Non-preemptive scheduling algorithms and schedulability conditions for real-time systems with precedence and latency constraints

Liliana Cucu, Yves Sorel

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Non-preemptive scheduling algorithms and schedulability conditions for real-time systems with precedence and latency constraints

Liliana Cucu — Yves Sorel

N° 5403

Decembre 2004

Thème COM

ISSN 0249-6399 ISRN INRIA/RR--5403--FR+ENG

Non-preemptive scheduling algorithms and schedulability conditions for real-time systems with precedence and latency constraints

Liliana Cucu *, Yves Sorel *

Thème COM — Systèmes communicants
Projet Aoste

Rapport de recherche n° 5403 — Decembre 2004 — 33 pages

Abstract: First we justify our concern in latency constraints for real-time systems with precedence constraints. We evoke the model based on graph theory used to state and solve the problem of non-preemptive scheduling systems with precedence and latency constraints. Because latency constraints involve pairs of operations related to the graph which models the precedence constraints, we define three relations between pairs of operations, relations which determine the way of scheduling the operations. We give a necessary and sufficient condition of schedulability for the particular case of systems with precedence and latency constraints, where all pairs, on which latency constraints are imposed, are in one of these relations. Finally, we give a general schedulability condition for real-time systems with precedence and latency contraints and we propose a scheduling algorithm which is proved optimal.

Key-words: real-time, graphs, scheduling, schedulability, algorithms, complexity, mono-processor

* INRIA Rocquencourt

Ordonnancement non-préemptif et conditions d'ordonnançabilité pour systèmes temps réel avec contraintes de précédences et de latences

Résumé : On commence par justifier notre intêret pour les contraintes de latences pour des systèmes temps réel avec contraintes de précédences. On rapelle notre modèle utilisant les graphes afin de poser et de résoudre le problème d'ordonnancement non-préemptif de systèmes avec contraintes de précédences et de latences. Puisque les contraintes de latence sont imposées sur des paires d'opérations du graphe, qui définit les contraintes de précédence, on définit trois relations entre paires d'opérations, relations qui détermine la manière dont les opérations vont être ordonnancées. On donne une condition nécessaire et sufisante d'ordonnançabilité pour le cas des sytèmes avec contraintes de précédences et de latences où les paires avec contraintes de latences sont en une de trois relations. Finalement, on donne une condition générale d'ordonnançabilité et on propose un algorithme d'ordonnancement qui est prouvé optimal.

Mots-clés : temps réel, graphes, ordonnancement, ordonnançabilité, algorithmes, complexité, monoprocesseur

1 Introduction

Real-time systems are first of all reactive systems [1], that is to each event arriving in the system through sensors corresponds an output event produced through actuators to the environment in reaction to this input event. The arrival of input events is indefinitly repeated, and an output event is obtained from one or several input events after several operations on which precedence constraints may be imposed. For example in the case of brake control in a car, when the driver presses the brake pedal, the wheels must stop before a fixed delay, corresponding to the execution of control laws relating "pedal-press" event and "wheels-stop" event. We call this constraint a *latency* constraint. This type of delay was already used by the real-time community, but either the models took into account only one global latency constraint [2, 3], or the delay may be imposed only on sensors and actuators [4]. This paper generalises these results, allowing to have several latency constraints which are not imposed only on sensors and actuators, but also on the operations which are necessary to obtain output events from input events. For example, it is possible to specify the brake system of a car (figure 1) by a graph where $PRESS$ represents a sensor giving the information which indicates that the driver presses the brake pedal, $SPEED$ represents a sensor giving the wheel speed, $DEAC$ represents the deactivation of the speed regulator, $ESTIM$ represents the estimation of the pressure on the brake pedal, ABS represents the anti-locking brake system (ABS) and $BRAKE$ represents the actuator giving the pressure on the brakes wheels. We may need to impose a delay between $PRESS$ and $BRAKE$, and another delay between $DEAC$ and $BRAKE$.

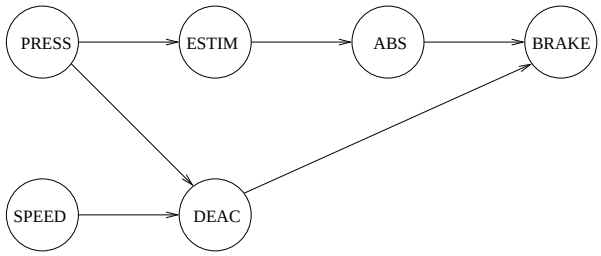

Figure 1: Example of brake control

In this paper we study the non-preemptive case and we plan to extend subsequently the results to the preemptive case evaluating, accurately, its overhead.

The next section gives some notations and existing results for systems with precedence constraints. Section 3 presents our model given in [5]. In this section we define the latency constraint. In the following section we give a schedulability condition for real-time systems with precedence and latency constraints. Section 5 presents an optimal scheduling algorithm. The last section concludes and proposes future work.

2 Notations and existing results

In order to clearly distinguish the specification level from the implementation level, we use the term *operation* instead of the commonly used "task" too closely related to the implementation level, and in turn we use the term *operator* instead of "processor" or "machine".

For an operation A we may specify a computation time C_A, a deadline D_A (defined from the start of the schedule) and a start time s_A. Also for pairs of operations, we may specify precedence constraints that may be represented by a directed acyclic graph where the vertices are the operations and the edges are the precedence constraints.

We evoke now some main results on systems with precedence constraints and deadlines, in the case of one operator. For the non-preemptive problem with precedence constraints, minimising a monotone function which allows to define deadlines, Lawler gives the optimal "first-to-last-rule" [6]. Also for the non-preemptive problem with precedence and deadlines, Blazewicz gives a polynomial solution [7]. For the preemptive problem with precedence constraints and deadlines, there are results on the schedulability of a system [8], [9], [10] and [11].

In this paper we give a schedulability condition for real-time systems with precedence and latency constraints using the model proposed in [5].

3 Model for systems with precedence and latency constraints

In our model a system with precedence constraints is defined by a infinite directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of operations and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ the set of edges which represents the precedence constraints between operations. Because a real-time system is reactive, we model the system by an infinite graph, issued from an infinite repetition of a finite subgraph called "pattern" [3]. If one consider only the pattern itself, according to its partial order the first operations are called inputs and the last operations are called outputs. These operations correspond to respectively sensors and actuators. The edges between operations belonging to the same pattern are called intra-pattern and the edges between operations belonging to different patterns are called extra-pattern. For example in figure 2, the edge from A to B is an intra-pattern edge and the edge from D belonging to the i^{th} pattern to A belonging to the $(i+1)^{th}$ pattern is an extra-pattern. The infinite repetition of the pattern of the graph induces an infinite repetition of all the operations.

Definition 1 *[12] A directed path from A_1 to A_n is an alternating sequence $A_1(A_1, A_2)$ $A_2 \ldots (A_{n-1}, A_n)A_n$ of vertices and distinct edges, where $A_i \in \mathcal{V}$ and $(A_i, A_{i+1}) \in \mathcal{E}$, $\forall i \in \{1, 2, \ldots, n\}$.*

For the sake of simplicity and because we use only directed graphs, from now on we will use the term path instead of directed path.

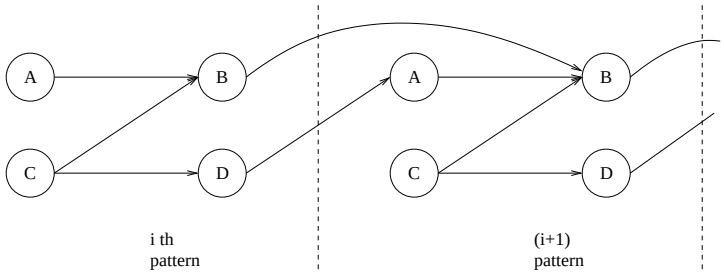

Figure 2: System of operations defined by a infinite directed graph

We denote by $\mathcal{P}$ the set of all paths of graph $\mathcal{G}$ and we say that $P(A,B) \in \mathcal{P}$, if there is at least a path from A to B. If $\nexists P(A,B) \in \mathcal{P}$, then there is no path from A to B. If there is, at least, a path from A to B, we denote by $\mathcal{M}(A,B)$ the set of all operations belonging to all the paths from A to B, $\mathcal{M}(A,B) = \{C \in \mathcal{V}$ with $P(A,C) \in \mathcal{P}$ and $P(C,B) \in \mathcal{P}\} \bigcup \{A,B\}$. In order to illustrate the latter notation we give the following example.

Example 1 *Let be the graph given by figure 3. The set $\mathcal{M}(A,G) = \{A,B,D,C,G\}$ is not empty because there is at least a path from A to G and we have $P(A,G) \in \mathcal{P}$. For example, from A to G there are four paths: $A(A,D)D(D,G)G$, $A(A,B)B(B,D)D(D,G)G$, $A(A,D)D(D,C)C(C,G)G$ and $A(A,B)B(B,D)D(D,C)C(C,G)G$. On the contrary the set $\mathcal{M}(C,E)$ is empty because there is no path from C to E.*

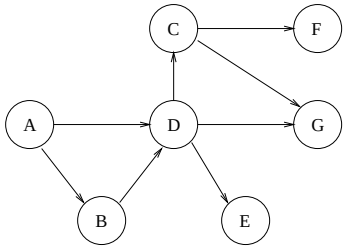

Figure 3: Set of operations belonging to all paths from A to G

Definition 2 *For a system of operations, a schedule S is a total ordered set of start times of all the operations $S = \{s_A \in \mathbb{N}, A \in \mathcal{V}\}$ such that the precedence constraints are satisfied. We denote by $\mathcal{S}$ the set of all schedules.*

Remark 1 *An operation is available when all its predecessors are scheduled. We consider schedules without idle time, i.e. we schedule available operations at the end of the last scheduled operation and we have $\forall A \in \mathcal{V}, \exists B \in \mathcal{V}$ such that $s_A + C_A = s_B$.*

The edges of a system of operations define a partial order on the set of operations. There are several total orders which may be obtained from this partial order. Because a schedule is a total order on the set of operations with a start time associated to each operation, then we may have several schedules for the same system of operations.

