日本ソフトウェア科学会第 27 回大会 (2010 年度) 講演論文集

ハイブリッド制約言語HydLaの宣言的意味論

上田 和紀　細部 博史　石井 大輔

時間の経過に伴って状態が連続変化したり，状態や方程式系自体が離散変化したりする系をハイブリッドシステムと呼ぶ．我々は，不確実値の扱い，シミュレーションと検証の統合などの観点から，制約概念に基づくハイブリッドシステムモデリング言語 HydLa の設計と実装を進めてきた．HydLa は，制約階層概念の採用によって制約条件を過不足なく与えることを容易にした点や，数学や論理学の記法を最大限活用する点を特徴とするが，種々の言語機能の相互作用のためにその意味論の定式化は自明ではない．本論文では，HydLa の宣言的意味論を定式化して考察を加えるとともに，宣言的意味論から導かれる性質や帰結を具体例を用いつつ論じる．

1　はじめに

我々は，時間の経過に伴って状態が連続変化したり，状態や方程式系自体が離散変化したりする**ハイブリッドシステム**のモデリングの枠組を，制約プログラミング (Constraint Programming) の概念に基づいて構築している．物理システムやサイバーフィジカルシステム，生命システムなどの諸現象を等式や不等式を組み合わせた論理式として記述し，制約伝播に代表される探索技術を駆使して解いたり検証したりするパラダイムの確立を目標としている．

その動機は，これまでのオートマトンやペトリネットに基づく記述体系 [4] [1] に対して，数式や論理式を用いた問題の高水準記述をそのままプログラムとして用いる宣言型パラダイムの有効性をハイブリッドシステムの分野において確立することである．類似の試みとしては Hybrid CC [3] があるが，多くの記述実験の結果，連続変化フェーズと離散変化フェーズを交互に繰り返す系が満たすべき制約条件を過不足なくかつ適切に与えることは必ずしも容易でないことが判明し，ハイブリッドシステムの簡潔な記述を可能とする言語を新たに設計することとした．

我々は，2008 年にハイブリッド制約言語 HydLa の大枠 [9] を構築して以来，多くの例題記述を通じた言語仕様の詳細検討 [5]，シミュレーションアルゴリズム [8] や処理系の構築，高信頼実装の要素技術の開拓 [6] などを進めてきた．これらを通じて，HydLa の言語仕様の本質的な部分や細心の注意を要する部分などが明らかになってきた．本論文ではこれを踏まえ，HydLa の核心部分の宣言的意味論を定式化し，その記述性や性質について例題を用いつつ議論する．

2　HydLa の概要

HydLa はハイブリッドシステムのための宣言型言語であり，与えられた問題の数学的定義をできるだけそのままの形で記述し，その実行や解析を行うことを目標としている．HydLa の設計指針や関連研究については [9] を参照してほしい．

HydLa プログラムが記述の対象とする系は，一般に可算個の実数値関数 $x_1(t)$, $x_2(t)$, $\ldots (t \geq 0)$ で表現される (その特別な場合として整数値関数も同

Declarative Semantics of the Hybrid Constraint Language HydLa.

Kazunori Ueda, 早稲田大学理工学術院情報理工学科, Dept. of Computer Science and Engineering, Waseda University.

Hiroshi Hosobe, 国立情報学研究所, National Institute of Informatics.

Daisuke Ishii, ナント大学, University of Nantes.

```
  INIT ⇔ ht=10 ∧ ht'=0.
PARAMS ⇔ □(g=9.8 ∧ c=0.5).
 FALL  ⇔ □(ht''=-g).
BOUNCE ⇔ □(ht- =0 ⇒ ht'= -c*(ht'-)).
 INIT, PARAMS, (FALL << BOUNCE).
```

図 1　床で弾むボール

様に扱える). HydLa プログラムは，時間の推移とともに連続的もしくは離散的に変化するこれらの関数 (以下軌道と呼ぶ) の挙動に関する制約条件を与える．HydLa プログラム P の宣言的意味は，軌道群 $\bar{x}(t) = \{x_i(t)\}_{i\geq 1}$ が P を充足するという関係，または (同じことであるが) P を満たす $\bar{x}(t)$ 全体の集合として定義される．

