A Uniform Approach to Three-Valued Semantics for \( \mu \)-Calculus on Abstractions of Hybrid Automata

(Haifa Verification Conference 2008)

K. Bauer, R. Gentilini, and K. Schneider

University of Kaiserslautern

October 28, 2008
Overview

1. Preliminaries and Motivation
2. Generic Semantics for $L_\mu$ on Abstractions of Hybrid Automata
   ▶ Generic Preservation Result
3. Specializations
   ▶ May-/Must Abstractions
   ▶ DBB Abstractions
   ▶ Monotonicity Issues
4. Conclusions and Future Work
A hybrid automaton consists of

- Graph with finitely many locations

Example: (Heating controller)
Hybrid Automata (HA)

A hybrid automaton consists of
- Graph with finitely many locations
- Finitely many continuous variables changing value within a location according to differential rules

Example: (Heating controller)

\[ \dot{x} = -0.1 \quad \text{off} \]
\[ \dot{x} = 5 \quad \text{on} \]
A hybrid automaton consists of

- Graph with finitely many locations
- Finitely many continuous variables changing value within a location according to differential rules
- Initial Conditions, Location invariants, guards and resets for discrete transitions

Example: (Heating controller)
A hybrid automaton consists of

- Graph with finitely many locations
- Finitely many continuous variables changing value within a location according to differential rules
- Initial Conditions, Location invariants, guards and resets for discrete transitions

Example: (Heating controller)

- Heating is off: temperature $x$ falls with $\dot{x} = -0.1$
- Heating is on: temperature $x$ rises with $\dot{x} = 5$
Problem: (Decidability vs Expressiveness)

- In general, hybrid automata are undecidable w.r.t. reachability
- Decidability results only exist, when discrete and/or continuous dynamics are highly restricted
Problem: (Decidability vs Expressiveness)

- *In general, hybrid automata are undecidable w.r.t. reachability*
- *Decidability results only exist, when discrete and/or continuous dynamics are highly restricted*

Example:

- *Timed automata are decidable*
Problem: (Decidability vs Expressiveness)

- In general, hybrid automata are undecidable w.r.t. reachability
- Decidability results only exist, when discrete and/or continuous dynamics are highly restricted

Example:

- Timed automata are decidable
  - Adding skewed clocks makes timed automata undecidable
Problem: (Decidability vs Expressiveness)

- In general, hybrid automata are undecidable w.r.t. reachability
- Decidability results only exist, when discrete and/or continuous dynamics are highly restricted

Example:

- Timed automata are decidable
  - Adding skewed clocks makes timed automata undecidable
  ⇒ Approximative techniques are needed
Goal and Perspective

Goal:

Developing a framework for the automated reasoning on hybrid automata outside the decidability realm, featuring:

- combined overapprox./underapprox. analysis
  ⇒ safety certification + counterexamples
- ability to both prove and disprove reactive system properties expressed in $L_\mu$. 

A Uniform Approach to Three-Valued Semantics for $\mu$-Calculus on Abstractions of Hybrid Automata

K. Bauer, R. Gentilini, and K. Schneider
Goal and Perspective

Goal:
Developing a framework for the automated reasoning on hybrid automata outside the decidability realm, featuring:

▶ combined overapprox./underapprox. analysis
⇒ safety certification + counterexamples
▶ ability to both prove and disprove reactive system properties expressed in $L_\mu$.

Method:

▶ Three-valued generic semantics for $L_\mu$ ‘adaptable’ to proper abstraction frameworks
▶ Specialization of the generic semantics to different types of abstractions providing over-/underapprox.
  ▶ DBB abstractions
  ▶ Modal abstractions
3-Valued $L_{\mu}$ on HA-Abstractions

- $A = \langle R, R_0, \delta, e \rangle$ abstraction of $H$ encoding over- and underapproximation of the runs in $H$
- $AP$ finite set of atomic propositions
- $R$ a partition w.r.t. $l_{AP} : Q \rightarrow 2^{AP}$

