

5.45 calendar

	DESCRIPTION	LINKS
Origin	[25]	
Constraint	calendar(INSTANTS, MACHINES)	
Type	UNAVAILABILITIES : collection(low-int, up-int)	
Arguments	INSTANTS : collection($\left(\begin{array}{l} \text{machine-dvar,} \\ \text{virtual-dvar,} \\ \text{ireal-dvar,} \\ \text{flagend-int} \end{array} \right)$ MACHINES : collection(id-int, cal - UNAVAILABILITIES)	
Restrictions	required(UNAVAILABILITIES, [low, up]) required(INSTANTS, [machine, virtual, ireal, flagend]) in_attr(INSTANTS, machine, MACHINES, id) INSTANTS.flagend \geq 0 INSTANTS.flagend \leq 1 required(MACHINES, [id, cal]) distinct(MACHINES, id)	
Purpose	Makes the link between an universal calendar and resource dependent calendars. Given a collection of machines MACHINES where each machine is defined by its identifier and its unavailability periods the calendar constraint maps items of real and virtual dates depending on the machine assignment as well as of the fact that we consider start (flagend = 0) or end (flagend = 1) times. Virtual dates on a given machine m do not consider the unavailability periods on m , while real dates consider all time points.	
Example	$\left(\begin{array}{l} \text{machine - 1} \quad \text{virtual - 2} \quad \text{ireal - 3} \quad \text{flagend - 0,} \\ \text{machine - 1} \quad \text{virtual - 5} \quad \text{ireal - 6} \quad \text{flagend - 1,} \\ \text{machine - 2} \quad \text{virtual - 4} \quad \text{ireal - 5} \quad \text{flagend - 0,} \\ \left\langle \begin{array}{l} \text{machine - 2} \quad \text{virtual - 6} \quad \text{ireal - 9} \quad \text{flagend - 1,} \\ \text{machine - 3} \quad \text{virtual - 2} \quad \text{ireal - 2} \quad \text{flagend - 0,} \\ \text{machine - 3} \quad \text{virtual - 5} \quad \text{ireal - 5} \quad \text{flagend - 1,} \\ \text{machine - 4} \quad \text{virtual - 2} \quad \text{ireal - 2} \quad \text{flagend - 0,} \\ \text{machine - 4} \quad \text{virtual - 7} \quad \text{ireal - 9} \quad \text{flagend - 1} \end{array} \right\rangle, \\ \text{id - 1} \quad \text{cal - } \langle \text{low - 2 up - 2, low - 6 up - 7} \rangle, \\ \left\langle \begin{array}{l} \text{id - 2} \quad \text{cal - } \langle \text{low - 2 up - 2, low - 6 up - 7} \rangle, \\ \text{id - 3} \quad \text{cal - } [], \\ \text{id - 4} \quad \text{cal - } \langle \text{low - 3 up - 4} \rangle \end{array} \right\rangle \end{array} \right)$	

Figure 5.84 illustrates the example. It present four machines with their respective unavailability periods (in grey) as well as four tasks (in blue and pink). Each item of the INSTANTS collection corresponds to the start or to the end of one of the previous four tasks. The calendar constraint holds since:

- The real date 3 (INSTANTS[1].ireal = 3) associated with the start (INSTANTS[1].flagend = 0) of task (a) in the universal time corresponds to the virtual date 2 (INSTANTS[1].virtual = 2) on machine 1 (INSTANTS[1].machine = 1).
- The real date 6 (INSTANTS[2].ireal = 6) associated with the end (INSTANTS[2].flagend = 1) of task (a) in the universal time corresponds to the virtual date 5 (INSTANTS[2].virtual = 5) on machine 1 (INSTANTS[2].machine = 1).
- The real date 5 (INSTANTS[3].ireal = 5) associated with the start (INSTANTS[3].flagend = 0) of task (b) in the universal time corresponds to the virtual date 4 (INSTANTS[3].virtual = 4) on machine 2 (INSTANTS[3].machine = 2).
- The real date 9 (INSTANTS[4].ireal = 9) associated with the end (INSTANTS[4].flagend = 1) of task (b) in the universal time corresponds to the virtual date 6 (INSTANTS[4].virtual = 6) on machine 2 (INSTANTS[4].machine = 2).
- The real date 2 (INSTANTS[5].ireal = 2) associated with the start (INSTANTS[5].flagend = 0) of task (c) in the universal time corresponds to the virtual date 2 (INSTANTS[5].virtual = 2) on machine 3 (INSTANTS[5].machine = 3).
- The real date 5 (INSTANTS[6].ireal = 5) associated with the end (INSTANTS[6].flagend = 1) of task (c) in the universal time corresponds to the virtual date 5 (INSTANTS[6].virtual = 5) on machine 3 (INSTANTS[6].machine = 3).
- The real date 2 (INSTANTS[7].ireal = 2) associated with the start (INSTANTS[7].flagend = 0) of task (d) in the universal time corresponds to the virtual date 2 (INSTANTS[7].virtual = 2) on machine 4 (INSTANTS[7].machine = 4).
- The real date 9 (INSTANTS[8].ireal = 9) associated with the end (INSTANTS[8].flagend = 1) of task (d) in the universal time corresponds to the virtual date 7 (INSTANTS[8].virtual = 7) on machine 4 (INSTANTS[8].machine = 4).

