Refinement and Continuous Behaviour

Mauno Rönkkö

Turku Centre for Computer Science,
Department of Computer Science,
Åbo Akademi University,
Lemminkäisenkatu 14A,
FIN-20520 Turku, Finland,
mronkko@abo.fi

Kaisa Sere

Department of Computer Science,
Åbo Akademi University,
Lemminkäisenkatu 14A,
FIN-20520 Turku, Finland,
Kaisa.Sere@abo.fi

Turku Centre for Computer Science
TUCS Technical Report No 198
September 1998
ISBN 952-12-0263-7
ISSN 1239-1891

Abstract

Refinement Calculus is a formal framework for the development of provably correct software. It is also used in the development of Action Systems, which is a predicate transformer based framework for constructing distributed and reactive software systems. Recently, Action Systems were extended with a new action called the differential action. It allows the modelling of continuous behaviour. Along with the differential action we may use Action Systems also to model hybrid systems.

In this paper we investigate how the use of differential action in Action Systems fits to the refinement ideology. As the main result we show that simple laws can be used for proving a refinement step involving continuous behaviour within the Refinement Calculus.

Keywords: actions, action systems, continuous behaviour, hybrid systems, refinement

TUCS Research Group
Programming Methodology Research Group

1 Introduction

Action Systems, originally proposed by Back and Kurki-Suonio [2], are predicate transformer based systems for modelling discrete computations. They have been extensively used in the development of reactive and concurrent programs, see Sere et al. [3, 7]. Action Systems support the stepwise refinement paradigm, which is formalised in the *Refinement Calculus* [5].

The *differential action* introduced recently by Rönkkö and Ravn [15] allows us to use differential equations directly in capturing continuous phenomena. The differential action has predicate transformer semantics like the other actions, and therefore, it fits seamlessly to the Action Systems framework. Hence, Action Systems with differential actions can be used for modelling hybrid systems, where a discrete controller interacts with some continuously evolving environment.

Laws for the refinement of actions involving differential actions have not been investigated, yet. The existence of such laws would not only justify the introduction of the differential action, but also open the refinement technique for the development of hybrid systems with Action Systems. For instance, a hybrid system specification could be developed from a discrete skeleton, details of continuous behaviour could be added in a stepwise manner to an existing specification, and some complex continuous behaviour could be replaced with a combination of simpler discrete and continuous behaviours. The contribution of this paper is to present such refinement laws for differential actions, and show that the refinement of continuous behaviour has a strong connection to path equivalence or *deformation* in topology: two paths are equivalent if there exists a continuous monotonic mapping such that the paths traverse the same values in the same order under the mapping [13]. We focus the investigation to systems in isolation. The refinement of components in a hybrid system is a topic of future research.

Overview.

We start by defining actions in Section 2. In Section 3 we investigate the refinement between differential actions. The refinement between continuous and discrete behaviour is investigated in Section 4. In Section 5 we present Action Systems along with the refinement laws for systems in isolation. In Section 6 we use a travelling train example to illustrate how the presented refinement laws are used. Section 7 concludes and discusses directions for further work as well as provides pointers to related work. In Appendix A we prove some algebraic properties for differential actions. These properties are used in Appendix B, where we show that the differential action fulfills the four healthiness conditions given by Dijkstra [8].

2 Actions

An action is any statement in Dijkstra's guarded command language [8]. Also, a pure guarded command can be used as an action. Actions operate on a fix set of program variables. Hence, predicates can be used for specifying states in a system.

When we consider a given action A, we may speak of a *postcondition* q. It is a predicate specifying desirable states after the execution of the action. The meaning of an action is defined with the *weakest precondition* predicate transformer $\mathrm{wp}\,(A,\ q)$. It returns a predicate describing the largest set of states from which the action A terminates and reaches the postcondition q. The *termination*, i.e. reaching an arbitrary state, is given by $\mathrm{t}A\ \hat{=}\ \mathrm{wp}\,(A,\ true)$. We may also speak of enabledness of an action, or *guard condition.* It is a predicate describing all those states from which the action terminates, i.e. $\mathrm{g}A\ \hat{=}\ \neg\mathrm{wp}\,(A,\ false)$.

Let $q[E/X]$ to denote the textual substitution of free variables X with expressions E in a predicate q. For predicate p and actions A and B we define:

$$
\begin{array}{llll}
\mathrm{wp}\,(\{p\},\ q) & \hat{=} & p \wedge q & (assert) \\
\mathrm{wp}\,(\mathbf{abort},\ q) & \hat{=} & false & (abort) \\
\mathrm{wp}\,(\mathbf{skip},\ q) & \hat{=} & q & (skip) \\
\mathrm{wp}\,(X := E,\ q) & \hat{=} & q[E/X] & (assignment) \\
\mathrm{wp}\,(A; B,\ q) & \hat{=} & \mathrm{wp}\,(A,\ \mathrm{wp}\,(B,\ q)\ ) & (sequential\ composition) \\
\mathrm{wp}\,(A \mathbin{[\!]} B,\ q) & \hat{=} & \mathrm{wp}\,(A,\ q)\ \wedge \mathrm{wp}\,(B,\ q) & (non-determin.\ choice) \\
\mathrm{wp}\,(p \rightarrow A,\ q) & \hat{=} & p \Rightarrow \mathrm{wp}\,(A,\ q) & (guarded\ action) \\
\mathrm{wp}\,(\mathbf{do}\ A\ \mathbf{od},\ q) & \hat{=} & \forall n \geq 0 \cdot \mathrm{wp}\,(A^n,\ \mathrm{g}A \vee q) & (iteration) \\
 & & \wedge\ \exists n \geq 0 \cdot \neg\, \mathrm{g}A^n &
\end{array}
$$

where $A^0 \mathrel{\hat{=}} \mathbf{skip}$ and $A^{n+1} \mathrel{\hat{=}} A; A^n$.

The differential action is used for modelling and reasoning about continuous behaviour in a system. Let e be a predicate. A differential action $e : \dot{X} = F(X)$ evolves continuously the values of variables X as long as an evolution guard e holds, see Figure 1. The guard may speak of the variables X. The evolution is given by a system of differential equations. We use the standard convention of writing a system of differential equations as $\dot{X} = F(X)$, where $\dot{X}$ is a componentwise first derivate of X, and $F(X)$ is a vector of functions with an equal number of components as in X.

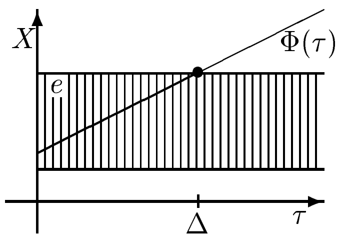

Figure 1: The differential action $e : \dot{X} = F(X)$ evolves the values of X as long as e holds. The new values are the values $\Phi(\Delta)$, where Φ is the evolution curve, and Δ is the first point of time when e ceases to hold.

Let τ and δ be real numbers not appearing in X or e. The differential action has the semantics

$$\begin{aligned}
&\mathrm{wp}\,(e : \dot{X} = F(X),\ q) \\
&\quad \mathrel{\widehat{=}} \exists\, \Phi \cdot \Phi(0) = X \ \wedge\ \dot{\Phi} = F(\Phi) \wedge \\
&\qquad (\forall\, \tau \geq 0 \cdot (e \vee q)[\Phi(\tau)/X] \vee \exists\, 0 \leq \delta < \tau \cdot \neg e[\Phi(\delta)/X]) \wedge \\
&\qquad \exists\, \delta \geq 0 \cdot \neg e[\Phi(\delta)/X]
\end{aligned}$$

In the definition the first two conjuncts require the existence of a solution Φ for the evolution $\dot{X} = F(X)$. The third conjunct captures all those states from which on either the evolution traverses values satisfying e or it reaches some values for which e does not hold but q holds. The last conjunct requires the termination for the continuous evolution, i.e. the evolution eventually reaches values for which e does not hold. Note that $\dot{X} = F(X)$ is *autonomous*, because τ and δ do not appear in it. The differential action has also some other important algebraic properties concerning the evolution disabledness, non-termination, and termination. These properties are shown in Appendix A.

Dijkstra gives four healthiness conditions for program statements. These are strictness, monotonicity, conjunctivity, and continuity. The differential action fulfills all of these conditions. The proofs are shown in Appendix B.

3 Continuous Refinement

The refinement of actions, as formalised in Refinement Calculus, is based on the following definition [3]. An action A is refined by an action B, denoted by $A \leq B$, if the action B can reach all the same post-states at least from the same pre-states as the action A, that is, $\forall\, q \cdot \mathrm{wp}\,(A,\ q) \ \Rightarrow\ \mathrm{wp}\,(B,\ q)$. This definition applies to any action, including the differential action, because it fulfills all the required healthiness conditions. However, the complex weakest precondition semantics for the differential action makes the use of this definition quite cumbersome. Therefore, we would like to have an alternative, but equal, characterisation for the refinement, which is more useful.

It might seem that the definition for the refinement, which speaks only of the pre-states and post-states of an evolution, does not cover the intermediate states. However, the semantics for the differential action imposes a strict ordering to the pre-states with respect to time: the closer the evolution is to the termination the smaller is the set of states that may still lead to a given post-state. This particular property makes the refinement of differential actions to speak also of the intermediate states indirectly. Therefore, one possibility is to look at the refinement of differential actions at the level of evolutions.

Consider two differential actions $0 \leq x < 1 : \dot{x} = 1$ and $0 \leq x < 1 : \dot{x} = 2$. If we think of the execution of these actions, all we can observe, is that both of the evolutions traverse the same values for x in the same order. We cannot say anything about the consumed time, since there is no variable measuring time. For instance, we can only observe that from an initial state $x = 0$ both of these actions traverse all the values from 0 to 1 for x. Since we cannot distinguish the evolutions from each other, it seems plausible that one of the actions is a refinement of the other.

In the following we show that proving the evolutions of two differential actions to be observably equal, i.e. path equivalent, is the same as proving the refinement using the definition above. Moreover, since it is easier to show path equivalence than to use the above definition for refinement, we also provide a convenient refinement law for differential actions.

