

A preliminary proposal for Sugar2.0 semantics
“Truncated/Untruncated Semantics”
draft version –
do not distribute outside the Accellera FVTC

Cindy Eisner¹ Dana Fisman^{1,2} John Havlicek³
Yoad Lustig¹ Anthony McIsaac⁴ David Van Campenhout⁵

¹ IBM Haifa Research Lab ² Weizmann Institute of Science
³ Motorola, Inc. ⁴ STMicroelectronics, Ltd. ⁵ Verisity Design, Inc.

December 10, 2002

1 Introduction

We present a semantics for LTL plus an abort operator on finite and infinite paths. The abort operator “cuts the path” at the appearance of an abort signal, and has the truth value of a weakened version of the formula on the resulting (finite) path.

The abort operator has been presented before [3, 2]. However, both previous presentations had problems. The semantics of abort in [3] were simple, but the complexity of model checking them was non-elementary [1]. The semantics of abort in [2], while of acceptable complexity, were presented in terms of a complicated seven-way relation.

In this document, we present semantics for LTL plus an abort operator that combine the advantages of [3] and [2]. Our contribution is an elegant formulation of the semantics of abort, which, while equivalent to that of [2] (and thus of acceptable complexity), uses the intuitions of [3] to avoid the complicated context-based semantics presented in [2].

An important goal of our semantics is to provide elegant and intuitive semantics for simulation as well as for model checking. The problem is that traditional LTL semantics over finite paths are defined for maximal paths in the model. That is, if we evaluate a formula over a finite path under traditional LTL finite semantics, it is because the last state of the path has no successor in the model. In simulation, however, all paths are finite, while most are not maximal. Rather, they end simply because the simulation itself has ended. We term such a path a *truncated* path.

There are two conflicting views of what the semantics should be for truncated paths. In one view, which we term the *neutral view*, truncated paths should be treated in the same way as finite maximal paths. In this view, the formula $\mathbf{F}p$ on a truncated path should hold iff p occurs somewhere on the path.

Another view, suggested by Carl Pixley [?], is that we should be able to guarantee that a formula which has been shown to hold by model checking will not fail under simulation. Under this view, which we term the *weak view*, the formula $\mathbf{F}p$ on a

truncated path should always hold, because the fact that p has not yet been seen is not evidence that the formula does not hold on some extension of the path.

There is an intimate relation between the abort operator, intended to be used for hardware resets, and the problem of truncated vs. maximal paths. In particular, the weak view holds for any formula of the form $f \text{ abort } e$, where e is a special atomic proposition that holds at the point after which the simulation trace ends.

The intuitive definition of $f \text{ abort } b$ is “check if the formula f holds and if in the middle of the check the abort signal b is encountered – stop the check”. If the evaluation of f proceeds smoothly without b interrupting in the middle, then the truth value of $f \text{ abort } b$ is the same as f . What is the truth value of $f \text{ abort } b$ if b interrupted the evaluation of f ? We can decide that it is *true* or refine the abort operator and have two kinds of *abort*: abort_T and abort_F such that if b occurs before the evaluation of f completed the truth value is *true* if the original formula used abort_T and *false* if the original formula used abort_F . Note that having two kinds of *abort* is not really a design choice. The two kinds abort_T and abort_F are dual to each other. Thus, if we decide to have only abort_T , then the formula $\neg(\neg f \text{ abort}_T b)$ gives the exact semantics of $f \text{ abort}_F b$. The resulting intuitive semantics of $f \text{ abort}_F b$ reads “ f holds with no intervention of b in the middle” as if b happens before the evaluation of f completed, the result is *false*.

The formal definition of LTL augmented with abort operators extends the model under which formulas are evaluated with an additional context, which can be *weak*, *neutral* or *strong*. Under a weak context eventualities are not required to occur as paths are considered truncated (for instance, due to an abort_T signal). Under a neutral context formulas are evaluated the same way as in classical LTL, thus eventualities are required to occur. Under a strong context not only eventualities are required to occur but invariants are required to hold forever (force a path to be infinite), this semantics is the dual of the weak semantics and matches the semantics of the abort_F operator.

