Appendix A: \(\mathcal{L}_{\mathbb{R}_x}\)-Formulas and \(\delta\)-Decidability

We will use a logical language over the real numbers that allows arbitrary computable real functions \([1]\). We write \(\mathcal{L}_{\mathbb{R}_x}\) to represent this language. Intuitively, a real function is computable if it can be numerically simulated up to an arbitrary precision. For the purpose of this paper, it suffices to know that almost all the functions that are needed in describing hybrid systems are Type 2 computable, such as polynomials, exponentiation, logarithm, trigonometric functions, and solution functions of Lipschitz-continuous ordinary differential equations.

More formally, \(\mathcal{L}_{\mathbb{R}_x} = (\mathcal{F}, >)\) represents the first-order signature over the reals with the set \(\mathcal{F}\) of computable real functions, which contains all the functions mentioned above. Note that constants are included as 0-ary functions. \(\mathcal{L}_{\mathbb{R}_x}\)-formulas are evaluated in the standard way over the structure \(\mathbb{R}_x = (\mathbb{R}, \mathcal{F}_{\mathbb{R}}, >)\).

It is not hard to see that we can put any \(\mathcal{L}_{\mathbb{R}_x}\)-formula in a normal form, such that its atomic formulas are of the form \(t(x_1, ..., x_n) > 0\) or \(t(x_1, ..., x_n) \geq 0\), with \(t(x_1, ..., x_n)\) composed of functions in \(\mathcal{F}\). To avoid extra preprocessing of formulas, we can explicitly define \(\mathcal{L}_\delta\)-formulas as follows.

**Definition 1 (\(\mathcal{L}_{\mathbb{R}_x}\)-Formulas).** Let \(\mathcal{F}\) be a collection of computable real functions. We define:

\[
t := x \mid f(t(x)), \text{ where } f \in \mathcal{F} \quad \text{(constants are 0-ary functions)};
\]

\[
\varphi := t(x) > 0 \mid t(x) \geq 0 \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \exists x_i \varphi \mid \forall x_i \varphi.
\]

In this setting \(\neg \varphi\) is regarded as an inductively defined operation which replaces atomic formulas \(t > 0\) with \(-t \geq 0\), atomic formulas \(t \geq 0\) with \(-t > 0\), switches \(\land\) and \(\lor\), and switches \(\forall\) and \(\exists\).

**Definition 2 (Bounded \(\mathcal{L}_{\mathbb{R}_x}\)-Sentences).** We define the bounded quantifiers \(\exists^{[u,v]}\) and \(\forall^{[u,v]}\) as:

\[
\exists^{[u,v]} x. \varphi := \exists x.(u \leq x \land x \leq v \land \varphi) \quad \text{and} \quad \forall^{[u,v]} x. \varphi := \forall x.(u \leq x \land x \leq v \rightarrow \varphi)
\]

where \(u \leq v\) denote \(\mathcal{L}_{\mathbb{R}_x}\) terms, whose variables only contain free variables in \(\varphi\) excluding \(x\). A bounded \(\mathcal{L}_{\mathbb{R}_x}\)-sentence is

\[
Q_1^{[u_1,v_1]} x_1 \cdots Q_n^{[u_n,v_n]} x_n \psi(x_1, ..., x_n),
\]

where \(Q_i^{[u_i,v_i]}\) are bounded quantifiers, and \(\psi(x_1, ..., x_n)\) is quantifier-free.

**Definition 3 (\(\delta\)-Variants).** Let \(\delta \in \mathbb{Q}^+ \cup \{0\}\), and \(\varphi\) an \(\mathcal{L}_{\mathbb{R}_x}\)-formula

\[
\varphi := Q_1^{[t_1]} x_1 \cdots Q_n^{[t_n]} x_n \psi|_{t_i(x,y) > 0; t_j(x,y) \geq 0},
\]

where \(i \in \{1, ..., k\}\) and \(j \in \{k+1, ..., m\}\). The \(\delta\)-weakening \(\varphi^\delta\) of \(\varphi\) is defined as the result of replacing each atom \(t_i > 0\) by \(t_i > -\delta\) and \(t_j \geq 0\) by \(t_j \geq -\delta\): \(\varphi^\delta := Q_1^{[t_1]} x_1 \cdots Q_n^{[t_n]} x_n \psi|_{t_i(x,y) > -\delta; t_j(x,y) \geq -\delta}\).

