数理解析研究所講究録
第591巻 1986年 53-64

イベントの2項関係に基づくハードウェア仕様記述

手嶋 茂晴, 平石 裕実, 矢島 脩三
Shigeharu TESHIMA, Hiromi HIRAISHI and Shuzo YAJIMA
(京都大学 工学部)

1. はじめに

時間やデータの抽象度の高いレベルで, ハードウェアの設計を支援する検証システム·DAシステムの開発が望まれており, そのためには仕様記述体系が不可欠である. 本論文では, 仕様はハードウェアの満たすべき要件を記述するものと考え, ハードウェアの機能的な動作についての形式的な仕様記述体系を提案する. 提案する記述体系では, ハードウェアのプリミティブな動作をイベントと考え, 時間やデータ値などをイベントの2項関係により表現する. また, その他にイベントの発生に関する因果関係を含めて動作を議論する.

時間と因果は半順序関係, 値は同値関係によって表現する. 時間を半順序関係によって表すことはハードウェアの並列動作を議論する上で有効である. イベント自体の抽象度は自由に設定できるので, 記述対象のハードウェアの記述レベルに独立な記述が可能である. システムの処理対象の相対的な関係に数学的な基礎をもつ仕様記述の意味論は抽象データ型において広く研究されている[1][2][4]. 抽象データ型ではデータの同値関係のみによってデータ型を定義する. データ型の議論はシステムの取り扱うデータに対象が限定させる. そこで本論文では抽象データ型の仕様記述体系を拡張し, ハードウェアの動的な振る舞いを取り扱うことができる記述体系を構築する.

2.では基本的な用語を定義する. **3.**ではハードウェアのモデル化およびの動作履歴の表現について考え, **4.**では仕様記述の構文と意味について議論する.

2. 諸定義

ここでは**3.**以後に用いる多ソート代数[3][5]と第一階述語論理[6][7]に関する用語を定義する.

[定義1] 集合名の集合を**So**, 関数記号の集合**Fs**とする. ただし $\mathbf{Fs}=\cup_{w\in\mathbf{So}^*,s\in\mathbf{So}}\mathbf{Fs}_{w,s}$ $w\neq w'$ または $s\neq s'$ のとき, $\mathbf{Fs}_{w,s}\cap\mathbf{Fs}_{w',s'}=\varnothing$ である. このとき, $<\mathbf{So},\mathbf{Fs}>$ をシグニチャという. $\mathbf{Fs}_{s_1\cdots s_n,s}$ は $s_1,\cdots,s_n$ を名前に持つ集合の積集合を定義域, s を名前に持つ集合を値域にもつ関数の記号より成る集合である. □

[定義2] $Sig=<\mathbf{So},\mathbf{Fs}>$ をシグニチャとするとき, Sig をシグニチャにもつ多ソート代数**A**は,

i) $s\in\mathbf{So}$ を集合名にもつ非空集合 $\mathbf{A}_s$,

ii) $f\in\mathbf{Fs}_{s_1\cdots s_n,s}$ $(n>0)$ を関数記号にもつ関数 $f_\mathbf{A}$,

$f_\mathbf{A}$: $\mathbf{A}_{s_1}\times\mathbf{A}_{s_2}\times\cdots\times\mathbf{A}_{s_n}\rightarrow\mathbf{A}_s$,

iii) $f\in\mathbf{Fs}_{\varepsilon,s}$ (εは空系列)に対応する $\mathbf{A}_s$ の要素, から構成される. 多ソート代数を以下では単に代数という. □

以下に定義する第一階述語論理は一般の述語論理に型の概念を導入したものである.

第一階述語論理は以下の記号をアルファベットにもつ言語のクラスである. 有限集合$\mathbf{S}$(型の名前の集合)に対して, 以下のように記号を定める:

i) 変数記号 変数記号の集合(可算無限)を$\mathbf{X}$とする. ただし$\mathbf{X}=\cup_{s\in\mathbf{S}}\mathbf{X}_s$, $s\neq s'$のとき$\mathbf{X}_s\cap\mathbf{X}_{s'}=\varnothing$, $\mathbf{X}_s$はsに対応する集合の要素をとる変数の集合である.

ii) 関数記号 関数記号の集合を$\mathbf{F}$とする. ただし, $\mathbf{F}=\cup_{w\in\mathbf{S}^*,s\in\mathbf{S}}\mathbf{F}_{w,s}$, $w\neq w'$または$s\neq s'$のとき$\mathbf{F}_{w,s}\cap\mathbf{F}_{w',s'}=\varnothing$.

iii) 述語記号 述語記号の集合を$\mathbf{P}$とする. ただし$\mathbf{P}=\cup_{i=1,2\cdots}\mathbf{P}_i$, $i\neq j$ のとき$\mathbf{P}_i\cap\mathbf{P}_j=\varnothing$, $\mathbf{P}_i$はi-引数述語記号の集合である.

iv) 論理記号 $\neg$, $\vee$, $\exists$

v) 補助記号 (,), ,

$\mathbf{S}$に対して, $\mathbf{X}$を変数記号の集合, $\mathbf{F}$を関数記号の集合, $\mathbf{P}$を述語記号の集合にもつ言語を$L(\mathbf{S},\mathbf{X},\mathbf{F},\mathbf{P})$と書く. $L(\mathbf{S},\mathbf{X},\mathbf{F},\mathbf{P})$には項と論理式の2種類の表現が定義される.

