Chapter 1

PROPERTIES OF CHANGE DIAGRAMS

Uwe Schwiegelshohn
Computer Engineering Institute
University Dortmund
D-44221 Dortmund, Germany
uwe@carla.e-technik.uni-dortmund.de

Lothar Thiele
Computer Engineering and Networks Laboratory (TIK)
Swiss Federal Institute of Technology (ETH) Zurich
CH-8092 Zürich, Switzerland
thiele@tik.ee.ethz.ch

Abstract The paper investigates properties of change diagrams. They are able to model a subclass of concurrent systems, for example asynchronous circuits or timing diagrams. The following results are described: Change diagrams are related to the class of dynamic min-max graphs. Efficient algorithms for timing analysis are derived. Liveness and boundedness properties are investigated.

1. INTRODUCTION

Change diagrams (CD) have been introduced and described in (Varshavsky et al., 1989; Kishinevsky et al., 1991; Kishinevsky et al., 1993). This model is characterized by the following properties, see also (Yakovlev et al., 1996):

- CDs can equally model two types of causality, i.e. AND causality and OR causality, see (Yakovlev et al., 1996). For example, if an event c has two cause events a and b, then event c occurs at most i times if the minimum of occurrences of a and b equals i (AND causality) or if the maximum of occurrences of a and b equals i (joint OR causality).

- CDs cannot directly represent processes with conflicts or choice.

Additional properties and a comparison to other models of computation as well as many examples for their application to the analysis and synthesis of circuits and protocols can be found in (Kishinevsky et al., 1993; Yakovlev, 1992; Yakovlev et al., 1996) and the references therein.

Initially, CDs have been introduced in a labeled and interpreted form in order to directly establish the link to signal transitions. In the present paper, we address the more general unlabeled version of the model.

Definition 1 (Untimed Change Diagram) *An untimed change diagram (CD) is a graph* $G = (V, E)$ *with a marking (or state) function* $d : E \rightarrow \mathcal{Z}$ *where* $d(e)$ *of edge* $e \in E$ *is called the number of tokens on* e *or the marking of* e*. The set of nodes* V *is partitioned into two sets* V^+ *and* V^- $(V = V^+ \cup V^-)$*.*

In order to avoid notational difficulties we assume that each node has at least one incoming edge and that G is connected. However note that all results in this paper can be easily generalized.

Definition 2 (State Transition) *A node* $v \in V$ *is enabled at marking (or state)* d *iff*

$$d(u,v) > 0 \text{ for all } (u,v) \in E \qquad \text{if } v \in V^+$$

$$\text{there exists } (u,v) \in E \text{ with } d(u,v) > 0 \qquad \text{if } v \in V^-$$

An enabled node v *may fire leading to a state transition with*

$$d'(u,v) = d(u,v) - 1 \quad \text{if } (u,v) \in E, u \neq v$$

$$d'(v,u) = d(v,u) + 1 \quad \text{if } (v,u) \in E, u \neq v$$

All others markings remain unchanged in a state transition. The state transition is denoted as $d \xrightarrow{v} d'$*.*

At any moment any edge of G may contain either some negative ($d(e) < 0$) or some positive ($d(e) > 0$) tokens. Note that there is an immediate "recombination" of negative and positive tokens on any edge such that they cannot coexist on the same edge at any time. More complex transitions with concurrent firing of several vertices can always be converted into a sequence of simple transitions. We say that a state or marking d' can be reached from d if there is a sequence of state transitions σ leading from d to d', denoted as $d \xrightarrow{\sigma} d'$.

The well known class of marked graphs is a subclass of change diagrams characterized by $V^- = \emptyset$ and $d(e) \geq 0$ for all edges e.

In Figure 1.1a we show an example from (Yakovlev et al., 1996) which corresponds to a low latency hardware structure with redundancy and buffers of length n. After firing of v_1, v_2, v_3, v_5 and v_6 3 times each, the state as shown in Figure 1.1b is obtained.

