5.344 sequence_folding

DESCRIPTION LINKS GRAPH AUTOMATON

Origin J. Pearson

Constraint sequence_folding(LETTERS)

Argument LETTERS : collection(index−int, next−dvar)

Restrictions
|LETTERS| ≥ 1
required(LETTERS, [index, next])
LETTERS.index ≥ 1
LETTERS.index ≤ |LETTERS|
increasing_seq(LETTERS, index)
LETTERS.next ≥ 1
LETTERS.next ≤ |LETTERS|

Purpose Express the fact that a sequence is folded in a way that no crossing occurs. A sequence is modelled by a collection of letters. For each letter l_1 of a sequence, we indicate the next letter l_2 located after l_1 that is directly in contact with l_1 (l_1 itself if such a letter does not exist).

Example

$$\left(\left\langle\begin{array}{ll} \texttt{index}-1 & \texttt{next}-1, \\ \texttt{index}-2 & \texttt{next}-8, \\ \texttt{index}-3 & \texttt{next}-3, \\ \texttt{index}-4 & \texttt{next}-5, \\ \texttt{index}-5 & \texttt{next}-5, \\ \texttt{index}-6 & \texttt{next}-7, \\ \texttt{index}-7 & \texttt{next}-7, \\ \texttt{index}-8 & \texttt{next}-8, \\ \texttt{index}-9 & \texttt{next}-9 \end{array}\right\rangle\right)$$

Figure 5.696 gives the folded sequence associated with the previous example. Each number represents the index of an item. The sequence_folding constraint holds since no crossing occurs.

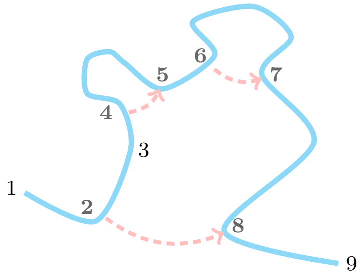
Typical
|LETTERS| > 2
range(LETTERS.next) > 1

Usage Motivated by RNA folding [167].

See also **implies (items to collection):** lex_alldifferent, lex_chain_less.

Keywords **application area:** bioinformatics.

characteristic of a constraint: automaton, automaton without counters, reified automaton constraint.



LETTERS

index − 1	next − 1
index − 2	next − 8
index − 3	next − 3
index − 4	next − 5
index − 5	next − 5
index − 6	next − 7
index − 7	next − 7
index − 8	next − 8
index − 9	next − 9

Figure 5.696: Folded sequence (in blue) of the **Example** slot: links from a letter to a distinct letter are represented by a dashed arc, while self-loops are not drawn

combinatorial object: sequence.

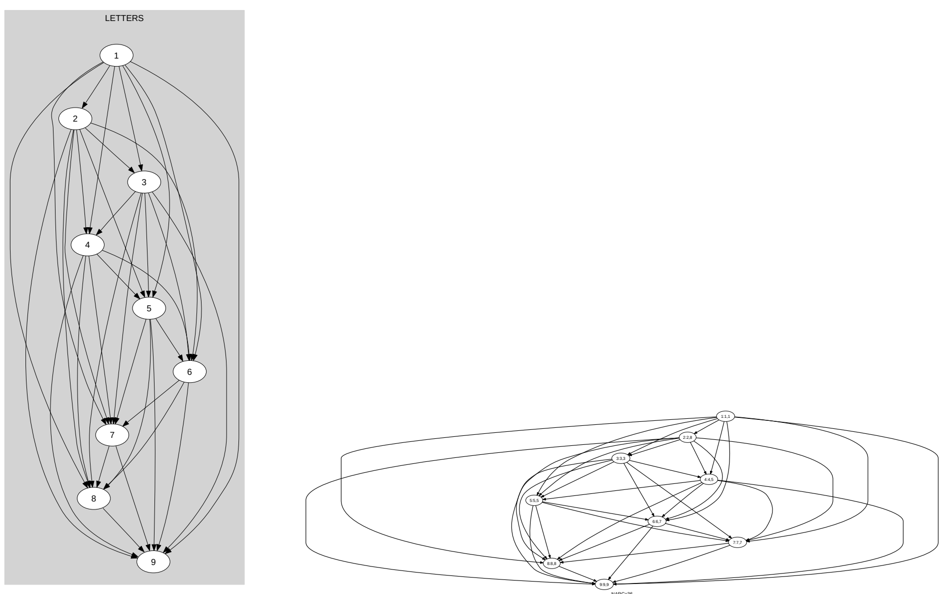
constraint type: decomposition.

geometry: geometrical constraint.