[定義3] $L(\mathbf{S},\mathbf{X},\mathbf{F},\mathbf{P})$において, 以下のi)~iii)を有限回繰り返して得られるものだけが項である: ここに, $\mathbf{T}_s(s\in\mathbf{S})$は以下のi)~iii)を有限回繰り返して得られる要素だけの集合であり, $\mathbf{T}=\cup_{s\in\mathbf{S}}\mathbf{T}_s$とする.

i) $a\in\mathbf{F}_{\varepsilon,s}$ は$\mathbf{T}_s$に含まれる項である.

ii) $x\in\mathbf{X}_s$ は$\mathbf{T}_s$に含まれる項である.

iii) $t_1\in\mathbf{T}_{s_1}, t_2\in\mathbf{T}_{s_2},\cdots, t_n\in\mathbf{T}_{s_n}, f\in\mathbf{F}_{s_1s_2\cdots s_n,s}$のとき,
$f(t_1,t_2,\cdots,t_n)$は$\mathbf{T}_s$に含まれる項である. □

[定義4] $L(\mathbf{S},\mathbf{X},\mathbf{F},\mathbf{P})$において, 以下のi)~iv)を有限回繰り返して得られるものだけが論理式である:

i) $p\in\mathbf{P}_n$, $t_1,t_2,\cdots,t_n\in\mathbf{T}$のとき, $p(t_1,t_2,\cdots,t_n)$は論理式である.

ii) lが論理式のとき, $(\neg l)$は論理式である.

iii) l_1,l_2が論理式のとき, $(l_1\vee l_2)$は論理式である.

iv) lが論理式, $x\in\mathbf{X}$のとき, $(\exists x l)$は論理式である.

また, $p\in\mathbf{P}_n$, $t_1,t_2,\cdots,t_n\in\mathbf{T}$のとき, $p(t_1,t_2,\cdots,t_n)$, $\neg p(t_1,t_2,\cdots,t_n)$をリテラルという. □

$(\neg(l_1\vee l_2))$を$((\neg l_1)\wedge(\neg l_2))$, $(\neg(\exists x l))$を$\forall x(\neg l)$, $(l\vee\neg(l))$を$true$と書くこともある. また$\exists$, $\neg$, $\wedge$, $\vee$の順に演算子の優先順位を定め, 不要なカッコは省くことができるものとする.

[定義5] $L(\mathbf{S},\mathbf{X},\mathbf{F},\mathbf{P})$の表現の有限集合を$Ex$に対して, Exに現れる$\mathbf{F}_{\varepsilon,s}$の要素からなる集合を$\mathbf{H}^0{}_s$とする. Exに$\mathbf{F}_{\varepsilon,s}$の要素が現れないときは適当な定数$a_s$をとり, $\mathbf{H}^0{}_s=\{a_s\}$とする. そして, $\mathbf{H}^i{}_s=\mathbf{H}^{i-1}{}_s\cup\{f(t_1,t_2,\cdots,t_n)\mid f\in\mathbf{F}_{s_1\cdots s_n,s}, t_1\in\mathbf{H}^{i-1}{}_{s_1}, t_2\in\mathbf{H}^{i-1}{}_{s_2},\cdots,t_n\in\mathbf{H}^{i-1}{}_{s_n}\}$と定義するとき, $\mathbf{H}=\cup_{s\in\mathbf{S}}\mathbf{H}^\infty{}_s$を$Ex$のHerbrand領域という. また, $\mathbf{H}^\infty{}_s$は$\mathbf{H}_s$と略記される. □

[定義6] $L(\mathbf{S},\mathbf{X},\mathbf{F},\mathbf{P})$の表現の有限集合を$Ex$, ExのHerbrand領域を$\mathbf{H}$とするとき, $HB=\{p(t_1,t_2,\cdots,t_n) \mid p\in\mathbf{P}_n, t_1,\cdots,t_n\in\mathbf{H}\ n=1,2,\cdots\}$をHerbrand基底集合という. ここで, pはExに現れる述語記号である. HBの要素を基礎例という. また, HBの部分集合をHerbrand解釈という. □

[定義7] IをHerbrand解釈とするとき, Iによる論理式の真偽を関数m_Iを用いて帰納的に定義する:

変数を含まないリテラル$p(t_1,t_2,\cdots,t_n)$に対して,

$$m_I(p(t_1,t_2,\cdots,t_n))=\begin{cases}1, & p(t_1,t_2,\cdots,t_n)\in I\text{のとき},\\ 0, & p(t_1,t_2,\cdots,t_n)\notin I\text{のとき}.\end{cases}$$

論理式lに対して,

$$m_I(\neg l)=1-m_I(l).$$

論理式l_1, l_2に対して,

$$m_I((l_1\wedge l_2))=max(m_I(l_1), m_I(l_2)).$$

論理式lと変数$x\in\mathbf{X}_s$ $(s\in\mathbf{S})$に対して,

$m_I((\exists x l))=max_\sigma(l\sigma)$, σは変数$x$をHerbrand領域の部分集合$\mathbf{H}_s$の要素に置き換える置換操作.

論理式lは$m_I(l)=1$のとき, 解釈Iにおいて真であり, $m_I(l)=0$のとき, 解釈Iにおいて偽である. □

3. ハードウェアモデル

3.1 イベントとキャリア

仕様記述体系の基となるハードウェアモデルについて述べる. ハードウェアは有限固定個のキャリアから構成され, イベントをキャリア上に発生させるものと考える. イベントはプリミティブな動作である. イベントは発生後, ある期間キャリア上に存在し, そして消滅する. イベントは時間的には期間を表す. 信号線, レジスタなどがキャリアであり, 信号線が論理値1をとっていることなどがイベントである.

キャリア上に発生するイベントは, 1つ1つ異なるイベントとして識別される. イベントにはイベントの種類を示す値が定義される. イベントの値は従来のハードウェア記述におけるデータに対応する. 設計者は設計のレベルや設計の内容に応じて値の定義される集合を設定する. イベントの値はイベントの発生するキャリアによって型が決まる.

