Timing Analysis of Combinational Circuits in Intuitionistic Propositional Logic

Michael Mendler*

Abstract

Classical logic has so far been the logic of choice in formal hardware verification. This paper proposes the application of intuitionistic logic to the timing analysis of digital circuits. The intuitionistic setting serves two purposes. The model-theoretic properties are exploited to handle the second-order nature of bounded delays in a purely propositional setting without need to introduce explicit time and temporal operators. The proof theoretic properties are exploited to extract quantitative timing information and to reintroduce explicit time in a convenient and systematic way.

We present a natural Kripke-style semantics for intuitionistic propositional logic, as a special case of a Kripke *constraint model* for Propositional Lax Logic [15], in which validity is validity *up to stabilisation*, and implication $\supset$ comes out as "*boundedly gives rise to*." We show that this semantics is equivalently characterised by a notion of realisability with *stabilisation bounds* as realisers. Following this second point of view an intensional semantics for proofs is presented which allows us effectively to compute quantitative stabilisation bounds.

We discuss the application of the theory to the timing analysis of combinational circuits. To test our ideas we have implemented an experimental prototype tool and run several examples.

1 Motivation

A recurring theme in Computer Science, as an engineering discipline, is to find the right level of abstraction in modelling real-world phenomena. There are as many levels of abstraction as there are modelling applications and to adjust the level properly in each case is a nontrivial undertaking. On the one hand it must not be so abstract as to compromise the correctness of the model by ignoring important low-level features. At the same time it must not be so concrete as to clutter up the model with irrelevant low-level details.

A typical case where this issue arises is the formal reasoning about digital circuits in the presence of timing constraints. The ideal abstract specification of a combinational block computing the function f, for instance, is the equation

$$y = f(x), \tag{1}$$

*Department of Computer Science, University of Sheffield, 211 Portobello Street, Sheffield S1 5DP, UK. Email M.Mendler@dcs.shef.ac.uk

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

where x stands for the input and y for the output of the circuit. Usually, x and y would be Boolean vectors and f a Boolean function over x. Though this algebraic specification certainly is adequate for many situations, it is but an approximation that completely elides the timing dimension of the real circuit, which cannot be ignored in practice. When we establish the correctness of a synchronous circuit, for instance, knowledge of the stabilisation behaviour of the combinational blocks is crucial to adjust the clock rate and to meet the setup and hold constraints of the register flipflops. In this case a more refined description of the combinational block is more adequate:

$$\forall t.\ \forall v.\ (\forall s \geq t.\ x(s) = v)\ \Rightarrow\ (\forall s \geq t + 100ns.\ y(s) = f(v)). \tag{2}$$

It says that whenever the input x stabilises at some value v then after a delay of $100ns$ the output y stabilises at the value $f(v)$. This is a safe description of a typical (functional) circuit representing the idealised behaviour (1) with a stabilisation imprecision of $100ns$. In choosing this level of description we are guaranteed to retain control of not only the functional but also the stabilisation behaviour. For instance, when this circuit is now composed with another one realizing $z = g(y)$ subject to an imprecision of $23ns$ (in the above sense) we can formally derive a composite behaviour $z = g(f(x))$ subject to an imprecision of $123ns$:

$$\begin{aligned}
&\forall t.\ \forall v.\ \mathit{stable}(x, t, v)\ \Rightarrow\ \mathit{stable}(y, t + 100ns, f(v))\ \wedge \\
&\quad \forall t.\ \forall v.\ \mathit{stable}(y, t, v)\ \Rightarrow\ \mathit{stable}(z, t + 23ns, g(v)) \\
&\Rightarrow\ \forall t.\ \forall v.\ \mathit{stable}(x, t, v)\ \Rightarrow\ \mathit{stable}(z, t + 123ns, g(f(v))),
\end{aligned}$$

where $\mathit{stable}(x, t, v)$ abbreviates $\forall s \geq t.\ x(s) = v$. From an engineering point of view, however, such a detailed low-level analysis, though it can be done, is not necessarily a good idea. It mixes up what usually are considered separate concerns: the abstract reasoning in terms of time-free algebraic equations, *i.e.* the inference of $z = g(f(x))$ from $y = f(x)$ and $z = g(y)$, and the analysis of the timing constraint, *i.e.* adding the delays $100ns + 23ns = 123ns$.

The purpose of this paper is to present a meet-in-the-middle framework for the formal verification of combinational circuits that lies between the two extremes (1) and (2). It is not as concrete as (2) which messes up the functional verification with timing, and it is not as abstract as (1) which looses track of the stabilisation behaviour altogether. We use the syntactically modest formalism of intuitionistic propositional logic to deal with the functional behaviour and exploit the richness of its underlying Kripke models and realisability semantics to account for timing.

2 Introduction

Let us agree, for the purpose of this paper, that we are interested only in Boolean combinational circuits, *i.e.* circuits built from components like INV, AND, OR, NAND gates. An extension to combinational circuits over arbitrary finite data domains is straightforward. A *signal* a, then, is a timed Boolean function, *i.e.* $a \in \mathbb{N} \to \mathbb{B}$, where time is represented

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

by the natural numbers. For convenience we will fix a countably infinite number of signals $\mathbb{S} = \{a, b, c, c_1, c_2, \cdots\}$ throughout, and conceive a *circuit* as a relation $C \subseteq \mathbb{S} \to \mathbb{N} \to \mathbb{B}$ which constrains the behaviour on signals. The elements $V \in C$ will be called *waveforms*.

For every $a \in \mathbb{S}$ let $a = 1$ and $a = 0$ be the atomic sentences stating that signal a is stable at 1 and 0, respectively. Thus, with $C \models \varphi$ expressing that circuit C satisfies formula φ, we stipulate

$$\begin{aligned} C \models a = 1 \quad &=_{df} \quad \forall V \in C.\, \mathit{stable}(V(a), 0, 1) \\ C \models a = 0 \quad &=_{df} \quad \forall V \in C.\, \mathit{stable}(V(a), 0, 0), \end{aligned}$$

where $\mathit{stable}(x, t, v)$, as before, abbreviates $\forall s \geq t.\, x(s) = v$. Our goal is to build up from these atomic sentences more complex propositional formulas that capture the stabilisation behaviour of circuits, without resorting to explicit time and explicit temporal operators. But how do we get nontrivial transition behaviour from purely propositional connectives? After all, the atomic sentences represent constant behaviour, whence a simple combination of such sentences by ordinary logical disjunction, conjunction, and implication does not yield anything exciting. The trick is to interpret an implication like $a = 1 \supset b = 1$ not as in classical logic "if a is stable 1 then b is stable 1" but as a *boundedly gives rise to* statement: "there exists a stabilisation bound δ so that whenever a becomes stable 1, b will become stable 1 with a maximal delay δ." Formally,

$$\begin{aligned} C \models a = 1 \supset b = 1 \quad =_{df} \quad & \exists \delta \in \mathbb{N}.\, \forall V \in C.\, \forall t \in \mathbb{N}. \\ & \mathit{stable}(V(a), t, 1) \Rightarrow \mathit{stable}(V(b), t + \delta, 1). \end{aligned} \tag{3}$$

Notice that this is a much stronger condition than

$$\forall V \in C.\, \forall t \in \mathbb{N}.\, \mathit{stable}(V(a), t, 1) \Rightarrow \exists \delta \in \mathbb{N}.\, \mathit{stable}(V(b), t + \delta, 1), \tag{4}$$

which might seem a more obvious generalisation of (2) to abstract from a particular stabilisation constraint, such as $100ns$. The difference between (3) and (4) is the swapping of the $\exists\delta$ and $\forall V$ quantifiers, which has quite a drastic effect: The quantification $\forall V.\forall t.\exists\delta$ in (4) means that C may exhibit unbounded response time, which is not what we want. In fact, (4) is logically equivalent to $\forall V.\, (\exists t.\, \mathit{stable}(V(a), t, 1)) \Rightarrow (\exists t.\, \mathit{stable}(V(b), t, 1))$, which simply states that "if a eventually stabilises to 1 then b eventually stabilises to 1." This would again make $\supset$ a classical implication which does not make for an interesting semantics. Contrast this with the quantification $\exists\delta.\forall V.\forall t$ of (3). It specifies an unknown, but fixed stabilisation delay for all waveforms of the circuit, which is the right generalisation of (2). It takes care of the fact that a propagation delay is a property of a circuit C rather than a property of an individual waveform, *i.e.* of an execution of a circuit.

But how can we squeeze an intrinsically second-order condition like (3) into a single propositional connective like $\supset$ and yet hope to get a useful logic? Surprisingly, it turns out that the particular semantic condition (3) indeed behaves like an implication, *viz.* intuitionistic one. In order to indicate why, let us first observe that (3) is closed under inclusion and time shift. Let us define a partial ordering $\sqsubseteq$ on circuits $C, D \subseteq \mathbb{S} \to \mathbb{N} \to \mathbb{B}$ such that $C \sqsubseteq D$ if every waveform in D is the time shift of some waveform in C. Formally,

$$C \sqsubseteq D \quad \textit{iff} \quad \forall W \in D.\, \exists t \in \mathbb{N}.\, \exists V \in C.\, W = V^t,$$

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

where V^t is the *time shift* of V, *i.e.* $V^t(a)(s) = V(a)(s+t)$. Thus, in passing from C to D with $C \sqsubseteq D$ we are taking a subset of C and perform an arbitrary time shift on the selected waveforms. Then we find that

$$C \models a = 1 \supset b = 1 \;\&\; C \sqsubseteq D \quad \Rightarrow \quad D \models a = 1 \supset b = 1.$$

This closure property reveals the intuitionistic nature of our interpretation. In fact, it is possible to show (*cf.* Section 5) that $C \models a = 1 \supset b = 1$ is equivalent to

$$\forall D.\, C \sqsubseteq D \;\&\; D \models a = 1 \quad \Rightarrow \quad \exists \delta.\, D^\delta \models b = 1,$$

where D^δ is obtained from D by time shifting all $W \in D$ by an amount of δ, $D^\delta = \{\, W^\delta \mid W \in D \,\}$. This formulation has the structure essentially of an intuitionistic implication (see *e.g.* [52]) interpreted on the Kripke frame induced by the set of circuits $C \subseteq \mathbb{S} \to \mathbb{N} \to \mathbb{B}$ ordered by the $\sqsubseteq$ relation. Using this Kripke model for LJ obtains a full-blooded intuitionistic semantics based on $\models$ in which non-trivial stabilisation behaviour can be expressed and verified.

To finish off this section let us point out that there is another arrangement of the quantifiers $\exists\delta.\forall V.\forall t$ in (3) that yields a non-trivial intuitionistic interpretation of implication. It is the specification

$$\begin{aligned} C \models_t a = 1 \supset b = 1 \quad =_{df} \quad & \forall t \in \mathbb{N}.\, \exists \delta \in \mathbb{N}.\, \forall V \in C. \\ & \mathit{stable}(V(a), t, 1) \Rightarrow \mathit{stable}(V(b), t+\delta, 1), \end{aligned} \tag{5}$$

which is a *time variant* weakening of (3). Here the delay with which b responds to a may depend on the time at which a stabilises, but still is uniform for all waveforms. As an aside, the reader may check that all other re-orderings of the quantifiers in (3) do not result in anything new. This time variant version $\models_t$ can be phrased as a Kripke semantics by taking a more rigid ordering on circuits, *viz.* the ordering $C \sqsubseteq_t D$ *iff* $\exists t \in \mathbb{N}.\, \forall W \in D.\, \exists V \in C.\, W = V^t$. Thus, for the time variant case a circuit D is required to inherit all properties from a circuit C, *i.e.* $C \sqsubseteq_t D$, if there exists a uniform time delay t such that all waveforms in D are time shifted by t from C. This is a strengthening of $\sqsubseteq$, which is the reason why validity $\models_t$ is weaker. Again, note that the difference is due to a quantifier swap: $\sqsubseteq$ has $\forall W.\exists t.\exists V$ and $\sqsubseteq_t$ has $\exists t.\forall W.\exists V$ in its definition.

In this paper we will stick to the *time invariant* formulation (3), which appears to be the more adequate setup. For all practical purposes the response time of digital circuits may be assumed to be time invariant. It is also a central assumption in standard digital design that the delay does not depend on the time when a circuit is used but only on the data that are computed. We point out, however, that the results presented in this paper can be generalised to $\models_t$ too. We will indicate in passing the modifications that need to be done.

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

3 Intuitionistic Propositional Logic LJ

We will be concerned with a non-standard semantics for standard intuitionistic propositional logic, the formulas of which are generated by the grammar

$$\varphi \quad ::= \quad a = 1 \mid a = 0 \mid \mathit{false} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \supset \varphi$$

where a ranges over the set of signal constants $\mathbb{S}$. We will use $\equiv$ to abbreviate bi-implication, and $\neg\varphi$ to abbreviate $\varphi \supset \mathit{false}$. The formulas $a = 1$, $a = 0$ are the atomic propositional sentences of the logic. Our formal calculus is a slight variant [12] of Gentzen's sequent calculus for the logic LJ [19]. The rules are shown in Fig. 1. A multi-succedent version of this calculus also appears in [11].

$$\frac{}{\Gamma, \varphi \vdash \varphi}\, id \qquad \frac{\Gamma, \Gamma', \varphi \vdash \theta}{\Gamma, \varphi, \Gamma' \vdash \theta}\, get$$

$$\frac{\Gamma \vdash \varphi \qquad \Gamma \vdash \psi}{\Gamma \vdash \varphi \wedge \psi} \wedge R \qquad \frac{\Gamma, \varphi, \psi \vdash \theta}{\Gamma, \varphi \wedge \psi \vdash \theta} \wedge L$$

$$\frac{\Gamma, \varphi \vdash \theta \qquad \Gamma, \psi \vdash \theta}{\Gamma, \varphi \vee \psi \vdash \theta} \vee L \qquad \frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \vee \psi} \vee R_1 \qquad \frac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \vee \psi} \vee R_2$$

$$\frac{}{\Gamma, \mathit{false} \vdash \theta}\, \mathit{false} L$$

$$\frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \supset \psi} \supset R \qquad \frac{\Gamma, \varphi \supset \psi \vdash \varphi \qquad \Gamma, \psi \vdash \theta}{\Gamma, \varphi \supset \psi \vdash \theta} \supset L$$

Figure 1: Gentzen Rules for LJ.

The system of Fig. 1 is a cut and contraction free presentation of LJ that directly lends itself to be mechanised in a Prolog fashion. The only difficulty lies in the fact that, in a blind application of the rules in a backwards search, the replication of the principal formula $\varphi \supset \psi$ in the left branch of rule $\supset L$ may lead to looping. This problem can be overcome by some form of loop detection, or more elegantly by using an equivalent system [12] in which rule $\supset L$ is split up into (replaced by) four rules without replication, as seen in Fig. 2. In a practical implementation of a theorem prover for LJ even more tricks will be built in (see *e.g.* [17].) However, to avoid irrelevant implementation details we will stick to the simple system of Fig. 1 in this paper.
For our purposes it will be important to have some notation for proofs and derivations. Derivation terms are linear notations for derivation trees in LJ with multiple conclusions. They are generated by the following simple grammar:

$$\begin{aligned} \mathit{Deduction} &::= \mathit{Rule} \cdot \mathit{Deduction} \mid (\mathit{Deduction}, \mathit{Deduction}) \\ \mathit{Rule} &::= id \mid get(i) \mid \wedge R \mid \wedge L \mid \vee L \mid \vee R_1 \mid \vee R_2 \mid \mathit{false} L \mid \supset R \mid \supset L, \end{aligned}$$

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

$$\frac{\Gamma, \alpha, \psi \vdash \theta}{\Gamma, \alpha, \alpha \supset \psi \vdash \theta} \supset L_1 \ (\alpha \text{ atomic sentence})$$

$$\frac{\Gamma, \varphi_1 \supset (\varphi_2 \supset \psi) \vdash \theta}{\Gamma, (\varphi_1 \wedge \varphi_2) \supset \psi \vdash \theta} \supset L_2$$

$$\frac{\Gamma, \varphi_1 \supset \psi, \varphi_2 \supset \psi \vdash \theta}{\Gamma, (\varphi_1 \vee \varphi_2) \supset \psi \vdash \theta} \supset L_3$$

$$\frac{\Gamma, \varphi_2 \supset \psi \vdash \varphi_1 \supset \varphi_2 \qquad \Gamma, \psi \vdash \theta}{\Gamma, (\varphi_1 \supset \varphi_2) \supset \psi \vdash \theta} \supset L_4$$

Figure 2: Replication-Free Replacement for $\supset L$.

where i is a natural number recording the principal formula picked out by an application of rule *get*. If D is a derivation of a sequent $\Gamma \vdash \varphi$ we will denote this by $D : \Gamma \vdash \varphi$. When Γ is empty we call D a *proof* of φ.