Arc input(s)	LETTERS
Arc generator	$SELF \mapsto$ `collection`(`letters`)
Arc arity	1
Arc constraint(s)	`letters.next` $\geq$ `letters.index`
Graph property(ies)	$\mathbf{NARC} = \lvert$LETTERS$\rvert$

Arc input(s)	LETTERS
Arc generator	$CLIQUE(<) \mapsto$ `collection`(`letters1`, `letters2`)
Arc arity	2
Arc constraint(s)	$\bigvee\begin{pmatrix} \texttt{letters2.index} \geq \texttt{letters1.next}, \\ \texttt{letters2.next} \leq \texttt{letters1.next} \end{pmatrix}$
Graph property(ies)	$\mathbf{NARC} = \lvert\text{LETTERS}\rvert * (\lvert\text{LETTERS}\rvert - 1)/2$

Graph model Parts (A) and (B) of Figure 5.697 respectively show the initial and final graph associated with the **Example** slot. Since we use the **NARC** graph property, the arcs of the final graph are stressed in bold.



(A) **(B)**

Figure 5.697: Initial and final graph of the `sequence_folding` constraint

Signature Consider the first graph constraint. Since we use the $SELF$ arc generator on the LETTERS collection the maximum number of arcs of the final graph is equal to $\lvert$LETTERS$\rvert$. Therefore we can rewrite the graph property $\mathbf{NARC} = |\mathtt{LETTERS}|$ to $\mathbf{NARC} \geq |\mathtt{LETTERS}|$ and simplify $\overline{\underline{\mathbf{NARC}}}$ to $\overline{\mathbf{NARC}}$.

Consider now the second graph constraint. Since we use the $CLIQUE(<)$ arc generator on the LETTERS collection the maximum number of arcs of the final graph is equal to $|\mathtt{LETTERS}| \cdot (|\mathtt{LETTERS}| - 1)/2$. Therefore we can rewrite the graph property $\mathbf{NARC} = |\mathtt{LETTERS}| \cdot (|\mathtt{LETTERS}| - 1)/2$ to $\mathbf{NARC} \geq |\mathtt{LETTERS}| \cdot (|\mathtt{LETTERS}| - 1)/2$ and simplify $\overline{\underline{\mathbf{NARC}}}$ to $\overline{\mathbf{NARC}}$.

Automaton Figure 5.698 depicts the automaton associated with the `sequence_folding` constraint. Consider the i^{th} and the j^{th} $(i < j)$ items of the collection LETTERS. Let $\texttt{INDEX}_i$ and $\texttt{NEXT}_i$ respectively denote the `index` and the `next` attributes of the i^{th} item of the collection LETTERS. Similarly, let $\texttt{INDEX}_j$ and $\texttt{NEXT}_j$ respectively denote the `index` and the `next` attributes of the j^{th} item of the collection LETTERS. To each quadruple $(\texttt{INDEX}_i, \texttt{NEXT}_i, \texttt{INDEX}_j, \texttt{NEXT}_j)$ corresponds a signature variable $S_{i,j}$, which takes its value in $\{0, 1, 2\}$, as well as the following signature constraint:

$(\texttt{INDEX}_i \leq \texttt{NEXT}_i) \wedge (\texttt{INDEX}_j \leq \texttt{NEXT}_j) \wedge (\texttt{NEXT}_i \leq \texttt{NEXT}_j) \Leftrightarrow S_{i,j} = 0 \wedge$

$(\texttt{INDEX}_i \leq \texttt{NEXT}_i) \wedge (\texttt{INDEX}_j \leq \texttt{NEXT}_j) \wedge (\texttt{NEXT}_i > \texttt{INDEX}_j) \wedge (\texttt{NEXT}_j \leq \texttt{NEXT}_i) \Leftrightarrow S_{i,j} = 1$.

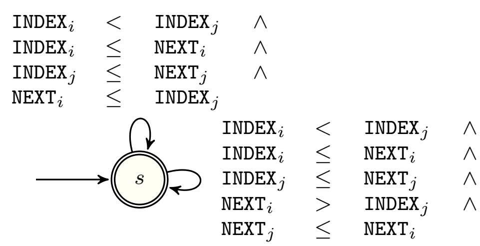


Figure 5.698: Automaton of the `sequence_folding` constraint