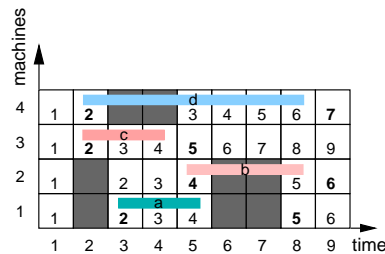


Figure 5.84: Four machines with their unavailability periods as well as four tasks assigned to these machines (virtual dates mentioned in the **Example** slot use a bold font)

Typical

```
|INSTANTS| > 1
|MACHINES| > 1
```

Symmetries

- Items of INSTANTS are [permutable](#).
- Items of MACHINES are [permutable](#).

Usage

The `calendar` constraint is used as a [channelling constraint](#) in resource scheduling problems where resources have unavailability periods that can preempt the execution of a task. In this context two time coordinates systems are used:

- A first coordinate system, so called the *virtual coordinate system*, ignores all unavailability periods on the different resources. All resource constraints are stated within this virtual coordinate system.
- A second coordinate system, so called the *real coordinate system*, corresponds to the real time. All temporal constraints (e.g., precedence constraints) are stated within this real coordinate system.

In this context, each task has a *virtual origin*, a *virtual duration*, a *virtual end*, a *real origin*, a *real duration*, a *real end* and the `calendar` constraint links together the virtual origin and the real origin as well as the virtual end and the real end. The virtual duration (i.e., the real duration plus the sum of the unavailability periods crossed by the task) is linked to the virtual end and the virtual origin through an equality constraint on the difference between the virtual end and the virtual origin. The real duration is linked in a similar way to the real end and the real origin. The keyword [scheduling with machine choice, calendars and preemption](#) provides a concrete example of resource scheduling problem using the `calendar` constraint.

Reformulation

The `calendar` constraint can be reformulated into two generalised case constraints (i.e., two case constraints augmented with linear constraints). Part (A) (respectively Part (B)) of Figure 5.85 provides the dag that allows mapping the virtual start and real start (respectively the virtual end and real end) of a task. This dag can be computed directly from the arguments of the `calendar` constraint:

1. We create an initial root node labelled by m and we partition the set of machines into classes of consecutive machines that all share exactly the same unavailability periods. For each such class we create an arc from the root node to a new node vs labelled by the corresponding interval of consecutive machines identifiers. In Part (A) this corresponds to node m and its three outgoing arcs respectively labelled by intervals $[1, 2]$, $[3, 3]$ and $[4, 4]$.
2. For each class of consecutive machines found previously, we label in increasing order each timepoint that is not part of an unavailability period. We create an arc from the corresponding node vs for each maximum interval of available timepoints to a new node labelled by rs . In Part (A) this translate to:
 - For the class corresponding to machines 1 and 2 we create three outgoing arcs labelled by the time intervals $[1, 1]$, $[2, 4]$ and $[5, 6]$.
 - For the class corresponding to machine 3 we create the outgoing arc labelled by time interval $[1, 9]$.
 - For the class corresponding to machine 4 we create the two outgoing arcs labelled by the time intervals $[1, 2]$ and $[3, 7]$.