Remark 2 *The directed pair of operations* $(A, B) \in \mathcal{E}$ *implies that* B *must be scheduled, only if* A *was already scheduled, and we have* $s_A + C_A \leq s_B$*, where* $s_A, s_B, C_B \in \mathbb{N}^+$*.*

Definition 3 *[5] on a pair of operations* (A, B) *with* $\exists P(A, B) \in \mathcal{P}$ *and* A *and* B *belonging to the same pattern, we say that a latency constraint* $L(A, B)$ *is imposed if the operations are scheduled such that* $s_B + C_B - s_A \leq L_{AB}$*,* $L_{AB} \in \mathbb{N}^+$*. Let* $\mathcal{L}$ *be the set of all latency constraints imposed for a system.*

Remark 3 *The condition imposing to* A *and* B *to belong to the same pattern is not restrictive. If we want to impose a latency constraint on two operations which do not belong to the same pattern, then it is sufficient to define a new pattern. Indeed because the value of a latency constraint is finite (otherwise it does not define a real-time constraint) we will never obtain an infinite pattern. The figure 4 gives such an example. We want to define the latency constraint* $L(C, B)$ *with* C *and* B *belonging to two different patterns. Therefore we choose a new pattern containing two old patterns.*

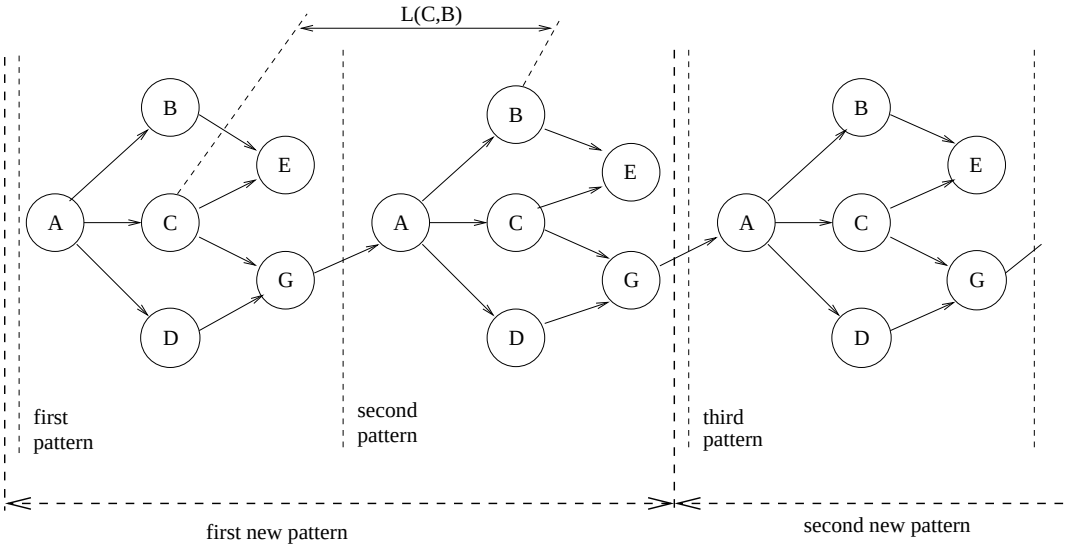

Figure 4: Latency imposed on several patterns

Definition 4 *if there is a schedule for a system of operations which satisfies all the latency constraints then the system is called schedulable.*

The definition 3 generalises the notion of latency constraint for two operations, which are not necessarily input and output operations.

4 Schedulability condition

We give a necessary and sufficient condition for a system with precedence and latency constraints to be schedulable. Because the graph is a pattern infinitely repeated and the latency constraints are imposed on operations belonging to the same pattern, studying only the pattern is enough to obtain the schedulability condition.

The latency constraints are imposed on pairs of operations related by a path which imposes a total order on the operations belonging to this path. So, if we want to schedule a system of operations with n latency constraints, we study the existence of paths between pairs of operations with latency constraints. Consequently in this section first we present relations between two pairs of operations relatively to scheduling. For these pairs on which latency constraints are imposed, we present a schedulability condition.

For the sake of simplicity, we call "an operation belonging to a pair (X, Y)" an operation belonging to at least one of the paths from X to Y. Also, we say that "a pair (X, Y) contains an operation Z" when Z belongs to the pair (X, Y).

For some pair of operations (X, Y) belonging to a system and a schedule of this system, we call "schedule length" the time between the start of an operation X and the end of an operation Y. We denote this time by $\delta_{XY}(S) = s_Y + C_Y - s_X$, where $s_X, s_Y \in S$. Without any loss of generality we consider that X is always scheduled before Y.

Because we consider that the available operations are scheduled at the end of the last scheduled operation (see remark 1) we have $\delta_{XY}(S) = \sum_{\forall C, s_X \leq s_C \leq s_Y} C_C$.

Theorem 1 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph defining a system and $L(A, B)$ a latency constraint where $A, B \in \mathcal{V}$. If the system is schedulable then the sum of the execution times of operations belonging to $\mathcal{M}(A, B)$ is smaller or equal to $L_{A,B}$.*

Proof If the system is schedulable then there is at least a schedule S satisfying all constraints of the system. So, for a schedule S we have

$$\delta_{AB}(S) \leq L_{A,B} \tag{1}$$

Because of the partial order, the operations belonging to $\mathcal{M}(A, B)$ must be scheduled between A and B and we have $\sum_{C \in \mathcal{M}(A,B)} C_C \leq \delta_{AB}(S) \leq L_{A,B}$. So $\sum_{C \in \mathcal{M}(A,B)} C_C \leq L_{A,B}$. The theorem is proved $\square$

The theorem 1 gives a necessary condition for a latency constraint to be satisfied. By imposing this condition we guarantee that we can always schedule between the first and

the last operation of a latency constraint at least the operations belonging to this latency constraint.

If between these operations there is no operation belonging to another pair on which a latency constraint is imposed then this condition is also sufficient (see theorem 2). This is possible because all schedules, satisfying the partial order defined by a graph for which we obtain the smallest time between the first and the last operation of the latency constraint, contain only operations belonging to the pair on which the latency constraint is imposed. All these schedules lead to the same set of operations scheduled between the first operation and the last operation of the latency constraint, set that we call "minimal set corresponding to the latency constraint".

On the contrary if between these operations there is at least one operation belonging to another pair on which a latency constraint is imposed then the condition given by theorem 1 is only necessary (see theorem 5). This is possible because all schedules, satisfying the partial order defined by a graph for which we obtain the smallest time between the first and the last operation of the latency constraint, may contain also operations belonging to another pair on which a latency constraint is imposed. All these schedules lead to different sets of operations scheduled between the first operation and the last operation of the latency constraint, sets that we call "minimal sets corresponding to the latency constraint".

The first consequence of these interpretations of theorem 1 is the corollary 1 concerning the case of a system with only one latency constraint. So we are in the case of a latency constraint imposed on a pair which does not contain any operation belonging to another pair on which a latency constraint is imposed and the condition given by theorem 1 is necessary and sufficient.

Corollary 1 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph defining a system and (A, B) the only latency constraint of the system. The system is schedulable if and only if the sum of execution times of operations belonging to $\mathcal{M}(A, B)$ is smaller or equal to $L_{A,B}$.*

Proof If we have only one latency constraint then studying the schedulability of the system amounts to consider all schedules satisfying the partial order and containing between the first and the last operation of the latency constraint only operations belonging to the pair on which the latency constraint is imposed. Indeed for these schedules we obtain the smallest time between the first and the last operation of the latency. Consequently it is sufficient to compare the sum of execution times of operations belonging to the pair to value of the latency constraint. The corollary is proved $\boxdot$

Example 2 *This example illustrates the case of one latency constraint. For the graph given in figure 5 we consider the latency constraint $L(A, D)$. We consider all execution times equal to 1. The minimal set $\{A, B, C, E, D\}$ contains between the first and the last operation of the latency constraint only operations belonging to the pair on which the latency constraint is imposed. The operation F does not belong to this set. The system is schedulable if and only if the value of the latency constraint is greater than 5.*

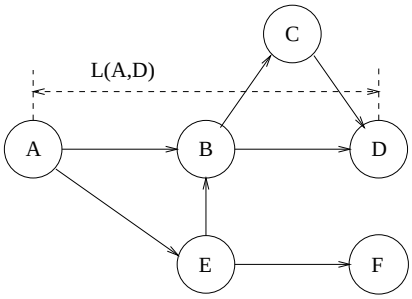

Figure 5: Graph illustrating the remark 1 concerning one latency constraint

Example 3 *This example illustrates the case of several latency constraints. For the graph given in figure 6 we consider the latency constraints* $L(A,C)$ *and* $L(J,L)$*. Because each pair, on which a latency constraint is imposed, does not contain operations belonging to another pair then each minimal set corresponding to these constraints contains only their operations. In this case the minimal set corresponding to each latency constraint is unique. For the latency constraint* $L(A,C)$ *we have the minimal set* $\{A,B,C\}$ *and for the latency constraint* $L(J,L)$ *the minimal set* $\{J,K,L\}$*.*

For the same graph, we consider now the latency constraints $L(D,I)$ *and* $L(J,L)$*. Because each pair, on which a latency constraint is imposed, contains operations belonging to another pair then each minimal set corresponding to these constraints contains also operations belonging to the other constraint. The minimal sets corresponding to these constraints are not unique. Because the precedence constraints* (F,L) *and* (K,H) *impose to* K *and* F *to be scheduled before* L *and* H *then either* F *must be scheduled between operations belonging to the other pair* (J,L)*, either* K *must be scheduled between operations belonging to the other pair* (D,I)*. We have the two following possibilities:*

- *either for the latency constraint* $L(D,I)$ *the minimal set is* $\{D,F,H,I,J,K\}$ *and for the latency constraint* $L(J,L)$ *the minimal set is* $\{J,K,L,H,I\}$*,*
- *either for the latency constraint* $L(D,I)$ *the minimal set is* $\{D,F,H,I,L\}$ *and for the latency constraint* $L(J,L)$ *the minimal set is* $\{J,K,L,D,F\}$*.*

Thus, studying the schedulability of a system amounts to find out for each latency constraint the minimal set, i.e. the operations which might be scheduled at the exterior of a latency constraint and which will increase unnecessarily the time between the first and the last operation of a latency constraint if they were scheduled at the interior of the latency constraint. By exterior of a latency constraint we understand, for some schedule, the operations scheduled before the first operation or after the last operation of the latency. By interior of a latency constraint we understand, for some schedule, the operations scheduled after the first operation and before the last operation of the latency.