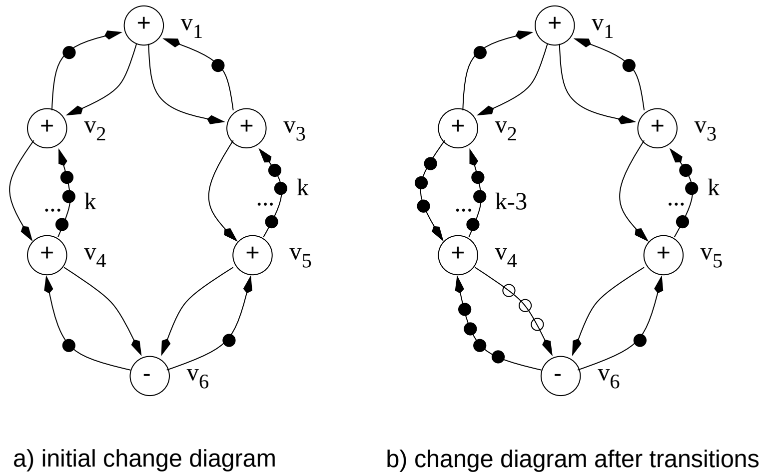

Figure 1.1 Change diagram corresponding to a redundant hardware system. The marking $d(e)$ is either shown as a number attached to e or as filled ($d(e) > 0$) or empty ($d(e) < 0$) circles.

In order to check the correctness of a specification or implementation, it is obviously desirable to derive properties of a given change diagram. In particular, because of the similarity to the model of marked graphs, it can be expected that comparable results can be obtained, see (Desel and Esparza, 1995; Commoner et al., 1971). Several results in this direction are already given in (Kishinevsky et al., 1993).

Until now, no general approach to characterize and determine liveness and boundedness properties of change diagrams is known. A CD is live for an initial marking d if from each reachable marking d' there exists for each node $v \in V$ a reachable marking d'' that enables v. A CD is bounded for an initial marking d if there exists an integer K such that for each reachable marking we have $|d(e)| \leq K$ for all edges $e \in E$.

The paper contains the following new results concerning untimed change diagrams:

- A simple algorithm for deciding the liveness of a change diagram is given.
- The boundedness is characterized and an efficient algorithm is provided.

- Several properties of change diagrams are derived.

In order to evaluate system performance in case of an implementation or to verify timing properties of a system specification, timed change diagrams can be defined as follows:

Definition 3 (Timed Change Diagram) *In a timed change diagram, a weight function $w : E \to \mathcal{Z}$ is associated with the edges. The time of the k^{th} firing of a node $v \in V$ is denoted as $p(v,k) \in \mathcal{Q}$. The firing times are defined by the following recurrence equations:*

$$p(v,k) = \max_{(u,v)\in E} \{p(u, k - d(u,v) + w(u,v))\} \quad \textit{if}\, v \in V^+$$

$$p(v,k) = \min_{(u,v)\in E} \{p(u, k - d(u,v) + w(u,v))\} \quad \textit{if}\, v \in V^-$$

In other words, a node $v \in V^+$ fires at a time t, if on each input edge $(u,v) \in E$ there is at least one token which has been on that edge for at least $w(u,v)$ time units. A node $v \in V^-$ fires, if on at least one input edge $(u,v) \in E$ there is a token which has been on that edge for at least $w(u,v)$ time units. This property can best be described with the unfolding of a change diagram. For each firing of a node $v \in V$ there is a new node in the unfolding. As the resulting graph is acyclic, the time of a firing can simply be calculated solving min- and max-equations for each node in the unfolding. An example is shown in Figure 1.2 where the ordered firing sequence is $((v_1,0),(v_4,0),(v_2,1),(v_3,1),(v_4,1),(v_3,2),(v_1,3),(v_2,4),...)$. It can be seen, that the system consists of two bounded subsystems, i.e. v_1, v_2 and v_3, v_4, which evolve with the different firing periods 3 and 1, respectively. The two subsystems are connected via edges with an unbounded number of token. This behavior is called quasi-periodic in (Schwiegelshohn and Thiele, 1997a; Schwiegelshohn and Thiele, 1997b).

The analysis of timing properties will be carried out by establishing a link between the class of timed change diagrams and dynamic min-max systems, see (Olsder, 1991; Olsder, 1992; Baccelli et al., 1992; Gunawardena, 1993; Gunawardena, 1994; Schwiegelshohn and Thiele, 1997a; Schwiegelshohn and Thiele, 1997b).

In particular, the present paper contains the following new results for the analysis of timed change diagrams:

- The first efficient (pseudo-polynomial) algorithm for the determination of the periods of a timed change diagram is derived.

- Results on the uniqueness of the periods are given.

- The boundedness of change diagrams is characterized and efficient (pseudo-polynomial) algorithms are derived.

