Verification via Digitized Models of Real-Time Hybrid Systems

Dang Van Hung *and Ko Kwang Il †
The United Nations University
International Institute for Software Technology
P.O.Box 3058, Macau
{dvh,kki}@iist.unu.edu

Abstract

The paper presents an approach to the specification and verification of real-time hybrid systems using Duration Calculus (DC) [11]. By introducing a formula int *to DC to express the intervals of time between two ticks of the system clock, it is shown that DC can be used to specify both the digital states and the analog states as well as to reason about time in distributed systems. We illustrate our approach by giving an intuitive model for the communication protocols in which the verification can be done by using the familiar natural induction rules.*

1. Introduction

In the Real-Time Hybrid systems, we have to reason about their timed behavior (see, e.g. [8, 9, 10]). Since in the such systems we have to deal with analog components whose states change continuously, and with digital components (computer programs) whose states change only at ticks of the computer clock, we have to deal with both continuous time and discrete time. Here we consider continuous time as the set R^+ of nonnegative real numbers and discrete time as the set N of natural numbers, which is a countable subset of R^+.

The Duration Calculus (DC) [11], an extension of interval temporal logic, is a notation to specify and reason about properties of systems and is a logic to verify theorems about such specifications. In DC, states of a system are modeled by boolean functions of continuous time which have finite variability, meaning that in any finite interval of time there is only a finite number of discontinuous points of the functions. Thus time in DC is continuous and we can reason about the behavioral timing properties of both analog components and computer programs. When restricted to discrete time DC can specify precisely the behavior of computer programs referring to unique clock, and many properties of the systems written in DC are decidable (see [2]). However, in this case it can only specify approximately the behavior of physical analog components, and if the systems are distributed, it cannot specify precisely the behavior of the computer programs that run on different components of the systems with different clocks. Thus, in specifying the hybrid and distributed systems, continuous time should be referred to. A problem with DC when using continuous time is that it cannot specify that a state can *change only at discrete time points*.

To overcome this shortcoming, we introduce a formula int into DC. The formula int is satisfied by an interval $[a, b]$ of time if and only if $a, b \in \mathsf{N}$, i.e., an interval from a tick to another tick of the reference clock. We say that a state is *discrete* if it can change its value only at discrete time points.

With discrete states we can define a digitization of a continuous state. Let us consider a model of communication protocols. In order to send a sequence of bits in $\{0, 1\}^*$, the sender encodes the sequence as a state and put it on the bus. The receiver then digitizes the signal and gets a discrete state according to its clock. Because of some disturbance, such as it taking a significant time to change the signal on the bus from high to low and vice-versa, and because of digitization, the state that the receiver gets is different from the signal sent by the sender. However, there is a relationship between the continuous state sent and the discrete state received, which should be satisfied for a protocol under the physical laws. Let $f(w)$ be a DC formula for the state resulted from encoding a sequence w of bits, D be the discrete state obtained by digitizing $f(w)$, and $g(D)$ be the sequence of bits obtained by decoding D. The relationship can be then expressed in DC as the formula $f(w) \Rightarrow D$. The correctness of the protocol is verified if $g(D) = w$. Hence, we can have a nice model for specification and verification of

*On leave from the Institute of Information Technology, Nghia Do, Tu Liem, Hanoi, Vietnam.

†On leave from POSTECH/CAST, South Korea from January to February 1995

the communication protocols in which the encoding and decoding can be written as functions between formal (regular) languages, the correctness of the protocols can be proved by using the natural induction rule on the length of words in the traditional way.

To illustrate our approach, we try the approach to model and verify the Biphase Mark (BPM) protocol and the Audio Control protocol which are widely used in practice for asynchronous communication between two digital hardware devices. In [7], the Biphase Mark protocol is verified formally using Boyer-Moore Logic for some specific encoding and decoding rules. In our model, however we can verify the BPM protocol for the more general encoding and decoding rules and we can make clear the assumption of the environments, namely we take into account the time it takes to change the signal from high to low and from low to high.

The paper is organized as follows. In Section 2, we present Discrete Duration Calculus (DC^d) which is an extension of DC with discrete states. In Section 3, a general model for physical environments of communication protocols is specified and reasoned in DC^d. Based on the model, a specification and verification of the Biphase Mark protocol in DC^d is presented in Section 4. The last section is a conclusion of the paper.

2. Discrete Duration Calculus: a Duration Calculus with discrete states

In this section, we give an extension of DC, Discrete DC, to reason about the hybridity. First, we give a brief summary of DC and then our extension. For more details of DC, readers are referred to [11, 3].

2.1. Summary of Duration Calculus

Time in DC is the set R^+ of the nonnegative real numbers. For $t, t' \in \mathsf{R}^+ (t \leq t')$, $[t, t']$ denotes the time interval from t to t'. Let $intv$ denote the set of all time intervals.

2.1.1 Syntax of Duration Calculus

Assume that $V = \{X, Y, \ldots\}$ is a set of state variables. We will use the special symbol ℓ to denote the length of a time interval, f_i^n and A_i^n $(i, n \geq 0)$ as n-ary function and n-ary predicate names respectively. The syntax classes *state expressions*, *terms*, and *formulas* will be then defined as follows.

State expressions: The set of state expressions is generated by the grammar

$$P \ \widehat{=}\ 0 \mid 1 \mid v \mid \neg P \mid P \wedge Q\,,$$

where v stands for state names in the set V.

Terms: The set of terms is generated by

$$r \ \widehat{=}\ \textstyle\int P \mid \ell \mid f_i^n(r_1, \ldots, r_n)\,,$$

where P stands for state expressions, f_i^n for an n-ary functions.

Atomic Formulas: if $r_1, \ldots, r_n$ are n terms then $A_i^n(r_1, \ldots, r_n)$ is an atomic formula.

Formulas: the set of formulas is generated by the grammar:

$$D \ \widehat{=}\ A \mid D; D \mid \neg D \mid D \vee D$$

where A stands for atomic formulas.

2.1.2 Semantics of Duration Calculus

Assume that each n-ary function name f_i^n is associated with a total function from R^n to R which is denoted by f_i^n also, and each n-ary predicate name A_i^n is associated with a total function from R^n to $\{tt, ff\}$ which is also denoted by A_i^n. In this paper, for simplicity we interpret the functions f as operators on reals, e.g. $+, *$, and the relations A as comparative operators between reals, e.g. $<, \leq, =, >, \geq$.

An interpretation $\mathcal{I}$ is a function $\mathcal{I} \in (V \to (\mathsf{R}^+ \to \{0,1\}))$, for which each $\mathcal{I}(X)$, $X \in V$ has at most finitely many discontinuity points in any interval $[a, b]$. We shall use the abbreviation $X_\mathcal{I} \widehat{=} \mathcal{I}(X)$. The semantics of state expression, terms, and formulas in an interpretation $\mathcal{I}$ are then defined as follows.

Semantics of state expressions: The semantics of a state expression P in an interpretation $\mathcal{I}$ is a function $\mathcal{I}_P \in Time \to \{0, 1\}$ defined inductively on the structure of state expressions by:

$$\begin{array}{lcl}
\mathcal{I}_0(t) & \widehat{=} & 0, \\
\mathcal{I}_1(t) & \widehat{=} & 1, \\
\mathcal{I}_X(t) & \widehat{=} & X_\mathcal{I}(t), \\
\mathcal{I}_{(\neg P)}(t) & \widehat{=} & 1 - \mathcal{I}_P(t), \text{ and} \\
\mathcal{I}_{(P \vee Q)}(t) & \widehat{=} & \begin{cases} 0 & \text{if } \mathcal{I}_P(t) = 0 \text{ and } \mathcal{I}_Q(t) = 0, \\ 1 & \text{otherwise.} \end{cases}
\end{array}$$

Semantics of terms: The semantics of a term r in an interpretation $\mathcal{I}$ is a function $\mathcal{I}_r \in intv \to \mathsf{R}$ defined inductively on the structure of terms by:

$$\begin{array}{lcl}
\mathcal{I}_{\int P}([a,b]) & \widehat{=} & \int_a^b \mathcal{I}_P(t)dt, \\
\mathcal{I}_\ell([a,b]) & \widehat{=} & b - a, \text{ and} \\
\mathcal{I}_{f_i^n(r_1,\ldots,r_n)}([a,b]) & \widehat{=} & f_i^n(\mathcal{I}_{r_1}([a,b]), \ldots, \mathcal{I}_{r_n}([a,b])).
\end{array}$$

Semantics of formulas: The semantics of a formula D in an interpretation $\mathcal{I}$ is a function $\mathcal{I}_D \in intv \to \{tt, ff\}$ defined inductively on the structure of formulas as follows. Using the following abbreviations:

$$\begin{array}{lcl}
\mathcal{I}, [a,b] \models D & \widehat{=} & \mathcal{I}_D([a,b]) = tt \text{ and} \\
\mathcal{I}, [a,b] \not\models D & \widehat{=} & \mathcal{I}_D([a,b]) = ff,
\end{array}$$

$\mathcal{I}_D$ is defined by:

$\mathcal{I}, [a,b] \models A_i^n(r_1, \ldots, r_n)$ iff
$A_i^n(\mathcal{I}_{r_1}([a,b]), \ldots, \mathcal{I}_{r_n}([a,b])) = tt$,
$\mathcal{I}, [a,b] \models true$ Always,
$\mathcal{I}, [a,b] \models (\neg D)$ iff $\mathcal{I}, [a,b] \not\models D$,
$\mathcal{I}, [a,b] \models (D_1 \vee D_2)$ iff
$\mathcal{I}, [a,b] \models D_1$ or $\mathcal{I}, [a,b] \models D_2$,
$\mathcal{I}, [a,b] \models (D_1; D_2)$ iff
$\mathcal{I}, [a,m] \models D_1$ and
$\mathcal{I}, [m,b] \models D_2$ for some $m \in [a,b]$.