4.1 Relations between two pairs of operations relatively to scheduling

We consider two pairs of operations and we study the case where they are connected or not by paths. The paths which connect two pairs are those who start from an operation belonging to one of the pairs to an operation belonging to the other pair.

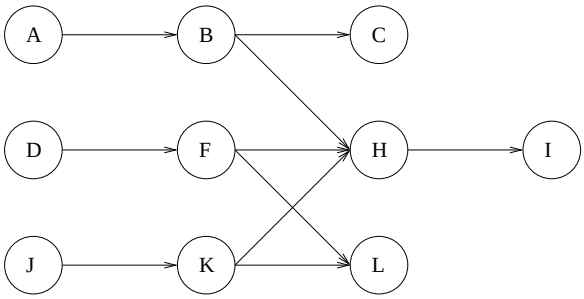

Figure 6: Example of relations between pairs of operations

Thus, for two pairs of operations, we have 2^2 possible combinations covered by the three following cases:

1. there is no path from an operation belonging to one of the paths between the considered pair, to an operation belonging to one of the paths between the other pair (for example, pairs (A, C) and (J, L) of figure 6). This relation is called $||$;
2. there is at least a path from one pair to the other one, and there is no path from the latter pair to the first one, or vice versa (for example, pairs (A, C) and (D, I) of figure 6). This relation is called Z. Note that it covers two cases;
3. there is at least a path from one pair to the other one and vice versa (for example, pairs (D, I) and (J, L) of figure 6). This relation is called X.

In the case of relation $||$ (see figure 6), no precedence constraint imposes to an operation belonging to $\mathcal{M}(A, C)$ to be scheduled before or after all operations belonging to $\mathcal{M}(J, L)$. For example, the schedules with A, B and C scheduled before J, K and L or vice versa satisfy the partial order.

In the case of relation Z (see figure 6), the precedence constraints impose to operations belonging to $\mathcal{M}(A, B)$ to be scheduled before operation belonging to $\mathcal{M}(H, I)$. However, they do not impose any order between operations belonging to $\mathcal{M}(B, C)\backslash\{B\}$ and operations belonging to $\mathcal{M}(D, H)\backslash\{H\}$. For example A, B and C may be scheduled before all operations belonging to $\mathcal{M}(D, I)$, and similarly D, F, H and I may be scheduled before all operations belonging to $\mathcal{M}(A, C)$.

In the case of relation X (see figure 6), the precedence constraints impose to operations belonging to $\mathcal{M}(D,F)$ to be scheduled before operations belonging to $\mathcal{M}(L,L)$ and to operations belonging to $\mathcal{M}(H,I)$ to be scheduled before operations belonging to $\mathcal{M}(J,K)$. However they do not impose any order between operations belonging to $\mathcal{M}(D,F)$ and operations belonging to $\mathcal{M}(J,K)$. For example D and F may be scheduled before or after J and K, and H and I must be scheduled after J and K, and vice versa.

By taking into account the relations between pairs of operations, we study the schedulability condition for latency constraints imposed on pairs in relation Z and/or in relation ||. We study the schedulability condition for latency constraints imposed on pairs in relation X. Finally, we give a general schedulability condition for latency constraints imposed on pairs in relation ||, Z and X.

4.2 Relations || and Z

We study the schedulability of systems with latency constraints imposed on pairs in relation Z and/or in relation ||. For example in figure 7 we have three latency constraints $L(A,C,)$, $L(D,I)$ and $L(J,L)$ imposed on pairs in relation Z and/or in relation ||.

Only the relation Z imposes a partial order on pairs whereas the relation || does not impose any order. For example the partial order imposes only to A and to B to be scheduled before H and I.

The schedulability studies search among all possible schedules those leading to minimal sets corresponding to latencies. The lemma 1 allows to find those schedules and lemma 2 proves that for pairs in relation || and in relation Z those minimal sets are identical. Consequently, even if relatively to the partial order the two cases are not identical, relatively to the schedulability, they are. This implies that we may give an unique schedulability condition for the two relations. This condition is given by theorem 2 which ends this subsection.

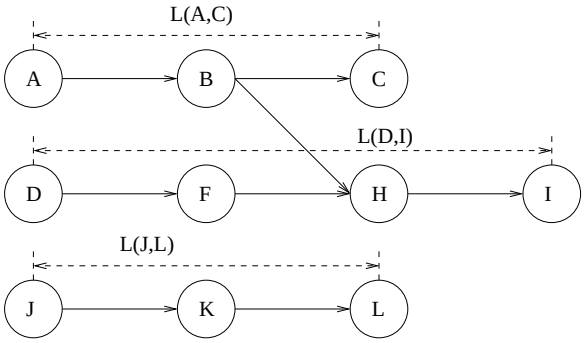

Figure 7: Latency constraints

Lemma 1 *Let $A, B \in \mathcal{V}$ be two operations belonging to a system given by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, such that $P(A, B) \in \mathcal{P}$. We denote by $S \in \mathcal{S}$ a schedule such that $\forall H \notin \mathcal{M}(A, B)$, either we have $s_H + C_H < s_A + C_A$, or we have $s_B + C_B < s_H + C_H$. If $\mathcal{S}'$ is the set of the schedules S' such that $\exists H \notin \mathcal{M}(A, B)$ with $s'_A + C_A < s'_H + C_H < s'_B + C_B$, then $\delta_{AB}(S) \leq \delta_{AB}(S')$, $\forall S' \in \mathcal{S}'$.*

Proof Because S is a schedule such that $\forall H \notin \mathcal{M}(A, B)$, either we have $s_H + C_H < s_A + C_A$, or we have $s_B + C_B < s_H + C_H$, only operations belonging to $\mathcal{M}(A, B)$ are scheduled from s_A to s_B and we have

$$\delta_{AB}(S) = \sum_{I \in \mathcal{M}(A,B)} C_I \tag{2}$$

For some schedule $S' \in \mathcal{S}'$, let $N = \{H \in \mathcal{V}$ such that $H \notin \mathcal{M}(A, B)$ and $s'_A + C_A < s'_H + C_H < s'_B + C_B\}$. We have

$$\delta_{AB}(S') = \sum_{I \in \mathcal{M}(A,B)} C_I + \sum_{H \in N} C_H, \forall S' \in \mathcal{S}' \tag{3}$$

By subtracting the equations (2) and (3), we obtain $\delta_{AB}(S') - \delta_{AB}(S) = \sum_{H \in N} C_H \geq 0, \forall S' \in \mathcal{S}$. Finally we have $\delta_{AB}(S) \leq \delta_{AB}(S')$, $\forall S' \in \mathcal{S}'$. The lemma is proved □

Lemma 2 *For four operations A, B, C and $D \in \mathcal{V}$, if (A, B) || (C, D) or if (A, B) Z (C, D) then we have $\mathcal{M}(A, B) \bigcap \mathcal{M}(C, D) = \emptyset$.*

Proof We prove by contradiction that if (A, B) Z (C, D) then $\mathcal{M}(A, B) \bigcap \mathcal{M}(C, D) = \emptyset$.

We suppose that $\exists F \in \mathcal{M}(A, B) \bigcap \mathcal{M}(C, D)$. Because $F \in \mathcal{M}(A, B)$, respectively $F \in \mathcal{M}(C, D)$, we have

$$P(A, F) \in \mathcal{P} \tag{4}$$

and

$$P(F, B) \in \mathcal{P} \tag{5}$$

respectively

$$P(C, F) \in \mathcal{P} \tag{6}$$

and

$$P(F, D) \in \mathcal{P} \tag{7}$$

From (4) and (7), respectively, from (5) and (6), we obtain $P(A, D) \in \mathcal{P}$ and $P(C, B) \in \mathcal{P}$. This is in contradiction with the definition of relation Z which says that there is at least one path from a pair to the other pair and there is no path from the last pair to the first pair. We have a contradiction so the supposition that there is $F \in \mathcal{M}(A, B) \bigcap \mathcal{M}(C, D)$ is not true and we have $\mathcal{M}(A, B) \bigcap \mathcal{M}(C, D) = \emptyset$.

The proof for the case of relation || is similar to case of relation X. The lemma is proved □

The lemma 2 proves that a latency constraint $L(A,B)$ in relation || or in relation Z with another latency constraint $L(C,D)$ does not contain any operation belonging to this latter constraint ($\mathcal{M}(A,B) \bigcap \mathcal{M}(C,D) = \emptyset$). This implies that each minimal set corresponding to a latency constraint contains only operations belonging to this constraint, so for the two cases, the minimal set corresponding to each latency constraint is identical.

Theorem 2 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph defining a system and each two pairs, on which latency constraints are imposed, are either in relation ||, either in Z. A schedule satisfies all the latency constraints if and only if $\sum_{H \in \mathcal{M}(A,B)} C_H \leq L_{AB}$, for all A and B having a latency constraint.*

Proof The lemma 1 shows that, for two operations A and B related by a path, scheduling only operations which belong to $\mathcal{M}(A,B)$ between A to B will decrease the value of $\delta_{AB}(.)$. This means that if a schedule, in which only operations which belong to $\mathcal{M}(A,B)$ are scheduled between A to B, do not satisfy a latency constraint L_{AB}, then no other schedule will. Therefore and because $\mathcal{M}(A,B) \bigcap \mathcal{M}(C,D) = \emptyset$, for all pairs which are in relation || or in relation Z (lemma 1 and lemma 2), from now on, we will consider only schedules of this type.

A schedule S satisfies all the latency constraints if and only if $\delta_{AB} \leq L_{AB}$, $\forall A$ and $\forall B$ having a latency constraint (definition 3). Because $\forall H \notin \mathcal{M}(A,B)$, either we have $s_H + C_H < s_A + C_A$, or we have $s_B + C_B < s_H + C_H$, then a schedule S satisfies all the latency constraints if and only if $\sum_{H \in \mathcal{M}(A,B)} C_H \leq L_{AB}$, for all A and B having a latency constraint. The theorem is proved □

Remark 4 *When the latency constraints are in relation || or in relation Z, the minimal set corresponding to a latency constraint is equal to the set of operations belonging to this latency constraint. The first part of example 3 illustrates this kind of set.*

Example 4 *This example illustrates the schedulability condition for a system of operations with latency constraints in relation || or in relation Z. For the graph given in figure 8 we consider the latency constraints $L(A,C)$, $L(D,F)$, $L(H,I)$ and, respectively, $L(J,L)$ imposed on pairs (A,C), (D,F), (H,I) and, respectively, (J,L). All latency constraints are equal to 3 and $C_A = 1, \forall A \in \mathcal{V}$. We notice that all pairs are either in relation ||, either in relation Z. The schedulability condition implies that latencies must be greater than the sum of execution times of operations. We suppose that this condition is satisfied so there is at least one schedule which satisfies every constraint of the system. Figure 9 gives such a schedule.*

4.3 Relation X

In order to present the results concerning systems with latency constraints imposed on pairs in relation X, we introduce two new notations.