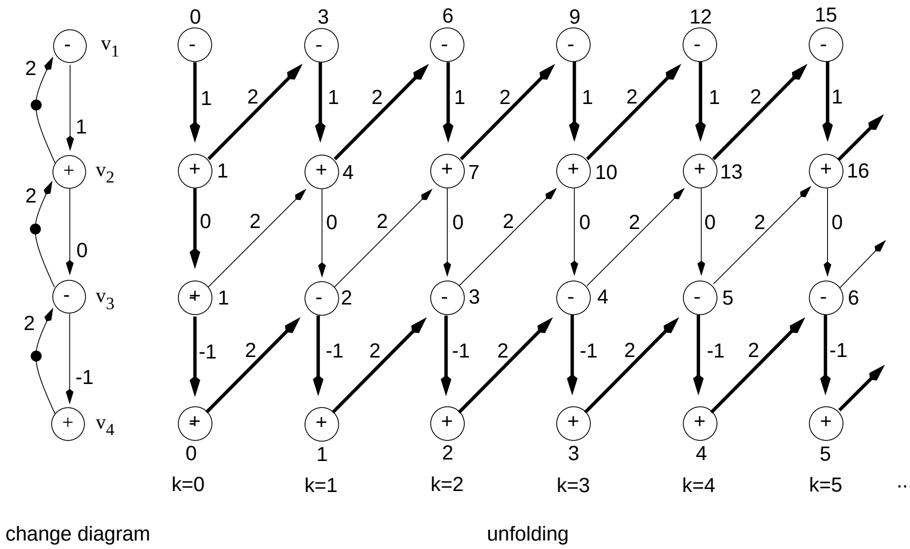

Figure 1.2 A change diagram and the corresponding unfolding. The edges of the change diagram and the unfolding are labeled with $w(e)$. The nodes of the unfolding are labeled with the associated firing time.

2. PROPERTIES OF UNTIMED CHANGE DIAGRAMS

2.1 BASIC PROPERTIES

Marked graphs are closely related to change diagrams. On the other hand, CD have two classes of nodes with different enabling conditions. As a consequence, the markings can be negative both at the initial state or during state evolution. Therefore, some of the simple properties of marked graphs, see e.g. (Desel and Esparza, 1995; Commoner et al., 1971), are no longer valid. Nevertheless, some properties can easily be derived.

Proposition 4 (Cycles) *Let C be a directed cycle of a change diagram with initial state d. For any reachable state d', the sum of tokens in C is constant, i.e.*

$$\sum_{e \in C} d(e) = \sum_{e \in C} d'(e)$$

Sketch of Proof: The proof as known for marked graphs only uses the firing rule and not the enabling conditions. Consequently, the result holds. ■

The next property characterizes firing sequences which reproduce the state of a change diagram.

Proposition 5 (Reproducibility) *Let σ be a sequence of state transitions which reproduces a state, i.e. $d \xrightarrow{\sigma} d$. If the change diagram is connected, then each node $v \in V$ occurs in σ equally often.*

Sketch of Proof: Again, the proof as known for marked graphs uses the firing rules only and not the enabling conditions, see (Desel and Esparza, 1995). Consequently, the result holds. ■

Finally, a simple condition for the liveness of a change diagram can be obtained. But first, a Lemma will be shown which stresses the importance of directed cycles in a change diagram.

Lemma 6 (Positiveness) *A change diagram with initial state d is given. If $\sum_{e \in C} d(e) > 0$ for all directed cycles C, then there exists a firing sequence σ with $d \xrightarrow{\sigma} d'$ such that $d'(e) \geq 0$ for all $e \in E$.*

Sketch of Proof:
The proof is constructive. In each step, the sum of negative markings, i.e. $\sum_{e \in E} \max\{0, -d(e)\}$ is reduced by one.

Let us choose one edge $(u, v) \in E$ with $d(u, v) \leq 0$. If the source node u can be fired, then the sum of negative markings has been reduced by one. If not, it has at least one input edge (w, u) with $d(w, u) \leq 0$. Again, if w can be fired, then the sum of negative markings has been reduced by one. Continuing this process, at least one node can be fired. Otherwise, a cycle with a non-positive sum of markings exists. ■

Consequently, if the sum of tokens is strictly positive in any cycle, then there is a firing sequence which drives the change diagram into a "normalized" form where no marking is negative.