3.1 Refinement

Consider two actions $DA \mathrel{\hat{=}} e : \dot{X} = F(X)$ and $DA' \mathrel{\hat{=}} i : \dot{X} = G(X)$. For simplicity, we assume that all the evolutions are finite, that is, $\forall X \cdot \mathrm{t}(DA) \wedge \mathrm{t}(DA')$. A necessary requirement for the refinement $DA \leq DA'$ to hold is that the evolution guards are equal.

Lemma 1. Let DA and DA' be as above. Then, $DA \leq DA'$ cannot hold, if $\neg(e \Leftrightarrow i)$.

Proof. Suppose $\neg(e \Rightarrow i)$. Then, due to finiteness there exists a predicate p describing such a point in the evolution of DA that $p \Rightarrow e$ holds, but $p \Rightarrow i$ does not. Now, consider a postcondition, q, for which $p \Rightarrow \mathrm{wp}\,(DA,\ q)$, however, for the other action $\neg(p \Rightarrow \mathrm{wp}\,(DA',\ q)\ )$, because $\neg(p \Rightarrow i)$, and therefore, $DA \leq DA'$ does not hold.

Suppose $\neg(i \Rightarrow e)$. Then, due to finiteness there exists a predicate p describing such a point in the evolution of DA' that $p \Rightarrow i$ holds, but $p \Rightarrow e$ does not. For this point $\mathrm{wp}\,(DA',\ p) \equiv \mathit{false}$, whereas $\neg(\mathrm{wp}\,(DA,\ p) \equiv \mathit{false})$, and therefore, $DA \leq DA'$ does not hold. Hence, $DA \leq DA'$ cannot hold, if $\neg(e \Leftrightarrow i)$. $\square$

Also, there exists a necessary requirement concerning the evolution curves.

Lemma 2. Let DA and DA' be as above, with $e \Leftrightarrow i$. The respective finite solution curves for these actions are denoted by Φ and Φ'. Let δ be the time of termination for DA, that is, $\neg e[\Phi(\delta)/X] \wedge \forall\, 0 \leq \tau < \delta \cdot e[\Phi(\tau)/X]$, and δ' be the time of termination for DA'. Then, $\Phi(\delta) = \Phi'(\delta')$, if the refinement $DA \leq DA'$ holds.

Proof. By unfolding $DA \leq DA'$ we get $\forall\, q \cdot q[\Phi(\delta)/X] \Rightarrow q[\Phi'(\delta')/X]$. Clearly, this holds, only if $\Phi(\delta) = \Phi'(\delta')$. $\square$

Lemma 2 shows an obvious consequence that follows, when the refinement holds between two differential actions. However, we may exploit even further the structure that is added to the state space by the differential action. By doing so, we gain more information concerning the intermediate states in the evolutions as shown by the following lemma.

Lemma 3. Let DA, DA', Φ, and Φ' be as in Lemma 2. Then, the evolutions Φ and Φ' are continuous, loop-free, and traverse the same states in the same order, if the refinement $DA \leq DA'$ holds.

Proof. First we conclude that Φ and Φ' traverse the same states. Because $DA \leq DA'$ holds, we know that DA' must reach all the same post-states at least from the same pre-states as DA. Therefore, Φ' traverses at least the same states as Φ. Since the differential equations appearing in DA and DA' are autonomous, we know that for any one state in the state space there is only one evolution curve that passes it. Furthermore, because all the evolutions are finite, we know that all the states are traversed by some evolution curve. Since the evolution guards are equal, $e \Leftrightarrow i$, DA' must reach all the same post-states from exactly the same pre-states as DA. Otherwise, there would be more than one evolution curve passing one state in the state space. Therefore, Φ and Φ' traverse the same states.

Next we show that Φ and Φ' traverse not only the same states but also in the same order. Consider a post-state p that is reachable by DA. Because the differential equations appearing in DA are autonomous and all the evolutions are finite, the states $\mathrm{wp}\,(DA,\ p)$ are connected by one and only one continuous finite loop-free evolution. Let that evolution be Φ_p. Similarly, all the states $\mathrm{wp}\,(DA',\ p)$ are connected by a continuous finite loop-free evolution Φ'_p. As noted, both Φ_p and Φ'_p traverse exactly the same states. But then, Φ_p and Φ'_p must traverse the same states in the same order, because otherwise Φ'_p either contains loops or is discontinuous. $\square$

With the help of Lemma 3 it is easy to show the main theorem, which states that the refinement between differential actions holds if and only if all the corresponding evolution curves are path equivalent.

Theorem 4. Let DA, DA', Φ, and Φ' be as in Lemma 2. Then, $DA \leq DA'$ holds, iff $\exists\, \mu \cdot \Phi \equiv \Phi' \circ \mu$, where μ is a continuous monotonic increasing function, and $\circ$ is the usual function composition.

Proof. First the proof to direction "$\Rightarrow$". By assumption and Lemma 3 we know that the evolutions Φ and Φ' are continuous, finite, loop-free, and traverse the same states in the same order. Therefore, there exists a mapping μ which is also a continuous monotonic increasing function such that $\Phi \equiv \Phi' \circ \mu$.

Next the proof to direction "$\Leftarrow$". As noted earlier, $DA \leq DA'$ unfolds to $\forall\, q \cdot q[\Phi(\delta)/X] \Rightarrow q[\Phi'(\delta')/X]$. This holds, if $\Phi(\delta) = \Phi'(\delta')$. By assumption there exists a continuous mapping μ such that $\Phi \equiv \Phi' \circ \mu$, and therefore,

$\Phi(\delta) = \Phi'(\mu(\delta))$ holds. But then, δ' is $\mu(\delta)$, and thus, $\Phi(\delta) = \Phi'(\delta')$ also holds. $\square$

This path equivalence property corresponds to the intuition of observability that was discussed earlier. It is also more convenient to use than the original definition for the refinement. For instance, we can now derive a result that proves the earlier example.

Corollary 5. The refinement $e : \dot{x} = c \ \leq \ e : \dot{x} = k$ with constants c and k holds, if $ck > 0$.

Proof. According to Theorem 4 the refinement $e : \dot{x} = c \ \leq \ e : \dot{x} = k$ holds, if there exists a μ such that $\Phi \equiv \Phi' \circ \mu$ holds, where $\Phi(\tau) = x + c\tau$ and $\Phi'(\tau) = x + k\tau$. Such a μ is $\mu(\tau) = \frac{c}{k}\tau$ and it exists, because $ck > 0$. $\square$

3.2 Data refinement

The results above can also be applied to *data refinement* that speaks of the refinement between actions operating on separate state spaces. Let an action A operate on variables X, an action B operate on separate variables Y, and a predicate R be a refinement relation that tells how the values of X are related to the values of Y. The action A is said to be data refined by the action B under R, denoted by $A \leq_R B$, if B reaches the same post-states as A at least from the same pre-states as A with respect to R [1]. Let q be a postcondition speaking only of variables X. The data refinement is captured by $\forall q \cdot R \wedge \mathrm{wp}\,(A,\ q) \ \Rightarrow \ \mathrm{wp}\,(B,\ \exists X \ \cdot \ R \wedge q)$, where $\exists X \ \cdot \ R \wedge q$ is a predicate on variables Y. Again, due to the complex differential action semantics it is hard to use this definition as such with differential actions. Therefore, we derive an alternative, but equal, characterisation for the data refinement of differential actions.

Consider two actions $DA \ \hat{=} \ e : \dot{X} = F(X)$ and $DB \ \hat{=} \ i : \dot{Y} = G(Y)$ with finite evolutions, and a refinement relation R for the disjoint variables X and Y. A necessary requirement for the refinement $DA \leq_R DB$ to hold is that the evolution guards are equal with respect to R.

Lemma 6. Let DA, DB, and R be as above. Then, $DA \leq_R DB$ cannot hold, if $\neg(R \wedge e \Leftrightarrow R \wedge i)$.

Proof. Suppose $\neg(R \wedge e \Rightarrow i)$. Due to finiteness there exists a predicate p speaking only of variables X that describes such a point in the evolution of DA, for which $p \Rightarrow e$ holds, but $R \wedge p \Rightarrow i$ does not. Consider a postcondition q, for which $p \Rightarrow \mathrm{wp}\,(DA,\ q)$, but then $\neg(R \wedge p \Rightarrow \mathrm{wp}\,(DB,\ \exists X \ \cdot \ R \wedge q)\ )$, and therefore, $DA \leq_R DB$ does not hold.

Suppose $\neg(R \wedge i \Rightarrow e)$. Again, due to finiteness, there exists a predicate p speaking only of variables X that describes such a point in the evolution of DB, for which $\exists X \cdot R \wedge p \Rightarrow i$ holds, but $p \Rightarrow e$ does not. For this point $\text{wp}(DB, \exists X \cdot R \wedge p) \equiv false$, whereas, $\neg(\text{wp}(DA, p) \equiv false)$, and therefore, $DA \leq_R DB$ does not hold. Hence, $DA \leq_R DB$ cannot hold, if $\neg(R \wedge e \Leftrightarrow R \wedge i)$. $\square$

Moreover, there exists a necessary requirement concerning the evolution curves.

Lemma 7. Let R, DA and DB be as in Lemma 6. Moreover, let the evolution guards of DA and DB be equal with respect to R. We denote the evolution curves of these actions by Φ and Ψ. Let α be the time of termination for DA, that is, $\neg e[\Phi(\alpha)/X] \wedge \forall 0 \leq \tau < \alpha \cdot e[\Phi(\tau)/X]$, and β be the time of termination for DB. Then, $R \Rightarrow R[\Phi(\alpha)/X, \Psi(\beta)/Y]$ holds, if $DA \leq_R DB$ holds.

Proof. $DA \leq_R DB$ unfolds to $\forall q \cdot R \wedge q[\Phi(\alpha)/X] \Rightarrow R[\Phi(\alpha)/X, \Psi(\beta)/Y]$, which holds, only if $R \Rightarrow R[\Phi(\alpha)/X, \Psi(\beta)/Y]$. $\square$

By assuming some properties on the data refinement relation R we may exploit further the structure that is added to the state spaces by the differential actions as shown by the following lemma.

Lemma 8. Let DA, DB, Φ, and Ψ be as in Lemma 7. Let E be some predicate describing a proper superset of states described by e such that $e \Rightarrow E \wedge \text{wp}(DA, E)$, and similarly I be a predicate describing a proper superset of states described by i such that $i \Rightarrow I \wedge \text{wp}(DB, I)$. Furthermore, let the data refinement relation R describe a continuous and bijective relation between E and I. Then, the evolutions Φ and Ψ are continuous, loop-free, and traverse the same states related by R in the same order, if the refinement $DA \leq_R DB$ holds.