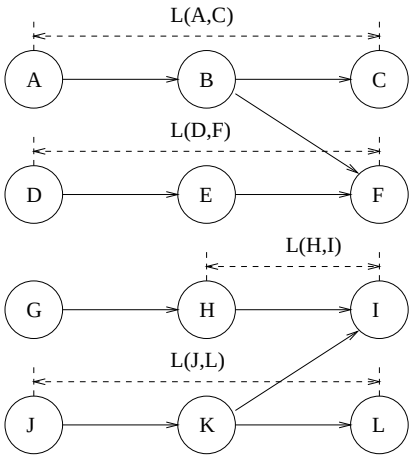

Figure 8: Systems of operations latency constraints in relation || or in relation Z

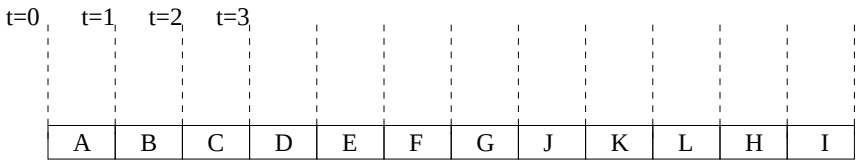

Figure 9: Schedule satisfying all constraint of the system

Let $(A,B),(C,D) \in \mathcal{V} \times \mathcal{V}$ with $P(A,B), P(C,D) \in \mathcal{P}$ be some pairs in relation X. Each pair has operations which must be scheduled before operations belonging to the other pair. Like in the previous case of pairs in relation || and/or in relation Z, the schedulability studies for pairs in relation X are based on minimal sets corresponding to the latency constraints.

For two pairs in relation X, ft denotes set of operations which are first on all paths between the first operation of a pair and those operations, comparatively to the existence of at least a path from one pair to these operations.

We denote by $ft_{CD}(A,B)$ the set of operations $E \in \mathcal{M}(A,B)$ for which $\exists F \in \mathcal{M}(C,D)$ such that $P(F,E) \in \mathcal{P}$ and $\nexists G \in \mathcal{M}(A,E)$ for which $\exists F \in \mathcal{M}(C,D)$ such that $P(F,G) \in \mathcal{P}$. This means that $ft_{CD}(A,B)$ is the set of operations E belonging to one of the paths from A to B such that there is a path to E from one of the operations belonging to one of the paths from C to D, and there is no operation G belonging to one of the paths from A to E such that there is a path to G from one of the operations belonging to one of the paths from C to D. We denote by $\Gamma^{-}_{CD}(A,B)$ the set of predecessors of $ft_{CD}(A,B)$, i.e. the set of operations $E \in \mathcal{V}$ for which there is an operation $F \in ft_{CD}(A,B)$ such that $(E,F) \in \mathcal{E}$.

For two pairs in relation X, lt denotes set of operations which are last on all paths between those operations and the last operation of a pair and, comparatively to the existence of at least a path from these operations to the pair.

We denote by $lt_{CD}(A,B)$ the set of operations $E \in \mathcal{M}(A,B)$ for which $\exists F \in \mathcal{M}(C,D)$ such that $P(E,F) \in \mathcal{P}$ and $\nexists G \in \mathcal{M}(E,B)$ for which $\exists F \in \mathcal{M}(C,D)$ such that $P(G,F) \in \mathcal{P}$. This means that $lt_{CD}(A,B)$ is the set of operations E belonging to one of the paths from A to B such that there is a path from E to one of the operations belonging to one of the paths from C to D, and there is no operation G belonging to one of the paths from E to B such that there is a path from G to one of the operations belonging to one of the paths from C to D. We denote by $\Gamma^{+}_{CD}(A,B)$ the set of successors of $lt_{CD}(A,B)$, i.e. the set of operations $E \in \mathcal{V}$ for which there is an operation $F \in lt_{CD}(A,B)$ such that $(F,E) \in \mathcal{E}$.

Example 5 *In order to illustrate these notations we give the following example. In figure 10, for* (A,B) X (C,D)*, we have* $ft_{CD}(A,B) = \{F,G\}$*,* $lt_{CD}(A,B) = \{G\}$*,* $ft_{AB}(C,D) = \{G\}$ *and* $lt_{AB}(C,D) = \{G,I\}$*.*

The schedulability studies is based again on minimal set belonging to a latency constraint L_1 which must be scheduled before the last operation of another latency constraint L_2 with L_1 X L_2. We study in a schedule the position of operations belonging to L_1 which only increase the schedule length between the first operation and the last operation of L_2 without modifying the schedule length between the first operation and the last operation of L_1, once the first operations or the last operations were scheduled.

For example, in figure 11, we consider that L_1 is a latency constraint on pair (A,B) and that L_2 is a latency constraint on pair (C,D). If we schedule A before C, then for all positions of operation E in the schedule, the schedule length between A and B stays identical. On the contrary the position of E is important for the other pair (C,D). If we schedule E after C we only increase the schedule length between C and D.

Similarly, in figure 12 we consider that L_1 is the latency constraint on pair (A,B) and that L_2 is latency constraint on pair (C,D). If we schedule B before D, then the position of

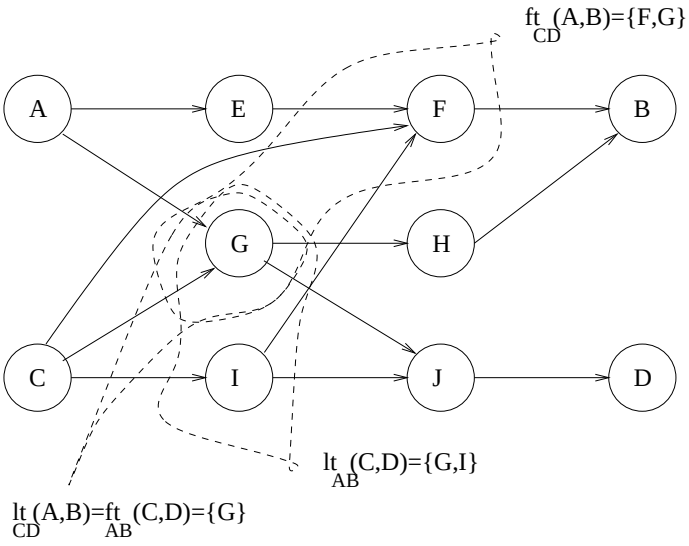

Figure 10: Example for ft and lt

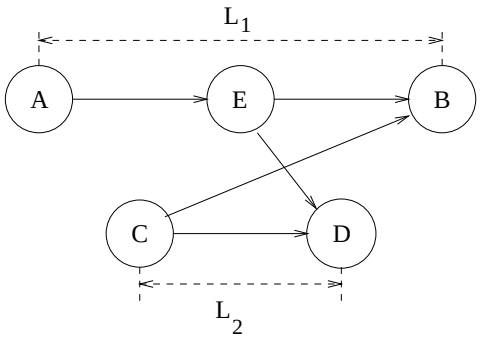

Figure 11: Operations only increasing the schedule length which are at the beginning of a latency constraint

operation F in the schedule is not important for the pair (C, D) because the schedule length time between C and D stays identical. On the contrary the position of F is important for the other pair (A, B) because if we schedule F before B we increase the schedule length between A and B.

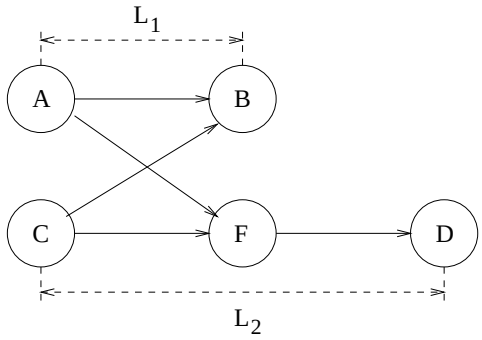

Figure 12: Operations only increasing the schedule length which are at the end of a latency constraint

Theorem 3 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph defining a system and there are two latency constraints imposed on pairs (A, B) and (C, D), where (A, B) X (C, D) and $L_{AB} = \sum_{H \in \mathcal{M}(\mathcal{A},\mathcal{B})} C_H$. The system is schedulable if and only if $L_{CD} \geq \sum_{H \in \mathcal{M}(A,B) \bigcup \mathcal{M}(C,D)} C_H$.*

Proof Because $L_{AB} = \sum_{H \in \mathcal{M}(\mathcal{A},\mathcal{B})} C_H$ then we may schedule between the operation A and the operation B only operations belonging to $\mathcal{M}(A, B)$. So no operation belonging to $\mathcal{M}(C, D)$ and not belonging to $\mathcal{M}(A, B)$ can be scheduled between A and B.

Consequently, when C and D do not belong to $\mathcal{M}(A, B)$ these operations must be scheduled before A and after B, which implies that all operations belonging to $\mathcal{M}(A, B)$ are scheduled between C and D. And then, we have $L_{CD} \geq \sum_{H \in \mathcal{M}(A,B) \bigcup \mathcal{M}(C,D)} C_H$. The theorem is proved □

Remark 5 *In theorem 3 we do not take into account the trivial case where C and D belong to $\mathcal{M}(A, B)$. Indeed if C and D belong to $\mathcal{M}(A, B)$, then all operations belonging to $\mathcal{M}(C, D)$ belong also to $\mathcal{M}(A, B)$. So, the latency constraint imposed on (A, B) must be at least greater or equal to the sum of execution times of operations belonging to $\mathcal{M}(A, B)$.*

Remark 6 *For the case studied by theorem 3 the minimal set of the latency constraint (A, B) is equal to the set of operations belonging to the latency constraint. For the same case, the minimal set of the latency constraint (C, D) is equal to the set of operations belonging to the two latency constraints. For example in figure 6 if we consider the latency constraint $L(D, I)$ is equal to the sum of execution times of operations belonging to $\mathcal{M}(D, I)$ then the*

minimal set of the constraint (D, I) *is equal to* $\{D, F, H, I\}$ *and the minimal set of the constraint* (J, L) *is equal to* $\{J, K, L, D, F, H, I\}$.