3.2 イベントの2項関係による動作履歴の表現

ハードウェアの動作履歴は動作における時間概念を直接に反映する. 前節の議論より履歴はどのキャリアでどのイベントがいつ発生し, いつ消滅したかを示す記録, つまりイベントの値, 時間の記録である. 本論文ではどのように, どうしてイベントが発生したかという因果関係も履歴に含める. 因果と時間はともにハードウェアの動的な振る舞いに関する概念であるが, 物理的な現象を表す時間と, 物理

的なハードウェアの構成に独立な概念である因果とを区別する．因果関係を用いることによってハードウェアの実現と独立で，抽象的なハードウェアの動作を記述することが可能になる．

イベントは動作の最小単位である．したがって，イベントより細かな単位を基準にして履歴を表現することはイベントがプリミティブな動作であることに反する．イベントの相対的な関係のみで表現しなければならない．イベントの関係は履歴を表現する十分な能力を持つ．

以下 (1)因果，(2)時間，(3)値の順に履歴の表現について述べる．

(1) 因果：因果を表わす2項関係は反射律の成立を仮定すると，イベントの半順序関係である．イベントaがイベントbの発生の原因，イベントbがイベントaの結果であるとき，かつそのときにのみ$a \geq_c b$とする．図1に示すDフリップ・フロップの場合，イベントe_1, e_2はQ上に発生するイベントe_3の原因であるので，$e_1 \geq_c e_3$ かつ $e_2 \geq_c e_3$が成立する．

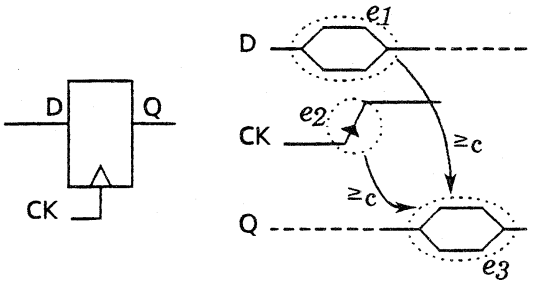

図 1 D-フリップ・フロップの動作

(2) 時間：『イベントaの消滅後にイベントbが発生する』という2項関係のみで時間を考える．時間を表す2項関係は，時間が過去から未来へ流れることに対応して，因果関係と同様に半順序関係である．イベントaの消滅後にイベントbが発生するとき，かつそのときにのみ$a \geq_t b$とする．図2に示すようにイベントが発生するとき，図3に示す関係が成立する．また，$a \not\geq_t b$かつ$a \not\leq_t b$なるイベントa, bは並列に処理される可能性をもつ．しかし，一般に$a \not\geq_t b$かつ$a \not\leq_t b$なる2つのイベントの組がすべて並列に処理できるとは限らない．

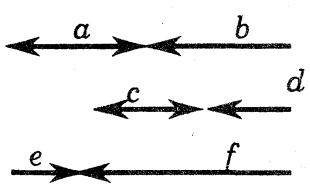

図2 図式的な時間表現

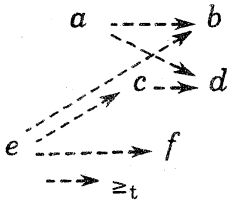

図3 2項関係による時間表現

(3) 値：値には型が定義されるとした．型の概念は多ソート代数によって形式化される．本モデルでは，イベントの相対的な関係を議論の対象としているので，代数の具体的な構成を知る必要はない．したがって，値は抽象データ型と考える．そこで

代数を構成する集合において要素d_1, d_2が同一の要素であるとき，かつそのときにのみ$d_1 =_v d_2$が成立する同値関係を用いて値を表現する．イベントからイベントの値を求める関数をvalとし，valを用いて$=_v$をイベントと値の集合との和集合における同値関係に拡張する．イベントaの値がdであるとき，つまり$val(a)=d$のとき，かつそのときにのみ$a=_v d$，またイベントaとイベントbが同じ値をもつとき，つまり$val(a)=val(b)$のとき，かつそのときにのみ$a=_v b$とする．

図1において，フリップ・フロップを機能させるキャリアCK上のイベント(信号値の立ち上がり)の値を$p\text{-}edge$とするとき，$e_1=_v e_3$かつ$e_2=_v p\text{-}edge$が成立する．

(1)~(3)の議論により，履歴の形式的な表現を定義する．

[定義8]　Cをキャリアの名前の集合，また$Sig=<\mathbf{So},\mathbf{Fs}>$をシグニチャとして，関数$v:\mathbf{C}\rightarrow\mathbf{So}$が定義されるとき，$<\mathbf{C},\mathbf{So},\mathbf{Fs},v>$をスキーマという．□

Cはハードウェアを構成するキャリア名の宣言である．Sigはイベントの値がつくる代数のシグニチャである．また，vはキャリア上に発生するイベントの値の型を決める関数である．スキーマはハードウェアの静的構成を明らかにする．

[定義9]　2項関係R_1, R_2をもつイベントの集合E，$Sc=<\mathbf{C},\mathbf{So},\mathbf{Fs},v>$なるスキーマに対して，$Sig=<\mathbf{So},\mathbf{Fs}>$をシグニチャにもつ代数を$\mathbf{A}$とし，関数$car, val$を$car: E\rightarrow\mathbf{C}$，$val: E\rightarrow\cup_{s\in\mathbf{So}}\mathbf{A}_s$とするとき，$<\mathbf{Sc},E,R_1,R_2,car,val,\mathbf{A}>$なる数学的構造が以下の条件を満たすならば，$<\mathbf{Sc},E,R_1,R_2,car,val,\mathbf{A}>$をスキーマ$Sc$に対する履歴という：

i) $e\in E$　ならば　$val(e)\in\mathbf{A}_{v(car(e))}$．

ii) R_1, R_2は半順序関係で，EはR_1，R_2に関して共通の最大元をもつ．

iii) $e_1,e_2\in E$，$e_1\neq e_2$，$e_1R_1e_2$ならば$e_2\not R_2 e_1$．□

履歴$<\mathbf{Sc},E,R_1,R_2,car,val,A>$において，$R_1$は因果関係$\geq_c$，$R_2$は時間関係$\geq_t$に対応し，$R_1$と$R_2$の最大元は処理の開始を表わす仮想的イベントである．以下では最大となるイベントをe_{beg}と書くことにする．

4. 仕様記述　-構文と意味-

4.1 第一階述語論理による関係の記述

仕様はハードウェアの満たすべき要件である．本論文では機能的動作を対象にし，その構文と意味について考察する．モデルにおいて動作に関する仕様は履歴において成立すべき因果，時間，値についての性質である．つまり履歴において，

