The *period* Constraint

Nicolas Beldiceanu and Emmanuel Poder

LINA FREE CNRS 2729, École des Mines de Nantes,

FR-44307 Nantes Cedex 3, France.

Email: {Nicolas.Beldiceanu,Emmanuel.Poder}@emn.fr

ICLP, 08-09-2004

OUTLINE

- CONTEXT
- WORK ON SEQUENCES
- FILTERING FOR THE *PERIOD* CONSTRAINT
- CONCLUSION

- **CONTEXT**
- WORK ON SEQUENCES
- FILTERING FOR THE *PERIOD* CONSTRAINT
- CONCLUSION

The *period* constraint

The ***period*** *per*(*s*,*ctr*) *of a sequence* $s=v_0,v_1,\ldots,v_{m-1}$ for a relation *ctr* is **the smallest integer** *p* such that $ctr(v_i,v_{i+p})$ for $i\in\{0,1,\ldots,m-p-1\}$.

Examples : *s* = **1 1 4 1 1 4 1 1** has a period of **3** for the relation *ctr* = '**=**' ,

s = **3 1 4 6 9 7** has a period of **2** for the relation *ctr* = '**<**' .

The *constraint* ***period***(***P***,***Sequence***,***ctr***) where ***P*** is a variable, ***Sequence*** a list $[V_0,V_1,\ldots,V_{m-1}]$ of *m* variables and ***ctr*** a relation, holds iff ***P*** is **the period** of the sequence $V_0V_1\ldots V_{m-1}$ for the relation ***ctr***.

(*By default* ***ctr*** *is the equality*)

Motivation: behind *period*

- From an algorithm to a constraint

  (**break** the distinction between input parameters and output parameters)

- Constraint parametrized by a relation

  (filtering algorithm **depends** of the **properties** of the relation)

Practical motivation

Cyclic timetable:

- Don't **know** the period (but want to minimise it)
- Not a **pure** period constraint (holidays **break** the period, …)

Example : ctr(X,Y) = (X=Y) ∨ (X=0) ∨ (Y=0).

s = **1 1 3 1 0 3 1 1** has a period **3** according to previous relation

(**0**: holiday, **1**: morning, **2**: afternoon, **3**: night)

Difficulties with the *period* constraint

- The concept is simple but **difficult to modelise**

  (network of constraints, graph property, automaton)

- *P* has to correspond to **the smallest** period and not to a period.

- CONTEXT
- **WORK ON SEQUENCES**
- FILTERING FOR THE *PERIOD* CONSTRAINT
- CONCLUSION

Work on sequences (algorithms)

- Areas: Localisation, Index, Patterns with wildcard, …

  (A. Apostolico, M. Crochemore, …)

- Trend coming from the applications:
  - Data are **not completely fixed**,
  - Use of wildcards.

Work on sequences (constraint area)

- ***sequence*** constraint of CHIP
- A Filtering Algorithm for ***global sequencing*** constraint [RÉGIN, PUGET CP'97]
- Constraints ***stretch***, ***pattern*** et ***regular*** [PESANT CP'2001, CP'2003, CP'2004]
- Constraint Reasoning over Strings [GOLDEN, PANG CP'2003]
- Deriving Filtering Algorithms from Constraint Checkers

  [BELDICEANU, CARLSSON, PETIT CP'2004]

- Context
- Work on Sequences
- **Filtering for the *Period* Constraint**
- Conclusion

Organization of the filtering algorithm

- **Pruning *P*** (according to the variables of the sequence)
  - Adjust *min*(*P*) : Prop. **2**
  - Adjust *max*(*P*) : Prop. **6**, **7**
  - Remove values from *dom*(*P*) : Prop. **3**, **4**, **5**(**3**), **8**, **9**

- **Pruning the variables of the sequence** (according to *P*)
  - Avoiding being below *min*(*P*) : Prop. **13**(**6**)
  - Achieving a period of *dom*(*P*) : Prop. **10**, **11**(**4**,**5**), **12**(**9**)

Some basic definitions about sequences

- A sequence r is a **factor** of a sequence s if there exist u and v such that $s=urv$.
  - if $u = \varepsilon$ r is a **prefix** of s,
  - if $v = \varepsilon$ r is a **suffix** of s,
  - if $r \neq s$ r is a **proper factor** of s.
- A **border** of a non-empty sequence s is a **proper factor of s** which both is **prefix and suffix of s**. *Bord*(s) designates the longest border of s.
- $|s| - |Bord(s)| = per(s)$ where $|t|$ is the length of the sequence t.

