

## 5.40 atmost1

	DESCRIPTION	LINKS
<b>Origin</b>	[365]	
<b>Constraint</b>	atmost1(SETS)	
<b>Synonym</b>	pair_atmost1.	
<b>Argument</b>	SETS : collection(s—svar, c—int)	
<b>Restrictions</b>	required(SETS, [s, c]) SETS.c ≥ 1	
<b>Purpose</b>	Given a collection of set variables $s_1, s_2, \dots, s_n$ and their respective cardinality $c_1, c_2, \dots, c_n$ , the <code>atmost1</code> constraint forces the following two conditions: <ul style="list-style-type: none"> <li>• <math>\forall i \in [1, n] :  s_i  = c_i</math>,</li> <li>• <math>\forall i, j \in [1, n] (i &lt; j) :  s_i \cap s_j  \leq 1</math>.</li> </ul>	
<b>Example</b>	$\left( \left\langle \begin{array}{ll} s - \{5, 8\} & c - 2, \\ s - \{5\} & c - 1, \\ s - \{5, 6, 7\} & c - 3, \\ s - \{1, 4\} & c - 2 \end{array} \right\rangle \right)$ <p>The <code>atmost1</code> constraint holds since:</p> <ul style="list-style-type: none"> <li>• <math> \{5, 8\}  = 2,  \{5\}  = 1,  \{5, 6, 7\}  = 3,  \{1, 4\}  = 2</math>.</li> <li>• <math> \{5, 8\} \cap \{5\}  \leq 1,  \{5, 8\} \cap \{5, 6, 7\}  \leq 1,  \{5, 8\} \cap \{1, 4\}  \leq 1,</math>  <math> \{5\} \cap \{5, 6, 7\}  \leq 1,  \{5\} \cap \{1, 4\}  \leq 1,</math>  <math> \{5, 6, 7\} \cap \{1, 4\}  \leq 1</math>.</li> </ul>	
<b>Typical</b>	SETS  > 1	
<b>Symmetries</b>	<ul style="list-style-type: none"> <li>• Items of SETS are <a href="#">permutable</a>.</li> <li>• All occurrences of two distinct values of SETS.s can be <a href="#">swapped</a>; all occurrences of a value of SETS.s can be <a href="#">renamed</a> to any unused value.</li> </ul>	
<b>Arg. properties</b>	<a href="#">Contractible</a> wrt. SETS.	
<b>Remark</b>	When we have only two set variables the <code>atmost1</code> constraint was called <code>pair_atmost1</code> in [428].	

- Algorithm** C. Bessière *et al.* have shown in [68] that it is NP-hard to enforce bound consistency for the `atmost1` constraint. Consequently, following the first filtering algorithm from A. Sadler and C. Gervet [365], W.-J. van Hove and A. Sabharwal have proposed an algorithm that enforces `bound-consistency` when the `atmost1` constraint involves only two sets variables [428].
- Systems** `at_most1` in **MiniZinc**.
- Keywords** **constraint arguments:** constraint involving set variables.  
**constraint type:** predefined constraint.  
**filtering:** bound-consistency.