Proposition 7 (Liveness) *A change diagram with initial state d is given. If $\sum_{e \in C} d(e) > 0$ for all directed cycles C, the change diagram is live.*

Sketch of Proof: We have shown that there exists a firing sequence which yields $d(e) \geq 0$ for all edges $e \in E$. Now, if we restrict ourselves to a change diagram which contains nodes $v \in V^+$ only, then the graph is a marked graph and the same proof as e.g. in (Desel and Esparza, 1995) holds.

If a change digram with $V = V^+$ is live, there exists a firing sequence such that any node can fire arbitrarily often. This property still holds if if some nodes change their type from $v \in V^+$ to $v \in V^-$.

It can be proven that a change diagram is live iff there is a firing sequence such that any node can fire arbitrarily often. This proof is based on the persistence of change diagrams. Therefore, the result of the Proposition holds. ∎

Unfortunately, the above result gives a sufficient condition for the liveness of a change diagram only. In fact, Figure 1.3 shows the example of a live CD which violates the condition of the above Proposition. The next section contains also necessary conditions for liveness.

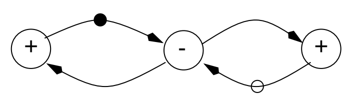

Figure 1.3 A live change diagram with a negative sum of markings in some directed cycle.

2.2 LIVENESS

Is a change diagram live for a given initial state d? We address this problem by presenting a constructive algorithm for its solution. A similar approach as in (Schwiegelshohn and Thiele, 1997b) is used. In particular, an algorithm "live" is defined which is particularly easy to analyze and which constructs a certain firing sequence. Note that we assumed at the beginning of the paper that each node has at least one incoming edge and that G is connected.

The algorithm uses the change diagram G and the initial state d and returns 'true' if G with d is live and 'false' otherwise. If it returns 'false', the graph G_t is a maximal live subgraph of G, i.e. the live subgraph with the maximal number of nodes.

Internally, it uses a 'potential' $p(v) \geq 0$ for any node $v \in V$ which denotes the number of firings of node v in the firing sequence constructed so far. Moreover, it uses an upper bound on the number of markings of any simple path in G:

$$s = \sum_{v \in V} (\max_{(u,v) \in E} \{|d(u,v)|\})$$

This algorithm simulates the change diagram by assuming that a vertex fires whenever it is able to do so. More precisely, a vertex $v \in V$ will fire k-times in a state d with

$$k = \min_{(u,v) \in E} d(u,v) \quad \text{if } v \in V^+$$

```
Boolean Function live(G, d, G_t) {
        in G, d; out G_t
        p(v) = 0 for all v ∈ V;
  loop: do in parallel (for all v_j ∈ V^+) {
            p(v_j) = max{0, min{p(v_i) + d(v_i, v_j) | (v_i, v_j) ∈ E}}; }
        do in parallel (for all v_i ∈ V^-) {
            p(v_j) = max{0, max{p(v_i) + d(v_i, v_j) | (v_i, v_j) ∈ E}}; }
        if (there was no change in any p(v) for all v ∈ V) {
            G_t = ∅; return 'false'; }
        if (p(v) > 0 for all v ∈ V) {  G_t = G; return 'true'; }
        if (there were only changes in p(v), v ∈ V with p(v) > 2s) {
            G_t = subgraph induced by nodes with p(v) > s;
            return 'false'; }
        goto loop;
}
```

Table 1.1 Algorithm *live*

$$k = \max_{(u,v)\in E} d(u,v) \quad \text{if } v \in V^-$$

provided $k > 0$ holds. The resulting sequence of states starting with the initial state d is called a *greedy* sequence. Note that the greedy sequence is equivalent to the sequence of the corresponding time CD with $w(e)) = 0$ for every edge e.

In order to proof the main result of this section, the firing sequences of change diagrams must be considered more closely.

Lemma 8 *A change diagram G with initial state d is live iff there exists a firing sequence which contains each node $v \in V$ at least once.*

Sketch of Proof: Let us suppose that there is no sequence which contains each node at least once. Then there is a node v which cannot be enabled and the change diagram is not live.

Let us suppose that a firing sequence σ exists which contains each node at least once, e.g. $d \xrightarrow{\sigma} d'$. Then one can simply construct sequences which fire each node at least k times for any given k. To this end, a sequence σ' is determined from σ which contains each node exactly once and reproduces d', i.e. $d \xrightarrow{\sigma} d' \xrightarrow{\sigma'} d'$. Obviously, σ' can be applied to d' infinitely often.

