Schedulability condition for real-time systems with precedence and periodicity constraints, without preemption

Liliana Cucu, Yves Sorel

INRIA Rocquencourt,
BP 105 - 78153 Le Chesnay Cedex, France
liliana.cucu@inria.fr, yves.sorel@inria.fr

Abstract

We present for one computing resource, the known schedulability conditions for problems with periodicity constraints with and without preemption. After reminding our model with periodicity and precedence constraints, based on graphs, we prove the existence of a hyper-period which gives a schedulability condition without preemption.

1 Introduction

The real-time systems, which are reactive systems, interact with the physical environment through sensors and actuators. These reactions correspond to an infinite repetition of operations, each of these operations may be finitely repeated. This behaviour is, usually, based on periodic operations. We choose to analyze strictly periodic systems because they are easier to study and the results might be extended to typical periodic systems with jitter. Identically, we choose the non-preemptive case because it is easier to study, and we plan to extend the results to the preemptive case evaluating, accurately, its cost.

In [1], we give an optimal scheduling algorithm which, in the case of a schedulable system does not stop due to the infinite nature of the reactive systems. The purpose of this paper is to prove the existence of a hyper-period for systems with periodicity and precedence constraints without preemption. Consequently, we obtain a schedulability condition and the scheduling algorithm stops, in the case of a schedulable system.

The paper starts with notations used to present the main results on systems with periodicity constraints (schedulability condition). The second section presents the model and the problem to solve followed by the section giving the condition of schedulability for this problem. The paper ends with conclusion and future research.

2 Notations and results

In order to clearly distinguish the specification level and its associated model 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 specify a computation time C_A, a release time r_A, a deadline D_A (defined from the start of the schedule), a period T_A 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 remind the main results on feasibility tests or schedulability conditions for systems with periodicity constraints, for one operator. In the non-preemptive case, Jeffay and al. give a schedulability condition for a set of periodic and sporadic operations with arbitrary release times [3] and Howell and al. study the complexity for the problem of non-preemptively scheduling periodic and sporadic task systems on one processor using inserted idle times [4] . In the preemptive case of fixed priority scheduling, there is a method to analyze schedulability given by Harbour [5] and a schedulability criterion for arbitrary deadlines [6]. For a model using Processing Graph Method, Baruah and al. give an efficient feasibility test. Leung and Merrill prove the existence of a hyper-period for a schedule obtained using Earliest Deadline First [7], which allows to have a feasibility test. To our best knowledge, there is no proof of the existence of a hyper-period for a system with precedence and periodicity constraints defined in [1] and the results presented ahead cannot be applied to this problem. Our paper proves the existence of a hyper-period, allowing to obtain a schedulability condition.

3 Model for systems with precedence and periodicity constraints

In [1] we give a model based on graph theory for systems of operations with precedence constraints, and also the definition of the periodicity constraint for an operation.

A directed graph G is the pair (V, E) where V is the set of operations and $E \subseteq V \times V$ the set of edges which represents the precedence constraints between operations. Therefore, the directed pair of operations $(A, B) \in E$ means that B must be scheduled, only if A was already scheduled, and we have $s_A + C_A \leq s_B$.

Each operation may belong to a precedence constraint, i.e. it belongs to a pair defining a partial order and the operations which do not belong to a precedence constraint define a potential parallelism [8]. Moreover, an operation may be repeated leading to several instances of the same operation either spatially without precedence constraint between consecutive repetitions, defining also a potential parallelism for this subgraph, or temporally with precedence constraints between consecutive repetitions. The repetition is useful when specifying some applications where operations must be executed several times in order to produce data results consumed by their successors.

Actually, the real-time system, which is reactive [9], interacts with the physical environment, therefore the graph of precedence constraints has a *pattern* infinitely repeated [8]. If we 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 inter-pattern and the edges between operations belonging to different patterns are called extra-pattern. For example in the pattern of figure 1, A is repeated temporally three times whereas C is repeated spatially two times, A_1 is an input operation, and C_1, C_2 are output operations. The edge from A_3 of the i^{th}

pattern to A_1 of the $(i+1)^{th}$ pattern is extra-pattern and the edge from B to C_1 is intra-pattern. Note that C_1 and C_2 define a potential parallelism. The infinite repetition of the pattern of the graph induces an infinite repetition of all the operations.