2.1.3 Abbreviations, axioms, and theorems

For a DC formulas D, D_1, and a state expression P, we use the following abbreviations:

$$\begin{array}{rcl} \Diamond D & \widehat{=} & (true; D); true \\ \Box D & \widehat{=} & \neg(\Diamond(\neg D)) \\ \lceil P \rceil & \widehat{=} & \int P = \ell \wedge \ell > 0 \\ \lceil\ \rceil & \widehat{=} & \ell = 0 \\ D \Rightarrow D_1 & \widehat{=} & (\neg D) \vee D_1 \\ D \wedge D_1 & \widehat{=} & \neg((\neg D) \vee (\neg D_1)) \end{array}$$

Some axioms and theorems in the calculus are:

DA 1 $\int 0 = 0$

DA 2 *For an arbitrary state* P, $\int P \geq 0$

DA 3 *For arbitrary states* P *and* Q, $\int P + \int Q = \int(P \vee Q) + \int(P \wedge Q)$

DA 4 *For a state* P, *for nonnegative real numbers* r, s, $(\int P = r + s) \Leftrightarrow (\int P = r; \int P = s)$

(Monotonicity) *If* $D_1 \Rightarrow A_1$ *and* $D_2 \Rightarrow A_2$ *then* $D_1; D_2 \Rightarrow A_1; A_2$

(Associativity) $(D_1; D_2); D_3 \Leftrightarrow D_1; (D_2; D_3)$

(Unit) $\lceil\ \rceil; D \Leftrightarrow D; \lceil\ \rceil \Leftrightarrow D$

(Zero) $false; D \Leftrightarrow D; false \Leftrightarrow false$

(Other) $(\ell = a; A) \wedge (\ell = a; B) \Rightarrow \ell = a; A \wedge B$.

2.2. Duration calculus with discrete states

Since we want to reason about discrete time in the same framework, we use the set N of natural numbers as a subset of R^+ to express discrete time. The set $Nintv$ of all intervals over R^+ with the endpoints in N is defined by $Nintv \widehat{=} \{[a,b] \mid a, b \in \mathsf{N}\}$, and then $Nintv \subset intv$. We introduce a formula int into DC to express the intervals in $Nintv$ resulting in a "discrete DC". It turns out later that discrete DC (DC^d) then is powerful enough to deal with discrete variables.

The syntax of DC^d is:

$$D \widehat{=} A \mid int \mid D; D \mid \neg D \mid D \vee D,$$

where A stands for atomic formulas of DC.

The semantics for DC^d is defined in the same way as for DC with semantics of int defined by:

$$\mathcal{I}, [a,b] \models int \text{ iff } [a,b] \in Nintv.$$

Axioms and rules for DC^d are those of DC. In addition, DC^d needs some others for int.

I1 $\ell \geq 1 \Rightarrow \ell < 1; int; \ell < 1$

I2 *For any formula* B, $int; B; int \Rightarrow int; (B \wedge int); int$

I3 *For any formulas* A *and* B,

$$\Rightarrow \quad \begin{pmatrix} (\ell < 1; (A \wedge (int; \ell < 1)))\wedge \\ (\ell < 1; (B \wedge (int; \ell < 1))) \end{pmatrix} \\ \ell < 1; \begin{pmatrix} A \wedge B \wedge \\ (int; \ell < 1) \end{pmatrix},$$

$$\Rightarrow \quad \begin{pmatrix} ((A \wedge (\ell < 1; int)); \ell < 1)\wedge \\ ((B \wedge (\ell < 1; int)); \ell < 1) \end{pmatrix} \\ \begin{pmatrix} (A \wedge B \wedge \\ (\ell < 1; int)) \end{pmatrix}; \ell < 1$$

I4 $int \wedge \ell > 0 \Leftrightarrow \ell = 1; int$ *and* $int \wedge \ell > 0 \Leftrightarrow int; \ell = 1$

I5 *For any formula* D, $int \Rightarrow (int; D \Rightarrow int; (D \wedge int))$ *and* $int \Rightarrow (D; int \Rightarrow (D \wedge int); int)$

(Natural Induction) *Let* X *be a formula letter occurring in the formula* $D(X)$. *Then*
If $D(int \wedge \ell = 0)$ *holds and* $D((int \wedge X) \vee (int \wedge X); \ell = 1)$ *is provable from* $D(int \wedge X)$, *then* $D(int)$ *holds,*
If $D(int \wedge \ell = 0)$ *holds and* $D((int \wedge X) \vee \ell = 1; (int \wedge X))$ *is provable from* $D(int \wedge X)$, *then* $D(int)$ *holds.*

It is easy to prove that the axioms introduced for int are sound.

Now we give some rules that will be used in this paper. The soundness of the rules can be proved easily from the definition of DC^d semantics.

Some additional rules: *For any state variables* P, Q, *for any formulas* D_1, D_2, D_1', D_2'

R1 *If* $P \Rightarrow Q$ *then* $\lceil P \rceil \Rightarrow \lceil Q \rceil$

R2 $(\lceil P \rceil \wedge \lceil Q \rceil) \Rightarrow \lceil P \wedge Q \rceil$

R3 $(\lceil P \rceil; true) \wedge (\lceil Q \rceil; true) \Rightarrow \lceil P \wedge Q \rceil; true$, *and* $(true; \lceil P \rceil) \wedge (true; \lceil Q \rceil) \Rightarrow true; \lceil P \wedge Q \rceil$

R4

$$\left(\begin{array}{c}\left(\begin{array}{c}D_1\wedge\\(true;\lceil P\rceil\wedge\ell\geq a)\end{array}\right);\\\left(\begin{array}{c}D_2\wedge\\(\lceil\neg P\rceil\wedge\ell\geq b;B)\end{array}\right)\end{array}\right)$$
$$\wedge$$
$$\left(\begin{array}{c}\left(\begin{array}{c}D'_1\wedge\\(true;\lceil P\rceil\wedge\ell\geq a)\end{array}\right);\\\left(\begin{array}{c}D'_2\wedge\\(\lceil\neg P\rceil\wedge\ell\geq b;B)\end{array}\right)\end{array}\right)$$
$$\Rightarrow\quad (D_1\wedge D'_1);(D_2\wedge D'_2),$$
$$\left(\begin{array}{c}\left(\begin{array}{c}D_2\wedge\\B;\lceil P\rceil\wedge\ell\geq a)\end{array}\right);\\\left(\begin{array}{c}D_1\wedge\\(\lceil\neg P\rceil\wedge\ell\geq b;true)\end{array}\right)\end{array}\right)$$
$$\wedge$$
$$\left(\begin{array}{c}\left(\begin{array}{c}D'_2\wedge\\B;\lceil P\rceil\wedge\ell\geq a)\end{array}\right);\\\left(\begin{array}{c}D'_1\wedge\\(\lceil\neg P\rceil\wedge\ell\geq b;true)\end{array}\right)\end{array}\right)$$
$$\Rightarrow\quad (D_2\wedge D'_2);(D_1\wedge D'_1)\,,$$

where $B\mathrel{\widehat{=}}\neg\Diamond(\lceil P\rceil\wedge\ell\geq a;\lceil\neg P\rceil\wedge\ell\geq b)\,.$

The rule **R4** needs some explanation. It says that the first or the last change of a state in an interval which satisfies a specific property will determine a unique point in the interval.

Discrete states

The states that can be discontinuous at discrete time points only are called *discrete states*. Such states could be assertions of program states, for instance, $x=6$, where x is a program variable. To specify that P is a discrete state, we can use the DC formula

$$I(P)\mathrel{\widehat{=}} int\wedge\ell=1\Rightarrow\lceil P\rceil\vee\lceil\neg P\rceil.$$

The following theorem can be proved easily.