ハイブリッドシステムを簡潔に表現するには，知識表現やオブジェクト指向設計などと同じように，階層化を用いたデフォルトや例外の表現が重要な役割を果たす．たとえば床に落下して弾むボールを考えると，ほとんどの時刻ではボールの速度変化は重力加速度から定まるが (デフォルト)，床との衝突のときだけは重力加速度ではなく衝突の方程式から定まる (例外)．このような系の解軌道を well-defined に定めるには，系がしたがうべき方程式の候補集合に対して半順序構造を与え，その構造の中で極大無矛盾 (maximally consistent) な集合を採用するのが数学的に簡明な方法であると考え，この方式を採用した．

図 1 に，弾むボールの HydLa による記述例を示す．最初の 4 行は制約モジュールの定義である．制約モジュールは，制約の集合を構成したり優先度をつけたりするときの単位となる．定義右辺の制約式の中の ' は時間微分，後置演算子 - は軌道の左からの極限，□ は時相演算子 always を表す．どの制約も時刻 0 における制約を表現しているが，INIT 以外は □ がついているため，時刻 0 以降のすべての時刻で成立する．BOUNCE のように含意演算子を含む制約を条件付き制約と呼ぶ．□ のついた条件付き制約は，前件 (ガード条件) が成り立つ各時点で後件の制約が課される．

最後の 1 行で 4 つの制約モジュールを組み合わせている．カンマは優先度をつけない合成で，<< は BOUNCE に FALL よりも高い優先度を与えている．この例では，ボールが空中にあるときは 4 つの制約がすべて採用され，床にぶつかった瞬間は FALL と BOUNCE とが矛盾するため {INIT, PARAMS, BOUNCE} が極大無矛盾集合として採用される．

この例は，Zeno 挙動，すなわち有限時間内に無限回の離散変化を繰り返してシミュレーションが進まなくなる例として知られ，現存の多くのハイブリッドシステム処理系では，離散変化の収束時刻 (Zeno 時刻) 以降の挙動が，ボールが床にめりこむなどの形で破綻する．

(program)	$P ::= (MS, DS)$
(module set)	$MS ::= M$ の集合を要素とする半順序集合
(definitions)	$DS ::=$ 互いに異なる左辺をもつ D の集合
(definition)	$D ::= M \Leftrightarrow C$
(constraint)	$C ::= A \mid C \wedge C$ $\mid G \Rightarrow C \mid \Box C \mid \exists x.C$
(guard)	$G ::= A \mid C \wedge C$
(atomic constraint)	$A ::= E \; relop \; E$
(expression)	$E ::=$ 通常の式 $\mid E' \mid E-$

図 2　基本 HydLa の構文

3　基本 HydLa

本論文で意味論付与の対象とする HydLa (基本 HydLa と呼ぶ) の構文を図 2 に示す．基本 HydLa は，HydLa [9] から以下のような単純化を行っている．

1. HydLa は，各時刻において，制約モジュールの無矛盾集合の中で，優先度制約を満たしかつ集合の包含関係に関して極大なものを採用する．つまり制約モジュール間の相対優先度から，制約モジュール集合のすべての部分集合の中で採用候補となるものと，それらの間の半順序関係を導出し [5]，その極大要素を採用する．基本 HydLa はこの導出は扱わずに「制約モジュール集合を要素とする狭義 (非反射的) 半順序集合」自体を，(制約モジュールの定義と組にして) プログラムとみなす．軌道の連続性のようなデフォルトの制約 (フレーム公理，第 6.1 節) も，この半順序集合の中に陽に表現する．また，制約階層の最上位の制約

は多くの場合必須制約 (必ず採用する制約) として扱うのが望ましいが，そのように扱うか否かも陽に表現できる．

2. 時間シフト (遅延) 演算子 ^ は扱わず，次項 3. の機能で代用する．
3. 軌道の動的生成を可能にするために局所変数生成機能 ∃ を導入する．これを用いてタイマーを動的に生成し，ある条件を検知してから新たな制約を発行するまでの遅延を実現できる．
4. プログラム定義は単なる展開機能なので扱わない．
5. 同種の軌道を複数個生成する ∀ 機能も同じ理由で扱わない．

