD_WD

FOR	**guard** G_ℓ of e of M WHERE
	$\ell \in 1 \mathinner{..} g$
ID	"$MDL/EVT/GRD_\ell/\mathbf{WD}$"
GPO	$\mathcal{U}(M);\; G_1;\; \ldots;\; G_{\ell-1} \vdash \mathsf{WD}(G_\ell)$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(M)$, and $G_1 \ldots G_{\ell-1}$ before by MDL_GRD_WD, and $t_1, \ldots t_j$ **nfin** $\mathcal{U}(M)$ by Theorem 7. □

3.3.8 Well-definedness of Event Actions

Proof Obligation: MDL_EVT_WD

FOR	**substitution** R_ℓ of e of M WHERE	
	$\ell \in 1 \mathinner{..} r$ AND $u_\ell = \mathsf{frame}(R_\ell)$	
ID	"$MDL/EVT/u_\ell/\mathbf{WD}$"	
GPO	$\top$	(if $R_\ell \sim \mathsf{skip}$)
GPO	$\mathcal{U}(M);\; G_1;\; \ldots;\; G_g \vdash \mathsf{WD}(E_\ell)$	(if $R_\ell \sim u_\ell := E_\ell$)
GPO	$\mathcal{U}(M);\; G_1;\; \ldots;\; G_g \vdash \mathsf{WD}(E_\ell)$	(if $R_\ell \sim u_\ell :\in E_\ell$)
GPO	$\mathcal{U}(M);\; G_1;\; \ldots;\; G_g \vdash \mathsf{WD}(A_\ell)$	(if $R_\ell \sim u_\ell :\mid A_\ell$)

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(M)$, and $G_1 \ldots G_g$ before by MDL_GRD_WD, and $t_1, \ldots t_j$ **nfin** $\mathcal{U}(M)$ by Theorem 7. □

3.3.9 Feasibility of Event Actions

Proof Obligation: MDL_EVT_FIS

FOR	**substitution** R_ℓ of e of M WHERE
	$\ell \in 1 \,..\, r$ AND $u_\ell = \mathsf{frame}(R_\ell)$
ID	"$MDL/EVT/u_\ell/$**FIS**"
GPO	$\top$ (if $R_\ell \sim \mathsf{skip}$)
GPO	$\top$ (if $R_\ell \sim u_\ell := E_\ell$)
GPO	$\mathcal{U}(M);\; G_1;\; \ldots;\; G_g \vdash E_\ell \neq \varnothing$ (if $R_\ell \sim u_\ell :\in E_\ell$)
GPO	$\mathcal{U}(M);\; G_1;\; \ldots;\; G_g \vdash \exists\, u'_\ell \cdot A_\ell$ (if $R_\ell \sim u_\ell :\mid A_\ell$)

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(M)$, and $G_1 \ldots G_g$ has be shown by MDL_GRD_WD, and that of E_ℓ (respectively A_ℓ) by MDL_EVT_WD, and $t_1, \ldots t_j$ **nfin** $\mathcal{U}(M)$ by Theorem 7. □

3.3.10 Invariant Preservation

Proof Obligation: MDL_EVT_INV

FOR	**event** e of M and **invariant** I_ℓ of M WHERE
	$\ell \in 1 \,..\, m$ AND $z = \mathsf{free}(I_\ell)$ AND $R_{\mid z}$ is not empty
ID	"$MDL/EVT/INV_\ell/$**INV**"
GPO	$\mathcal{U}(M);\; G_1;\; \ldots;\; G_g \vdash \mathsf{BA}(T_{\mid z}) \Rightarrow [S_{\mid z}]\,[v_{T_{\mid z}} := v'_{T_{\mid z}}]\, I_\ell$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(M)$, and T and S by MDL_EVT_WD, and I_ℓ by MDL_INV_WD, and $G_1 \ldots G_g$ has be shown by MDL_GRD_WD, and $t_1, \ldots t_j$ **nfin** $\mathcal{U}(M)$ by Theorem 7. □

Remark. If $R_{\mid z}$ is the empty multiple substitution, this proof obligation should not be generated because I_ℓ would appear in the antecedent and the consequent. This holds when the free variables of I_ℓ are not in the frame of R.

Remark. We cannot reduce the number of guards in the hypotheses because they can be transitively dependent. So we could render a provable proof obligation unprovable.

3.3.11 Deadlock Freedom (Optional)

Proof Obligation: MDL_DLK

FOR	**model** M WHERE	
	$e_1, \ldots, e_k$ are all internal events of M	
ID	"*MDL*/**DLK**"	
GPO	$\mathcal{U}(M) \vdash \mathsf{GD}(e_1) \vee \ldots \vee \mathsf{GD}(e_k)$	

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(M)$, and $\mathsf{GD}(e_1) \ldots \mathsf{GD}(e_k)$ follows from MDL_GRD_WD and the fact that the local variables t_{e_ℓ} of each event e_ℓ are bound by an existential quantifier in $\mathsf{GD}(e_\ell)$. ☐

Remark. We could equivalently generate the proof obligation:

$$\mathcal{U}(M);\ \neg\, \mathsf{GD}(e_1);\ \ldots;\ \neg\, \mathsf{GD}(e_{k-1}) \vdash \mathsf{GD}(e_k)\ .$$

Remark. This proof obligation should only be generated when all guards of all events of the model M are well-formed and well-typed. It should be avoided to present the user with proof obligations that may not be stable. For this proof obligation we know that if some evnets have not passed static-checking, then it will certainly change. If the user would prove a such proof obligation before it is stable, this would be nuisance.

Remark. The user who is creating a model has to decide whether or not to prove deadlock freedom. The corresponding information must be available to the proof obligation generator.

4 Proof Obligations for Refinements

4.1 Description

Let M be a model and N a refinement of M, i.e. $M \sqsubseteq N$. Assume N sees context C with name CTX (or no context at all). Let x be the variables that appear only in M, y be the variables that appear only in N, v all variables of M, and w all variables of N. In other words, x are the variables that disappear in the refinement step, and y are the newly introduced variables. Furthermore, let o be the variables occurring in M and N.

<table>
<tr><td colspan="2">M</td><td></td></tr>
<tr><td colspan="2">v</td><td></td></tr>
<tr><td>x</td><td>o</td><td>y</td></tr>
<tr><td></td><td colspan="2">w</td></tr>
<tr><td></td><td colspan="2">N</td></tr>
</table>

Let N contain the following sequences of invariants and theorems:

$$\boxed{\begin{array}{l}\textbf{invariant } INV_1 \; I_1 \\ \vdots \\ \textbf{invariant } INV_m \; I_m\end{array}} \qquad \boxed{\begin{array}{l}\textbf{theorem } THM_1 \; Q_1 \\ \vdots \\ \textbf{theorem } THM_n \; Q_n\end{array}}$$

One part of the invariant $I_1 \wedge \ldots \wedge I_m$ is called *external invariant*, denoted by $J_1 \wedge .. \wedge J_\sigma$. An external invariant can only refer to external variables of the refined and the abstract model. The remaining part of the invariant $I_1 \wedge \ldots \wedge I_m$ is called *internal*, and can refer to all variables of the refined model and the abstract model except the disappearing abstract external variables. Initialisation of M and N is partitioned into two parts corresponding to internal and external initialisation. The initialisations of have the form:

$$\boxed{\begin{array}{l}R_{M_1} \\ \vdots \\ R_{M_p}\end{array}} \qquad \boxed{\begin{array}{l}R_{N_1} \\ \vdots \\ R_{N_q}\end{array}}$$

for some $p \geq 1$ and $q \geq 1$. All other events e^M (with name EVT_M), respectively e^N (with name EVT_N), have the form

$$\boxed{\begin{array}{l}\textbf{any} \\ \quad t_1^M, \ldots, t_i^M \\ \textbf{where} \\ \quad GRM_1 \; G_1 \\ \quad \vdots \\ \quad GRM_g \; G_g \\ \textbf{then} \\ \quad R_{M_1} \\ \quad \vdots \\ \quad R_{M_p} \\ \textbf{end}\end{array}} \qquad \boxed{\begin{array}{l}\textbf{any} \\ \quad t_1^N, \ldots, t_j^N \\ \textbf{where} \\ \quad GRN_1 \; H_1 \\ \quad \vdots \\ \quad GRN_g \; H_h \\ \textbf{then} \\ \quad R_{N_1} \\ \quad \vdots \\ \quad R_{N_q} \\ \textbf{end}\end{array}}$$

for some $p \geq 1$ and $q \geq 1$; where $t_1^M, \ldots, t_i^M$ are the local variables (possibly none), $G_1, \ldots, G_g$ the guards (possibly none), and $R_{M_1} \parallel \ldots \parallel R_{M_p}$ is the action of event e^M; and $t_1^N, \ldots, t_j^N$ are the local variables (possibly none), $H_1, \ldots, H_h$ the guards (possibly none), and $R_{N_1} \parallel \ldots \parallel R_{N_q}$ is the action of event e^N.

External Variables. We refer to external variables u of M or N by $\overset{\times}{u}$.

4.1.1 Actions

We use similar conventions an notations as described in Section 3.1.2. The only difference is that we propagate the subscripts M and N, for instance, partitioning R_M into S_M and T_M.

4.1.2 Split and Merge

Split. For a split refinement of an event we do not need special notation. In fact, we treat this as the standard case of refinement.

Merge. For a merge refinement of a set of events $e_1^M, \ldots, e_k^M$ we need some more complicated notation for their guards. We let $G_{\ell,1}, \ldots, G_{\ell,g_\ell}$ be the guards of event e_ℓ^M for $\ell \in 1\,..\,k$. There is no need for further extra notation for merge refinements because all the events $e_1^M, \ldots, e_k^M$ are required to have identical local variables (in particular, identically typed) and identical actions (except for permutation of substitutions). Furthermore, no explicit use of guard names of $e_1^M, \ldots, e_k^M$ is made.

4.1.3 Witnesses

Witnesses serve to instantiate existential quantifiers in consequents. They are an important technique for decomposing complex proof obligations. We distinguish *explicit* and *default* witnesses.

Explicit Witnesses. Explicit witnesses are associated with events. The are two kinds of explicit witnesses, called *local* and *global*, used with events in a refined model:

Local witnesses of the form $t_\ell^M := E$, where t_ℓ^M is a local variable of the corresponding abstract event e^M, and E is an expression over constants, sets, local variables t^N, and global variables w of the refined model and their post-values w';

Local witnesses of the form $t_\ell^N := E$, where t_ℓ^N is a local variable of the corresponding refined event e^N, and E is an expression over constants, sets, local variables t^M, and global variables v of the abstract model and their post-values v';

Global witnesses of the form $u := E$, where u is contained in the disappearing abstract variables x, and E is an expression over constants, sets, variables w of the refined model and their post-values w', and local variables t^N of the event of the refined model (to which the witness belongs).

Abstract and Concrete Local Witnesses. Witnesses for abstract local variables t^M are used in the guard strengthening proof obligation. Witnesses for concrete local variables t^N are used in the guard equivalence proof obligation of external events (REF_GRD_EXT).

Derived Witnesses. The user interface could suggest certain invariants and theorems to be global witnesses if they are equations of the form $u = E$ where expression E must be an expression over constants, sets, and variables w of the refined model. This equation could be turned into a global witness by renaming the variables and rewriting the equation into a substitution: $u := E'$. The proof obligation generator does not do this. Similarly, the user interface could search for equalities in guards to suggest local witnesses.

Witnessed Variable. We call the variable occurring on the left hand side of a witness (i.e. its frame) the *witnessed variable.*

Default Local Witnesses. If local variables are repeated in a refined event, then they are required to be the same, i.e. the default local witness

$$u := u$$

is assumed. Note, that in order for this to be well-defined, the types of identically named local variables must also have identical types.

Default Global Witnesses. If global variables are repeated in a refined model, then they are required to be the same, i.e. the default global witness

$$u := u$$

is assumed. This corresponds just to the glueing invariant for identically named global variables (that is not stated explicitly in the refined model). Note, that in order for this to be well-defined, the types of identically named global variables must also have identical types. (This is checked by the static-checker.) This must be true transitively along the chain of abstractions of a model (as is already required for $\mathcal{I}(M)$ for some model M to be well-defined).

Use of Default Witnesses. Because default witnesses are identity substitutions they do not need to be explicitly part of generated proof obligations. However, if a default witness exists, it is not possible for the user to provide another witness for the concerned local or global variable.

Combined Local Witness. For local variables t^M of the abstract model M the *combined local witness* is defined to be the multiple substitution consisting of all non-default local witnesses $t_\ell^M := E$. The combined witness for abstract local variables is denoted by V_{t^M}. The combined witness for the local variables t^N of the concrete model V_{t^N} is defined similarly.

Combined Global Witness. For (disappearing) global variables x of the abstract model M the *combined global witness* is defined to be the multiple substitution consisting of all non-default global witnesses $u := E$. The combined witness for disappearing abstract variables ids denoted by W_x.

4.2 Theory

We have to prove that model N is a refinement of model M.

4.2.1 Model Theorems

We must prove that each theorem Q_ℓ is implied by properties of C and properties and theorems of its abstractions, and the invariants of M and the invariants and theorems of the abstractions of M. This proof obligation is similar to that for initial models.

Theorem 10

$$\mathcal{Q}(C);\ \mathcal{I}(M);\ \mathcal{J}(N);\ Q_1;\ \ldots;\ Q_{\ell-1} \vdash Q_\ell$$

Proof: This is trivially implied by REF_THM. □

4.2.2 External Invariant

The external invariant $J_1 \wedge \ldots \wedge J_\sigma$ must be functional from concrete to abstract disappearing variables, total, and surjective. Theorem 11 shows that it is functional, Theorem 12 shows that it is total, and Theorem 13 shows that it is surjective.

Theorem 11

$$\mathcal{Q}(C);\ [\overset{\times}{x} := \overset{\times}{x}]\, J_1;\ \ldots;\ [\overset{\times}{x} := \overset{\times}{x}]\, J_\sigma;\ [\overset{\times}{x} := \overset{\times}{x}']\, J_1;\ \ldots;\ [\overset{\times}{x} := \overset{\times}{x}']\, J_\sigma \vdash \overset{\times}{x} = \overset{\times}{x}'$$

Proof: The claim just corresponds to REF_EXT_FUN. □

Theorem 12

$$\mathcal{Q}(C) \vdash \forall\, \overset{\times}{x} \cdot \exists\, \overset{\times}{y} \cdot J_1 \wedge \ldots \wedge J_\sigma$$

Proof: The claim just corresponds to REF_EXT_TOT. □

Theorem 13

$$\mathcal{Q}(C) \vdash \forall\, \overset{\times}{y} \cdot \exists\, \overset{\times}{x} \cdot J_1 \wedge \ldots \wedge J_\sigma$$

Proof: The claim just corresponds to REF_EXT_SRJ. □

4.2.3 Feasibility of Initialisation

We must show that the combined initialisation of N is feasible assuming that only properties (and theorems) of the context C hold. This is the same proof obligation as for initial models.