Definition: ($L_{\mu}$ for generic HA-Abstractions)

$\phi \in AP$:

- $J_{\phi}^{K}(r) = \begin{cases} 
1 & \phi \in l_{AP}(r) \\
0 & \phi \not\in l_{AP}(r)
\end{cases}$
- $J_{\neg \phi}^{K} := \neg J_{\phi}^{K}$
- $J_{\phi \lor \psi}^{K} := J_{\phi}^{K} \lor J_{\psi}^{K}$
- $J_{\phi \land \psi}^{K} := J_{\phi}^{K} \land J_{\psi}^{K}$

Parametrized modal operators

- $\star \in \{ \langle \delta \rangle \phi, \langle e \rangle \phi, [\delta] \phi, [e] \phi, E(\phi U \psi), A(\phi U \psi) \}$
- $J_{\star}^{K}(r) = 1 \Rightarrow \forall x \in r: J_{\star}^{K}H(x) = 1$
- $J_{\star}^{K}(r) = 0 \Rightarrow \forall x \in r: J_{\star}^{K}H(x) = 0$
A $= \langle R, R_0, \delta, e \rangle$ abstraction of $H$ encoding over- and underapproximation of the runs in $H$

$AP$ finite set of atomic propositions

$R$ a partition w.r.t. $l_{AP} : Q \rightarrow 2^{AP}$

**Definition:** ($L_{\mu}$ for generic HA-Abstractions)

- $\phi \in AP$: $\llbracket \phi \rrbracket (r) = \begin{cases} 1 & \phi \in l_{AP}(r) \\ 0 & \phi \notin l_{AP}(r) \end{cases}$

- $\llbracket \neg \phi \rrbracket := \neg 3 \llbracket \phi \rrbracket$
  
- $\llbracket \phi \lor \psi \rrbracket := \llbracket \phi \rrbracket \lor 3 \llbracket \psi \rrbracket$,  
  $\llbracket \phi \land \psi \rrbracket := \llbracket \phi \rrbracket \land 3 \llbracket \psi \rrbracket$

**Parametrized modal operators**

- $\star \in \{ \langle \delta \rangle \phi, \langle e \rangle \phi, [\delta] \phi, [e] \phi, E(\phi \cup \psi), A(\phi \cup \psi) \}$:
  
- $\llbracket \star \rrbracket (r) = 1 \Rightarrow \forall x \in r : \llbracket \star \rrbracket_H(x) = 1$
  
- $\llbracket \star \rrbracket (r) = 0 \Rightarrow \forall x \in r : \llbracket \star \rrbracket_H(x) = 0$
Motivation

general Framework

Specializations
May/Must Abstractions
DBB-Abstractions

Conclusions

Definition: ($L_\mu$ for generic HA Abstractions)

Fixpoints: Let $\sigma \in \{\mu, \nu\}$

$$\llbracket \sigma Z. \phi \rrbracket := \llbracket apx_{\hat{k}}\sigma Z. \phi \rrbracket$$ satisfying

- $\hat{k}$ is the smallest index with

$$\llbracket apx_{\hat{k}}(\sigma Z. \phi) \rrbracket = \llbracket apx_{\hat{k}+1}(\sigma Z. \phi) \rrbracket$$

$A \models \phi :\iff \forall r \in R_0 : \llbracket \phi \rrbracket (r) = 1$

$A \not\models \phi :\iff \exists r \in R_0 : \llbracket \phi \rrbracket (r) = 0$. 