プログラム (MS, DS) は，$\bigcup MS \subseteq \mathrm{dom}(DS)$ ($\bigcup MS$ は MS に出現するモジュールの集合，$\mathrm{dom}(DS)$ は DS の左辺の集合) を満たすものとする．また以下では，DS をモジュール名から制約への関数と同一視する．

図 2 のように，ガード条件は原子制約またはその連言に限定している．記述可能な制約式のクラスは定めない．この意味で HydLa は制約系がパラメタ化された**言語スキーム**である．

4 基本 HydLa の宣言的意味論

第 2 節のように，HydLa の宣言的意味は，与えられた軌道 (解釈) がプログラム (仕様) を満たすという関係を定義することによって与える．宣言的意味は，どのようなクラスのプログラムを扱うのか，合成可能性 (compositionality，全体の意味を各構成要素の意味から合成できるか) をどこまで目指すか，などの設計基準によって，保持すべき情報が変化する．文献 [9] の意味論は，□ 演算子を条件付き制約の後件に含まないプログラムを対象にしていた．有限個の構成要素をもち遅延を含まない系は，冠頭だけに □ をもつ制約で系のパラメタや挙動を記述できる．条件付き制約を含む場合も，後件が成立するのはガード条件の成立と同時刻だけであるので，制約の無矛盾極大集合の選択をその時点で行うことができる．

しかし，後件に □ を含む制約は，その前件が成立した後にも，後件が採用候補制約として残る．将来のある時点でどの後件を採用するかを前件成立の時点で判断することは将来の先読みにほかならず，極大無矛盾集合の選択は，制約が生まれた時点ではなくその制約を実際に適用する時点で行わなければならない．そこで本論文では以前の意味論を以下の手順で詳細化する．

まず，以下では制約の連言を制約の集合と同一視する．すなわち図 2 の制約の構文を

$$C ::= \{A\} \mid C \cup C \mid \{G \Rightarrow C\} \mid \{\Box C\} \mid \{\exists x.C\}$$

とみなし，さらに空集合も許す．また，条件つき制約の後件以外に出現する存在限量子 ∃ は Skolem 化によって再帰的に除去する．

さらに，制約集合は一般に時間の関数と考える．たとえば，プログラムに出現する制約 C は $C(0) = C$, $C(t) = \{\}$ $(t > 0)$ なる関数とみなす．

時間の関数である制約 $C(t)$ に対して，□-閉包 $C^*(t)$ を以下の条件を満たす関数と定義する．

- (拡張条件) $\forall t(C(t) \subseteq C^*(t))$
- (□-閉包条件)
  $\forall t(\Box a \in C^*(t) \Rightarrow \forall t' \geq t\,(a \subseteq C^*(t')))$
- (最小条件) 各 t について，$C^*(t)$ は上記 2 条件を満たす最小の集合である

たとえば $C = \{\texttt{f=0}, \Box\{\texttt{f'=1}\}\}$ とすると，$C^*(0) = \{\texttt{f=0}, \texttt{f'=1}, \Box\{\texttt{f'=1}\}\}$, $C^*(t) = \{\texttt{f'=1}\}$ $(t > 0)$ となる．

HydLa プログラムの解軌道を規定する制約集合は，二つの理由で時間進行とともに変化してゆく．一つは極大となる無矛盾集合の変化である．もう一つは条件付き制約の条件が成立して，後件が新たに制約として加わることによる変化である．前者の極大無矛盾集合の選択は，各時刻ごとに独立に行われる．これに対して後者のガード条件の成立は，後件が □ から始まる制約を持つ場合，その制約の有効性が過去におけるガード条件の成否によって決まるため，ガード条件の成立履歴を保持する必要がある．したがって，解軌道 $\bar{x} = \bar{x}(t)$ と，ガード条件の成立履歴を保持した制約モジュール定義 $Q = Q(t)$ とを対にした $\langle \bar{x}, Q \rangle$ がプログラム $P = (MS, DS)$ を満たす，という関係を考えるのが適当である．この関係を図 3 のように定める．