Proof. First we conclude that the claim holds for all evolutions which terminate at the same state where they start. Since $DA \leq_R DB$ holds and R describes a continuous and bijective relation between E and I, we know that any state outside e has a state outside i that is related by R. Therefore, any evolution of DA that terminates at the same state where it starts has a corresponding evolution of DB that terminates at the same state where it starts with respect to R. Thus, the claim holds for any evolutions which terminate at the same state where they start

Next we conclude that Ψ, which traverses more than one state, traverses at least the same states uniquely related by R as Φ. By definition of the differential action, if an evolution traverses more than one state, it must start within the evolution guard. Therefore, both Φ and Ψ must start within their evolution guards e and i. Due to the properties of R each state in E has a unique corresponding state in I that is related by R. Moreover, because by assumption $DA \leq_R DB$ holds, $e \Rightarrow E \wedge \mathrm{wp}\,(DA,\ E)$, and $i \Rightarrow I \wedge \mathrm{wp}\,(DB,\ I)$, we know that Ψ traverses at least the same states uniquely related by R as Φ.

Next we can conclude that Ψ, which traverses more than one state, traverses exactly the same states uniquely related by R as Φ. We have concluded that Ψ traverses at least the same states uniquely related by R as Φ. Since the differential equations appearing in DA and DB are autonomous, we know that for any one state in either of the state spaces there is only one evolution curve that passes it. Furthermore, because all the evolutions are finite, we know that all the states are traversed by some evolution curve. Therefore, Φ and Ψ traverse the same states uniquely related by R, otherwise there would be more than one evolution curve passing one state in either of the state spaces.

Finally we show that Φ, which traverses more than one state, traverses the same states uniquely related by R in the same order as Φ. Consider a post-state p that is reachable by DA. Because the differential equations appearing in DA are autonomous and all the evolutions are finite, the states $\mathrm{wp}\,(DA,\ p)$ are connected by one and only one continuous finite loop-free evolution. Let that evolution be Φ_p. Similarly, all the states $\mathrm{wp}\,(DB,\ p)$ are connected by a continuous finite loop-free evolution Ψ_p. As noted, both Φ_p and Ψ_p traverse exactly the same states uniquely related by R. But then, Φ_p and Ψ_p must traverse the states in the same order, because R described a continuous mapping between the state spaces.

Therefore, the evolutions Φ and Ψ are continuous, loop-free, and traverse the same states related by R in the same order. $\square$

With the help of Lemma 8 it is easy to show the theorem, which states that the refinement between differential actions in different state spaces reduces to a path equivalence problem.

Theorem 9. Let DA, DB, Φ, Ψ, α, and β be as in Lemma 7, and R be as in Lemma 8. Then, refinement $DA \leq_R DB$ holds, iff $\exists\, \mu \cdot \forall\, 0 \leq \tau \leq \alpha \cdot R \Rightarrow R[\Phi(\tau)/X, (\Psi \circ \mu)(\tau)/Y]$, where μ is a continuous monotonic increasing function, and $\circ$ is the usual function composition.

Proof. First the proof to direction "$\Rightarrow$". By assumption and Lemma 8 we know that the evolutions Φ and Ψ are continuous, finite, loop-free, and traverse the same states uniquely related by R in the same order. Moreover, we know that R describes a continuous mapping between the evolutions. Therefore, there exists a mapping μ which is also a continuous monotonic increasing function such that $\forall\, 0 \leq \tau \leq \alpha \cdot R \Rightarrow R[\Phi(\tau)/X, (\Psi \circ \mu)(\tau)/Y]$.

Next the proof to direction "$\Leftarrow$". As noted earlier, $DA \leq DB$ unfolds to $\forall\, q \cdot R \wedge q[\Phi(\alpha)/X] \Rightarrow R[\Phi(\alpha)/X, \Psi(\beta)/Y]$. This holds, if $R \Rightarrow R[\Phi(\alpha)/X, \Psi(\beta)/Y]$. By assumption there exists a continuous mapping

μ such that $\forall\, 0 \leq \tau \leq \alpha \cdot R \Rightarrow R[\Phi(\tau)/X, (\Psi \circ \mu)(\tau)/Y]$, and therefore, $R \Rightarrow R[\Phi(\alpha)/X, (\Psi \circ \mu)(\alpha)/Y]$ holds. But then, β is $\mu(\alpha)$, and thus, $R \Rightarrow R[\Phi(\alpha)/X, \Psi(\beta)/Y]$ also holds. $\square$

4 Mixed Refinement

Although the differential action is the only action modelling continuous behaviour, it is just an action among the other actions defined with the weakest precondition predicate transformer. This permits us to reason about *mixed refinement* where continuous behaviour is refined by some discrete behaviour or vice versa. Hence, we may develop hybrid systems from discrete skeletons or from fully continuous mathematical constructions by means of refinement steps. Moreover, we may use refinement to simplify existing hybrid specifications to gain easier analysis for specific purposes.

In this section we focus on the refinement between an assignment statement and a differential action. Without loss of generality, we limit the investigation to that of data refinement.

4.1 Discretisation

In *discretisation* a differential action is refined by some non-differential action. Thus, the intermediate continuous behaviour is removed by refinement. This type of refinement is required when some analytical component, say an ordinary electrical motor, is replaced with a similar kind of discrete component, like for instance a step motor. Discretisation may also be used for simplifying the specification to gain simpler analysis of some overall behaviour.

Theorem 10. Let a differential action $e : \dot{X} = F(X)$ with finite evolutions Φ operate on variables X. The time of termination for this differential action is denoted by Δ. Let an assignment statement $Y := E$ operate on variables Y, which are disjoint from X, and let R be the refinement relation. Then, the data refinement $e : \dot{X} = F(X) \;\; \leq_R \;\; Y := E$ holds iff $R \Rightarrow R[\Phi(\Delta)/X, E/Y]$.

Proof. First to the direction "$\Rightarrow$". By unfolding $e : \dot{X} = F(X) \;\leq_R\; Y := E$ we get an expression $\forall q \cdot R \wedge q[\Phi(\Delta)/X] \Rightarrow R[\Phi(\Delta)/X, E/Y]$, which cannot hold, unless $R \Rightarrow R[\Phi(\Delta)/X, E/Y]$. The claim holds trivially to the direction "$\Leftarrow$" if the condition $R \Rightarrow R[\Phi(\Delta)/X, E/Y]$ holds. $\square$

4.2 Continuation

The reverse refinement step for discretisation is called *continuation*. In continuation a non-differential action is refined by a differential action causing the introduction of intermediate continuous behaviour to a discrete state change. By using continuation one can develop a hybrid system specification starting from an entirely discrete specification. The following theorem shows that continuation is applicable under the same condition as discretisation.

Theorem 11. Let a differential action $e : \dot{X} = F(X)$ with finite evolutions Φ operate on variables X. The time of termination for this differential action is denoted by Δ. Let an assignment statement $Y := E$ operate on variables Y, which are disjoint from X. Furthermore, let R be the refinement relation. Then, the data refinement $Y := E \;\; \leq_R \;\; e : \dot{X} = F(X)$ holds iff $R \Rightarrow R[E/Y, \Phi(\Delta)/X]$.

The proof for this theorem is analogous to the proof for Theorem 10, and thus, is omitted here.

5 Action Systems

So far we have been discussing the refinement of individual actions. However, a hybrid system can very seldom be modelled by an action. For this purpose we use Action Systems. An *action system* $\mathcal{A}$ is an initialised block of the form

$$\mathcal{A} \;=\; |[\; \mathbf{var} \;\; X : T \;\; \bullet \; X := E; \;\; \mathbf{do}\, A_1 \,[\!]\; \ldots \,[\!]\; A_n \;\mathbf{od}\;]|$$

The expression **var** $X : T$ declares a list of variables X with types T. These variables form the *state space* of $\mathcal{A}$. Some of the variables X may be *observable* from outside. These variables are decorated in the declaration with a superscript asterisk, e.g. a^*. The action $X := E$ initialises all the variables X by expressions E.

After initialisation, the actions A_1 ... A_n are repeatedly executed when enabled. The selection of an enabled action for execution is non-deterministic. There are no fairness assumptions about selection of actions. The execution of any action, including the differential action, is always atomic. This means that an evolution continues without interruption till the evolution guard becomes disabled. Only after that the other enabled actions have a chance for execution. If two enabled actions refer to disjoint variables, their execution can be in any order or in parallel. Hence, this models parallelism by interleaving.

The execution of $\mathcal{A}$ terminates when none of the actions is enabled, i.e. $\neg\, \mathrm{g}(\; A_1 \,[\!]\; \ldots \,[\!]\; A_n \;)$. Similarly, the execution of an action system aborts when an executed action aborts.

5.1 Refinement

The action system formalism allows also the refinement of entire action systems. A refining action system preserves the overall behaviour with respect to the observable variables [4]. Consider two action systems

$$
\begin{array}{lcl}
\mathcal{A} & \hat{=} & |[\ \textbf{var}\quad X^*, Y : T, V \quad \bullet\ X, Y := E, I;\quad \textbf{do}\, A\ \ \textbf{od}\ ]| \\
\mathcal{A}' & \hat{=} & |[\ \textbf{var}\quad X^*, Z : T, W \quad \bullet\ X, Z := E, O;\quad \textbf{do}\, A' \,[\!]\, B\ \ \textbf{od}\ ]|
\end{array}
$$

and a refinement relation R. As shown by Back [1], the refinement $\mathcal{A} \leq_R \mathcal{A}'$ holds if

(1) the initialisations do not contradict with the refinement relation,
(2) all the actions A' have the same effect to the variables as the actions A,
(3) the additional actions B do not have an effect on the state space of $\mathcal{A}$,
(4) the additional actions terminate, and
(5) $\mathcal{A}'$ does not terminate, unless $\mathcal{A}$ terminates as well.