2 Syntax

The formulas of $\text{LTL}^{\text{abort}}$ are defined over a set \mathbf{B} of boolean expressions¹ as follows:

Definition 1. $\text{LTL}^{\text{abort}}$ formulas

- Every boolean expression is an $\text{LTL}^{\text{abort}}$ formula.
- If b is a boolean expression, f , f_1 , and f_2 are $\text{LTL}^{\text{abort}}$ formulas then the following are $\text{LTL}^{\text{abort}}$ formulas:
 - $\neg f$
 - $f_1 \wedge f_2$
 - $X! f$
 - $[f_1 U f_2]$
 - $f \text{ abort}_T b^2$

¹ We assume two designated boolean expressions T and F belong to \mathbf{B} .

² In [4, 3] we referred to the abort_T operator as *abort*.

Additional operators are defined as syntactic sugaring of the above operators:

- $f \vee g \stackrel{\text{def}}{=} \neg(\neg f \wedge \neg g)$
- $\mathbf{F} f \stackrel{\text{def}}{=} [\mathbf{T} \mathbf{U} f]$
- $\mathbf{G} f \stackrel{\text{def}}{=} \neg \mathbf{F} \neg f$
- $[f \mathbf{W} g] \stackrel{\text{def}}{=} [f \mathbf{U} g] \vee \neg[\mathbf{T} \mathbf{U} \neg f]$
- $f \text{ abort}_{\mathbf{F}} b \stackrel{\text{def}}{=} \neg(\neg f \text{ abort}_{\mathbf{T}} b)$
- $\mathbf{X} f \stackrel{\text{def}}{=} \neg(\mathbf{X}! \neg f)$

3 Semantics

The semantics of an $\text{LTL}^{\text{abort}}$ formula are defined with respect to finite or infinite words and a context which can be *weak*, *neutral* or *strong*. By abuse of the \models notation we use $\omega \models^- f$, $\omega \models f$ and $\omega \models^+ f$ to denote that f holds on ω under weak, neutral or strong interpretation, respectively. We assume a given alphabet Σ and a mapping from each letter in Σ to the set \mathbf{B} of boolean expressions the letter *satisfies*. We use the notation $\ell \models^{\mathbf{B}} b$ to say that the letter ℓ satisfies the boolean expression b . We assume that for the designated boolean expressions \mathbf{T} and \mathbf{F} , $\ell \models^{\mathbf{B}} \mathbf{T}$ and $\ell \not\models^{\mathbf{B}} \mathbf{F}$ (for every letter ℓ).

We use u, v, w and ω to denote (possibly empty) finite/infinite words. We use ℓ (possibly with subscripts) to denote letters. We denote the length of word ω as $|\omega|$. A finite word $\omega = \ell_0 \ell_1 \ell_2 \dots \ell_n$ has length $n + 1$, an infinite word has length ∞ and the empty word ϵ has length 0. We denote by $\omega^{i..}$ the suffix of ω starting at ℓ_i . That is, for every $i < |\omega|$, $\omega^{i..} = \ell_i \ell_{i+1} \dots \ell_n$ (or $\omega^{i..} = \ell_i \ell_{i+1} \dots$). We denote by $\omega^{i..j}$ the finite sequence of letters starting from ℓ_i and ending in ℓ_j . That is, for $j \geq i$, $\omega^{i..j} = \ell_i \ell_{i+1} \dots \ell_j$ and for $j < i$, $\omega^{i..j} = \epsilon$. We use ω^i as a shorthand for $\omega^{i..i}$ (thus $\omega^i = \ell_i$). We make use of an ‘‘overflow’’ for the indices of ω . That is, ω^j , $\omega^{j..}$, and $\omega^{j..k}$ are defined for $j \geq |\omega|$ as: $\omega^j = \omega^{j..} = \omega^{j..k} = \epsilon$. For example, in the semantics of $[f_1 \mathbf{U} f_2]$ under weak context, the k that is required to exist is not necessarily in the range of ω .