Theorem 1. *Assume that* $I(P),\ I(Q)$ *hold. Let* D, D' *be* DC^d *formulas. Then*

1. $I(P\wedge Q),\ I(\neg P)$
2. $int\wedge(\lceil P\rceil;\lceil Q\rceil)\Rightarrow(\lceil P\rceil\wedge int);(int\wedge\lceil Q\rceil)$
3. $\lceil P\rceil;\lceil\neg P\rceil;\lceil P\rceil\Rightarrow\lceil P\rceil;(int\wedge\lceil\neg P\rceil);\lceil P\rceil\,.$

Digitization of states

The idea of digitization here is taken from [6]. Suppose that a device monitors the state Q over cycles of clock, returning 1 if Q is true over the whole cycle, 0 if it is false for the whole cycle, and any value chosen nondeterministically in $\{0,1\}$ otherwise. The result is said to be a digitization of Q. We could define formally the digitization as follows. A state P is a digitization of a state Q iff $I(P)$ and

$$int\wedge\ell=1\Rightarrow\lceil Q\rceil\Rightarrow\lceil P\rceil\text{ and}$$
$$int\wedge\ell=1\Rightarrow\lceil\neg Q\rceil\Rightarrow\lceil\neg P\rceil.$$

More generally, for a state P and a discrete state R_P we define a discrete state S_P to be a sampling state of P w.r.t. R_P by:

$$I(S_P)\wedge I(R_P),$$
$$\lceil R_P\Rightarrow(P\Leftrightarrow S_P)\rceil,\text{ and}$$
$$int\wedge\ell=1\Rightarrow(\lceil P\rceil\vee\lceil\neg P\rceil\Rightarrow\lceil R_P\rceil).$$

With these discrete state and digitization concepts, a general model of physical environment where a communication protocol works is defined in the next section.

3. A model of communication at physical layer

Consider a model for communication at the physical layer. A sender and a receiver are connected via a bus. Their clocks are running at different rates. We refer to the clock of the receiver as a reference time. The receiver receives signals by digitizing. The signals sent by the sender are modeled by a state X (we recall that a state is a boolean function from R^+ to $\{0,1\}$). Let Y be the state representing signals received by the receiver by digitizing the signals on the bus. Without loss of generality, assume that the delay between the sender and the receiver is 0. Due to the fact that it takes a significant amount of time to change the signal on the bus from high to low or vice-versa, Y is not only for expressing the digitization of X as shown in Figure 1. The value of Y is chosen arbitrarily from $\{0,1\}$ if the digitized value is neither 0 nor 1. Let R be the state expressing the reliability of the digitization. Then Y and R are discrete states according to the receiver's clock, while X is not (although it can be considered as a discrete state according to the sender clock) (see Figure 1). In this paper we assume that the states expressing signals are continuous from the left hand side.

Since a state can be specified by a DC formula, a communication protocol can be modeled as consisting of a language $\mathcal{L}_s$ whose elements are DC formulas over state variable X, a language $\mathcal{L}_r$ whose elements are DC formulas over discrete state variable Y, a coding function $f:\{0,1\}^+\to\mathcal{L}_s$ and a decoding function $g:\mathcal{L}_r\to\{0,1\}^+$ such that

1. for any $w\in\{0,1\}^+$ there exists a formula $D\in\mathcal{L}_r$ for which $f(w)\Rightarrow D$, and
2. for any $D\in\mathcal{L}_r$, if $f(w)\Rightarrow D$ then $g(D)=w$.

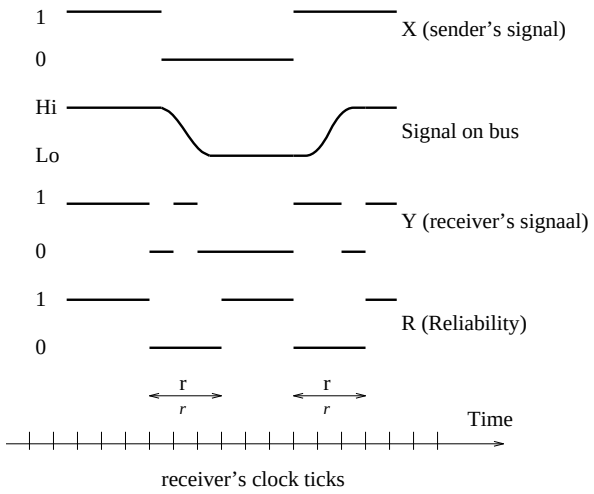

Figure 1. Signal Patterns

Let us use DC^d to express the relationship between the states X, R and Y before specifying any communication protocol.

(i) *Y and R are discrete states, and R expresses the reliability of the digitization*:

D1 $I(Y) \wedge I(R)$,

D2 $\lceil R \Rightarrow (X \Leftrightarrow Y)\rceil$,

D3 $int \wedge \ell = 1 \wedge (\lceil X\rceil;\lceil\neg X\rceil \vee \lceil\neg X\rceil;\lceil X\rceil) \Rightarrow \lceil\neg R\rceil$.

(ii) *Assumptions about physical laws and digitizations*: Let us adopt the following phenomenon: If the sender changes the signal from low to high or high to low, the receiver's signal will be unreliable for r cycles, $r \geq 1$, $r \in \mathsf{N}$ during which it can be any value chosen nondeterministically among 0 and 1; otherwise, if the sender keeps the signal stable then the same signal will be stable for the receiver (see Figure 1). These assumptions can be expressed by the following DC^d formulas:

A1

$$\begin{array}{rl} & \ell \geq r+1 \wedge (\lceil X\rceil \vee \lceil\neg X\rceil) \\ \Rightarrow & \ell \leq r; \left(\begin{array}{c} int \wedge \\ R \end{array}\right); \ell < 1, \end{array}$$

A2

$$\begin{array}{rl} & (\ell \geq r+1 \wedge \lceil X\rceil);(\ell \geq r+1 \wedge \lceil\neg X\rceil) \\ \Rightarrow & \left(\begin{array}{c} \ell \leq r; \left(\begin{array}{c} int \wedge \\ \lceil R\rceil \end{array}\right); \left(\begin{array}{c} int \wedge \\ \lceil\neg R\rceil \wedge \\ \ell = r \end{array}\right); \\ \left(\begin{array}{c} int \wedge \\ \lceil R\rceil \end{array}\right); \ell < 1 \end{array}\right), \end{array}$$

$$\begin{array}{rl} & (\ell \geq r+1 \wedge \lceil\neg X\rceil);(\ell \geq r+1 \wedge \lceil X\rceil) \\ \Rightarrow & \left(\begin{array}{c} \ell \leq r; \left(\begin{array}{c} int \wedge \\ \lceil R\rceil \end{array}\right); \left(\begin{array}{c} int \wedge \\ \lceil\neg R\rceil \wedge \\ \ell = r \end{array}\right); \\ \left(\begin{array}{c} int \wedge \\ \lceil R\rceil \end{array}\right); \ell < 1 \end{array}\right). \end{array}$$

The assumption **A1** says that if the sender stays high or low for enough long time then the signal received by the receiver is also the same and stable for enough time. The assumption **A2** says that if the sender changes the signal then the signal received by the receiver may not be stable for r sampling cycles provided the sender has kept the signal stable before the change and keeps it stable after the change for enough long time. From these assumptions it follows that if the signal changes at a tick of the receiver's clock then it will not be stable for r cycles starting from the tick.

There are only two possibilities for coding information: changes of the signal from high to low or low to high, which are called edges, and distances between edges.

An edge is expressed in DC as:

1. for the sender

$$\lceil X\rceil;\lceil\neg X\rceil \text{ or } \lceil\neg X\rceil;\lceil X\rceil$$

2. for the receiver

$$\lceil Y\rceil;\lceil\neg Y\rceil \text{ or } \lceil\neg Y\rceil;\lceil Y\rceil$$

From the assumption **A2**, an edge of the sender, in an interval that is long enough, can introduce several edges of the receiver. However, if we consider those edges whose first signal is stable for a long time then we can have a one-to-one corresponding between edges of the sent signal and received signal (see Corollary 1). If we refer to an edge which has a long stable first signal as a *long edge*, we can formally define long edges by

$$\begin{array}{l} l\_Hi\_lo_s \mathrel{\widehat{=}} (\ell \geq 2r+1 \wedge \lceil X\rceil);(\ell \geq r+1 \wedge \lceil\neg X\rceil), \\ l\_Lo\_Hi_s \mathrel{\widehat{=}} (\ell \geq 2r+1 \wedge \lceil\neg X\rceil);(\ell \geq r+1 \wedge \lceil X\rceil), \\ l\_Hi\_Lo_r \mathrel{\widehat{=}} (\ell \geq r+1 \wedge \lceil Y\rceil);(\ell = 1 \wedge \lceil\neg Y\rceil), \\ l\_Lo\_Hi_r \mathrel{\widehat{=}} (\ell \geq r+1 \wedge \lceil\neg Y\rceil);(\ell = 1 \wedge \lceil Y\rceil). \end{array}$$

Corollary 1.

$$\begin{array}{l} l\_Hi\_lo_s \Rightarrow \Diamond l\_Hi\_Lo_r \wedge \neg(\Diamond l\_Hi\_Lo_r;\Diamond l\_Hi\_Lo_r)\,, \\ l\_Lo\_Hi_s \Rightarrow \Diamond l\_Lo\_Hi_r \wedge \neg(\Diamond l\_Lo\_Hi_r;\Diamond l\_Lo\_Hi_r)\,. \end{array}$$

This means that there is a one-to-one correspondence between long edges of the sender and the receiver, and the first long edge of the receiver is the corresponding edge. Notice that long edges of the same kind (low to high or high to low) can be distinguished from each other. The corollary says that long edges can be used in encoding information. Actually we can have a more precise relationship between X and Y, as in the following corollaries, which will be used in proving the correctness of the Biphase Mark protocol later.