These proof obligations are stated formally as

(1) $R[E, I, O/X, Y, Z]$,
(2) $A \leq_R A'$,
(3) $\textbf{skip} \leq_R B$,
(4) $R \Rightarrow \mathrm{t}(\textbf{do}\, B\ \ \textbf{od})$, and
(5) $R \wedge \mathrm{g}A \Rightarrow \mathrm{g}A' \vee \mathrm{g}B$.

It is also shown by Back [1] that for proving the refinement of guarded actions $g \to A \ \leq_R\ g' \to A'$ it is adequate to show that the guard is not weakened, $R \wedge g' \Rightarrow g$, and that the body of the action is also refined in the context of the new guard, $\{g'\}; A \leq_R A'$.

Since the semantics for the differential action is given with the weakest precondition, any of the actions in A, A', and B may contain differential actions. Moreover, we have shown in this paper how to prove a refinement of actions involving a differential action. Therefore, the proof obligations for action systems above cover also the cases, where the action systems are modelling some hybrid systems.

6 Example: Travelling Train

We use a simple train example to illustrate how the given theorems are used in practise. In this example a train travels a distance of 2560 units. It starts by accelerating to travelling velocity, 20 units per one time unit, which is then kept for the most of the journey. At the end, the train decelerates to a full stop. Figure 2 depicts how the velocity is changed during the journey.

In the following we will develop an action system modelling such a travelling train. We start with a discrete action system describing only the state changes, and add the details of continuous behaviour by refinement steps. At the end of the example, we show how to use discretisation to analyse the timing in the developed action system.

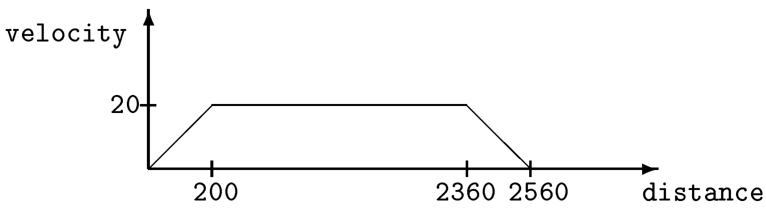

Figure 2: Velocity during the journey.

6.1 Initial specification

The discrete action system below is the initial specification describing only the state changes during the journey.

$$
\begin{array}{lcl}
\mathcal{T}rain & \hat{=} & |[\ \mathbf{var}\quad x : \mathbb{R} \quad \bullet \quad x := 0; \\
& & \quad \mathbf{do}\ 0 \leq x < 200 \qquad \rightarrow \quad x := 200 \\
& & \quad [\!]\ \ 200 \leq x < 2360 \quad \rightarrow \quad x := 2360 \\
& & \quad [\!]\ \ 2360 \leq x < 2560 \quad \rightarrow \quad x := 2560 \\
& & \quad \mathbf{od} \\
& &]|
\end{array}
$$

In this action system the variable x denotes the travelled distance. The initialisation states that the train is originally located at the very beginning. The first action models the acceleration state, the second action models the state, where the train travels with a constant velocity, and the last action models the deceleration state. The action system terminates when the train has travelled the whole distance.

6.2 Introducing continuous movement

In this first refinement step we introduce continuous movement without the notion of acceleration and deceleration. Such an action system is the one below.

$$
\begin{array}{lcl}
\mathcal{T}rain' & \hat{=} & |[\ \mathbf{var}\quad x' : \mathbb{R} \quad \bullet \quad x' := 0; \\
& & \quad \mathbf{do}\ 0 \leq x' < 200 \qquad \rightarrow \quad x' < 200 : \dot{x}' = 1 \\
& & \quad [\!]\ \ 200 \leq x' < 2360 \quad \rightarrow \quad x' < 2360 : \dot{x}' = 1 \\
& & \quad [\!]\ \ 2360 \leq x' < 2560 \quad \rightarrow \quad x' < 2560 : \dot{x}' = 1 \\
& & \quad \mathbf{od} \\
& &]|
\end{array}
$$

In order to prove that this is a refinement of the original specification, we need to define a refinement relation capturing the features that are to be preserved. Because the only requirement for the action system above is to retain the meaning of the variable measuring the distance, the refinement relation is $R1 \ \hat{=}\ \ x = x'$.

Since both $\mathcal{T}rain$ and $\mathcal{T}rain'$ have an equal number of actions, we can prove the refinement $\mathcal{T}rain \leq_{R1} \mathcal{T}rain'$ in three steps.

i) Initialisations. The initialisations in the action systems do not contradict with the refinement relation, because $R1[0/x, 0/x']$ holds.

ii) Actions. Each action in $\mathcal{T}rain$ has a corresponding refining action in $\mathcal{T}rain'$. For instance, consider the case $0 \le x < 200 \to x := 200 \quad \le_{R1} \quad 0 \le x' < 200 \to x' < 200 : \dot{x}' = 1$. The guard is not weakened, because $R1 \wedge 0 \le x' < 200 \Rightarrow 0 \le x < 200$ holds. Therefore, we only need to prove the refinement of the body $\{0 \le x' < 200\}; x := 200 \quad \le_{R1} \quad x' < 200 : \dot{x}' = 1$. Here, we can use Theorem 11. According to that theorem we only need to prove that $0 \le x' < 200 \wedge R1 \Rightarrow R1[200/x, \Phi'(\Delta')/x']$ holds, where $\Phi'(\tau) = x' + \tau$ and $\Delta' = 200 - x'$. The proof of this comes by substitution. The refinement of other actions is proven similarly.

iii) Termination. The termination condition is not strengthened, because the condition $R1 \wedge \mathrm{g}\mathcal{T}rain \Rightarrow \mathrm{g}\mathcal{T}rain'$, which unfolds to $R1 \wedge 0 \le x < 2560 \Rightarrow 0 \le x' < 2560$, holds.

6.3 Introducing velocity

In this refinement step we add the notion of acceleration and deceleration to the specification. Such an action system, where v denotes the velocity, is the one below.

$$
\begin{array}{lll}
\mathcal{T}rain'' & \widehat{=} & |[\ \mathbf{var}\ \ x'', v'' : \mathbb{R}, \mathbb{R} \ \bullet\ x'', v'' := 0, 0; \\
 & & \quad \mathbf{do}\ 0 \le x'' < 200 \quad \to\ x'' < 200 : \dot{x}'', \dot{v}'' = v'', 1 \\
 & & \quad \talloblong\ \ 200 \le x'' < 2360 \quad \to\ x'' < 2360 : \dot{x}'' = v'' \\
 & & \quad \talloblong\ \ 2360 \le x'' < 2560 \quad \to\ x'' < 2560 : \dot{x}'', \dot{v}'' = 1, -1 \\
 & & \quad \mathbf{od} \\
 & &]|
\end{array}
$$

The refinement relation used for proving that this action system is indeed a refinement of $\mathcal{T}rain'$ must describe what we mean by acceleration and deceleration. Such a refinement relation is the following.

$$
\begin{array}{lll}
R2 & \widehat{=} & (x' = x'') \ \wedge \\
 & & (0 \le x'' < 200 \Rightarrow v'' = \sqrt{2x''}) \ \wedge \\
 & & (200 \le x'' < 2360 \Rightarrow v'' = 20) \ \wedge \\
 & & (2360 \le x'' < 2560 \Rightarrow v'' = \sqrt{5120 - 2x''}) \ \wedge \\
 & & (x'' = 2560 \Rightarrow v'' = 0).
\end{array}
$$

$R2$ describes a continuous and bijective relation between variable x' and variables x'' and v'' within an interval $0 \le x' \le 2560$. The first conjunct states that the meaning of the variable measuring the distance must remain the same. The second, the third, and the fourth conjuncts describe how the velocity behaves during the acceleration, the travelling with constant speed, and the deceleration. The last conjunct reflects the fact that once the train is in its destination it is fully stopped. Note that the primes after the variables tell into which action system the variables belong to and have no other meaning.

The proof of the refinement $\mathcal{T}rain' \leq_{R2} \mathcal{T}rain''$ is done in three steps.

i) Initialisations. The initialisations in the action systems do not contradict with the refinement relation, because $R2[0/x', 0/x'', 0/v'']$ holds.

ii) Actions. Each action in $\mathcal{T}rain'$ has again a corresponding refining action in $\mathcal{T}rain''$. For instance, consider the case $0 \leq x' < 200 \to x' < 200 : \dot{x}' = 1 \quad \leq_{R2} \quad 0 \leq x'' < 200 \to x'' < 200 : \dot{x}'', \dot{v}'' = v'', 1$. The guard is not weakened, because $R2 \wedge 0 \leq x'' < 200 \Rightarrow 0 \leq x' < 200$ holds. Therefore we only need to prove the refinement of the body $\{0 \leq x'' < 200\}; x' < 200 : \dot{x}' = 1 \quad \leq_{R2} \quad x'' < 200 : \dot{x}'', \dot{v}'' = v'', 1$. Because $R2$ describes a continuous and bijective relation between variable x' and variables x'' and v'' within a proper superset of an interval $0 \leq x' < 200$, we can use Theorem 9. According to that theorem, we have to show that there exists such a μ that $\forall\, 0 \leq \tau \leq \alpha \cdot R \Rightarrow R[\Phi_{x'}(\tau)/x', \Psi_{x''}(\mu(\tau))/x'', \Psi_{v''}(\mu(\tau))/v'']$ holds, where $\alpha = 200 - x'$, $\Phi_{x'}(\tau) = x' + \tau$, $\Psi_{x''}(\tau) = x'' + v''\tau + \frac{1}{2}\tau^2$, and $\Psi_{v''}(\tau) = v'' + \tau$. Such a μ is $\mu(\tau) = \sqrt{2x'' + 2\tau} - \sqrt{2x''}$ for the context $0 \leq x'' < 200$, and hence, the refinement holds. The refinement of other actions is proven similarly.

iii) Termination. The termination condition is not strengthened, because the condition $R2 \wedge 0 \leq x' < 2560 \Rightarrow 0 \leq x'' < 2560$ holds.