holds neutrally

1. $\omega \models b \iff |\omega| > 0$ and $\omega^0 \models^{\mathbf{B}} b$
2. $\omega \models \neg f \iff \omega \not\models f$
3. $\omega \models f_1 \wedge f_2 \iff \omega \models f_1$ and $\omega \models f_2$
4. $\omega \models \mathbf{X}! f \iff |\omega| > 1$ and $\omega^{1..} \models f$
5. $\omega \models [f_1 \mathbf{U} f_2] \iff$ there exists $k < |\omega|$ such that $\omega^{k..} \models f_2$, and for every $j < k$, $\omega^{j..} \models f_1$
6. $\omega \models f \text{ abort}_{\mathbf{T}} b \iff$ either $\omega \models f$ or there exists k such that $\omega^k \models b$ and for every $j < k$, $\omega^j \not\models b$ and $\omega^{0..k-1} \models^- f$

holds weakly

1. $\omega \models^- b \iff \text{either } |\omega| = 0 \text{ or } \omega^0 \models^{\mathbb{B}} b$
2. $\omega \models^- \neg f \iff \omega \not\models^+ f$
3. $\omega \models^- f_1 \wedge f_2 \iff \omega \models^- f_1 \text{ and } \omega \models^- f_2$
4. $\omega \models^- \mathbf{X}! f \iff \text{either } |\omega| \leq 1 \text{ or } \omega^{1..} \models^- f$
5. $\omega \models^- [f_1 \mathbf{U} f_2] \iff \text{there exists } k \text{ such that } \omega^{k..} \models^- f_2, \text{ and for every } j < k, \omega^{j..} \models^- f_1$
6. $\omega \models^- f \text{ abort.}_{\top} b \iff \text{either } \omega \models^- f \text{ or there exists } k \text{ such that } \omega^k \models b \text{ and for every } j < k, \omega^j \not\models b \text{ and } \omega^{0..k-1} \models^- f$

holds strongly

1. $\omega \models^+ b \iff |\omega| > 0 \text{ and } \omega^0 \models^{\mathbb{B}} b$
2. $\omega \models^+ \neg f \iff \omega \not\models^- f$
3. $\omega \models^+ f_1 \wedge f_2 \iff \omega \models^+ f_1 \text{ and } \omega \models^+ f_2$
4. $\omega \models^+ \mathbf{X}! f \iff |\omega| > 1 \text{ and } \omega^{1..} \models^+ f$
5. $\omega \models^+ [f_1 \mathbf{U} f_2] \iff \text{there exists } k \text{ such that } \omega^{k..} \models^+ f_2, \text{ and for every } j < k, \omega^{j..} \models^+ f_1$
6. $\omega \models^+ f \text{ abort.}_{\top} b \iff \text{either } \omega \models^+ f \text{ or there exists } k \text{ such that } \omega^k \models b \text{ and for every } j < k, \omega^j \not\models b \text{ and } \omega^{0..k-1} \models^- f$

4 Theorems on the semantics

Notations:

- We say that u is a *prefix* of v and denote $u \preceq v$ if there exists a word u' such that $uu' = v$.
- We say that w is an *extension* of v and denote $w \succeq v$ if there exists a word v' such that $vv' = w$.

Theorem 1 (strength relation theorem).

1. $w \models^+ f \implies w \models f$
2. $w \models f \implies w \models^- f$

Theorem 2 (prefix/extension theorem).

1. $v \models^+ f \iff \forall w \succeq v, w \models^+ f$
2. $v \models^- f \iff \forall u \preceq v, u \models^- f$

Proposition 1.

1. $\omega \models f \vee g \iff \omega \models f \text{ or } \omega \models g$
2. $\omega \models^+ f \vee g \iff \omega \models^+ f \text{ or } \omega \models^+ g$

$$3. \omega \models^- f \vee g \iff \omega \models^- f \text{ or } \omega \models^- g$$

Proposition 2.

1. $\omega \models Ff \iff \exists k < |\omega| \text{ s.t. } \omega^{k..} \models f$
2. $\omega \models^+ Ff \iff \exists k \text{ s.t. } \omega^{k..} \models^+ f$
3. $\omega \models^- Ff \iff \exists k \text{ s.t. } \omega^{k..} \models^- f$

Note: for every formula f and every **finite** word ω , $\omega \models^- Ff$.

Proposition 3.

1. $\omega \models Gf \iff \forall k < |\omega|, \omega^{k..} \models f$
2. $\omega \models^+ Gf \iff \forall k, \omega^{k..} \models^+ f$
3. $\omega \models^- Gf \iff \forall k, \omega^{k..} \models^- f$

Note: Gf holds strongly only on infinite paths.

Proposition 4.

1. $\omega \models [fUg] \iff \omega \models [\neg gW(\neg f \wedge \neg g)]$
2. $\omega \models^+ [fUg] \iff \omega \models^+ [\neg gW(\neg f \wedge \neg g)]$
3. $\omega \models^- [fUg] \iff \omega \models^- [\neg gW(\neg f \wedge \neg g)]$

Proposition 5.

