

contraction

[/Reference manual/Z-related commands/In situ replacement commands](#)

The *contraction* command performs *in situ* replacements that are each like *expansion* steps but performed in the opposite direction. The earliest applicable *contraction* step in the list is the one that is applied.

0.0.0.1. Predicates

$$\begin{aligned}
 \neg p_1 \vee p_2 &\implies p_1 \Rightarrow p_2 \\
 p_1 \Rightarrow p_2 \wedge p_2 \Rightarrow p_1 &\implies p_1 \Leftrightarrow p_2 \\
 \neg (p_1 \Leftrightarrow p_2) &\implies p_1 \not\equiv p_2 \\
 \forall x : \tau \bullet x \in e_1 \Leftrightarrow x \in e_2 &\implies e_1 = e_2
 \end{aligned}$$

where e_1 and e_2 are expressions of type $\mathbb{P} \tau$.

$$\exists s \bullet e_1 = e_2 \implies e_1 \in \{s \bullet e_2\}$$

where no references in e_1 are bound to declarations in s (the variable release problem).

$$\begin{aligned}
 e.1 \in e_1 \wedge \dots \wedge e.n \in e_n &\implies e \in (e_1 \times \dots \times e_n) \\
 e.1 = e_1 \wedge \dots \wedge e.n = e_n &\implies e = (e_1, \dots, e_n) \\
 e.i_1 = e_1 \wedge \dots \wedge e.i_n = e_n &\implies e = \langle i_1 == e_1, \dots, i_n == e_n \rangle \\
 e = e_1 \vee \dots \vee e = e_n &\implies e \in \{e_1, \dots, e_n\} \\
 e \in (relation_operator) &\implies relation\ operator\ application
 \end{aligned}$$

0.0.0.2. Expressions

Page 2 of 2

◀ ▶

Go Back

Full Screen

Close

Quit

$$\begin{aligned}
 \{i_1 : e_1; \dots; i_n : e_n\} &\Longrightarrow e_1 \times \dots \times e_n \\
 \neg e_1 \vee e_2 &\Longrightarrow e_1 \Rightarrow e_2 \\
 e_1 \Rightarrow e_2 \wedge e_2 \Rightarrow e_1 &\Longrightarrow e_1 \Leftrightarrow e_2 \\
 \neg (e_1 \Leftrightarrow e_2) &\Longrightarrow e_1 \not\equiv e_2 \\
 (i.1, \dots, i.n) &\Longrightarrow i \\
 \langle i_1 == i.i_1, \dots, i_n == i.i_n \rangle &\Longrightarrow i
 \end{aligned}$$

1. Tactic example

“contraction” $p_1 \ p_2$

This example applies the *contraction* command to predicates p_1 and p_2 .

IT 5-Apr-2000