4 Circuits as Timed Kripke Models

By considering circuits as a particular class of Kripke models we will interpret formulas of LJ as statements about circuits. These Kripke models are induced by the ordering $\sqsubseteq$ on circuits and the operation of time shift. Let φ be a formula and C a circuit. We say that C *validates* φ *up to stabilisation*, written $C \models \varphi$, if

$\varphi = \mathit{false}$	and $C = \emptyset$, or
$\varphi = a = 1$	and for all $V \in C$ and $t \in \mathbb{N}$, $V(a)(t) = 1$, or
$\varphi = a = 0$	and for all $V \in C$ and $t \in \mathbb{N}$, $V(a)(t) = 0$, or
$\varphi = \psi_1 \wedge \psi_2$	and both $C \models \psi_1$ and $C \models \psi_2$, or
$\varphi = \psi_1 \vee \psi_2$	and $C \models \psi_1$ or $C \models \psi_2$, or
$\varphi = \psi_1 \supset \psi_2$	and for all D such that $C \sqsubseteq D$ and $D \models \psi_1$, there exists $\delta \in \mathbb{N}$ such that $D^\delta \models \psi_2$.

A formula φ is *valid up to stabilisation*, written $\models \varphi$, if $C \models \varphi$ for all circuits C. This nails down the Kripke semantics sketched in the Introduction. In this semantics $\varphi \supset \psi$ is not the classical "if φ then ψ" but "φ gives rise to ψ in bounded time," where the stress is on *bounded*. We will elaborate on the connection between this Kripke semantics and bounded stabilisation in the next section.

The semantic clauses above defining $\models$ are almost exactly the standard clauses (see *e.g.* [52]) for intuitionistic validity in Kripke models. Compared to the standard setting there are two important things to note, though.

(i) We are considering a particular class of Kripke models generated from sets of waveforms under the $\sqsubseteq$ ordering. This means that our interpretation of LJ is rather

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

specialised. It is a dedicated stabilisation theory of combinational circuits with distinguished propositional constants $a = 0$ and $a = 1$ (for all $a \in \mathbb{S}$) that have the fixed semantic interpretation "signal a is stable 1" and "signal a is stable 0." Our theory, for instance, satisfies the theorem $\neg(a = 0 \wedge a = 1)$ which is not a theorem of LJ.

(*ii*) The clause for validity of implication $\supset$ is a modification of the standard one, which would be $C \models \varphi \supset \psi$ *iff* $\forall D.\ C \sqsubseteq D \ \ \& \ \ D \models \varphi \Rightarrow D \models \psi$. In our case the implication $D \models \varphi \Rightarrow D \models \psi$ is extended to allow for an additional time shift: $D \models \varphi \Rightarrow \exists \delta.\ D^{\delta} \models \psi$. This modification is essential and gives us the "boundedly gives rise to" interpretation of implication.

Because of the nonstandard treatment of implication, as pointed out in (*ii*), it is not immediately obvious that LJ actually is sound for our semantics. Of course it better be and indeed we have

Theorem 4.1 (Soundness) *If* $\vdash \varphi$ *then* $\models \varphi$.

Proof: The theorem can be proven by induction on the derivation of $\vdash \varphi$. It directly follows also from Theorems 5.1 and 6.4. ∎

The first point (*i*) noted above, *viz.* that our semantics induces a special theory of LJ, raises the question of completeness. For the calculus of LJ to become complete for our semantics additional axioms and possibly also rules must be added. In this paper, however, all we need is soundness. The question of whether there exists a complete axiomatisation of our LJ theory is left open. We will come back to this point briefly at the end of this paper.

Here are a few simple observations about the semantics: First, notice that if $C = \emptyset$, then $C \models \varphi$ for all φ. Since the empty circuit validates all formulas it is semantically irrelevant. Still it is technically convenient to include it as a degenerate, but harmless, case. Another degenerate but more interesting case is when the circuit only consists of a single constant waveform, *i.e.* $C = \{V\}$ where $\forall a \in \mathbb{S}.\ \exists v \in \mathbb{B}.\ \forall t \in \mathbb{N}.\ V(a)(t) = v$. Then validity up to stabilisation coincides with ordinary classical validity: $C \models \varphi$ *iff* φ is classically valid for V, where an atomic sentence $a = v$, is read as "signal a is constant v." This special case corresponds to the usual static two-valued model of circuits. There is yet another way in which the classical two-valued model can be embedded into this semantics: As one verifies readily, $C \models \neg\neg\varphi$ *iff* φ is classically valid on all $V \in C$, where $a = v$ is read as "signal a will stabilise to v." This means that double negated formulas are classical statements about the stationary state of a circuit. To be more precise, these are statements in a classical three-valued setting in which a signal a can be stable 1, stable 0, or oscillate. The latter value is represented by the formula $\neg a = 1 \wedge \neg a = 0$. If C is a circuit in which all signals eventually stabilise, then we get back, under double negation, the classical two-valued model of the final stable state. Note that the double negated $C \models \neg\neg(a = 1 \supset b = 1)$ is exactly the classical (and thus trivial) semantic condition (4) discussed in Section 2.

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

In view of (*ii*) our claim that $\models$ is a Kripke semantics for LJ, of course, deserves some justification. It is not immediately obvious since the existential quantification $\exists\delta$ and the time shift D^δ is not part of the usual Kripke interpretation of implication. However, it is possible to show that validity up to stabilisation can be rephrased in terms of a *two-frame* intuitionistic Kripke semantics. More precisely, every circuit C induces a canonical Kripke *constraint model* [15] C^* such that for all formulas φ, $C \models \varphi$ *iff* $C^* \models^* \varphi^*$, where φ^* is obtained from φ by replacing each occurrence of a sub-formula $\psi \supset \theta$ by $\psi \supset \bigcirc\theta$. Here, $\bigcirc$ is the constraint modality and $\models^*$ validity of Propositional Lax Logic [15].

5 Stabilisation Bounds as Realisers

Let us unfold the meaning of the formula $a = 1 \supset b = 1$ for a circuit C. By definition $C \models a = 1 \supset b = 1$ is equivalent to

$$\forall D.\ C \sqsubseteq D\ \ \&\ \ (\forall V \in D.\ \mathit{stable}(V(a), 0, 1)) \Rightarrow \exists\delta.\ \forall V \in D.\ \mathit{stable}(V(b), \delta, 1). \quad (6)$$

In particular, consider the time invariant subset $D^* = \{\, V^t \mid V \in C \ \&\ t \in \mathbb{N} \ \&\ \mathit{stable}(V(a), t, 1) \,\}$, which is defined so that $\forall V \in D^*.\ \mathit{stable}(V(a), 0, 1)$ is trivially true. If we instantiate D in (6) by D^*, then (6) reduces to $\exists\delta.\ \forall V \in D^*.\ \mathit{stable}(V(b), \delta, 1)$, which implies

$$\exists\delta.\ \forall V \in C. \forall t \in \mathbb{N}.\ \mathit{stable}(V(a), t, 1) \Rightarrow \mathit{stable}(V(b), t + \delta, 1). \quad (7)$$

This is precisely the second-order semantical condition (3) that we are aiming at. The converse can be shown, too, *viz.* that (7) implies (6). Technically, we have proven the equivalence between two different semantics of the intuitionistic implication $a = 1 \supset b = 1$: The Kripke semantics $C \models a = 1 \supset b = 1$ and a *realisability* interpretation (7). We may think of the stabilisation bound δ in (7) as a *realiser* for the formula $a = 1 \supset b = 1$ and the implication $\forall t \in \mathbb{N}.\ \mathit{stable}(V(a), t, 1) \Rightarrow \mathit{stable}(V(b), t + \delta, 1)$ as the *refinement* of $a = 1 \supset b = 1$ by this realiser δ. This refinement, also called the *realisability predicate*, expresses a timing property of a waveform V. According to this reading the semantic condition (7) stipulates that there exist a realiser δ such that the refinement of $a = 1 \supset b = 1$ by this realiser δ is valid for all waveforms $V \in C$. In other words: $a = 1 \supset b = 1$ is uniformly realisable on C. This realisability interpretation can be lifted systematically to all formulas and proven to be equivalent to $\models$. This we will show next.

To every formula φ we assign a set $[\![\varphi]\!]$ representing the set of realisers for φ:

$$\begin{array}{rcl} [\![a = 1]\!] & = & \{*\} \ = \ [\![a = 0]\!] \\ [\![\mathit{false}]\!] & = & \{*\} \\ [\![\varphi \wedge \psi]\!] & = & [\![\varphi]\!] \times [\![\psi]\!] \\ [\![\varphi \vee \psi]\!] & = & [\![\varphi]\!] + [\![\psi]\!] \\ [\![\varphi \supset \psi]\!] & = & [\![\varphi]\!] \to \mathbb{N} \times [\![\psi]\!]. \end{array}$$

The set $\{*\}$ is the distinguished singleton set with sole element $*$, the operations $\times, +, \to$ are the standard set-theoretic constructions of Cartesian product, disjoint sum, function

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

space, respectively. As usual the elements of the disjoint sum $[\![\varphi]\!] + [\![\psi]\!]$ are pairs $(1, c_1)$ where $c_1 \in [\![\varphi]\!]$ or $(2, c_2)$ where $c_2 \in [\![\psi]\!]$. Note that $[\![\varphi]\!]$ always is non-empty, so that every formula has at least one bound. Let us call an element $c \in [\![\varphi]\!]$ a *stabilisation bound* or simply a *bound* for φ. The bounds can be used to refine formulas into dynamic statements in the time domain. We say that φ is *valid* with bound $c \in [\![\varphi]\!]$ for a waveform $V \in \mathbb{S} \to \mathbb{N} \to \mathbb{B}$, written $c, V \Vdash \varphi$, according to the following inductive clauses:

$$
\begin{array}{rcll}
*, V & \Vdash & a = 1 & \textit{iff} \quad \text{for all } t \in \mathbb{N},\ V(a)(t) = 1 \\
*, V & \Vdash & a = 0 & \textit{iff} \quad \text{for all } t \in \mathbb{N},\ V(a)(t) = 0 \\
(c_1, c_2), V & \Vdash & \varphi \wedge \psi & \textit{iff} \quad c_1, V \Vdash \varphi \text{ and } c_2, V \Vdash \psi \\
(1, c_1), V & \Vdash & \varphi \vee \psi & \textit{iff} \quad c_1, V \Vdash \varphi \\
(2, c_2), V & \Vdash & \varphi \vee \psi & \textit{iff} \quad c_2, V \Vdash \psi \\
f, V & \Vdash & \varphi \supset \psi & \textit{iff} \quad \text{for all } t \in \mathbb{N} \text{ and } c \in [\![\varphi]\!], \\
 & & & \qquad \text{if } c, V^t \Vdash \varphi \text{ then } \pi_2(f\ c), V^{\pi_1(f\ c)+t} \Vdash \psi.
\end{array}
$$

A formula φ is said to be *valid for a circuit* C with stabilisation bound $c \in [\![\varphi]\!]$, if $c, V \Vdash \varphi$ for all waveforms $V \in C$. We call c a *uniform bound* for φ if φ is valid with bound c for all circuits C. Note that there are formulas φ, such as *false*, that do not have any uniform bound, even though $[\![\varphi]\!]$ is nonempty. It can be checked easily by induction on φ, that if $c, V \Vdash \varphi$ then $c, V^t \Vdash \varphi$ for all $t \in \mathbb{N}$.

The following theorem states the equivalence between the Kripke semantics introduced in the previous section and the realisability interpretation. It implies that a formula φ is valid up to stabilisation *iff* it has a uniform bound.

Theorem 5.1 (Equivalence of Kripke and Realisability Semantics) *Let C be a circuit and φ a formula. Then C validates φ up to stabilisation iff there exists a stabilisation bound $c \in [\![\varphi]\!]$ such that φ is valid for C with bound c.*

Proof: The proof proceeds by induction on φ, showing that $C \models \varphi$ *iff* $\exists c \in [\![\varphi]\!].\ \forall V \in C.\ c, V \Vdash \varphi$.

• $C \models \mathit{false}$ is true *iff* $C = \emptyset$. Since $[\![\mathit{false}]\!] = \{*\}$ and $*, V \not\Vdash \mathit{false}$ this is equivalent to $\exists c \in [\![\mathit{false}]\!].\ \forall V \in C.\ c, V \Vdash \mathit{false}$.

• $C \models a = i$ is the condition $\forall V \in C.\ \forall t \in \mathbb{N}.\ V(a)(t) = i$. But this is equivalent to $\forall V \in C.\ *, V \Vdash a = i$, which is equivalent to $\exists c \in [\![a = i]\!].\ \forall V \in C.\ *, V \Vdash a = i$ as desired.

• Suppose $C \models \varphi \wedge \psi$, *i.e.* $C \models \varphi$ and $C \models \psi$. By induction hypothesis this implies the existence of $c_1 \in [\![\varphi]\!]$ and $c_2 \in [\![\psi]\!]$ such that for all $V \in C$, $c_1, V \Vdash \varphi$ and $c_2, V \Vdash \psi$. Thus, by definition $(c_1, c_2), V \Vdash \varphi \wedge \psi$. Since V is arbitrary and $(c_1, c_2) \in [\![\varphi \wedge \psi]\!] = [\![\varphi]\!] \times [\![\psi]\!]$ we are done.

Vice versa, suppose $c \in [\![\varphi]\!] \times [\![\psi]\!]$ such that for all $V \in C$, $c, V \Vdash \varphi \wedge \psi$. Then, $c = (c_1, c_2)$ for some $c_1 \in [\![\varphi]\!]$ and $c_2 \in [\![\psi]\!]$ and further, $c_1, V \Vdash \varphi$ and $c_2, V \Vdash \psi$. Thus, by induction hypothesis, $C \models \varphi$ and $C \models \psi$, whence $C \models \varphi \wedge \psi$.

• Suppose $C \models \varphi \vee \psi$, *i.e.* $C \models \varphi$ or $C \models \psi$. Let us assume the first is true. Then, by induction hypothesis, there is $c \in [\![\varphi]\!]$ such that for all $V \in C$, $c, V \Vdash \varphi$. This implies

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

$(1,c), V \Vdash \varphi \vee \psi$ for all $V \in C$, which is what we need. The case that $C \models \psi$ is completely symmetric, replace $(1,c)$ by $(2,c)$ and φ by ψ.

Vice versa, suppose $c \in [\![\varphi \vee \psi]\!] = [\![\varphi]\!] + [\![\psi]\!]$ and $c, V \Vdash \varphi \vee \psi$ for all $V \in C$. If $c = (1, c_1)$ this means $c_1, V \Vdash \varphi$ for all $V \in C$, so that by induction hypothesis, $C \models \varphi$. Symmetrically, if $c = (2, c_2)$ we conclude $C \models \psi$.