1. $\omega \models f \text{ abort}_{.F} b \iff \omega \models f \text{ and } \forall k \text{ if } \omega^k \models^B b \text{ and } \forall j < k, \omega^j \not\models^B b \text{ then } \omega^{0..k-1} \models^+ f$
2. $\omega \models^+ f \text{ abort}_{.F} b \iff \omega \models^+ f \text{ and } \forall k \text{ if } \omega^k \models^B b \text{ and } \forall j < k, \omega^j \not\models^B b \text{ then } \omega^{0..k-1} \models^+ f$
3. $\omega \models^- f \text{ abort}_{.F} b \iff \omega \models^- f \text{ and } \forall k \text{ if } \omega^k \models^B b \text{ and } \forall j < k, \omega^j \not\models^B b \text{ then } \omega^{0..k-1} \models^+ f$

Proposition 6.

1. $\omega \models Xf \iff \omega \models \neg(X! \neg f)$
2. $\omega \models^+ Xf \iff \omega \models^+ X!f$
3. $\omega \models^- Xf \iff \omega \models^- X!f$

Note: under the weak and strong semantics, there is a single *next* operator which is its own dual, while under the neutral semantics, there are two different *next* operators, one of which is the dual of the other.

A Proofs

Lemma 1. *Let f be a formula in $\text{LTL}^{\text{abort}}$. Then both $\epsilon \models^- f$ and $\epsilon \not\models^+ f$.*

Proof. The proof is by induction on the structure of the formula. Most cases are easy to see, we show here the case where f is a formula of the form $g \text{ abort_T } b$.

$$\begin{aligned}
& \epsilon \not\models^+ g \text{ abort_T } b \\
\iff & \text{not}(\text{either } \epsilon \models^+ g \text{ or there exists } k \text{ such that } \epsilon^k \models b \text{ and } \epsilon^{0..k-1} \models^- g \text{ and } \forall j < k, \epsilon^j \not\models b) \\
\iff & \text{not}(\text{either } \epsilon \models^+ g \text{ or there exists } k \text{ such that } \epsilon \models b \text{ and } \epsilon \models^- g \text{ and } \forall j < k, \epsilon \not\models b) \\
\iff & [\epsilon \models b \text{ is FALSE}] \\
& \text{not}(\epsilon \models^+ g) \\
\iff & [\text{induction}] \\
& \text{not}(\text{FALSE}) \\
\iff & \text{TRUE}
\end{aligned}$$

□

Theorem 1 [strength relation theorem]

1. $w \models^+ f \implies w \models f$
2. $w \models f \implies w \models^- f$

Proof. By induction on the structure of the formula.

1. $f = b$
 - (a) $w \models^+ b \iff |\omega| > 0 \text{ and } \omega^0 \models^{\mathbf{B}} b \iff w \models b$
 - (b) $w \models b \iff |w| > 0 \text{ and } w^0 \models^{\mathbf{B}} b \implies |w| = 0 \text{ or } w^0 \models^{\mathbf{B}} b \iff w \models^- b$.
2. $f = \neg g$
 - (a) $w \models^+ \neg g \iff w \not\models^- g \implies [\text{induction}] w \not\models g \iff w \models \neg g$
 - (b) $w \models \neg g \iff w \not\models g \implies [\text{induction}] w \not\models^+ g \iff w \models^- \neg g$
3. $f = g \wedge h$
 - (a) $w \models^+ g \wedge h \iff w \models^+ g \text{ and } w \models^+ h \implies [\text{induction}] w \models g \text{ and } w \models h \iff w \models g \wedge h$
 - (b) $w \models g \wedge h \iff w \models g \text{ and } w \models h \implies [\text{induction}] w \models^- g \text{ and } w \models^- h \iff w \models^- g \wedge h$
4. $f = \mathbf{X!}g$
 - (a) $w \models^+ \mathbf{X!}g \iff |w| > 1 \text{ and } w^{1..} \models^+ g \implies [\text{induction}] |w| > 1 \text{ and } w^{1..} \models g \iff w \models \mathbf{X!}g$
 - (b) $w \models \mathbf{X!}g \iff |w| > 1 \text{ and } w^{1..} \models g \implies [\text{induction}] |w| > 1 \text{ and } w^{1..} \models^- g \implies |w| \leq 1 \text{ or } w^{1..} \models^- g \iff w \models^- \mathbf{X!}g$
5. $f = [g \mathbf{U} h]$