Theorem 14

$$\mathcal{Q}(C) \vdash \exists\, w' \cdot \mathsf{BA}_w(R_N)$$

Proof: Similarly to Theorem 5 it is sufficient to prove: $\mathcal{Q}(C) \vdash \mathsf{FIS}(R_{N_1}) \wedge \ldots \wedge \mathsf{FIS}(R_{N_r})$. We decompose this sequent into r sequents of the form $\mathcal{Q}(C) \vdash \mathsf{FIS}(R_{N_\ell})$ where $\ell \in 1 \mathrel{..} r$. Applying the definition of FIS this means we have nothing to prove in case $R_{N_\ell} \sim \mathsf{skip}$ or $R_{N_\ell} \sim u_\ell := E_\ell$. In the remaining two cases we have to prove $\mathcal{Q}(C) \vdash E_\ell \neq \varnothing$ if $R_{N_\ell} \sim u_\ell :\in E_\ell$, and $\mathcal{Q}(C) \vdash \exists\, u_\ell' \cdot A_\ell$ if $R_{N_\ell} \sim u_\ell :\mid A_\ell$. This corresponds to proving MDL_INI_FIS for all ℓ. □

4.2.4 Simulation of Initialisation and Invariant Establishment

This proof obligation comprises simulation of initialisation and invariant establishment. We have to show that the combined initialisation of M can simulate the combined initialisation of N and that invariant of N holds after initialisation assuming only properties (and theorems) of the context C and its abstractions. Let R_M be the combined initialisation of M, and R_N be the combined initialisation of N. We use v'' to denote the after-state of abstract initialisation, and w' to denote the after-state of the refined initialisation.

Theorem 15

$$\begin{aligned}&\mathcal{Q}(C);\ \mathsf{BA}_v(R_N) \vdash \\ &\qquad \exists\, v'' \cdot [v' := v'']\,\mathsf{BA}_v(R_M) \wedge o' = o'' \wedge [x := x''][w := w']\,(I_1 \wedge \ldots \wedge I_m)\end{aligned}$$

Proof: Note that v_{R_M} equals v (resp. w_{R_N} equals w) in a initialisation, hence, we can rewrite the sequent replacing BA_v by BA:

$$\begin{aligned}&\mathcal{Q}(C);\ \mathsf{BA}(R_N) \vdash \\ &\qquad \exists\, v''_{R_M} \cdot [v'_{R_M} := v''_{R_M}]\,\mathsf{BA}(R_M) \wedge o' = o'' \wedge \\ &\qquad\qquad [x := x'']\,[w_{R_N} := w'_{R_N}]\,(I_1 \wedge \ldots \wedge I_m)\ .\end{aligned}$$

First we apply the one-point rule for the common variables o:

$$\begin{aligned}&\mathcal{Q}(C);\ \mathsf{BA}(R_N) \vdash \\ &\qquad \exists\, x'' \cdot [x' := x'']\,\mathsf{BA}(R_M) \wedge \\ &\qquad\qquad [x := x'']\,[w_{R_N} := w'_{R_N}]\,(I_1 \wedge \ldots \wedge I_m)\ .\end{aligned}$$

The abstract action R_M can be split into a deterministic part S_M and a non-deterministic part T_M:

$$\begin{aligned}&\mathcal{Q}(C);\ \mathsf{BA}(R_N) \vdash \\ &\qquad \exists\, x'' \cdot [x' := x'']\,(\mathsf{BA}(S_M) \wedge \mathsf{BA}(T_M)) \wedge \\ &\qquad\qquad [x := x'']\,[w_{R_N} := w'_{R_N}]\,(I_1 \wedge \ldots \wedge I_m)\ .\end{aligned}$$

Application of the one-point rule for $S_{M|x}$ yields:

$$\begin{aligned}&\mathcal{Q}(C);\ \mathsf{BA}(R_N) \vdash \\ &\qquad \exists\, x''_{T_M} \cdot [x'_{T_M} := x''_{T_M}]\,\mathsf{BA}(T_M) \wedge \mathsf{BA}(S_{M|o}) \wedge \\ &\qquad\qquad [S''_{M|x}]\,[x := x'']\,[w_{R_N} := w'_{R_N}]\,(I_1 \wedge \ldots \wedge I_m)\ .\end{aligned}$$

Now we instantiate the remaining disappearing variables x''_{T_M} using the global witness W_x. We assume the global witnesses have been chosen for the proof to succeed.

$$\begin{aligned}&\mathcal{Q}(C);\ \mathsf{BA}(R_N) \vdash \\ &\qquad [W''_x]\,[x'_{T_M} := x''_{T_M}]\,\mathsf{BA}(T_M) \wedge \mathsf{BA}(S_{M|o}) \wedge \\ &\qquad [W''_x]\,[S''_{M|x}]\,[x := x'']\,[w_{R_N} := w'_{R_N}]\,(I_1 \wedge \ldots \wedge I_m)\ .\end{aligned}$$

This simplifies to:

$$
\begin{aligned}
&\mathcal{Q}(C);\ \ \mathsf{BA}(R_N) \vdash \\
&\qquad [W'_x]\, \mathsf{BA}(T_M) \wedge \mathsf{BA}(S_{M|o}) \wedge \\
&\qquad [W_x]\, [S_{M|x}]\, [w_{R_N} := w'_{R_N}]\, (I_1 \wedge \ldots \wedge I_m)\ .
\end{aligned}
$$

We partition R_N into a deterministic part S_N and a non-deterministic part T_N, and rewrite the claim: $\mathcal{Q}(C);\ \ \mathsf{BA}(S_N);\ \ \mathsf{BA}(T_N) \vdash \ldots$. The predicate $\mathsf{BA}(S_N)$ consists of a set of equations of the form $w'_{S_N} = \ldots$, hence, we can apply the equalities to the conclusion,

$$
\begin{aligned}
&\mathcal{Q}(C);\ \ \mathsf{BA}(T_N) \vdash \\
&\qquad [S'_N]\, ([W'_x]\, \mathsf{BA}(T_M) \wedge \mathsf{BA}(S_{M|o})) \wedge && (1)\\
&\qquad [S'_N]\, [W_x]\, [S_{M|x}]\, [w_{R_N} := w'_{R_N}]\, (I_1 \wedge \ldots \wedge I_m)\ . && (2)
\end{aligned}
$$

In order to prove (1) we rewrite it to:

$$\mathcal{Q}(C);\ \ \mathsf{BA}(T_N) \vdash [S'_N]\, [W'_x]\, (\mathsf{BA}(T_M) \wedge \mathsf{BA}(S_{M|o}))\ .$$

This possible because x' does not occur free in $\mathsf{BA}(S_{M|o})$. We decompose this sequent into the sequents: $\mathcal{Q}(C);\ \ \mathsf{BA}(T_N) \vdash [S'_N]\, [W'_x]\, \mathsf{BA}(R_{M_\ell})$, where R_{M_ℓ} is not in $S_{M|x}$. Note, that (primed) abstract disappearing variables x' do not occur free in $\mathsf{BA}(S_{M|o})$. Finally, with $f = \mathsf{frame}(R_{M_\ell})$ and $z = \mathsf{primed}(W_{x|f})$, it is sufficient to prove:

$$\mathcal{Q}(C);\ \ \mathsf{BA}(T_{N|f \cup z}) \vdash [S'_{N|f \cup z}]\, [W'_{x|f}]\, \mathsf{BA}(R_{M_\ell})\ ,$$

i.e. REF_INI_SIM. In order to prove (2), we decompose the sequent

$$\mathcal{Q}(C);\ \ \mathsf{BA}(T_N) \vdash [S'_N]\, [W_x]\, [S_{M|x}]\, [w_{R_N} := w'_{R_N}]\, (I_1 \wedge \ldots \wedge I_m)$$

into m sequents of the form $\mathcal{Q}(C);\ \mathsf{BA}(T_N) \vdash [S'_N]\, [W_x]\, [S_{M|x}]\, [w_{R_N} := w'_{R_N}]\, I_\ell$ for $\ell \in 1 \,..\, m$. Letting $z = \mathsf{free}(I_\ell)$ and $\theta = \mathsf{primed}(W_{x|z}) \cup \mathsf{primed}(S_{M|x \cap z})$, it is thus sufficient to prove

$$\mathcal{Q}(C);\ \ \mathsf{BA}(T_{N|\theta \cup z}) \vdash [S'_{N|\theta \cup z}]\, [W_{x|z}]\, [S_{M|x \cap z}]\, [(w_{R_N} := w'_{R_N})_{|z}]\, I_\ell\ ,$$

i.e. REF_INI_INV. $\square$

4.2.5 Equivalent External Initialisation

We have to prove that the refined external initialisation is not less non-deterministic than the abstract external initialisation.

Theorem 16

$$
\begin{aligned}
&\mathcal{Q}(C);\ \ [\overset{\times}{v}{}' := \overset{\times}{v}{}'']\, \mathsf{BA}_{\overset{\times}{v}}(R_M); \\
&\qquad \overset{\times}{o}{}' = \overset{\times}{o}{}'';\ \ [\overset{\times}{x} := \overset{\times}{x}{}'']\, [\overset{\times}{y} := \overset{\times}{y}{}']\, J_1;\ \ \ldots;\ \ [\overset{\times}{x} := \overset{\times}{x}{}'']\, [\overset{\times}{y} := \overset{\times}{y}{}']\, J_\sigma \vdash \\
&\quad \mathsf{BA}_{\overset{\times}{w}}(R_N)
\end{aligned}
$$

Proof: We apply the equalities $\overset{\times}{o}' = \overset{\times}{o}''$:

$$\begin{aligned}
&\mathcal{Q}(C);\ \ [\overset{\times}{\mathrm{x}}' := \overset{\times}{\mathrm{x}}''] \,\mathsf{BA}_{\overset{\times}{v}}(R_M); \\
&\qquad\qquad [\overset{\times}{\mathrm{x}} := \overset{\times}{\mathrm{x}}''] \,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}'] \; J_1;\ \ \ldots;\ \ [\overset{\times}{\mathrm{x}} := \overset{\times}{\mathrm{x}}''] \,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}'] \; J_\sigma \vdash \\
&\qquad \mathsf{BA}_{\overset{\times}{w}}(R_N)\ .
\end{aligned}$$

Because x and w are distinct, we can rename x'' to x':

$$\begin{aligned}
&\mathcal{Q}(C);\ \ \mathsf{BA}_{\overset{\times}{v}}(R_M); \\
&\qquad\qquad [\overset{\times}{\mathrm{x}} := \overset{\times}{\mathrm{x}}'] \,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}'] \; J_1;\ \ \ldots;\ \ [\overset{\times}{\mathrm{x}} := \overset{\times}{\mathrm{x}}'] \,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}'] \; J_\sigma \vdash \\
&\qquad \mathsf{BA}_{\overset{\times}{w}}(R_N)\ .
\end{aligned}$$

We replace the relative before-after operators by relative before-after operators which is possible because external initialisations assign to all external variables (and only those).

$$\begin{aligned}
&\mathcal{Q}(C);\ \ \mathsf{BA}(R_M); \\
&\qquad\qquad [\overset{\times}{\mathrm{x}}_{R_M} := \overset{\times}{\mathrm{x}}'_{R_M}] \,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}'] \; J_1;\ \ \ldots;\ \ [\overset{\times}{\mathrm{x}}_{R_M} := \overset{\times}{\mathrm{x}}'_{R_M}] \,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}'] \; J_\sigma \vdash \\
&\qquad \mathsf{BA}(R_N)\ .
\end{aligned}$$

We split R_M into the deterministic part S_M and the non-deterministic part T_M, and apply the equalities $\mathsf{BA}(S_M)$:

$$\begin{aligned}
&\mathcal{Q}(C);\ \ \mathsf{BA}(T_M); \\
&\qquad\qquad [S_{M|x}] \,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}] \,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}'] \; J_1;\ \ \ldots;\ \ [S_{M|x}] \,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}] \,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}'] \; J_\sigma \vdash \\
&\qquad [S_{M|o}]\mathsf{BA}(R_N)\ .
\end{aligned}$$

We split this sequent into q sequents:

$$\begin{aligned}
&\mathcal{Q}(C);\ \ \mathsf{BA}(T_M); \\
&\qquad\qquad [S_{M|x}] \,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}] \,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}'] \; J_1;\ \ \ldots;\ \ [S_{M|x}] \,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}] \,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}'] \; J_\sigma \vdash \\
&\qquad [S_{M|o}] \,\mathsf{BA}(R_{N_\ell})\ .
\end{aligned}$$

where $\ell \in 1 \mathinner{..} q$. For all ℓ it is sufficient to prove

$$\begin{aligned}
&\mathcal{Q}(C);\ \ \mathsf{BA}(T_{M|x\cup f}); \\
&\qquad\qquad [S_{M|x}] \,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}] \,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}'] \; J_1;\ \ \ldots;\ \ [S_{M|x}] \,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}] \,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}'] \; J_\sigma \vdash \\
&\qquad [S_{M|f}] \,\mathsf{BA}(R_{N_\ell})\ .
\end{aligned}$$

where $f = \mathsf{frame}(R_{N_\ell})$, i.e. REF_INI_EXT. □

4.2.6 Feasibility of Events

We must show that all events of M are feasible assuming that all of $\mathcal{U}(M)$ hold. This is the same proof obligation as for initial models. For each event we must prove:

Theorem 17

$$\mathcal{U}(N) \vdash \forall\, t \cdot H_1 \wedge \ldots \wedge H_h \Rightarrow \exists\, w' \cdot \mathsf{BA}(R_N)$$

Proof: Similarly to Theorem 7 we only need to prove

$$\mathcal{U}(N) \vdash \forall\, t \cdot H_1 \wedge \ldots \wedge H_h \Rightarrow \mathsf{FIS}(R_{N_1}) \wedge \ldots \wedge \mathsf{FIS}(R_{N_q}) \ .$$

Using Theorem 7 rewriting yields: $\mathcal{U}(N);\ H_1;\ \ldots;\ H_h \vdash \mathsf{FIS}(R_{N_1}) \wedge \ldots \wedge \mathsf{FIS}(R_{N_q})$. We decompose this sequent into q sequents of the form $\mathcal{U}(N);\ H_1;\ \ldots;\ H_h \vdash \mathsf{FIS}(R_\ell)$ where $\ell \in 1\,..\,q$. Applying the definition of FIS this means we have nothing to prove in case $R_\ell \sim \mathsf{skip}$ or $R_\ell \sim u_\ell := E_\ell$. In the remaining two cases we have to prove $\mathcal{U}(N);\ H_1;\ \ldots;\ H_h \vdash E_\ell \neq \varnothing$ if $R_\ell \sim u_\ell :\in E_\ell$, and $\mathcal{U}(N);\ H_1;\ \ldots;\ H_h \vdash \exists\, u'_\ell \cdot A_\ell$ if $R_\ell \sim u_\ell :|\ A_\ell$. This corresponds to proving REF_EVT_FIS for all ℓ. □

4.2.7 Before-States and After-States

In proof obligations that deal with refinement of events we must rename global variables of the abstract model in order to achieve disjoint state spaces, for instance, $[o := o_1]\mathcal{U}(M)$. Furthermore, we add a predicate $o = o_1$ assuming equality of the before states to the antecedent. So we have a sequent like: $[o := o_1]\mathcal{U}(M);\ o = o_1;\ \ldots \vdash \ldots$. We can apply the equalities $o = o_1$ to the entire sequent to remove o_1 from all predicates. We state all proof obligations after this renaming has been carried out and o_1 does not appear anymore.