A Uniform Approach to Three-Valued Semantics for $\mu$-Calculus on Abstractions of Hybrid Automata

K. Bauer, R. Gentilini, and K. Schneider
Theorem: (Preservation)

Let $H$ be a hybrid automaton and $A$ be an abstraction of $H$. Then for all $\phi \in L_\mu$:

- $[\phi](r) = 1 \Rightarrow \forall x \in r : [\phi]_H(x) = 1$
- $[\phi](r) = 0 \Rightarrow \forall x \in r : [\phi]_H(x) = 0$
Preservation Results

Theorem: (Preservation)

Let $H$ be a hybrid automaton and $A$ be an abstraction of $H$. Then for all $\phi \in L_\mu$:

- $[\phi](r) = 1 \Rightarrow \forall x \in r : [\phi]_H(x) = 1$
- $[\phi](r) = 0 \Rightarrow \forall x \in r : [\phi]_H(x) = 0$

Proof: (Sketch)

By structural induction:

- boolean operators: obvious
- modal operators: by assumption
- fixpoint operators:
  $[\sigma Z. \phi] = [apx_k(\sigma Z. \phi)]$ for some $k \in \mathbb{N}$
  $\Rightarrow$ structural induction + monotonicity of fixpoints yield the claim
May/Must Abstractions

General Idea:
Adapt ideas for may/must transitions from discrete systems

- All transitions in $A$ are may-transitions
- $r \delta \rightarrow$ must $r' \ if \ all \ x \in r \ have \ a \ direct \ succ. \ x \Rightarrow x' \in r'$
- $r e \rightarrow$ must $r' \ if \ all \ x \in r \ have \ a \ succ. \ x e \rightarrow x' \in r'$
May/Must Abstractions

General Idea:
Adapt ideas for may/must transitions from discrete systems

Definition:
Let $A = \langle R, R_0, \delta, e \rangle$ be an abstraction. Then,

- All transitions in $A$ are may-transitions
- $r \xrightarrow{\delta} _{\text{must}} r'$ if all $x \in r$ have a direct succ. $x \leadsto x' \in r'$
- $r \xrightarrow{e} _{\text{must}} r'$ if all $x \in r$ have a succ. $x \xrightarrow{e} x' \in r'$

Lemma:
$A_{\text{must}} \leq_S T_H \leq_S A^*$
($A^*$ uses the transitive closure $\delta^*$ of $\delta$)
Semantics Completion of 3-valued $L_\mu$ on May/Must Abstractions:

Definition:

Let $A$ be a may/must abstraction. Then:

$\llangle \delta \rrangle \phi (r) = \begin{cases} 
1 & \exists r \xrightarrow{\delta} \text{must } r' : r' \text{ satisfies } \phi \\
0 & \nexists r \xrightarrow{\delta^*} r' : r' \text{ satisfies } \phi \\
\perp & \text{else}
\end{cases}$

$\llangle e \rrangle \phi (r) = \begin{cases} 
1 & \exists r \xrightarrow{e} \text{must } r' : r' \text{ satisfies } \phi \\
0 & \nexists r \xrightarrow{e} r' : r' \text{ satisfies } \phi \\
\perp & \text{else}
\end{cases}$

$a \in \{ e, \delta \}: \llbracket [a] \phi \rrbracket = \llbracket \neg (\llangle a \rrangle \neg \phi) \rrbracket$
Semantics Completion of 3-valued $L_\mu$ on May/Must Abstractions:

**Definition:**

Let $A$ be a may/must abstraction. Then:

$$\begin{align*}
\llbracket E(\phi U \psi) \rrbracket(r) &= \begin{cases} 
1 & \exists r \xrightarrow{\text{must}} r' \text{ satisfying } \phi U \psi \\
0 & \forall \text{ may-paths } \phi U \psi \text{ can be disproven} \\
\bot & \text{else}
\end{cases} \\
\llbracket A(\phi U \psi) \rrbracket(r) &= \begin{cases} 
1 & \text{all may-paths satisfy } \phi U \psi \\
0 & \exists r \xrightarrow{\text{must}} r' \text{ not satisfying } \phi U \psi \\
\bot & \text{else}
\end{cases}
\end{align*}$$
Corollary: (Preservation)

Let $H$ be a hybrid automaton and $A$ be a may/must abstraction of $H$. Then for all $\phi \in L_\mu$:

- $A \models \phi \Rightarrow H \models \phi$
- $A \not\models \phi \Rightarrow H \not\models \phi$

Remark:

May/must abstractions do not provide monotonicity results
Example: Heating Controller

\[ x = 20 \rightarrow \begin{cases} \dot{x} = -0.1 & \text{off} \\ \dot{x} = 5 & \text{on} \end{cases} \]

\[ x > 22, x' = x \]

\[ x < 20, x' = x \]

\[ \mu\text{-calculus formula: } \phi := \mu Z.(\text{on} \times [22, 24]) \lor \Diamond Z \]
Example: Heating Controller

\[ x = 20 \rightarrow \dot{x} = -0.1 \text{ off} \]

\[ x > 22, \dot{x} = x \text{ on} \]

\[ x < 20, \dot{x} = x \]

\[ \mu\text{-calculus formula: } \phi := \mu Z. (\text{on} \times [22, 24]) \lor \diamondsuit Z \]

Abstraction:
Example: Heating Controller

\[ x = 20 \rightarrow \dot{x} = -0.1 \text{ off} \]
\[ x < 20, x' = x \]
\[ x > 22, x' = x \]
\[ x < 24, x' = 5 \text{ on} \]

\[ \mu \text{-calculus formula: } \phi := \mu Z. (\text{on} \times [22, 24]) \lor \Diamond Z \]

Abstraction:
\[ A \models_3 \phi = 1 \]
\[ \Rightarrow H \models \phi = 1 \]
Example: Heating Controller

$\mu$-calculus formula: $\phi := \mu Z . (\text{on} \times [22, 24]) \lor \Diamond Z$

Refinement:
Example: Heating Controller

\[
\begin{align*}
\mu\text{-calculus formula: } & \phi := \mu Z.(\text{on} \times [22, 24]) \lor \Box Z \\
\text{Refinement: } & A \models_3 \phi = \bot
\end{align*}
\]
Definition: (Discrete Bounded Bisimulation)

Let $H$ be a hybrid automaton with state space $Q$. Let $P$ be a partition of $Q$.

\[ \equiv_0 \in Q \times Q \text{ is the max. relation on } Q \text{ s.t. for all } p \equiv_0 q:\]

- $[p]_P = [q]_P \text{ and } p \in Q_0 \iff q \in Q_0$
- $\forall p \xrightarrow{\delta} p' \exists q' : p' \equiv_0 q' \land q \xrightarrow{\delta} q'$
- $\forall q \xrightarrow{\delta} p' \exists p' : p' \equiv_0 q' \land p \xrightarrow{\delta} p'$
Definition: (Discrete Bounded Bisimulation)

Let \( H \) be a hybrid automaton with state space \( Q \). Let \( P \) be a partition of \( Q \).

\[\equiv_n \in Q \times Q \text{ is the max. relation on } Q \text{ s.t. for all } p \equiv_n q:\]

\[\begin{align*}
\triangleright & \; p \equiv_{n-1} q \\
\triangleright & \; \forall p \xrightarrow{\delta} p' \exists q' : p' \equiv_n q' \land q \xrightarrow{\delta} q' \\
& \forall q \xrightarrow{\delta} p' \exists p' : p' \equiv_n q' \land p \xrightarrow{\delta} p' \\
\triangleright & \; \forall p \xrightarrow{e} p' \exists q' : p' \equiv_{n-1} q' \land q \xrightarrow{e} q' \\
& \forall q \xrightarrow{e} q' \exists p' : p' \equiv_{n-1} q' \land p \xrightarrow{e} p'
\end{align*}\]

The relation \( \equiv_n \) is called \( n \)-DBB equivalence.
Semantics Completion of three-valued $L_\mu$:

**Definition:**

Let $H \equiv_n$ be an $n$-DBB abstraction. Then:

1. $\llangle \delta \rrangle \equiv_n ([x] \equiv_n) = 1$ iff 
   \[ \exists [x] \equiv_n \xrightarrow{\delta} [x'] \equiv_n : [x'] \equiv_n \text{satisfies } \phi \]

2. $\llangle \delta \rrangle \equiv_n ([x] \equiv_n) = 0$ iff 
   \[ \nexists [x] \equiv_n \xrightarrow{\delta^*} [x'] \equiv_n : [x'] \equiv_n \text{satisfies } \phi \]