- (a) $w \models^{\pm} g \cup h \iff \exists k \text{ s.t. } w^{k\cdot} \models^{\pm} h \text{ and } \forall j < k, w^{j\cdot} \models^{\pm} g \iff$
 [if $k \geq |w|$, then w^k is empty, hence by Lemma 1 $w^{k\cdot} \not\models^{\pm} h$]
 $\exists k < |w| \text{ s.t. } w^{k\cdot} \models^{\pm} h \text{ and } \forall j < k, w^{j\cdot} \models^{\pm} g \implies$ [induction]
 $\exists k < |w| \text{ s.t. } w^{k\cdot} \models h \text{ and } \forall j < k, w^{j\cdot} \models g \iff w \models g \cup h$
- (b) $w \models g \cup h \iff \exists k < |w| \text{ s.t. } w^{k\cdot} \models h \text{ and } \forall j < k, w^{j\cdot} \models g \implies$ [induction]
 $\exists k < |w| \text{ s.t. } w^{k\cdot} \models h \text{ and } \forall j < k, w^{j\cdot} \models g \implies \exists k \text{ s.t. } w^{k\cdot} \models h \text{ and } \forall j <$
 $k, w^{j\cdot} \models g \iff w \models g \cup h$
6. $f = g \text{ abort}_{\top} b$
- (a) $w \models^{\pm} g \text{ abort}_{\top} b \iff$
 either $w \models^{\pm} g$ or $\exists k \text{ s.t. } w^k \models b \text{ and } w^{0..k-1} \models^{-} g \text{ and } \forall j < k, w^j \not\models b \implies$
 [induction]
 either $w \models g$ or $\exists k \text{ s.t. } w^k \models b \text{ and } w^{0..k-1} \models^{-} g \text{ and } \forall j < k, w^j \not\models b \iff$
 $w \models g \text{ abort}_{\top} b$
- (b) $w \models g \text{ abort}_{\top} b \iff$
 either $w \models g$ or $\exists k \text{ s.t. } w^k \models b \text{ and } w^{0..k-1} \models^{-} g \text{ and } \forall j < k, w^j \not\models b \implies$
 [induction]
 either $w \models^{-} g$ or $\exists k \text{ s.t. } w^k \models b \text{ and } w^{0..k-1} \models^{-} g \text{ and } \forall j < k, w^j \not\models b \iff$
 $w \models^{-} g \text{ abort}_{\top} b$

□

Theorem 2 [prefix/extension theorem]

1. $v \models^{\pm} f \iff \forall w \succeq v, w \models^{\pm} f$
2. $v \models^{-} f \iff \forall u \preceq v, u \models^{-} f$

Proof. The proof is by induction on the structure of the formula f .

1. $f = b$
 - (a) $v \models^{\pm} b$
 \iff [definition]
 $|v| > 0 \text{ and } v^0 \models^{\mathbb{B}} b$
 $\implies [w \succeq v \text{ implies } |w| \geq |v| > 0 \text{ and } w^0 = v^0]$
 forall $w \succeq v$: if $|w| > 0$ then $w^0 \models^{\mathbb{B}} b$
 \iff [definition]
 forall $w \succeq v$: $w \models^{\pm} b$
 - (b) $v \models^{-} b$
 \iff [definition]
 $|v| = 0 \text{ or } v^0 \models^{\mathbb{B}} b$
 $\implies [u \preceq v \text{ implies } |u| \leq |v| \text{ and, if } |u| > 0 \text{ then } u^0 = v^0]$
 forall $u \preceq v$: $|u| = 0 \text{ or } u^0 \models^{\mathbb{B}} b$