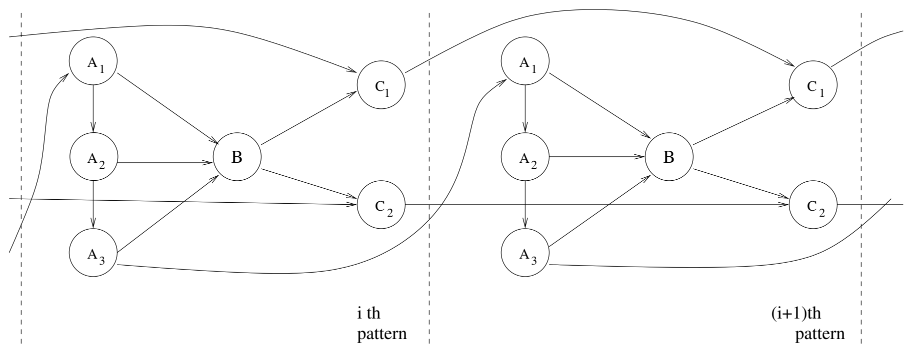

Figure 1: System of operations given by a graph

Definition 1 *[10] A directed path starting with A_1 and ending with 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.*

Because we work only with directed graphs, for the sake of simplicity, we will use the term path instead of directed path. We denote by $\mathcal{P}$ the set of all the paths of the graph G and we say that $P(A, B) \in \mathcal{P}$, if there is at least a path starting with A and ending with B. If $\nexists P(A, B) \in \mathcal{P}$, then there is no path starting with A and ending with B.

Definition 2 *[1] For two consecutive temporal repetitions A_i and A_{i+1} of the same operation A (belonging to the same pattern or to consecutive patterns), we say that A has a constraint of periodicity (is periodic with) T_A if $s_{A_{i+1}} - s_{A_i} = T_A, \forall i \in \mathbb{N}$. We denote by A_1 the first repetition of A.*

Remark 1 *On the contrary, an operation spatially repeated cannot have a periodicity constraint, because there is no total order between the repeated operations. This is the case of operation C in figure 1.*

Definition 3 *For a system of operations, a schedule S is the set of the start times of all the operations $S = \{s_A, A \in V\}$ such that the precedence and the periodicity constraints are satisfied. If there is a schedule for a system, the system is called schedulable; if not, then it is unschedulable.*

Without any loss of generality [11], we assume that all the operation characteristics, i.e. the fixed computation time (exactly known), the periodicity constraints, are defined as multiples of a clock tick τ (time is discrete). Afterwards the integer values of the given characteristics are implicitly multiplied by τ.

Remark 2 *We assume that the periodic operations are scheduled strictly with their exact periodicity constraints. Thus, it amounts to not allow any jitter. Also, we assume that all the periodicity constraints are not coprime to each other and there is, at least, one periodic operation in each connected component.*

For two periodic operations A, B with $(A, B) \in E$ belonging to a system given by a graph $G = (V, E)$, it is obvious that we must have

$$T_A \leq T_B \tag{1}$$

Indeed, because the first repetition of A must be scheduled before the first repetition of B due to the precedence constraint, then satisfying the infinitely repeated precedence constraints of the next repetitions of A and B, implies satisfying equation (1). Identically, if the operations A and B are finitely repeated, respectively, n and m times in the infinitely repeated pattern, and there is a precedence constraint from, at least, a repetition of A to a repetition of B, then we must have

$$nT_A \leq mT_B \tag{2}$$

In the next section we give a condition of schedulability for the problem of a system with periodicity and precedence constraints, in the non-preemptive case.

4 Condition of schedulability

The condition of schedulability for a system of operations with precedence and periodicity constraints, is a direct consequence of the existence of a hyper-period, i.e. the existence of a pattern of start times infinitely repeated in the schedule. This latter pattern includes or is equal to the pattern of the graph which gives the system. When a precedence constraint is due to a data transfer between operations, because the size of the data buffers is limited, the inequality (2) becomes an equality $nT_A = mT_B$. In this case, the pattern of a schedule is equal to the pattern of the graph which gives the system.