1) 値のつくる代数
2) イベント発生
3) 時間的制約　　について成立すべき性質が記述される．

因果，時間，値の2項関係に2引数述語を対応させ1)~3)の性質について第一階述語論理を用いて記述する．

$Sc=<\mathbf{C},\mathbf{So},\mathbf{Fs},v>$をもつハードウェアの履歴$<\mathbf{Sc},E,R_1,R_2,car,val,\mathbf{A}>$に対して，$L_{Spec}=L(\mathbf{C}\cup\mathbf{So},\mathbf{X},\mathbf{F},\mathbf{P})$の論理式を用いる．ここで$\mathbf{Fs}_{w,s}\subseteq\mathbf{F}_{w,s}$ $(w\in\mathbf{So}^*, s\in\mathbf{So})$，$\{val_c\}\subseteq\mathbf{F}_{c,v(c)}$ $(c\in\mathbf{C})$，$\{beg\}\subseteq\mathbf{F}_{\varepsilon,car(e_{beg})}$，$\{Eqv, Cas, Bfr\}\subseteq\mathbf{P}_2$である．$Eqv$は，代

数と付値関数valによって定義されるイベントの同値関係$=_v$に対応し，Casは因果関係$\geq_c$($x \geq_c y : Cas(x,y)$, x causes y), Bfrは時間関係$\geq_t$($x \geq_t y : Bfr(x,y)$, x before y)にそれぞれ対応する述語記号でる．val_cはvalの定義域を$car(e)=c$ ($c \in \mathbf{C}$)となるイベントに制限した関数の関数記号であり，$beg \in \mathbf{F}_{\varepsilon,car(e_{beg})}$は開始のイベント$e_{beg}$に対応する定数である．しかし，一般の$L_{Spec}$の論理式は履歴を解釈にして，真偽が判定できない．履歴は動作過程において発生したイベントについて値，因果，時間の関係を表現するものであり，発生していないイベントについては記述されていない．そこで履歴に対してL_{Spec}の論理式が意味をもつように構文に制限を加える．

[定義8]　L_{Spec}の論理式l, lに出現する拘束変数xが項t以前(以降)に制限されているということを次のように定義する:

i) xがlの冠頭標準形において$\forall$で拘束されているとき，以下の条件を満たす冠頭和積標準形l'が存在する:

xの出現するすべての和項において，$\neg Cas(t_i,t_{i+1})$ ($\neg Cas(t_{i+1},t_i)$) $i=1,\cdots,n-1$, $t_1=x$, $t_n=t$ なるリテラルがすべて，または$\neg Cas(x,y)$ ($\neg Cas(y,x)$)なるリテラルがその和項に出現する．ここで，$t_1, t_2, \cdots, t_n$は項であり，yは項t以前(以降)に制限されている変数である．これらは和項ごとに定まる．

ii) xがlの冠頭標準形において$\exists$で拘束されるとき，以下の条件を満たす冠頭積和標準形l''が存在する:

xが出現するすべての積項において，$Cas(t_i,t_{i+1})$ ($Cas(t_{i+1},t_i)$) $i=1,\cdots,n-1$, $t_1=x$, $t_n=t$ なるリテラルがすべて，または$Cas(x,y)$ ($Cas(y,x)$)なるリテラルがその積項に出現する．ここで，$t_1, t_2, \cdots, t_n, y$はi)と同様である．

また，lの自由変数については，i)の条件($\forall$で拘束されることは除く)を満たすとき$\forall$の意味で項t以前(以降)に制限されるといい，またii)の条件(同様)を満たすとき$\exists$の意味で項t以前(以降)に制限されるという．　□

変数xが項t以前に制限されるとき，項tの原因でないイベントへの変数xの置換は論理式の真偽に影響しない．

スキーマを$Sc=<\mathbf{C}, \mathbf{So}, \mathbf{Fs}, v>$とするとき，$Sc$をスキーマにもつハードウェアの動作に対する仕様は，L_{Spec}の論理式と$\Rightarrow_c, \Rightarrow_t$の記号からなるsyn1)~syn3)の構文をもつ表現の集合である:

syn1)　$Eqv(t,t')$, $t,t' \in \mathbf{T}_s$ ($s \in \mathbf{So}$)

syn2)　$l_1 \Rightarrow_c l_2$　　l_1, l_2はL_{Spec}の論理式．

l_1は次の条件を満たす:

i) 自由変数は$\exists$の意味でbeg以降に制限される．

ii) 拘束変数はいずれかの自由変数以後に制限され，かつ，いずれかの自由変数以前に制限される．

iii) 冠頭積和(和積)標準形で$\neg Eqv(t,t')$ $t,t' \in \mathbf{T}$なるリテラルを含まない．

iv) $t \in \mathbf{T}_s$ ($s \in \mathbf{So}$)はCas,Bfrの引数ではない．

l_2は$Eqv(x,t)$なる論理式である．ここでxは$x \in \mathbf{X}_c$, $c \in \mathbf{C}$なるl_1の自由変数で，tはl_1の自由変数のみを変数に含む項である．

syn3) $l_3 \Rightarrow_t l_4$　l_3, l_4はL_{Spec}の論理式.
l_3は次の条件を満たす:

i) 変数はすべて自由変数で∃の意味で*beg*以降に制限されている.

ii) 正リテラルの積である.

l_4は$Bfr(x,x')$なる論理式である. x, x'はl_3に出現する自由変数であり, l_3の自由変数はxまたはx'の以前に制限される. □

syn1)の表現$Eqv(t,t')$は直観的には, 項t,t'を評価するとき, 値が等しいことを意味する. syn2)の表現$l_1 \Rightarrow_c l_2$はl_1の自由変数を履歴のイベントに置き換えたとき, l_1が真ならば, 値についてl_2を満たすイベントが発生することを意味する. つまりl_1の自由変数に割りあてられたイベントに対して, l_2を満たすイベントが存在し, その間に因果関係が成り立つ. syn3)の表現$l_3 \Rightarrow_t l_4$はl_3の自由変数を履歴のイベントに置き換えたとき, l_3が真ならばl_4で表わされる時間的制約が成り立つことを意味する.