6.4 Timing analysis

As the last part of this example, we show how discretisation is used for simplifying the analysis of a hybrid system. Here we investigate the timing of the travelling train example. For this we need to add an auxiliary variable c measuring the passage of time. Such an extension of the travelling train specification is the one below.

$$
\begin{array}{lll}
\mathcal{T}imed\mathcal{T}rain & \hat{=} & |[\ \textbf{var}\ \ x, v, c : \mathbb{R}, \mathbb{R}, \mathbb{R} \ \bullet\ x, v, c := 0, 0, 0; \\
& & \quad \begin{array}{lll}
\textbf{do}\ 0 \leq x < 200 & \to & 0 \leq x < 200 : \dot{x}, \dot{v}, \dot{c} = v, 1, 1 \\
\ [\!]\ \ 200 \leq x < 2360 & \to & x < 2360 : \dot{x}, \dot{c} = v, 1 \\
\ [\!]\ \ 2360 \leq x < 2560 & \to & x < 2560 : \dot{x}, \dot{v}, \dot{c} = 1, -1, 1 \\
\textbf{od} & &
\end{array} \\
& &]|
\end{array}
$$

In the discretisation step we remove all the continuous behaviour, as well as the variable v used for stating the velocity of the train. The obtained completely discrete action system is the following.

$$
\begin{array}{lcll}
\mathit{TimedTrain'} & \hat{=} & |[\ \mathbf{var}\ \ x', c' : \mathbb{R}, \mathbb{R} & \bullet\ \ x', c' := 0, 0; \\
& & \quad \mathbf{do}\ 0 \le x' < 200 & \rightarrow\ \ x', c' := 200, c' + 20 - \sqrt{2x'} \\
& & \quad [\!]\ \ 200 \le x' < 2360 & \rightarrow\ \ x', c' := 2360, c' + 118 - \frac{x'}{20} \\
& & \quad [\!]\ \ 2360 \le x' < 2560 & \rightarrow\ \ x', c' := 2560, c' + \sqrt{5120 - 2x'} \\
& & \quad \mathbf{od} & \\
& &]| &
\end{array}
$$

The refinement relation that we use is otherwise the same as in the previous refinement step except that we add a condition stating that the new clock variable c' must retain the behaviour of the old clock variable c. Thus, the refinement relation is

$$
\begin{array}{lcl}
R3 & \hat{=} & (x = x')\ \wedge \\
& & (0 \le x < 200\ \Rightarrow\ v = \sqrt{2x})\ \wedge \\
& & (200 \le x < 2360\ \Rightarrow\ v = 20)\ \wedge \\
& & (2360 \le x < 2560\ \Rightarrow\ v = \sqrt{5120 - 2x})\ \wedge \\
& & (x = 2560\ \Rightarrow\ v = 0) \wedge \\
& & c = c' \quad .
\end{array}
$$

The proof of $\mathit{TimedTrain} \le_{R3} \mathit{TimedTrain'}$ is done in three steps.

i) Initialisations. The initialisations in the action systems do not contradict with the refinement relation, because $R3[0/x, 0/v, 0/c, 0/x', 0/c']$ holds.

ii) Actions. Each action in $\mathit{TimedTrain}$ has again a corresponding refining action in $\mathit{TimedTrain'}$. For instance, consider the case $0 \le x < 200 \rightarrow 0 \le x < 200 : \dot{x}, \dot{v}, \dot{c} = v, 1, 1 \quad \le_{R3} \quad 0 \le x' < 200 \rightarrow x', c' := 200, c' + 20 - \sqrt{2x'}$. The guard is not weakened, because $R3 \wedge 0 \le x' < 200 \Rightarrow 0 \le x < 200$ holds. Therefore, we only need to prove the refinement of the body $\{0 \le x' < 200\}; 0 \le x < 200 : \dot{x}, \dot{v}, \dot{c} = v, 1, 1 \quad \le_{R3} \quad x', c' := 200, c' + 20 - \sqrt{2x'}$. Here, we can use Theorem 10. According to that theorem we only need to prove that $0 \le x' < 200 \wedge R3 \Rightarrow R3[\Phi_c(\Delta)/c, \Phi_v(\Delta)/v, \Phi_x(\Delta)/x, 200/x', c' + 20 - \sqrt{2x'}]$ holds, where $\Phi_x(\tau) = x + v\tau + \frac{1}{2}\tau^2$, $\Phi_v(\tau) = v + \tau$, $\Phi_c(\tau) = c + \tau$, and $\Delta = 20 - v$. The proof of this comes by substitution. The refinement of other actions is proven similarly.

iii) Termination. The termination condition is not strengthened, because the condition $R3 \wedge 0 \le x < 2560 \Rightarrow 0 \le x' < 2560$ holds.

One way of analysing the timing of a simple system, such as the timed train above, is to use traces with clock information. In general, a trace is a sequence showing how the values of the variables change during the execution. Such trace semantics of Action Systems have been studied by Back and von Wright [4]. A trace is extracted from an action system by recording into a sequence the values of the variables after the execution of each enabled action. In the presence of non-determinism there are several possible traces.

Because the $\mathcal{T}imedTrain'$ is a deterministic system, there is only one trace, which is

$$\langle\,(x,c)\,\rangle = \langle\,(0,0),\ (200,20),\ (2360,128),\ (2560,148)\,\rangle \quad .$$

From this trace we can see directly the timing of the travelling train. According to the trace the train has reached travelling velocity after 20 time units of acceleration, starts decelerating after 128 time units, and stops at the final destination after 148 time units from the very start.

7 Conclusion

In this paper we presented simple, yet powerful, refinement laws for differential actions. Although these laws require the differential equations in the differential actions to be solvable, one can derive specialised laws with no need of solving the differential equations. This was illustrated in Corollary 5. We also showed how the refinement laws for differential actions are used in the development of hybrid systems. For this purpose we used a travelling train example. We limited the investigation of using refinement in hybrid setting to systems in isolation. The refinement of parallel components in a hybrid system is a topic of future research.

There are several formalisms that are used in hybrid systems. The connection between some of these formalisms and Action Systems with differential actions has been studied earlier. For instance, the relation to Branicky's unified model [6] was established by Rönkkö and Ravn [16]. Also, the use of HyTech [10] in analysis of Action Systems with differential actions is discussed elsewhere [15]. Another closely related formalism is Duration Calculus: the refinement method as well as the use of differential equations in Duration Calculus is discussed by Ravn [14]. Refinement in timed and hybrid settings has also been studied in other formalisms [12, 11, 9]. An interesting topic for future research is to see, how the refinement ideas presented here relate to these approaches.

Acknowledgements

We would like to thank Anders P. Ravn and Emil Sekerinski for helpful discussions. The work of Kaisa Sere is supported by the Academy of Finland.

References

[1] R. J. R. Back. Refinement calculus, part II: Parallel and reactive programs. In J. W. de Bakker, W.–P. de Roever, and G. Rozenberg, editors,

Stepwise Refinement of Distributed Systems: Models, Formalisms, Correctness. Proceedings. 1989, volume 430 of *Lecture Notes in Computer Science*. Springer–Verlag, 1990.

[2] R. J. R. Back and R. Kurki-Suonio. Decentralization of process nets with centralized control. In *Proc. of the 2nd ACM SIGACT–SIGOPS Symp. on Principles of Distributed Computing*, pp. 131–142, 1983.

[3] R. J. R. Back and K. Sere. Stepwise refinement of action systems. *Structured Programming*,**12**, pp. 17-30, 1991.

[4] R. J. R. Back and J. von Wright. *Trace Refinement of Action Systems.* In J. Parrow, editor, *CONCUR94*, LNCS 666, Springer-Verlag, 1994.

[5] R. J. R. Back and J. von Wright. *Refinement Calculus: A Systematic Introduction.* Graduate Texts in Computer Science, Springer-Verlag,1998.

[6] M. S. Branicky. *Studies in Hybrid Systems: Modeling, Analysis, and Control.* PhD Thesis. Massachusetts Institute of Technology, EECS Dept, 1995.

[7] M. Butler, E. Sekerinski, and K. Sere. An Action System Approach to the Steam Boiler Problem. In J-R. Abrial, E. Borger, H. Langmaack (eds.) *Formal Methods for Industrial Applications: Specifying and Programming the Steam Boiler Control*, LNCS 1165, Springer-Verlag, 1996.

[8] E. W. Dijkstra. *A Discipline of Programming.* Prentice–Hall International, 1976.

[9] T. A. Henzinger, Z. Manna, and A. Pnueli. *Towards Refining Temporal Specifications into Hybrid Systems.* In R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel (eds.) *Hybrid Systems*, LNCS 736, Springer-Verlag, 1993.

[10] T. A. Henzinger and P.-H. Ho. HyTech: The Cornell hybrid technology tool. In P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors, *Hybrid Systems II*, LNCS 999, pp. 265-293, Springer-Verlag, 1995.

[11] N.A. Lynch and F.W. Vaandrager. *Forward and backward simulations Part II: Timing-based systems.* Information and Computation, 128(1):1-25, July 1996.

[12] Brendan P. Mahony and Ian J. Hayes. *A Case-Study in Timed Refinement: A Mine Pump.* IEEE Transactions on Software Engineering, vol. 18, no. 9, September 1992.

[13] M.H.A. Newman. *Elements of the Topology of Plane Sets of Points.* Cambridge University Press, 1961 (First edition 1939).

[14] A. P. Ravn. *Design of Embedded Real-Time Computing Systems.* Technical University of Denmark, DK-2800 Lyngby, Denmark, Doctoral Dissertation, ID-TR: 1995-170, 1995.

[15] M. Rönkkö and A. P. Ravn. *Action Systems with Continuous Behaviour.* To appear in the *Proceedings of Hybrid Systems V - Fifth International Hybrid Systems Workshop*, LNCS, Springer-Verlag, 1998.

[16] M. Rönkkö and A. P. Ravn. *Switches and Jumps in Hybrid Action Systems.* Proceedings of the Estonian Academy of Sciences. Engineering, June 1998, 4, 2, pp. 106-118.