After-states of refined model events are named w', and after-states of the abstract model events are named v''. Abstract model event after-states only appear existentially quantified in the consequent. After application of the global witnesses all abstract after-states v'' are removed.

4.2.8 Guard Strengthening of Events

We have to prove that the guards of refined events are stronger than the guards of their abstract counterparts. We have two cases, one for events that are split (perhaps only into one event) and for events that are merged. We deal with the split case first:

Theorem 18

$$\mathcal{U}(N);\ H_1;\ \ldots;\ H_h \vdash \exists\, t^M \cdot G_1 \wedge \ldots \wedge G_g$$

Proof: Because of the feasibility of the event of the refined model we can add its before-after predicate to the hypotheses. This implies that this theorem is proved as part of Theorem 20. (In fact, we must prove it as part of Theorem 20 because the witnesses must be the same.) □

The merge case is similar:

Theorem 19

$$\mathcal{U}(N);\ H_1;\ \ldots;\ H_h \vdash \exists\, t^M \cdot ((G_{1,1} \wedge \ldots \wedge G_{1,g_1}) \vee \ldots \vee (G_{k,1} \wedge \ldots \wedge G_{k,g_k}))$$

Proof: Because of the feasibility of the event of the refined model we can add its before-after predicate to the hypotheses. This implies that this theorem is proved as part of Theorem 21. □

4.2.9 Simulation of Events and Invariant Preservation

We have to show that the action of the abstract event can simulate the action of the refined event and the resulting after-states satisfy the invariant (provided the before-states satisfy the invariant).

Split case. In case of a split refinement the following must hold:

Theorem 20

$$
\begin{aligned}
&\mathcal{U}(N);\ (\exists\, t^N \cdot H_1 \wedge \ldots \wedge H_h);\ (\forall\, t^N \cdot H_1 \wedge \ldots \wedge H_h \Rightarrow \mathsf{BA}_w(R_N)) \vdash \\
&\qquad \exists\, t^M \cdot G_1 \wedge \ldots \wedge G_g \wedge (\exists\, v'' \cdot [v' := v'']\, \mathsf{BA}_v(R_M)) \wedge \\
&\qquad\qquad o' = o'' \wedge \\
&\qquad\qquad [x := x'']\,[w := w']\,(I_1 \wedge \ldots \wedge I_m)
\end{aligned}
$$

Proof: Using Theorem 7 on the local variables t^N rewriting yields:

$$
\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ (\forall\, t^N \cdot H_1 \wedge \ldots \wedge H_h \Rightarrow \mathsf{BA}_w(R_N)) \vdash \\
&\qquad \exists\, t^M \cdot G_1 \wedge \ldots \wedge G_g \wedge (\exists\, v'' \cdot [v' := v'']\, \mathsf{BA}_v(R_M)) \wedge \\
&\qquad\qquad o' = o'' \wedge \\
&\qquad\qquad [x := x'']\,[w := w']\,(I_1 \wedge \ldots \wedge I_m)\ .
\end{aligned}
$$

We instantiate t^N in the antecedent and apply modus ponens to produce the simpler sequent:

$$
\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}_w(R_N) \vdash \\
&\qquad \exists\, t^M \cdot G_1 \wedge \ldots \wedge G_g \wedge (\exists\, v'' \cdot [v' := v'']\, \mathsf{BA}_v(R_M)) \wedge \\
&\qquad\qquad o' = o'' \wedge \\
&\qquad\qquad [x := x'']\,[w := w']\,(I_1 \wedge \ldots \wedge I_m)\ .
\end{aligned}
$$

We assume the combined witness V_{t^M} has been chosen for the proof to succeed:

$$
\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}_w(R_N) \vdash \\
&\qquad [V_{t^M}]\, G_1 \wedge \ldots \wedge [V_{t^M}]\, G_g \wedge \\
&\qquad (\exists\, v'' \cdot [V_{t^M}]\,[v' := v'']\, \mathsf{BA}_v(R_M)) \wedge \\
&\qquad\quad o' = o'' \wedge \\
&\qquad\quad [x := x'']\,[w := w']\,(I_1 \wedge \ldots \wedge I_m)\ .
\end{aligned}
$$

We replace BA_w by BA denoting by w_{Ξ_N} the variables that are not in the frame of R_N, and similarly for the abstract action R_M where w_{Ξ_M} denotes the variables not in the frame. This yields:

$$
\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(R_N);\ w_{\Xi_N} = w'_{\Xi_N} \vdash \\
&\qquad [V_{t^M}]\, G_1 \wedge \ldots \wedge [V_{t^M}]\, G_g \wedge && (1) \\
&\qquad \exists\, v'' \cdot [V_{t^M}]\,[v'_{R_M} := v''_{R_M}]\, \mathsf{BA}(R_M) \wedge && (2) \\
&\qquad\quad v_{\Xi_M} = v''_{\Xi_M} \wedge o' = o'' \wedge \\
&\qquad\quad [x := x'']\,[w := w']\,(I_1 \wedge \ldots \wedge I_m)\ .
\end{aligned}
$$

To prove sequent (1) we split R_N into a deterministic part S_N and a non-deterministic part T_N, and apply the equalities $w_{\Xi_N} = w'_{\Xi_N}$ and $\mathsf{BA}(S_N)$:

$$\begin{array}{l} \mathcal{U}(N);\; H_1;\; \ldots;\; H_h;\; \mathsf{BA}(T_N) \vdash \\ \qquad [S'_N]\,[w'_{\Xi_N} := w_{\Xi_N}]\,[V_{t^M}]\,G_1 \wedge \ldots \wedge [S'_N]\,[w'_{\Xi_N} := w_{\Xi_N}]\,[V_{t^M}]\,G_g \end{array}$$

We split this sequent into g sequents:

$$\mathcal{U}(N);\; H_1;\; \ldots;\; H_h;\; \mathsf{BA}(T_N) \vdash [S'_N]\,[w'_{\Xi_N} := w_{\Xi_N}]\,[V_{t^M}]\,G_\ell$$

for $\ell \in 1 \,..\, g$. Letting $z = \mathsf{free}(G_\ell)$ and $\psi = \mathsf{primed}(V_{t^M|z})$ it is sufficient to prove

$$\mathcal{U}(N);\; H_1;\; \ldots;\; H_h;\; \mathsf{BA}(T_{N|\psi}) \vdash [S'_{N|\psi}]\,[(w'_{\Xi_N} := w_{\Xi_N})_{|\psi}]\,[V_{t^M|z}]\,G_\ell$$

i.e. REF_GRD_REF (see also Theorem 18). Sequent (2) is proved by Theorem 22. $\square$

Merge case. In case of a merge refinement the following must hold (Remember that for events to be merged we require their actions to be identical.):

Theorem 21

$$\begin{array}{l} \mathcal{U}(N);\; (\exists\, t^N \cdot H_1 \wedge \ldots \wedge H_h);\; (\forall\, t^N \cdot H_1 \wedge \ldots \wedge H_h \Rightarrow \mathsf{BA}_w(R_N)) \vdash \\ \qquad \exists\, t^M \cdot ((G_{1,1} \wedge \ldots \wedge G_{1,g_1}) \vee \ldots \vee (G_{k,1} \wedge \ldots \wedge G_{k,g_k})) \wedge \\ \qquad\qquad (\exists\, v'' \cdot [v' := v'']\,\mathsf{BA}_v(R_M)) \wedge \\ \qquad\qquad\qquad o' = o'' \wedge \\ \qquad\qquad\qquad [x := x'']\,[w := w']\,(I_1 \wedge \ldots \wedge I_m) \end{array}$$

Proof: The proof is almost identical to that of Theorem 18. Using Theorem 7 on the local variables t^N rewriting yields:

$$\begin{array}{l} \mathcal{U}(N);\; H_1;\; \ldots;\; H_h;\; (\forall\, t^N \cdot H_1 \wedge \ldots \wedge H_h \Rightarrow \mathsf{BA}_w(R_N)) \vdash \\ \qquad \exists\, t^M \cdot ((G_{1,1} \wedge \ldots \wedge G_{1,g_1}) \vee \ldots \vee (G_{k,1} \wedge \ldots \wedge G_{k,g_k})) \wedge \\ \qquad\qquad (\exists\, v'' \cdot [v' := v'']\,\mathsf{BA}_v(R_M)) \wedge \\ \qquad\qquad\qquad o' = o'' \wedge \\ \qquad\qquad\qquad [x := x'']\,[w := w']\,(I_1 \wedge \ldots \wedge I_m) \end{array}$$

We instantiate t^N in the antecedent and apply modus ponens to produce the simpler sequent:

$$\begin{array}{l} \mathcal{U}(N);\; H_1;\; \ldots;\; H_h;\; \mathsf{BA}_w(R_N) \vdash \\ \qquad \exists\, t^M \cdot ((G_{1,1} \wedge \ldots \wedge G_{1,g_1}) \vee \ldots \vee (G_{k,1} \wedge \ldots \wedge G_{k,g_k})) \wedge \\ \qquad\qquad (\exists\, v'' \cdot [v' := v'']\,\mathsf{BA}_v(R_M)) \wedge \\ \qquad\qquad\qquad o' = o'' \wedge \\ \qquad\qquad\qquad [x := x'']\,[w := w']\,(I_1 \wedge \ldots \wedge I_m) \end{array}$$

We assume the combined witness V_{t^M} has been chosen for the proof to succeed:

$$\begin{array}{l}\mathcal{U}(N);\; H_1;\; \ldots;\; H_h;\; \mathsf{BA}_w(R_N) \vdash \\ \quad [V_{t^M}]\,((G_{1,1} \wedge \ldots \wedge G_{1,g_1}) \vee \ldots \vee (G_{k,1} \wedge \ldots \wedge G_{k,g_k})) \wedge \\ \quad (\exists\, v'' \cdot [v' := v'']\,\mathsf{BA}_v(R_M)) \wedge \\ \qquad o' = o'' \wedge \\ \qquad [x := x'']\,[w := w']\,(I_1 \wedge \ldots \wedge I_m)\end{array}$$

We replace BA_w by BA denoting by w_{Ξ_N} the variables that are not in the frame of R_N, and similarly for the abstract action R_M where w_{Ξ_M} denotes the variables not in the frame. This yields:

$$\begin{array}{ll}\mathcal{U}(N);\; H_1;\; \ldots;\; H_h;\; \mathsf{BA}(R_N);\; w_{\Xi_N} = w'_{\Xi_N} \vdash & \\ \quad [V_{t^M}]\,((G_{1,1} \wedge \ldots \wedge G_{1,g_1}) \vee \ldots \vee (G_{k,1} \wedge \ldots \wedge G_{k,g_k})) \wedge & (1) \\ \quad \exists\, v'' \cdot [V_{t^M}]\,[v'_{R_M} := v''_{R_M}]\,\mathsf{BA}(R_M) \wedge & (2) \\ \qquad v_{\Xi_M} = v''_{\Xi_M} \wedge o' = o'' \wedge & \\ \qquad [x := x'']\,[w := w']\,(I_1 \wedge \ldots \wedge I_m)\ . & \end{array}$$

To prove sequent (1) we split R_N into a deterministic part S_N and a non-deterministic part T_N, and apply the equalities $w_{\Xi_N} = w'_{\Xi_N}$ and $\mathsf{BA}(S_N)$:

$$\begin{array}{l}\mathcal{U}(N);\; H_1;\; \ldots;\; H_h;\; \mathsf{BA}(T_N) \vdash \\ \quad [S'_N]\,[w'_{\Xi_N} := w_{\Xi_N}]\,[V_{t^M}]\,((G_{1,1} \wedge \ldots \wedge G_{1,g_1}) \vee \ldots \vee (G_{k,1} \wedge \ldots \wedge G_{k,g_k}))\ .\end{array}$$

Letting $\psi = \mathsf{primed}(V_{t^M})$ it is sufficient to prove

$$\begin{array}{l}\mathcal{U}(N);\; H_1;\; \ldots;\; H_h;\; \mathsf{BA}(T_{N|\psi}) \vdash \\ \quad [S'_{N|\psi}]\,[(w'_{\Xi_N} := w_{\Xi_N})_{|\psi}]\,[V_{t^M}]\,((G_{1,1} \wedge \ldots \wedge G_{1,g_1}) \vee \ldots \vee (G_{k,1} \wedge \ldots \wedge G_{k,g_k}))\ .\end{array}$$

i.e. REF_GRD_MRG (see also Theorem 19). Sequent (2) is proved by Theorem 22. □

Theorem 22

$$\begin{array}{l}\mathcal{U}(N);\; H_1;\; \ldots;\; H_h;\; \mathsf{BA}(R_N);\; w_{\Xi_N} = w'_{\Xi_N} \vdash \\ \quad \exists\, v'' \cdot [V_{t^M}]\,[v'_{R_M} := v''_{R_M}]\,\mathsf{BA}(R_M) \wedge \\ \qquad v_{\Xi_M} = v''_{\Xi_M} \wedge o' = o'' \wedge \\ \qquad [x := x'']\,[w := w']\,(I_1 \wedge \ldots \wedge I_m)\ .\end{array}$$

Proof: We apply the one-point rule for the common variables o:

$$\begin{array}{l}\mathcal{U}(N);\; H_1;\; \ldots;\; H_h;\; \mathsf{BA}(R_N);\; w_{\Xi_N} = w'_{\Xi_N} \vdash \\ \quad \exists\, x'' \cdot [V_{t^M}]\,[x'_{R_M} := x''_{R_M}]\,\mathsf{BA}(R_M) \wedge \\ \qquad [o'' := o']\,v_{\Xi_M} = v''_{\Xi_M} \wedge \\ \qquad [x := x'']\,[w := w']\,(I_1 \wedge \ldots \wedge I_m)\ .\end{array}$$

We split v_{Ξ_M} into to sets of disappearing variables x_{Ξ_M} and common variables o_{Ξ_M}:

$$\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(R_N);\ w_{\Xi_N} = w'_{\Xi_N} \vdash \\
&\quad \exists\, x'' \cdot [\,V_{t^M}]\,[x'_{R_M} := x''_{R_M}]\,\mathsf{BA}(R_M) \wedge \\
&\qquad x_{\Xi_M} = x''_{\Xi_M} \wedge o_{\Xi_M} = o'_{\Xi_M} \wedge \\
&\qquad [x := x'']\,[w := w']\,(I_1 \wedge \ldots \wedge I_m)\ .
\end{aligned}$$

We apply the one-point law to $x_{\Xi_M} = x''_{\Xi_M}$ (note, that primed variables do not occur free in V_{t^M}):

$$\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(R_N);\ w_{\Xi_N} = w'_{\Xi_N} \vdash \\
&\quad \exists\, x''_{R_M} \cdot [\,V_{t^M}]\,[x'_{R_M} := x''_{R_M}]\,\mathsf{BA}(R_M) \wedge \\
&\qquad o_{\Xi_M} = o'_{\Xi_M} \wedge \\
&\qquad [x_{R_M} := x''_{R_M}]\,[w := w']\,(I_1 \wedge \ldots \wedge I_m)\ .
\end{aligned}$$

Now we split the abstract action R_M into a deterministic part S_M and a non-deterministic part T_M:

$$\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(R_N);\ w_{\Xi_N} = w'_{\Xi_N} \vdash \\
&\quad \exists\, x''_{R_M} \cdot [\,V_{t^M}]\,[x'_{R_M} := x''_{R_M}]\,(\mathsf{BA}(S_M) \wedge \mathsf{BA}(T_M)) \wedge \\
&\qquad o_{\Xi_M} = o'_{\Xi_M} \wedge \\
&\qquad [x_{R_M} := x''_{R_M}]\,[w := w']\,(I_1 \wedge \ldots \wedge I_m)\ .
\end{aligned}$$