$$\iff \text{[definition]}$$

$$\forall u \preceq v : u \Vdash^- b$$

2. $f = \neg g$

(a) $\text{not}(\text{forall } w \succeq v : w \Vdash^+ \neg g)$

$$\iff$$

$$\text{exists } w \succeq v : \text{not}(w \Vdash^+ \neg g)$$

$$\iff \text{[definition]}$$

$$\text{exists } w \succeq v : w \Vdash^- g$$

$$\iff \text{[induction]}$$

$$\text{exists } w \succeq v : \text{forall } u \preceq w : u \Vdash^- g$$

$$\implies$$

$$v \Vdash^- g$$

$$\iff \text{[definition]}$$

$$\text{not}(v \Vdash^+ \neg g)$$

(b) $\text{not}(\text{forall } u \preceq v : u \Vdash^- \neg g)$

$$\iff$$

$$\text{exists } u \preceq v : \text{not}(u \Vdash^- \neg g)$$

$$\iff \text{[definition]}$$

$$\text{exists } u \preceq v : u \Vdash^+ g$$

$$\iff \text{[induction]}$$

$$\text{exists } u \preceq v : \text{forall } w \succeq u : w \Vdash^+ g$$

$$\implies$$

$$v \Vdash^+ g$$

$$\iff \text{[definition]}$$

$$\text{not}(v \Vdash^- \neg g)$$

3. $f = g \wedge h$

(a) $v \Vdash^+ g \wedge h$

$$\iff \text{[definition]}$$

$$v \Vdash^+ g \text{ and } v \Vdash^+ h$$

$$\iff \text{[induction]}$$

$$\text{forall } w \succeq v : w \Vdash^+ g \text{ and } \text{forall } w \succeq v : w \Vdash^+ h$$

$$\iff$$

$$\text{forall } w \succeq v : w \Vdash^+ g \text{ and } w \Vdash^+ h$$

$$\iff \text{[definition]}$$

$$\text{forall } w \succeq v : w \Vdash^+ g \wedge h$$

(b) Similar.

4. $f = \mathbf{X!}g$

(a) $v \Vdash^+ \mathbf{X!}g$

$$\iff \text{[definition]}$$

$$|v| > 1 \text{ and } v^{1..} \Vdash^+ g$$

$$\iff \text{[induction]}$$

- $|v| > 1$ and forall $w \succeq v^{1..}$: $w \Vdash^+ g$
 $\implies [w \succeq v \text{ implies } |w| \geq |v| \text{ and } w^{1..} \succeq v^{1..}]$
 forall $w \succeq v$: $|w| > 1$ and $w^{1..} \Vdash^+ g$
 \iff [definition]
 forall $w \succeq v$: $w \Vdash^+ \mathbf{X}!g$
- (b) $v \Vdash^- \mathbf{X}!g$
 \iff [definition]
 $|v| \leq 1$ or $v^{1..} \Vdash^- g$
 \iff [induction]
 $|v| \leq 1$ or forall $u \preceq v^{1..}$: $u \Vdash^- g$
 $\implies [u \preceq v \text{ implies } |u| \leq |v| \text{ and } u^{1..} \preceq v^{1..}]$
 forall $u \preceq v$: $|u| \leq 1$ or $u^{1..} \Vdash^+ g$
 \iff [definition]
 forall $u \preceq v$: $u \Vdash^- \mathbf{X}!g$
5. $f = [g\mathbf{U}h]$
- (a) $v \Vdash^+ [g\mathbf{U}h]$
 \iff [definition]
 there exists k such that $v^{k..} \Vdash^+ h$ and for all $j < k$, $v^{j..} \Vdash^+ g$
 \iff [induction]
 there exists k s.t. forall $w \succeq v^{k..}$: $w \Vdash^+ h$ and forall $j < k$ forall $w \succeq v^{j..}$:
 $w \Vdash^+ g$
 $\implies [w \succeq v \text{ implies } w^{k..} \succeq v^{k..} \text{ and } w^{j..} \succeq v^{j..}]$
 there exists k such that forall $w \succeq v$ both $w^{k..} \Vdash^+ h$ and forall $j < k$: $w^{j..} \Vdash^+ g$
 \implies
 forall $w \succeq v$: there exists k such that $w^{k..} \Vdash^+ h$ and forall $j < k$: $w^{j..} \Vdash^+ g$
 \iff [definition]
 forall $w \succeq v$: $w \Vdash^+ [g\mathbf{U}h]$
- (b) $v \Vdash^- [g\mathbf{U}h]$
 \iff [definition]
 there exists k such that $v^{k..} \Vdash^- h$ and for all $j < k$ $v^{j..} \Vdash^- g$
 \iff [induction]
 there exists k s.t. forall $u \preceq v^{k..}$: $u \Vdash^- h$ and forall $j < k$ forall $u \preceq v^{j..}$: $u \Vdash^- g$
 $\implies [u \preceq v \text{ implies } u^{k..} \preceq v^{k..} \text{ and } u^{j..} \preceq v^{j..}]$
 there exists k s.t. forall $u \preceq v$ both $u^{k..} \Vdash^- h$ and forall $j < k$: $u^{j..} \Vdash^- g$
 \implies
 forall $u \preceq v$: there exists k s.t. $u^{k..} \Vdash^- h$ and forall $j < k$ $u^{j..} \Vdash^- g$
 \iff [definition]
 forall $u \preceq v$: $u \Vdash^- [g\mathbf{U}h]$
6. $f = g \text{ abort_T } b$
- (a) $v \Vdash^+ g \text{ abort_T } b$
 \iff [definition]