$\langle \overline{x}, Q\rangle \models (MS, DS) \Leftrightarrow$ (i) $\wedge$ (ii) $\wedge$ (iii) $\wedge$ (iv)，ここで

(i) $\forall M\,(Q(M) = Q(M)^*)$

(ii) $\forall M\,(DS(M)^* \subseteq Q(M))$

(iii) $\forall t\,\exists E \in MS\,($ (s0)

$\quad (\overline{x}(t) \Rightarrow \{Q(M)(t) \mid M \in E\})$ (s1)

$\quad \wedge\ \neg\exists \overline{x}'\,\exists E' \in MS\,($ (s2)

$\quad\quad \forall t' < t\,(\overline{x}'(t') = \overline{x}(t'))$ (s2)

$\quad\quad \wedge\ E \prec E'$ (s2)

$\quad\quad \wedge\ \overline{x}'(t) \Rightarrow \{Q(M)(t) \mid M \in E'\})$ (s2)

$\quad \wedge \forall d \forall e \forall M \in E\,($ (s3)

$\quad\quad (\overline{x}(t) \Rightarrow d) \wedge ((d \Rightarrow e) \in Q(M)(t))$ (s3)

$\quad\quad \Rightarrow e \subseteq Q(M)(t)))$ (s3)

(iv) 各 t における $Q(t)$ は (i)–(iii) を満たす最小の集合である

図 3 $\langle \overline{x}, Q\rangle \models P$ **の定義**

図 3 の宣言的意味論は無矛盾性に基づく制約情報の採用を基本原理としており，優先度が極大な制約モジュール集合を時々刻々満たすことを求めている．

(i) は $Q(M)$ が $\Box$-閉包条件を満たすことを，(ii) は $Q(M) = Q(M)^*$ が $DS(M)^*$ の拡張であることを求めている．(iii) を細かく見ると，行 (s0) の限量子の順序は，$\overline{x}$ が制約階層のどの候補モジュール集合を採用するかが時々刻々変化してもよいことを表す．行 (s1) は，$\overline{x}$ が時刻 t で制約階層のある候補モジュール集合を満たすことを表す．行 (s2) は，時刻 t よりも前については $\overline{x}$ と同じに振る舞い，かつ時刻 t で $\overline{x}$ よりも上位の候補モジュール集合を満たす軌道 $\overline{x}'$ がないことを表す．行 (s3) は，採用した条件付き制約の前件が成立する場合，後件が Q の中の当該モジュール M の定義の中で展開されて Q が拡張されることを表す．後件 (制約の集合とみなしている) の構成要素が $\Box$ から始まる場合はさらに $\Box$-閉包条件 (i) で展開され，また $\exists$ から始まる場合は Skolem 関数を使って限量子除去がなされる．(iv) は最小性の要請である．

5 例

簡単な記述例を用いて，宣言的意味論が実際にどのように解軌道やそれを規定する制約集合を定めるのかを説明する．

例 1: 最初の例は，単調増加関数のしきい値への到達を，遅延をはさんで別の関数に反映させるものである．

$$
\begin{aligned}
P_1 &= (MS_1, DS_1)\\
MS_1 &= (\{\{\texttt{A,B}\}, \{\texttt{A,B,C}\}\}, \{\{\texttt{A,B}\} \prec \{\texttt{A,B,C}\}\})\\
DS_1 &= \{\texttt{A} \Leftrightarrow \texttt{f=0} \wedge \Box(\texttt{f'=1}),\\
&\qquad \texttt{B} \Leftrightarrow \Box(\texttt{g=0}),\\
&\qquad \texttt{C} \Leftrightarrow \Box(\texttt{f=5} \Rightarrow \exists \texttt{a}\,.\,(\texttt{a=0} \wedge \Box(\texttt{a'=1})\\
&\qquad\qquad \wedge \Box(\texttt{a=2} \Rightarrow \texttt{g=1})))\}
\end{aligned}
$$