The existence of a hyper-period is obtained by proving that a non-periodic operation inherits the period of its successors and of its predecessors, which allows, thanks to equation (2), to conclude that the partial order of the precedence constraints implies an increasing order of the periodicity constraints of operations.

Theorem 1 *Let A, B with $(A, B) \in E$ be two operations belonging to a system given by a graph $G = (V, E)$. We suppose that A has no periodicity constraint and B is the only successor of A which has a periodicity constraint T_B. Let $G' = (V, E')$ the graph obtained from G by adding a precedence constraint from the operation B of the i^{th} pattern to the operation A of the $(i+1)^{th}$ pattern, $\forall i \in \mathbb{N}$. If the system given by G is schedulable, then the system given by G' is, also, schedulable and A inherits the periodicity constraint of B (both schedules have to satisfy the same periodicity constraint for B and use the same computation times for A and B).*

Proof Let $S = \{s_A | A \in V\}$ be some schedule for the system given by G. Then, S must satisfy the precedence constraints of A and B and the periodicity constraint of B. We write the periodicity constraint of B:

$$s_{B_i} - s_{B_{i-1}} = T_B, \forall i \geq 1 \tag{3}$$

where s_B is the start time of B and B_i is the i-th repetition of B. We write the precedence constraints:

$$s_{A_i} + C_A \leq s_{B_i}, \forall i \geq 1 \tag{4}$$

Because of the precedence constraints in the infinitely repeated pattern, a schedule S may contain several repetitions of A before the first repetition of B. Let n_0 be the number of repetitions of A scheduled in S before the first repetition of B. So, we have $s_{A_1} + n_0 C_A \leq s_{B_1}$.
The inequality (4) implies that A_{n_0+1} must be scheduled before B_{n_0+1}. We have, at least, one repetition of A scheduled between two consecutive repetitions of B, i.e. $\exists i_0 \leq n_0$ such that

$$s_{B_{i_0}} + C_B \leq s_{A_{n_0+1}} \tag{5}$$

$$s_{A_{n_0+1}} + C_A \leq s_{B_{i_0+1}} \tag{6}$$

By adding the inequalities (5) and (6), we obtain $s_{B_{i+1}} - s_{B_i} \geq C_A + C_B$ and, by using (3), we obtain $T_B \geq C_A + C_B$. This inequality shows that between two consecutive repetitions of B, there is enough time to schedule one repetition of A.

Without loss of generality, we may suppose that $s_{A_{n_0}}$ is equal to $s_{B_{i_0}} + \mathcal{C}$, where $\mathcal{C} < T_B - C_B$ is an integer constant. Because all the periodicity constraints are not coprime to each other, since A was scheduled at $s_{B_{i_0}} + \mathcal{C}$, there is no periodic operation which must be scheduled at $s_{B_{i_0}} + \mathcal{C} + iT_B$. This means that there is a schedule S' for which the start times of A and B satisfies:

$$S' = \begin{cases} s_{B_i} = s_{A_i} + T_B - \mathcal{C}, \forall i \geq 1 \\ s_{A_{i+1}} = s_{A_i} + T_B, \forall i \geq 1 \end{cases} \tag{7}$$

The two equations of (7) shows that S' is a schedule satisfying, beside the constraints given by G, also, a precedence constraint from B_i to A_{i+1}, so the schedule S' is a schedule for the system given by G'. Consequently, all the start times of A in the schedule S' satisfy a periodicity constraint $T = T_B$. So, A inherits the periodicity constraint of B $\boxdot$

Theorem 2 *(generalization of theorem 1)*
Let $A, B_i, \forall i \in \{1, 2, \ldots, n\}$ with $(A, B_i) \in E, \forall i \in \{1, 2, \ldots, n\}$ be $n+1$ operations belonging to a system given by a graph $G = (V, E)$. We suppose that A has no periodicity constraint and $B_i, \forall i \geq 1$ are the only periodic successors (B_1 has the smallest value of periodicity constraints among all B_i). Let $G' = (V, E')$ the graph obtained from G by adding a precedence constraint from the operation B_1 of the j^{th} pattern to the operation A of the $(j+1)^{th}$ pattern, $\forall j \in \mathbb{N}$. If the system given by G is schedulable, then the system given by G' is, also, schedulable. A inherits the periodicity constraint $T_{B_1} = min_{i \in \{1,\ldots,n\}}\{T_{B_i}\}$ (both schedules have to satisfy the same periodicity constraint for $B_i, \forall i \geq 1$ and use the same computation times for A and $B_i, \forall i \geq 1$).