3. $\llangle [\delta] \phi \rrangle \equiv_n = \llangle \neg (\llangle \delta \rrangle \neg \phi) \rrangle \equiv_n$
Semantics Completion of three-valued $L_\mu$:

**Definition:**

Let $H \equiv_n$ be an $n$-DBB abstraction. Then:

- $\llangle e \phi \rrangle \equiv_n ([x] \equiv_n) = 1$ iff
  - $\exists [x] \equiv_n \overset{e}{\rightarrow} [x'] \equiv_n : [x'] \equiv_{n-1}$ satisfies $\phi$
- $\llangle e \phi \rrangle \equiv_n ([x] \equiv_n) = 0$ iff
  - $\not\exists [x] \equiv_n \overset{e}{\rightarrow} [x'] \equiv_n : [x'] \equiv_{n-1}$ satisfies $\phi$
- $\llangle e \phi \rrangle \equiv_n ([x] \equiv_n) = \perp$ else
- $[[e] \phi] \equiv_n = [[\neg(e) \neg \phi]] \equiv_n$
Semantics Completion on DBB-Abs.

Semantics Completion of three-valued $L_{\mu}$:

**Definition:**

Let $H \equiv n$ be an $n$-DBB abstraction. Then:

- For $\llbracket E(\phi U \psi) \rrbracket \equiv n$:
  \[
  \llbracket E(\phi U \psi) \rrbracket \equiv n([x] \equiv n) = 1 \iff
  \exists [x] \equiv n \xrightarrow{\delta^*} [x'] \equiv n \text{ satisfying } \phi U \psi \text{ in } H \equiv n
  \]
  or
  \[
  \exists [x] \equiv n \xrightarrow{\delta^*} [x'] \equiv n \xrightarrow{e} [x''] \equiv n-1 \text{ satisfying } \phi \text{ on the first part and } [x''] \equiv n-1 \text{ satisfying } E(\phi U \psi)
  \]
  \[
  \llbracket E(\phi U \psi) \rrbracket \equiv n([x] \equiv n) = 0 \iff
  \forall \text{ paths in } H \equiv n \phi U \psi \text{ can be disproven}
  \]
  \[
  \llbracket E(\phi U \psi) \rrbracket \equiv n([x] \equiv n) = \bot \text{ otherwise}
  \]
Semantics Completion of three-valued $L_\mu$:

**Definition:**

Let $H\equiv_n$ be an $n$-DBB abstraction. Then:

- For $[[A(\phi U \psi)]]\equiv_n$:
  
  $[[A(\phi U \psi)]]\equiv_n([x]\equiv_n) = 1$ iff
  
  all paths in $H\equiv_n$ starting in $[x]\equiv_n$ satisfy $\phi U \psi$

  $[[A(\phi U \psi)]]\equiv_n([x]\equiv_n) = 0$ iff

  - $\exists [x]\equiv_n \delta^* \leadsto [x']\equiv_n$ not satisfying $\phi U \psi$ in $H\equiv_n$ or
  - $\exists [x]\equiv_n \delta^* \leadsto [x']\equiv_n \xrightarrow{e} [x'']\equiv_n$ satisfying $\phi$ on the first part and $[x'']\equiv_{n-1}$ not satisfying $A\phi U \psi$

  $[[A(\phi U \psi)]]\equiv_n([x]\equiv_n) = \bot$ otherwise
Corollary: (Preservation)

Let $H$ be a hybrid automaton and $H_{≡ n}$ be an $n$-DBB abstraction of $H$. Then for all $ϕ ∈ L_µ$:

- $H_{≡ n} ⊨ ϕ \Rightarrow H ⊨ ϕ$
- $H_{≡ n} \not⊨ ϕ \Rightarrow H \not⊨ ϕ$
Preservation Results for DBB-Abs.