f は経過時刻を表す関数，a は f=5 をトリガとして生成起動するタイマー，g は通常は 0 で，タイマー起動から 2 秒後に瞬間的に 1 となるパルス関数を表す．解軌道 $\overline{x}$ は f，a (対応する Skolem 関数の名前もここでは a とする)，g の上記の挙動を表したものである．$Q(t)$ に登録される全制約 $Q(*)(t) = \bigcup\{Q(M)(t) \mid M \in \mathrm{dom}(Q)\}$ を見てゆくと，$0 < t < 5$ では $Q(*)(t)$ は f'=1, g=0 および C の頭の $\Box$ を外したものからなり，$t = 5$ ではそれに a=0, $\Box$(a'=1), a'=1, $\Box$(a=2 $\Rightarrow$ g=1) および a=2 $\Rightarrow$ g=1 が加わる．$5 < t < 7$ では a=0, $\Box$(a'=1), $\Box$(a=2 $\Rightarrow$ g=1) が除かれ，$t = 7$ では g=0 のかわりに g=1 が加わる．$t > 7$ では g=1 が除かれ再び g=0 が入るが，他の制約として f'=1, a'=1 および二つの条件付き制約がそのまま残っている．

例 2: 前節の宣言的意味論は，時間の遡及する方向に制約が伝播することを排除している．遡及の排除は意味論の構成法から明らかかもしれないが，重要な性質なので例題を使ってそのことを確認する．

$$
\begin{aligned}
P_2 &= ((\mathcal{P}(\{\texttt{D,E,F}\}), \subsetneq), DS_2)\\
DS_2 &= \{\texttt{D} \Leftrightarrow \texttt{y=0},\\
&\qquad \texttt{E} \Leftrightarrow \Box(\texttt{y'=1} \wedge \texttt{x'=0}),\\
&\qquad \texttt{F} \Leftrightarrow \Box(\texttt{y=5} \Rightarrow \texttt{x=1})\}
\end{aligned}
$$

P_2 は x の初期値が不定であるが，F が $t = 5$ で課する制約 x=1 が E 中の x'=0 によって過去に伝播するかどうかを見る．解軌道の候補として以下の三つを検討する．

1. $\mathtt{y}(t) = t$ $(t \geq 0)$, $\mathtt{x}(t) = 1$ $(t \geq 0)$ は全時刻で全制約 D, E, F を満たす．
2. $\mathtt{y}(t) = t$ $(t \geq 0)$, $\mathtt{x}(t) = 2$ $(t \geq 0)$ は $t = 5$ 以外で全制約を，$t = 5$ では D, E を満たす．
3. $\mathtt{y}(t) = t$ $(t \geq 0)$, $\mathtt{x}(t) = 2$ $(t < 5)$, $\mathtt{x}(t) = 1$

$(t \geq 5)$ は $t = 5$ 以外で全制約を，$t = 5$ では D, F を満たす．

1. は明らかに極大性を満たしているので解である．2. と 3. は $t = 5$ 以外では明らかに極大性を満たしており，これらよりも優れた解はない．$t = 5$ でもどちらかが劣ることはなく，全制約を満たす他の解もないのでどちらも極大性を満たす．つまり 1. 〜3. はどれも「極大性を満たすように解を時間軸方向に延ばして」いった結果となっており，P_2 の解である．

6 言語仕様と意味論に関する考察

6.1 微分制約の扱い

数学と論理学の既存の記法を活用するのが HydLa の基本設計指針であるが，記法の定義も数学の標準的慣習に合わせるのが望ましいと考えた．たとえば，区分的に連続な軌道が離散変化する時刻では，その軌道が右微分可能かつ左微分可能であっても微分可能とはみなさず，微係数に関する制約をその時刻でだけ無視することもしない．また，初期時刻での軌道の連続性や微分可能性は，右連続性や右微分可能性だけを要請する．