Proof The proof is made by mathematical induction using as induction integer i the number of successors of the operation A and using Theorem 1 to verify the statement for $i = 1$ $\boxdot$

The results obtained until now are verified by all the schedules. The following theorems are satisfied only by a schedule obtained with the optimal algorithm presented in [1]. We recall that this algorithm schedules the schedulable periodic operations in the increasing order of their periodicity constraints and the schedulable non-periodic operations as soon as possible, minimizing the idle time.

Theorem 3 *Let B be one operation belonging to a system given by a graph $G = (V, E)$. We suppose that B has no periodicity constraint and $\exists n$ periodic predecessors A_i. If the system is schedulable, then B inherits the periodicity constraint $T = max_{i \in \{1,\ldots,n\}}\{T_{A_i}\}$.*

Proof Without loss of generality, we consider that A_n is the operation with the biggest value of the periodicity constraint among all $A_i, \forall i \in \{1, \dots, n\}$. Because A_n has the biggest value of the periodicity constraint, it will be scheduled last among all $A_i, \forall i \in \{1, \dots, n\}$ and the next repetition of A_n will be the last scheduled among all $A_i, \forall i \in \{1, \dots, n\}$). Because the system is schedulable (all operations are scheduled), B is scheduled between two consecutive repetitions of A_n, once A_n and all its non-periodic predecessors are scheduled. Because all the periodicity constraints are not coprime to each other, since B was scheduled at s_B, there is no periodic operation which must be scheduled at $s_B + iT_{A_n}, \forall i \geq 1$. This means that B inherits the periodicity constraint of A $\boxdot$

Corollary 1 *Let B be an operation belonging to a system given by a graph $G = (V, E)$. We suppose that B has no periodicity constraint. We suppose that $\exists n$ periodic operations A_i such that $\exists P(A_i, B) \in \mathcal{P}$ and that $\exists m$ periodic operations C_j such that $\exists P(B, C_j) \in \mathcal{P}$. If the system is schedulable, then B inherits the periodicity constraint $T = min\{min_{i \in \{1,\dots,n\}}\{T_{A_i}\}, max_{j \in \{1,\dots,m\}}\{T_{C_j}\}\}$*

Proof The existence of, at least, a path from A_i to $B, \forall i \in \{1, 2, \dots, n\}$ and from B to $C_j, \forall j \in \{1, 2, \dots, m\}$ imposes a total order relation on the set of operations belonging to all these paths and the transitivity of this order allows to generalize the result obtained in Theorem 2 and Theorem 3 $\boxdot$

Example 1 *We present an example of a system given by the graph G depicted in figure 2. We have $T_A = 5$ and $T_D = 10$. Using Corollary 1, C inherits the periodicity constraints of A and D, E inherits the periodicity constraint of A, also B inherits the periodicity constraint of C. So, we have $T_A = T_B = T_C = T_E = 5$ and $T_D = 10$. For $C_A = C_B = C_C = C_D = C_E = 1$, we obtain the schedule S of figure 3.*

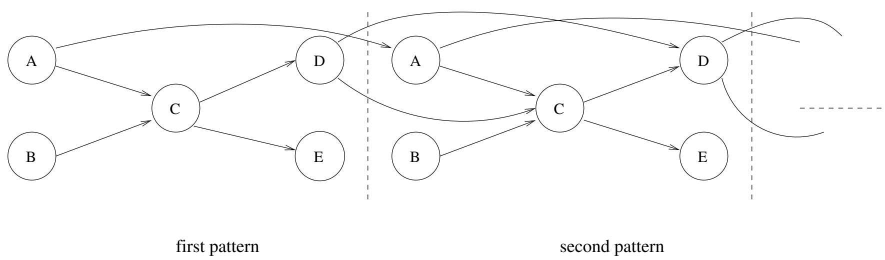

Figure 2: Graph G

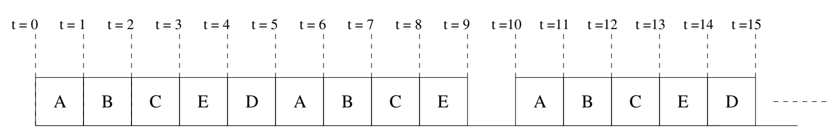

Figure 3: Schedule S for the system given by G

We denote by T the lcm (least common multiple) of all periodicity constraints of the system.