Corollary 2. *Let* $a \geq r+1,\ b \geq r+1,\ c \geq r+1$. *Then*

$$(\lceil X \rceil \wedge \ell = a); (\lceil \neg X \rceil \wedge \ell = b); (\lceil X \rceil \wedge \ell = c) \Rightarrow$$

$$\left(\begin{array}{c} \ell \leq 1; \left(\begin{array}{c} int \wedge \\ \ell = r-1 \end{array} \right); \left(\begin{array}{c} int \wedge \\ \lceil R \wedge X \rceil \wedge \\ |\ell - a + r| < 1 \end{array} \right); \\ \left(\begin{array}{c} int \wedge \\ \lceil \neg R \rceil \wedge \\ \ell = r \end{array} \right); \left(\begin{array}{c} int \wedge \\ \lceil R \wedge \neg X \rceil \wedge \\ |\ell - b + r| < 1 \end{array} \right); \\ \left(\begin{array}{c} int \wedge \\ \lceil \neg R \rceil \wedge \\ \ell = r \end{array} \right); \left(\begin{array}{c} int \wedge \\ \lceil R \wedge X \rceil \wedge \\ |\ell - c + r| < 1 \end{array} \right); \ell < 1 \end{array} \right)$$

Corollary 3. *For* $a \geq r+1$, $b \geq r+1$, *for any* $d \in \mathbf{N}$ *and* c *such that* $b + r \leq d \leq c + b - r - 3$,

$$(\lceil X \rceil \wedge \ell = a); (\lceil \neg X \rceil \wedge \ell = b); (\lceil X \rceil \wedge \ell = c) \Rightarrow$$

$$\left(\begin{array}{c} \ell \leq 1; \left(\begin{array}{c} int \wedge \\ \ell = r-1 \end{array} \right); \\ \left(\begin{array}{c} int \wedge \\ \lceil Y \rceil \wedge \\ \ell > a - r - 1 \end{array} \right); \left(\begin{array}{c} int \wedge \\ \lceil \neg Y \rceil \wedge \\ \ell = 1 \end{array} \right) \end{array} \right);$$

$$\left(\begin{array}{c} int \wedge \\ (\ell = d; \left(\begin{array}{c} int \wedge \\ \lceil Y \rceil \wedge \\ \ell > c + b - 3 - r - d \end{array} \right)) \\ \wedge (true; \left(\begin{array}{c} int \wedge \\ \lceil Y \rceil \wedge \\ \ell > c - r - 1 \end{array} \right)) \end{array} \right); \ell < 1$$

$$(\lceil X \rceil \wedge \ell \geq a); (\lceil \neg X \rceil \wedge \ell = b); (\lceil \neg X \rceil \wedge \ell = c) \Rightarrow$$

$$\left(\begin{array}{c} \ell \leq 1; \left(\begin{array}{c} int \wedge \\ \ell = r-1 \end{array} \right); \\ \left(\begin{array}{c} int \wedge \\ \lceil Y \rceil \wedge \\ \ell > a - r - 1 \end{array} \right); \left(\begin{array}{c} int \wedge \\ \lceil \neg Y \rceil \wedge \\ \ell = 1 \end{array} \right) \end{array} \right);$$

$$\left(\begin{array}{c} int \wedge \\ (\ell = d; \left(\begin{array}{c} int \wedge \\ \lceil \neg Y \rceil \wedge \\ \ell > c + b - 3 - r - d \end{array} \right)) \\ \wedge (true; \left(\begin{array}{c} int \wedge \\ \lceil \neg Y \rceil \wedge \\ \ell > c - r - 1 \end{array} \right)) \end{array} \right); \ell < 1$$

The corollaries above are of importance in designing communication protocols. A formal proof of them in DC can be found in [5].

For a protocol, to make the algorithm concrete and deterministic, long edges for the sender are defined as follows:

$$\begin{array}{lcl} Hi\_Lo_s & \widehat{=} & (\lceil X \rceil \wedge \ell = a); (\lceil \neg X \rceil \wedge \ell = b) \text{ and} \\ Lo\_Hi_s & \widehat{=} & (\lceil \neg X \rceil \wedge \ell = a); (\lceil X \rceil \wedge \ell = b). \end{array}$$

In the next sections, we give a formal specification and verification of some concrete protocols as an application of our model.

4. Biphase Mark Protocols

In this section, a formal model for encoding and decoding rules of the Biphase Mark (BPM) protocol is provided, and it is also verified that with some restriction to the rules the BPM protocol is always correct.

The BPM protocol encodes a bit as a cell consisting of a mark subcell and a code subcell. If the signal in the mark subcell is the same as the one in the code subcell then the information carried by that cell is 0. Otherwise, if the signals are different then the information carried is 1. There is a phase reverse between two consecutive cells. It means that the signal at the beginning of the following cell is held as the negation of the signal at the end of the previous cell. The receiver, after detecting the beginning of a cell, skips some cycles called as *sampling distance* and samples the signal. If the sampled signal is the same as the signal at the beginning of the cell, it decodes the cell as 0; otherwise it decodes the cell as 1. Figure 2 illustrates the sending and the receiving of the sequence $w = 01$.

We will formalize the protocol by using Discrete Duration Calculus and the denotations from the theory of formal languages (see, e.g. [4]). From now on, in order to simplify our denotations, we will identify regular languages with their regular expressions.

Let us denote in this section,

$$Hi\_Lo_r \ \widehat{=} \ \left(\begin{array}{c} int \wedge \\ \lceil Y \rceil \wedge \\ 1 \leq \ell \leq c' - 1 \end{array} \right); \left(\begin{array}{c} int \wedge \\ \lceil \neg Y \rceil \wedge \\ \ell = 1) \end{array} \right),$$

$$Lo\_Hi_r \ \widehat{=} \ \left(\begin{array}{c} int \wedge \\ \lceil \neg Y \rceil \wedge \\ 1 \leq \ell \leq c' - 1 \end{array} \right); \left(\begin{array}{c} int \wedge \\ \lceil Y \rceil \wedge \\ \ell = 1 \end{array} \right),$$

where c' is a natural number, $c' > 1$.

Encoding information

From now on, to distiguish with the symbols from the alphabets we use the symboles $\oplus$, **(**, **)** and $^+$ as standard connectives for building regular exspressions. Let Σ_s be the

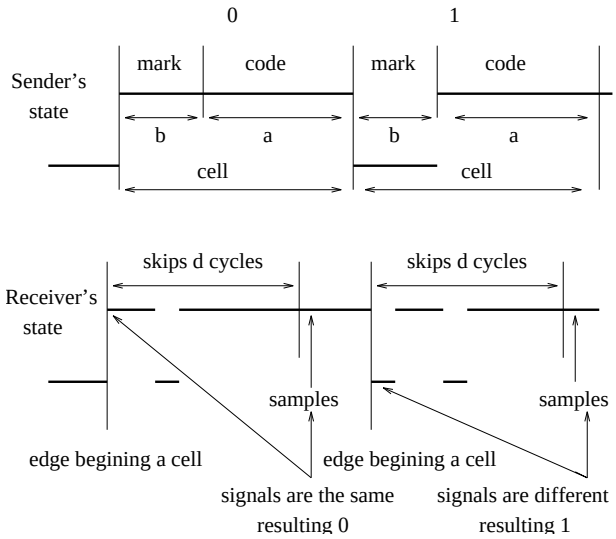

Figure 2. Biphase Mark Protocol

alphabet

$$\Sigma_s \;\widehat{=}\; \left\{ \begin{array}{c} \lceil\neg X\rceil, Lo\_Hi_s, Hi\_Lo_s, ;, \\ (\lceil\neg X\rceil \wedge \ell \geq c), (\lceil X\rceil \wedge \ell \geq c) \end{array} \right\},$$

and let $\mathcal{L}_s$ be the set of DC formulas represented by the following regular expression over Σ_s,

$$\begin{array}{rcl} \mathcal{L}_s & \widehat{=} & \lceil\neg X\rceil; Lo\_Hi_s; \mathbf{(}Hi\_Lo_s; \oplus Lo\_Hi_s; \mathbf{)^+} \\ & & \mathbf{(}(\lceil\neg X\rceil \wedge \ell \geq c) \oplus (\lceil X\rceil \wedge \ell \geq c)\mathbf{)}. \end{array}$$