• Suppose there exists a function $f \in [\![\varphi \supset \psi]\!] = [\![\varphi]\!] \to \mathbb{N} \times [\![\psi]\!]$ such that for all $V \in C$, $f, V \Vdash \varphi \supset \psi$. By definition of $\Vdash$ this means that for all $V \in C$, $t \in \mathbb{N}$, and $x \in [\![\varphi]\!]$ if $x, V^t \Vdash \varphi$ then $\pi_2\,(f\,x), V^{t+\pi_1\,(f\,x)} \Vdash \psi$. We wish to show that $C \models \varphi \supset \psi$. To this end let D be an arbitrary circuit such that $C \sqsubseteq D$ and $D \models \varphi$. By induction hypothesis, then, we have $c \in [\![\varphi]\!]$ so that for all $W \in D$, $c, W \Vdash \varphi$. Let $W \in D$ be given. Since $C \sqsubseteq D$ there exists $V \in C$ and $t \in \mathbb{N}$ such that $W = V^t$. Then $c, V^t \Vdash \varphi$. Using the property of function f we conclude that $\pi_2\,(f\,c), V^{t+\pi_1\,(f\,c)} \Vdash \psi$, which in turn means $\pi_2\,(f\,c), W^{\pi_1\,(f\,c)} \Vdash \psi$. As $W \in D$ was arbitrary, this is the same as saying that for all $W \in D^{\pi_1\,(f\,c)}$, $\pi_2\,(f\,c), W \Vdash \psi$, which by induction hypothesis means $D^{\pi_1\,(f\,c)} \models \psi$. Hence, we have found, for all D with $C \sqsubseteq D$ and $D \models \varphi$, a $\delta \in \mathbb{N}$ such that $D^\delta \models \psi$. This, by definition, proves $C \models \varphi \supset \psi$.

Vice versa, assume that $C \models \varphi \supset \psi$, *i.e.* for all circuits D with $C \sqsubseteq D$ and $D \models \varphi$ there exists $\delta \in \mathbb{N}$, $D^\delta \models \psi$. Now let $x \in [\![\varphi]\!]$ be arbitrary and $D(x)$ the set of waveforms

$$D(x) \quad =_{df} \quad \{\, V^t \mid V \in C \;\&\; t \in \mathbb{N} \;\&\; x, V^t \Vdash \varphi \,\},$$

which obviously satisfies $C \sqsubseteq D(x)$. $D(x)$ may be empty, but in any case we have that for all $W \in D(x)$, $x, W \Vdash \varphi$, which by induction hypothesis means $D(x) \models \varphi$. Since $C \sqsubseteq D(x)$ we can make use of our assumption to obtain a $\delta \in \mathbb{N}$, $D(x)^\delta \models \psi$. By induction hypothesis again, this implies

$$\exists c \in [\![\psi]\!].\ \forall W \in D(x)^\delta.\ c, W \Vdash \psi.$$

This is equivalent to the statement that there exists a $c \in [\![\psi]\!]$ such that for all $V \in C$ and $t \in \mathbb{N}$, $x, V^t \Vdash \varphi$ implies $c, V^{t+\delta} \Vdash \psi$. Since we got the existence of $c \in [\![\psi]\!]$ and $\delta \in \mathbb{N}$ for any $x \in [\![\varphi]\!]$, by the Axiom of Choice, there exists a function $f \in [\![\varphi]\!] \to \mathbb{N} \times [\![\psi]\!]$ such that for all $V \in C$ we have

$$\forall x \in [\![\varphi]\!].\ \forall t \in \mathbb{N}.\ x, V^t \Vdash \varphi \;\Rightarrow\; \pi_2\,(f\,x), V^{t+\pi_1\,(f\,x)} \Vdash \psi.$$

But this is nothing but the statement that $f, V \Vdash \varphi \supset \psi$. ∎

We noted that double negated formulas are classical statements about the stationary state of a circuit. Such statements, too, may have uniform bounds, but they do not contain any information. More precisely, it can be shown that if $c \in [\![\neg\neg\varphi]\!]$ and $c, V \Vdash \neg\neg\varphi$ then for all $d \in [\![\neg\neg\varphi]\!]$, $d, V \Vdash \neg\neg\varphi$. Thus, $\neg\neg\varphi$ either has a uniform bound or it does not; and if it does all bounds are uniform bounds. Moreover, they cannot be distinguished from each other by the relation $\Vdash$. Hence they might just as well all be identified. This information collapse reflects the classical nature of $\neg\neg\varphi$ from the realisability point of view.

There is a useful formal characterisation of validity up to stabilisation which can be derived from the realisability semantics. We define a syntactic translation that turns a formula φ

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

of intuitionistic logic into an equivalent formula $\varphi^\sharp(t,x)$ of classical (typed) higher-order logic, with free variables t, x, V of type $\mathbb{N}, [\![\varphi]\!], \mathbb{S} \to \mathbb{N} \to \mathbb{B}$, respectively:

$$\begin{aligned}
(a=1)^\sharp(t,x) &= \mathit{stable}(V(a),t,1)\\
(a=0)^\sharp(t,x) &= \mathit{stable}(V(a),t,0)\\
\mathit{false}^\sharp(t,x) &= \mathit{false}\\
(\varphi\wedge\psi)^\sharp(t,x) &= \varphi^\sharp(t,\pi_1\,x)\wedge\psi^\sharp(t,\pi_2\,x)\\
(\varphi\vee\psi)^\sharp(t,x) &= \forall y.\ x=(1,y)\Rightarrow\varphi^\sharp(t,y)\wedge\forall y.\ x=(2,y)\Rightarrow\psi^\sharp(t,y)\\
(\varphi\supset\psi)^\sharp(t,x) &= \forall s\geq t.\ \forall y.\ \varphi^\sharp(s,y)\Rightarrow\psi^\sharp(s+\pi_1(x\,y),\pi_2(x\,y)).
\end{aligned}$$

Notice, in general, $\varphi^\sharp(t,x)$ is higher-order since in the last clause for implication the quantifier $\forall y$ ranges over elements of type $[\![\varphi]\!]$ which can be function spaces of arbitrary order.

Proposition 5.2 *φ is valid for C with bound c iff the formula $\varphi^\sharp(0,c)$ is (classically) valid for all waveforms $V \in C$.*

Proof: One shows by induction on φ that for all $t \in \mathbb{N}$ and $x \in [\![\varphi]\!]$, $\varphi^\sharp(t,x)$ *iff* $x, V^t \Vdash \varphi$. ∎

By Thm. 5.1 and Prop. 5.2, a formula φ is valid up to stabilisation *iff* there exists a $c \in [\![\varphi]\!]$ such that $\varphi^\sharp(0,c)$ is a classical tautology. Thus, the translation $\varphi \mapsto \varphi^\sharp$ gives us a way of embedding our timing semantics of intuitionistic logic into classical predicate logic. All in all we now have available three equivalent ways of presenting our timing semantics:

Kripke Style: $C \models \varphi$

Realisability: $\exists c \in [\![\varphi]\!].\ \forall V \in C.\ c, V \Vdash \varphi$

Predicate Logic: $\exists c \in [\![\varphi]\!].\ \forall V \in C.\ \varphi^\sharp(0,c)$

Each of these presentations has its specific technical advantages and disadvantages. In the following we adoptthe realisability point of view since it corresponds most closely to the standard fashion of separating the intensional aspect of timing from from the extensional aspect of function. This separation of concerns is formalised directly in the distinction between formulas and realisers: the formula φ represents the functional information and the realizers $[\![\varphi]\!]$ the timing information in form of stabilisation bounds.

As pointed out in Sec. 4 our treatment of circuits as timed Kripke models is a modification of the standard Kripke semantics. It deviates from the standard method in its treatment of implication which involves an implicit modal operator. The realisability semantics introduced in this section, too, is nonstandard. It adds another variant to the many notions of realisability discussed for intuitionistic logic [51]. The notion that comes closest to ours is the set-theoretic realisability introduced by Medvedev [36] as an attempt to formalise Kolmogoroff's original explanation [30] of the intuitionistic connectives, called the *Aufgabenrechnung* by Kolmogoroff. According to Medvedev's interpretation of the

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

Aufgabenrechnung every atomic proposition α represents a basic *problem* given by an associated set $F(\alpha)$ of *admissible possibilities* and a subset $X(\alpha) \subseteq F(\alpha)$ of distinguished *solutions*. Starting from assignments F and X for atomic propositions every non-atomic formula φ inductively defines a set $F(\varphi)$ of admissible possibilities for φ and a realisability relation $r, X \Vdash \varphi$ which picks out those admissible $r \in F(\varphi)$ that are solutions of the composite problem φ when X gives the solutions to all atomic problems. A composite problem φ then is called *solvable* for F *iff* $\exists r \in F(\varphi).\ \forall X.\ r, X \Vdash \varphi$, *i.e.* there exists a uniform solution $r \in F(\varphi)$ for φ independently of what the solutions X of atomic propositions are. On a formal level our notion of validity up to stabilisation is obtained in a very similar way: $C \models \varphi$ *iff* $\exists c \in [\![\varphi]\!].\ \forall V.\ c, V \Vdash \varphi$. In our setting the uniform stabilisation bounds $c \in [\![\varphi]\!]$ are the uniform solutions, the waveforms V define the solutions of the atomic propositions and thus play the role of X, and the relation $c, V \Vdash \varphi$ gives the solutions $c \in [\![\varphi]\!]$ for composite φ relative to V. At a more detailed level, however, there are three main differences here to Medvedev's realisability interpretation of intuitionistic logic:

(*i*) Medvedev's interpretation quantifies over all interpretations F that associate arbitrary finite sets $F(\alpha)$ of realisers with propositional atoms. Our semantics is more specific in that it uses a fixed choice of singleton sets $[\![a = 1]\!] = [\![a = 0]\!] = \{*\}$ for the propositional atoms. This has to do with the fact that the atoms $a = 1$ and $a = 0$ here are not propositional variables but propositional constants with a fixed semantic interpretation.

(*ii*) Medvedev's as well as many other notions of realisability represent *false* as the empty set, $F(\mathit{false}) = \emptyset$. In our framework $[\![\mathit{false}]\!] = \{*\}$, *i.e. false* has a (single) realiser. A simple technical reason for this is that in our semantics *false* is logically equivalent to $a = 0 \wedge a = 1$ and $[\![a = 0 \wedge a = 1]\!] = \{*\} \times \{*\}$ which is isomorphic to $\{*\}$.

(*iii*) The set $F(\varphi \supset \psi)$ of realisers of an implication in Medvedev's setting is the set $F(\varphi) \to F(\psi)$ of all functions from $F(\varphi)$ to $F(\psi)$. Here, we take $[\![\varphi \supset \psi]\!] = [\![\varphi]\!] \to \mathbb{N} \times [\![\psi]\!]$, so the realisers $f \in [\![\varphi \supset \psi]\!]$ not only consist of a function $\pi_2 \circ f \in [\![\varphi]\!] \to [\![\psi]\!]$ mapping realisers of φ to realisers of ψ but also carry additional timing information $\pi_1 \circ f \in [\![\varphi]\!] \to \mathbb{N}$ that associates a stabilisation delay with every realiser of φ.

(*iv*) Medvedev applies a classical reading of implication whereby $f, X \Vdash \varphi \supset \psi$ *iff* $\forall r \in F(\varphi).\ r, X \Vdash \varphi \Rightarrow f\, r, X \Vdash \psi$. In our semantics we have $f, V \Vdash \varphi \supset \psi$ *iff* $\forall t \in \mathbb{N}.\ \forall c \in [\![\varphi]\!].\ c, V^t \Vdash \varphi \Rightarrow \pi_2(f\, c), (V^t)^{\pi_1(f\, c)} \Vdash \psi$. The key difference to Medvedev (besides the delay by $\pi_1(f\, c)$) is the time shift V^t and universal quantification over t. This amounts to an intuitionistic reading of realisability on waveforms V as linear Kripke models.

To sum up, our timing semantics of intuitionistic logic may be thought of as an *intuitionistic version* of a Medvedev style realisability semantics of *singleton problems* on *linear* Kripke models. For a systematic study of Medvedev's logic of singleton problems the reader is referred to [40]. There, an intuitionistic reading of Medvedev's semantics based on arbitrary Kripke models rather than linear models has been suggested.

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

It is interesting to note that the time variant notion of Kripke validity $\models_t$ based on the ordering relation $\sqsubseteq_t$ on circuits, too, has an equivalent realisability presentation. We define the corresponding relation $\Vdash_t$ as before but take as the realisers of an implication the set $[\![\varphi \supset \psi]\!] = \mathbb{N} \times [\![\varphi]\!] \to \mathbb{N} \times [\![\psi]\!]$ and put $f, V \Vdash_t \varphi \supset \psi$ *iff* for all $t \in \mathbb{N}$ and $c \in [\![\varphi]\!]$ such that $c, V^t \Vdash_t \varphi$ we have $\pi_2(f\,(t,c)), V^{\pi_1(f\,(t,c))+t} \Vdash_t \psi$.

6 Extracting Stabilisation Bounds

When φ represents the specification of a combinational circuit, the existence of a timing bound c is reassuring but really we would like to compute and evaluate c in order to obtain quantitative information about the timing behaviour of the circuit. The Completeness Theorem 5.1, which is derived by classical methods, does not *per se* yield a method for finding a bound. Fortunately, often we do not only know that φ is valid up to stabilisation, but also have a formal proof for it, *i.e.* some extra intensional information about "why" φ is true. We will now show that if the proof is a proof in the calculus LJ (*cf.* Sec. 3) we can indeed extract a uniform bound for φ from this proof. In general, there may be many different ways of translating proofs into timing bounds. In the sequel we will present one such method that produces useful timing information, and discuss its application to combinational circuits.

We first translate derivations into typed lambda terms with explicit finite sums, finite products, and delay constructs, generated by the syntax

$$t \;::=\; x \mid \mathit{del}(n) \mid (t,t) \mid \pi_1\, t \mid \pi_2\, t \mid ?(t) \mid \mathit{case}_{x,y}(t,t,t) \mid \iota_1\, t \mid \iota_2\, t \mid t\, t \mid \lambda x.\, t,$$

where n is an arbitrary natural number. The types are

$$\tau \;::=\; \mathbf{1} \mid \mathbf{0} \mid \tau + \tau \mid \tau \times \tau \mid \tau \to \tau.$$

The formal definition of well-formed terms is as usual(*cf.* [20]). If $\Delta = x_1{:}\tau_1, \ldots, x_n{:}\tau_n$ is a non-repeating list of variables x_i, each one associated with a type τ_i, we write $\Delta \vdash t : \tau$ to denote that t is a well-formed term of type τ with free variables in Δ. Such a Δ is called a *context*. The typing rules are given in Figure 3.
The type of a variable is generic and thus needs to be given explicitly, as well as the type of the terms $?(t)$. The first is dealt with in the contexts. In the last case we write $?^\tau(t)$ to denote this type τ. The nonstandard part of our syntax are the terms $\mathit{del}(n)$, which are the germs for our intensional timing semantics for terms and proofs. They are canonical terms of type $\mathbf{1}$, where the standard λ-calculus only has a single canonical element, say $*$. Intuitively, $\mathit{del}(n)$ represents this standard element $*$ suffering from a delay of n time units. The notion of *bound* and *free* variables and *closed* terms are assumed to be understood.