The two properties concerning the position of the first and the last operations of two constraints are proved by lemma 3 and lemma 4. The theorem 5 which follows the two lemma uses it for the case where all latency constraints are in relation X.

Lemma 3 *We have a system given by a graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *such that all pairs, on which latency constraints are imposed, are in relation* X. *For* (A, B) X (C, D) *and some schedule* $S \in \mathcal{S}$ *with* $s_A \leq s_C$, *if all operations* $E \in \bigcup_{H \in \Gamma^-_{CD}(A,B)} \mathcal{M}(A, H)$ *are scheduled between* s_A *and* s_C, *then* $\delta_{AB}(S)$ *is not modified compared with the case when at least an operation* $E \in \bigcup_{H \in \Gamma^-_{CD}(A,B)} \mathcal{M}(A, H)$ *is scheduled after* s_C. *Also, if at least an operation* $E \in \bigcup_{H \in \Gamma^-_{CD}(A,B)} \mathcal{M}(A, H)$ *is scheduled after* s_C, *then* $\delta_{CD}(S)$ *is increased compared with the case when all operations* $E \in \bigcup_{H \in \Gamma^-_{CD}(A,B)} \mathcal{M}(A, H)$ *are scheduled between* s_A *and* s_C.

Proof Because $E \in \bigcup_{H \in \Gamma^-_{CD}(A,B)} \mathcal{M}(A, H) \subset \mathcal{M}(A, B)$, E is, always, scheduled between s_A and s_B, so $\delta_{AB}(S)$ will, always, contain the computation time of E and the position of E in the schedule does not modify $\delta_{AB}(S)$.

Because $E \in \bigcup_{H \in \Gamma^-_{CD}(A,B)} \mathcal{M}(A, H)$, there is no operation $F \in \mathcal{M}(C, D)$ which must be scheduled before E. So, E may be scheduled before C. If we schedule any operation $E \in \bigcup_{H \in \Gamma^-_{CD}(A,B)} \mathcal{M}(A, H)$ after C, it is obvious that, for some schedule S, $\delta_{CD}(S)$ is increased with the sum of computation times of these operations E. The lemma is proved ⊡

Lemma 4 *We have a system given by a graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *such that all pairs, on which latency constraints are imposed, are in relation* X. *For* (A, B) X (C, D) *and some schedule* $S \in \mathcal{S}$ *with* $s_D \leq s_B$, *if all operations* $E \in \bigcup_{H \in \Gamma^+_{CD}(A,B)} \mathcal{M}(H, B)$ *are scheduled between* s_D *and* s_B, *then* $\delta_{AB}(S)$ *is not modified compared with the case when at least an operation* $E \in \bigcup_{H \in \Gamma^+_{CD}(A,B)} \mathcal{M}(H, B)$ *is scheduled before* s_D. *Also, if at least an operation* $E \in \bigcup_{H \in \Gamma^+_{CD}(A,B)} \mathcal{M}(H, B)$ *is scheduled before* s_D, *then* $\delta_{CD}(S)$ *is increased compared with the case when all operations* $E \in \bigcup_{H \in \Gamma^+_{CD}(A,B)} \mathcal{M}(H, B)$ *are scheduled between* s_D *and* s_B.

Proof Because $E \in \bigcup_{H \in \Gamma^+_{CD}(A,B)} \mathcal{M}(H,B) \subset \mathcal{M}(A,B)$, E is, always, scheduled between s_A and s_B, so $\delta_{AB}(S)$ will, always, contain the computation time of E and the position of E in the schedule does not modify $\delta_{AB}(S)$. Because $E \in \bigcup_{H \in \Gamma^+_{CD}(A,B)} \mathcal{M}(H,B)$, there is no operation $F \in \mathcal{M}(C,D)$ which must be scheduled after E. So, D may scheduled before all operations $E \in \bigcup_{H \in \Gamma^+_{CD}(A,B)} \mathcal{M}(H,B)$. If we schedule any operation $E \in \bigcup_{H \in \Gamma^+_{CD}(A,B)} \mathcal{M}(H,B)$ before D, it is obvious that, for some schedule S, $\delta_{CD}(S)$ is increased with the sum of computation times of these operations E. The lemma is proved ⊡

Theorem 4 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph defining a graph and there are two latency constraints imposed on the pairs (A,B) and (C,D), where (A,B) X (C,D). The two latency constraints are not equal to the sum of execution times of operations belonging to the constraint. The system is schedulable if and only if*

$$\begin{cases} L_{AB} \geq \sum_{H \in \mathcal{M}(A,B)} C_H + \sum_{H \in \bigcup_{E \in ft_{AB}(C,D)} \mathcal{M}(E,D)} C_H \\ L_{CD} \geq \sum_{H \in \mathcal{M}(C,D)} C_H + \sum_{H \in \bigcup_{E \in lt_{CD}(A,B)} \mathcal{M}(A,E)} C_H \end{cases} \tag{8}$$

or

$$\begin{cases} L_{CD} \geq \sum_{H \in \mathcal{M}(C,D)} C_H + \sum_{H \in \bigcup_{E \in ft_{CD}(A,B)} \mathcal{M}(E,B)} C_H \\ L_{AB} \geq \sum_{H \in \mathcal{M}(A,B)} C_H + \sum_{H \in \bigcup_{E \in lt_{AB}(C,D)} \mathcal{M}(C,E)} C_H \end{cases} \tag{9}$$

Proof The proof is made by double implication.

First, we prove that if the system is schedulable then 8 or 9 are satisfied. Because the system is schedulable then there is at least one schedule S satisfying all constraints. For this schedule, A is scheduled before or after C.

Because we are in the case of latency constraints which are not equal to the sum of execution times of operations belonging to the constraint, when A is scheduled before C, B is scheduled before D (see figure 13).

Using lemma 3, because A is scheduled before C then all operations $E \in \bigcup_{H \in \Gamma^-_{CD}(A,B)} \mathcal{M}(A,H)$ are scheduled between s_A and s_C. We obtain

$$L_{AB} \geq \sum_{H \in \mathcal{M}(A,B)} C_H + \sum_{H \in \bigcup_{E \in ft_{AB}(C,D)} \mathcal{M}(E,D)} C_H \tag{10}$$

Using lemma 4, because B is scheduled before D then all operations $E \in \bigcup_{H \in \Gamma^+_{AB}(C,D)} \mathcal{M}(H,D)$ are scheduled before s_D. We obtain

$$L_{CD} \geq \sum_{H \in \mathcal{M}(C,D)} C_H + \sum_{H \in \bigcup_{E \in lt_{CD}(A,B)} \mathcal{M}(A,E)} C_H \quad (11)$$

From 10 and 11, we obtain that 8 is satisfied.

Because we are in the case of latency constraints which are not equal to the sum of execution times of operations belonging to the constraint, when A is scheduled after C, B is scheduled after D (see figure 14) and the proof follows the same principles as for the first case in order to prove that 9 is satisfied. The direct implication is proved.

We prove now that if 8 or 9 is satisfied then the system is schedulable. The idea of this proof is to build a schedule which satisfies the latency constraints. First, we suppose that 8 is satisfied. We consider the schedule given in figure 13 with all operations $E \in \bigcup_{H \in \Gamma^+_{AB}(C,D)} \mathcal{M}(H,D)$ scheduled before s_D and all operations $E \in \bigcup_{H \in \Gamma^-_{CD}(A,B)} \mathcal{M}(A,H)$ scheduled between s_A and s_C. Because of 8 this schedule satisfies the latency constraints.

We suppose that 9 is satisfied. The proof follows the same principles as before using the schedule given in figure 14 in order to prove that the system is schedulable. The theorem is proved □

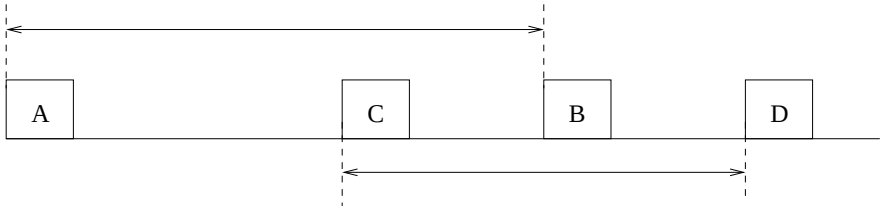

Figure 13: Schedule illustrating theorem 4

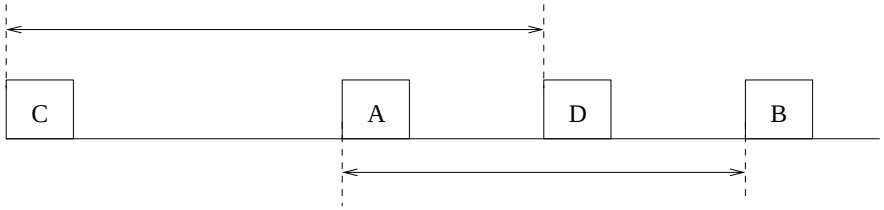

Figure 14: Schedule illustrating theorem 4

Remark 7 *for the case studied by theorem 4 the minimal sets corresponding to the two latency constraints do not contain only operations belonging to the latency constraint but also operations belonging to the other latency constraint. The example 3 illustrates such sets.*