4.2 仕様記述の不動点意味

仕様記述の意味を入力および初期値から動作した履歴を求める関数として定義する. 本論文では入力・初期値は履歴の中であらかじめわかっている有限個のイベントの関係と考える. 入力と初期値をあわせて入力パターンということにする. 仕様を公理と考え, そこからイベント間の関係を導出することによって, ハードウェアの動作をシミュレートする. 導出可能なすべての関係が入力パターンに対する履歴である. 我々は入力パターンをHerbrand解釈に対応づけ, Herbrand領域で意味を議論する. まずそのためのHerbrand領域を設定する.

[定義10] スキーマ$Sc=<\mathbf{C},\mathbf{So},\mathbf{Fs},v>$の仕様$Spec$に対して, 関数$cond : Spec \rightarrow L_{Spec}$の論理式の集合, と関数$lit : Spec \rightarrow$正リテラルの集合 を次のように定義する:
$exp \in Spec$に対して,

i) expが $Eqv(t,t')$なる表現(syn1))のとき,

$cond(exp)=true$,

$lit(exp)=\{Eqv(t, t')\}$.

ii) expが $l \Rightarrow_c Eqv(x,t)$なる表現(syn2))のとき,

$cond(exp)=l$,

$lit(exp)=$ $\{Cas(x_1, f_{exp}(x_1,\cdots,x_n)), Cas(x_2, f_{exp}(x_1,\cdots,x_n)), \cdots, Cas(x_n, f_{exp}(x_1,\cdots,x_n)),$
$Eqv(f_{exp}(x_1,\cdots x_n),t), Bfr(beg, f_{exp}(x_1,\cdots,x_n))\}$

ここで, $x_1,\cdots,x_n$はlの自由変数, f_{exp}はexpに対してあらかじめ決められている関数記号で, $\mathbf{Fs}_{c_1c_2\cdots c_n,c}$ ($x \in \mathbf{X}_c, x_1 \in \mathbf{X}_{c_1}, x_2 \in \mathbf{X}_{c_2}, \cdots, x_n \in \mathbf{X}_{c_n}$, $c, c_1, c_2, \cdots, c_n \in \mathbf{C}$)の要素である. また$exp_1 \neq exp_2$ならば$fexp_1 \neq fexp_2$とする.

iii) expが $l \Rightarrow_t Bfr(x,x')$なる表現(syn3))のとき,

$cond(exp)=l$,

$lit(exp)=\{Bfr(x,x')\}$. □

キャリアごとに$f_c \in \mathbf{F}_{c,c}, a_c \in \mathbf{F}_{\varepsilon,c}$ ($c \in \mathbf{C}$)なる関数記号を選ぶとき, 表現の集合 $\cup_{exp \in Spec}\{cond(exp)\} \cup lit(exp) \cup_{c \in \mathbf{C}}\{f_c(a_c)\}$のHerbrand領域H, Herbrand基底集合

HBは仕様$Spec$の意味を議論するための十分な能力をもつ．以後これらを単に$\mathbf{H}$, HBと記す．

仕様の表現は動作の立場から見ると，syn1)~syn3)のいずれの表現も，「途中までの履歴が条件を満たせば，新しくイベントに関係が生じる」ということを記述している．このことをHerbrand解釈上で考えると，表現はHerbrand解釈で条件が満たされるとき，新しく生じる関係に対応する基礎例を解釈に追加する手続きを意味する．関数$cond$は表現の意味する手続きが適用される条件，関数litは条件が成立するとき，Herbrand解釈に追加される基礎例をリテラルで表す．syn2)の場合，新しく発生するイベントを$f_{exp}(x_1,\cdots,x_n)$によって表している．f_{exp}はイベントを返す関数の関数記号であるが，ここでは関数の内容は考えず，イベントを識別するために名前をつける働きをする．しかし，定義9よりすべてのHerbrand解釈が履歴を表現するわけではない．Cas, Bfr, Eqvによって表される述語 をHerbrand領域上の2項関係と見なしたとき，Cas, Bfrが半順序関係に，Eqvが同値関係にならなければならない．また，Eqvによる同値類分割が代数を構成しなければならない．これらの条件を満たすHerbrand解釈の集合を$\mathrm{U_{his}}$とする．

任意の解釈$I\epsilon\mathrm{U_{his}}$は以下のように履歴$<Sc, E, \geq_c, \geq_t, car, val, \mathbf{A}>$に対応づけることができる：

1) $Cas(beg,t)\epsilon I$となる項tのみを含む集合E，またそのとき，$t\epsilon\mathbf{H}_c$ ($c\epsilon\mathbf{C}$)ならば，$car(t)=c$と定義される関数car.

2) Eの要素t, t'に対して，Cas, Bfrに対応する$\geq_c, \geq_t$.

3) Eqvに対応する同値関係$=_v$による$\mathbf{H}_s$ ($s\epsilon\mathbf{So}$)の同値類分割$\mathbf{H}_s/=_v$と，$f\epsilon\mathbf{Fs}$に対応して同値類分割上に定義される定数，関数から構成される代数$\mathbf{A}$.

4) Eの要素tに対して，$Eqv(t,t')\epsilon I$かつ$t'\epsilon\mathbf{H}_s$ ($s\epsilon\mathbf{So}$)のとき，t'が含まれる$\mathbf{H}_s/=_v$の同値類dとすると，$val(t)=d$により定義される関数val.

$\mathrm{U_{his}}$の要素は履歴$<Sc, E, \geq_c, \geq_t, car, val, \mathbf{A}>$に現れるイベントおよび代数の要素にHerbrand領域の要素によって名前をつけたものである．

一方，入力パターン$<Sc, E_{IP}, \geq_c, \geq_t, car, val, \mathbf{A}_{IP}>$は次のような$\mathrm{U_{his}}$の要素$I$に対応づけることができる．入力パターンにおいてキャリア$c\epsilon\mathbf{C}$上に発生するイベントに対して項$a_c, f_c(a_c), f_c(f_c(a_c)), \cdots\epsilon\mathbf{H}_c$を割り当てる．$I$は以下のようなHerbrand解釈である：

任意のイベント$e, e'\epsilon E_{IP}$に対して，

1) $e\geq_c e'$,のとき，かつそのときにのみ $Cas(t_e, t_{e'})\epsilon I$.