The translation of derivations into terms is controlled by a translation of formulas into types. The type $|\varphi|$ associated with a formula φ is obtained as follows: $|a = 1| = |a = 0| = \mathbf{1}$, $|\mathit{false}| = \mathbf{0}$, $|\varphi \wedge \psi| = |\varphi| \times |\psi|$, $|\varphi \vee \psi| = |\varphi| + |\psi|$, and $|\varphi \supset \psi| = |\varphi| \to |\psi|$. Now, given a derivation $D : \varphi_n, \ldots, \varphi_1 \vdash \psi$, and a context $\Delta = x_1 : |\varphi_1|, \ldots, x_n : |\varphi_n|$ we

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

$$\frac{}{\Delta, x:\tau, \Delta' \vdash x:\tau} \qquad \frac{}{\Delta \vdash del(n):\mathbf{1}}\ n \in \mathbb{N} \qquad \frac{\Delta \vdash p:\mathbf{0}}{\Delta \vdash ?^{\tau}(p):\tau}$$

$$\frac{\Delta \vdash p:\sigma \quad \Delta \vdash q:\tau}{\Delta \vdash (p,q):\sigma\times\tau} \qquad \frac{\Delta \vdash r:\sigma\times\tau}{\Delta \vdash \pi_1 r:\sigma} \qquad \frac{\Delta \vdash r:\sigma\times\tau}{\Delta \vdash \pi_2 r:\tau}$$

$$\frac{\Delta \vdash r:\sigma+\tau \quad \Delta, y:\sigma \vdash p:\rho \quad \Delta, z:\tau \vdash q:\rho}{\Delta \vdash case_{y,z}(r,p,q):\rho}$$

$$\frac{\Delta \vdash p:\sigma}{\Delta \vdash \iota_1 p:\sigma+\tau} \qquad \frac{\Delta \vdash p:\tau}{\Delta \vdash \iota_2 p:\sigma+\tau}$$

$$\frac{\Delta, z:\sigma \vdash p:\tau}{\Delta \vdash \lambda z:\sigma.\ p:\sigma\to\tau} \qquad \frac{\Delta \vdash p:\sigma\to\tau \quad \Delta \vdash q:\sigma}{\Delta \vdash p\,q:\tau}$$

Figure 3: Typing Rules.

compute a well-formed term $|D|_\Delta$ of type $|\psi|$ with free variables in Δ as follows:

$$\begin{aligned}
|id|_{x,\Delta} &= x \\
|get(i)\cdot D|_\Delta &= |D|_{\Delta^i} \\
|\wedge R\cdot(D_1,D_2)|_\Delta &= (|D_1|_\Delta, |D_2|_\Delta) \\
|\wedge L\cdot D|_{x,\Delta} &= |D|_{y_2,y_1,\Delta}\{y_2/\pi_2\, x\}\{y_1/\pi_1\, x\} \\
|\vee L\cdot(D_1,D_2)|_{x,\Delta} &= case_{y_1,y_2}(x, |D_1|_{y_1,\Delta}, |D_2|_{y_2,\Delta}) \\
|\vee R_1\cdot D|_\Delta &= \iota_1\,|D|_\Delta \\
|\vee R_2\cdot D|_\Delta &= \iota_2\,|D|_\Delta \\
|false L|_{x,\Delta} &= ?(x) \\
|\supset R\cdot D|_\Delta &= \lambda y.\,|D|_{y,\Delta} \\
|\supset L\cdot(D_1,D_2)|_{x,\Delta} &= |D_2|_{y,\Delta}\{y/x\ |D_1|_{x,\Delta}\},
\end{aligned}$$

where Δ^i is the list Δ in which the term at the i-th position has been moved to the front of the list. The variables y, y_1, y_2 are generated by the translation and need to be fresh in each case, *i.e.* must not already occur in Δ and be different from x. Up to the choices of bound variables, the term $|D|_\Delta$ is unique. To enforce this uniqueness we follow the usual custom of identifying terms up to renaming of bound variables.

The next step is to give a denotational semantics for λ-terms so that if D is a proof of φ then the denotational semantics of $t = |D|$, which is abbreviated by $[t]$, is an element of $[\![\varphi]\!]$ and a uniform bound for φ. In general, a term t of type τ is mapped into an element $[t] \in \mathbb{N}\times[\tau]$, where $[\tau]$ is defined so that $[\mathbf{1}] = [\mathbf{0}] = \{*\}$, $[\tau_1\times\tau_2] = [\tau_1]\times[\tau_2]$, $[\tau_1+\tau_2] = [\tau_1]+[\tau_2]$, and $[\tau_1\to\tau_2] = [\tau_1]\to\mathbb{N}\times[\tau_2]$. Note that $[|\varphi|] = [\![\varphi]\!]$.
In order to interpret terms with free variables we need the notion of an *environment*, which is a map ρ that assigns to every variable x of type τ an element $\rho(x) \in \mathbb{N}\times[\tau]$. For an

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

environment ρ, a variable $x : \tau$, and a pair $(t, v) \in \mathbb{N} \times [\tau]$ we write $\rho[x \leftarrow (t, v)]$ for the updated environment that has $\rho[x \leftarrow (t, v)](x) = (t, v)$ and $\rho[x \leftarrow (t, v)](y) = \rho(y)$ for all $y \neq x$. Since for every type τ the set $[\tau]$ is nonempty we are entitled to assume for every τ a predefined choice of an element $\kappa_\tau \in [\tau]$. The semantics is given by the following inductive definition:

- $[x]_\rho = \rho(x)$
- $[del(\delta)]_\rho = (\delta, *)$
- If $[t]_\rho = (\delta, v)$ then $[?^\tau(t)]_\rho = (\delta, \kappa_\tau)$
- If $[t_1]_\rho = (\delta_1, v_1)$ and $[t_2]_\rho = (\delta_2, v_2)$ then $[(t_1, t_2)]_\rho = (max(\delta_1, \delta_2), (v_1, v_2))$
- If $[t]_\rho = (\delta, (v_1, v_2))$ then $[\pi_i\, t]_\rho = (\delta, v_i)$, for $i = 1, 2$
- If $[t]_\rho = (\delta, v)$ then $[\iota_i\, t]_\rho = (\delta, (i, v))$, for $i = 1, 2$
- If $[t]_\rho = (\delta_i, (i, v_i))$ and $[t_i]_{\rho[x_i \leftarrow (0, v_i)]} = (\delta, v)$, then $[case_{x_1,x_2}(t, t_1, t_2)]_\rho = (\delta_i + \delta, v)$, for $i = 1, 2$
- $[\lambda x.\, t]_\rho = (0, f)$ where f is the function that maps every element $v \in [\![\tau_1]\!]$, τ_1 being the type of x, to the value $[t]_{\rho[x \leftarrow (0,v)]}$
- If $t_1 : \tau_2 \to \tau_1$ and $t_2 : \tau_2$ such that $[t_1]_\rho = (\delta_1, f_1)$ with $f_1 \in [\![\tau_2]\!] \to \mathbb{N} \times [\![\tau_1]\!]$, and $[t_2]_\rho = (\delta_2, v_2)$, and $f_1(v_2) = (\delta, v)$, then $[t_1\, t_2]_\rho = (\delta + max(\delta_1, \delta_2), v)$.

It is not difficult to see that the result $[p]_\rho$ only depends on the values $\rho(x)$ for the variables x that actually occur free in p. If p is a closed term then $[p]_\rho$ does not depend on ρ at all, so we may just as well write $[p]$ for it. For the time-variant semantics we would put $[t_1\, t_2]_\rho = (\pi_1((\pi_2[t_1]_\rho)[t_2]_\rho) + max(\pi_1[t_1]_\rho, \pi_1[t_2]_\rho), \pi_2((\pi_2[t_1]_\rho)[t_2]_\rho))$ to translate an application term $t_1\, t_2$ in the last clause above. The soundness lemma below also holds for this time-variant translation.

Proposition 6.1 *Let t be a well-formed term of type τ. Then for every environment ρ the above inductive rules define a uniquely determined element $[t]_\rho \in \mathbb{N} \times [\![\tau]\!]$.*

Proof: Simple, by induction on the structure of t. ∎

If $\pi_1\, [t]_\rho = 0$ then t is called *stable* in environment ρ. For a stable term $t : \tau$ we may confuse $[t]_\rho \in \mathbb{N} \times [\![\tau]\!]$ and $\pi_2\, [t]_\rho \in [\![\tau]\!]$.

Lemma 6.2 (Substitution Lemma) *Let $t_1 : \tau_1$ and $t_2 : \tau_2$ well-formed terms and $x : \tau_1$ a free variable of t_2. Then $[t_2\{x/t_1\}]_\rho = [t_2]_{\rho[x \leftarrow [t_1]_\rho]}$.*

Proof: By induction on t_2. It is important to be aware that the substitution $t_2\{x/t_1\}$ possibly needs to rename bound variables in order to avoid name capture. ∎

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

Lemma 6.3 (Soundness Lemma) *Let $D : \Gamma \vdash \psi$ be a derivation with $\Gamma = \varphi_n, \ldots, \varphi_1$, and $\Delta = x_1 : |\varphi_1|, \ldots, x_n : |\varphi_n|$ a context of variables for Γ. Further let $t_D = |D|_\Delta$ be the extracted λ-term for D. Then for all environments ρ, all $t \in \mathbb{N}$, and all waveforms V,*

$$\forall i \leq n.\ \pi_2\, \rho(x_i),\ V^{t+\pi_1\, \rho(x_i)} \Vdash \varphi_i \quad \Rightarrow \quad \pi_2\, [t_D]_\rho,\ V^{t+\pi_1\, [t_D]_\rho} \Vdash \psi.$$

Proof: Let us first introduce some useful notation. For $c \in \mathbb{N} \times [\![\varphi]\!]$ and $V \in \mathbb{S} \to \mathbb{N} \to \mathbb{B}$ we define $c, V \Vdash^* \varphi$ to stand for the condition $\pi_2\, c, V^{\pi_1\, c} \Vdash \varphi$. Now, let $D : \Gamma \vdash \psi$ be a derivation with $\Gamma = \varphi_n, \ldots, \varphi_1$, and $\Delta = x_1 : |\varphi_1|, \ldots, x_n : |\varphi_n|$ a context of variables for Γ. Let us take $[\![D]\!]_\rho^\Delta$ to abbreviate $[|D|_\Delta]_\rho$, and $\rho(\Delta), V \Vdash^* \Gamma$ to abbreviate $\forall i \leq n.\ \rho(x_i),\ V \Vdash^* \varphi_i$. The proof that for all ρ, t, V,

$$\rho(\Delta), V^t \Vdash^* \Gamma \ \Rightarrow\ [\![D]\!]_\rho^\Delta, V^t \Vdash^* \psi$$

is by induction of the structure of D. We will only treat the cases where D ends in an application of $\supset L$, $\supset R$, or $\vee L$. All other cases, *viz.* rules in $\{id, get(i), \wedge R, \wedge L, \vee R_1, \vee R_2, falseL\}$ are omitted as they are rather simple.

• Suppose D ends in an application of the rule $\supset L$, *i.e.* $D = \supset L \cdot (D_1, D_2)$ with the associated proof tree

$$\frac{D_1 : \Gamma, \varphi \supset \psi \vdash \varphi \qquad D_2 : \Gamma, \psi \vdash \theta}{D : \Gamma, \varphi \supset \psi \vdash \theta}\ \supset L$$

Let Δ be an arbitrary, but fixed, context for Γ and $x : |\varphi \supset \psi|$. The goal is to show

$$\forall \rho, t, V.\quad \rho(\Delta), V^t \Vdash^* \Gamma \quad \&\quad \rho(x), V^t \Vdash^* \varphi \supset \psi \quad \Rightarrow \quad [\![D]\!]_\rho^{x,\Delta}, V^t \Vdash^* \theta.$$

To this end let ρ, t, and V be fixed, and assume

$$\textit{(i)}\quad \rho(\Delta), V^t \Vdash^* \Gamma \qquad\qquad \textit{(ii)}\quad \rho(x), V^t \Vdash^* \varphi \supset \psi.$$

We wish to show that $[\![D]\!]_\rho^{x,\Delta}, V^t \Vdash^* \theta$, where $[\![D]\!]_\rho^{x,\Delta}$ is determined as follows: By definition of the translation $|D|_{x,\Delta} = |\supset L \cdot (D_1, D_2)|_{x,\Delta} = |D_2|_{y,\Delta}\{y/x\,|D_1|_{x,\Delta}\}$, where y is some fresh variable of type $|\psi|$ different from x and not occurring in Δ. Now, by the Substitution Lemma 6.2, we get $[\![D]\!]_\rho^{x,\Delta} = [\![D_2]\!]_{\rho[y \leftarrow c]}^{y,\Delta}$ where $c = [x\,|D_1|_{x,\Delta}]_\rho$. Using our semantic definition we find that $c = (\delta, v)$ such that

$$\begin{aligned} \delta &= \pi_1\,((\pi_2\,\rho(x))(\pi_2\,[\![D_1]\!]_\rho^{x,\Delta})) \ +\ max(\pi_1\,\rho(x), \pi_1 [\![D_1]\!]_\rho^{x,\Delta}) \\ v &= \pi_2\,((\pi_2\,\rho(x))(\pi_2\,[\![D_1]\!]_\rho^{x,\Delta})). \end{aligned}$$

Thus our goal is to verify

$$[\![D_2]\!]_{\rho[y \leftarrow c]}^{y,\Delta}, V^t \Vdash^* \theta. \tag{8}$$

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

The proof of (8) may use the induction hypothesis for both $D_1 : \Gamma, \varphi \supset \psi \vdash \varphi$ with context x, Δ and $D_2 : \Gamma, \psi \vdash \theta$ with context y, Δ, *viz.*

$$\begin{array}{lllll} \forall \rho, t, V. & \rho(\Delta), V^t \Vdash^* \Gamma & \& & \rho(x), V^t \Vdash^* \varphi \supset \psi & \Rightarrow \quad [\![D_1]\!]_\rho^{x,\Delta}, V^t \Vdash^* \varphi \\ \forall \rho, t, V. & \rho(\Delta), V^t \Vdash^* \Gamma & \& & \rho(y), V^t \Vdash^* \psi & \Rightarrow \quad [\![D_2]\!]_\rho^{y,\Delta}, V^t \Vdash^* \theta. \end{array}$$

By choosing the ρ in the second induction hypothesis to be $\rho[y \leftarrow c]$ we can reduce our goal (8) to

$$\rho[y \leftarrow c](\Delta), V^t \Vdash^* \Gamma \quad \& \quad c, V^t \Vdash^* \psi.$$

The first of these two conditions is immediate from *(i)*, observing that the values of both environments $\rho[y \leftarrow c]$ and ρ on the context Δ are the same. Thus we are left with the problem of showing

$$c, V^t \Vdash^* \psi. \tag{9}$$

This is done as follows. We first use assumptions *(i)* and *(ii)* and the first induction hypothesis to infer $[\![D_1]\!]_\rho^{x,\Delta}, V^t \Vdash^* \varphi$, which more explicitly is

$$\textit{(iii)} \quad \pi_2\, [\![D_1]\!]_\rho^{x,\Delta}, V^{t+\pi_1\, [\![D_1]\!]_\rho^{x,\Delta}} \Vdash \varphi.$$

Now, by definition of $\Vdash^*$ *(ii)* implies that for all $(s, d) \in \mathbb{N} \times [\![\varphi]\!]$,

$$d, V^{t+\pi_1\, \rho(x)+s} \Vdash \varphi \quad \Rightarrow \quad (\pi_2\, \rho(x))\, d, V^{t+\pi_1\, \rho(x)+s} \Vdash^* \psi. \tag{10}$$

In particular we may choose $d = \pi_2\, [\![D_1]\!]_\rho^{x,\Delta}$ and $s = max(\pi_1\, [\![D_1]\!]_\rho^{x,\Delta} - \pi_1\, \rho(x), 0)$. Then $s \in \mathbb{N}$ and $s + \pi_1\, \rho(x) \geq \pi_1\, [\![D_1]\!]_\rho^{x,\Delta}$, *i.e.* by *(iii)*, $\pi_2\, [\![D_1]\!]_\rho^{x,\Delta}, V^{t+\pi_1\, \rho(x)+s} \Vdash \varphi$. Now we can apply (10) to obtain $(\pi_2\, \rho(x))\, (\pi_2\, [\![D_1]\!]_\rho^{x,\Delta}), V^{t+\pi_1\, \rho(x)+s} \Vdash^* \psi$. But since $\pi_1\, \rho(x) + s = max(\pi_1\, [\![D_1]\!]_\rho^{x,\Delta}, \pi_1\, \rho(x))$, this is nothing but (9) as desired.