The coding function $f : \{0,1\}^+ \rightarrow \mathcal{L}_s$ is defined inductively as follows.
Basic:

$$\begin{array}{rcl} f(0) & \widehat{=} & \lceil\neg X\rceil; Lo\_Hi_s; (\lceil X\rceil \wedge \ell \geq c)\,, \\ f(1) & \widehat{=} & \lceil\neg X\rceil; Lo\_Hi_s; (\lceil\neg X\rceil \wedge \ell \geq c)\,. \end{array}$$

Induction:

F1 Let $f(w) = D; (\lceil X\rceil \wedge \ell \geq c) (\in \mathcal{L}_s)$, where $D \in \Sigma_s^*$. Then,

$$\begin{array}{rcl} f(w0) & \widehat{=} & D; Hi\_Lo_s; (\lceil\neg X\rceil \wedge \ell \geq c)\,, \\ f(w1) & \widehat{=} & D; Hi\_Lo_s; (\lceil X\rceil \wedge \ell \geq c)\,. \end{array}$$

F2 Let $f(w) = D; (\lceil\neg X\rceil \wedge \ell \geq c)$. Then,

$$\begin{array}{rcl} f(w0) & \widehat{=} & D; Lo\_Hi_s; (\lceil X\rceil \wedge \ell \geq c)\,, \\ f(w1) & \widehat{=} & D; Lo\_Hi_s; (\lceil\neg X\rceil \wedge \ell \geq c)\,. \end{array}$$

For example, by the definition of f,

$$f(01) \;=\; \lceil\neg X\rceil; Lo\_Hi_s; Hi\_Lo_s; (\lceil X\rceil \wedge \ell \geq c)$$

which expresses the sender's state in Fig. 2.

Decoding information

Let

$$\Sigma_r \;\widehat{=}\; \left\{ \begin{array}{c} \ell \leq 1\,,\, (int \wedge \ell = r-1)\,,\, ;\,, \\ Lo\_Hi_r, \ell = d\,, \\ Hi\_Lo_r\,,\, \ell < 1\,, \\ \lceil\neg Y\rceil\,,\, (\lceil\neg Y\rceil \wedge \ell \geq c' \wedge int)\,, \\ (\lceil Y\rceil \wedge \ell \geq c' \wedge int) \end{array} \right\},$$

where d is a natural number. The decoding function $g : \mathcal{L}_r \rightarrow \{0,1\}^+$ is defined below, where $\mathcal{L}_r$ is the set of DC formulas represented by the following regular expression over Σ_r

$$\begin{array}{rcl} \mathcal{L}_r & \widehat{=} & \ell \leq 1; (int \wedge \ell = r-1); (\lceil\neg Y\rceil \wedge int); \\ & & Lo\_Hi_r; \ell = d; \\ & & \mathbf{((}Lo\_Hi_r \;\oplus\; Hi\_Lo_r\mathbf{)}; \ell = d; \mathbf{)^*} \\ & & \mathbf{(}(\lceil\neg Y\rceil \wedge \ell \geq c' \wedge int) \\ & & \oplus (\lceil Y\rceil \wedge \ell \geq c' \wedge int); \mathbf{)}; \ell < 1. \end{array}$$

First, we define a function g', $g' : ((Lo\_Hi_r \oplus Hi\_Lo_r); \ell = d;)^+ ((\lceil\neg Y\rceil \wedge \ell \geq c' \wedge int) \oplus (\lceil Y\rceil \wedge \ell \geq c' \wedge int)); \ell < 1 \rightarrow \{0,1\}^*$ inductively as follows.
Basic:

1. Let $D = Hi\_Lo_r; \ell = d; (\lceil Y\rceil \wedge \ell \geq c' \wedge int); \ell < 1$ then $g'(D) = 1$
2. Let $D = Hi\_Lo_r; \ell = d; (\lceil\neg Y\rceil \wedge \ell \geq c' \wedge int); \ell < 1$ then $g'(D) = 0\,,$
3. Let $D = Lo\_Hi_r; \ell = d; (\lceil Y\rceil \wedge \ell \geq c' \wedge int); \ell < 1$ then $g'(D) = 0\,,$
4. Let $D = Lo\_Hi_r; \ell = d; (\lceil\neg Y\rceil \wedge \ell \geq c' \wedge int); \ell < 1$ then $g'(D) = 1\,.$

Induction:

1. Let $D = Lo\_Hi_r; \ell = d; Lo\_Hi_r; D'$ then $g'(D) = 1g'(Lo\_Hi_r; D')\,,$
2. Let $D = Lo\_Hi_r; \ell = d; Hi\_Lo_r; D'$ then $g'(D) = 0g'(Hi\_Lo_r; D')\,,$
3. Let $D = Hi\_Lo_r; \ell = d; Lo\_Hi_r; D'$ then $g'(D) = 0g'(Lo\_Hi_r; D')\,,$
4. Let $D = Hi\_Lo_r; \ell = d; Hi\_Lo_r; D'$ then $g'(D) = 1g'(Hi\_Lo_r; D')\,.$

Let $D = \ell \leq 1; (int \wedge \ell = r-1); Lo\_Hi_r; D'$. Then the decoding function g is defined as

$$g(D) \,\widehat{=}\, g'(Lo\_Hi_r; D')\,.$$

For axample, suppose that the sender wants to send a sequence of bits 011. It encodes the sequence as the DC formulas $f(011)$,

$$\begin{array}{rcl} f(011) & = & (\lceil \neg X \rceil \wedge \ell \geq c); Lo\_Hi_s; Hi\_Lo_s; Hi\_Lo_s; \\ & & (\lceil X \rceil \wedge \ell \geq c)\,. \end{array}$$

If we can prove that

$$\begin{array}{rcl} f(011) & \Rightarrow & D, \text{ where} \\ D & \widehat{=} & \ell \leq 1; (int \wedge \ell = r-1); (\lceil \neg Y \rceil \wedge int); \\ & & Lo\_Hi_r; \ell = d; Hi\_Lo_r; \ell = d; Hi\_Lo_r; \\ & & \ell = d; (\lceil Y \rceil \wedge \ell \geq c' \wedge int); \ell < 1\,, \end{array}$$

then the same sequence 011 will be received by the receiver, since according to the definition of g, we have

$$g(D) \quad = \quad 011\,.$$

A verification which shows that the BPM protocol is correct with some restriction to the encoding and decoding rules, i.e., to the lengths of a, b, c, c', and d, is given below.

Verification

To verify that the protocol for the case $b + r \leq d \leq a + b - r - 3$, $b \geq r + 1$, $a \geq 2r + 1$, $c \geq c' + 1$ and $c' \geq a + r + 1$ is correct, we have to prove:

1. For all $D, D' \in \mathcal{L}_r$ such that D and D' are different from each other syntactically, $D \wedge D' \Rightarrow$ *false*
2. For all $w \in \{0,1\}^+$, there exists a formula $D \in \mathcal{L}_r$ such that $f(w) \Rightarrow D$ and $g(D) = w$.

Proof of item 1

Let $\mathcal{R}$ be the following regular expression over Σ_r:

$$\mathcal{R} \widehat{=} Lo\_Hi_r; \ell = d\mathbf{(;(}Lo\_Hi_r \ \oplus \ Hi\_Lo_r\mathbf{)}; \ell = d\mathbf{)^*}\,.$$

For any $C \in \mathcal{R}$, using the rules **R1**-**R4** it can be proved easily by the induction on the length of the 'word' C that for any DC^d formulas D, D',

$$(C; D) \wedge (C; D') \Rightarrow C; (D \wedge D')\,. \qquad (1)$$

Now, let D, D' be different elements of in $\mathcal{L}_r$ (syntactically). Then, there exists $C \in \mathcal{R}$ such that

$$\begin{array}{rcl} D & = & \ell \leq 1; (int \wedge \ell = r-1); (\lceil \neg Y \rceil \wedge int); C; A; E \\ D' & = & \ell \leq 1; (int \wedge \ell = r-1); (\lceil \neg Y \rceil \wedge int); C; A'; E'\,, \end{array}$$

where A and A' are different symbols of Σ_r, E, E are DC formulas. By applying rules **I3**, **R1**-**R4** and (1),

$$D \wedge D' \Rightarrow \ell \leq 1; (int \wedge \ell = r-1); C; ((A; E) \wedge (A'; E'))\,.$$