Corollary: (Preservation)

Let $H$ be a hybrid automaton and $H_{\equiv n}$ be an $n$-DBB abstraction of $H$. Then for all $\phi \in L_\mu$:

- $H_{\equiv n} \models \phi \Rightarrow H \models \phi$
- $H_{\equiv n} \not\models \phi \Rightarrow H \not\models \phi$

Theorem: (Monotonicity)

Let $H_{\equiv n}$ and $H_{\equiv k}$, $n > k$, be DBB abstractions. Then for all $\phi \in L_\mu$ and all $x$ in the state space of $H$:

- $[\phi]_{\equiv k}([x]_{\equiv k}) = 1 \Rightarrow [\phi]_{\equiv n}([x]_{\equiv n}) = 1$
- $[\phi]_{\equiv k}([x]_{\equiv k}) = 0 \Rightarrow [\phi]_{\equiv n}([x]_{\equiv n}) = 0$
Example: Waterlevel Controller

\[ y \leq 10 \]
\[ \dot{x} = 1 \]
\[ \dot{y} = 1 \]
\[ \text{shut} \]
\[ y = 10 \]
\[ x_1 \geq 0 \]
\[ \dot{x} = -1 \]
\[ \dot{y} = -2 \]
\[ \text{open} \]

\( \phi = \mu Z.r \vee \Diamond Z \)
\( r = \text{shut} \times [0, 6] \times \{10\} \)

1-DBB Abstraction:
Example: Waterlevel Controller

\[ y \leq 10 \]
\[ x = 1 \]
\[ y = 1 \]
\[ \text{shut} \]
\[ x_1 \geq 0 \]
\[ \dot{x} = -1 \]
\[ \dot{y} = -2 \]
\[ \text{open} \]

\[ \mu\text{-calculus formula:} \]
\[ \phi = \mu Z.r \lor \Diamond Z \]
\[ r = \text{shut} \times [0, 6] \times \{10\} \]

1-DBB Abstraction: \( A \models_3 \phi = \bot \)
Example: Waterlevel Controller

\[
\begin{align*}
\mu\text{-calculus formula:} & \quad \phi = \mu Z.r \lor \Diamond Z \\
& \quad r = \text{shut} \times [0,6] \times \{10\}
\end{align*}
\]

2-DBB Abstraction:
Example: Waterlevel Controller

$\mu$-calculus formula:
$\phi = \mu Z.r \lor \Diamond Z$
$r = \text{shut} \times [0, 6] \times \{10\}$

2-DBB Abstraction: $A \models_3 \phi = 0 \Rightarrow H \models \phi = 0$
Conclusions:

- A parametrized three-valued interpretation of $L_{\mu}$ has been developed
- Preservation results have been proved
  ⇒ Safety certification + counterexamples
- Different applications for the general framework have been provided:
  - May/must abstractions
  - DBB abstractions

Future Work

- Development of a three-valued model-checking tool for hybrid automata
- Property driven abstraction refinements
- …
A Uniform Approach to Three-Valued Semantics for $\mu$-Calculus on Abstractions of Hybrid Automata

K. Bauer, R. Gentilini, and K. Schneider
Discrete time frameworks: $\U$-Operator redundant

- $E(\phi \U \psi) = \mu Z. \psi \lor \phi \land \lozenge Z$
- $A(\phi \U \psi) = \mu Z. \psi \lor \phi \land \square Z$

Continuous time frameworks: $\U$-operator not redundant
The $\mu$-Operator on Hybrid Automata

Discrete time frameworks: $\mu$-Operator redundant

- $E(\phi U \psi) = \mu Z. \psi \lor \phi \land \Diamond Z$

- $A(\phi U \psi) = \mu Z. \psi \lor \phi \land \Box Z$

Continuous time frameworks: $\mu$-operator not redundant

Example:

$x = 0 \rightarrow \dot{x} = 1$

Lemma: In the setting of hybrid automata the language $L_\mu$ with the temporal operators $E(\phi U \psi)$ and $A(\phi U \psi)$ is strictly more expressive than $L_\mu$ without these operators.
The $\mathbf{U}$-Operator on Hybrid Automata