Example : The borders of s = **1 1 4 1 1 4 1 1** are: ε, **1**, **1 1**, **1 1 4 1 1** .
($|s|=8$, $|Bord(s)|=5$, $per(s)=3$)

Generalisation of the border concept

Definition. The *border* of a non-empty sequence ***s*** according to a relation ***ctr*** is defined by two proper factors ***u*** and ***v*** of ***s*** such that:

– ***u*** is a prefix of ***s***,

– ***v*** is a suffix of ***s***,

– ***u*** and ***v*** have the same length,

– *ctr*(***u[i]***,***v[i]***) holds $\forall i \in \{0,1,\ldots,|u|-1\}$.

Notation. *Lbord*(***s***,*ctr*) designates the length of the longest border of ***s***,

it has ***Pbord(s,ctr)*** as prefix and ***Sbord(s,ctr)*** as suffix.

Example : ***s*** = **3 1 4 6 9 7** and *ctr* = ‘<‘ .

Pbord(s,ctr) = **3 1 4 6**

4 6 9 7 = ***Sbord(s,ctr)***

We have: **3<4, 1<6, 4<9, 6<7**

Computing the border table

Prop. 1. Let *ctr* be binary constraint **satisfying the transitivity property**.

Then:

$$Lbord(ua,ctr)=\begin{cases} Lbord(u,ctr)+1 & \text{if } ctr(u[Lbord(u,ctr)],a) \text{ holds} \\ Lbord(Pbord(u,ctr)a,ctr) & \text{else} \end{cases}$$

Example : *s* = 3 1 4 6 9 7 and *ctr* = '<'

k	0 1 2 3 4 5
s[*k*]	3 1 4 6 9 7
Lbord(*s*[0..*k*])	0 0 1 2 3 4

Computes the border table by using previous recurrence formula.

Evaluating *min*(*P*)

Prop. 2. *Let r be a factor of a sequence s. Then per(s) ≥ per(r).*

- Fixed variables
- Valid for any binary relation
- *O*(*m*) (border table)

1 3 2

Example : *s* = • 5 • 0 2 1 0 • • 3 5 3 5 • • • ⇒ **pers(*s*) ≥ 3**.

Pruning *dom*(*P*)

Prop. 3. *Let* $s = V_0\ V_1\ \dots\ V_{m-1}$ *and* i, j *such* $(0 \leq i < j \leq m-1$ *and* $V_i \neq V_j)$
Then $(j-i)$ *can't be a period of s.*

- Fixed variables
- O(*m*) (for each variable which get fixed)
- Valid for all binary relation *ctr* by replacing $V_i \neq V_j$ by $\neg$ ctr(V_i, V_j).

Example : s = • **5** • • • • • **9** • • • ⇒ **(7–1) = 6 not a period of *s*.**

Position 1 **Position 7**

Pruning *dom*(*P*)

Prop. 4. *Let s = $V_0\ V_1\ \ldots\ V_{m-1}$ and p such that $1 \leq p < m-1$. If p is not a period of s then each number which* **exactly divides p** *can't be a period of s .*

- Fixed variables
- (table *forbid*[1..*m*])
- Valid for all binary relations satisfying the transitivity property

Example : *s* = • **5** • • • • • **9** • • •

↑ **Position 1** ↑ **Position 7**

6 not a period of *s*

⇒ **1,2 and 3 also not a period of *s*.**

Notation

Let $s = V_0\ V_1\ \ldots\ V_{m-1}$ and p such that $1 \le p < m-1$.