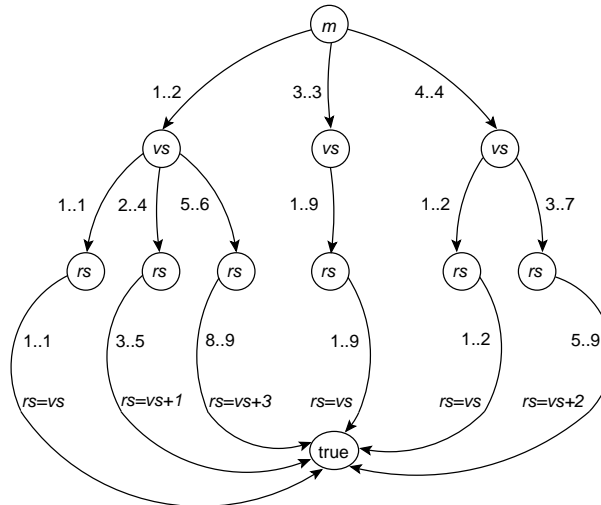
3. For each class of consecutive machines and for each maximum interval $[i, j]$ of available timepoints previously computed, we find out the number of unavailable timepoints b_i on the same class of machines that are located before the virtual coordinate i . We create an outgoing arc from the corresponding node r_s to a new node labelled by true (there is one single true node for the full dag). This arc is labelled by the interval $[i + b_i, j + b_i]$ and by the linear constraint $r_s = v_s + b_i$. In Part (A) this translate to:
- For the class corresponding to machines 1 and 2 and for each r_s node associated with the time intervals $[1, 1]$, $[2, 4]$ and $[5, 6]$ we respectively create an outgoing arc labelled by intervals $[1, 1]$, $[3, 5]$ and $[8, 9]$. To each of these arcs we also respectively associate the linear constraints $r_s = v_s + 0$ (+0 since on machines 1 and 2 there is no unavailability period before the virtual coordinate 1), $r_s = v_s + 1$ (+1 since on machines 1 and 2 there is one single unavailable timepoint before the virtual coordinate 2) and $r_s = v_s + 3$ (+3 since on machines 1 and 2 there is three unavailable timepoints before the virtual coordinate 5).
 - For the class corresponding to machine 3 and for the r_s node associated with the time interval $[1, 9]$ we create the outgoing arc labelled by time interval $[1, 9]$ and by $r_s = v_s + 0$ (i.e., since there is no unavailability period at all on machine 3).
 - For the class corresponding to machine 4 and for each r_s node associated with the time intervals $[1, 2]$ and $[3, 7]$ we respectively create an outgoing arc labelled by $[1, 2]$ and $[5, 9]$. To each of these arcs we also respectively associate the linear constraints $r_s = v_s + 0$ (+0 since on machine 4 there is no unavailability period before the virtual coordinate 1) and $r_s = v_s + 2$ (+2 since on machine 4 there is two unavailable timepoints before the virtual coordinate 3).

See also

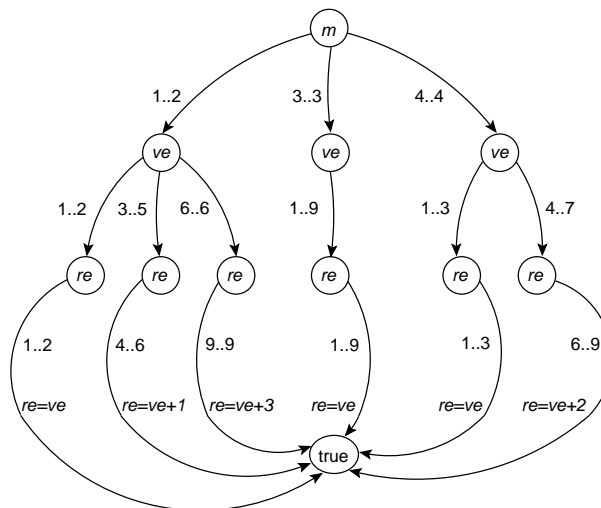
common keyword: [cumulative \(scheduling constraint\)](#), [cumulatives](#), [diffn \(scheduling with machine choice, calendars and preemption\)](#), [disjunctive \(scheduling constraint\)](#), [geost \(scheduling with machine choice, calendars and preemption\)](#).

Keywords

constraint type: [predefined constraint](#), [temporal constraint](#), [scheduling constraint](#).
modelling: [channelling constraint](#), [scheduling with machine choice, calendars and preemption](#), [assignment dimension](#).
modelling exercises: [scheduling with machine choice, calendars and preemption](#).



(A) linking the machine attribute, the virtual start and the real start of a task



(B) linking the machine attribute, the virtual end and the real end of a task

Figure 5.85: The two generalised case constraints for respectively mapping (1) the virtual start and real start of a task corresponding to the **Example** slot as well as (2) the virtual end and real end