• Suppose D ends in an application of the rule $\supset R$, *i.e.* $D = \supset R \cdot D_1$ with the associated proof tree

$$\frac{D_1 : \Gamma, \varphi \vdash \psi}{D : \Gamma \vdash \varphi \supset \psi}\, \supset R$$

Let Δ be an arbitrary but fixed context for Γ. Let ρ, t, V be given such that *(i)* $\rho(\Delta), V^t \Vdash^* \Gamma$. We must show that

$$[\![D]\!]_\rho^\Delta, V^t \Vdash^* \varphi \supset \psi. \tag{11}$$

We compute $|D|_\Delta = |\supset R \cdot D_1|_\Delta = \lambda x.\ |D_1|_{x,\Delta}$, where x is a fresh variable of type $|\varphi|$. Thus, by the semantics clauses, $[\![D]\!]_\rho^\Delta = [\lambda x.\ |D_1|_{x,\Delta}]_\rho = (0, f)$, where f is the function that maps every $c \in [\![\varphi]\!]$ to the element $f(c) = [\![D_1]\!]_{\rho[x \leftarrow (0,c)]}^{x,\Delta}$. With this information we find that (11) is equivalent to the statement

$$\forall d \in \mathbb{N} \times [\![\varphi]\!].\ d, V^t \Vdash^* \varphi \ \Rightarrow\ [\![D_1]\!]_{\rho[x \leftarrow (0, \pi_2\, d)]}^{x,\Delta}, V^{t+\pi_1\, d} \Vdash^* \psi.$$

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

So, let $d \in \mathbb{N} \times [\![\varphi]\!]$ with *(ii)* $d, V^t \Vdash^* \varphi$ be given. We wish to show

$$[\![D_1]\!]^{x,\Delta}_{\rho[x \leftarrow (0,\pi_2\, d)]}, V^{t+\pi_1\, d} \Vdash^* \psi. \tag{12}$$

We employ the induction hypothesis for $D_1 : \Gamma, \varphi \vdash \psi$ in context x, Δ,

$$\forall \rho, t, V. \quad \rho(\Delta), V^t \Vdash^* \Gamma \quad \& \quad \rho(x), V^t \Vdash^* \varphi \quad \Rightarrow \quad [\![D_1]\!]^{x,\Delta}_{\rho}, V^t \Vdash^* \psi$$

with the particular environment $\rho[x \leftarrow (0, \pi_2\, d)]$ and time $t + \pi_1\, d$ to reduce our goal (12) to

$$\rho[x \leftarrow (0, \pi_2\, d)](\Delta), V^{t+\pi_1\, d} \Vdash^* \Gamma \quad \& \quad (0, \pi_2\, d), V^{t+\pi_1\, d} \Vdash^* \varphi.$$

The second condition is just our assumption *(ii)*, while the first follows from *(i)* by observing that ρ and $\rho[x \leftarrow (0, \pi_2\, d)]$ obtain the same values in context Δ, and that $\rho(\Delta), V^t \Vdash^* \Gamma$ implies $\rho(\Delta), V^{t+\pi_1\, d} \Vdash^* \Gamma$. Thus, we have verified (12), which completes the proof.

• Suppose ends in an application of rule $\vee L$, *i.e.* $D = \vee L \cdot (D_1, D_2)$ with the associated proof tree

$$\frac{D_1 : \Gamma, \varphi \vdash \theta \qquad D_2 : \Gamma, \psi \vdash \theta}{D : \Gamma, \varphi \vee \psi \vdash \theta}\ \vee L$$

Let Δ be a context for Γ and x a variable of type $|\varphi \vee \psi| = |\varphi| + |\psi|$ not occurring in Δ. Let environment ρ, $t \in \mathbb{N}$ and waveform V be fixed such that *(i)* $\rho(\Delta), V^t \Vdash^* \Gamma$ and *(ii)* $\rho(x), V^t \Vdash^* \varphi \vee \psi$. We have to check that

$$[\![D]\!]^{x,\Delta}_{\rho}, V^t \Vdash^* \theta. \tag{13}$$

We prove this by case analysis on $\rho(x) \in \mathbb{N} \times ([\![\varphi]\!] + [\![\psi]\!])$. The first case is that $\rho(x) = (\delta_1, (1, v_1))$ for $\delta_1 \in \mathbb{N}$ and $v_1 \in [\![\varphi]\!]$. We compute

$$|D|_{x,\Delta} = |\vee L \cdot (D_1, D_2)|_{x,\Delta} = case_{y_1,y_2}(x, |D_1|_{y_1,\Delta}, |D_2|_{y_2,\Delta})$$

where y_1, y_2 are fresh variables different from x and not occurring in Δ. Then, with $\rho(x) = (\delta_1, (1, v_1))$, we get $[\![D]\!]^{x,\Delta}_{\rho} = (\delta_1 + \pi_1\, [\![D_1]\!]^{y_1,\Delta}_{\rho[y_1 \leftarrow (0,v_1)]}, \pi_2\, [\![D_1]\!]^{y_1,\Delta}_{\rho[y_1 \leftarrow (0,v_1)]})$. This means, our goal (13) becomes

$$[\![D_1]\!]^{y_1,\Delta}_{\rho[y_1 \leftarrow (0,v_1)]}, V^{t+\delta_1} \Vdash^* \theta. \tag{14}$$

Using the induction hypothesis for $D_1 : \Gamma, \varphi \vdash \theta$ this can be reduced to

$$\rho[y_1 \leftarrow (0, v_1)](\Delta), V^{t+\delta_1} \Vdash^* \Gamma \quad \& \quad (0, v_1), V^{t+\delta_1} \Vdash^* \varphi.$$

The second condition follows from *(ii)* and the first from *(i)*. This finishes the case that $\rho(x) = (\delta_1, (1, v_1))$. The other case, $\rho(x) = (\delta_2, (2, v_2))$ is symmetrical, using the induction hypothesis for D_2. ∎

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

Theorem 6.4 *Let D be a proof of φ and $t_D = |D|$ the corresponding closed λ-term for D. Then t_D is stable and $[t_D]$ is a uniform bound for φ.*

Proof: The first part of the theorem claims that the extracted term t_D for a proof D is stable, *i.e.* $\pi_1\,[t_D] = 0$. This is easy to see since the extracted term t_D does not contain any sub-term of form $del(\delta)$ which are the only terms that could introduce a nonzero delay value into the term. The rest of the theorem, then, follows from the previous Lem. 6.3 as a special case. ∎

With Thm. 6.4 the goal of this section, *viz.* to compute uniform stabilisation bounds for theorems of LJ, is achieved. The remaining task now is to demonstrate that the bounds obtained by this method produce the desired results in concrete circuit verifications. This will be done in the next section.

Before we turn to applications it appears appropriate to spend some words on semantical issues. The obvious question one might ask is what kind of semantics $[\_]$ induces on our terms and *a fortiori* on our proofs in LJ, and whether indeed this justifies calling it a lambda calculus. In fact, it turns out that we get a partial lambda calculus in the sense of Moggi [43], in which the existence predicate $t \downarrow_\tau$ stating that t is a value (rather than a computation) is interpreted as "t is stable."

Proposition 6.5 *Let t be stable. Then,*

$$\begin{array}{rcccl} [(\lambda x.\, s)\, t]_\rho & = & [s\{x/t\}]_\rho & & \\ [\pi_2(t,s)]_\rho & = & [s]_\rho & = & [\pi_1(s,t)]_\rho \\ [case_{x,y}(\iota_1\, t, s_1, s_2)]_\rho & = & [s_1\{x/t\}]_\rho & = & [case_{y,x}(\iota_2\, t, s_2, s_1)]_\rho. \end{array}$$

Proof: By simple check of definition and application of Substitution Lemma 6.2. ∎

Proposition 6.5 lists the standard β-reductions of λ-calculus. The side condition that t be stable is necessary in the sense that all equations have (simple) counter examples where t is not stable. We also have the usual η-equations, also subject to stability conditions:

Proposition 6.6 *Let t be stable. Then,*

$$\begin{array}{rcl} [\lambda x.\, (t\, x)]_\rho & = & [t]_\rho \\ [(\pi_1\, t, \pi_2\, t)]_\rho & = & [t]_\rho \\ [case_{x,y}(t, \iota_1\, x, \iota_2\, y)]_\rho & = & [t]_\rho \end{array}$$

Proof: By simple check of definition. ∎

As far as the η-equations are concerned the side condition of t being stable is only needed for lambda abstraction. Propositions 6.5 and 6.6 establish the soundness of the standard equational calculus for the lambda calculus with function types, products and sums, with the following restriction: variables denote stable terms and hence substitution is restricted

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

to stable terms too. Also a result of the computational nature of our lambda calculus is the fact that $\mathbf{1}$ and $\mathbf{0}$ are not terminal and initial types as usual. If they were, then all terms of type $\mathbf{0} \to \mathbf{1}$ would have to be equivalent. This is not the case. For instance, the terms $\lambda x : \mathbf{0}.\ del(n)$ for $n \in \mathbb{N}$ have type $\mathbf{0} \to \mathbf{1}$ but they all denote different functions $[\lambda x : \mathbf{0}.\ del(n)] \in [\mathbf{0} \to \mathbf{1}]$.

Our semantics can be seen as an interpretation in a particular *computational* λ*-model* [44]: *viz.* in the category of sets with a strong monad $T : A \mapsto \mathbb{N} \times A$ that has unit $\eta_A : A \to \mathbb{N} \times A$ with $\eta(a) = (0, a)$, multiplication $\mu_A : \mathbb{N} \times \mathbb{N} \times A \to \mathbb{N} \times A$ with $\mu(n_1, n_2, a) = (n_1 + n_2, a)$, and tensorial strength $\sigma_{A,B} : (\mathbb{N} \times A \times \mathbb{N} \times B) \to \mathbb{N} \times A \times B)$ defined as $\mu(n_1, a, n_2, b) = (max(n_1, n_2), a, b)$. Within our λ-calculus this special monad is reflected in the structure of the terms of type $\mathbf{1}$: They can be viewed as elements of $\mathbb{N}$ endowed with operations $+$ and max. Given terms $s, t : \mathbf{1}$ we can define $s + t =_{df} (\lambda x.\ s)\, t$ and $max(s, t) = \pi_1\,(s, t)$. These syntactic operations satisfy the standard properties of addition and maximum. In particular, we have $[del(n) + del(m)] = [del(n + m)]$ and $[max(del(n), del(m))] = [del(max(n, m))]$.

In the context of functional programming languages the intensional view is a natural idea. For instance, Douglas Gurr [24] considers non-extensional semantics for a lambda calculus in which explicit complexity information is added. Our delay time, at least *grosso modo*, may be seen as a special case. However, Gurr's notion of complexity only involves the operations of a monoid whereas for our purposes a richer delay algebra, *viz.* $(\mathbb{N}, 0, +, max)$, is needed. Both operations $+$ and max are essential to derive sensible timing information, as we will see in the next section. Thus, our framework applies a more refined view of (time) complexity than Gurr's.

7 Application to Combinational Circuits

We have implemented an experimental prototype system for the timing analysis of combinational circuits based on the ideas presented in the previous sections. Here we wish to illustrate some of the basic ideas on which the potential application of our experimental system rests. In particular, our aim is to demonstrate that

1. in the presence of propagation delays intuitionistic functional verification may be more adequate than classical two-valued reasoning, without being cluttered up by timing details. This contrasts with the other options available, *e.g.* using classical temporal logic.

2. the stabilisation bounds obtained from a proof yield data-dependent timing information of the circuit that has been verified, *i.e.* the computed propagation delays are specific to a particular function and particular input stimulus of the circuit. This contrasts with computing the worst-case (= topological) delay, *i.e.* the longest path through the circuit.

To keep matters short we will not enlarge on the application in general, but focus on the simple combinational circuit shown in Fig. 4 as a suggestive example, leaving the

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

generalisation to the imagination of the reader. We wish to stress that all the analysis steps presented in this section can be done automatically by our implementation. More on the application of our timing semantics for intuitionistic logic can be found in [38, 37].

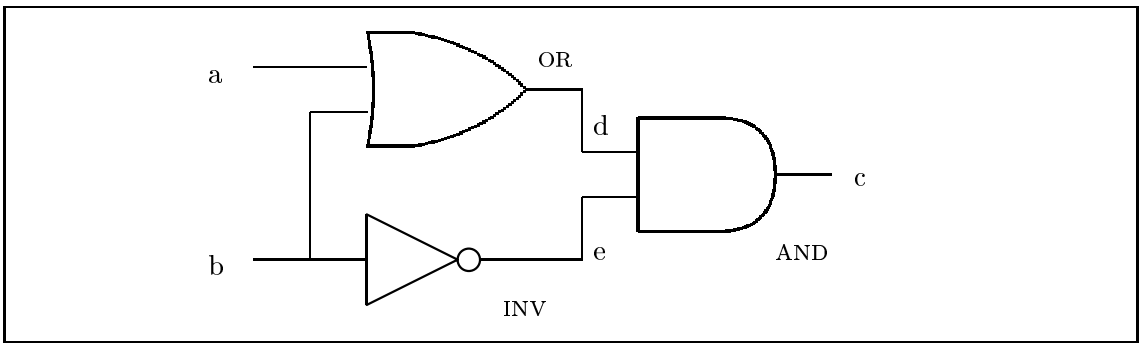

Figure 4: A Simple Combinational Circuit CIRC.