Theorem 5 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph defining a system and all pairs, on which latency constraints are imposed, are in relation* X*. The system is schedulable if and only if, for each pair (A, B) on which a latency constraint is imposed, and which is in relation* X *with n pairs $(C_i, D_i), \forall i\{1, 2, \ldots, n\}$, on which latency constraints are imposed, we have*

$$L_{AB} \geq \sum_{H \in \mathcal{W}} C_H \tag{12}$$

where:

$$\mathcal{W} = \mathcal{M}(A,B) \bigcup \bigcup_{i \in \{1,\ldots,j\}; E \in lt_{AB}(C_i,D_i)} \mathcal{M}(C_i,E) \bigcup \bigcup_{i \in \{j+1,\ldots,k\}; E \in ft_{AB}(C_i,D_i)} \mathcal{M}(E,D_i) \bigcup \bigcup_{i \in \{k+1,\ldots,m\}} \mathcal{M}(C_i,D_i),$$

where $j < k < m < n$ are the natural numbers for which the following relations are satisfied:

$$L_{C_iD_i} \geq \sum_{H \in \{\mathcal{M}(C_i,D_i) \bigcup \bigcup_{E \in ft_{C_iD_i}(AB)} \mathcal{M}(E,B)\}} C_H, \forall i \in \{1, \ldots, j\}$$

$$L_{C_iD_i} \geq \sum_{H \in \{\mathcal{M}(C_i,D_i) \bigcup \bigcup_{E \in lt_{C_iD_i}(A,B)} \mathcal{M}(A,E)\}} C_H, \forall i \in \{j+1, \ldots, k\}$$

$$L_{C_iD_i} \geq \sum_{H \in \mathcal{M}(C_i,D_i)} C_H, \forall i \in \{k+1, \ldots, m\} \text{ and } L_{C_iD_i} \geq \sum_{H \in \{\mathcal{M}(C_i,D_i) \bigcup \mathcal{M}(A,B)\}} C_H, \forall i \in \{m+1, \ldots, n\}.$$

Proof The proof is made by double implication. First, we prove the direct implication. We have that the system is schedulable which means there is a schedule $S \in \mathcal{S}$ which satisfies all the latency constraints: $s_B - s_A + C_B \leq L_{AB}$, $\forall L(A,B) \in \mathcal{S}$. But $s_B - s_A + C_B$ is the sum of computation times of operations which are scheduled from s_A to s_B. Because of the partial order, the set of the operations scheduled from s_A to s_B contains at least some operations belonging to $\mathcal{M}(C_i, D_i), \forall i \in \{1, 2, \ldots, n\}$. The operations of $\mathcal{M}(C_i, D_i)$ which are scheduled from s_A to s_B are given by scheduling order of operations C_i, D_i, A, B, $\forall i \in \{1, 2, \ldots, n\}$ in schedule S. Therefore we have to study all the possible orders for these two pairs of operations, i.e. six orders such that A is always scheduled before B, and C before D. Also, among the six orders, because (A, B) is in relation X with (C_i, D_i), the orders $s_A + C_A \leq s_B + C_B \leq s_{C_i} + C_{C_i} \leq s_{D_i} + C_{D_i}$ and $s_{C_i} + C_{C_i} \leq s_{D_i} + C_{D_i} \leq s_A + C_A \leq s_B + C_B$ are forbidden. It remains 4 possible cases for scheduling the pairs (A, B) and $(C_i, D_i), \forall i \in \{1, 2, \ldots, n\}$:

1. $s_A + C_A \leq s_{C_i} + C_{C_i} \leq s_B + C_B \leq s_{D_i} + C_{D_i}, \forall i \in \{1, 2, \ldots, j\}$;
2. $s_{C_i} + C_{C_i} \leq s_A + C_A \leq s_{D_i} + C_{D_i} \leq s_B + C_B, \forall i \in \{j, \ldots, k\}$;
3. $s_{C_i} + C_{C_i} \leq s_A + C_A \leq s_B + C_B \leq s_{D_i} + C_{D_i}, \forall i \in \{k+1, \ldots, m\}$;
4. $s_A + C_A \leq s_{C_i} + C_{C_i} \leq s_{D_i} + C_{D_i} \leq s_B + C_B, \forall i \in \{m+1, \ldots, n\}$;

Because of lemma 3, the case 1 implies that all operations belonging to $\mathcal{M}(A, E), E \in \Gamma^-_{C_i,D_i}(AB)$,
$\forall i \in \{1, 2, \ldots, j\}$ are scheduled from A to C. Consequently, because C_i is scheduled before B, the rest of operations belonging to $\mathcal{M}(A, B)$, i.e. operations belonging to $\mathcal{M}(E, B), E \in ft_{C_iD_i}(A, B)$ are scheduled between C_i and $D_i, \forall i \in \{1, 2, \ldots, j\}$.

Because of lemma 4, the case 1 implies that all operations belonging to $\mathcal{M}(E, D_i), E \in \Gamma^+_{AB}(C_i, D_i)$,
$\forall i \in \{1, 2, \ldots, j\}$ are scheduled from B to D_i. Consequently, because C_i is scheduled before B, the rest of operations belonging to $\mathcal{M}(C_i, D_i)$, i.e. operations belonging to $\mathcal{M}(C_i, E), E \in lt_{AB}(C_i, D_i)$ are scheduled between A and B.

Because of lemma 3, the case 2 implies that all operations belonging to $\mathcal{M}(C_i, E), E \in \Gamma^-_{AB}(C_i, D_i)$,
$\forall i \in \{j+1, \ldots, k\}$ are scheduled from C_i to A. Consequently, because A is scheduled before D_i, the rest of operations belonging to $\mathcal{M}(C_i, D_i)$, i.e. operations belonging to $\mathcal{M}(E, D_i), E \in ft_{AB}(C_i, D_i)$ are scheduled between A and B.

Because of lemma 4, the case 2 implies that all operations belonging to $\mathcal{M}(E, B), E \in \Gamma^+_{C_iD_i}(A, B)$,
$\forall i \in \{j+1, \ldots, k\}$ are scheduled from C_i to D_i. Consequently, because A is scheduled before D_i, the rest of operations belonging to $\mathcal{M}(A, B)$, i.e. operations belonging to $\mathcal{M}(A, E), E \in lt_{C_i,D_i}(A, B)$, are scheduled between C and D.

The case 3 implies that all operations belonging to $\mathcal{M}(C_i, D_i), \forall i \in \{k+1, \ldots, m\}$ are scheduled from A to B. The case 3 does not allow to say if operations belonging to $\mathcal{M}(A, B)$ are scheduled from C_i to D_i.

The case 4 implies that all operations belonging to $\mathcal{M}(A, B)$ are scheduled from C_i to $D_i, \forall i \in \{m+1, \ldots, n\}$. The case 4 does not allow to say if operations belonging to $\mathcal{M}(C_i, D_i)$ are scheduled from A to B.

In conclusion, the set of operations which are scheduled from A to B contains, at least, the operations belonging to

$$\mathcal{W} = \mathcal{M}(A, B) \bigcup \bigcup_{i \in \{1,\ldots,j\}; E \in lt_{AB}(C_i,D_i)} \mathcal{M}(C_i, E) \bigcup \bigcup_{i \in \{j+1,\ldots,k\}; E \in ft_{AB}(C_i,D_i)} \mathcal{M}(E, D_i) \bigcup \bigcup_{i \in \{k+1,\ldots,\}} \mathcal{M}(C_i, D_i), \text{ where:}$$

$j, k, m \in \mathbb{N}$ are the natural numbers for which the following relations are satisfied:

$$s_{D_i} - s_{C_i} + C_{D_i} \geq \sum_{H \in \{\mathcal{M}(C_i,D_i) \bigcup \bigcup_{E \in ft_{C_iD_i}(A,B)} \mathcal{M}(E,B)\}} \mathrm{C}_H, \forall i \in \{1, \ldots, j\},$$

$$s_{D_i} - s_{C_i} + C_{D_i} \geq \sum_{H \in \{\mathcal{M}(C_i,D_i) \bigcup \bigcup_{E \in lt_{C_i D_i}(A,B)\}} \mathcal{M}(A,E)\}} C_H, \forall i \in \{j+1,\ldots,k\},$$

$$s_{D_i} - s_{C_i} + C_{D_i} \leq \sum_{H \in \mathcal{M}(C_i,D_i)} C_H, \forall i \in \{k+1,\ldots,m\} \text{ and}$$

$$s_{D_i} - s_{C_i} + C_{D_i} \geq \sum_{H \in \{\mathcal{M}(C_i,D_i) \bigcup \bigcup_{E \in \mathcal{M}(A,B)} C_H} , \forall i \in \{m+1,\ldots,n\}.$$

Using $L(A,B) \geq s_{D_i} - s_{C_i} + C_{D_i}$, we obtain inequality (12). The direct implication is proved.

We prove the inverse implication. We have inequality (12) satisfied for each pair (A,B) on which latency constraint is imposed. This means that we may schedule from A to B the operations belonging to
$\mathcal{W} = \mathcal{M}(A,B) \bigcup \bigcup_{i \in \{1,\ldots,j\}; E \in lt_{AB}(C_i,D_i)} \mathcal{M}(C_i,E) \bigcup \bigcup_{i \in \{j+1,\ldots,k\}; E \in ft_{AB}(C_i,D_i)} \mathcal{M}(E,D_i) \bigcup$
$\bigcup_{i \in \{k+1,\ldots,\}} \mathcal{M}(C_i,D_i)$. The set $\mathcal{W}$ contains all the operations belonging to $\mathcal{M}(C_i,D_i)$, $\forall i \in \{1,\ldots,n\}$ which must be scheduled from A to B in order to satisfy the partial order. Using inequality (12) for each pair (A,B), we obtain a schedule which satisfies the partial order. Because latency constraints are, already, satisfied by inequality (12), the system is schedulable. The inverse implication is proved and the theorem is proved ⊡

Remark 8 *The theorem 5 presents schedulability conditions which allows to obtain minimal sets corresponding to the latency constraints. Because building these sets for a general case is probably a NP-hard problem, we will consider from now on only cases where each latency constraint is in relation* X *with at most one latency constraint. For this case in the next section we give a polynomial scheduling algorithm.*

We give an example of a system with latency constraints in relation X.