We can apply the one-point rule for $[\,V_{t^M}]\,\mathsf{BA}(S''_{M|x})$:

$$\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(R_N);\ w_{\Xi_N} = w'_{\Xi_N} \vdash \\
&\quad \exists\, x''_{T_M} \cdot [\,V_{t^M}]\,[x'_{T_M} := x''_{T_M}]\,(\mathsf{BA}(T_M) \wedge \mathsf{BA}(S_{M|o})) \wedge \\
&\qquad o_{\Xi_M} = o'_{\Xi_M} \wedge \\
&\qquad [\,V_{t^M}]\,[S''_{M|x}]\,[x_{R_M} := x''_{R_M}]\,[w := w']\,(I_1 \wedge \ldots \wedge I_m)\ .
\end{aligned}$$

We instantiate the remaining disappearing variables x''_{T_M} using the global witness W_x, assuming they have been chosen for the proof to succeed:

$$\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(R_N);\ w_{\Xi_N} = w'_{\Xi_N} \vdash \\
&\quad [\,W''_x]\,[\,V_{t^M}]\,[x'_{T_M} := x''_{T_M}]\,(\mathsf{BA}(T_M) \wedge \mathsf{BA}(S_{M|o})) \wedge \\
&\quad o_{\Xi_M} = o'_{\Xi_M} \wedge \\
&\quad [\,W''_x]\,[\,V_{t^M}]\,[S''_{M|x}]\,[x_{R_M} := x''_{R_M}]\,[w := w']\,(I_1 \wedge \ldots \wedge I_m)\ .
\end{aligned}$$

We can swap W''_x and V_{t^M} because $\mathsf{frame}(W''_x) \cap \mathsf{frame}(V_{t^M})$ is empty, $x'' \notin \mathsf{free}(V_{t^M})$, and $t^M \setminus t^N \notin \mathsf{free}(W''_x)$:

$$\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(R_N);\ w_{\Xi_N} = w'_{\Xi_N} \vdash \\
&\quad [\,V_{t^M}]\,[\,W''_x]\,[x'_{T_M} := x''_{T_M}]\,(\mathsf{BA}(T_M) \wedge \mathsf{BA}(S_{M|o})) \wedge \\
&\quad o_{\Xi_M} = o'_{\Xi_M} \wedge \\
&\quad [\,V_{t^M}]\,[\,W''_x]\,[S''_{M|x}]\,[x_{R_M} := x''_{R_M}]\,[w := w']\,(I_1 \wedge \ldots \wedge I_m)\ .
\end{aligned}$$

We simplify and apply the equalities $w_{\Xi_N} = w'_{\Xi_N}$:

$$\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(R_N) \vdash \\
&\qquad [w'_{\Xi_N} := w_{\Xi_N}]\,[V_{t^M}]\,[W'_x]\,(\mathsf{BA}(T_M) \wedge \mathsf{BA}(S_{M|o})) \wedge \\
&\qquad [w'_{\Xi_N} := w_{\Xi_N}]\,o_{\Xi_M} = o'_{\Xi_M} \wedge \\
&\qquad [w'_{\Xi_N} := w_{\Xi_N}]\,[V_{t^M}]\,[W_x]\,[S_{M|x}]\,[w_{R_N} := w'_{R_N}]\,(I_1 \wedge \ldots \wedge I_m)\ .
\end{aligned}$$

We partition R_N into a deterministic part S_N and a non-deterministic part T_N, and rewrite the claim:

$$\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(T_N) \vdash \\
&\qquad [S'_N]\,[w'_{\Xi_N} := w_{\Xi_N}]\,[V_{t^M}]\,[W'_x]\,(\mathsf{BA}(T_M) \wedge \mathsf{BA}(S_{M|o})) \wedge && (1)\\
&\qquad [S'_N]\,[w'_{\Xi_N} := w_{\Xi_N}]\,o_{\Xi_M} = o'_{\Xi_M} \wedge && (2)\\
&\qquad [S'_N]\,[w'_{\Xi_N} := w_{\Xi_N}]\,[V_{t^M}]\,[W_x]\,[S_{M|x}]\,[w_{R_N} := w'_{R_N}]\,(I_1 \wedge \ldots \wedge I_m)\ . && (3)
\end{aligned}$$

This sequent can be decomposed into three sequents: (1) deals with simulation by R_M, (2) deals with simulation by Ξ_M, and (3) deals with invariant preservation. Sequent (1), i.e. $\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(T_N) \vdash [S'_N]\,[w'_{\Xi_N} := w_{\Xi_N}]\,([W'_x]\,\mathsf{BA}(T_M) \wedge \mathsf{BA}(S_{M|o}))$ can be decomposed into the sequents

$$\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(T_N) \vdash \\
&\qquad [S'_N]\,[w'_{\Xi_N} := w_{\Xi_N}]\,[V_{t^M}]\,[W'_x]\,\mathsf{BA}(R_{M_\ell})
\end{aligned}$$

for $R_{M_\ell} \notin S_{M|x}$. Letting $f = \mathsf{frame}(R_{M_\ell})$, $\psi = \mathsf{free}(R_{M_\ell})$, and $\chi = \mathsf{primed}(W_{x|f})$, it is sufficient to prove:

$$\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(T_{N|f\cup\chi}) \vdash \\
&\qquad [S'_{N|f\cup\chi}]\,[(w'_{\Xi_N} := w_{\Xi_N})_{f\cup\chi}]\,[V_{t^M|\psi}]\,[W'_{x|f}]\,\mathsf{BA}(R_{M_\ell})\ ,
\end{aligned}$$

i.e. REF_EVT_SIM_Δ. Sequent (2) is proved by REF_EVT_SIM_Ξ for the common variables u of M and N that are not in the frame of R_M but are in the frame of R_N (in other words $u \in o \cap (\mathsf{frame}(R_N) \setminus \mathsf{frame}(R_M))$:

$$\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(T_{N|u}) \vdash [S'_{N|u}]\,u = u'\ .$$

In the case where u is not in either frame, sequent (1) is trivially true. Sequent (3) can be decomposed into m sequents:

$$\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(T_N) \vdash \\
&\qquad [S'_N]\,[w'_{\Xi_N} := w_{\Xi_N}]\,[V_{t^M}]\,[W_x]\,[S_{M|x}]\,[w_{R_N} := w'_{R_N}]\,I_\ell\ ,
\end{aligned}$$

for $\ell \in 1\,..\,m$, and for each ℓ it is sufficient to prove:

$$\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(T_{N|\eta\cap z}) \vdash \\
&\qquad [S'_{N|\eta\cap z}]\,[(w'_{\Xi_N} := w_{\Xi_N})_{|\eta\cap z}]\,[V_{t^M|\phi}]\,[W_{x|z}]\,[S_{M|x\cap z}]\,[(w_{R_N} := w'_{R_N})_{|z}]\,I_\ell\ .
\end{aligned}$$

where $z = \mathsf{free}(I_\ell)$, $\phi = \mathsf{free}(S_{M|x\cap z})$, and $\eta = \mathsf{primed}(W_{x|z}) \cup \mathsf{primed}(S_{M|x\cap z})$, i.e. proof obligation REF_EVT_INV. $\square$

4.2.10 Guard Weakening of External Events

Theorem 23

$$\mathcal{Q}(C);\;\; J_1;\;\; \ldots;\;\; J_\sigma;\;\; G_1;\;\; \ldots;\;\; G_g \vdash \exists\, t^N \cdot H_1 \land \ldots \land H_h$$

Proof: Because of the feasibility of the abstract event and surjectivity of $J_1 \land \ldots \land J_\sigma$ interpreted as a mapping from states of the refined model to states of the abstract model, we can add the abstract before-after predicate and the external invariant to the hypotheses:

$$\begin{aligned}
&\mathcal{Q}(C);\;\; J_1;\;\; \ldots;\;\; J_\sigma;\;\; G_1;\;\; \ldots;\;\; G_g;\\
&\qquad [\overset{\times}{\mathsf{x}} := \overset{\times}{\mathsf{x}}''\,]\, \mathsf{BA}_{\overset{\times}{v}}(R_M);\;\; [\overset{\times}{\mathsf{x}} := \overset{\times}{\mathsf{x}}''\,]\, [\overset{\times}{\mathsf{y}} := \overset{\times}{\mathsf{y}}'\,]\, J_1;\;\; \ldots;\;\; [\overset{\times}{\mathsf{x}} := \overset{\times}{\mathsf{x}}''\,]\, [\overset{\times}{\mathsf{y}} := \overset{\times}{\mathsf{y}}'\,]\, J_\sigma \vdash\\
&\quad \exists\, t^N \cdot H_1 \land \ldots \land H_h\ .
\end{aligned}$$

This is proved as part of Theorem 24. □

Remark. Guard strengthening (Theorem 18) and guard weakening (Theorem 23) of external events together imply that the guards of external events are equivalent.

4.2.11 Equivalent External Events

Theorem 24

$$\begin{aligned}
&\mathcal{Q}(C);\;\; J_1;\;\; \ldots;\;\; J_\sigma;\;\; G_1;\;\; \ldots;\;\; G_g;\;\; [\overset{\times}{v} := \overset{\times}{v}''\,]\, \mathsf{BA}_{\overset{\times}{v}}(R_M);\\
&\qquad \overset{\times}{o}' = \overset{\times}{o}'';\;\; [\overset{\times}{x} := \overset{\times}{x}''\,]\, [\overset{\times}{y} := \overset{\times}{y}'\,]\, J_1;\;\; \ldots;\;\; [\overset{\times}{x} := \overset{\times}{x}''\,]\, [\overset{\times}{y} := \overset{\times}{y}'\,]\, J_\sigma \vdash\\
&\quad \exists\, t^N \cdot H_1 \land \ldots \land H_h \land \mathsf{BA}_{\overset{\times}{w}}(R_N)
\end{aligned}$$

Proof: We apply the equalities $\overset{\times}{o}' = \overset{\times}{o}''$, yielding:

$$\begin{aligned}
&\mathcal{Q}(C);\;\; J_1;\;\; \ldots;\;\; J_\sigma;\;\; G_1;\;\; \ldots;\;\; G_g;\;\; [\overset{\times}{\mathsf{x}} := \overset{\times}{\mathsf{x}}''\,]\, \mathsf{BA}_{\overset{\times}{v}}(R_M);\\
&\qquad [\overset{\times}{\mathsf{x}} := \overset{\times}{\mathsf{x}}''\,]\, [\overset{\times}{\mathsf{y}} := \overset{\times}{\mathsf{y}}'\,]\, J_1;\;\; \ldots;\;\; [\overset{\times}{\mathsf{x}} := \overset{\times}{\mathsf{x}}''\,]\, [\overset{\times}{\mathsf{y}} := \overset{\times}{\mathsf{y}}'\,]\, J_\sigma \vdash\\
&\quad \exists\, t^N \cdot H_1 \land \ldots \land H_h \land \mathsf{BA}_{\overset{\times}{w}}(R_N)
\end{aligned}$$

Because x and w are distinct, we can rename x'' to x':

$$\begin{aligned}
&\mathcal{Q}(C);\;\; J_1;\;\; \ldots;\;\; J_\sigma;\;\; G_1;\;\; \ldots;\;\; G_g;\;\; \mathsf{BA}_{\overset{\times}{v}}(R_M);\\
&\qquad [\overset{\times}{\mathsf{x}} := \overset{\times}{\mathsf{x}}'\,]\, [\overset{\times}{\mathsf{y}} := \overset{\times}{\mathsf{y}}'\,]\, J_1;\;\; \ldots;\;\; [\overset{\times}{\mathsf{x}} := \overset{\times}{\mathsf{x}}'\,]\, [\overset{\times}{\mathsf{y}} := \overset{\times}{\mathsf{y}}'\,]\, J_\sigma \vdash\\
&\quad \exists\, t^N \cdot H_1 \land \ldots \land H_h \land \mathsf{BA}_{\overset{\times}{w}}(R_N)\ .
\end{aligned}$$

We assume that the witnesses for t^N have been chosen for the proof to succeed:

$$\begin{aligned}
&\mathcal{Q}(C);\;\; J_1;\;\; \ldots;\;\; J_\sigma;\;\; G_1;\;\; \ldots;\;\; G_g;\;\; \mathsf{BA}_{\overset{\times}{v}}(R_M);\\
&\qquad [\overset{\times}{\mathsf{x}} := \overset{\times}{\mathsf{x}}'\,]\, [\overset{\times}{\mathsf{y}} := \overset{\times}{\mathsf{y}}'\,]\, J_1;\;\; \ldots;\;\; [\overset{\times}{\mathsf{x}} := \overset{\times}{\mathsf{x}}'\,]\, [\overset{\times}{\mathsf{y}} := \overset{\times}{\mathsf{y}}'\,]\, J_\sigma \vdash\\
&\quad [\,V_{t^N}]\, H_1 \land \ldots \land [\,V_{t^N}]\, H_h \land \qquad (1)\\
&\quad [\,V_{t^N}]\, \mathsf{BA}_{\overset{\times}{w}}(R_N)\ . \qquad (2)
\end{aligned}$$

We split sequent (1) into h sequents:

$$
\begin{array}{l}
\mathcal{Q}(C);\ J_1;\ \ldots;\ J_\sigma;\ G_1;\ \ldots;\ G_g;\ \mathsf{BA}_{\check{v}}(R_M);\\
\qquad [\check{x} := \check{x}']\,[\check{y} := \check{y}']\,J_1;\ \ldots;\ [\check{x} := \check{x}']\,[\check{y} := \check{y}']\,J_\sigma \vdash\\
\quad [V_{t^N}]\,H_\ell\ .
\end{array}
$$

for $\ell \in 1\,..\,h$. The before-after predicate can be split according to the frame of R_M, and the latter can split into a deterministic part S_M and a non-deterministic part T_M:

$$
\begin{array}{l}
\mathcal{Q}(C);\ J_1;\ \ldots;\ J_\sigma;\ G_1;\ \ldots;\ G_g;\ \mathsf{BA}(T_M);\ \mathsf{BA}(S_M);\ \mathsf{BA}(\Xi_M);\\
\qquad [\check{x} := \check{x}']\,[\check{y} := \check{y}']\,J_1;\ \ldots;\ [\check{x} := \check{x}']\,[\check{y} := \check{y}']\,J_\sigma \vdash\\
\quad [V_{t^N}]\,H_\ell\ .
\end{array}
$$

We apply the equalities $\mathsf{BA}(S_M)$ and $\mathsf{BA}(\Xi_M)$ to yield:

$$
\begin{array}{l}
\mathcal{Q}(C);\ J_1;\ \ldots;\ J_\sigma;\ G_1;\ \ldots;\ G_g;\ \mathsf{BA}(T_M);\\
\qquad [S_{M|x}]\,[\check{x}_{T_M} := \check{x}'_{T_M}]\,[\check{y} := \check{y}']\,J_1;\ \ldots;\ [S_{M|x}]\,[\check{x}_{T_M} := \check{x}'_{T_M}]\,[\check{y} := \check{y}']\,J_\sigma \vdash\\
\quad [S'_M]\,[\check{v}'_{\Xi_M} := \check{v}_{\Xi_M}]\,[V_{t^N}]\,H_\ell\ .
\end{array}
$$