$I_i^p = \bigcap_{\substack{0 \le k \le m-1 \\ k \equiv i \bmod p}} dom(V_k)$ ($i \in \{0,1,\ldots,p-1\}$) is the **intersection** of the domains of the variables which belong to the ***i*** **th group** of variables according to *p*.

Example : $s = V_0\ V_1\ V_2\ V_3\ V_4\ V_5\ V_6\ V_7\ V_8\ V_9\ V_{10}$ and $p = 3$

$$I_0^3 = dom(V_0) \cap dom(V_3) \cap dom(V_6) \cap dom(V_9)$$

$$I_1^3 = dom(V_1) \cap dom(V_4) \cap dom(V_7) \cap dom(V_{10})$$

$$I_2^3 = dom(V_2) \cap dom(V_5) \cap dom(V_8)$$

Pruning *dom*(*P*)

Prop. 5. *Let s = $V_0 V_1 \ldots V_{m-1}$ and p such that $1 \leq p < m-1$.*

*(p not **a** period of s)* ⇔ *(∃ i ∈ {0,1,…,p-1}* $I_i^p = \varnothing$ *).*

- **Fixed variables**
- **At most $m \cdot (m-1)/2$ domain intersections**
- **Valid for all relations *ctr* for which one provides a NSC for checking if a conjunction of constraints** $\bigwedge_{\substack{0 \leq k \leq m-1 \\ k \equiv i \bmod p}} ctr(V_k, V_{k+p})$ **$i \in \{0,1,\ldots,p\text{-}1\}$ has at most one solution.**

Example :

	V_0	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}
s =	{1, 2}	•	•	•	{0, 2}	•	•	•	{0, 1}	•	•

I_0^4 = *dom*(V_0) ∩ *dom*(V_4) ∩ *dom*(V_8) = ∅ ⇒ **4 not a period of *s*.**

Pruning *max*(*P*)

Prop. 6. *Let* $s = V_0\ V_1\ \ldots\ V_{m-1}$, *b such that* $1 \leq b < \lfloor m/2 \rfloor$ *and p the period of* $s\,[0..b-1]\ s\,[m-b..m-1]$.

Then $per(s) \leq \lceil b/p \rceil \cdot p + m - 2 \cdot b$.

- Fixed variables
- Valid for all binary relations if p ≥ b.
  (else valid if the binary relation satisfies the transitivity property)
- $O(m)$

Example : *s* = 3 1 • • 8 • • • 2 3 1

3 1 3 1 ha a period = 2 ⇒ $per(s) \leq \lceil 2/2 \rceil \cdot 2 + 11 - 2 \cdot 2 = 9$.

Pruning *max*(*P*)

Prop. 7. *Let* $s = V_0\ V_1\ \ldots\ V_{m-1}$ *and* $s' = s\,[0..b\text{-}1]\ s\,[m\text{-}b..m\text{-}1]$, *where b is the largest integer such that we have exactly one non-fixed variable in s'.*

Then $per(s) \leq p = \max_{v \in dom(V)} \left(\lceil b / p_v \rceil \cdot p_v \right) + m - 2 \cdot b$ where $p_v = per(s')$ when $V=v$.

- **Not all variables are fixed**
- **Valid for all binary relations if** $\min_{v \in dom(V)} (p_v) \geq b$ **.**

  (else valid if the binary relation satisfies the transitivity property)

- **$O(m)$**

Pruning *max*(*P*)

Example (Prop. 7)

$V_0\ V_1\ V_2\ V_3\ V_4\ V_5\ V_6\ V_7\ V_8$

$s = 1\ \begin{bmatrix}1\\2\end{bmatrix}\ 2\ \begin{bmatrix}0\\1\\\cdots\\9\end{bmatrix}\ \begin{bmatrix}0\\1\\\cdots\\9\end{bmatrix}\ 2\ 1\ 1\ 2$