It is clear that \(\varphi \rightarrow \varphi^\delta\) (see [2]).
In [3], we have proved that the following $\delta$-decision problem is decidable, which is the basis of our framework.

**Theorem 1 ($\delta$-Decidability [3]).** Let $\delta \in \mathbb{Q}^+$ be arbitrary. There is an algorithm which, given any bounded $\mathcal{L}_{\mathcal{R},\mathcal{F}}$-sentence $\varphi$, correctly returns one of the following two answers:

- $\delta$-True: $\varphi^\delta$ is true.
- False: $\varphi$ is false.

When the two cases overlap, either answer is correct.

The following theorem states the (relative) complexity of the $\delta$-decision problem. A bounded $\Sigma_n$ sentence is a bounded $\mathcal{L}_{\mathcal{R},\mathcal{F}}$-sentence with $n$ alternating quantifier blocks starting with $\exists$.

**Theorem 2 (Complexity [2]).** Let $S$ be a class of $\mathcal{L}_{\mathcal{R},\mathcal{F}}$-sentences, such that for any $\varphi$ in $S$, the terms in $\varphi$ are in Type 2 complexity class $\mathcal{C}$. Then, for any $\delta \in \mathbb{Q}^+$, the $\delta$-decision problem for bounded $\Sigma_n$-sentences in $S$ is in $(\Sigma^P_\delta)^\mathcal{C}$.

Basically, the theorem says that increasing the number of quantifier alternations will in general increase the complexity of the problem, unless $P = NP$ (recall that $\Sigma^P_0 = P$ and $\Sigma^P_1 = NP$). This result can specialized for specific families of functions. For example, with polynomially-computable functions, the $\delta$-decision problem for bounded $\Sigma_n$-sentences is $(\Sigma^P_\delta)$-complete. For more details and results we again point the interested reader to [2].

**References**

Appendix B: BCF Model in dReach

As an example of dReach’s modeling language, we report below the actual dReach file for one of the BCF models (Run#7) analyzed in the paper.

```dreach
#define EPI_TVP 1.4506
#define EPI_TV1M 60.0
#define EPI_TV2M 1150.0
#define EPI_TWP 200.0
#define EPI_TW1M 60.0
#define EPI_TW2M 15.0
#define EPI_TSI 2.7342
#define EPI_TSO2 16.0
#define EPI_TFI 0.11
#define EPI_TD1 400
#define EPI_TD2 6.0
#define EPI_TSO1 30.0181
#define EPI_TS1 1.9957
#define EPI_TSI 2.7342
#define EPI_TW1NF 0.07
#define EPI_THV 0.3
#define EPI_THVINF 0.006
#define EPI_THW 0.13
#define EPI_THWINF 0.006
#define EPI_US 0.9087
#define EPI_THU 0.13
#define EPI_TSO 0.13
#define EPI_KWS 65.0
#define EPI_KS 2.0994
#define EPI_KSO 2.0458
#define EPI_KWU 0.03
#define EPI_KUS 0.9087
#define EPI_KU 0.0
#define EPI_USO 0.65
#define jfi1 0.0
#define jso1 (u/EPI_TD1)
#define jsi1 0.0
#define jfi2 0.0
#define jso2 (u/EPI_TD2)
#define jsi2 0.0
#define jfi3 0.0
#define jso3 1.0/(EPI_TXS1+((EPI_TXS2 - EPI_TXS1)*(1/(1+exp(-2*EPI_KSO*(u - EPI_USO))))))
#define jsi3 (0 - (w * s)/EPI_TSI)
#define jfi4 (0 - v * (u - EPI_THV) * (EPI_UU - u)/EPI_TFI)
#define jso4 1.0 / (EPI_TXS0 + (1/(1+exp(-2*EPI_KSO*(u - EPI_USO)))))
#define jsi4 (0 - (w * s)/EPI_TSI)
#define stim 1.0

[0, 2.0] u;
[0, 2.0] v;
[0, 2.0] w;
[0, 2.0] s;
[0, 1] tau;
[0, 1] time;

{mode 1;
  invt: (u >= 0);
  (u <= 0.006);
  (v >= 0);
  (w >= 0);
  (s >= 0);
  (tau >= 0);
flow: d/dt[tau] = 1.0;
  d/dt[u] = (stim - jfi1) - (jso1 + jsi1);