As change diagrams are persistent, this proves the liveness. The detailed proof will be contained in the full paper. ■

Now, the main results of this section can be described.

Proposition 9 (Liveness) *Given a change diagram G with initial state d. The change diagram is live iff algorithm* live *returns 'true'.*

Sketch of Proof: At first note that $p(v)$ does not decrease for any node during the execution of the loop. Therefore, the function definitely terminates.

Because of the previous Lemma, the change diagram is live if the algorithm returns 'true'.

Moreover, a change diagram is not live if no node can fire. In addition, if only nodes with $p(v) > s$ fire in an iteration, no node with $p(v) \leq s$ will fire in further iterations as a firing of nodes with $p(v) > s$ cannot lead to an enabling. ■

Up to $|V| \max_{(v_i,v_j)\in E}\{|d(v_i,v_j)|\}$ iterations of Algorithm 1.1 may be necessary before it can be concluded that a min-max Petri net is live. As vertices may fire more than once per iteration, some vertices may be required to fire up to $O(|V| \max_{(v_i,v_j)\in E}\{|d(v_i,v_j)|\}^2$ times. An example for such a case is given in Figure 1.4. In our figures any vertex of V^+ is denoted by a small + sign inside the vertex circle. A similar convention is used for the vertices from V^-. If cycle A consists of k vertices, vertex x will only fire at iteration $kn+2$ of Algorithm *live*. At this time vertex y has already fired $\lceil \frac{kn^2+2n}{2} \rceil$ times.

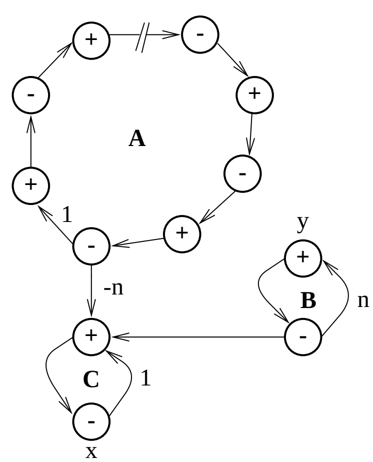

Figure 1.4 Change diagram

Note that several simple improvements of Algorithm 'live' are possible. In addition, the established relationship between the liveness of change diagrams and the static min-max problem also allows the use of more efficient algorithms like those described in (Schwiegelshohn and Thiele, 1997a). Those algorithms determine after k iterations that Cycle A in Figure 1.4 is independent of any other vertex. Then the potential of all vertices of Cycle A is set to infinity, thus causing vertex x to fire far earlier.

2.3 BOUNDEDNESS

Next assume that we are given a live min-max Petri net. Note that in general there are no timing constraints which uniquely determine the fire sequence of vertices. Hence, tokens may accumulate on some edges as the target vertex does not fire. This leads to our second key problem.

Is the number of tokens on any edge of a live min-max Petri net bounded for any firing sequence?

We start to address this question by assuming that the graph of the change diagram is strongly connected.

Proposition 10 *A live change diagram is unbounded if the underlying graph is not strongly connected.*

Sketch of Proof: Let us suppose that the graph is not strongly connected. Then a tree of strongly connected subgraphs can be formed and an unbounded number of tokens will accumulate on some edge if no vertex of a leaf component is fired. ■

But an unbounded accumulation of tokens on some edge may not only be due to the refusal to fire some vertices. In Figure 1.5 a simple live change diagram is given where even the greedy sequence results in an accumulation of positive tokens on edge (x, y) and a corresponding accumulation of negative tokens on edge (y, x) if $n > 2$.

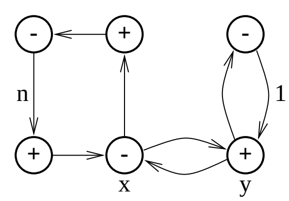

Figure 1.5 Unbounded change diagram

The next lemma gives a necessary and sufficient condition for the existence of sequences with an unbounded number of tokens on some edges.