Letting $z = \mathsf{free}(H_\ell)$ and $\psi = \mathsf{primed}(V_{t^N|z})$ it is sufficient to prove:

$$
\begin{array}{l}
\mathcal{Q}(C);\ J_1;\ \ldots;\ J_\sigma;\ G_1;\ \ldots;\ G_g;\ \mathsf{BA}(T_{M|x\cup\psi});\\
\qquad [S_{M|x}]\,[\check{x}_{T_M} := \check{x}'_{T_M}]\,[\check{y} := \check{y}']\,J_1;\ \ldots;\ [S_{M|x}]\,[\check{x}_{T_M} := \check{x}'_{T_M}]\,[\check{y} := \check{y}']\,J_\sigma \vdash\\
\quad [S'_{M|\psi}]\,[(\check{v}'_{\Xi_M} := \check{v}_{\Xi_M})_{|\psi}]\,[V_{t^N|z}]\,H_\ell\ .
\end{array}
$$

i.e. REF_GRD_EXT. Sequent (2) remains to be proved:

$$
\begin{array}{l}
\mathcal{Q}(C);\ J_1;\ \ldots;\ J_\sigma;\ G_1;\ \ldots;\ G_g;\ \mathsf{BA}_{\check{v}}(R_M);\\
\qquad [\check{x} := \check{x}']\,[\check{y} := \check{y}']\,J_1;\ \ldots;\ [\check{x} := \check{x}']\,[\check{y} := \check{y}']\,J_\sigma \vdash\\
\quad [V_{t^N}]\,\mathsf{BA}_{\check{w}}(R_N)\ .
\end{array}
$$

This equivalent to:

$$
\begin{array}{l}
\mathcal{Q}(C);\ J_1;\ \ldots;\ J_\sigma;\ G_1;\ \ldots;\ G_g;\ \mathsf{BA}(R_M);\ \mathsf{BA}(\Xi_M);\\
\qquad [\check{x} := \check{x}']\,[\check{y} := \check{y}']\,J_1;\ \ldots;\ [\check{x} := \check{x}']\,[\check{y} := \check{y}']\,J_\sigma \vdash\\
\quad [V_{t^N}]\,\mathsf{BA}(R_N) \land\\
\quad [V_{t^N}]\,\mathsf{BA}(\Xi_N)\ .
\end{array}
$$

We apply the equalities $\mathsf{BA}(\Xi_M)$. This yields (Ξ_N does not refer to local variables):

$$
\begin{array}{l}
\mathcal{Q}(C);\ J_1;\ \ldots;\ J_\sigma;\ G_1;\ \ldots;\ G_g;\ \mathsf{BA}(R_M);\\
\qquad [\check{x}_{R_M} := \check{x}'_{R_M}]\,[\check{y} := \check{y}']\,J_1;\ \ldots;\ [\check{x}_{R_M} := \check{x}'_{R_M}]\,[\check{y} := \check{y}']\,J_\sigma \vdash\\
\quad [\check{v}'_{\Xi_M} := \check{v}_{\Xi_M}]\,[V_{t^N}]\,\mathsf{BA}(R_N) \land\\
\quad [\check{o}'_{\Xi_M} := \check{o}_{\Xi_M}]\,(\check{w}'_{\Xi_N} = \check{w}_{\Xi_N})\ .
\end{array}
$$

We split R_M into a deterministic substitution S_M and a non-deterministic substitution T_M, and apply the equalities $\mathsf{BA}(S_M)$:

$$\begin{aligned}
&\mathcal{Q}(C);\ J_1;\ \ldots;\ J_\sigma;\ G_1;\ \ldots;\ G_g;\ \mathsf{BA}(T_M);\\
&\qquad [S_{M|x}]\,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}]\,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}']\,J_1;\ \ldots;\ [S_{M|x}]\,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}]\,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}']\,J_\sigma \vdash\\
&\quad [S'_M]\,[\overset{\times}{\mathrm{v}}'_{\Xi_M} := \overset{\times}{\mathrm{v}}_{\Xi_M}]\,[V_{t^N}]\,\mathsf{BA}(R_N)\ \wedge && (3)\\
&\quad [S'_{M|o}]\,[\overset{\times}{\mathrm{o}}'_{\Xi_M} := \overset{\times}{\mathrm{o}}_{\Xi_M}]\,(\overset{\times}{\mathrm{w}}'_{\Xi_N} = \overset{\times}{\mathrm{w}}_{\Xi_N})\ . && (4)
\end{aligned}$$

We prove sequent (3) by splitting it into q sequents:

$$\begin{aligned}
&\mathcal{Q}(C);\ J_1;\ \ldots;\ J_\sigma;\ G_1;\ \ldots;\ G_g;\ \mathsf{BA}(T_M);\\
&\qquad [S_{M|x}]\,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}]\,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}']\,J_1;\ \ldots;\ [S_{M|x}]\,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}]\,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}']\,J_\sigma \vdash\\
&\quad [S'_M]\,[\overset{\times}{\mathrm{v}}'_{\Xi_M} := \overset{\times}{\mathrm{v}}_{\Xi_M}]\,[V_{t^N}]\,\mathsf{BA}(R_{N_\ell})\ ,
\end{aligned}$$

where $\ell \in 1\,..\,q$. Letting $f = \mathsf{frame}(R_{N_\ell})$ it is sufficient to prove:

$$\begin{aligned}
&\mathcal{Q}(C);\ J_1;\ \ldots;\ J_\sigma;\ G_1;\ \ldots;\ G_g;\ \mathsf{BA}(T_{M|x\cup f});\\
&\qquad [S_{M|x}]\,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}]\,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}']\,J_1;\ \ldots;\ [S_{M|x}]\,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}]\,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}']\,J_\sigma \vdash\\
&\quad [S'_{M|f}]\,[(\overset{\times}{\mathrm{v}}'_{\Xi_M} := \overset{\times}{\mathrm{v}}_{\Xi_M})_{|f}]\,[V_{t^N}]\,\mathsf{BA}(R_{N_\ell})\ ,
\end{aligned}$$

i.e. REF_EVT_GEN_Δ. Sequent (4) is proved by

$$\begin{aligned}
&\mathcal{Q}(C);\ J_1;\ \ldots;\ J_\sigma;\ G_1;\ \ldots;\ G_g;\ \mathsf{BA}(T_M);\\
&\qquad [S_{M|x}]\,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}]\,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}']\,J_1;\ \ldots;\ [S_{M|x}]\,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}]\,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}']\,J_\sigma \vdash\\
&\quad [S'_{M|o}]\,(u = u')\ ,
\end{aligned}$$

for all $u \in o \cap (\mathsf{frame}(R_M) \setminus \mathsf{frame}(R_N))$. Thus it is sufficient to prove:

$$\begin{aligned}
&\mathcal{Q}(C);\ J_1;\ \ldots;\ J_\sigma;\ G_1;\ \ldots;\ G_g;\ \mathsf{BA}(T_{M|x\cup u});\\
&\qquad [S_{M|x}]\,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}]\,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}']\,J_1;\ \ldots;\ [S_{M|x}]\,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}]\,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}']\,J_\sigma \vdash\\
&\quad [S'_{M|u}]\,(u = u')\ ,
\end{aligned}$$

i.e. REF_EVT_GEN_Ξ. $\square$

4.2.12 Simulation of Skip and Invariant Preservation

If an ordinary event is introduced we only need to prove that it preserves the invariant and refines skip.

Theorem 25

$$\begin{aligned}
&\mathcal{U}(N);\ (\exists\, t^N \cdot H_1 \wedge \ldots \wedge H_h);\ (\forall\, t^N \cdot H_1 \wedge \ldots \wedge H_h \Rightarrow \mathsf{BA}_w(R_N)) \vdash\\
&\quad \exists\, v'' \cdot [v' := v'']\,\mathsf{BA}_v(\mathsf{skip})\ \wedge\\
&\qquad\quad o' = o''\ \wedge\\
&\qquad\quad [x := x'']\,[w := w']\,(I_1 \wedge \ldots \wedge I_m)
\end{aligned}$$

Proof: We proceed similarly to the proof of Theorem 20. After simplifying the antecedent we obtain:

$$\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}_w(R_N) \vdash \\
&\qquad \exists\, v'' \cdot [v' := v'']\, \mathsf{BA}_v(\mathsf{skip}) \wedge \\
&\qquad\qquad o' = o'' \wedge \\
&\qquad\qquad [x := x'']\, [w := w']\, (I_1 \wedge \ldots \wedge I_m)\ .
\end{aligned}$$

The predicate $\mathsf{BA}_v(\mathsf{skip})$ is $v' = v$, hence, we can simplify using the one-point rule:

$$\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}_w(R_N) \vdash \\
&\qquad [v'' := v]\, o' = o'' \wedge \\
&\qquad [v'' := v]\, [x := x'']\, [w := w']\, (I_1 \wedge \ldots \wedge I_m)\ .
\end{aligned}$$

We continue simplifying:

$$\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}_w(R_N) \vdash \\
&\qquad o' = o \wedge \\
&\qquad [w := w']\, (I_1 \wedge \ldots \wedge I_m)\ .
\end{aligned}$$

We replace BA_w by BA, and apply the equalities:

$$\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(R_N) \vdash \\
&\qquad [w'_{\Xi_N} := w_{\Xi_N}]\, o' = o \wedge \\
&\qquad [w'_{\Xi_N} := w_{\Xi_N}]\, [w := w']\, (I_1 \wedge \ldots \wedge I_m)\ .
\end{aligned}$$

Thus,

$$\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(R_N) \vdash \\
&\qquad o'_{R_N} = o_{R_N} \wedge \\
&\qquad [w_{R_N} := w'_{R_N}]\, (I_1 \wedge \ldots \wedge I_m)\ .
\end{aligned}$$

We split R_N into a deterministic part S_N and a non-deterministic part T_N, apply the equalities $\mathsf{BA}(S_N)$, and simplify:

$$\begin{aligned}
&\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(T_N) \vdash \\
&\qquad [S'_N]\, o'_{R_N} = o_{R_N} \wedge & (1)\\
&\qquad [S_N]\, [w_{T_N} := w'_{T_N}]\, (I_1 \wedge \ldots \wedge I_m)\ . & (2)
\end{aligned}$$

In order to prove (1), it is sufficient to show:

$$\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(T_{N|u}) \vdash [S'_{N|u}]\, u' = u$$

for all $u \in \mathsf{frame}(R_N) \cap o$, i.e. REF_NEW_SIM. To show (2) we prove m sequents:

$$\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(T_N) \vdash [S_N]\, [w_{T_N} := w'_{T_N}]\, I_\ell\ ,$$

where $\ell \in 1 \mathrel{..} m$. Letting $z = \mathsf{free}(I_\ell)$, it suffices to prove:

$$\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(T_{N|z}) \vdash [S_{N|z}]\, [(w_{T_N} := w'_{T_N})_{|z}]\, I_\ell\ ,$$

i.e. REF_NEW_INV. □

4.2.13 Reduction of a Set Variant

The variant of a model must be a finite set. It is decreased by convergent events; it is not increased by anticipated events.

Theorem 26

$$\mathcal{U}(N) \vdash \text{finite}(D)$$

Proof: This is trivially proven by REF_VAR_FIN_$\mathbb{P}$. $\square$

Theorem 27

$$\mathcal{U}(N);\ \ (\exists\, t^N \cdot H_1 \wedge \ldots \wedge H_h);\ \ (\forall\, t^N \cdot H_1 \wedge \ldots \wedge H_h \Rightarrow \mathsf{BA}_w(R_N)) \vdash ([w := w']\, D) \subseteq D$$

Proof: We proceed similarly to the first steps of the proof of Theorem 20 to obtain:

$$\mathcal{U}(N);\ \ H_1;\ \ \ldots;\ \ H_h;\ \ \mathsf{BA}_w(R_N) \vdash ([w := w']\, D) \subseteq D\ .$$

Thus,

$$\mathcal{U}(N);\ \ H_1;\ \ \ldots;\ \ H_h;\ \ \mathsf{BA}(R_N) \vdash ([w_{R_N} := w'_{R_N}]\, D) \subseteq D\ .$$

We split R_N into a deterministic part S_N and a non-deterministic part T_N, apply the equalities $\mathsf{BA}(S_N)$, and simplify:

$$\mathcal{U}(N);\ \ H_1;\ \ \ldots;\ \ H_h;\ \ \mathsf{BA}(T_N) \vdash ([S_N][w_{T_N} := w'_{T_N}]\, D) \subseteq D\ ,$$

thus, letting $z = \mathsf{free}(D)$:

$$\mathcal{U}(N);\ \ H_1;\ \ \ldots;\ \ H_h;\ \ \mathsf{BA}(T_{N|z}) \vdash ([S_{N|z}][(w_{T_N} := w'_{T_N})_{|z}]\, D) \subseteq D\ ,$$

i.e. REF_ANT_VAR_$\mathbb{P}$. $\square$

Theorem 28

$$\mathcal{U}(N);\ \ (\exists\, t^N \cdot H_1 \wedge \ldots \wedge H_h);\ \ (\forall\, t^N \cdot H_1 \wedge \ldots \wedge H_h \Rightarrow \mathsf{BA}_w(R_N)) \vdash ([w := w']\, D) \subset D$$

Proof: Following the same steps as in the proof of Theorem 27 we obtain:

$$\mathcal{U}(N);\ \ H_1;\ \ \ldots;\ \ H_h;\ \ \mathsf{BA}(T_{N|z}) \vdash ([S_{N|z}][(w_{T_N} := w'_{T_N})_{|z}]\, D) \subset D\ ,$$

i.e. REF_CVG_VAR_$\mathbb{P}$, where $z = \mathsf{free}(D)$. $\square$

4.2.14 Reduction of a Natural Number Variant

In the case when the variant can be expressed as a number specialised proof obligations can be used. If $D_{\mathbb{Z}}$ describes an integer number, then $0 \,..\, D_{\mathbb{Z}}$ is a set. So, all we have to do is to state the equivalents of Theorems 26 to 28 for natural numbers.