Definition 4 *A configuration of a system from time t_1 to time t_2, denoted by $S(t_1, t_2)$, is the set of start times of operations scheduled from t_1 to t_2.*

Theorem 4 *for a system with periodicity and precedence constraints, we have:*

$$S(iT, iT+t) = S((i+1)T, (i+1)T+t), \forall 0 < t \leq T \text{ and } i \geq 0$$

Proof We prove the theorem by double mathematical induction using the induction integers i and t. Let $P(i), \forall i \geq 0$ be the statement $S(iT, iT+t) = S((i+1)T, (i+1)T+t), \forall 0 < t \leq T$. First, we verify that $P(0)$ is true, by mathematical induction using the induction integer t. Let $Q(t) = P(0)$, i.e. $Q(t)$ is the statement $S(0,t) = S(T, T+t), \forall 0 < t \leq T$. We verify that $S(0,1) = S(T, T+1)$. Indeed, $S(0,1)$ contains, only, the first operation scheduled which has either its own periodicity constraint, either the inherited periodicity constraint from its successors. Because T is the lcm of all periodicity constraints, the first operation is scheduled, also, at time T. Therefore, $S(T, T+1)$ contains, exactly the same operation as $S(0,1)$ (we recall that the computation times are integers). So, $S(0,1) = S(T, T+1)$ is true.

Applying the induction principle, we suppose that $Q(t)$ is true and we verify that $Q(t+1)$ is, also, true ($Q(t)$ true implies $Q(t+1)$ true). We have $S(0,t) = S(T, T+t)$ and we want to prove that $S(0, t+1) = S(T, T+t+1)$, i.e. which means that the same operation is scheduled at t and $T+t$. We have two possibilities: either the last operation scheduled finished until t, either not.

In the first case, since we have the same operations scheduled from 0 to t as from T to $T+t$, at times t and $T+t$ we will have the same set of schedulable operations. We assume that we use a scheduling algorithm which guarantees that if it is applied several times to the same system of operations, it will give the same schedule. This means that the same operation will be chosen at t and $T+t$.

In the second case, because the scheduling is non-preemptive, the last operation will continue to use the resource at time t and $T+t$.

So, we proved that $Q(t+1)$ is true, if $Q(t)$ is true, and the induction using the induction integer t is proved. Since $Q(t)$ is verified, $P(0)$ is, also, verified.

Applying the induction principle, we suppose that $P(i)$ is true and we verify that $Q(i+1)$ is, also, true ($Q(t)$ true implies $P(i+1)$ true). The proof of the implication is similar to the implication $Q(t)$ true implies $Q(t+1)$ true.

Finally, $P(i)$ is true, $\forall i \geq 0$. The theorem is proved $\boxdot$

Corollary 2 *If a system with precedence and periodicity constraints given by $G = (V, E)$ is schedulable, then $\sum_{A \in V} \frac{C_A}{T_A} \leq 1$.*

Proof As seen in Theorem 4, a schedule for a system with precedence and periodicity constraints has a hyper-period T equal to the lcm of all its periodicity constraints. So, if a system with precedence and periodicity constraints is schedulable, then there is a schedule from 0 to T.

Because all the periodicity constraints are not coprime to each other, two start times of two different operations could not be equal. An operation A with a periodicity constraint T_A will be scheduled $n_A = \frac{T}{T_A}$ times from 0 to T. So, if a system is schedulable then all the operations are scheduled from 0 to T, and we have $\sum_{A \in V} n_A C_A \leq T$. Because $n_A = \frac{T}{T_A}$, we obtain that a system with precedence and periodicity constrains is schedulable implies that $\sum_{A \in V} \frac{C_A}{T_A} \leq 1$. The corollary is proved ⊡

Corollary 3 *Applying the scheduling algorithm proposed in [1] from* 0 *to* T*, it is possible to decide if a system with precedence and periodicity constraints is schedulable or not.*