Specifications The circuit's behaviour, as well as the behaviour of its components, is captured by formulas of LJ. For instance, the OR gate in Fig. 4 which has inputs a and b and output d may be specified by

$$\text{OR} \quad = \quad \text{OR}\uparrow \ \wedge \ \text{OR}\downarrow \quad = \quad ((a = 1 \vee b = 1) \supset d = 1) \ \wedge \ ((a = 0 \wedge b = 0) \supset d = 0).$$

In a sense, this specification is nothing but the input-output function table of the OR: The first conjunct OR $\uparrow$ captures the conditions for a rising output transition: "If *one* of the inputs a and b is stable 1, then this gives rise to output d being stable 1 in bounded time." The second conjunct OR $\downarrow$ sums up our assumption about the falling output transition: "If *both* a and b are stable 0, then in bounded time d is stable 0." Let us analyse this in more detail for the rising output transition of OR. By Thm. 5.1 a circuit C satisfies OR $\uparrow$ *iff* there exists a stabilisation bound $c \in [\![\text{OR}\uparrow]\!] = \{*\} + \{*\} \to \mathbb{N} \times \{*\}$ such that OR $\uparrow$ is valid for C with bound c. This, by Prop. 5.2 means that for all $V \in C$ the following formula is true:

$$\begin{aligned}
&(\text{OR}\uparrow)^\sharp(0, c) \\
&\quad = \quad ((a = 1 \vee b = 1) \supset d = 1)^\sharp(0, c) \\
&\quad = \quad \forall t.\, \forall x.\, (a = 1 \vee b = 1)^\sharp(t, x) \Rightarrow (d = 1)^\sharp(t + \pi_1(c\,x), \pi_2(c\,x)) \\
&\quad = \quad \forall t.\, \forall x. \\
&\qquad\qquad (\forall y.\, x = (1, y) \Rightarrow (a = 1)^\sharp(t, y) \ \wedge \ \forall y.\, x = (2, y) \Rightarrow (b = 1)^\sharp(t, y)) \\
&\qquad\qquad\quad \Rightarrow (d = 1)^\sharp(t + \pi_1(c\,x), \pi_2(c\,x)) \\
&\quad \equiv \quad \forall t.\, ((a = 1)^\sharp(t, *) \Rightarrow (d = 1)^\sharp(t + \pi_1(c\,(1, *)), \pi_2(c\,(1, *)))) \ \wedge \\
&\qquad\qquad ((b = 1)^\sharp(t, *) \Rightarrow (d = 1)^\sharp(t + \pi_1(c\,(2, *)), \pi_2(c\,(2, *)))) \\
&\quad = \quad \forall t.\, (\mathit{stable}(V(a), t, 1) \Rightarrow \mathit{stable}(V(d), t + \pi_1(c\,(1, *)), 1)) \ \wedge \\
&\qquad\qquad (\mathit{stable}(V(b), t, 1) \Rightarrow \mathit{stable}(V(d), t + \pi_1(c\,(2, *)), 1)).
\end{aligned}$$

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

The fourth equivalence uses the fact that $\forall x \in \{*\} + \{*\}.\ \varphi(x)$ is semantically the same as $\varphi((1,*))\ \&\ \varphi((2,*))$. Note that $\{*\} + \{*\} \to \mathbb{N} \times \{*\} \cong \mathbb{N} \times \mathbb{N}$, so that c is uniquely determined by two numbers $\delta^1_{\mathrm{OR}\uparrow} = \pi_1(c\,(1,*))$ and $\delta^2_{\mathrm{OR}\uparrow} = \pi_1(c\,(2,*))$. Our derivation shows that if c is the stabilisation bound of the OR gate then $\delta^1_{\mathrm{OR}\uparrow}$ and $\delta^2_{\mathrm{OR}\uparrow}$ are the propagation delays for a rising output transition triggered by the first and second input, respectively. For the falling output transition OR $\downarrow$, in contrast, a stabilisation bound $c \in [\![\mathrm{OR}\downarrow]\!] = \{*\} \times \{*\} \to \mathbb{N} \times \{*\} \cong \mathbb{N}$ is a single number $\delta_{\mathrm{OR}\downarrow}$, *viz.* $\delta_{\mathrm{OR}\downarrow} = \pi_2(c(*,*))$. To sum up we find that $C \models \mathrm{OR}$ is equivalent to the existence of a triple of natural numbers $(\delta^1_{\mathrm{OR}\uparrow}, \delta^2_{\mathrm{OR}\uparrow}, \delta_{\mathrm{OR}\downarrow})$ such that

$$
\begin{array}{l}
\mathrm{OR}^\sharp(0,c) \\
\quad = \quad \forall t.\ (\mathit{stable}(V(a),t,1) \Rightarrow \mathit{stable}(V(d), t+\delta^1_{\mathrm{OR}\uparrow}, 1))\ \wedge \\
\qquad\quad (\mathit{stable}(V(b),t,1) \Rightarrow \mathit{stable}(V(d), t+\delta^2_{\mathrm{OR}\uparrow}, 1))\ \wedge \\
\qquad \forall t.\ \mathit{stable}(V(a),t,0) \wedge \mathit{stable}(V(b),t,0) \Rightarrow \mathit{stable}(V(d), t+\delta_{\mathrm{OR}\downarrow}, 0)
\end{array}
$$

is true for all $V \in C$. This means that a stabilisation bound for OR may be viewed as a data-dependent delay table distinguishing between rising and falling output transitions and between different inputs. In practice these differences do occur, for instance in CMOS technology where one may face a relative variation of more than 50% [23]. Accounting for these differences in the timing model is crucial in wave pipelining applications or in the optimisation of synchronous designs [1]. Even if the primitive gates are mode-led with single fixed delays, the propagation delay of a composite circuit, in general, will be data-dependent, since not all subcomponents are used for every function of the circuit.

It is useful to have a syntactic representation for the stabilisation bounds, alias delay tables. In fact one can show that

$$
T_{\mathrm{OR}} : [\mathrm{OR}] \quad =_{df} \quad (\lambda y.\ \mathit{case}_{x_1,x_2}(y, \mathit{del}(\delta^1_{\mathrm{OR}\uparrow}), \mathit{del}(\delta^2_{\mathrm{OR}\uparrow}))\ ,\ \lambda y.\ \mathit{del}(\delta_{\mathrm{OR}\downarrow}))
$$

is the generic λ-term for $[\![\mathrm{OR}]\!]$, *i.e.* for every element $c \in [\![\mathrm{OR}]\!]$ there exist $\delta^1_{\mathrm{OR}\uparrow}, \delta^2_{\mathrm{OR}\uparrow}, \delta_{\mathrm{OR}\downarrow}$ such that $[T_{\mathrm{OR}}] = c$.

To complete our specification of the gates, here are the formulas for AND and INV

$$
\begin{array}{rcl}
\mathrm{AND} & = & \mathrm{AND}\uparrow\ \wedge \mathrm{AND}\downarrow \quad = \quad (d=1 \wedge e=1 \supset c=1)\ \wedge\ (d=0 \vee e=0 \supset c=0) \\
\mathrm{INV} & = & \mathrm{INV}\uparrow\ \wedge \mathrm{INV}\downarrow \quad = \quad (b=0 \supset e=1)\ \wedge\ (b=1 \supset e=0),
\end{array}
$$

which again are nothing but the formula-isation of their respective input-output function tables. Typical instances of associated delay tables are

$$
\begin{array}{rcl}
T_{\mathrm{AND}} & = & (\lambda y.\ \delta_{\mathrm{AND}\uparrow}\ ,\ \lambda y.\ \mathit{case}_{x_1,x_2}(y, \delta^1_{\mathrm{AND}\downarrow}, \delta^2_{\mathrm{AND}\downarrow})) \\
T_{\mathrm{INV}} & = & (\lambda y.\ \delta_{\mathrm{INV}\uparrow}\ ,\ \lambda y.\ \delta_{\mathrm{INV}\downarrow}).
\end{array}
$$

Let us consider the behaviour of the composite circuit CIRC of Fig. 4. If we used classical 2-valued logic, then, regarding a falling output transition of the circuit, we would be entitled to conclude that $a = 0$ implies $c = 0$. For if $a = 0$ then no matter which

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

value input b assumes, one of the two inputs of the AND is 0. However, in reality this reasoning is deceptive. In fact, assuming $a = 0$, any $1 \to 0$ transition on b may produce a spurious $0 \to 1 \to 0$ glitch at output c, which contradicts $c = 0$. The glitch occurs when the propagation of the $1 \to 0$ edge through the INV is faster than through the OR gate (which is not unreasonable). The flaw in the classical reasoning, of course, is the hidden assumption that input b is either stable 0 or stable 1. Hence, to mend the argument we must enforce explicitly that b is stable. But this is not expressible in a two-valued propositional logic. In our intuitionistic logic this *can* be expressed, *viz.* by the formula $b = 1 \vee b = 0$. The output transitions of the circuit, then, are safely specified by

$$\begin{aligned} \text{CIRC} &= \text{CIRC}\uparrow \ \wedge \text{CIRC}\downarrow \\ &= (a = 1 \wedge b = 0 \supset c = 1) \ \wedge \ (a = 0 \wedge (b = 0 \vee b = 1) \supset c = 0). \end{aligned} \tag{15}$$

Formal Verification The verification goal is the construction of a proof that the composition of the AND, OR, and INV gates results in a circuit satisfying CIRC, *i.e.* we seek a formal derivation of the sequent

$$\vdash (\text{OR} \wedge \text{INV} \wedge \text{AND}) \supset \text{CIRC}. \tag{16}$$

A search for this derivation would apply the rules of LJ given in Fig. 1 in the usual backward manner by unification, and eventually end up with a complete derivation, such as the one seen in Fig. 5. Here, for the sake of compactness some obvious abbreviations have been adopted. For instance, O, I, A stand for the formulas OR, INV, AND, respectively. The sub-derivations

$$\begin{aligned} D_1 &: \text{OR}, \text{INV}, \text{AND}, \vdash \text{CIRC}\uparrow \\ D_2 &: \text{OR}, \text{INV}, \text{AND}, \vdash \text{CIRC}\downarrow \end{aligned}$$

indicated in Fig. 5 establish the two main parts of the circuit's input-output function table. They are composed in an appropriate way to form a derivation D of sequent (16).

If we were concerned only with the functional behaviour of the circuit we could stop at this point. We have certainly invested more work compared to a classical two-valued analysis, but in return we get a verified statement about the functionality of the circuit that is safe even in the presence of (possibly unbounded) propagation delays and oscillations. There is a strong connection between intuitionistic theorem proving in the simple Horn clause fragment and ternary circuit simulation which is discussed in [38].

Translation into Lambda Terms The main advantage of the proposed constructive framework is that we can derive timing information, *viz.* stabilisation bounds, from the derivations, which is not possible in a classical two-valued world. Since all relevant information is already contained in D_1 and D_2 we will focus on these sub-derivations in the sequel.

Intuitively, the derivations D_1 and D_2 not only witness the mere fact that the composition of the components satisfies the functionality of CIRC, but also record the information just

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

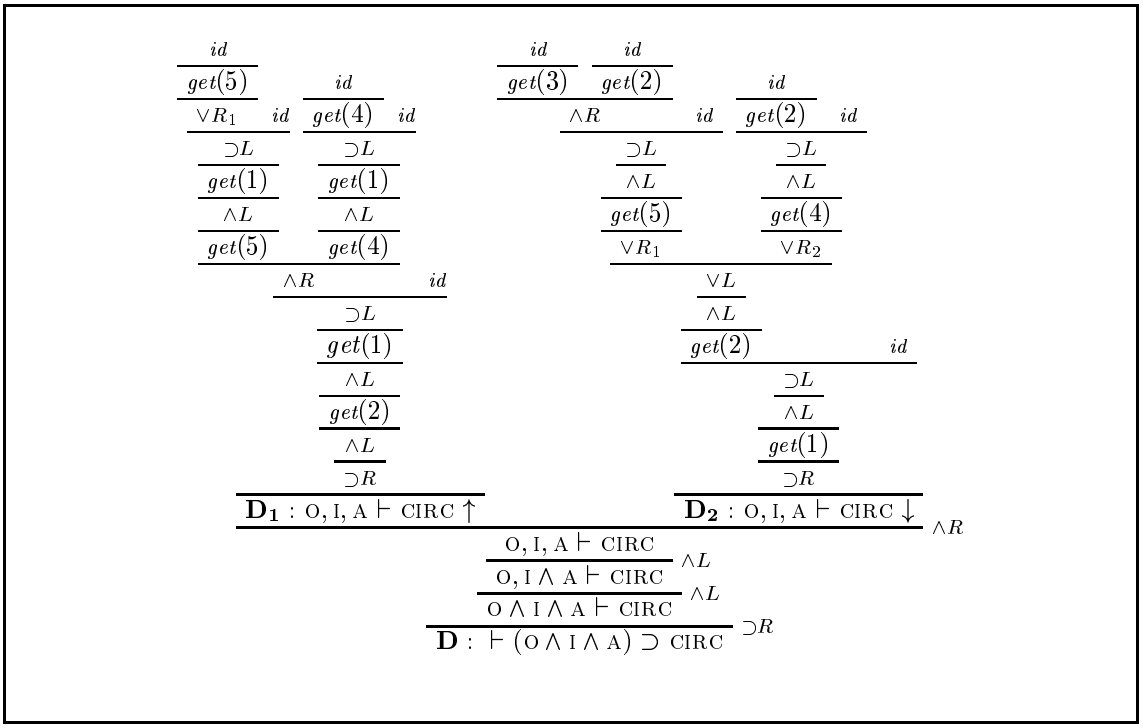

Figure 5: A Possible Proof.

how the goal is met. More precisely, for each part of the functional specification CIRC it is recorded how this part is achieved, what components are used, and what aspect of a component's functionality is involved. Our translation of the derivation into a λ-term, then, extracts this extra information: It represents a timing function which translates the delay tables of OR, INV, and AND to produce a delay table of CIRC.

The translation follows Sec. 6. Since the D_i are open derivations, *i.e.* with premisses on the left-hand side of the sequent turn-stile, the λ-terms obtained are open terms, with free variables being the parameters that represent stabilisation bounds of the sequent's premisses. Let $x_O : [\text{OR}]$, $x_I : [\text{INV}]$, and $x_A : [\text{AND}]$ be variables denoting stabilisation bounds for OR, INV, and AND, respectively. Then the translation as defined in Sec. 6 applied to D_1 yields:

$$
\begin{aligned}
&|D_1|_{x_A\ ,\ x_I\ ,\ x_O} \\
&\quad = \quad |\supset R \cdot \wedge L \cdot \mathit{get}(2) \cdot \wedge L \cdot \mathit{get}(1) \cdot \supset L \cdot (\wedge R \cdot \cdots\ ,\ \mathit{id})|_{x_A\ ,\ x_I\ ,\ x_O} \\
&\quad = \quad \lambda y.\ |\wedge L \cdot \mathit{get}(2) \cdot \wedge L \cdot \mathit{get}(1) \cdot \supset L \cdot (\wedge R \cdot \cdots\ ,\ \mathit{id})|_{y\ ,\ x_A\ ,\ x_I\ ,\ x_O} \\
&\quad = \quad \lambda y.\ |\mathit{get}(2) \cdot \wedge L \cdot \mathit{get}(1) \cdot \supset L \cdot (\wedge R \cdot \cdots\ ,\ \mathit{id})|_{y_2\ ,\ y_1\ ,\ x_A\ ,\ x_I\ ,\ x_O} \\
&\qquad\qquad \{y_2/\pi_2\ y\}\{y_1/\pi_1\ y\} \\
&\quad = \quad \lambda y.\ |\wedge L \cdot \mathit{get}(1) \cdot \supset L \cdot (\wedge R \cdot \cdots\ ,\ \mathit{id})|_{x_A\ ,\ y_2\ ,\ y_1\ ,\ x_I\ ,\ x_O} \\
&\qquad\qquad \{y_2/\pi_2\ y\}\{y_1/\pi_1\ y\} \\
&\quad = \quad \lambda y.\ |\mathit{get}(1) \cdot \supset L \cdot (\wedge R \cdot \cdots\ ,\ \mathit{id})|_{x_2\ ,\ x_1\ ,\ y_2\ ,\ y_1\ ,\ x_I\ ,\ x_O}
\end{aligned}
$$

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

$$
\begin{array}{rcl}
 & & \{x_2/\pi_2\, x_A\}\{x_1/\pi_1\, x_A\}\{y_2/\pi_2\, y\}\{y_1/\pi_1\, y\} \\
= & \lambda y.\, |{\supset}L \cdot (\wedge R \cdot \cdots\ ,\ id)|_{x_1\ ,\, x_2\ ,\, y_2\ ,\, y_1\ ,\, x_I\ ,\, x_O} & \\
 & & \{x_2/\pi_2\, x_A\}\{x_1/\pi_1\, x_A\}\{y_2/\pi_2\, y\}\{y_1/\pi_1\, y\} \\
= & \lambda y.\, |id|_{x\ ,\, x_2\ ,\, y_2\ ,\, y_1\ ,\, x_I\ ,\, x_O} & \\
 & & \{x/x_1\, |\wedge R \cdot \cdots|_{x_1\ ,\, x_2\ ,\, y_2\ ,\, y_1\ ,\, x_I\ ,\, x_O}\}\{x_2/\pi_2\, x_A\}\{x_1/\pi_1\, x_A\}\{y_2/\pi_2\, y\}\{y_1/\pi_1\, y\} \\
= & \lambda y.\, x & \\
 & & \{x/x_1\, |\wedge R \cdot \cdots|_{x_1\ ,\, x_2\ ,\, y_2\ ,\, y_1\ ,\, x_I\ ,\, x_O}\}\{x_2/\pi_2\, x_A\}\{x_1/\pi_1\, x_A\}\{y_2/\pi_2\, y\}\{y_1/\pi_1\, y\} \\
= & \lambda y.\, (\pi_1\, x_A)\, |\wedge R \cdot \cdots|_{x_1\ ,\, x_2\ ,\, y_2\ ,\, y_1\ ,\, x_I\ ,\, x_O} & \\
 & & \{x_2/\pi_2\, x_A\}\{x_1/\pi_1\, x_A\}\{y_2/\pi_2\, y\}\{y_1/\pi_1\, y\} \\
\vdots & & \vdots \\
= & \lambda y.\, (\pi_1\, x_A)\, ((\pi_1\, x_O)\, (\iota_1\, (\pi_1\, y))\ ,\ (\pi_1\, x_I)\, (\pi_2\, y)). &
\end{array}
$$

One verifies that this is a well-formed term of type $\mathbf{1} \times \mathbf{1} \to \mathbf{1}$. In a similar way we translate the other derivation and obtain:

$$|D_2|_{x_A\ ,\, x_I\ ,\, x_O} \quad = \quad \lambda y.\, (\pi_2\, x_A)\, case_{x_1,x_2}(\pi_2\, y, \iota_1\, ((\pi_2\, x_O)\, (\pi_1\, y, x_1)), \iota_2\, ((\pi_2\, x_I)\, x_2)),$$

which has type $\mathbf{1} \times (\mathbf{1} + \mathbf{1}) \to \mathbf{1}$.