Appendix A. Algebraic Properties for Differential Action

A.1 Property of evolution disabledness

If the evolution guard e of a differential action $e : \dot{X} = F(X)$, for which the solution curve exists, is disabled in a state, the differential action has no effect to the state space. Hence, we may use the equivalence

$$\neg e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \Leftrightarrow \neg e \wedge q \quad .$$

Proof.

$$\begin{array}{cl}
 & \neg e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \\
= & \{\ \mathit{unfolding\ with\ a\ solution}\ \} \\
 & \neg e \wedge \\
 & (\forall \tau \geq 0 \cdot (e \vee q)[\Phi(\tau)/X] \vee \exists\, 0 \leq \delta < \tau \cdot \neg e[\Phi(\delta)/X]) \wedge \\
 & \exists\, \delta \geq 0 \cdot \neg e[\Phi(\delta)/X] \\
= & \{\ \mathit{unfolding\ quantifiers\ by\ fact}\ \Phi(0) = X\ \} \\
 & \neg e \wedge \\
 & (\ (e \vee q) \wedge \forall \tau > 0 \cdot (e \vee q)[\Phi(\tau)/X] \vee \\
 & \quad (\neg e \vee \exists\, 0 < \delta < \tau \cdot \neg e[\Phi(\delta)/X])\ ) \wedge \\
 & (\neg e \vee \exists\, \delta > 0 \cdot \neg e[\Phi(\delta)/X]) \\
= & \{\ \mathit{simplifying\ second\ row,}\ e\ \mathit{does\ not\ contain}\ \tau\ \mathit{or}\ \delta\ \} \\
 & \neg e \wedge \\
 & (e \vee q) \wedge (\ \neg e \vee (\ \forall \tau > 0 \cdot (e \vee q)[\Phi(\tau)/X] \vee \\
 & \quad \exists\, 0 < \delta < \tau \cdot \neg e[\Phi(\delta)/X]\ )\ ) \wedge \\
 & (\neg e \vee \exists\, \delta > 0 \cdot \neg e[\Phi(\delta)/X]) \\
= & \{\ \mathit{simplifying\ by}\ \neg e \wedge (\neg e \vee p) \Leftrightarrow \neg e\ \} \\
 & \neg e \wedge q
\end{array}$$

□

A.2 Property of non-termination

Consider a differential action $e : \dot{X} = F(X)$ for which the solution curve Φ exists. The differential action $e : \dot{X} = F(X)$ terminates if $\mathrm{t}(e : \dot{X} = F(X))$ holds. If an evolution for a differential action does not terminate, the differential action acts as an **abort** statement, that is, for any predicate q

$$\neg\mathrm{t}(e : \dot{X} = F(X)) \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \Leftrightarrow \mathit{false} \quad .$$

Proof.

$$\begin{array}{cl}
 & \neg\mathrm{t}(e : \dot{X} = F(X)) \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \\
= & \{\ \mathit{unfolding\ termination}\ \} \\
 & \neg(\ \exists\,\delta \geq 0 \cdot \neg e[\Phi(\delta)/X]\ ) \wedge \\
 & \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \\
= & \{\ \mathit{unfolding\ with\ a\ solution}\ \} \\
 & \neg(\ \exists\,\delta \geq 0 \cdot \neg e[\Phi(\delta)/X]\ ) \wedge \\
 & (\forall\,\tau \geq 0 \cdot (e \vee q)[\Phi(\tau)/X] \vee \exists\,0 \leq \delta < \tau \cdot \neg e[\Phi(\delta)/X]) \wedge \\
 & \exists\,\delta \geq 0 \cdot \neg e[\Phi(\delta)/X] \\
= & \{\ \mathit{contradiction\ between\ first\ and\ last\ row}\ \} \\
 & \mathit{false}
\end{array}$$

□

A.3 Property of termination

Consider a differential action $e : \dot{X} = F(X)$, for which a solution Φ exists and which terminates. We denote the time of termination for this differential action by Δ, that is, $\neg e[\Phi(\Delta)/X] \wedge \forall\,0 \leq \tau < \Delta \cdot e[\Phi(\tau)/X]$. Such a differential action acts as a substitution similarly to an assignment statement, that is, for any predicate q

$$\mathrm{t}(e : \dot{X} = F(X)) \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \Leftrightarrow \mathrm{t}(e : \dot{X} = F(X)) \wedge q[\Phi(\Delta)/X].$$

Proof.

$$
\begin{array}{ll}
& \mathrm{t}(e : \dot{X} = F(X)) \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \\
= & \{\ \textit{unfolding termination}\ \} \\
& \exists\, \delta \geq 0 \cdot \neg e[\Phi(\delta)/X] \wedge \\
& \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \\
= & \{\ \textit{unfolding with a solution}\ \} \\
& \exists\, \delta \geq 0 \cdot \neg e[\Phi(\delta)/X] \wedge \\
& (\forall\, \tau \geq 0 \cdot (e \vee q)[\Phi(\tau)/X] \vee \exists\, 0 \leq \delta < \tau \cdot \neg e[\Phi(\delta)/X]) \wedge \\
& \exists\, \delta \geq 0 \cdot \neg e[\Phi(\delta)/X] \\
= & \{\ \textit{simplification}\ \} \\
& \exists\, \delta \geq 0 \cdot \neg e[\Phi(\delta)/X] \wedge \\
& \forall\, \tau \geq 0 \cdot (e \vee q)[\Phi(\tau)/X] \vee \exists\, 0 \leq \delta < \tau \cdot \neg e[\Phi(\delta)/X] \\
= & \{\ \textit{case analysis around}\ \Delta\ \} \\
& \exists\, \delta \geq 0 \cdot \neg e[\Phi(\delta)/X] \wedge \\
& (\ \forall\, 0 \leq \tau < \Delta \cdot (e \vee q)[\Phi(\tau)/X] \vee \exists\, 0 \leq \delta < \tau \cdot \neg e[\Phi(\delta)/X]\ ) \wedge \\
& (\ \forall\, \tau = \Delta \cdot (e \vee q)[\Phi(\tau)/X] \vee \exists\, 0 \leq \delta < \tau \cdot \neg e[\Phi(\delta)/X]\ ) \wedge \\
& (\ \forall\, \Delta < \tau \cdot (e \vee q)[\Phi(\tau)/X] \vee \exists\, 0 \leq \delta < \tau \cdot \neg e[\Phi(\delta)/X]\ ) \\
= & \{\ \textit{property of}\ \Delta\ \} \\
& \exists\, \delta \geq 0 \cdot \neg e[\Phi(\delta)/X] \wedge \\
& (\ \forall\, 0 \leq \tau < \Delta \cdot \textit{true} \vee q[\Phi(\tau)/X] \vee \textit{false}\ ) \wedge \\
& (\ \forall\, \tau = \Delta \cdot \textit{false} \vee q[\Phi(\tau)/X] \vee \textit{false}\ ) \wedge \\
& (\ \forall\, \Delta < \tau \cdot (e \vee q)[\Phi(\tau)/X] \vee \textit{true}\ ) \\
= & \{\ \textit{simplification}\ \} \\
& \exists\, \delta \geq 0 \cdot \neg e[\Phi(\delta)/X] \wedge \\
& \textit{true} \wedge q[\Phi(\Delta)/X] \wedge \textit{true} \\
= & \{\ \textit{simplification and rewriting}\ \} \\
& \mathrm{t}(e : \dot{X} = F(X)) \wedge q[\Phi(\Delta)/X]
\end{array}
$$

□

Appendix B. Proofs for Healthiness of Differential Action

The proofs shown in Appendix B rely on the shorthand for the time of termination, Δ, which is introduced in Appendix A.3, as well as properties of evolution disabledness, non-termination, and termination, which are shown in Appendices A.1, A.2, and A.3 respectively.

B.1 Strictness

An action A is strict if $\mathrm{wp}\,(A,\ false) \ = false$. The differential action is strict.

Proof. Let e and q be predicates. If a differential action $e : \dot{X} = F(X)$ has no solution curve the weakest precondition $\mathrm{wp}\,(e : \dot{X} = F(X),\ q)$ is identically $false$. Hence, in this case the differential action is trivially strict. The proof for the strictness of a differential action $e : \dot{X} = F(X)$, for which the solution curve Φ exists, goes as follows.

$$
\begin{array}{cl}
 & \mathrm{wp}\,(e : \dot{X} = F(X),\ false) \\
= & \{\ case\ analysis\ around\ evolution\ enabledness\ \} \\
 & (e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ false)\ ) \vee \\
 & (\neg e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ false)\ ) \\
= & \{\ property\ of\ evolution\ disabledness\ (A.1)\ \} \\
 & (e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ false)\ ) \vee \\
 & (\neg e \wedge false) \\
= & \{\ simplification\ \} \\
 & e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ false) \\
= & \{\ case\ analysis\ around\ termination\ \} \\
 & (\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ false)\ ) \vee \\
 & (\neg \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ false)\ ) \\
= & \{\ property\ of\ non-termination\ (A.2)\ \} \\
 & (\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ false)\ ) \vee \\
 & (\neg \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge false) \\
= & \{\ simplification\ \} \\
 & \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ false) \\
= & \{\ property\ of\ termination\ (A.3)\ \} \\
 & \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge false[\Phi(\Delta)/X] \\
= & \{\ substitution\ is\ strict\ \} \\
 & false
\end{array}
$$

□

B.2 Monotonicity

Let q and p be predicates. An action A is monotonic if $\mathrm{wp}\,(A,\ q) \Rightarrow \mathrm{wp}\,(A,\ p)$ whenever $q \Rightarrow p$. The differential action is monotonic.