We have

$$((A; E) \wedge (A'; E'))$$

$$\Rightarrow \left(\begin{array}{c} ((Lo\_Hi_r; true) \wedge (Hi\_Lo_r; true)) \vee \\ \left(\begin{array}{c} (Lo\_Hi_r; true) \wedge \\ (\lceil Y \rceil \wedge int \wedge \ell \geq c'); true) \end{array} \right) \vee \\ \left(\begin{array}{c} (Hi\_Lo_r; true) \wedge \\ ((\lceil Y \rceil \wedge int \wedge \ell \geq c'); true) \end{array} \right) \vee \\ \left(\begin{array}{c} (Lo\_Hi_r; true) \wedge \\ ((\lceil \neg Y \rceil \wedge int \wedge \ell \geq c'); true) \end{array} \right) \vee \\ \left(\begin{array}{c} (Hi\_Lo_r; true) \wedge \\ ((\lceil \neg Y \rceil \wedge int \wedge \ell \geq c'); true) \end{array} \right) \vee \\ \left(\begin{array}{c} ((\lceil \neg Y \rceil \wedge int \wedge \ell \geq c'); true)) \wedge \\ ((\lceil Y \rceil \wedge int \wedge \ell \geq c'); true) \end{array} \right) \end{array} \right)$$

By the rule **R3**, DC rule 9, and the definitions of Lo_Hi_r, Hi_Lo_r all the disjuncts of the disjunction are equivalent to *false*. Thus,

$$((A; E) \wedge (A'; E')) \quad \Rightarrow \textit{false}.$$

Consequently by the DC rule 8 (**Zero**),

$$\begin{array}{rcl} D \wedge D' & \Rightarrow & \ell \leq 1; (int \wedge \ell = r-1); \\ & & (\lceil \neg Y \rceil \wedge int); C; \textit{false} \\ & \Rightarrow & \textit{false}. \end{array}$$

Proof of item 2

First, we give an another definition of the decoding function g via the following lemma.

Lemma 1. *Let D be a word of the language $\mathcal{L}_r$ such that $g(D) = w$. If $D = D'; (\lceil Y \rceil \wedge int) \wedge \ell \geq c'; \ell < 1$, then*

G1 $g(D'; Hi\_Lo_r; \ell = d; (\lceil Y \rceil \wedge int \wedge \ell \geq c'); \ell < 1) = w1$

G2 $g(D'; Hi\_Lo_r; \ell = d; (\lceil \neg Y \rceil \wedge int \wedge \ell \geq c'); \ell < 1) = w0\,.$

If $D = D'; (\lceil \neg Y \rceil \wedge int \wedge \ell \geq c'); \ell < 1$, then

G3 $g(D'; Lo\_Hi_r; \ell = d; (\lceil Y \rceil \wedge int \wedge \ell \geq c'); \ell < 1) = w0$

G4 $g(D'; Lo\_Hi_r; \ell = d; (\lceil \neg Y \rceil \wedge int \wedge \ell \geq c'); \ell < 1) = w1$

Proof: The lemma is proved easily by the induction on the length of the 'word' D using the definition of the function g and is omitted here.

The proof of item 2 now is by induction on the length of w. The inductive assertion is that if $f(w) = B; (\lceil X \rceil \wedge \ell \geq c)$ then we can find a 'word' $D' \in \mathcal{L}_r$,

$$D' = A; \ell = d; (\lceil Y \rceil \wedge \ell \geq c' \wedge int); \ell < 1$$

such that

$$
\begin{array}{l}
g(D') = w, \\
f(w) \Rightarrow D', \\
B; (\lceil X \rceil \wedge \ell = a) \Rightarrow \\
A; \left(\begin{array}{c} int \wedge \\ (\ell = d; (\lceil Y \rceil \wedge \ell \geq 1)) \wedge \\ (true; (\lceil Y \rceil \wedge \ell \geq a - r - 1)) \end{array} \right); \ell < 1,
\end{array}
$$

and if $f(w) = B; (\lceil \neg X \rceil \wedge \ell \geq c)$ then we can find a 'word' $D' \in \mathcal{L}_r$

$$D' = A; \ell = d; (\lceil \neg Y \rceil \wedge \ell \geq c' \wedge int); \ell < 1$$

such that

$$
\begin{array}{l}
g(D') = w, \\
f(w) \Rightarrow D', \\
B; (\lceil \neg X \rceil \wedge \ell = a) \Rightarrow \\
A; \left(\begin{array}{c} int \wedge \\ (\ell = d; \lceil (\neg Y \rceil \wedge \ell \geq 1)) \wedge \\ (true; (\lceil \neg Y \rceil \wedge \ell \geq a - r - 1)) \end{array} \right); \ell < 1.
\end{array}
$$

Basic case: Let $|w| = 1$, where $|w|$ denotes the length of w. By the definition of the encoding function f:

$$
\begin{array}{lcl}
f(1) & = & \lceil \neg X \rceil; Lo\_Hi_s; (\lceil X \rceil \wedge \ell \geq c), \\
f(0) & = & \lceil \neg X \rceil; Lo\_Hi_s; (\lceil \neg X \rceil \wedge \ell \geq c).
\end{array}
$$

Now, by Corollary 3,

$$
\begin{array}{l}
\lceil \neg X \rceil; Lo\_Hi_s; (\lceil X \rceil \wedge \ell \geq c) \Rightarrow \\
\ell \leq 1; (int \wedge \ell = r - 1); (\lceil \neg Y \rceil \wedge int); Lo\_Hi_r; \ell = d; \\
(\lceil Y \rceil \wedge \ell \geq c' \wedge int); \ell < 1.
\end{array}
$$

Denote the conclusion of the implication by D. By the definition of g, we have $g(D) = 1$. Also by Corollary 3,

$$
\begin{array}{rl}
 & \lceil \neg X \rceil; Lo\_Hi_s; (\lceil X \rceil \wedge \ell = a) \\
\Rightarrow & \ell \leq 1; (int \wedge \ell = r - 1); (\lceil \neg Y \rceil \wedge int); Lo\_Hi_r; \\
 & \left(\begin{array}{c} int \wedge \\ (\ell = d; \lceil Y \rceil) \wedge \\ true; \left(\begin{array}{c} int \wedge \\ \lceil Y \rceil \wedge \\ \ell \geq a - r - 1 \end{array} \right) \end{array} \right); \ell < 1.
\end{array}
$$

Similarly,

$$
\begin{array}{l}
f(0) \Rightarrow \\
\ell \leq 1; (int \wedge \ell = r - 1); (\lceil \neg Y \rceil \wedge int); Lo\_Hi_r; \ell = d; \\
(\lceil \neg Y \rceil \wedge \ell \geq c' \wedge int); \ell < 1,
\end{array}
$$

and

$$
\begin{array}{l}
g(\ell \leq 1; (int \wedge \ell = r - 1); (\lceil \neg Y \rceil \wedge int); Lo\_Hi_r; \\
\ell = d; (\lceil \neg Y \rceil \wedge \ell \geq c' \wedge int); \ell \leq 1) = 0,
\end{array}
$$

$$
\begin{array}{l}
\lceil \neg X \rceil; Lo\_Hi_s; (\lceil \neg X \rceil \wedge \ell = a) \Rightarrow \\
\ell \leq 1; (int \wedge \ell = r - 1); (\lceil \neg Y \rceil \wedge int); Lo\_Hi_r; \\
\left(\begin{array}{c} int \wedge \\ (\ell = d; \lceil \neg Y \rceil) \wedge \\ true; \left(\begin{array}{c} int \wedge \\ \lceil \neg Y \rceil \wedge \\ \ell \geq a - r - 1 \end{array} \right) \end{array} \right); \ell < 1.
\end{array}
$$

Hence, the basic case is verified.

Induction step: According to the definition of the function f, there are two cases to consider. We verify for the case **F1** only. The other is similar and is left to the reader.