Timing Evaluation Now that we have in our hands the lambda terms D_1, D_2, we can evaluate them to work out the timing table for CIRC. This timing table is parametric in the timing tables T_{OR}, T_{INV}, and T_{AND} for the components OR, INV, and AND, respectively, which must be substituted for the free variables x_O, x_I, and x_A. The timing table for CIRC then is uniquely determined by three numbers $\delta_{\text{CIRC}\uparrow}$, $\delta^1_{\text{CIRC}\downarrow}$, and $\delta^2_{\text{CIRC}\downarrow}$ which are computed as follows:

$$
\begin{array}{rcl}
\delta_{\text{CIRC}\uparrow} & = & \pi_1\, [(|D_1|_{x_A,x_I,x_O}\, \{x_O/T_{\text{OR}}\}\{x_I/T_{\text{INV}}\}\{x_A/T_{\text{AND}}\})\, (del(0), del(0))] \\
\delta^1_{\text{CIRC}\downarrow} & = & \pi_1\, [(|D_2|_{x_A,x_I,x_O}\, \{x_O/T_{\text{OR}}\}\{x_I/T_{\text{INV}}\}\{x_A/T_{\text{AND}}\})\, (del(0), \iota_1(del(0)))] \\
\delta^2_{\text{CIRC}\downarrow} & = & \pi_1\, [(|D_2|_{x_A,x_I,x_O}\, \{x_O/T_{\text{OR}}\}\{x_I/T_{\text{INV}}\}\{x_A/T_{\text{AND}}\})\, (del(0), \iota_2(del(0)))].
\end{array}
$$

It is not difficult to compute these delays by unravelling the semantics in Sec. 6:

$$
\begin{array}{rcl}
\delta_{\text{CIRC}\uparrow} & = & max(\delta^1_{\text{OR}\uparrow}, \delta_{\text{INV}\uparrow}) + \delta_{\text{AND}\uparrow} \\
\delta^1_{\text{CIRC}\downarrow} & = & \delta_{\text{OR}\downarrow} + \delta^1_{\text{AND}\downarrow} \\
\delta^2_{\text{CIRC}\downarrow} & = & \delta_{\text{INV}\downarrow} + \delta^2_{\text{AND}\downarrow}.
\end{array}
$$

The delays are data-dependent and take account of what part of the functionality is achieved by which components and along which path through the circuit. For instance, the falling output transition triggered by $a = 0$ and $b = 0$ has delay $\delta^1_{\text{CIRC}\downarrow} = \delta_{\text{OR}\downarrow} + \delta^1_{\text{AND}\downarrow}$ since the falling edge in this case propagates through the OR and the *first* input of the AND (*cf.* Fig. 4). On the other hand, if the falling output transition is triggered by $a = 0$ and $b = 1$, then the falling edge travels through INV and the *second* input of AND, whence

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

$\delta^2_{\text{CIRC}\downarrow} = \delta_{\text{INV}\downarrow} + \delta^2_{\text{AND}\downarrow}$. Only for a rising output transition all three gates are used, and we get $\delta_{\text{CIRC}\uparrow} = max(\delta^1_{\text{OR}\uparrow}, \delta_{\text{INV}\uparrow}) + \delta_{\text{AND}\uparrow}$. Note that all three delays $\delta_{\text{CIRC}\uparrow}$, $\delta^1_{\text{CIRC}\downarrow}$, and $\delta^2_{\text{CIRC}\downarrow}$ may actually be smaller than the so-called *topological* delay through the circuit, *i.e.* the longest path. To sum up, the delays obtained may depend on the input value, at which gate input this value is imposed, and also (though this has not been shown in the example) they may depend on the value of the side inputs. For instance we could blow up the specification of AND $\downarrow$ (*cf.* page 22) to $(d = 0 \vee e = 0 \vee (d = 0 \wedge e = 0) \vee (d = 0 \wedge e = 1) \vee (e = 0 \wedge d = 1)) \supset c = 0$ in which we can distinguish 5 different delays for the falling output transition on c, depending on whether we use information on both inputs or only on one. It is known that real circuits indeed show context dependent delay behaviour. This granularity of data dependency certainly reaches beyond most existing techniques for automated timing analysis. More on the different timing models and their relation to standard analyses can be found in [37].

8 Why Intuitionistic Logic?

In proposing the formalism of intuitionistic logic for the combined functional and temporal analysis of combinational circuits the present paper deviates from consolidated tradition. This demands explanation. To justify our approach we shall first make some general comments that put this work into a larger perspective and then give a number of more pragmatic arguments in favour of the intuitionistic approach.

Combinational digital circuits are usually modelled using Boolean algebra or classical propositional logic. The idea of identifying the truth-values of classical propositional logic with the signal values of binary switching circuits is old and successful. According to Church [10] it was suggested first by Ehrenfest in 1910. Today digital hardware design can hardly be imagined without the mathematical tools of Boolean algebra and classical propositional logic. The success of the two-valued model is due to its simplicity and the drastic abstraction it offers from the physical behaviour of an electric circuit. It conveniently ignores low-level effects such as propagation delays, transistor threshold, signal strengths, power consumption, heat dissipation. The abstraction, however, has its price. The correctness of combinational systems established by abstract reasoning in the two-valued model is only valid in limited contexts and operating conditions. Not all these constraints are equally explicit for a given design style. Surely, combinational circuits in synchronous design, *e.g.* must not contain feed-back loops, be clock-driven with sufficiently long clock phases, and all external inputs must be synchronised. Less explicit constraints are the assumption that the voltage of the power supply and the external temperature do not exceed certain limits. When these abstraction constraints cannot be met the system breaks down, or, what is the same thing, fails to satisfy its abstract specification. In order to stay in control of correctness we are thus forced to leave the classical two-valued setting and use a many-valued model that includes sufficient lower-level details.

There are several many-valued extensions of the Boolean model in hardware design relating both to static as well as dynamic lower-level effects. Consider the *static* behaviour of a CMOS transistor. Its classical two-valued switch model [22] is adequate under the

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

constraint that every point in the transistor network is either connected to power or to ground through complementary pull-up and pull-down networks. Sometimes this restrictive design rule is violated on purpose to permit more optimal designs. In such a case the two-valued switch model would make false predictions and hence the circuit malfunction w.r.t. the abstract specification. To resolve the inconsistency many-valued extensions to the classical transistor switch model have been suggested. They increase the precision of the abstract model to deal with more general transistor networks that do not follow the strict CMOS design rules. The simplest is the four-valued model $\mathbb{V} = \{0, 1, X, Z\}$ [16, 58] in which the two extra values stand for "short-circuit" (X) and "floating" (Z), or the four-valued model [28, 5] in which every signal has both a "value" and a "strength" bit. Other, more sophisticated many-valued models including strength and connectivity are discussed in [55, 54, 7]. Many non-Boolean simulation models also capture *dynamic* low-level effects and timing information. This is crucial when a logic gate is to be used as a component of an asynchronous circuit, or when the correctness of a synchronous circuit itself is to be verified. In such cases stabilisation delays become relevant as well and delay-related transient phenomena like *hazards*, *races*, *glitches* may need to be accounted for. To do this the time dimension must be modelled in one way or other. Kleene's three-valued signal algebra $\mathbb{K} = \{0, \frac{1}{2}, 1\}$ [29] has long been used in hardware design in order to analyse hazards or oscillations, as in [57, 13] or to simulate circuits with feed-back, on the gate level as in [56, 32] or transistor level as in [8]. In these applications, the third value $\frac{1}{2}$ subsumes all the transient behaviours, in particular oscillations, that fall outside the static two-valued world. When quantitative timing is important then Boolean algebra may be replaced by *Timed Boolean Functions* [31], timed transition systems [18], monadic second-order logic over linear chains [4, 33], temporal logic [53, 45], or predicate logic with time variables [49, 27, 14].

These classical approaches may all be seen as careful attempts to include into the Boolean algebra or classical propositional logic a minimum amount of additional low-level parameters (strength, impedance, intermediate voltage, connectivity, transients, *etc.*). How much ballast needs to be taken on board depends on the context. The classical rule is that all semantic details for which we wish to obtain reliable information must be explicitly present in the formalism and all of them must be anticipated in the logic at the outset. However, this means that the classical approach of *just-another-signal-value* or *just-another-model-parameter* is inherently incomplete. It is incomplete since for every new low-level coupling between components that is discovered to be relevant for the correctness of the system in another previously unforeseen design context, another set of signal values must be devised or another physical parameter must be included and made explicit in the model. In other words, we must consider as many different models as there are different design contexts. In some situations finite models are insufficient and we may even be forced to give up all abstractness and resort to real number calculus and differential equations. One such example is the hardware arbiter. Its correct functioning depends on implicit timing constraints on external inputs which often cannot be guaranteed. The result is that arbiters may malfunction [34] and violate their abstract specification [39]. The problem is the fundamental mismatch between the discrete abstract specification of the arbiter and its continuous realisation as an electronic circuit. Therefore, no finite-valued extension of the

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

classical model can bridge the gap. It seems we must reintroduce analog electronics and use dynamic system and control theory to give a satisfactory account of the correctness of arbiters [41].

From a methodological point of view the classical ad-hoc extensions of the Boolean model are unsatisfactory. There is, however, a more robust way to go about taking care of correctness across abstraction levels. Rather than introducing ever more structure into our representation of signal values we may keep the ideal two-valued signals but apply a more refined notion of truth and truth values. This is where intuitionistic logic comes into play. Reasoning in intuitionistic logic only relies on positive information, *i.e*, on what can be deduced from the structure of the formulas alone, without using implicit assumptions on the context. Intuitionistic logic provides for an infinity of truth values (via Kripke models) and permits many notions of realisability. In this way intuitionistic truth is sustainable under extensions of the context and refinement of the semantic models. In contrast, classical logic has a closed world assumption to justify indirect arguments deducing the presence of some features from the absence of others. This is why classical correctness, if taken strictly, assumes complete knowledge of the context of a system and the interaction of all of its components at all levels of detail.

The intuitionistic viewpoint is not completely new. There is a distinct intuitionistic flavour already to the classical many-valued models. In fact, the elements of the extended signal domains $\mathbb{V} = \{0, 1, X, Z\}$ or $\mathbb{K} = \{0, \frac{1}{2}, 1\}$ mentioned above are not actually signal values but bits of information about a signal. They are not represented in the circuit in any concrete sense by physical parameters, but by the fact that the system has a particular property related to a given signal. This is formally reflected by the fact that the values in $\mathbb{V}$ or $\mathbb{K}$ are not independent but related by an ordering that measures this information. $\mathbb{K}$ is a domain of information with $\frac{1}{2} \sqsubseteq 0$ and $\frac{1}{2} \sqsubseteq 1$ in which $\frac{1}{2}$ stands for "unknown" rather than any particular undefined behaviour such as oscillation. Thus, the third value is given a special status and no longer on a par with 0 and 1. This is the original reading of Kleene [29], and it is implicit in most ternary simulation approaches like [9, 32, 8]. In $\mathbb{V}$ the ordering is $Z \sqsubseteq 0$, $Z \sqsubseteq 1$, $1 \sqsubseteq X$, $0 \sqsubseteq X$. Again, it does not make sense to read Z (floating) and X (short-circuit) as any specific behaviour. For surely it depends on the context and circumstances what physical signal Z or X is an abstract representation of. The reader is referred to [6, 42] for a discussion of some of the problems that arise when $\frac{1}{2}$ or Z, respectively, are naively interpreted as concrete signal values. Indeed, the sets $\mathbb{K}$ and $\mathbb{V}$ are better thought of as domains of truth values rather than signal values. All properties that hold of a given system where some signal has an associated value x must also hold of the system in which this value is increased to y, $x \sqsubseteq y$. Fourman's response model [16] generalises this intuitionistic point of view by replacing $\mathbb{K}$ and $\mathbb{V}$ by arbitrary complete lattices. It has also been argued [26] that by introducing uncertainty and energy considerations into the two-valued switch-level model one naturally obtains Heyting algebras, *i.e.* models of intuitionistic logic. The properties that we can express about a single fixed signal in the timing interpretation of intuitionistic logic presented in this paper also generates a Heyting algebra. One can show that the propositions (up to semantic equivalence) built from $a = 0, a = 1$ for every fixed $a \in \mathbb{S}$ correspond to the set

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

of all upper-closed (non-empty) subsets of the three-dimensional grid $\{0,1\} \times \{0,1,2\} \times \{0,1,2\}$, where $\sqsubseteq$ is the component-wise ordering of natural numbers. Note this is not a Boolean algebra and thus cannot be represented by classical properties over a (finite) set of signal values.

The timing semantics discussed in this paper is only one of many possible intensional interpretations of intuitionistic logic. We believe that this openness in the non-Boolean dimension of the intuitionistic approach is very profitable for formal hardware verification, in order to preserve the validity of correctness arguments in passing from higher to lower levels of abstraction. In the following we sketch some concrete methodological advantages of the intuitionistic over the classical approach.

- **Constructiveness.** In contrast to classical logic proofs in intuitionistic logic are constructive. Following the *propositions-as-types* or *proofs-as-programs* principle they encode computations. By interpreting proofs as terms of the lambda calculus one can extract programs [46, 25] or circuits [3]. In this paper we use a more specialised version which might be called the *proofs-as-delays* principle. Every proof p of an atomic proposition $a = 1$ represents a computation of a natural number $[\![p]\!] \in [\![a = 1]\!] = \mathbb{N} \times \{*\} \cong \mathbb{N}$ that gives an upper bound on the time when signal a stabilises to 1.

- **Abstractness.** Rather than introducing explicit time and time variables as in classical predicate and temporal logics our intuitionistic approach sticks with the abstract two-valued model in the sense that it is built only on the two atomic statements $a = 1$ and $a = 0$ for "signal a is stable 1" and "signal a is stable 0," respectively. The necessary model-theoretic structure to deal with time and stabilisation is introduced separately by a specific choice of a Kripke and realisability semantics in Sections 4 and 5. It shares with many-valued signal models the property that the negation $\neg a = 0$ is not the same as $a = 1$. Knowing that signal a is never stabilising to 0 does not imply it must be constant 1. Other non-classical interpretations which correspond to other semantic refinements of the two-valued setting are possible without the need to extend the abstract language. We just use a different computational interpretation of proofs.