Theorem 29

$$\mathcal{U}(N) \vdash \text{finite}(0 \,..\, D_{\mathbb{Z}})$$

Proof: The set $0 \,..\, D_{\mathbb{Z}}$ is finite. □

Theorem 30

$$\begin{array}{l}\mathcal{U}(N);\ \ (\exists\, t^N \cdot H_1 \wedge \ldots \wedge H_h);\ \ (\forall\, t^N \cdot H_1 \wedge \ldots \wedge H_h \Rightarrow \mathsf{BA}_w(R_N)) \vdash \\ \qquad ([w := w']\, 0 \,..\, D_{\mathbb{Z}}) \subseteq 0 \,..\, D_{\mathbb{Z}}\end{array}$$

Proof: We obtain (see Theorem 27):

$$\mathcal{U}(N);\ \ H_1;\ \ \ldots;\ \ H_h;\ \ \mathsf{BA}(T_N) \vdash ([S_N][w_{T_N} := w'_{T_N}]\, 0 \,..\, D_{\mathbb{Z}}) \subseteq 0 \,..\, D_{\mathbb{Z}}\ .$$

The consequent can be expressed equivalently:

$$\begin{array}{ll}\mathcal{U}(N);\ \ H_1;\ \ \ldots;\ \ H_h;\ \ \mathsf{BA}(T_N) \vdash & \\ \qquad D_{\mathbb{Z}} \in \mathbb{N} \wedge & (1) \\ \qquad ([S_N][w_{T_N} := w'_{T_N}]\, D_{\mathbb{Z}}) \leq D_{\mathbb{Z}}\ . & (2)\end{array}$$

Letting $z = \mathsf{free}(D_{\mathbb{Z}})$ the first sequent becomes

$$\mathcal{U}(N);\ \ H_1;\ \ \ldots;\ \ H_h \vdash D_{\mathbb{Z}} \in \mathbb{N}\ ,$$

i.e. REF_ANT_VAR_$\mathbb{N}$ and the second sequent:

$$\mathcal{U}(N);\ \ H_1;\ \ \ldots;\ \ H_h;\ \ \mathsf{BA}(T_{N|z}) \vdash ([S_{N|z}][(w_{T_N} := w'_{T_N})_{|z}]\, D_{\mathbb{Z}}) \leq D_{\mathbb{Z}}\ ,$$

i.e. REF_ANT_VAR_Δ. □

Theorem 31

$$\begin{array}{l}\mathcal{U}(N);\ \ (\exists\, t^N \cdot H_1 \wedge \ldots \wedge H_h);\ \ (\forall\, t^N \cdot H_1 \wedge \ldots \wedge H_h \Rightarrow \mathsf{BA}_w(R_N)) \vdash \\ \qquad ([w := w']\, 0 \,..\, D_{\mathbb{Z}}) \subset 0 \,..\, D_{\mathbb{Z}}\end{array}$$

Proof: We proceed as in the proof of Theorem 30 and with $z = \mathsf{free}(D_{\mathbb{Z}})$ obtain the sequents:

$$\mathcal{U}(N);\ \ H_1;\ \ \ldots;\ \ H_h \vdash D_{\mathbb{Z}} \in \mathbb{N}\ ,$$

i.e. REF_CVG_VAR_$\mathbb{N}$, and:

$$\mathcal{U}(N);\ \ H_1;\ \ \ldots;\ \ H_h;\ \ \mathsf{BA}(T_{N|z}) \vdash ([S_{N|z}][(w_{T_N} := w'_{T_N})_{|z}]\, D_{\mathbb{Z}}) \leq D_{\mathbb{Z}}$$

i.e. REF_CVG_VAR_Δ. □

4.2.15 Introduction of New Events

Ordinary Events. If an ordinary event is introduced we must prove Theorem 25.

Anticipated Events. If an anticipated event is introduced we must prove Theorem 25 and that the event does not increase the variant. If there are no convergent events either by refinement or introduction, there is no variant for the model and, hence, there is nothing to prove. In the other case Theorem 26 and Theorem 27 must hold (or alternatively only Theorem 30).

Convergent Events. If a convergent event is introduced we must prove Theorem 25 and that the event decreases the variant. I.e. we must also prove Theorem 26 and Theorem 28 must hold (or alternatively only Theorem 31).

4.2.16 Refinement of Events

External Events. External events can neither be anticipated nor convergent. They must, however, not have a stronger guard or be less deterministic. We must prove Theorem 20 and Theorem 24.

Ordinary Events. If the refined event is ordinary we must prove Theorem 20 or Theorem 21.

Anticipated Events. If the refined event is anticipated we must prove Theorem 20 or Theorem 21, and that the event does not increase the variant (if there is a variant in the refined model). If there is a variant we must also prove Theorem 26 and Theorem 27 (or alternatively only Theorem 30).

Convergent Events. If the refined event is convergent we must prove Theorem 20 or Theorem 21, and that the event decreases the variant. I.e. we must also prove Theorem 26 and Theorem 28 must hold (or alternatively only Theorem 31).

4.2.17 Relative Deadlock-Freedom

We must prove that the disjunction of the guards of the internal events of the refined model implies the disjunction of the guards of the internal events of the abstract model. Let $e_1^N, \ldots, e_\ell^N$ be the internal events of the refined model, and $e_1^M, \ldots, e_k^M$ be the internal events of the abstract model.

Theorem 32

$$\mathcal{U}(M);\ \mathsf{GD}(e_1^N) \vee \ldots \vee \mathsf{GD}(e_\ell^N) \vdash \mathsf{GD}(e_1^M) \vee \ldots \vee \mathsf{GD}(e_k^M)$$

Proof: By REF_DLK. □

4.3 Generated Proof Obligations

4.3.1 Well-definedness of Invariants

Proof Obligation: REF_INV_WD

FOR	**invariant** I_ℓ of N WHERE
	$\ell \in 1 \,..\, m$
ID	"$\mathit{REF}/\mathit{INV}_\ell/\mathbf{WD}$"
GPO	$\mathcal{Q}(C);\ \mathcal{I}(M);\ I_1;\ \ldots;\ I_{\ell-1} \vdash \mathsf{WD}(I_\ell)$

Proof of WDEF: Analogously to MDL_INV_WD. □

Remark. REF_INV_WD is identical to MDL_INV_WD (3.3.1 on page 18) apart from renaming.

Remark. See remarks on MDL_INV_WD.

4.3.2 Well-definedness of Theorems

Proof Obligation: REF_THM_WD

FOR	**theorem** Q_ℓ of N WHERE
	$\ell \in 1 \,..\, n$
ID	"$\mathit{REF}/\mathit{THM}_\ell/\mathbf{WD}$"
GPO	$\mathcal{Q}(C);\ \mathcal{I}(M);\ \mathcal{J}(N);\ Q_1;\ \ldots;\ Q_{\ell-1} \vdash \mathsf{WD}(Q_\ell)$

Proof of WDEF: Analogously to MDL_THM_WD. □

Remark. REF_THM_WD is identical to MDL_THM_WD (3.3.2 on page 19) apart from renaming.

Remark. See remarks on MDL_THM_WD.

4.3.3 Model Theorems

Proof Obligation: REF_THM

FOR	**theorem** Q_ℓ of N WHERE
	$\ell \in 1\,..\,n$
ID	"$\mathit{REF}/\mathit{THM}_\ell/$**THM**"
GPO	$\mathcal{Q}(C);\ \mathcal{I}(M);\ \mathcal{J}(N);\ Q_1;\ \ldots;\ Q_{\ell-1} \vdash Q_\ell$

Proof of WDEF: Analogously to MDL_THM. □

Remark. REF_THM is identical to MDL_THM (3.3.3 on page 19) apart from renaming.

Remark. See remarks on MDL_THM.

4.3.4 Functional External Invariant

Proof Obligation: REF_EXT_FUN

FOR	**external invariants** $J_1, .., J_\sigma$ of N WHERE
	$\top$
ID	"$\mathit{REF}/$**EXT/FUN**"
GPO	$\mathcal{Q}(C);\ [\overset{\times}{\mathrm{x}} := \overset{\times}{\mathrm{x}}]\, J_1;\ \ldots;\ [\overset{\times}{\mathrm{x}} := \overset{\times}{\mathrm{x}}]\, J_\sigma;\ [\overset{\times}{\mathrm{x}} := \overset{\times}{\mathrm{x}}']\, J_1;\ \ldots;\ [\overset{\times}{\mathrm{x}} := \overset{\times}{\mathrm{x}}']\, J_\sigma \vdash \overset{\times}{\mathrm{x}} = \overset{\times}{\mathrm{x}}'$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{Q}(C)$, and $J_1, \ldots, J_\sigma$ before by REF_INV_WD. □

4.3.5 Total External Invariant

Proof Obligation: REF_EXT_TOT

FOR	**external invariants** $J_1, .., J_\sigma$ of N WHERE
	$\top$
ID	"$\mathit{REF}/$**EXT/TOT**"
GPO	$\mathcal{Q}(C) \vdash \forall \overset{\times}{\mathrm{x}} \cdot \exists \overset{\times}{\mathrm{y}} \cdot J_1 \wedge \ldots \wedge J_\sigma$

Proof of WDEF: Similarly to REF_EXT_FUN. □

4.3.6 Surjective External Invariant

Proof Obligation: REF_EXT_SRJ

FOR	**external invariants** $J_1, .., J_\sigma$ of N WHERE
	$\top$
ID	"*REF*/**EXT**/**SRJ**"
GPO	$\mathcal{Q}(C) \vdash \forall \check{y} \cdot \exists \check{x} \cdot J_1 \wedge \ldots \wedge J_\sigma$

Proof of WDEF: Similarly to REF_EXT_FUN. □

4.3.7 Well-definedness of Initialisation

Proof Obligation: REF_INI_WD

FOR	**substitution** R_ℓ of the **combined initialisation** of N WHERE	
	$\ell \in 1 \,..\, n$ AND $u_\ell = \mathsf{frame}(R_\ell)$	
ID	"*REF*/**INIT**/u_ℓ/**WD**"	
GPO	$\top$	(if $R_\ell \sim \mathsf{skip}$)
GPO	$\mathcal{Q}(C) \vdash \mathsf{WD}(E_\ell)$	(if $R_\ell \sim u_\ell := E_\ell$)
GPO	$\mathcal{Q}(C) \vdash \mathsf{WD}(E_\ell)$	(if $R_\ell \sim u_\ell :\in E_\ell$)
GPO	$\mathcal{Q}(C) \vdash \mathsf{WD}(A_\ell)$	(if $R_\ell \sim u_\ell :\mid A_\ell$)

Proof of WDEF: Analogously to MDL_INI_WD. □

Remark. REF_INI_WD is identical to MDL_INI_WD (3.3.4 on page 19) apart from renaming.

Remark. See remarks on MDL_INI_WD.

4.3.8 Feasibility of Initialisation

Proof Obligation: REF_INI_FIS

FOR	**substitution** R_ℓ of the **combined initialisation** of N WHERE	
	$\ell \in 1 \mathinner{..} n$ AND $u_\ell = \mathsf{frame}(R_\ell)$	
ID	"$\mathit{REF}/\mathbf{INIT}/u_\ell/\mathbf{FIS}$"	
GPO	$\top$	(if $R_\ell \sim \mathsf{skip}$)
GPO	$\top$	(if $R_\ell \sim u_\ell := E_\ell$)
GPO	$\mathcal{Q}(C) \vdash E_\ell \neq \varnothing$	(if $R_\ell \sim u_\ell :\in E_\ell$)
GPO	$\mathcal{Q}(C) \vdash \exists\, u'_\ell \cdot A_\ell$	(if $R_\ell \sim u_\ell :\mid A_\ell$)

Proof of WDEF: Analogously to MDL_INI_FIS. □

Remark. REF_INI_FIS is identical to MDL_INI_FIS (3.3.5 on page 20) apart from renaming.

Remark. See remarks on MDL_INI_FIS.

4.3.9 Simulation of Initialisation

Proof Obligation: REF_INI_SIM

FOR	**combined initialisation** of N and **combined initialisation** of M WHERE
	$\ell \in 1 \mathinner{..} p$ AND $R_{M_\ell} \notin S_{M\mid x}$ AND $f = \mathsf{frame}(R_{M_\ell})$ AND $z = \mathsf{primed}(W_{x\mid f})$
ID	"$\mathit{REF}/\mathbf{INIT}/u/\mathbf{SIM}$"
GPO	$\mathcal{Q}(C);\ \mathsf{BA}(T_{N\mid f\cup z}) \vdash [S'_{N\mid f\cup z}]\,[W'_{x\mid f}]\,\mathsf{BA}(R_{M_\ell})$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{Q}(C)$, and T_N and S_N by REF_INI_WD, and combined witness W_x by REF_GWIT_WD, and R_{M_ℓ} by MDL_INI_WD/REF_INI_WD. □

Remark. This proof obligation should only be generated when the initialisations, external and internal, of the models M and N are well-formed and well-typed. It should be avoided to present the user with proof obligations that may not be stable.

Remark. Note also, that the initialisation of a model must assign values to variables of that model. This means there no variables outside its frame.

4.3.10 Unreduced External Initialisation

Proof Obligation: REF_INI_EXT

FOR **subst.** R_{N_ℓ} of **ext. initialisation** of N and **ext. initialisation** of M WHERE

$\ell \in 1 \mathinner{..} q$ AND $f = \mathsf{frame}(R_{N_\ell})$

ID "$\mathit{REF}/\mathbf{INIT}/f/\mathbf{EXT}$"

GPO $\mathcal{Q}(C);\ \mathsf{BA}(T_{M|x\cup f});$

$[S_{M|x}]\,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}]\,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}']\,J_1;\ \ldots;\ [S_{M|x}]\,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}]\,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}']\,J_\sigma \vdash$

$[S_{M|f}]\,\mathsf{BA}(R_{N_\ell})$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{Q}(C)$, and $J_1 \ldots J_\sigma$ by REF_INV_WD, and substitution R_{N_ℓ} by REF_INI_WD, and S_M and T_M by MDL_INI_WD/REF_INI_WD. □

4.3.11 Invariant Establishment

Proof Obligation: REF_INI_INV

FOR **combined initialisation** of N and **invariant** I_ℓ of N WHERE

$\ell \in 1 \mathinner{..} i$ AND $z = \mathsf{free}(I_\ell)$ AND $\theta = \mathsf{primed}(W_{x|z}) \cup \mathsf{primed}(S_{M|x\cap z})$

ID "$\mathit{REF}/\mathbf{INIT}/\mathit{INV}_\ell/\mathbf{INV}$"

GPO $\mathcal{Q}(C);\ \mathsf{BA}(T_{N|\theta\cup z}) \vdash [S'_{N|\theta\cup z}]\,[W_{x|z}]\,[S_{M|x\cap z}]\,[(w_{R_N} := w'_{R_N})_{|z}]\,I_\ell$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{Q}(C)$, and T_N and S_N by REF_INI_WD, and W_x by REF_GWIT_WD, and invariant I_ℓ by REF_INV_WD. □

4.3.12 Well-definedness of Guards

Proof Obligation: REF_GRD_WD

FOR **guard** H_ℓ of **event** e^N of N WHERE

$\ell \in 1 \mathinner{..} h$

ID "$\mathit{REF}/\mathit{EVT}/\mathit{GRN}_\ell/\mathbf{WD}$"

PO $\mathcal{U}(N);\ H_1;\ \ldots;\ H_{\ell-1} \vdash \mathsf{WD}(H_\ell)$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and $H_1, \ldots, H_{\ell-1}$ before by REF_GRD_WD, and $t_1^N, \ldots, t_j^N$ **nfin** $\mathcal{U}(N)$ by Theorem 7. $\square$

Remark. REF_GRD_WD is identical to MDL_GRD_WD (3.3.7 on page 21) apart from renaming.

4.3.13 Well-definedness of Local Witnesses

Remark. There are two kinds of local witnesses: witnesses for local variables of the abstract event, and for external events also witnesses for local variables of the refined event.