Lemma 11 (Boundedness) *Assume a live change diagram with initial state d based on a graph $G = (V, E)$. There are sequences causing an accumulation of an unbounded number of tokens on some edge if V can be partitioned into V_1 and V_2 such that*

1. *V_1 is a live change diagram when only edges between nodes of V_1 are considered.*

2. *There are no edges $(u, v) \in E$ with $u \in V_2$ and $v \in V_1 \cap V^+$.*

Sketch of Proof: Let us suppose that there exists a firing sequence which leads to an number of markings $|d'(e)| > K$ with

$$K = \sum_{(u,v)\in E} |d(u, v)|$$

in some edges. These edges will be called unbounded edges. If a firing sequence leads to an unbounded edge, G with initial state d is unbounded, see the proof of correctness for algorithm 'live'. By cutting all unbounded edges the resulting graph is not connected any more. Therefore, we can partition V according to $V = V_1 \cup V_2$ such that V_1 contains all sources of unbounded edges with $d(e) > 0$, all targets of unbounded edges with $d(e) < 0$ and all nodes which are connected to them via bounded edges. The nodes in V_1 can fire arbitrarily often and independent of those in V_2 as the connecting edges are unbounded, and the nodes in V_1 fired already at most K times more often than those in V_2. Edges $(u, v) \in E$ with $v \in V_1 \cap V^+$ cannot exists as otherwise, $v \in V_1$ would not have fired sufficiently often. Therefore the partitioning in the Lemma exists.

Let us suppose that the partitioning exists, then the subgraph induced by nodes V_1 (without edges between V_1 and V_2) is live. There exists a sequence of firings which contains all nodes in V_1 arbitrarily often and nodes in V_2 a finite number of times as there are no edges $(u, v) \in E$ with $v \in V_1 \cap V^+$. This sequence leads to unbounded edges which connect V_1 and V_2. ■

Unfortunately, Lemma 11 by itself does not guarantee an efficient algorithm. Although Algorithm *live* allows the separation of dead vertices from live ones, there remains the problem of selecting those edges which must be deleted to perform the partition.

The following algorithm *bounded* returns 'true' if the CD is bounded and 'false' otherwise. It simply tests the liveness of the change diagram after inhibiting the firing of each of its nodes one after the other.

Proposition 12 *A given connected change diagram G with initial state d is bounded iff algorithm* bounded *returns 'true'.*

```
Boolean Function bounded(G, d) {
        in G, d;
        for each v ∈ V do
                G' = G;
                remove all edges (u, v) from G';
                live(G,d,G_t);
                if ( G_t ≠ ∅ ) return 'false';
        return 'true';
}
```

Table 1.2 Algorithm *bounded*

Sketch of Proof: The algorithm obviously terminates. Let us suppose that *bounded* returns a subgraph G_t. Obviously, there exists a firing sequence which contains all nodes in this subgraph arbitrarily often and nodes not in G_t a bounded number of times. As the change diagram is connected, there is at least one edge connecting a node in G_t and one not in G_t. The marking on this edge would be unbounded.

Now, let us suppose that the change diagram is not bounded. Then according to Lemma 11 there exists a partition $V = V_1 \cup V_2$ where the subgraph induced by V_1 is live. Consequently, if we prevent any of the nodes $v \in V$ from firing, the subgraph induced will still be live, the algorithm 'live' returns $G_t \supseteq V_1$ and Function 'bounded' returns 'false'. ∎

3. PROPERTIES OF TIMED CHANGE DIAGRAMS

The analysis of timed change diagrams, see e.g. Figure 1.2, is closely related to the area of interface timing verification, see (McMillan and Dill, 1992; Walkup and Borriello, 1994; Yen et al., 1995). In these problems, the solution to a set of equations involving min and max operators is required. But caused by the dynamic nature of change diagrams, a much more general problem must be solved here. On the other hand, for timed marked graphs corresponding results are known for a long time, see e.g. (Cunninghame-Green, 1962; Reiter, 1968; Ramamoorthy, 1980; Cohen et al., 1985; Baccelli et al., 1992).

After several attempts to solve the analysis of dynamic min-max problems, see (Olsder, 1991; Olsder, 1993; Baccelli et al., 1992; Gunawardena, 1993; Gunawardena, 1994), the first efficient (pseudo-polynomial) algorithm for the determination of the periods of dynamic min-max systems has been described in (Schwiegelshohn and Thiele, 1997a; Schwiegelshohn and Thiele, 1997b). In addition, a more general class of dynamic min-max systems is considered and results on the uniqueness and existence of their periods are derived.