- **Expressiveness.** It is known that a complete semantic characterisation of intuitionistic propositional logic requires an infinite number of truth values [21]. One possible such set of truth values may be obtained in terms of Kripke model structures [47]. Since Kripke models are also the basis for modal and temporal logics it is not surprising that intuitionistic propositional logic can be used to capture time-dependent dynamic behaviour. This paper gives a particular instance of this idea. Note that a simple implication such as $a = 0 \supset b = 1$, which in our semantics formalises bounded reaction, has second-order expressiveness. In fact, the property of fixed but unknown response time cannot be expressed in first-order predicate logic, propositional temporal and modal logics, or even monadic second-order successor logic. Our specific semantics of intuitionistic propositional logic corresponds to a fragment of higher-order logic.

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

- **Separation of Concerns.** As demonstrated in this paper intuitionistic logic permits a natural separation between the intensional aspect of timing and the extensional aspect of function. The timing is obtained from the proofs and the function is specified in the formula. In separating these two aspects our approach follows more closely the usual engineering practice. In a classical setting time would have to represented explicitly by syntactic constructs such as time variables, quantifiers, or next-state operator, and mixed up with the functional specification. This means that the timing abstraction of the ideal Boolean specification is lost and the separation between functional and timing aspects destroyed.

Having stressed the advantages of the intuitionistic over the classic method we must not forget to mention the disadvantages, too. Timing analysis in the intuitionistic setting as proposed here means we must evaluate and store proof terms or realisers. This is expensive computationally and may be a limiting factor in applications. However, as long as manipulating the proof terms (or realisers) corresponds to manipulating essential semantic information the intuitionistic should be computationally conservative over traditional methods. For, after all, the timing information in the classical and algebraic setting must also be represented and handled in one way or other. The characteristic feature of the intuitionistic way of organising affairs is to push into the proofs those parts of the relevant semantic information that are mainly algorithmic while the functional (or non-algorithmic) aspects remain in the formulas.

9 Discussion

In this paper we proposed an application of constructive theorem proving to the timing analysis of hardware. A concrete framework has been outlined based on a special semantic interpretation of intuitionistic propositional logic, combined with an appropriate stabilisation semantics for proofs. Via the semantic interpretation intuitionistic logic is turned into an adequate means to reason about the functional behaviour of combinational circuits in the presence of propagation delays, oscillations, and glitches. We have shown, by way of an example, that data-dependent timing information can be extracted using the proposed stabilisation semantics of proofs. The usual Boolean steady-state analysis is contained as a special case, *viz.* the double negated formulas. Thus we can mix classical reasoning about the steady state and intuitionistic reasoning about stabilisation behaviour within one and the same formalism. The interplay with respect to timing extraction between the classical and the intuitionistic analysis needs further exploration.

Given a circuit specification φ, every proof of φ gives rise to a uniform stabilisation bound for φ. Hence, this bound will be a property not only of the circuit but also of a particular proof. For instance, if a circuit has several alternative ways of achieving a particular function we may have different proofs and consequently different stabilisation bounds. In these cases the actual delay of the circuit can be characterised as the *minimal* uniform stabilisation bound for φ.

The stabilisation theory of circuits defined as the set of all φ such that $\models \varphi$ is a deductively closed set of formulas, that deserves to be investigated in its own right. We know by

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

Thm. 4.1 that LJ (over the atomic sentences $a = i$) is contained in LJS. It is not difficult to see that in fact LJS is properly contained between LJ and classical logic, whence it is a so-called *intermediate* theory. The axiom scheme KP of Kreisel and Putnam, $(\neg\varphi \supset \psi_1 \vee \psi_2) \supset ((\neg\varphi \supset \psi_1) \vee (\neg\varphi \supset \psi_2))$, for instance, is contained in LJS, but not in LJ. It is beyond the scope of this paper to discuss the issue of whether it is possible to give a complete axiomatisation for LJS. The results in [38] (for Propositional Lax Logic) essentially show that the sequent-style formulation of SLD resolution is intensionally complete for the Horn fragment of LJS.

We have implemented an experimental prototype system for the combined timing and functional analysis of combinational circuits based on the ideas presented in this paper. The code for the central part of the system, the LJ theorem prover, was obtained from Roy Dyckhoff. It implements a variant of his contraction-free system [12] with optimisations due to Torkel Franzén [17]. Among the other parts of the program are a lambda compiler for proof scripts, and a timing evaluator for lambda terms. All code is written in Prolog and implemented in BinProlog [50]. Though the applicability has been established in theory and small circuits have been analysed, it is too early to assess the feasibility of the approach on realistic circuit sizes. Let us point out, however, that our method algorithmically is much better behaved than it would appear in view of the fact that decidability of intuitionistic logic is PSPACE-complete [48]. Firstly, we are concerned not with intuitionistic logic proper but with a particular intuitionistic theory. Secondly, in many cases we can do with a fragment of intuitionistic logic. The examples of Section 7, for instance, use Horn formulas to specify essentially the function table of a component. The decision problem for propositional Horn formulas is polynomial and for acyclic theories such as the ones obtained from loop-free finite circuits even linear. Since the naive encoding of complete binary function tables results in exponential size Horn formulas timing analysis by intuitionistic theorem-proving is exponential. This is not surprising since full data-dependent timing analysis for combinational circuits is NP-complete [35]. Using more sophisticated encoding techniques for function tables, *e.g.* analogous to binary decision diagrams [2], it should be possible to achieve similar speed ups as in classical two-valued logic. A distinct algorithmic advantage, so we believe, of the proposed logical method is its flexibility. It provides a uniform framework in which components may be specified at different levels of abstraction with different degrees in the data-dependency of the delays. In this way it should be possible to implement timing analyses of high precision by combining various forms of approximative and hierarchical analyses of polynomial complexity. In [37] it has been shown (in the more general setting of Propositional Lax Logic) how timing abstractions can be obtained with restricted specification styles, and how these can be used to characterise the correctness and completeness of some standard timing analysis algorithms.

Acknowledgements I am very grateful to Matt Fairtlough and Pierangelo Miglioli for the many useful discussions on timing semantics and intermediate intuitionistic logic. Thanks also go to Roy Dyckhoff for making available his Prolog theorem prover for LJ and to Terry Stroup for his comments on a draft of this paper. I would like to thank one of the referees for especially careful reading and astute questions that helped me improve

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

the presentation of this paper. This work was supported by the *Deutsche Forschungsgemeinschaft* under grant number Me-1427 1-3.

References

[1] ACM. *International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, Intermar Hotel Malente, Germany, September 1993.

[2] S. Akers. Binary decision diagrams. *IEEE Transactions on Computers*, C-27:509–516, 1978.

[3] D. Basin. Extracting circuits from constructive proofs. In *Int'l Workshop on: Formal Methods for VLSI Design*, Miami, USA, January 1991. IFIP-IEEE.

[4] D. Basin and N. Klarlund. Hardware verification using monadic second-order logic. In *Proc. Computer Aided Verification CAV'95*, pages 31–41. Springer, 1995. LNCS 939.

[5] J. A. Bergstra and J. W. Klop. A proof rule for restoring logic circuits. *INTEGRATION, the VLSI Journal*, 1:161–178, 1983.

[6] M. A. Breuer. A note on three-valued logic simulation. *IEEE Transactions on Computers*, C-21(4):399–402, April 1972.

[7] R. E. Bryant. A switch-level model and simulator for MOS digital systems. *IEEE Transactions on Computers*, C-33:160–177, February 1984.

[8] R. E. Bryant. Boolean analysis of MOS circuits. *IEEE Transactions on Computer-Aided Design*, 6(4):634–649, July 1987.

[9] J. A. Brzozowski and M. Yoeli. On a ternary model of gate networks. *IEEE Transactions on Computers*, C-28:178–184, 1979.

[10] A. Church. Logic, arithmetic and automata. In *Proc. of the Int. Congr. of Mathematicians, Stockholm 1962*, pages 23–35. Almquist and Wiksells, 1963.

[11] A. G. Dragalin. *Mathematical Intuitionism. Introduction to Proof Theory.* American Mathematical Society, 1988.

[12] R. Dyckhoff. Contraction-free sequent calculi for intuitionistic logic. *The Journal of Symbolic Logic*, 57(3):795–807, September 1992.

[13] E. B. Eichelberger. Hazard detection in combinational and sequential switching circuits. *IBM J. Res. Div.*, 9:90–99, 1965.

[14] H. Eveking and Ch. Mai. Formal verification of timing conditions. In *European Design Automation Conference*, pages 512–517, 1990.

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

[15] M. Fairtlough and M. Mendler. Propositional Lax Logic. *Information and Computation*, 137(1):1–33, August 1997.

[16] M. Fourman. Proof and design. In M. Broy, editor, *Deductive Program Design*, pages 397–439. Springer, 1996.

[17] Torkel Franzén. Algorithmic aspects of intuitionistic propositional logic. Research Report SICS R87010, Swedish Institute of Computer Science, 1987.

[18] J. Frößl and Th. Kropf. A new model to uniformly represent function and timing of MOS circuits and its application to VHDL simulation. In *European Design Automation Conference*, 1994.

[19] G. Gentzen. Untersuchungen über das Logische Schließen. *Math. Z.* , 39:176–210, 405–431, 1934–1935.

[20] J. Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types.* Cambridge University Press, 1989.

[21] K. Gödel. Zum Intuitionistischen Aussagenkalkül. *Akademie der Wissenschaften in Wien, Mathematisch-Naturwissenschaftliche Klasse. Anzeiger.*, 69:65–66, 1932.

[22] M. Gordon. Why higher-order logic is a good formalism for specifying and verifying hardware. In P. A. Subrahmanyam and G. Milne, editors, *Formal Aspects of VLSI design*, pages 153–178. North Holland, 1986.

[23] C. T. Gray, W. Liu, and R. K. Cavin III. Exact timing analysis considering data dependent delays. In *International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, Malente, September 1993. ACM/GMD.

[24] D. J. Gurr. *Semantic Frameworks for Complexity.* PhD thesis, Edinburgh University, Department of Computer Science, January 1991.

[25] S. Hayashi and H. Nakano. *PX, A Computational logic.* MIT press, 1989.

[26] J. P. Hayes. Uncertainty, energy, and multiple-valued logics. *IEEE Transactions on Computers*, C-35(2):107–114, February 1986.

[27] J. Herbert. Formal verification of basic memory devices. Technical Report 124, University of Cambridge, Computer Laboratory, February 1988.

[28] C. A. R. Hoare. A theory for the derivation of C-mos circuit designs. In W. H. J. Feijen et. al., editor, *Beauty is our Business.* Springer, 1990.

[29] S. C. Kleene. *Introduction to Metamathematics.* North Holland, Amsterdam, 1952. Chap. XII, Par. 64.

[30] A. Kolmogoroff. Zur Deutung der intuitionistischen Logik. *Mathematische Zeitschrift*, 35:58–65, 1932.

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

[31] K. C. Lam and R. K. Brayton. *Timed Boolean Functions. A Unified Formalism for Exact Timing Analysis.* Kluwer, 1994.

[32] S. Malik. Analysis of cyclic combinational circuits. In *International Conference on Computer-Aided Design*, pages 618–625. IEEE, 1993.

[33] T. Margaria and M. Mendler. Model-based automatic synthesis and analysis in second-order monadic logic. In *Proceedings First ACM SIGPLAN Workshop on Automated Analysis of Software*, pages 99–112, Paris, France, 14. January 1997.

[34] L. R. Marino. General theory of metastable operation. *IEEE Transactions on Computers*, 30(2):107–115, February 1981.

[35] P. C. McGeer and R. K. Brayton. *Integrating Functional and Temporal Domains in Logic Design.* Kluwer, 1991.

[36] Ju. T. Medvedev. Interpretation of logical formulas by means of finite problems. *Soviet Math. Dokl.*, 7(4):857–860, 1966.

[37] M. Mendler. Characterising timing analysis in intuitionistic modal logic. In R.J.G.B. de Queiroz and M. Finger, editors, *Workshop on Logic, Language, Information, and Computation*, pages 132–140. Department of Computer Science, IME/USP University of São Paulo, Brazil, 1998.

[38] M. Mendler and M. Fairtlough. Ternary simulation: A refinement of binary functions or an abstraction of real-time behaviour? In M. Sheeran and S. Singh, editors, *Proceedings of the 3rd Workshop on Designing Correct Circuits (DCC96)*. Springer, October 1996. Springer Electronic Workshops in Computing.

[39] M. Mendler and T. Stroup. Newtonian arbiters cannot be proven correct. *Formal Methods in System Design*, 3(3):233–257, November/December 1993.

[40] P. Miglioli, U. Moscato, M. Ornaghi, S. Quazza, and G. Usberti. Some results on intermediate constructive logics. *Notre Dame Journal of Formal Logic*, 30(4):543–562, 1989.

[41] I. Mitchell and M. R. Greenstreet. Proving newtonian arbiters correct, almost surely. In M. Sheeran and S. Singh, editors, *Designing Correct Circuits.* Springer Electronic Workshops in Computing, 1996.

[42] M.J.C. Gordon, P. Loewenstein, and M. Shahaf. Formal verification of a cell library: A case study in technology transfer. In L.J.M. Claesen, editor, *Proceedings of the IFIP International Workshop on Applied Formal Methods for Correct VLSI Design*, volume 2, pages 409–417, Leuven, Belgium, 1990. North-Holland.

[43] E. Moggi. *The partial lambda calculus.* PhD thesis, Edinburgh University, Department of Computer Science, August 1988.

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.

[44] E. Moggi. Notions of computation and monads. *Information and Computation*, 93:55–92, 1991.

[45] B. Moszkowski. A temporal logic for multilevel reasoning about hardware. *IEEE Computer*, 18(2):10–19, 1985.

[46] M. Parigot. Programming with proofs: A second-order type theory. In H. Ganzinger, editor, *European Symposium on Programming*, pages 145–159. Springer, 1988. LNCS 300.

[47] K. Schütte. *Systeme Modaler und Intuitionistischer Logik*, volume 42 of *Ergebnisse der Mathematik und ihrer Grenzgebiete*. Springer, 1968.

[48] R. Statman. Intuitionistic propositional logic is polynomial space complete. *Theoretical Computer Science*, 9:67–72, 1973.

[49] P. A. Subrahmanyam. Towards a framework for dealing with system timing in very high level silicon compilers. In G. Birtwistle and P. Subrahmanyam, editors, *VLSI Specification, Verification, and Synthesis*, pages 159–215. Workshop on Hardware Verification, Kluwer, 1988.

[50] P. Tharau. BinProlog 4.0 User Guide. Departement d'Informatique, Université de Moncton, Canada, September 1995.

[51] A. S. Troelstra. Realizability. In S. R. Buss, editor, *Handbook of Proof Theory*, chapter VI, pages 407–474. Elsevier, 1998.

[52] D. van Dalen. Intuitionistic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume III, chapter 4, pages 225–339. Reidel, 1986.

[53] Gregor von Bochmann. Hardware specification with temporal logic: an example. *IEEE Transactions on Computers*, C-31(3):223–231, March 1982.

[54] D. Weise. Multilevel verification of MOS circuits. *IEEE Transactions on Computer-Aided Design*, 9(4):341–351, April 1990.

[55] G. Winskel. A compositional model of MOS circuits. In G. M. Birtwistle and P. A. Subrahmanyam, editors, *VLSI Specification, Verification and Synthesis*. Kluwer, 1987.

[56] M. Yoeli and J. A. Brzozowski. Ternary simulation of binary gate networks. In J. M. Dunn and G. Epstein, editors, *Modern Uses of Multiple-Valued Logic*, pages 41–50. D. Reidel, 1977.

[57] M. Yoeli and S. Rinon. Application of ternary algebra to the study of static hazards. *Journal of the ACM*, 11:84–97, 1964.

[58] Chaochen Zhou and C. A. R. Hoare. A model for synchronous switching circuits and its theory of correctness. *Formal Methods in System Design*, 1(1):7–28, July 1992.

M. Mendler: Timing analysis of combinational circuits in intuitionistic propositional logic.
Formal Methods in System Design, Vol. 17, Nr. 1, pp. 5-37, 2000.