$V_0\ V_1\ V_2\ V_6\ V_7\ V_8$

Then $s' = 1\ \begin{bmatrix}1\\2\end{bmatrix}\ 2\ 1\ 1\ 2$

3

$s' = 1\ 1\ 2\ \ 1\ 1\ 2$

4

$s' = 1\ 2\ 2\ \ 1\ 1\ 2$

$\Rightarrow$ ***per*(*s*) ≤ *max*(3,4) + 9 – 2 ·3 = 7.**

Pruning *dom*(*P*)

Prop. 8. *Let s* = $V_0 V_1 \ldots V_{m-1}$ *et* $p \in \{2,3,\ldots,m\}$ and assume each I_k^p is reduced to one single value. *If there exist a period q* < *p of* $I_0^p I_1^p \ldots I_{p-1}^p$ *where q divides exactly p then p is not the period of s.*

- **Not all variables are fixed**
- **$O(p)$ if $I_0^p I_1^p \ldots I_{p-1}^p$ already computed**
- **Valid if the binary relation satisfies the equivalence property.**

Pruning *dom*(*P*)

Example (Prop. 8):

	V_0	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	V_{11}
$s =$	0, 1, …, 9	0, 1, …, 9	1	0	1	0, 1, …, 9	0, 1, …, 9	0, 1, …, 9	0, 1, …, 9	0	0, 1, …, 9	0, 1, …, 9

For ***p* = 4** : $I_0^4 = I_2^4 = \{1\},\ I_1^4 = I_3^4 = \{0\}$

$per(I_0^4 I_1^4 I_2^4 I_3^4) = per(1010)$ ⇒ **4 not the period of *s*.**

Pruning *dom*(*P*)

Prop. 9. *Let* $s = V_0\ V_1\ \dots\ V_{m-1}$ *containing a factor* a^k *(k>1) where a stands for an element of the alphabet A and p such that* $1 \leq p < m-1$.

Then, pers(s) can't be equal to 2,3,…k.

- **Fixed variables**
- **Weak form of Prop. 8**
- $O(m)$

Example : s = • **0 0 0 0** • • contains the factor $\mathbf{0}^k$

⇒ ***pers(s)*** **can't be equal to 2,3,4**.

Pruning Variables from *dom*(*P*)

Prop. 10. *Let* $s = V_0\ V_1\ \ldots\ V_{m-1}$.

1. *For all* i *such that* $max(P) \leq i \leq m-1$, *we remove from* $dom(V_i)$ *all values that don't belong to* $\bigcup_{\substack{0 \leq l < i \\ (i-l) \in dom(P)}} dom(V_l)$.

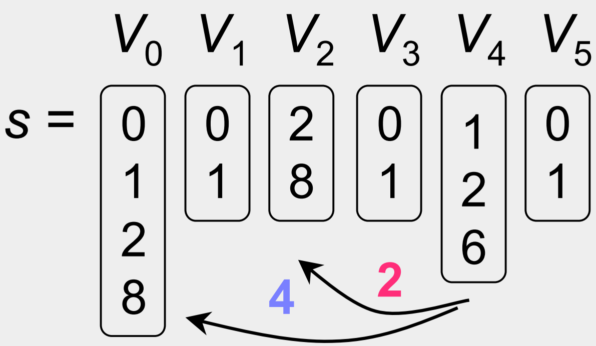
2. *For all* j *such that* $0 \leq j \leq m-1 - max(P)$, *we remove from* $dom(V_j)$ *all values that don't belong to* $\bigcup_{\substack{j < k \leq m-1 \\ (k-j) \in dom(P)}} dom(V_k)$.

Pruning Variables from *dom*(*P*)

- Not all variables are fixed
- Valid for all binary relations for which one can provide AC for the pattern:

$$ctr(X,Y_1) \wedge ctr(X,Y_2) \wedge \ldots \wedge ctr(X,Y_n) \quad .$$

Example :



and assume $dom(P)= \{2,4\}$.