Example 6 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the graph given in figure 15 with the execution times $C_A = 1, \forall A \in \mathcal{V}$ and the latency constraints $L_{AC} = 3$, $L_{DF} = 9$ and $L_{HJ} = 5$. We have that (A,C)* X *(D,F) and (D,F)* X *(H,J). The schedulability conditions are satisfied and the figure 16 gives a schedule which satisfies all constraints.*

4.4 General schedulability condition

Giving a general schedulability condition amounts to prove that there is a schedule which satisfies all the latency constraints if and only if each latency constraint in relation ||, Z or X with other latency constraints is satisfied. Previously, we obtained, separately, schedulability condition for systems with all pairs in relation || and Z and schedulability condition for

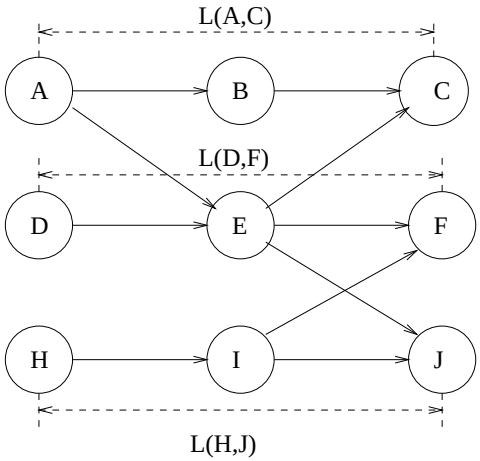

Figure 15: System with latency constraints in relation X

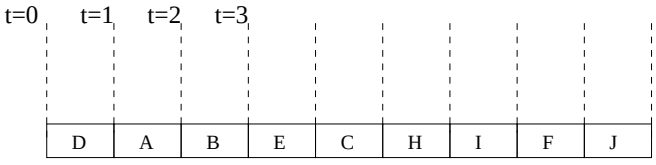

Figure 16: Schedule satisfying all constraints

systems with all pairs in relation X. From now on, we will study only the schedulability condition of systems with all latency constraint $L_{X,Y}$ greater than the sum of computation times of operations belonging to $\mathcal{M}(X,Y)$ because the systems which do not satisfy this latter condition are, obviously, not schedulable. The following theorem gives the general schedulability condition.

Because all latency constraint $L_{X,Y}$ are greater than the sum of computation times of operations belonging to $\mathcal{M}(X,Y)$. Because of lemma 1, we have that a schedule, in which between s_A and s_B there are no operation belonging to pairs in relation || or Z with (A,B), does not satisfy the latency constraints with all pairs in relation || and X on which latency constraints are imposed, we proved that a system with all the latency constraints well-defined is, always, schedulable. This remaining true for the general case, the general schedulability condition will concerns only pairs of operations which are in relation.

Theorem 6 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph defining a system and $\mathcal{L}$ a set of latency constraints imposed on pairs of operations. There is a schedule satisfying all the latency constraints if and only if inequality (12) is satisfied for each pair (A,B) in relation X with n pairs $(C_i, D_i), \forall i \in \{1, 2, \ldots n\}$.*

Proof The proof is made by double implication and follows the same principles as the proof of theorem 5. The direct implication assumes that the system is schedulable and proves that inequality (12) is satisfied, taking into account for each pair on which a latency constraint is imposed all pairs such that this pair is in relation ||, Z or X. The inverse implication builds a schedule which satisfies all latency constraints, using inequality (12) ⊡

Remark 9 *For a system with n latency constraints, the number of inequalities of type(12) which must be verified is at most $n(n-1)$.*

Example 7 *For the graph of figure 17, we have all computation times equal to 1, $C_A = 1, \forall A \in \mathcal{V}$. The following latency constraints are imposed: $L_{AB} = 7$, $L_{CD} = 12$, $L_{MF} = 3$ and $L_{OR} = 4$. Note that (A,B)X(C,D), (A,B)Z(M,F), (C,D)X(M,F) and (O,R) is in relation || with all the other latency constraints. First, note that all latency constraints $L_{X,Y}$ are greater than the sum of computation times of operations belonging to $\mathcal{M}(X,Y)$. Then applying theorem 6, the system is schedulable if and only if the following relations are satisfied:*

$$L_{AB} \geq \sum_{X \in \{\mathcal{M}(A,B) \bigcup \bigcup_{Y \in lt_{AB}(C,D)} \mathcal{M}(C,Y)\}} C_X$$

$$L_{CD} \geq \sum_{X \in \{\mathcal{M}(C,D) \bigcup \bigcup_{Y \in ft_{CD}(A,B)} \mathcal{M}(Y,B) \bigcup \mathcal{M}(M,F)\}} C_X$$

$$L_{MF} \geq \sum_{X \in \mathcal{M}(M,F)} C_X$$

We obtain:

$$\begin{cases} L_{AB} \geq C_A + C_G + C_H + C_I + C_B + C_C + C_K \\ L_{CD} \geq C_C + C_I + C_J + C_K + C_L + C_D + C_M + \\ C_N + C_F + C_H + C_B \\ L_{MF} \geq C_M + C_N + C_F \end{cases}$$

Because all three inequations are verified, the system is schedulable. Indeed the schedule $S = \{s_O \leq s_P \leq s_R \leq s_E \leq s_A \leq s_G \leq s_C \leq s_K \leq s_I \leq s_H \leq s_B \leq s_M \leq s_N \leq s_F \leq s_J \leq s_L \leq s_D\}$ *satisfies all the latency constraints. In the following section we give an algorithm which allows to obtain a schedule which satisfies all the latency constraints for the case where each latency constraint is in relation* X *with at most one latency constraint.*

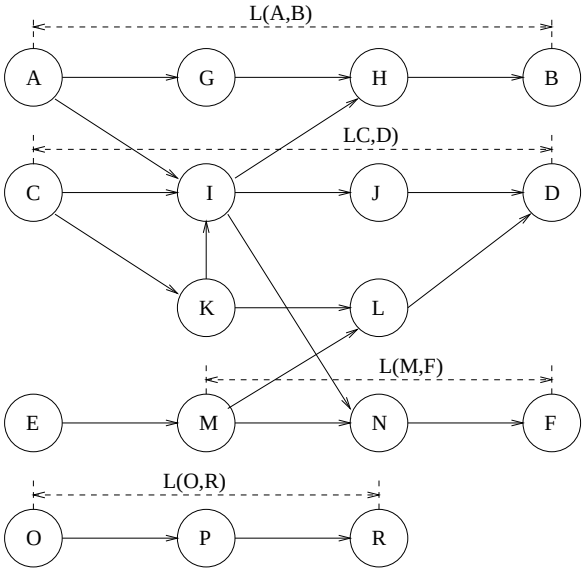

Figure 17: Example for general schedulability condition

5 Optimal scheduling algorithm

This section presents an optimal scheduling algorithm for the case where each latency constraint is in relation X with at most one latency constraint. This algorithm builds schedules satisfying the schedulability condition obtained in the previous section.

A scheduling algorithm transforms the partial order defined by the graph in a total order (one of possibles total orders) which satisfies the latency constraints. In our case the

scheduling algorithm 2 is applied only to the pattern and it uses algorithm 1 that we call "marking operations algorithm". This algorithm "anticipates" which operations are the most important for the latency constraints.

We denote by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ the oriented acyclic graph-pattern, where $\mathcal{V}$ is the set of operations and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges and by $\mathcal{D}(A) = \{B \in \mathcal{V}$ such that $\exists$ at least a path from A to $B\}$. We denote by $\mathcal{W}$ the working-set used during the algorithm.

The purpose of the marking algorithm is to associate to each operation A a mark which indicates if an operation C belonging to a pair (B, C) with a latency constraint will be scheduled after A. This is possible thanks to the existence of at least one path between A and C and to the non-existence of any path between A and B. If for an operation there are several operations satisfying this property then the mark is equal to the smallest latency constraint. We denote this mark by $mark(A) \in \mathbb{N}^* \bigcup \{\infty\}$, where ∞ is a positive integer greater than any other number belonging to $\mathbb{N}^*$. These marks may change their values during the marking algorithm. This latter is applied only on the pattern of the graph like the scheduling algorithm.

Algorithm 1

Initialisation: if (A, B) has a latency constraint $L(A, B)$ then $mark(B) = L(A, B)$ and $mark(A) = \infty$, else $mark(A) = mark(B) = \infty$. Moreover if B belongs to several pairs with latency constraints then $mark(B) = min_{(C,B)\in\mathcal{L}}\{L(C, B)\}$ and $mark(A) = \infty$. Let $\mathcal{W} = \mathcal{L}$.

Step 1: for $(A, B) \in \mathcal{W}$ and for each operation $C \in V \backslash \{A, B\}$, there are three possibilities:

(a) if $A \in \mathcal{D}(C)$ then $mark(C) = mark(C)$;
(b) if $A \notin \mathcal{D}(C)$ and $B \in \mathcal{D}(C)$, then $mark(C) = min(mark(C), mark(B))$;
(c) if $A \notin \mathcal{D}(C)$ and $B \notin \mathcal{D}(C)$ then $mark(C) = mark(C)$.

Let $\mathcal{W} = \mathcal{W} \backslash \{(A, B)\}$.

Step 2: if $\mathcal{W} \neq \emptyset$ then goto Step 1, else the algorithm stops.

We denote by $\mathcal{L}_X$ the set of latency constraints which are in relation X with other latency constraints and which do not have the difference between the value of the latency and the sum of operations belonging to the pair is equal to 0. Without any loss of generality, we consider this set ordered, i.e. if $L(A, B)$ is before $L(C, D)$ in $\mathcal{L}_X$ then $L_{AB} < L_{CD}$.

The scheduling algorithm uses the marks obtained using the marking algorithm. We denote by $\mathcal{W}$ the working set, by s_T the start time of the last scheduled operation, by C_T its execution time and by $Prec(A)$ the set of predecessors of operation A. During the application of the algorithm, the set $\mathcal{W}$ contains the available operations. We notice that between two operations which are available in the same time there is no path. Also, during the application of the algorithm each time that an operation is scheduled, s_T (respectively C_T) is replaced by the start time (execution time) of this operation.

Algorithm 2

Initialisation: $\mathcal{W} = \bigcup_{A \in V \text{ and } Prec(A)=\emptyset} \{A\}$ and $s_T = 0, C_T = 0$.

Step 1 *(operation with latency constraint)*:

if $\exists A \in \mathcal{W}$ such that $mark(A) \neq \infty$ when we chose A such that

$$mark(A) = min_{B \in \mathcal{W}}\{mark(B)\}$$

If $\exists L(C_1, D_1) \in \mathcal{L}_X$ and $\exists L(C_i, D_i) \in \mathcal{L}_X, \forall i = 2, 3, \ldots n$ such that $A \in \bigcup_{E \in \Gamma^+_{C_i D_i}(C_1, D_1)} \mathcal{M}(E, D_1)$, $\forall i = 2, 3, \ldots n$ then let $B \in \mathcal{W}$ be the operation with the smallest mark such that $\exists i \in \{2, 3, \ldots, n\}$ with $B \in \bigcup_{E \in \Gamma^+_{C_1 D_1}(C_i, D_i)} \mathcal{M}(E, D_i)$. If the operation B exists then $s_B = s_T + C_T$, we remove B from $\mathcal{W}$, all operations which become available are added to $\mathcal{W}$ and we go to step 4. If the operation B does not exist then $s_A = s_T + C_T$, we remove A from $\mathcal{W}$, all operations which become available are added to $\mathcal{W}$ and we go to Step 4.