Discrete time frameworks: $\mathbf{U}$-Operator redundant

- $E(\phi \mathbf{U} \psi) = \mu Z. \psi \lor \phi \land \Diamond Z$
- $A(\phi \mathbf{U} \psi) = \mu Z. \psi \lor \phi \land \Box Z$

Continuous time frameworks: $\mathbf{U}$-operator not redundant

Example:

$$\phi := E(x < 2) \mathbf{U} (x = 3)$$
The $\mathbf{U}$-Operator on Hybrid Automata

Discrete time frameworks: $\mathbf{U}$-Operator redundant

- $E(\phi \mathbf{U} \psi) = \mu Z.\psi \lor \phi \land \lozenge Z$
- $A(\phi \mathbf{U} \psi) = \mu Z.\psi \lor \phi \land \square Z$

Continuous time frameworks: $\mathbf{U}$-operator not redundant

Example:

$x = 0 \rightarrow \dot{x} = 1$

$\phi := E(x < 2) \mathbf{U} (x = 3)$
$\psi := \mu Z.(x = 3) \lor (x < 2) \land \lozenge Z$
The $U$-Operator on Hybrid Automata

Discrete time frameworks: $U$-Operator redundant

$E(\phi U \psi) = \mu Z.\psi \lor \phi \land \Diamond Z$

$A(\phi U \psi) = \mu Z.\psi \lor \phi \land \Box Z$

Continuous time frameworks: $U$-operator not redundant

Example:

$x = 0 \rightarrow \dot{x} = 1$

$\phi := E(x < 2) U (x = 3)$

$\psi := \mu Z.(x = 3) \lor (x < 2) \land \Diamond Z$

Lemma:

In the setting of hybrid automata the language $L_\mu$ with the temporal operators $E(\phi U \psi)$ and $A(\phi U \psi)$ is strictly more expressive than $L_\mu$ without these operators.

K. Bauer, R. Gentilini, and K. Schneider
Let the modal operator $\langle \delta \rangle$ satisfy:
$$\models \langle \delta \rangle \phi(r) = 1 \iff \text{a direct successor of } r \text{ satisfies } \phi \, (*)$$

**Theorem: (Redundancy)**

Let $H$ be a hybrid automaton and $A$ be an abstraction of $H$ satisfying $(*)$. Then for all $\phi, \psi \in L_\mu$:

1. $A \models \mu Z. \psi \lor \phi \land \Diamond Z \implies H \models E(\phi U \psi)$
   $$A \not\models \mu Z. \psi \lor \phi \land \Diamond Z \implies H \not\models E(\phi U \psi)$$

2. $A \models \mu Z. \psi \lor \phi \land \Box Z \implies H \models A(\phi U \psi)$
   $$A \not\models \mu Z. \psi \lor \phi \land \Box Z \implies H \not\models A(\phi U \psi)$$
Let the modal operator $\langle \delta \rangle$ satisfy:

$\llbracket \langle \delta \rangle \phi \rrbracket (r) = 1 \iff$ a direct successor of $r$ satisfies $\phi$  

(*)

**Theorem: (Redundancy)**

Let $H$ be a hybrid automaton and $A$ be an abstraction of $H$ satisfying (*). Then for all $\phi, \psi \in L_{\mu}$:

1. $A \models \mu Z. \psi \lor \phi \land \diamond Z \Rightarrow H \models E(\phi \mathcal{U} \psi)$
   $A \not\models \mu Z. \psi \lor \phi \land \diamond Z \Rightarrow H \not\models E(\phi \mathcal{U} \psi)$
2. $A \models \mu Z. \psi \lor \phi \land \Box Z \Rightarrow H \models A(\phi \mathcal{U} \psi)$
   $A \not\models \mu Z. \psi \lor \phi \land \Box Z \Rightarrow H \not\models A(\phi \mathcal{U} \psi)$

**Corollary:**

- For may/must abstractions the $\mathcal{U}$-operator is redundant
- For DBB-abstractions the $\mathcal{U}$-operator is redundant