Proof Obligation: REF_LWIT_WD_A

FOR	**witness** $W_{t_\ell^M}$ of **event** e^N of N WHERE
	$\ell \in 1\,..\,i$ AND $W_{t_\ell^M} \sim t_\ell^M := E$
ID	"$\mathit{REF/EVT}/t_\ell^M/\mathbf{WWD}$"
GPO	$\mathcal{U}(N);\ H_1;\ \ldots;\ H_h \vdash \mathsf{WD}(E)$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and $H_1, \ldots, H_h$ before by REF_GRD_WD, and $t_1^N, \ldots, t_j^N$ **nfin** $\mathcal{U}(N)$ by Theorem 7. $\square$

Remark. This proof obligation does not apply to index ℓ for $t_\ell^M \in t^N$ because it is not possible to specify explicit witnesses for local variables for which default witnesses are used.

Proof Obligation: REF_LWIT_WD_R

FOR	**witness** $W_{t_\ell^N}$ of **event** e^N of N WHERE
	$\ell \in 1\,..\,i$ AND $W_{t_\ell^N} \sim t_\ell^N := E$
ID	"$\mathit{REF/EVT}/t_\ell^N/\mathbf{WWD}$"
GPO	$\mathcal{Q}(C);\ J_1;\ \ldots;\ J_\sigma;\ G_1;\ \ldots;\ G_g \vdash \mathsf{WD}(E)$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{Q}(C)$, and the external invariants $J_1, \ldots, J_\sigma$ by REF_INV_WD, and the guards $G_1, \ldots, G_g$ by MDL_GRD_WD/REF_GRD_WD, and $t_1^M, \ldots, t_j^M$ **nfin** $\mathcal{U}(N)$ by Theorem 7. $\square$

Remark. This proof obligation does not apply to index ℓ for $t^N_\ell \in t^M$ because it is not possible to specify explicit witnesses for local variables for which default witnesses are used.

4.3.14 Well-definedness of Global Witnesses of Events

Proof Obligation: REF_GWIT_WD

FOR	**witness** W_u of **event** e^N of N WHERE
	$W_u \sim u := E$ AND $z = \mathsf{primed}(E)$
ID	"$REF/EVT/u/$**WWD**"
GPO	$\mathcal{U}(N);\ H_1;\ \ldots;\ H_h \vdash \mathsf{BA}(T_{N\|z}) \Rightarrow [S'_{N\|z}]\,\mathsf{WD}(E)$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and the guards $H_1, \ldots, H_h$ by REF_GRD_WD, and T_N and S_N by REF_EVT_WD, and $t^N_1, \ldots, t^N_j$ **nfin** $\mathcal{U}(N)$ by Theorem 7. □

4.3.15 Guard Strengthening (Split Case)

Proof Obligation: REF_GRD_REF

FOR	**event** e^N of N and **guard** G_ℓ of **event** e^M of M WHERE
	$\ell \in 1 \mathinner{..} g$ AND $z = \mathsf{free}(G_\ell)$ AND $\psi = \mathsf{primed}(V_{t^M\|z})$
ID	"$REF/EVT/GRM_\ell/$**REF**"
GPO	$\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(T_{N\|\psi}) \vdash [S'_{N\|\psi}]\,[(w'_{\Xi_N} := w_{\Xi_N})_{\|\psi}]\,[V_{t^M\|z}]\ G_\ell$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and $H_1, \ldots, H_h$ by REF_GRD_WD, and that of S_N and T_N by REF_EVT_WD, and V_{t^M} by REF_LWIT_WD_A, and guard G_ℓ by MDL_GRD_WD/REF_GRD_WD, and $t^N_1, \ldots, t^N_j$ **nfin** $\mathcal{U}(N)$ by Theorem 7. □

4.3.16 Guard Weakening of External Events

Proof Obligation: REF_GRD_EXT

FOR **guard** H_ℓ of **external event** e^N of N and **external event** e^M of M WHERE

$\ell \in 1 \mathinner{..} h$ AND $z = \mathsf{free}(H_\ell)$ AND $\psi = \mathsf{primed}(V_{t^N|z})$

ID "$REF/EVT/GRN_\ell/$**EXT**"

GPO $\mathcal{Q}(C)$; J_1; $\ldots$; J_σ; G_1; $\ldots$; G_g; $\mathsf{BA}(T_{M|x\cup\psi})$;

$$[S_{M|x}]\,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}]\,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}']\,J_1;\ \ldots;\ [S_{M|x}]\,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}]\,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}']\,J_\sigma \vdash$$

$$[S'_{M|\psi}]\,[(\overset{\times}{\mathrm{v}}'_{\Xi_M} := \overset{\times}{\mathrm{v}}_{\Xi_M})_{|\psi}]\,[V_{t^N|z}]\,H_\ell$$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and H_ℓ by REF_GRD_WD, and V_{t^N} by REF_LWIT_WD_R, and the guards $G_1 \ldots G_g$ by MDL_GRD_WD/REF_GRD_WD, and S_M and T_M by MDL_EVT_WD/REF_EVT_WD, and $t_1^M, \ldots, t_i^M$ **nfin** $\mathcal{U}(N)$ by Theorem 7. □

Remark. This proof obligation applies to all external events of a model. In conjunction with REF_GRD_REF it shows that the guards of an external event and the corresponding refined event are equivalent.

Remark. External events can neither be split nor be merged. The proof obligation that applies is that for the split case (where the abstract event is split into only one event).

Remark. The combined witnesses V_{t^M} and V_{t^N} are used for both proof obligations concerning guards REF_GRD_REF and REF_GRD_EXT. This is possible because identically named local variables u must denote the same objects. They are associated with default witnesses of the form $u := u$. These are applied in both directions. For the remaining variables with distinct names it is clear for which proof obligation they are to be applied because they only occur either in the guard of the abstract event or in the guard of the refined event.

4.3.17 Guard Strengthening (Merge Case)

Proof Obligation: REF_GRD_MRG

FOR	**event** e^N of N and **events** $e^M_1, \ldots, e^M_k$ of M WHERE
	$\psi = \mathsf{primed}(V_{t^M})$
ID	"*REF*/*EVT*/**MRG**"
GPO	$\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(T_{N\vert\psi}) \vdash$ $[S'_{N\vert\psi}]\,[(w'_{\Xi_N} := w_{\Xi_N})_{\vert\psi}]\,[V_{t^M}]\,((G_{1,1} \wedge \ldots \wedge G_{1,g_1}) \vee \ldots \vee (G_{k,1} \wedge \ldots \wedge G_{k,g_k}))$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and $H_1, \ldots, H_h$ by REF_GRD_WD, and that of S_N and T_N by REF_EVT_WD, and the combined witness V_{t^M} by REF_LWIT_WD_A, and $G_{1,1} \ldots G_{1,g_1} \ldots G_{k,1} \ldots G_{k,g_k}$ by MDL_GRD_WD/REF_GRD_WD, and $t^N_1, \ldots, t^N_j$ **nfin** $\mathcal{U}(N)$ by Theorem 7. □

Remark. Unfortunately this proof obligation cannot be further decomposed.

4.3.18 Well-definedness of Event Actions

Proof Obligation: REF_EVT_WD

FOR	**substitution** R_ℓ of **event** e^N of N WHERE	
	$\ell \in 1\,..\,n$ AND $u_\ell = \mathsf{frame}(R_\ell)$	
ID	"*REF*/*EVT*/u_ℓ/**WD**"	
GPO	$\top$	(if $R_\ell \sim \mathsf{skip}$)
GPO	$\mathcal{U}(N);\ H_1;\ \ldots;\ H_h \vdash \mathsf{WD}(E_\ell)$	(if $R_\ell \sim u_\ell := E_\ell$)
GPO	$\mathcal{U}(N);\ H_1;\ \ldots;\ H_h \vdash \mathsf{WD}(E_\ell)$	(if $R_\ell \sim u_\ell :\in E_\ell$)
GPO	$\mathcal{U}(N);\ H_1;\ \ldots;\ H_h \vdash \mathsf{WD}(A_\ell)$	(if $R_\ell \sim u_\ell :\mid A_\ell$)

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and $H_1 \ldots H_h$ before by REF_GRD_WD, and $t_1, \ldots t_j$ **nfin** $\mathcal{U}(N)$ by Theorem 7. □

Remark. REF_EVT_WD is identical to MDL_EVT_WD (3.3.8 on page 21) apart from renaming.

4.3.19 Feasibility of Event Actions

Proof Obligation: REF_EVT_FIS

FOR	**substitution** R_ℓ of **event** e^N of N WHERE
	$\ell \in 1 \mathbin{..} n$ AND $u_\ell = \mathsf{frame}(R_\ell)$
ID	"$REF/EVT/u_\ell/$**FIS**"
GPO	$\top$ (if $R_\ell \sim \mathsf{skip}$)
GPO	$\top$ (if $R_\ell \sim u_\ell := E_\ell$)
GPO	$\mathcal{U}(N);\; H_1;\; \ldots;\; H_h \vdash E_\ell \neq \varnothing$ (if $R_\ell \sim u_\ell :\in E_\ell$)
GPO	$\mathcal{U}(N);\; H_1;\; \ldots;\; H_h \vdash \exists\, u'_\ell \cdot A_\ell$ (if $R_\ell \sim u_\ell :\mid A_\ell$)

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and $H_1 \ldots H_h$ has be shown by REF_GRD_WD, and that of E_ℓ (respectively A_ℓ) by REF_EVT_WD, and $t_1, \ldots t_j$ **nfin** $\mathcal{U}(N)$ by Theorem 7. □

Remark. REF_EVT_FIS is identical to MDL_EVT_FIS (3.3.9 on page 22) apart from renaming.

4.3.20 Simulation of Refined-Event Actions

Remark. There are two cases of simulation to be treated as indicated in the proof obligations REF_EVT_SIM_(Δ/Ξ) by underlining the corresponding conditions. This happens because an event behaves like skip on variables that are not in its frame. For each event, the generated simulation proof obligations must cover all abstract variables v.

Proof Obligation: REF_EVT_SIM_$\boldsymbol{\Delta}$

FOR	**refined event** e^N of N and **substitution** R_{M_ℓ} of **event** e^M of M WHERE
	$\ell \in 1 \mathbin{..} p$ AND $R_{M_\ell} \notin S_{M\mid x}$ AND
	$f = \mathsf{frame}(R_{M_\ell})$ AND $\psi = \mathsf{free}(R_{M_\ell})$ AND $\chi = \mathsf{primed}(W_{x\mid f})$
ID	"$REF/EVT/u/$**SIM**"
GPO	$\mathcal{U}(N);\; H_1;\; \ldots;\; H_h;\; \mathsf{BA}(T_{N\mid f\cup\chi}) \vdash$ $[S'_{N\mid f\cup\chi}]\,[(w'_{\Xi_N} := w_{\Xi_N})_{f\cup\chi}]\,[V_{t^M\mid\psi}]\,[W'_{x\mid f}]\,\mathsf{BA}(R_{M_\ell})$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and $H_1 \dots H_h$ has be shown by REF_GRD_WD, and that of substitutions S_N and T_N by REF_EVT_WD, , and that of R_{M_ℓ} by MDL_EVT_WD/REF_EVT_WD, and W_x by REF_GWIT_WD, and $t_1, \dots t_j$ **nfin** $\mathcal{U}(N)$ by Theorem 7. $\square$

Remark. We have the choice to add either proved invariant preservation as lemmas to the antecedent of this generated proof obligation, or the simulations as lemmas to the antecedents of the invariant preservation proof obligations REF_EVT_INV. We have decided for the second choice because, empirically, the simulation proof obligation is usually straightforward whereas invariant preservation proofs are more difficult and profit from the addition antecedents. See the remarks on REF_EVT_INV.

Split GPO. In case of a split refinement we can add some useful additional hypotheses to REF_EVT_SIM_Δ, assuming that REF_GRD_REF (Theorem 18) has been proven as a lemma (for all G_ℓ):

$$\begin{aligned}&\mathcal{U}(N);\ [V_{t^M|\theta_1}]\,G_1;\ \dots;\ [V_{t^M|\theta_g}]\,G_g;\ H_1;\ \dots;\ H_h;\ \mathsf{BA}(T_{N|f\cup\chi}) \vdash\\ &\qquad [S'_{N|f\cup\chi}]\,[(w'_{\Xi_N} := w_{\Xi_N})_{f\cup\chi}]\,[V_{t^M|\psi}]\,[W'_{x|f}]\,\mathsf{BA}(R_{M_\ell})\end{aligned}$$

where $\theta_\ell = \mathsf{free}(G_\ell)$ for $\ell \in 1 \,..\, g$. This is still well-defined because we have shown well-definedness of $G_1, \dots, G_g$ has be shown by MDL_GRD_WD/REF_GRD_WD, and V_{t^M} by REF_LWIT_WD_A.

Merge GPO. In case of a merge refinement we can add some useful additional hypotheses to REF_EVT_SIM_Δ, assuming that REF_GRD_MRG (Theorem 19) has been proven as a lemma:

$$\begin{aligned}&\mathcal{U}(N);\\ &[V_{t^M}]\,((G_{1,1} \wedge \dots \wedge G_{1,g_1}) \vee \dots \vee (G_{k,1} \wedge \dots \wedge G_{k,g_k}));\\ &H_1;\ \dots;\ H_h;\ \mathsf{BA}(T_{N|f\cup\chi}) \vdash\\ &\qquad [S'_{N|f\cup\chi}]\,[(w'_{\Xi_N} := w_{\Xi_N})_{f\cup\chi}]\,[V_{t^M|\psi}]\,[W'_{x|f}]\,\mathsf{BA}(R_{M_\ell})\end{aligned}$$

This is still well-defined because we have shown well-definedness of $(G_{1,1}, \dots, G_{1,g_1})$, $\dots$, $(G_{k,1}, \dots, G_{k,g_k})$ has be shown by MDL_GRD_WD/REF_GRD_WD, and the combined witness V_{t^M} by REF_LWIT_WD_A.

Remark. There must only be global witnesses for variables that do occur in the frame of are non-deterministic assignment in the abstract action. Extra witnesses would break the correctness of REF_EVT_INV.