Step 2 *(operation which is not first in a latency constraint)*:

if $\exists A \in \mathcal{W}$ such that there is no operation $B \in \mathcal{V}$ with $(A, B) \in \mathcal{L}$ then $s_A = s_T + C_T$, we remove it from $\mathcal{W}$, all operations which become available are added to $\mathcal{W}$ and we go to Step 4.

Step 3 *(operation which is first in a latency constraint)*:

if $\exists A \in \mathcal{W}$ such that there is $B \in V$ avec $(A, B) \in \mathcal{L}$ alors $s_A = s_T + C_T$ with $L(A, B) = max_{(C,D) \in \mathcal{L} \text{ et } C \in \mathcal{W}}\{L(C, D)\}$, we remove it from $\mathcal{W}$, all operations which become available are added to $\mathcal{W}$ and we go to Step 4.

Step 4:

if $\mathcal{W} = \emptyset$ then the algorithm stops, else we go to Step 1.

The scheduling algorithm 2 is proved optimal for the case where each latency constraint is in relation X with at most one latency constraint. Before proving this optimality, we give an example of schedule obtained with this algorithm.

Example 8 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the graph given in figure 18 with $C_A = 2$, $C_B = 1$ and $C_C = 2$ and the latency constraints $L(A_2, C_2) = 10$ and $L(B, C_1) = 9$. After applying the marking algorithm we obtain the marks given in table 1 where $m(A)$ denotes $mark(A)$.*

We obtain the table 2 using the scheduling algorithm 2.

Theorem 7 *The scheduling algorithm 2 is optimal (if there is a schedule which satisfies the constraints then the algorithm will find it).*

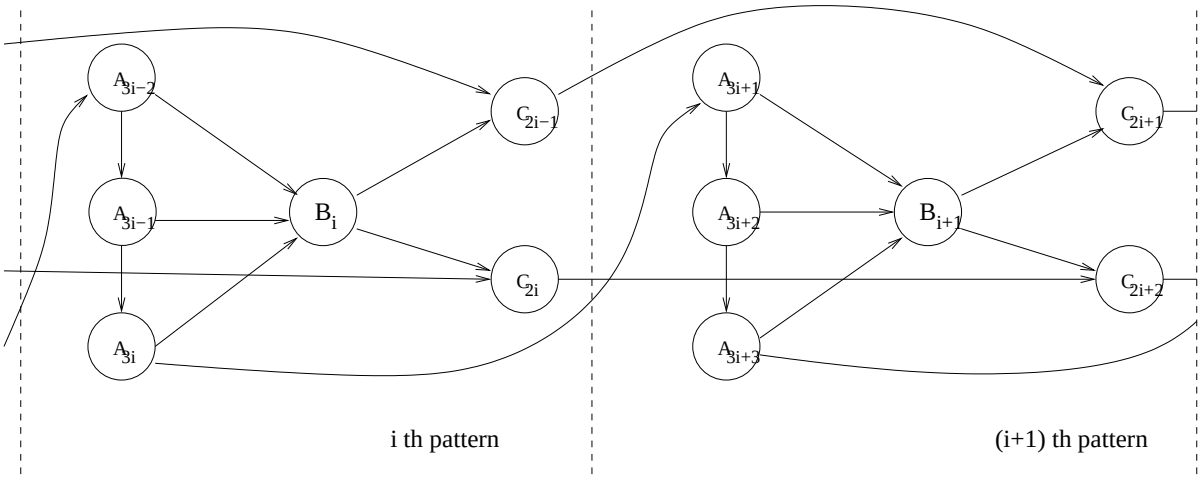

Figure 18: Graph illustrating the scheduling algorithm 2

	$m(A_1)$	$m(A_2)$	$m(A_3)$	$m(B)$	$m(C_1)$	$m(C_2)$
Initialisation	∞	∞	∞	∞	9	10
$L(B, C_1)$	∞	∞	∞	∞	9	10
$L(A_2, C_2)$	∞	∞	10	10	9	10

Table 1: Marks obtained using the marking algorithm

$\mathcal{W}$	s_T	C_T	Step used utilisé	Choice
$\{A_1\}$	0	0	Step 2	$s_{A_1} = 0$
$\{A_2\}$	0	2	Step 3	$sA_2 = 2$
$\{A_3\}$	2	2	Step 1	$s_{A_3} = 4$
$\{B\}$	4	2	Step 1	$s_B = 6$
$\{C_1, C_2\}$	6	1	Step 1	$s_{C_2} = 7$
$\{C_1\}$	7	2	Step 1	$s_{C_1} = 9$

Table 2: Schedule obtained using the algorithm 2

Proof We prove that the scheduling algorithm builds 2 schedules which satisfy the schedulability conditions and if these schedules do not satisfy the constraints then no other will. The proof has three parts : a first one concerning the latency constraints imposed on pairs

in relation ||, a second part concerning the latency constraints imposed on pairs in relation Z and a last part concerning the latency constraints imposed on pairs in relation X.

Without loss of generality, we consider that all latency constraints have different values.

Let $L(A,B)$ and $L(C,D)$ be some two latency constraints imposed on pairs in relation ||. Because there is no path from an operation belonging to (A,B) to an operation belonging to (C,D) then the mark of operations belonging to (A,B) is not modified by the latency constraint $L(C,D)$. Likewise because there is no path from an operation belonging to (C,D) to an operation belonging to (A,B) then the mark of operations belonging to (C,D) is not modified by the latency constraint $L(A,B)$. Therefore the operations belonging to the two pairs have different marks and they will be scheduled either in the following order $s_A + C_A < s_B + C_B < s_C + C_C < s_D + C_D$, either in the order $s_C + C_C < s_D + C_D < s_A + C_A < s_B + C_B$. The two orders lead to the schedulability condition for the case of latency constraints imposed on pairs in relation ||.

Let $L(A,B)$ and $L(C,D)$ be some two latency constraints imposed on pairs which are in relation Z. Because there is no path from an operation belonging to (C,D) to an operation belonging to (A,B) then the mark of operations belonging to (C,D) is not modified by the latency constraint $L(A,B)$. Because there is a path from an operation belonging to (A,B) to at least an operation belonging to (C,D), then the mark of A may be modified by the latency constraint $L(C,D)$ only if $L_{AB} > L_{CD}$. In this case the operations belonging to the two pairs will be scheduled in the following order $s_A+C_A < s_B+C_B < s_C+C_C < s_D+C_D$. In the opposite case, the mark of A is not modified by the latency constraint $L(C,D)$ and the operations belonging to the two pairs will be scheduled either in this order $s_A + C_A < s_B+C_B < s_C+C_C < s_D+C_D$, either in this order $s_C+C_C < s_D+C_D < s_A+C_A < s_B+C_B$. The two orders lead to the schedulability condition for the case of latency constraints imposed on pairs in relation Z.

Let $L(A,B)$ and $L(C,D)$ be some two latency constraints imposed on pairs which are in relation X. Because there is a path from an operation belonging to (A,B) to at least an operation belonging to (C,D), then the mark of A may be modified by the latency constraint $L(C,D)$ only if $L_{AB} > L_{CD}$. In this case the existence of a path between operations belonging to (C,D) to at least an operation belonging to (A,B) does not this mark. Likewise we obtain the same conclusion if $L_{AB} < L_{CD}$.

For the three cases the schedulability conditions are satisfied by the schedules obtained using the algorithm 2. The theorem is proved $\boxdot$

6 Conclusion and future work

We give a schedulability condition for systems with latency and precedence constraints, using the model based on graphs presented in [5]. This result is obtained by studying three relations ||, Z and X between pairs of operations and by studying the latency constraints imposed on the pairs of operations satisfying the latter relations.

We give a scheduling algorithm which is proved optimal for the case where the latency constraints are in X with at most one other latency constraint.

Presently we study the complexity of the scheduling problem for the case where the latency constraints may be in X with at least one other latency constraint. We plan to generalise this schedulability condition to real-time systems with precedence, periodicity and latency constraints.

Contents

References

[1] D Harel and A Pnueli. On the development of reactive systems. In K. R. Apt, editor, *Logics and Models of Concurrent Systems*. Springer Verlag, New York, 1985.

[2] S. Goddard. *Constraints on data-flow*. PhD thesis, University of North Carolina, 2000.

[3] T. Grandpierre, C. Lavarenne, and Y. Sorel. Optimized rapid prototyping for real time embedded heterogeneous multiprocessors. *Codes'99 7th International Workshop on Hardware/Software Co-Design*, 1999.

[4] R. Gerber, S. Hong, and M. Saksena. Guaranteeing real-time requirements with resource-based calibration of periodic processes. *IEEE Transactions on Software Engineering(21)7*, 1995.

[5] L. Cucu, R. Kocik, and Y. Sorel. Real-time scheduling for systems with precedence, periodicity and latency constraints. *Real-time and Embedded Systems*, 2002.

[6] E.L. Lawler. Recent results in the theory of machine scheduling. *Mathematical Programming: the State of the Art, Springer-Verlag*, 1983.

[7] J. Blazewicz. Scheduling dependent tasks with different arrival times to meet deadlines. *Modelling and Performance Evaluation of Computer Systems*, 1976.

[8] Jun Sun and Jane W.S. Liu. Bounding completion times of jobs with arbitrary release times and variable execution times. *Proceedings of 17th IEEE Real-Time Systems Symposium*.

[9] M. Gonzalez Harbour, M. H. Klein, and J.P. Lehoczky. Timing analysis for fixed-priority scheduling of hard real-time systems. *IEEE Transactions on Software Engineering*, 1994.

[10] M. Gonzalez Harbour, M. H. Klein, and J.P. Lehoczky. Fixed priority scheduling of periodic tasks with varying execution priority. *proceedings of the 12th IEEE Real-Time Systems Symposium*, 1991.

[11] H. Chetto, M. Silly, and T. Bouchentouf. Dynamic scheduling of real-time tasks under precedence contraints. *The Journal of Real-Time Systems*, 1990.

[12] R. Balakrishnan and K. Ranganathan. *A Textbook of Graph Theory*. Springer, 2000.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399