$e\geq_t e'$,のとき，かつそのときにのみ $Bfr(t_e, t_{e'})\epsilon I$.

ただし，$t_e, t_{e'}$はそれぞれイベントe, e'に割り当てられた項である．

2) $e=_v e'$, $car(e)=c$かつ$car(e')=c'$のとき，かつそのときにのみ

$Eqv(t_e, t_{e'})$, $Eqv(val_c(t_e), t_{e'})$, $Eqv(t_e, val_{c'}(t_{e'}))$, $Eqv(val_c(t_e), val_{c'}(t_{e'}))\ \epsilon I$.

ただし，$=_v$は代数$\mathbf{A}_{IP}$と関数valから得られる同値関係である．

3) $val(e)=d$かつ$car(e)=c$のとき，かつそのときにのみ

$Eqv(t_e, t_d), Eqv(val_c(t_e), t_d) \in \Gamma$. ただし，$d$は代数$\mathbf{A_{IP}}$の要素であり，$t_d$は代数$\mathbf{A_{IP}}$において$d$と評価される項である．

4) 項$t, t' \in \mathbf{H}_s$ $(s \in \mathbf{So})$が代数$\mathbf{A_{IP}}$において同一の要素に評価されるとき，かつそのときにのみ$Eqv(t, t') \in \Gamma$.

[定理1] Tを任意の解釈$I \in \mathbf{U_{his}}$に対して$\mathbf{T} \supseteq I$となるHerbrand解釈とし，$\mathbf{U} = \mathbf{U_{his}} \cup \{\mathbf{T}\}$とするとき，Uは$\supseteq$に関して完備束である．

証明は自明である．□

次に関数$lit, cond$を用いて仕様と入力パターンからイベントの関係を導き出す関数を定義する．

[定義11] スキーマ$Sc = <\mathbf{C}, \mathbf{So}, \mathbf{Fs}, v>$の仕様$Spec$に含まれるsyn1) syn2) の形の表現すべてから成る部分集合を$Spec^{V,C}$, syn3)の形の表現すべてから成る部分集合を$Spec^{T}$とするとき，次のように関数$\tau_1, \tau_2, \tau_3 : 2^{HB} \to 2^{HB}$を定義する：

$I \in 2^{HB}$に対して，

$\tau_1(I) = I \cup \{L\sigma \mid L \in lit(exp)\}$ $exp \in Spec^{V,C}, m_I(cond(exp)\sigma) = 1$のとき，

I 任意の$exp \in Spec^{V,C}$, 任意の置換σに対して，$m_I(cond(exp)\sigma) = 0$のとき．

$\tau_2(I) = I \cup \{L\sigma \mid L \in lit(exp)\}$ $exp \in Spec^{T}, m_I(cond(exp)\sigma) = 1$のとき，

I 任意の$exp \in Spec^{T}$, 任意の置換σに対して，$m_I(cond(exp)\sigma) = 0$のとき．

$$\tau_3(I) = I \cup \{Cas(x, x)\sigma \mid m_I(Cas(beg, x)\sigma) = 1\}$$
$$\cup \{Cas(x, z)\sigma \mid m_I(Cas(x, y) \wedge Cas(y, z) \wedge Cas(beg, x)\sigma) = 1\}$$
$$\cup \{Bfr(x, x)\sigma \mid m_I(Cas(beg, x)\sigma) = 1\}$$
$$\cup \{Bfr(x, z)\sigma \mid m_I(Bfr(x, y) \wedge Bfr(y, z) \wedge Cas(beg, x) \wedge Cas(beg, y) \wedge Cas(beg, z)\sigma) = 1\}$$
$$\cup \{Bfr(beg, x)\sigma \mid m_I(Cas(beg, x)\sigma) = 1\} \cup \{Eqv(w, w)\sigma\}$$
$$\cup \{Eqv(w, v)\sigma \mid m_I(Eqv(v, w)\sigma) = 1\} \cup \{Eqv(w, u)\sigma \mid m_I(Eqv(w, v) \wedge Eqv(v, u)\sigma) = 1\}$$
$$\cup \{Eqv(t'' \; t_1/x_1 \; t_2/x_2 \cdots t_n/x_n, \; t'' \; t'_1/x_1 \; t'_2/x_2 \cdots t'_n/x_n) \mid t'' \in \mathbf{T},$$
$$Eqv(t_1, t'_1), Eqv(t_2, t'_2), \cdots, Eqv(t_n, t'_n) \in I, \; x_1, x_2, \cdots, x_n \text{は} t'' \text{に現れる変数}\}$$
$$\cup \{Eqv(x, w)\sigma \mid m_I(Eqv(val_c(x), w)\sigma) = 1\} \cup \{Eqv(val_c(x), w)\sigma \mid m_I(Eqv(x, w)\sigma) = 1\}$$

ここで，$x, y, z \in \cup_{c \in \mathbf{C}} \mathbf{X}_c$ $w, v, u \in \mathbf{X}$, ただし，σは論理式の自由変数$x \in \mathbf{X}_s$ $(s \in \mathbf{C} \cup \mathbf{So})$のHerbrand領域の部分集合$\mathbf{H}_s$の要素への置換である． □

τ_1はsyn1), syn2) の形の表現に対応する関数であり，τ_2はsyn3) に対応する関数である．τ_3は解釈に履歴の条件を満足させるようとする関数である．ただし，半順序関係の反対称律の成立は保証されない．τ_1, τ_2, τ_3はσの選び方について定義されていないが，σの選び方は以後の議論において本質でない．τ_2, τ_3についてはチャーチ・ロッサ性が成り立つ．τ_2, τ_3はリテラル追加の条件が正リテラルのみによって表さ

れていることにより容易に理解される.また,τ_1についてもチャーチ・ロッサ性が保証される範囲で用いられる.