The results obtained in (Schwiegelshohn and Thiele, 1997a) and (Schwiegelshohn and Thiele, 1997b) hold for the class of timed change diagrams as well. Change diagrams are called static graphs and their unfoldings are called dynamic graphs. For the rest of this paper we suppose that $\sum_{(u,v)\in C} d(e) > 0$ and $\sum_{(u,v)\in C} w(e) > 0$ for all cycles C of the change diagram. As has been shown in the present paper, this property guarantees the liveness of the change diagram.

The following results from (Schwiegelshohn and Thiele, 1997a; Schwiegelshohn and Thiele, 1997b) can be applied to the timing analysis of change diagrams.

Periodic Case Suppose that only periodic firing times of the nodes are considered, i.e.

$$p(v, k+1) = p(v, k) + \lambda \quad \text{for all } v \in V$$

where $\lambda \in \mathcal{Q}$ is called the (common) period of the timed change diagram.

Then either the change diagram has no periodic firing times or the period is unique. Moreover, there is an algorithm which computes the period in pseudo-polynomial time.

The algorithm uses an function similar to 'live' and a binary search to determine the period of the change diagram.

Even if no common period of all nodes exists there are individual periods of the nodes. This property is called quasi-periodicity.

Quasi-Periodic Case Suppose that an individual period is associated with each node of the change diagram, i.e.

$$p(v, k+1) = p(v, k) + \lambda(v) \quad \text{for all } v \in V$$

where $\lambda(v) \in \mathcal{Q}$ is called the period of node $v \in V$. An example for this situation is shown in Figure 1.2. The periods of the nodes satisfy $\lambda(v_1) = \lambda(v_2) = 3$ and $\lambda(v_3) = \lambda(v_4) = 1$.

It can be derived from the results in (Schwiegelshohn and Thiele, 1997a; Schwiegelshohn and Thiele, 1997b) that any timed change diagram has quasi-periodic firing times as already mentioned above. Moreover, the periods $\lambda(v)$ for all $v \in V$ are unique. Again, a pseudo-polynomial algorithm is given to determine these periods.

These results can be applied to determine the period (or cycle time, throughput) of asynchronous circuits which can be modeled with change diagrams. In addition, the throughput of iterative protocols involving AND and OR causality can be calculated.

The results shown in (Schwiegelshohn and Thiele, 1997a) and (Schwiegelshohn and Thiele, 1997b) apply to specific initial (first) firing times of the nodes, namely those which lead to a periodic or quasi-periodic firing. But very often one is interested in the firing times under arbitrary initial conditions. Though it is an open problem whether in this case the firing times become eventually periodic in a steady state, useful bounds on the firing times are derived in the present paper. As a consequence, a strong statement on the boundedness of timed change diagrams can be made.

Proposition 13 (Firing Times) *Given a timed change diagram G with initial state d and weights w.*

Then there exist constants k_1 and k_2 such that the firing times $p(v, k)$ of a node $v \in V$ can be bounded by

$$k\lambda(v) + k_1 \leq p(v, k) \leq k\lambda(v) + k_2$$

As a direct consequence of the above Proposition, the periods $\lambda(v)$ are unique and can be computed in pseudo-polynomial time with the algorithm given in (Schwiegelshohn and Thiele, 1997a; Schwiegelshohn and Thiele, 1997b). Intuitively, the nodes v fire with different 'speeds' determined by $\lambda(v)$.

Finally it is interesting to determine the boundedness of a timed change diagram. Of course, any bounded change diagram is bounded also in its timed version. On the other hand, as the timing restricts the set of possible firing sequences, the converse is not true. For example, a structurally not bounded change diagram may be bounded if it is executed with an appropriate timing.

Considering the example shown in Figure 1.2, it is obvious that the edges connecting the nodes v_2 and v_3 are unbounded. This observation can be generalized as follows.

Proposition 14 (Boundedness) *Given a timed change diagram G with initial state d, weights w and periods $\lambda(v)$ for all $v \in V$. The change diagram is bounded iff all periods are equal.*

Sketch of Proof: The difference of firing times between two nodes u and v satisfies

$$|p(v, k) - p(u, k)| \leq k|\lambda(v) - \lambda(u)| + 2s$$

If all periods are equal, the firing times of all nodes differ by a constant only. Therefore, all connecting edges have a bounded number of token.