Proof. Let e and q be predicates. If a differential action $e : \dot{X} = F(X)$ has no solution curve the weakest precondition $\mathrm{wp}\,(e : \dot{X} = F(X),\ q)$ is identically $false$. Hence, in this case the differential action is trivially monotonic. The proof for the monotonicity of a differential action $e : \dot{X} = F(X)$, for which the solution curve Φ exists, goes as follows.

$$
\begin{array}{ll}
 & \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \Rightarrow \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
= & \{\ \mathit{case\ analysis\ around\ evolution\ enabledness}\ \} \\
 & (e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \Rightarrow \mathrm{wp}\,(e : \dot{X} = F(X),\ p)\ ) \wedge \\
 & (\neg e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \Rightarrow \mathrm{wp}\,(e : \dot{X} = F(X),\ p)\ ) \\
= & \{\ \mathit{propagation\ over\ implication}\ \} \\
 & (e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \Rightarrow e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p)\ ) \wedge \\
 & (\neg e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \Rightarrow \neg e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p)\ ) \\
= & \{\ \mathit{property\ of\ evolution\ disabledness}\ (A.1)\ \} \\
 & (e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \Rightarrow e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p)\ ) \wedge \\
 & (\neg e \wedge q \Rightarrow \neg e \wedge p) \\
= & \{\ \mathit{assumption}\ q \Rightarrow p,\ \mathit{and\ simplification}\ \} \\
 & e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \Rightarrow e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
= & \{\ \mathit{case\ analysis\ around\ termination}\ \} \\
 & (\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \\
 & \quad \Rightarrow e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p)\ ) \wedge \\
 & (\neg\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \\
 & \quad \Rightarrow e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p)\ ) \\
= & \{\ \mathit{propagation\ over\ implication}\ \} \\
 & (\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \\
 & \quad \Rightarrow \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p)\ ) \wedge \\
 & (\neg\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \\
 & \quad \Rightarrow \neg\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p)\ ) \\
= & \{\ \mathit{property\ of\ non-termination}\ (A.2)\ \} \\
 & (\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \\
 & \quad \Rightarrow \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p)\ ) \wedge \\
 & (\neg\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge false \\
 & \quad \Rightarrow \neg\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge false) \\
= & \{\ \mathit{simplification}\ \} \\
 & \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \\
 & \quad \Rightarrow \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p)
\end{array}
$$

$$
\begin{aligned}
= \quad & \{ \textit{ property of termination } (A.3) \ \} \\
& \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge q[\Phi(\Delta)/X] \\
& \quad \Rightarrow \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge p[\Phi(\Delta)/X] \\
= \quad & \{ \textit{ substitution is monotonic with respect to } q \Rightarrow p \ \} \\
& \textit{true}
\end{aligned}
$$

□

B.3 Conjunctivity

Let q and p be predicates. An action A is conjunctive if $\mathrm{wp}\,(A,\ q) \ \wedge\ \mathrm{wp}\,(A,\ p) \ \Leftrightarrow \mathrm{wp}\,(A,\ q \wedge p)$. The differential action is conjunctive.

Proof. Let e and q be predicates. If a differential action $e : \dot{X} = F(X)$ has no solution curve the weakest precondition $\mathrm{wp}\,(e : \dot{X} = F(X),\ q)$ is identically $false$. Hence, in this case the differential action is trivially conjunctive. The proof for the conjunctivity of a differential action $e : \dot{X} = F(X)$, for which the solution curve Φ exists, goes as follows.

$$
\begin{aligned}
& \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \\
& \Leftrightarrow \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
= \quad & \{ \textit{ case analysis } \} \\
& (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \\
& \quad \Rightarrow \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \ \ ) \wedge \\
& (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
& \quad \Rightarrow \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \ \ ) \\
= \quad & \{ \textit{ case analysis around evolution enabledness } \} \\
& (\ (\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \\
& \qquad \Rightarrow \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \ \ ) \wedge \\
& \quad (\ \neg e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \\
& \qquad \Rightarrow \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \ \ ) \ ) \\
& \wedge \\
& (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
& \quad \Rightarrow \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \ \ )
\end{aligned}
$$

$$
\begin{array}{ll}
= & \{\ \textit{propagation over implication}\ \} \\
 & (\ (\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \\
 & \qquad \Rightarrow\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \quad) \wedge \\
 & \quad (\ \neg e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \\
 & \qquad \Rightarrow\ \neg e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \quad)\ ) \\
 & \wedge \\
 & (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
 & \quad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \quad) \\
= & \{\ \textit{property of evolution disabledness}\ (A.1)\ \} \\
 & (\ (\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \\
 & \qquad \Rightarrow\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \quad) \wedge \\
 & \quad (\ \neg e \wedge q \wedge p \\
 & \qquad \Rightarrow\ \neg e \wedge q \wedge p\ )\ ) \\
 & \wedge \\
 & (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
 & \quad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \quad) \\
= & \{\ \textit{simplification}\ \} \\
 & (\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \\
 & \quad \Rightarrow\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \quad) \\
 & \wedge \\
 & (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
 & \quad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \quad) \\
= & \{\ \textit{case analysis around termination}\ \} \\
 & (\ (\ \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \\
 & \qquad \Rightarrow\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \quad) \wedge \\
 & \quad (\ \neg\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \\
 & \qquad \Rightarrow\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \quad)\ ) \wedge \\
 & (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
 & \quad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \quad) \\
= & \{\ \textit{propagation over implication}\ \} \\
 & (\ (\ \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \\
 & \qquad \Rightarrow\ \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \\
 & \qquad\quad \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \quad) \wedge \\
 & \quad (\ \neg\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \\
 & \qquad \Rightarrow\ \neg\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \\
 & \qquad\quad \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \quad)\ ) \wedge \\
 & (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
 & \quad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \quad)
\end{array}
$$

$$
\begin{aligned}
= \;& \{\ \mathit{property\ of\ non-termination}\ (A.2)\ \} \\
& (\ (\ \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \\
& \qquad \Rightarrow \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \\
& \qquad\quad \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \ \ ) \wedge \\
& \quad (\ \neg\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathit{false} \\
& \qquad \Rightarrow \neg\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \\
& \qquad\quad \mathit{false} \wedge \mathit{false}\ )\ ) \wedge \\
& (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
& \quad \Rightarrow \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \ \ ) \\
= \;& \{\ \mathit{simplification}\ \} \\
& (\ \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \\
& \qquad \Rightarrow \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \\
& \qquad\quad \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \ \ ) \wedge \\
& (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
& \quad \Rightarrow \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \ \ ) \\
= \;& \{\ \mathit{property\ of\ termination}\ (A.3)\ \} \\
& (\ \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge (q \wedge p)[\Phi(\Delta)/X] \\
& \qquad \Rightarrow \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \\
& \qquad\quad q[\Phi(\Delta)/X] \wedge p[\Phi(\Delta)/X]\ ) \wedge \\
& (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
& \quad \Rightarrow \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \ \ ) \\
= \;& \{\ \mathit{substitution\ is\ conjunctive}\ \} \\
& \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
& \Rightarrow \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \\
= \;& \{\ \mathit{case\ analysis\ around\ evolution\ enabledness}\ \} \\
& (\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
& \quad \Rightarrow \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \ \ ) \wedge \\
& (\ \neg e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
& \quad \Rightarrow \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \ \ ) \\
= \;& \{\ \mathit{propagation\ over\ implication}\ \} \\
& (\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
& \quad \Rightarrow e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \ \ ) \wedge \\
& (\ \neg e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
& \quad \Rightarrow \neg e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \ \ ) \\
= \;& \{\ \mathit{property\ of\ evolution\ disabledness}\ (A.1)\ \} \\
& (\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
& \quad \Rightarrow e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p) \ \ ) \wedge \\
& (\ \neg e \wedge q \wedge p \\
& \quad \Rightarrow \neg e \wedge q \wedge p\ ) \\
= \;& \{\ \mathit{simplification}\ \} \\
& e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
& \quad \Rightarrow e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \wedge p)
\end{aligned}
$$

$$
\begin{array}{ll}
= & \{ \, \textit{case analysis around termination} \, \} \\
& (\; \mathrm{t}(e : \dot{X} = F(X)) \; \wedge \; e \wedge \\
& \quad \mathrm{wp}\,(e : \dot{X} = F(X), \; q) \quad \wedge \; \mathrm{wp}\,(e : \dot{X} = F(X), \; p) \\
& \qquad \Rightarrow \; e \; \wedge \; \mathrm{wp}\,(e : \dot{X} = F(X), \; q \; \wedge \; p) \quad) \wedge \\
& (\; \neg\mathrm{t}(e : \dot{X} = F(X)) \; \wedge \; e \wedge \\
& \quad \mathrm{wp}\,(e : \dot{X} = F(X), \; q) \quad \wedge \; \mathrm{wp}\,(e : \dot{X} = F(X), \; p) \\
& \qquad \Rightarrow \; e \; \wedge \; \mathrm{wp}\,(e : \dot{X} = F(X), \; q \; \wedge \; p) \quad) \\
= & \{ \, \textit{propagation over implication} \, \} \\
& (\; \mathrm{t}(e : \dot{X} = F(X)) \; \wedge \; e \wedge \\
& \quad \mathrm{wp}\,(e : \dot{X} = F(X), \; q) \quad \wedge \; \mathrm{wp}\,(e : \dot{X} = F(X), \; p) \\
& \qquad \Rightarrow \; \mathrm{t}(e : \dot{X} = F(X)) \; \wedge \; e \; \wedge \; \mathrm{wp}\,(e : \dot{X} = F(X), \; q \; \wedge \; p) \quad) \wedge \\
& (\; \neg\mathrm{t}(e : \dot{X} = F(X)) \; \wedge \; e \; \wedge \quad \mathrm{wp}\,(e : \dot{X} = F(X), \; q) \quad \wedge \; \mathrm{wp}\,(e : \dot{X} = F(X), \; p) \\
& \qquad \Rightarrow \; \neg\mathrm{t}(e : \dot{X} = F(X)) \; \wedge \; e \; \wedge \; \mathrm{wp}\,(e : \dot{X} = F(X), \; q \; \wedge \; p) \quad) \\
= & \{ \, \textit{property of non-termination} \; (A.2) \, \} \\
& (\; \mathrm{t}(e : \dot{X} = F(X)) \; \wedge \; e \wedge \\
& \quad \mathrm{wp}\,(e : \dot{X} = F(X), \; q) \quad \wedge \; \mathrm{wp}\,(e : \dot{X} = F(X), \; p) \\
& \qquad \Rightarrow \; \mathrm{t}(e : \dot{X} = F(X)) \; \wedge \; e \; \wedge \; \mathrm{wp}\,(e : \dot{X} = F(X), \; q \; \wedge \; p) \quad) \wedge \\
& (\; \neg\mathrm{t}(e : \dot{X} = F(X)) \; \wedge \; e \wedge \\
& \quad \mathit{false} \; \wedge \; \mathit{false} \\
& \qquad \Rightarrow \; \neg\mathrm{t}(e : \dot{X} = F(X)) \; \wedge \; e \; \wedge \; \mathit{false} \;) \\
= & \{ \, \textit{simplification} \, \} \\
& \mathrm{t}(e : \dot{X} = F(X)) \; \wedge \; e \wedge \\
& \mathrm{wp}\,(e : \dot{X} = F(X), \; q) \quad \wedge \; \mathrm{wp}\,(e : \dot{X} = F(X), \; p) \\
& \quad \Rightarrow \; \mathrm{t}(e : \dot{X} = F(X)) \; \wedge \; e \; \wedge \; \mathrm{wp}\,(e : \dot{X} = F(X), \; q \; \wedge \; p) \\
= & \{ \, \textit{property of termination} \; (A.3) \, \} \\
& \mathrm{t}(e : \dot{X} = F(X)) \; \wedge \; e \wedge \\
& q[\Phi(\Delta)/X] \; \wedge \; p[\Phi(\Delta)/X] \\
& \quad \Rightarrow \; \mathrm{t}(e : \dot{X} = F(X)) \; \wedge \; e \; \wedge \; (q \; \wedge \; p)[\Phi(\Delta)/X] \\
= & \{ \, \textit{substitution is conjunctive} \, \} \\
& \mathit{true}
\end{array}
$$

□

B.4 Continuity

Let q and p be predicates. For an action A to satisfy continuity the predicate $\mathrm{wp}\,(A, \; q) \;\; \vee \; \mathrm{wp}\,(A, \; p) \;\; \Leftrightarrow \; \mathrm{wp}\,(A, \; q \vee p)$ must hold. The differential action satisfies continuity.