τ_2, τ_3に関して次の定理が成り立つ.

[定理2] 任意の解釈$I\epsilon 2^{HB}$に対して,$X=\{I, \tau_3\circ\tau_2(I), (\tau_3\circ\tau_2)^2(I), \cdots\}$とするとき,$sup X$($X$の上限)は一意に決まり,かつ$\tau_3\circ\tau_2$の不動点である.

(証明)$\tau_3\circ\tau_2$は基礎例を追加する関数であるから,$I\subseteq I'$ならば,$\tau_3\circ\tau_2(I)\subseteq\tau_3\circ\tau_2(I')$したがって,$I\subseteq\tau_3\circ\tau_2(I)\subseteq(\tau_3\circ\tau_2)^2(I)\subseteq\cdots\subseteq sup X$であり,また$sup X\subseteq\tau_3\circ\tau_2(sup X)$も成立する.いま,$\tau_3\circ\tau_2(sup X)\supset sup X$と仮定し,$g\epsilon\tau_3\circ\tau_2(sup X)$, $g\notin sup X$とする.$\tau_3\circ\tau_2$の定義より,gが追加される条件は,有限個の基礎例が解釈に含まれているか否かで真偽が決められる.$sup X=sup\{x\mid x\epsilon X\}$より,真偽を決定する基礎例を含む解釈を$I_1, \cdots, I_n\epsilon X$すると,$I_i\subseteq I'$ $(i=1,2,\cdots,n)$なる$I'\epsilon X$が存在し,I'においてもgが追加される条件は真である.つまり$g\epsilon\tau_3\circ\tau_2(I')\subseteq sup X$ であり,矛盾する.また,$sup X$の一意性はτ_2, τ_3のチャーチ・ロッサ性より自明である. □

定理2より,$fix[\tau_3\circ\tau_2](I)=sup\{I, \tau_3\circ\tau_2(I), (\tau_3\circ\tau_2)^2(I), \cdots\}$なる関数$fix[\tau_3\circ\tau_2]$を定義することができる.

[定義12] $\tau=fix[\tau_3\circ\tau_2]\circ\tau_1$とするとき,関数$\tau_U$:U→Uを解釈$I\epsilon$Uに対して次のように定義する:

$\tau_U(I)=$ T $I=$Tのとき,または$\tau(I)$が履歴の条件を満たさないとき,

$\tau(I)$ その他のとき. □

τ_Uはイベントを発生させ,そのイベントについての関係を導き出す関数である.τ_Uを無限回適用することによってハードウェアの動作をシミュレートすることを考える.τ_Uの性質を議論するまえにτ_1において成り立ついくつかの性質を明らかにする.

[補題1] 解釈$I\epsilon 2^{HB}$において,Herbrand領域Hの部分集合H'をH'$=\cup_{s\epsilon C\cup So}\{t\mid t\epsilon H_s, Cas(beg, t)\epsilon I\}$とする.解釈$I$が以下の条件(L1)を満たすとき,H'に含まれる任意の項t_1, t_2対して,$Cas(t_1, t_2)\notin I$ならば$Cas(t_1, t_2)\notin\tau_1(I)$である.

ここで,条件(L1)とはexpに現れる自由変数のH'の要素への任意の置換σと任意の$exp\epsilon Spec^{V,C}$に対して,Casの引数として$lit(exp)\sigma$に現れる項がすべてH'に含まれるならば$lit(exp)\sigma\subseteq I$が成り立つことである.

(証明)解釈Iが条件(L1)を満たし,$Cas(t_1, t_2)\notin I$ $(t_1, t_2\epsilon$H'$)$かつτ_1により$Cas(t_1, t_2)$なる基礎例がIに追加されたと仮定する.つまり,$m_I(cond(exp)\sigma)=1$, $Cas(t_1, t_2)\epsilon lit(exp)\sigma$となる置換$\sigma$が存在する.$exp\epsilon Spec$はsyn2)の構文をとる表現である.

τ_1の定義より,t_2を除きCasの引数として$lit(exp)\sigma$に現れる項はすべてH'に含まれる.また仮定より$t_2\epsilon$H'. したがって,条件(L1)より$lit(exp)\sigma\subseteq I$. つまり$Cas(t_1, t_2)\epsilon I$となり仮定に反する. □

[補題2] 補題1と同じH', 条件(L1)を満たす解釈$I\epsilon 2^{HB}$に対して,$Bfr(t_1,t_2)\notin I$ $(t_1, t_2\epsilon$H'$)$かつ$\tau_3\circ\tau_2(I)=I$ならば $Bfr(t_1, t_2)\notin fix[\tau_3\circ\tau_2]\circ\tau_1(I)$である.

(証明) $Bfr(t_1,t_2)\in fix[\tau_3\circ\tau_2]\circ\tau_1(I)$ となるためには, τ_3 と τ_2 の定義より $Bfr(t_1, t_2)$ は $Cas(t_1',t_1)$, $Cas(t_2',t_2)$ $(t_1',t_2'\in H')$ なる基礎例が $\tau_1(I)$ に含まれなければならないが, 補題1より τ_1 はそのような基礎例を I に追加しない. □

[定理3] (L1)を満たす解釈 $I\in U$ に対して, τ_U はチャーチ・ロッサ性をもつ.

(証明) 表現 $exp\in Spec$ が自由変数に対する置換 σ によって, $m_I(cond(exp)\sigma)=1$ であるとする. また σ により $cond(exp)$ の自由変数が置換されたHerbrand領域の項を t_1, t_2, ..., t_n とする. 補題1, 2より Cas, Bfr を述語記号として, 項 $t_1, t_2, ..., t_n$ を引数にもつ基礎例は, τ_U によって I に追加されることはない. 一方 τ_U において基礎例追加の条件では, Eqv については負リテラルの出現を許さない. したがって, $m_{\tau_U(I)}(cond(exp)\sigma)=1$ であり, チャーチ・ロッサ性が満たされる. □

[定理4] 条件(L1)式を満たす解釈 $I\in U$ に対して $X=\{I, \tau_U(I), \tau_U^2(I), \cdots\}$ とすると, $supX$ は一意に決まり, かつ τ_U の不動点である.