After some calculations it can also be shown that at some time t the difference between the number of firings of u and v is larger than

$$k_1|\lambda(v) - \lambda(u)|t + k_2$$

for some constants $k_1 > 0$ and k_2. If there are nodes with different periods in G, then there exists at least one edge connecting two nodes with different periods. Remember that we consider connected change diagrams only. The difference in the number of firings grows with t which leads to an unbounded number of tokens on this edge. ■

4. CONCLUDING REMARKS

The paper presents new results on the dynamic behavior of timed and untimed change diagrams. Major results concern the liveness and boundedness of timed and untimed CDs as well as a characterization of the firing times. These results can be applied to analyze properties of a certain class of concurrent systems.

It is an open problem whether timed change diagrams eventually show a periodic behavior independent of the initial firing times.

References

Baccelli, F., Cohen, G., Olsder, G., and Quadrat, J.-P. (1992). *Synchronization and Linearity*. John Wiley, Sons, New York.

Cohen, G., Dubois, D., Quadrat, J. P., and Viot, M. (1985). A linear-system-theoretic view of discrete-event processes and its use for performance evaluation in manufacturing. *IEEE Transactions on Automatic Control*, AC-30, No. 3:210–220.

Commoner, F., Holt, A., Even, S., and Pnueli, A. (1971). Marked directed graphs. *Journla of Computer and System Sciences*, 5:511–523.

Cunninghame-Green, R. (1962). Describing industrial processes and approximating their steady-state behaviour. *Opt. Res. Quart.*, 13:95–100.

Desel, J. and Esparza, J. (1995). *Free Choice Petri Nets*. Number 40 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press.

Gunawardena, J. (1993). Timing analysis of digital circuits and the theory of min-max functions. In *TAU'93, ACM International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems.*

Gunawardena, J. (1994). Min-max functions. Technical Report to be published in Discrete Event Dynamic Systems, Department of Computer Science, Stanford University, Stanford, CA 94305, USA.

Kishinevsky, M., Kondratyev, A., and Taubin, A. (1991). Formal methods for self-timed design. In *Proc. EDAC 91*, pages 197–201.

Kishinevsky, M., Kondratyev, A., Taubin, A., and Varshavsky, V. (1993). *Concurrent Hardware: The Theory and Practice of Self-Timed Design*. John Wiley and Sons.

McMillan, K. and Dill, D. (1992). Algorithms for interface timing verification. In *IEEE Int. Conference on Computer Design*, pages 48–51.

Olsder, G. (1991). Eigenvalues of dynamic max–min systems. *Discrete Event Dynamic Systems: Theory and Applications*, pages 1:177–207.

Olsder, G. (1992). About difference equations, algebras and discrete events. *Topics in Engineering Mathematics*, pages 121–150.

Olsder, G. (1993). Analyse de systèmes min-max. Technical Report 1904, Institut National de Recherche en Informatique et en Automatique.

Ramamoorthy, C. (1980). Performance evaluation of asynchronous concurrent systems using Petri nets. *IEEE Transactions on Software Engineering*, pages 440–449.

Reiter, R. (1968). Scheduling parallel computations. *Journal of the Association for Computing Machinery*, 15, No. 4:590–599.

Schwiegelshohn, U. and Thiele, L. (1997a). Dynamic min-max problems. Technical Report 23, Swiss Federal Institute of Technology (ETH Zurich), TIK (Computer Engineering and Networks Laboratory). accepted for publication in Journal on Discrete Event Dynamic Systems.

Schwiegelshohn, U. and Thiele, L. (1997b). Periodic and non-periodic mion-max equations. In *International Colloquium on Automata, Languages, and Programming*, ICALP 97, pages 379–389.

Varshavsky, V., Kishinevsky, M., Kondratyev, A., Rosenblum, L., and Taubin, A. (1989). Models for specification and analysis of processes in asynchronous circuits. *Soviet Journal of Computer and System Sciences*, 26(2):61–76.

Walkup, E. and Borriello, G. (1994). Interface timing verification with application to synthesis. In *IEEE/ACM Design Automation Conference*, pages 106–112.

Yakovlev, A. (1992). On limitations and extensions of stg model for designing asynchronous control circuits. In *Proceedings of ICCD 92*, Cambridge, MA.

Yakovlev, A., Kishinevsky, M., Kondratyev, A., Lavagno, L., and Pietkiewicz-Koutny, M. (1996). On the models for asynchronous circuit behaviour with OR causality. *Formal Methods in System Design*, 9:189–233.

Yen, T., Ishii, A., Casavant, A., and Wolf, W. (1995). Efficient algorithms for interface timing verification. Technical report, Princeton University.