

## 5.1 abs\_value

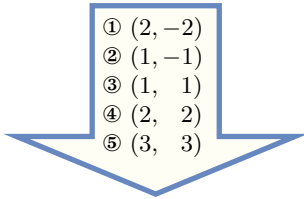
	DESCRIPTION	LINKS
<b>Origin</b>	Arithmetic.	
<b>Constraint</b>	<code>abs_value(Y, X)</code>	
<b>Usual name</b>	<code>abs</code>	
<b>Synonym</b>	<code>absolute_value.</code>	
<b>Arguments</b>	<code>Y : dvar</code> <code>X : dvar</code>	
<b>Restriction</b>	$Y \geq 0$	
<b>Purpose</b>	Enforce the fact that the first variable is equal to the absolute value of the second variable.	
<b>Example</b>	<div style="border: 1px solid black; padding: 2px; display: inline-block;">(8, -8)</div> The <code>abs_value</code> constraint holds since 8 is equal to $ -8 $ .	
<b>All solutions</b>	Figure 5.1 gives all solutions to the following non ground instance of the <code>abs_value</code> constraint: $Y \in [1, 6]$ , $X \in [-2, 3]$ , <code>abs_value(Y, X)</code> .	
		
<b>Arg. properties</b>	Functional dependency: Y determined by X.	
<b>Systems</b>	<code>abs</code> in <b>Choco</b> , <code>abs</code> in <b>Gecode</b> .	
<b>See also</b>	<b>implied by:</b> <code>eq</code> . <b>implies:</b> <code>geq</code> , <code>zero_or_not_zero</code> . <b>implies (if swap arguments):</b> <code>opposite_sign</code> , <code>zero_or_not_zero</code> .	

Figure 5.1: All solutions corresponding to the non ground example of the `abs_value` constraint of the **All solutions** slot

**Keywords**

**constraint arguments:** binary constraint, pure functional dependency.

**constraint type:** predefined constraint, arithmetic constraint.

**filtering:** arc-consistency.

**modelling:** functional dependency.