上記のことから，区分的に連続な関数の微分制約は，離散変化を記述した制約よりも一般に優先度を下げる必要がある．一方，離散変化後の軌道が常微分方程式の初期値問題として well-defined になるためには，軌道の右連続性を，離散変化時刻も含めて仮定する必要もある．このため HydLa では，微分制約をもつ軌道の右連続性を仮定する．さらに，微分制約が無効になる離散変化時点では，関数値の左連続性も仮定しないとその値を決めることができない．このため，微分制約を持つ軌道については左連続性も仮定する．この二つの制約は微分可能な時刻では自動的に成立するため，微分制約よりも高い優先度の制約としてそれぞれ独立に仮定する．

6.2 表現力

HydLa の最重要用途は区分的に連続な軌道を記述することであるが，制約と制約階層を用いて以下のような軌道や軌道群も定義できる．

6.2.1 微分方程式によらない軌道

HydLa は，微分制約を用いない軌道や軌道群の記述を許すこととした．たとえば，値が変動するパラメタを $\Box$(0.9<a ∧ a<1.1) のように表現でき，これは区間 $(0.9, 1.1)$ を値域とする全軌道の集合を表す．

ただしこのように定義した軌道は連続とは限らない．したがって f=0 ∧ $\Box$(f'=1) と定義された関数があったとき，時刻 0.9 から 1.1 の間に f'=a が成立する保証がない．これに制約 $\Box$(a'=b) を追加すると (b に関する制約は与えない)，a は $(0.9, 1.1)$ を値域とする連続かつ微分可能な全軌道の集合となり，f との交差が保証される．

微分方程式でない制約が重要な役割を果たすもう一つの用途は，パルス関数の定義である．第 5 節の例 1 の g がその例である．パルス関数はイベント発生の表現に重要な役割を果たすが，離散変化点で右連続でないため，変化後の軌道を微分方程式で定義することはできないと予想される．実際，その一つの試みが失敗することを以下に示す．

$$
\begin{aligned}
&P_3 = (MS_3, DS_3) \\
&MS_3 = (\{\{\texttt{G,J,K}\}, \{\texttt{G,H,J,K}\}\}, \{\{\texttt{G,J,K}\} \prec \{\texttt{G,H,J,K}\}\}) \\
&DS_3 = \{\, \texttt{G} \Leftrightarrow \texttt{a=0} \wedge \texttt{b=0} \wedge \Box(\texttt{a'=1}), \\
&\qquad \texttt{H} \Leftrightarrow \Box(\texttt{b'=0}), \\
&\qquad \texttt{J} \Leftrightarrow \Box(\texttt{a-=1} \Rightarrow \texttt{b=1}), \\
&\qquad \texttt{K} \Leftrightarrow \Box(\texttt{b-=1} \Rightarrow \texttt{b=0})\}
\end{aligned}
$$

MS_3 の二つの候補集合の間には，6.1 節の考察から，b の右連続性を始めとするいくつかの連続性制約を追加した集合がいくつか存在する．$t = 1$ では G,H,J,K は充足不可能だが G,J,K に b の右連続性を追加したものは充足可能で，かつ J から $\texttt{b}(1) = 1$ である．しかしそうすると，K のガード条件が成り立つ時刻の下限 (最大下界) が $t = 1$ となり，後件から $\texttt{b}(t) = 0$ $(t > 1)$ が出てくるので右連続性と矛盾する．そこで右連続性の仮定を外すと，詳細は省くが，すべての $c \neq 1$ について $\texttt{b}(t) = c$ $(t > 1)$ が無矛盾であることがわかる．つまり，解軌道は存在するものの，一意性を保証することに失敗している．

6.2.2 Zeno 挙動

第 2 節で床で弾むボールの Zeno 挙動にふれたが，第 4 節の宣言的意味論を丹念に調べると，図 1 の

プログラムは Zeno 時刻まで，つまりボールが止まるまでの解軌道を一意に定めるものの，それより後の解軌道としては床にめり込むものを許容していることがわかる．Zeno 時刻よりも後の軌道を規定するには追加のルールが必要であり [10], HydLa でも $\Box(\texttt{ht-=0} \wedge \texttt{ht'-=0} \Rightarrow \Box(\texttt{ht=0}))$ と定義できる．ただしガード条件の判定には，通常のシミュレーション手法とは異なる方法が必要となる [7].