Proof Obligation: REF_EVT_SIM_Ξ

FOR	**refined event** e^N of N and **event** e^M of M WHERE
	$\ell \in 1 \,..\, p$ AND $u \in o \cap (\mathsf{frame}(R_N) \setminus \mathsf{frame}(R_M)$
ID	"$REF/EVT/u/$**SIM**"
GPO	$\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(T_{N\|u}) \vdash [S'_{N\|u}]\, u = u'$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and $H_1 \ldots H_h$ has been shown by REF_GRD_WD, and that of substitutions S_N and T_N by REF_EVT_WD, and $t_1, \ldots t_j$ **nfin** $\mathcal{U}(N)$ by Theorem 7. □

4.3.21 Unreduced External-Event Actions

Proof Obligation: REF_EVT_GEN_Δ

FOR	**subst.** R_{N_ℓ} **ext. event** e^N of N and **ext. event** e^M of M WHERE
	$\ell \in 1 \,..\, q$ AND $f = \mathsf{frame}(R_{N_\ell})$
ID	"$MDL/EVT/f/$**EXT**"
GPO	$\mathcal{Q}(C);\ J_1;\ \ldots;\ J_\sigma;\ G_1;\ \ldots;\ G_g;\ \mathsf{BA}(T_{M\|x\cup f});$ $[S_{M\|x}]\,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}]\,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}']\, J_1;\ \ldots;\ [S_{M\|x}]\,[\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}]\,[\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}']\, J_\sigma \vdash$ $[S'_{M\|f}]\,[(\overset{\times}{\mathrm{v}}'_{\Xi_M} := \overset{\times}{\mathrm{v}}_{\Xi_M})_{\|f}]\,[V_{t^N}]\,\mathsf{BA}(R_{N_\ell})$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and the external invariants $J_1, \ldots, J_\sigma$ by REF_INV_WD, and $G_1 \ldots G_g$ has be shown by MDL_GRD_WD/REF_GRD_WD, and V_t^N by REF_LWIT_WD_R, and that of S_M and T_M by MDL_EVT_WD/REF_EVT_WD, and that of R_{N_ℓ} by REF_EVT_WD, and $t_1^M, \ldots t_j^M$ **nfin** $\mathcal{U}(N)$ by Theorem 7. □

Proof Obligation: REF_EVT_GEN_Ξ

FOR **external event** e^N of N and **external event** e^M of M WHERE

$u \in o \cap (\mathsf{frame}(R_M) \setminus \mathsf{frame}(R_N))$

ID "$MDL/EVT/u/$**EXT**"

GPO $\mathcal{Q}(C);\; J_1;\; \ldots;\; J_\sigma;\; G_1;\; \ldots;\; G_g;\; \mathsf{BA}(T_{M|x \cup u});$

$$[S_{M|x}]\, [\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}]\, [\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}'] \, J_1;\; \ldots;\; [S_{M|x}]\, [\overset{\times}{\mathrm{x}}_{T_M} := \overset{\times}{\mathrm{x}}'_{T_M}]\, [\overset{\times}{\mathrm{y}} := \overset{\times}{\mathrm{y}}'] \, J_\sigma \vdash$$

$[S'_{M|u}]\,(u = u')$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and $G_1 \ldots G_g$ has be shown by MDL_GRD_WD/REF_GRD_WD, and that of S_M and T_M by MDL_EVT_WD/REF_EVT_WD, and $t_1^M, \ldots t_j^M$ **nfin** $\mathcal{U}(N)$ by Theorem 7. □

4.3.22 Invariant Preservation of Refined-Event Actions

Proof Obligation: REF_EVT_INV

FOR **refined event** e^N of N and **event** e^M of M and **invariant** I_ℓ of N WHERE

$\ell \in 1 \mathinner{..} m$ AND

$z = \mathsf{free}(I_\ell)$ AND $\phi = \mathsf{free}(S_{M|x \cap z})$ AND $\eta = \mathsf{primed}(W_{x|z}) \cup \mathsf{primed}(S_{M|x \cap z})$

ID "$REF/EVT/INV_\ell/$**INV**"

GPO $\mathcal{U}(N);\; H_1;\; \ldots;\; H_h;\; \mathsf{BA}(T_{N|\eta \cap z}) \vdash$

$$[S'_{N|\eta \cap z}]\, [(w'_{\Xi_N} := w_{\Xi_N})_{|\eta \cap z}]\, [V_{t^M|\phi}]\, [W_{x|z}]\, [S_{M|x \cap z}]\, [(w_{R_N} := w_{R_N})'_{|z}]\, I_\ell$$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$ and V_{t^M} by REF_LWIT_WD_A, and S_N and T_N by REF_EVT_WD, and W_x by REF_GWIT_WD, and I_ℓ by REF_INV_WD, and $H_1 \ldots H_h$ has be shown by REF_GRD_WD, and $t_1^N, \ldots t_j^N$ **nfin** $\mathcal{U}(N)$ by Theorem 7. □

Remark. If $R_{N|z}$ is the empty multiple substitution and $z \cap x$ is empty, this proof obligation should not be generated because I_ℓ would appear in the antecedent and in the consequent.

Remark. The frame of the combined witness W_x must not be larger than x_{R_M}.

Remark. We can add additional hypotheses to proof obligation REF_EVT_INV, assuming that REF_EVT_SIM_Δ has been proven as a lemma (for all $R_{M_k} \notin S_{M|x}$):

$$[S'_{N|f\cup\chi}]\,[(w'_{\Xi_N} := w_{\Xi_N})_{f\cup\chi}]\,[V_{t^M|\psi}]\,[W'_{x|f}]\,\mathsf{BA}(R_{M_k})\ .$$

This is still valid if the option **Split GPO** or **Merge GPO** has been used to for proof obligation REF_EVT_SIM_Δ. This corresponds to an application of the cut rule. Furthermore, these hypotheses can be add in addition to those suggested in the options **Split GPO** or **Merge GPO** for this proof obligation.

Split GPO. In case of a split refinement we can add some useful additional hypotheses to REF_EVT_INV, assuming that REF_GRD_REF (Theorem 18) has been proven as a lemma (for all G_ℓ):

$$\begin{aligned}&\mathcal{U}(N);\ [V_{t^M|\theta_1}]\,G_1;\ \ldots;\ [V_{t^M|\theta_g}]\,G_g;\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(T_{N|\eta\cap z}) \vdash\\ &\qquad [S'_{N|\eta\cap z}]\,[(w'_{\Xi_N} := w_{\Xi_N})_{|\eta\cap z}]\,[V_{t^M|\phi}]\,[W'_{x|z}]\,[S'_{M|x\cap z}]\,[(w_{R_N} := w_{R_N})'_{|z}]\,I_\ell\end{aligned}$$

where $\theta_\ell = \mathsf{free}(G_\ell)$ for $\ell \in 1 \,..\, g$. This still well-defined because well-definedness of $G_1 \ldots G_g$ has be shown by MDL_GRD_WD/REF_GRD_WD and V_{t^M} by REF_LWIT_WD_A.

Merge GPO. In case of a merge refinement we can add some useful additional hypotheses to REF_EVT_INV, assuming that REF_GRD_MRG (Theorem 19) has been proven as a lemma:

$$\begin{aligned}&\mathcal{U}(N);\\ &[V_{t^M}]\,((G_{1,1} \wedge \ldots \wedge G_{1,g_1}) \vee \ldots \vee (G_{k,1} \wedge \ldots \wedge G_{k,g_k}));\\ &H_1;\ \ldots;\ H_h;\ \mathsf{BA}(T_{N|\eta\cap z}) \vdash\\ &\qquad [S'_{N|\eta\cap z}]\,[(w'_{\Xi_N} := w_{\Xi_N})_{|\eta\cap z}]\,[V_{t^M|\phi}]\,[W'_{x|z}]\,[S'_{M|x\cap z}]\,[(w_{R_N} := w_{R_N})'_{|z}]\,I_\ell\end{aligned}$$

This is still well-defined because we have shown well-definedness of $(G_{1,1}, \ldots, G_{1,g_1})$, …, $(G_{k,1}, \ldots, G_{k,g_k})$ has be shown by MDL_GRD_WD/REF_GRD_WD, and the combined witness V_{t^M} by REF_LWIT_WD_A.

4.3.23 Simulation of New-Event Actions

Proof Obligation: REF_NEW_SIM

FOR	**new event** e^N of N WHERE
	$\ell \in 1 \,..\, p$ AND $u \in \mathsf{frame}(R_N) \cap o$
ID	"$REF/EVT/u/$**SIM**"
GPO	$\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(T_{N\|u}) \vdash [S'_{N\|u}]\,u' = u$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and $H_1 \ldots H_h$ has be shown by REF_GRD_WD, and that of substitutions S_N and T_N by REF_EVT_WD, and $t_1, \ldots t_j$ **nfin** $\mathcal{U}(N)$ by Theorem 7. □

Remark. This is a simplified variant of of REF_EVT_SIM_Ξ, where we have used the fact that a new event refines skip, i.e. the abstract event has the guard $\top$ and the action skip, and $\mathsf{frame}(\mathsf{skip})$ is empty.

4.3.24 Invariant Preservation of New-Event Actions

Proof Obligation: REF_NEW_INV

FOR	**new event** e^N of N and **invariant** I_ℓ of N WHERE
	$\ell \in 1 \mathinner{..} m$ AND $z = \mathsf{free}(I_\ell)$
ID	"$REF/EVT/INV_\ell/$**INV**"
GPO	$\mathcal{U}(N);\ H_1;\ \ldots;\ H_h;\ \mathsf{BA}(T_{N\|z}) \vdash [S_{N\|z}]\,[(w_{T_N} := w'_{T_N})_{\|z}]\,I_\ell$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and S_N and T_N by REF_EVT_WD, and I_ℓ by REF_INV_WD, and $H_1 \ldots H_h$ has be shown by REF_GRD_WD, and $t_1, \ldots t_j$ **nfin** $\mathcal{U}(N)$ by Theorem 7. □

Remark. If $R_{N|z}$ is the empty multiple substitution, this proof obligation should not be generated because I_ℓ would appear in the antecedent and in the consequent.

4.3.25 Well-definedness of the Variant

Proof Obligation: REF_VAR_WD

FOR	**variant** D of N WHERE
	$\top$
ID	"$REF/$**VWD**"
GPO	$\mathcal{U}(N) \vdash \mathsf{WD}(D)$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$. □

4.3.26 Well-foundedness of the (Set) Variant

Proof Obligation: REF_VAR_FIN_$\mathbb{P}$

FOR	**variant** D of N WHERE
	$\top$
ID	"REF/**VFIN**"
GPO	$\mathcal{U}(N) \vdash \text{finite}(D)$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and D has been shown by REF_VAR_WD. $\square$

4.3.27 Strong (Set) Variant

Proof Obligation: REF_CVG_VAR_$\mathbb{P}$

FOR	**variant** of N and **event** e^N of N WHERE
	$z = \mathsf{free}(D)$
ID	"REF/EVT/**VAR**"
GPO	$\mathcal{U}(M);\ H_1;\ \ldots;\ H_h \vdash \mathsf{BA}(T_{N\|z}) \Rightarrow ([S_{N\|z}]\,[w_{T_{N\|z}} := w'_{T_{N\|z}}]\,D) \subset D$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and $H_1 \ldots H_h$ has be shown by REF_GRD_WD, and that of substitutions S_N and T_N by REF_EVT_WD, and D by REF_VAR_WD, and $t_1, \ldots t_j$ **nfin** $\mathcal{U}(N)$ by Theorem 7. $\square$

Remark. This proof obligation must be generated for each convergent event (where the variant is a set expression).

4.3.28 Strong (Natural Number) Variant

Proof Obligation: REF_CVG_VAR_$\boldsymbol{\Delta}$

FOR	**variant** of N and **event** e^N of N WHERE
	$z = \mathsf{free}(D)$
ID	"REF/EVT/**VAR**"
GPO	$\mathcal{U}(M);\ H_1;\ \ldots;\ H_h \vdash \mathsf{BA}(T_{N\|z}) \Rightarrow ([S_{N\|z}]\,[w_{T_{N\|z}} := w'_{T_{N\|z}}]\,D) < D$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and $H_1 \ldots H_h$ has be shown by REF_GRD_WD, and that of substitutions S_N and T_N by REF_EVT_WD, and D by REF_VAR_WD, and $t_1, \ldots t_j$ **nfin** $\mathcal{U}(N)$ by Theorem 7. □

Proof Obligation: REF_CVG_VAR_$\mathbb{N}$

FOR	**variant** of N and **event** e^N of N WHERE
	$\top$
ID	"*REF/EVT/***NAT**"
GPO	$\mathcal{U}(M);\ H_1;\ \ldots;\ H_h \vdash D \in \mathbb{N}$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and $H_1 \ldots H_h$ has be shown by REF_GRD_WD, and D by REF_VAR_WD, and $t_1, \ldots t_j$ **nfin** $\mathcal{U}(N)$ by Theorem 7. □

Remark. These proof obligations must be generated for each convergent event (where the variant is a set expression).

4.3.29 Weak (Set) Variant

Proof Obligation: REF_ANT_VAR_$\mathbb{P}$

FOR	**variant** of N and **event** e^N of N WHERE
	$z = \mathsf{free}(D)$
ID	"*REF/EVT/***VAR**"
PRE	$\top$
GPO	$\mathcal{U}(M);\ H_1;\ \ldots;\ H_h \vdash \mathsf{BA}(T_{N\vert z}) \Rightarrow ([S_{N\vert z}]\,[w_{T_{N\vert z}} := w'_{T_{N\vert z}}]\,D) \subseteq D$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and $H_1 \ldots H_h$ has be shown by REF_GRD_WD, and that of substitutions S_N and T_N by REF_EVT_WD, and D by REF_VAR_WD, and $t_1, \ldots t_j$ **nfin** $\mathcal{U}(N)$ by Theorem 7. □

Remark. This proof obligation must be generated for each anticipated event if the refined model has (set) variant.

4.3.30 Weak (Natural Number) Variant

Proof Obligation: REF_ANT_VAR_$\boldsymbol{\Delta}$

FOR	**variant** of N and **event** e^N of N WHERE
	$z = \mathsf{free}(D)$
ID	"$REF/EVT/$**VAR**"
PRE	$\top$
GPO	$\mathcal{U}(M);\ H_1;\ \ldots;\ H_h \vdash \mathsf{BA}(T_{N\|z}) \Rightarrow ([S_{N\|z}]\,[w_{T_{N\|z}} := w'_{T_{N\|z}}]\,D) \leq D$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and $H_1 \ldots H_h$ has be shown by REF_GRD_WD, and that of substitutions S_N and T_N by REF_EVT_WD, and D by REF_VAR_WD, and $t_1, \ldots t_j$ **nfin** $\mathcal{U}(N)$ by Theorem 7. □

Proof Obligation: REF_ANT_VAR_$\mathbb{N}$

FOR	**variant** of N and **event** e^N of N WHERE
	$\top$
ID	"$REF/EVT/$**NAT**"
PRE	$\top$
GPO	$\mathcal{U}(M);\ H_1;\ \ldots;\ H_h \vdash D \in \mathbb{N}$

Proof of WDEF: The sequent is well-defined because context abstraction and model abstraction are acyclic directed graphs, and we can assume that we have shown well-definedness of $\mathcal{U}(N)$, and $H_1 \ldots H_h$ has be shown by REF_GRD_WD, and that of substitutions S_N and T_N by REF_EVT_WD, and D by REF_VAR_WD, and $t_1, \ldots t_j$ **nfin** $\mathcal{U}(N)$ by Theorem 7. □

Remark. This proof obligation is identical to REF_CVG_VAR_$\mathbb{N}$.

Remark. This proof obligation must be generated for each anticipated event if the refined model has a (natural number) variant.

4.3.31 Deadlock-Freedom

Proof Obligation: REF_DLK

FOR	**model** M WHERE
	$e_1^N, \ldots, e_\ell^N$ are the internal events of N AND $e_1^M, \ldots, e_k^M$ the internal events of M
ID	"*REF*/**DLK**"
GPO	$\mathcal{U}(M);\ \mathsf{GD}(e_1^N) \vee \ldots \vee \mathsf{GD}(e_\ell^N) \vdash \mathsf{GD}(e_1^M) \vee \ldots \vee \mathsf{GD}(e_k^M)$

Remark. Deadlock-freedom proof obligations need only be generated for events whose guard has been changed. The two sets of events can be chosen accordingly.

Remark. One could alternatively generate the proof obligation:

$$\mathcal{U}(M);\ \neg\,\mathsf{GD}(e_2^M);\ \ldots;\ \neg\,\mathsf{GD}(e_k^M);\ \mathsf{GD}(e_1^N) \vee \ldots \vee \mathsf{GD}(e_\ell^N) \vdash \mathsf{GD}(e_1^M)$$

where event e_1^M is arbitrarily chosen.