either $v \models^+ g$ or
 there exists k s.t. $v^k \models b$ and for every $j < k$: $v^j \not\models b$ and $v^{0..k-1} \models^- g$
 \iff [induction]
 either forall $w \succeq v : w \models^+ g$ or
 there exists k s.t. $v^k \models b$ and for every $j < k$: $v^j \not\models b$ and $v^{0..k-1} \models^- g$
 $\implies [w \succeq v \text{ implies } w^j = v^j \text{ for } j < k \text{ and } w^{0..k-1} = v^{0..k-1}]$
 either forall $w \succeq v : w \models^+ g$ or forall $w \succeq v$:
 there exists k s.t. $w^k \models b$ and for every $j < k$: $w^j \not\models b$ and $w^{0..k-1} \models^- g$
 \implies
 forall $w \succeq v$: either $w \models^+ g$ or
 there exists k s.t. $w^k \models b$ and for every $j < k$: $w^j \not\models b$ and $w^{0..k-1} \models^- g$
 \iff [definition]
 forall $w \succeq v : w \models^+ g \text{ abort}_T b$
 (b) $v \models^- g \text{ abort}_T b$
 \iff [definition]
 either $v \models^- g$ or there exists k such that $v^k \models b$ and for every $j < k$ $v^j \not\models b$
 and $v^{0..k-1} \models^- g$
 \iff [induction]
 A: either
 i. forall $u \preceq v : u \models^- g$ or
 ii. there exists k such that $v^k \models b$ and for every $j < k$ $v^j \not\models b$ and forall
 $u \preceq v^{0..k-1} : u \models^- g$
 Let $u \preceq v$. If A.i. holds then we get $u \models^- g$, hence $u \models^- g \text{ abort}_T b$. Suppose
 now that A.ii. holds. [Note that $k < |v|$ (since $v^k \models b$)]. If $u \preceq v^{0..k-1}$, then [by
 the last part of A.ii.] we get $u \models^- g$, hence again $u \models^- g \text{ abort}_T b$. Otherwise,
 $v^{0..k} \preceq u$. Then $u^j = v^j$ for all $j \leq k$, and so A.ii. implies there exists k such
 that $u^k \models b$ and for every $j < k$: $u^j \not\models b$ and $u^{0..k-1} \models^- g$. From this we again
 get $u \models^- g \text{ abort}_T b$. Therefore, $A \implies$ forall $u \preceq v : u \models^- g \text{ abort}_T b$

□

References

1. R. Armoni, D. Bustan, O. Kupferman, and M. Y. Vardi. Aborts vs resets in linear temporal logic. In <http://www.cs.rice.edu/vardi/misc/abortreset.pdf>, 2002.
2. R. Armoni, L. Fix, A. Flaisher, R. Gerth, B. Ginsburg, T. Kanza, A. Landver, S. Mador-Haim, E. Singerman, A. Tiemeyer, M. Y. Vardi, and Y. Zbar. The ForSpec temporal logic: A new temporal property-specification language. In J.-P. Katoen and P. Stevens, editors, *Proc. 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 2280 of *Lecture Notes in Computer Science*. Springer, 2002.
3. C. Eisner and D. Fisman. Sugar 2.0 proposal presented to the Accellera Formal Verification Technical Committee. In http://www.haifa.il.ibm.com/projects/verification/sugar/Sugar_2.0_Accellera.ps, 2002.

4. C. Eisner, D. Fisman, M. Gordon, J. Havlicek, A. McIsaac, and D. Van Campenhout. Proposal for sugar 2.0 temporal layer formal syntax and semantics SEM_1. December 2002.