Prop. 10 (1). prune V_i for $4 \le i \le 5$:

- V_4 : $dom(V_0) \cup dom(V_2) = \{0,1,2,8\}$ and $6 \in dom(V_4) \Rightarrow$ **remove 6 de $dom(V_4)$**
- V_5 : $dom(V_1) \cup dom(V_3) = \{0,1\} = dom(V_5) \Rightarrow$ **no pruning**

Pruning Variables from *dom*(*P*)

Prop. 11. *Let q = lcm(dom(P)) and assume q < m.*

For each i $\in$ *{0,1,…, min(q, m-q) - 1)}, the domains of the variables*

$V_i, V_{i+q}, \ldots, V_{i+\lfloor (m-i-1)/q \rfloor \cdot q}$ *are restricted to* I_i^q .

- Not all variables are fixed
- Derived from prop. 4 and 5
- Valid for all transitive binary relations for which one provides a filtering algorithm for handling a conjunction of constraints :

$$\bigwedge_{\substack{0 \le k \le m-1 \\ k \equiv i \bmod q}} ctr(V_k, V_{k+q}) \quad i \in \{0,1,\ldots,q\text{-}1\} \ .$$

Pruning Variables from *dom*(*P*)

Example (Prop. 11) : *s* =

V_0	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9
1 2	0 1 2	1 2	1 2	1 2	0 2	1 4	0 2	0 2	1 2

Assume that *dom*(*P*) = {1,2,3}. Then *lcm*(*dom*(*P*)) = 6.

⇒
- dom(V_0) and dom(V_6) are restricted to dom(V_0) ∩ dom(V_6) = {1}
- dom(V_1) is restricted to dom(V_1) ∩ dom(V_7) = {0,2}
- dom(V_2) and dom(V_8) are restricted to dom(V_2) ∩ dom(V_8) = {2}
- dom(V_3) and dom(V_9) are not restricted (dom(V_3) ∩ dom(V_9) = {1,2})

⇒ s =

1	**0 2**	**2**	1 2	1 2	0 2	**1**	0 2	**2**	1 2

Pruning Variables from *dom*(*P*)

Prop. 12. *Let s =* $V_0\ V_1\ \ldots\ V_{m-1}$ *containing a factor* $a^k\ V_i\ a^{k'}$ *where a is an element of an alphabet A, k and k' > 1 and* V_i *is not fixed.*
If dom(P) $\subseteq$ {2,3,…, *k* + *k'*+1} *then V can't take value a.*

- **Not all variables are fixed**
- **Derived from Prop. 9.**
- $O(m)$

Example : $s = 0\ 0\ V_2\ 0\ V_4\ V_5$ and *dom*(*P*) = {2,3,4}

s contains the factor $0^2\ V_2\ 0^1 \Rightarrow V_2$ can't take value 0.

Pruning Variables from *min*(*P*)

Prop. 13. *Let* $s = V_0\ V_1\ \dots\ V_{m-1}$ *and* $s' = s[0..b-1]\ s[m-b..m-1]$ *where b is the largest integer* $1 \leq b < \lfloor m/2 \rfloor$ *such that all variables of s' but one (V) are fixed. Let* $v \in dom(V)$ and *p the period of s' for V = v. If* $\lceil b/p \rceil \cdot p + m - 2 \cdot b < min(P)$ *then V can't be assigned to value v.*

- **Fixed variables**
- **Derived from Prop.6**
- $O(m)$

Example : s **= 1 2 3 6 • 9 1** V_7 **3** and assume $6 < min(P)$.

$\Rightarrow$ V_7 can't be assigned to value **2**.

(else *s* would have a border *b* = **1 2 3** and therefore (Prop. 6) per(*s*) ≤ 6)

- CONTEXT
- WORK ON SEQUENCES
- FILTERING FOR THE *PERIOD* CONSTRAINT
- **CONCLUSION**

Conclusion

RESULTS:

- A constraint which can be **parametrazided** by a relation,
- Filtering **depends of the properties** of this relation.

OPEN-QUESTIONS:

- **Arc-consistency**,
- **Efficient** implementation for Prop.5, 8, 10, 11 ?