```
\[
\begin{align*}
d/dt[w] &= ((1.0 - (u/EPI_TWINF) - w)/(EPI_TW1M + (EPI_TW2M - EPI_TW1M) * (1/(1+exp(-2*EPI_KWM*(u - EPI_UWM)))))); \\
d/dt[v] &= ((1.0 - v)/EPI_TV1M); \\
d/dt[s] &= (((1/(1+exp( -2 * EPI_KS * (u - EPI_US) ))) - s)/EPI_TS1); \\
\text{jump:} \\
(u >= 0.006) \Rightarrow @2 \text{ (and (tau' = tau) (u' = u) (v' = v) (w' = w) (s' = s))}; \\
\end{align*}
\]

\{mode 2; \\
\text{invt:} \\
(u >= 0.006); \\
(u <= 0.13); \\
(v >= 0); \\
(w >= 0); \\
(s >= 0); \\
(tau >= 0); \\
flow: \\
d/dt[tau] = 1.0; \\
d/dt[u] = (stim - jfi2) - (jso2 + jsi2); \\
d/dt[v] = ((0.94-u)/(EPI_TW1M + (EPI_TW2M - EPI_TW1M) * (1/(1+exp(-2*EPI_KWM*(u - EPI_UWM)))))); \\
d/dt[v] = (-v/EPI_TV2M); \\
d/dt[s] = (((1/(1+exp( -2 * EPI_KS * (u - EPI_US) ))) - s)/EPI_TS1); \\
\text{jump:} \\
(u >= 0.13) \Rightarrow @3 \text{ (and (tau' = tau) (u' = u) (v' = v) (w' = w) (s' = s))}; \\
\}

\{mode 3; \\
\text{invt:} \\
(u >= 0.13); \\
(u <= 0.3); \\
(v >= 0); \\
(w >= 0); \\
(s >= 0); \\
(tau >= 0); \\
flow: \\
d/dt[tau] = 1.0; \\
d/dt[u] = (stim - jfi3) - (jso3 + jsi3); \\
d/dt[v] = (-v/EPI_TV2M); \\
d/dt[v] = ((0.94-u)/(EPI_TW1M + (EPI_TW2M - EPI_TW1M) * (1/(1+exp(-2*EPI_KWM*(u - EPI_UWM)))))); \\
d/dt[v] = (-v/EPI_TV2M); \\
d/dt[s] = (((1/(1+exp( -2 * EPI_KS * (u - EPI_US) ))) - s)/EPI_TS2); \\
\text{jump:} \\
(u >= 0.3) \Rightarrow @4 \text{ (and (tau' = tau) (u' = u) (v' = v) (w' = w) (s' = s))}; \\
\}

\{mode 4; \\
\text{invt:} \\
(u >= 0.3); \\
(v >= 0); \\
(w >= 0); \\
(s >= 0); \\
(tau >= 0); \\
flow: \\
d/dt[tau] = 1.0; \\
d/dt[u] = (stim - jfi4) - (jso4 + jsi4); \\
d/dt[v] = (-v/EPI_TV2M); \\
d/dt[v] = (-v/EPI_TV2M); \\
d/dt[v] = ((0.94-u)/(EPI_TW1M + (EPI_TW2M - EPI_TW1M) * (1/(1+exp(-2*EPI_KWM*(u - EPI_UWM)))))); \\
d/dt[v] = (-v/EPI_TV2M); \\
\text{jump:} \\
(u > 2.0) \Rightarrow @4 \text{ (and (tau' = tau) (u' = u) (v' = v) (w' = w) (s' = s))}; \\
\}

\text{init:} @1 \text{ (and (tau = 0) (u = 0.0) (v = 1.0) (w = 1.0) (s = 0.0))};

\text{goal:} @4 \text{ (and (tau = 1) (u >= 0.3) (u <= 2) (v >= 0) (v <= 2) (w >= 0) (w <= 2) (s >= 0) (s <= 2))};