Proof. Let e and q be predicates. If a differential action $e : \dot{X} = F(X)$ has no solution curve the weakest precondition $\mathrm{wp}\,(e : \dot{X} = F(X), \; q)$ is identically false. Hence, in this case the the differential action is trivially satisfies the continuity. The proof of continuity for a differential action $e : \dot{X} = F(X)$, for which the solution curve Φ exists, goes as follows.

$$
\begin{array}{ll}
 & \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \\
 & \Leftrightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \vee\ \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
= & \{\ \textit{case analysis}\ \} \\
 & (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \\
 & \quad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \vee\ \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \quad) \wedge \\
 & (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \vee\ \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
 & \quad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \quad) \\
= & \{\ \textit{case analysis around evolution enabledness}\ \} \\
 & (\ (\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \\
 & \qquad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \vee\ \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \quad) \wedge \\
 & \quad (\ \neg e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \\
 & \qquad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \vee\ \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \quad)\ ) \\
 & \wedge \\
 & (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \vee\ \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
 & \quad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \quad) \\
= & \{\ \textit{propagation over implication}\ \} \\
 & (\ (\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \\
 & \qquad \Rightarrow\ e \wedge (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \vee\ \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \quad)\ ) \wedge \\
 & \quad (\ \neg e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \\
 & \qquad \Rightarrow\ \neg e \wedge (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \vee\ \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \quad)\ )\ ) \\
 & \wedge \\
 & (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \vee\ \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
 & \quad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \quad) \\
= & \{\ \textit{property of evolution disabledness}\ (A.1)\ \} \\
 & (\ (\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \\
 & \qquad \Rightarrow\ e \wedge (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \vee\ \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \quad)\ ) \wedge \\
 & \quad (\ \neg e \wedge (q \vee p) \\
 & \qquad \Rightarrow\ \neg e \wedge (q \vee p)\ )\ ) \\
 & \wedge \\
 & (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \vee\ \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
 & \quad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \quad) \\
= & \{\ \textit{simplification}\ \} \\
 & (\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \\
 & \quad \Rightarrow\ e \wedge (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \vee\ \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \quad)\ ) \wedge \\
 & (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \vee\ \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
 & \quad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \quad) \\
= & \{\ \textit{case analysis around termination}\ \} \\
 & (\ (\ \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \\
 & \qquad \Rightarrow\ e \wedge (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \vee\ \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \quad)\ ) \wedge \\
 & \quad (\ \neg \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \\
 & \qquad \Rightarrow\ e \wedge (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \vee\ \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \quad)\ )\ ) \wedge \\
 & (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \quad \vee\ \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
 & \quad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \quad)
\end{array}
$$

$$
\begin{array}{ll}
= & \{\ \mathit{propagation\ over\ implication}\ \} \\
& (\ (\ \mathrm{t}(e : \dot{X} = F(X))\ \wedge\ e\ \wedge\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \\
& \qquad \Rightarrow\ \mathrm{t}(e : \dot{X} = F(X))\ \wedge\ e \wedge \\
& \qquad\qquad (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q)\ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p)\ \ )\ ) \wedge \\
& \quad (\ \neg\mathrm{t}(e : \dot{X} = F(X))\ \wedge\ e\ \wedge\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \\
& \qquad \Rightarrow\ \neg\mathrm{t}(e : \dot{X} = F(X))\ \wedge\ e \wedge \\
& \qquad\qquad (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q)\ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p)\ \ )\ )\ ) \wedge \\
& (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q)\ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
& \quad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p)\ \ ) \\
= & \{\ \mathit{property\ of\ non-termination}\ (A.2)\ \} \\
& (\ (\ \mathrm{t}(e : \dot{X} = F(X))\ \wedge\ e\ \wedge\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \\
& \qquad \Rightarrow\ \mathrm{t}(e : \dot{X} = F(X))\ \wedge\ e \wedge \\
& \qquad\qquad (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q)\ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p)\ \ )\ ) \wedge \\
& \quad (\ \neg\mathrm{t}(e : \dot{X} = F(X))\ \wedge\ e\ \wedge\ \mathit{false} \\
& \qquad \Rightarrow\ \neg\mathrm{t}(e : \dot{X} = F(X))\ \wedge\ e \wedge \\
& \qquad\qquad (\ \mathit{false} \vee \mathit{false}\ )\ )\ ) \wedge \\
& (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q)\ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
& \quad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p)\ \ ) \\
= & \{\ \mathit{simplification}\ \} \\
& (\ \mathrm{t}(e : \dot{X} = F(X))\ \wedge\ e\ \wedge\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \\
& \quad \Rightarrow\ \mathrm{t}(e : \dot{X} = F(X))\ \wedge\ e \wedge \\
& \qquad\quad (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q)\ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p)\ \ )\ ) \wedge \\
& (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q)\ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
& \quad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p)\ \ ) \\
= & \{\ \mathit{property\ of\ termination}\ (A.3)\ \} \\
& (\ \mathrm{t}(e : \dot{X} = F(X))\ \wedge\ e\ \wedge\ (q \vee p)[\Phi(\Delta)/X] \\
& \quad \Rightarrow\ \mathrm{t}(e : \dot{X} = F(X))\ \wedge\ e \wedge \\
& \qquad\quad (\ q[\Phi(\Delta)/X] \vee p[\Phi(\Delta)/X]\ )\ ) \wedge \\
& (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q)\ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
& \quad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p)\ \ ) \\
= & \{\ \mathit{substitution\ satisfies\ continuity}\ \} \\
& \mathrm{wp}\,(e : \dot{X} = F(X),\ q)\ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \\
& \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \\
= & \{\ \mathit{case\ analysis\ around\ evolution\ enabledness}\ \} \\
& (\ e\ \wedge\ (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q)\ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p)\ \ ) \\
& \quad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p)\ \ ) \wedge \\
& (\ \neg e\ \wedge\ (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q)\ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p)\ \ ) \\
& \quad \Rightarrow\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p)\ \ ) \\
= & \{\ \mathit{propagation\ over\ implication}\ \} \\
& (\ e\ \wedge\ (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q)\ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p)\ \ ) \\
& \quad \Rightarrow\ e\ \wedge\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p)\ \ ) \wedge \\
& (\ \neg e\ \wedge\ (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q)\ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p)\ \ ) \\
& \quad \Rightarrow\ \neg e\ \wedge\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p)\ \ )
\end{array}
$$

$$
\begin{array}{cl}
= & \{\ \mathit{property\ of\ evolution\ disabledness}\ (A.1)\ \} \\
& (\ e \wedge (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \ \ ) \\
& \quad \Rightarrow\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \ \ ) \wedge \\
& (\ \neg e \wedge (q \vee p) \\
& \quad \Rightarrow\ \neg e \wedge (q \vee p)\ ) \\
= & \{\ \mathit{simplification}\ \} \\
& e \wedge (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \ \ ) \\
& \Rightarrow\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \\
= & \{\ \mathit{case\ analysis\ around\ termination}\ \} \\
& (\ \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \\
& \ \ (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \ \ ) \\
& \quad \Rightarrow\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \ \ ) \wedge \\
& (\ \neg\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \\
& \ \ (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \ \ ) \\
& \quad \Rightarrow\ e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \ \ ) \\
= & \{\ \mathit{propagation\ over\ implication}\ \} \\
& (\ \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \\
& \ \ (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \ \ ) \\
& \quad \Rightarrow\ \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \ \ ) \wedge \\
& (\ \neg\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \\
& \ \ (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \ \ ) \\
& \quad \Rightarrow\ \neg\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \ \ ) \\
= & \{\ \mathit{property\ of\ non-termination}\ (A.2)\ \} \\
& (\ \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \\
& \ \ (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \ \ ) \\
& \quad \Rightarrow\ \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \ \ ) \wedge \\
& (\ \neg\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \\
& \ \ (\mathit{false} \vee \mathit{false}) \\
& \quad \Rightarrow\ \neg\mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathit{false}\ ) \\
= & \{\ \mathit{simplification}\ \} \\
& \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \\
& (\ \mathrm{wp}\,(e : \dot{X} = F(X),\ q) \ \ \vee \mathrm{wp}\,(e : \dot{X} = F(X),\ p) \ \ ) \\
& \Rightarrow\ \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \mathrm{wp}\,(e : \dot{X} = F(X),\ q \vee p) \\
= & \{\ \mathit{property\ of\ termination}\ (A.3)\ \} \\
& \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge \\
& (\ q[\Phi(\Delta)/X] \vee p[\Phi(\Delta)/X]\ ) \\
& \Rightarrow\ \mathrm{t}(e : \dot{X} = F(X)) \wedge e \wedge (q \vee p)[\Phi(\Delta)/X] \\
= & \{\ \mathit{substitution\ satisfies\ conjunctivity}\ \} \\
& \mathit{true}
\end{array}
$$

□

Turku Centre for Computer Science
Lemminkäisenkatu 14
FIN-20520 Turku
Finland

http://www.tucs.abo.fi

University of Turku
- **Department of Mathematical Sciences**

Åbo Akademi University
- **Department of Computer Science**
- **Institute for Advanced Management Systems Research**

Turku School of Economics and Business Administration
- **Institute of Information Systems Science**