以下は Zeno 時刻判定法の別解であり，直前のはね返り速度を保持する関数 vmax の収束を判定している．

$$\Box(\texttt{vmax'=0}) \ll \Box(\texttt{ht'-!=ht'} \Rightarrow \texttt{vmax=ht'}) \wedge \Box(\texttt{vmax-=0} \Rightarrow \Box(\texttt{ht=0}))$$

この例は，左極限演算子 - が，離散変化しか起こさない関数に対しても実用的な意味をもつことを示している．

7 まとめと今後の課題

本論文では，階層構造をもつハイブリッド制約言語 HydLa の宣言的意味論を与え，その仕組みや帰結を例題を通じて検討するとともに，言語機能および記述力に関するいくつかの考察を与えた．

本論文の意味論は通常の単純な時間概念の採用を動機として，軌道を時刻の関数とみなしている．一方，ハイブリッドシステムの理論では，一つの時刻に複数回の離散変化を許すハイブリッド時刻が採用されることもある [2]．その動機の一つは単一の離散変化時点での複数ステップの計算を適切にモデル化することであるが，HydLa は制約に基づく言語であるため，状態変化ではなく制約伝播を用いてそのような計算を表現できる．ハイブリッド時刻のもう一つの動機として，軌道の安定性や収束性の宣言的意味論を通じた扱いがあるが，こちらを実現するには今回の意味論をハイブリッド時刻に基づく意味論に拡張することが必要と考えられ，今後の検討課題である．

本宣言的意味論に対応するシミュレーションアルゴリズムの整備 [8] とその実装も整備を進めており，制約プログラミングの柔軟性および区間演算との親和性を活かしたシステムとする予定である．

謝辞 HydLa の設計は，上田研究室 HydLa 班メンバの理論・実装研究および日常の議論から多くのフィードバックを受けてきた．本研究の一部は，科学研究費補助金 (基盤研究 (B)20300013) の補助を得て行った．

参 考 文 献

[1] David, R. and Alla, H.: On Hybrid Petri Nets, *Discrete Event Dynamic Systems*, Vol. 11, No. 1–2 (2001), pp. 9–40.

[2] Goebel, R., Sanfelice, R. G., Teel, A. R.: Hybrid Dynamical Systems, *IEEE Control Systems Magazine*, Vol. 29, No. 2 (2009), pp. 28–93.

[3] Gupta V., Jagadeesan, R., Saraswat, V. and Bobrow, D.: Programming in Hybrid Constraint Languages, in *Hybrid Systems II*, LNCS 999, Springer-Verlag, 1995, pp. 226–251.

[4] Henzinger, T. A.: The Theory of Hybrid Automata, in *Proc. LICS'96*, 1996, pp. 278–292.

[5] 廣瀬賢一，大谷順司，石井大輔，細部博史，上田和紀：制約階層によるハイブリッドシステムのモデリング手法，日本ソフトウェア科学会第 26 回大会講演論文集，2D-2, 2010.

[6] Ishii, D., Ueda, K., Hosobe, H., Goldsztejn, A: Interval-based Solving of Hybrid Constraint Systems, in *Proc. ADHS'09*, pp. 144–149, 2009.

[7] 大野善之，石井大輔，上田和紀：数式処理・Quantifier Elimination を用いたハイブリッドシステムの Zeno 状態の導出手法，人工知能学会第 22 回全国大会論文集，1D1-3, 2008.

[8] 渋谷俊，高田賢士郎，細部博史，上田和紀：ハイブリッドシステムモデリング言語 HydLa 処理系における実行アルゴリズム，日本ソフトウェア科学会第 27 回大会講演論文集，2010.

[9] 上田和紀，石井大輔，細部博史：制約概念に基づくハイブリッドシステムモデリング言語 HydLa，第 5 回システム検証の科学技術シンポジウム予稿集，pp. 1–6, 2008.

[10] Zheng, H., Lee, E. A., Ames, A. D.: Beyond Zeno: Get on with It!, in *Proc. HSCC 2006*, LNCS 3927, Springer-Verlag, pp. 568–582, 2006.