Proof As seen in Theorem 4, a schedule for a system with precedence and periodicity constraints has a hyper-period T equal to the lcm of all its periodicity constraints. We remind that the scheduling algorithm proposed in [1] finds a schedule, if there is one. If the scheduling algorithm finds a schedule from 0 to T, then the system is schedulable. On the contrary, if the scheduling algorithm does not find a schedule from 0 to T, then the system is not schedulable ⊡

Remark 3 *We remind that, in the case of a schedulable system, the scheduling algorithm proposed in [1] does not stop. Applying Corollary 3, in order to find a schedule, this algorithm is used only from* 0 *to* T*, instead of from* 0 *to* ∞*.*

5 Conclusion and further research

In this paper, we proved the existence of a hyper-period for a system with precedence and periodicity constraints without preemption, allowing to our scheduling algorithm to find a schedule, applying it from 0 to T, instead of from 0 to ∞. Also, we give a schedulability condition for these systems, in the case without preemption.

We want to extend these results to typical systems with precedence and periodicity constraints with jitter modeled by a latency constraint. Moreover, when the condition of schedulability is not satisfied, we plan to study if introducing preemption allows to find a schedule.

References

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

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

[3] K. Jeffay, D. Stanat, and C. Martel. On non-preemptive scheduling of periodic and sporadic tasks. *IEEE Real-Time Systems Symposium*, 1991.

[4] R.R. Howell and M.K. Venkatrao. On non-preemptive scheduling of recurring tasks using inserted idle time. *Information and computation Journal*, 1995.

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

[6] J.P. Lehoczky, L. Sha, and Y Ding. The rate monotonic sheduling algorithm: exact characterization and average case bahavior. *Proceedings of the IEEE Real-Time Systems Symposium*, 1989.

[7] Joseph Y.-T. Leung and M. L. Merrill. A note on preemptive scheduling of periodic, real-time tasks. *Information Processing Letters*, 1980.

[8] 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.

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

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

[11] S.K. Baruah, R.R. Howell, and L.E. Rosier. Algorithms and complexity concerning the preemptive scheduling of periodic real-time tasks on one processor. *Journal of Real-Time Systems*, 1990.

Appendix

The following theorem gives a result which is independent of the rest of the paper. Its only purpose is your pleasure for reading.

Theorem 5 *Let A, B with $(A, B) \in E$ be two operations belonging to a system given by a graph $G = (V, E)$. We suppose that A has a periodicity constraint T_A and B has a periodicity constraint T_B. If $T_A > T_B$ then the system is not schedulable.*

Proof We prove by contradiction. We suppose that there is a schedule S which satisfy the periodicity and precedence constraints of the system, implicitly, the constraints for A and B. The periodicity constraints of A and B imply:

$$s_{A_i} = s_{A_1} + (i-1)T_A, \forall i \geq 1 \tag{8}$$

$$s_{B_i} = s_{B_1} + (i-1)T_B, \forall i \geq 1 \tag{9}$$

The precedence constraint from A to B is equivalent to:

$$s_{A_i} + C_A \leq s_{B_i}, \forall i \geq 1 \tag{10}$$

Because of the precedence constraints, a schedule S may contain several repetitions of A before the first repetition of B. Let $n_0 > 0$ be the number of repetitions of A scheduled in S before the first repetition of B and we have:

$$s_{A_{n_0}} + C_A \leq s_{B_1} \leq s_{A_{n_0+1}} \tag{11}$$

Writing (10) for $\forall m > n_0$, we obtain $s_{A_m} + C_A \leq s_{B_m}$. By using (8), (9) and (11), we have: $s_{A_{n_0}} + (m - n_0)T_A + C_A \leq s_{B_1} + (m-1)T_B \leq s_{A_{n_0+1}} + (m-1)T_B$. So, we have $m \leq \frac{(n_0+1)T_A - T_B - C_A}{T_A - T_B}$ which is in contradiction with $\forall m > n_0$. Also, it shows, that starting from $i \geq \frac{(n_0+1)T_A - T_B - C_A}{T_A - T_B}$, A_i must be scheduled after B_i in order to satisfy its periodicity constraints, but the schedule does not satisfy the precedence, anymore. In conclusion, there is no schedule for the system and the theorem is proved $\boxdot$