(証明) τU は基礎例を追加する関数であるから, $supX\subseteq\tau U(supX)$. また, 定理2より, $supX\neq T$ のとき, $\tau_3\circ\tau_2(supX)=supX$ が示される.

$supX\supseteq\tau U(supX)$ を証明する. $supX=T$ のとき, 明らかに成り立つ. $supX\neq T$ のとき, $g\in\tau_U(supX)$ $g\notin supX$ なる基礎例 g が存在すると仮定する. $\tau_1(supX)=supX$ とすると, $\tau_3\circ\tau_2(supX)=supX$ となり仮定に反するに反する. したがって, τ_1 によって基礎例が $supX$ に追加されたことになる. つまり $supX$ において, $cond(exp)\sigma$ が真となり, $g\in lit(exp)\sigma$ である置換 σ が存在する. exp の自由変数が $y_i(i=1,2,\cdots,n)$ であるとき, $I'=\{Cas(beg, y_i)\sigma, Eqv(y_i, t'_i)\sigma \mid i=1, 2, \cdots, n.$ $Eqv(y_i,t'_i)$ は $cond(exp)$ に現れるリテラルのうち述語記号 Eqv をもつもの$\}$ $(I'\subseteq supX)$ は要素数有限だから, $I'\subseteq I''$, $I''\in X$ となる I'' が存在する. 補題1,2より $supX$ において $cond(exp)\sigma$ が真ならば, I'' においても, $cond(exp)\sigma$ が真となり $g\notin supX$ に反する.

また解釈 I が(L1)を満たすとき, $\tau_U(I)$ も(L1)を満たす. したがって, 任意の X の要素も同様に(L1)を満たす. 定理3より $supX$ の一意性が示される. □

[定義13] 仕様 $Spec$ の意味を $fix[\tau_U]\circ fix[\tau_3\circ\tau_2]$ と定義する. □

入力パターンから変換されるHerbrand解釈 I_P に対しては $fix[\tau_3\circ\tau_2](I_P)$ が条件(L1)を満たすので, $fix[\tau U]\circ fix[\tau_3\circ\tau_2](I_P)$ の存在は定理4により保証される. τ_U は τ_1 によってイベントを発生させ, τ_2 と τ_3 によって制約条件, 履歴としての条件から生じる関係をすべて求める. τ_U をくり返すことは, ハードウェアが動作することに対応し, τ_U の不動点はハードウェアが無限に動くことに対応する.

5. むすび

本論文ではハードウェアの機能的な動作を対象とした仕様記述体系を提案した. 提案する仕様記述では, ハードウェアの並列動作を考慮せず, 直列的に行わなければならない処理のみを記述する. 直列処理の制約が満たされる範囲で最大の並列度をもつハードウェアの動作が記述から得られる履歴である. また, 動作履歴は時系列によって表現されることが多い. 時系列において, 時間は系列を構成する要素間の

全順序関係によって表現される．本論文のモデルでは時間を半順序関係に拡張し，より柔軟な記述を可能にしている．

Top-Down設計において，仕様から得られる実現はそれ自身以後の設計段階での仕様である．したがって，実現もハードウェアモデルに従いモデル化され，仕様とそれから得られた実現は本論文の提案する同一の記述体系で記述可能である．実現の仕様に対する正当性は仕様で成立するイベント間の関係が実現においても成立することである．

意味となる関数の定義は，記述の記号シミュレーションの手続きとみることができる．記号シミュレーションでは，データに記号値を許すが，本論文での入力パターンの考え方に従えば，記号値を含む入力パターンは，値についての関係の一部しか含まない入力パターンであり，手続きはそのまま対象を記号値に拡張することができる．

謝辞 日頃から御議論いただく京都大学矢島研究室の諸氏に感謝いたします．

参考文献

(1) J. V. Guttag: "Abstruct Data Types and the Develpoment of Data Structure", CACM **20**, 6, pp. 396-404 (June 1977).

(2) J. A. Goguen, J. W. Thatcher, E. G. Wagner: "Initial Algebra Approach to the Specification, Correcrness and Implementation of Abstruct Data Types", IBM Reserch Report RC 6487 (Oct. 1976).

(3) 稲垣,坂部: "多ソート代数と等式論理", 情報処理, **25**, 1, pp.47-53 (1984-01).

(4) 稲垣,坂部: "抽象データタイプ", 情報処理, **25**, 5, pp.491-501 (1984-05)

(5) H. Ehrig, B. Mahr : "Fundamentals of Algebraic Specification 1", Springer-Verlag (1985).

(6) 長尾, 淵: "論理と意味", 岩波書店 (1983).

(7) C. Chang, R. C. Lee: "Symbolic Logic and Mechanical Theorem Proving", Academic Press (1973).

(8) D. Bjøner, C. Jones: "Formal Specification & Software Development", Prentice-Hall (1982).

(9) C. A. R. Hoare: "Communicating Sequentail Processes", CACM, **21**, 8 (Aug. 1978).

(10)C. Hewitt: "Laws for Communicating Parallel Processes", Proc. IFIP Congress, Toronto (1977).

(11)D. Scott: "Out Line of a Mathematical Theory of Comptation", Tch. Mono. PRG-2, Oxford Univ, Comp. Lab (1970).

(12)中島玲二: "数理情報学入門", 朝倉書店 (1982).

(13)M. H. Emden, R. A. Kowalski: "The Semantics of Predicate Logic as a Programming Language", J. ACM, **23**, 4, pp. 733-742 (Oct. 1976).

(14)J. Staples, V. L. Nguyen: "A Fixpoint Semantics for Nondeterministic Data Flow", J. ACM, **32**, 2, pp. 411-444 (April 1985).