Let $f(w) = B; (\lceil X \rceil \wedge \ell \geq c) \in \mathcal{L}_s$, where $B \in \Sigma_s^*$. Then, by the definition **F1** of f,

$$
\begin{array}{lcl}
f(w0) & \widehat{=} & B; Hi\_Lo_s; (\lceil \neg X \rceil \wedge \ell \geq c) \\
f(w1) & \widehat{=} & B; Hi\_Lo_s; (\lceil X \rceil \wedge \ell \geq c).
\end{array}
$$

By the inductive hypothesis, we can find a 'word' $D' = A; \ell = d; (\lceil Y \rceil \wedge \ell \geq c' \wedge int); \ell < 1 \in \mathcal{L}_r$ such that

$$
\begin{array}{l}
f(w) \Rightarrow D', \\
g(D') = w, \text{ and} \\
B; (\lceil X \rceil \wedge \ell = a) \Rightarrow \\
A; \left(\begin{array}{c} int \wedge \\ (\ell = d; (\lceil Y \rceil \wedge \ell \geq 1)) \wedge \\ (true; (\lceil Y \rceil \wedge \ell \geq a - r - 1)) \end{array} \right); \ell < 1.
\end{array}
$$

Thus,

$$
\begin{array}{rl}
 & f(w1) = \\
 & B; (\lceil X \rceil \wedge \ell = a); (\lceil \neg X \rceil \wedge \ell = b); (\lceil X \rceil \wedge \ell \geq c) \\
\Rightarrow & A; \left(\begin{array}{c} int \wedge \\ (\ell = d; (\lceil Y \rceil \wedge \ell \geq 1)) \wedge \\ (true; (\lceil Y \rceil \wedge \ell \geq a - r - 1)) \end{array} \right); \\
 & \left(\begin{array}{c} int \wedge \\ \ell < r + 1 \ \wedge (\lceil Y \rceil \vee \\ \lceil Y \rceil^*; \lceil \neg Y \rceil; true) \end{array} \right); \left(\begin{array}{c} int \wedge \\ \lceil \neg Y \rceil \wedge \\ \ell \geq 1 \end{array} \right); \\
 & \ell = r; \left(\begin{array}{c} int \wedge \\ \lceil Y \rceil \wedge \\ \ell \geq c - r - 1 \end{array} \right); \ell < 1 \\
 & \\
\Rightarrow & A; \left(\begin{array}{c} int \wedge \\ (\ell = d; (\lceil Y \rceil \wedge \ell \geq 1)) \wedge \\ (true; (\lceil Y \rceil \wedge \ell \geq a - r - 1)) \end{array} \right); \\
 & \left(\begin{array}{c} int \wedge \\ \lceil \neg Y \rceil \wedge \\ \ell \geq 1 \end{array} \right); \\
 & \left(\begin{array}{c} int \wedge \\ (\neg \Diamond ((\lceil Y \rceil \wedge \ell \geq r); (\lceil \neg Y \rceil \wedge \ell \geq 1))) \end{array} \right); \quad (2) \\
 & \ell < 1. \quad (3)
\end{array}
$$

Here we have used the implication $\ell < 1; \ell = r \Rightarrow \ell < r+1 \wedge (\lceil Y \rceil \vee \lceil Y \rceil^*; \lceil \neg Y \rceil; true)$. On the other hand, by Corollary 3 (and also Corollary 1),

$$f(w1) = B; \lceil X \rceil \wedge \ell = a; (\lceil \neg X \rceil \wedge \ell = b); (\lceil X \rceil \wedge \ell \geq c)$$

$$\begin{array}{lll} \Rightarrow & true; \ell \leq r; \left(\begin{array}{c} int \wedge \\ \lceil Y \rceil \wedge \\ \ell \geq a - r - 1 \end{array} \right); & \\ & \left(\begin{array}{c} int \wedge \\ \lceil \neg Y \rceil \wedge \\ \ell = 1 \end{array} \right); & \\ & \left(\begin{array}{c} int \wedge \\ (\ell = d; \left(\begin{array}{c} int \wedge \\ \lceil Y \rceil \\ \ell \geq c' \end{array} \right)) \wedge \\ (\neg \Diamond ((\lceil Y \rceil \wedge \ell \geq r); (\lceil \neg Y \rceil \wedge \ell \geq 1))) \end{array} \right); & \\ & \ell < 1. & (4) \end{array}$$

By the rules **I3, R3, R4** and because $a \geq 2r+1$, it follows from (2), (3) that

$$\begin{array}{ll} f(w1) & \Rightarrow \\ & A; \ell = d; Hi\_Lo_r; \ell = d; (\lceil Y \rceil \wedge \ell \geq c'); \ell < 1\,. \end{array}$$

By **G1** of Lemma 1 and the inductive hypothesis

$$g(A; \ell = d; Hi\_Lo_r; \ell = d; (\lceil Y \rceil \wedge \ell \geq c'); \ell < 1) \quad = \quad w1.$$

Let

$$\begin{array}{lcl} A' & = & A; \ell = d; Hi\_Lo_r, \\ B' & = & B; Hi\_Lo_s. \end{array}$$

By the same arguments (replacing c with a), it can be verified also that

$$\begin{array}{ll} & B'; (\lceil X \rceil \wedge \ell = a) \\ \Rightarrow & A'; \left(\begin{array}{c} int \wedge \\ (\ell = d; \lceil Y \rceil \wedge \ell \geq 1) \wedge \\ (true; \lceil Y \rceil \wedge \ell \geq a - r - 1) \end{array} \right); \ell < 1. \end{array}$$

Similarly, we can verify the statement for the word $w0$.

Reasoning about clock rates of the sender and the reader

We have verified the correctness of the Biphase Mark Protocols for the case $b + r \leq d \leq a + b - r - 3$, $b \geq r + 1$, $a \geq 2r + 1$, $c \geq c' + 1$. The values of a, b, c, d, r are numbers of the receiver's clock cycles. However, the values a and b should be given according to the sender's clock. Since the sender's clock rate and the receiver's clock rate are different, we cannot have the same values a and b according to the receiver's clock. Thus, the values of a and b should be such that when transformed into the values according to the receiver's clock, the above mentioned conditions are satisfied. Depending on specific values of a, b, c, d, r the ratio of the rates of the sender's and the receiver's clock for which the protocol is correct can be determined easily. Let us consider some examples.

Let $d = 10$, $r = 3$. Then, for $4 \leq b \leq 7$, $c \geq a$, $a + b \geq 16$, the protocol is correct. Let $a = 13$, $b = 5$ according to the sender's clock. Then, if the ratio of the rates of the receiver's clock and the sender's clock is between $4/5$ and $7/5$ then the above inequalities are satisfied and the protocol is still correct.

Let $d = 10$, $r = 2$. Then, for $3 \leq b \leq 8$, $a + b \geq 15$, the protocol is correct. Let $a = 13$, $b = 5$ according to the sender's clock. Then, if the ratio of the rates of the receiver's clock and the sender's clock is between $3/5$ and $8/5$ then the above inequalities are satisfied and the protocol is still correct.

5 Audio Protocol

In this section, we use our approach to specify and verify the correctness of the audio control protocol. The protocol uses the well-known Manchester encoding of bit strings. Each bit is encoded as a cell consisting $4Q$ cycles of the receiver clock. Bit 1 is encoded as an upgoing edge at the middle of the cell, and bit 0 is encoded as a downgoing edge at the middle of the cell. If the same bit is sent twice in a row then an additional edge is required, which is placed exactly in between the corresponding two cells (see Fig. 3).

The receiver scans only for upgoing edges ($l\_Lo\_Hi_r$) using the Philips decoding. It has to decode the signal without seeing the downgoing edges. This is always possible except that a message ending with "10" cannot be distinguished from the same message ending with "1". To solve the problem, we require that all messages either end with "00" or have an odd length. In addition, all messages should begin with "1" for the receiver to detect the beginning of the messages.

Let us denote in this section,

$$\begin{array}{lcl} Hi\_Lo_r & \widehat{=} & (\lceil Y \rceil \wedge int \wedge r \leq \ell); (\lceil \neg Y \rceil \wedge \ell = 1)\,, \\ Lo\_Hi_r & \widehat{=} & (\lceil \neg Y \rceil \wedge int \wedge r \leq \ell); (\lceil Y \rceil \wedge \ell = 1)\,. \end{array}$$

Let $\mathcal{L}_s$ be defined as in the Biphase Mark protocols. Now, we can model the Audio Protocol as follows.

Manchester encoding

$f : \{0,1\}^+ \to \mathcal{L}_s$ in this case is a partial function only. Its domain is defined by a regular expression:

$$1(00 \oplus 01 \oplus 10 \oplus 11)^* \oplus 1(0 \oplus 1)^* 00.$$

Let f' be a homomorphism defined by

$$\begin{array}{lcl} f'(1) & \widehat{=} & Lo\_Hi_s \\ f'(0) & \widehat{=} & Hi\_Lo_s \\ f'(w0) & \widehat{=} & f'(w); f'(0) \\ f'(w1) & \widehat{=} & f'(w); f'(1). \end{array}$$

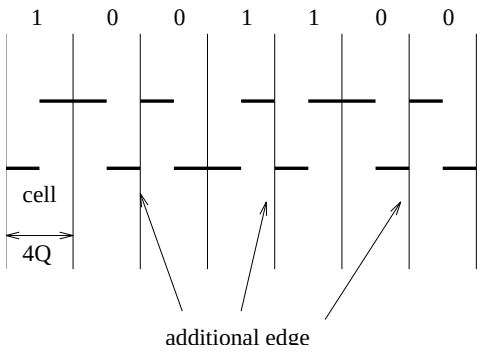

Figure 3. Audio Protocol

Then the Manchester encoding function f is defined as

$$f(w) \;\widehat{=}\; \lceil\neg X\rceil; f'(w); (\lceil\neg X\rceil \wedge \ell \geq c)\,.$$

Philips decoding

Let us denote:

$FourH \mathrel{\widehat{=}} ((\neg\Diamond Lo\_Hi_r); Lo\_Hi_r) \wedge 3Q \leq \ell < 5Q$
$SixH \mathrel{\widehat{=}} ((\neg\Diamond Lo\_Hi_r); Lo\_Hi_r) \wedge 5Q \leq \ell < 7Q$
$EightH \mathrel{\widehat{=}} ((\neg\Diamond Lo\_Hi_r); Lo\_Hi_r) \wedge 7Q \leq \ell$
$NineH \mathrel{\widehat{=}} (\neg\Diamond Lo\_Hi_r) \wedge 9Q \leq \ell\,.$

Let

$$\Sigma_r \;\widehat{=}\; \left\{ \begin{array}{c} FourH, SixH, EightH, NineH, \\ ;, (\neg\Diamond Lo\_Hi_r) \end{array} \right\}$$

and $\mathcal{L}_r$ be represented by the following regular expression over Σ_r:

$$\mathcal{L}_r \;\widehat{=}\; (\neg\Diamond Lo\_Hi_r); Lo\_Hi_r; \\ \mathbf{(}FourH \oplus SixH \oplus EightH\mathbf{)}^*; NineH\,.$$

Then, the decoding function g, $g : \mathcal{L}_r \rightarrow \{0,1\}^+$, is defined below. First, the function $g' : (\neg\Diamond Lo\_Hi_r); Lo\_Hi_r\mathbf{(};\mathbf{(}FourH \oplus SixH \oplus EightH\mathbf{))}^* \rightarrow \{0,1\}^+$ is defined inductively as follows.
Basic case:

$$g'((\neg\Diamond Lo\_Hi_r); Lo\_Hi_r) \;\widehat{=}\; 1\,.$$

Induction step: Assume that $g'(D)$ has been defined and $g'(D) = w0$.

$$g'(D; FourH) \mathrel{\widehat{=}} w00$$
$$g'(D; SixH) \mathrel{\widehat{=}} w01\,.$$

Assume that $g'(D)$ has been defined and $g'(D) = w1$.

$$g'(D; FourH) \mathrel{\widehat{=}} w11$$
$$g'(D; SixH) \mathrel{\widehat{=}} w100$$
$$g'(D; EightH) \mathrel{\widehat{=}} w101\,.$$

Now, the function g'' is defined as:

$$g''(D; NineH) \;\widehat{=}\; g'(D)$$

Let $Just : \{0,1\}^* \rightarrow \{0,1\}^*$ be a homomorphism defined by:

1. if $w = x1$ and $|w|$ is odd then $Just(w) = w$,
2. if $w = x1$ and $|w|$ is even then $Just(w) = w0$,
3. if $w = x0$ and $|w|$ is odd then $Just(w) = w$,
4. if $w = x0$ and $|w|$ is even then
   - $x = y0$ then $Just(w) = w$,
   - $x = y1$ then $Just(w) = w0$.

The Philips decoding function $g : \mathcal{L}_r \rightarrow \{0,1\}^*$ is defined by

$$g(D) \;\widehat{=}\; Just(g''(D))\,.$$

Verification

We have to find the values of a, b, c, r, and prove that for those a, b, c, r

1. For all $D, D' \in \mathcal{L}_r$, $D \neq D'$ implies that $D \wedge D' \Leftrightarrow false$,
2. For all $w \in dom(f)$, there is a formula $D \in \mathcal{L}_r$ such that $f(w) \Rightarrow D$ and $g(D) = w$.

The proof of items 1 and 2 are similar to the proof for the Biphase Mark protocols. We give here a rough sketch of our proof only.

Lemma 2. *Let $D = A; B \in \mathcal{L}_r$, $D' = A; B' \in \mathcal{L}_r$, where $A, B, B' \in \Sigma_r^*$. Then $D \wedge D' \Rightarrow A; (B \wedge B')$.*

The lemma can be proved easily by induction on the length of the word A using the rule **R4**.

The first item follows directly from Lemma 2 with the note that for any $C, C' \in \Sigma_r$, $C \neq C'$ implies that $C \wedge C' \Rightarrow false$.

Based on Corollaries 2,3, we then can prove easily by induction on the length $|w|$ of $w \in \{0,1\}^*$ that if $a = b$, $a \geq 2r + 1$, $4a - r - 1 \geq 7Q$, $4a + r + 1 \leq 9Q$, then

- $w = x1$ implies that there exists $D' \in \mathcal{L}_r$ such that $f(w) \Rightarrow D'$ and $g''(D') = x1$,
- $w = x10$ implies that there exists $D' \in \mathcal{L}_r$ such that $f(w) \Rightarrow D'$ and $g''(D') = x1$,
- $w = x00$ implies that there exists $D' \in \mathcal{L}_r$ such that $f(w) \Rightarrow D'$ and $g''(D') = x0$.

Item 2 then follows immediately from these properties and the definition of the function g.

Reasoning about clock rates of the sender and the reader

In the proof of the correctness of the protocol, we have assumed that the reference time is according to the receiver's clock. However, in practice, a should be implemented according to the sender's clock. Thus if we take a to be $2Q$ according to the sender's clock, then it will be that $a = 2Q + 2\delta$ according to the receiver's clock. Thus, to guarantee that the protocol is correct, it should be the case that

$$\begin{aligned} &2Q + 2\delta \geq 2r + 1, \\ &8Q + 8\delta - r - 1 \geq 7Q, \\ &8Q + 8\delta + r + 1 \leq 9Q. \end{aligned}$$

Thus, if we can ensure that

$$\begin{aligned} &2r + 1 - 2\delta \leq 2Q \\ &|\delta| \leq (Q - r - 1)/8 \end{aligned}$$

then the protocol is correct. For instance, let $r = 3$, $Q = 5$. $a = 2Q$. In this case, $|\delta| \leq 1/8$, which means that the ratio of the rates of the sender's and the receiver's clock should be between $39/40$ and $41/40$.

In practice, Q is very big (≥ 200, see e.g. [1]), and we have much looser constraints on the rates of the clock of the sender and the reader.

6. Conclusion

We have presented our approach to the specification and verification of real-time hybrid systems using Duration Calculus. By introducing the formula int into the calculus, we are able to specified both discrete states and continuous states. Apart from this, we are also able to reason about the different clock rates of different components in the systems, which is necessary in distributed systems. By trying our approach in the Biphase Mark protocol and Audio Control protocol, we have shown that we can specify the protocol in a natural and intuitive way. In our model of communication protocols, we can verify the correctness in a familiar way using just the induction techniques on the length of strings of a formal languages and some obvious rules of the calculus. Since formal proofs are often tedious work, our syntactical approach enables the use of a mechanical prover in the verification of the real-time systems.

References

[1] D. Bosscher, I. Polak, and F. Vaandrager. Verification of an Audio Control Protocol. In *Formal Techniques in Real-Time and Fault-Tolerant Systems*, volume 863 of *Lecture Notes in Computer Science*, pages 170–192. Springer-Verlag, 1994.

[2] Z. Chaochen, M. Hansen, and P. Sestoft. Decidability and Undecidability Results for Duration Calculus. In A. F. P. Enjalbert and K. Wagner, editors, *STACS'93*, volume 665 of *Lecture Notes in Computer Science*, pages 58–68. Springer-Verlag, 1993.

[3] Z. Chaochen, D. V. Hung, and L. Xiaoshan. A Duration Calculus with Infinite Intervals. In H. Reichel, editor, *Fundamentals of Computing Theory*, volume 965 of *Lecture Notes in Computer Science*, pages 16–41. Springer-Verlag, 1995.

[4] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages and Computation*. Reading, MA, Addison-Wesley, 1979.

[5] D. Hung and K. K. Il. Verification via Digitized Models of Real-Time Hybrid Systems. Technical Report 54, UNU/IIST, P.O.Box 3058 Macau, 1996.

[6] B. Mahony and I. Hayes. Using Continuous Real Functions to Model Timed Histories. In *6th Australian Software Engineering Conference, ASWEC91*, 1991.

[7] J. S. Moore. A Formal Model of Asynchronous Communication and Its Use in Mechanically Verifying a Biphase Mark Protocol. *Formal Aspects of Computing*, 6, 1994.

[8] A. Nerode and W. Kohn. Models for hybrid systems: automata, topologies, controllability, observability. In *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 317–356. Springer-Verlag, 1993.

[9] X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. An approach to the description and analysis of hybrid systems. In *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 149–178. Springer-Verlag, 1993.

[10] A. Pnueli. Development of Hybrid Systems. In *Formal Techniques in Real-Time and Fault-Tolerant Systems*, volume 863 of *Lecture Notes in Computer Science*. Springer-Verlag, 77-80 1994.

[11] A. P. R. Zhou Chaochen, C. A. R. Hoare. A Calculus of Durations. *Information Processing Letters*, 40:269–276, 1992.