expansion

/Reference manual/Z-related commands/In situ replacement commands

The *expansion* command performs the following *in situ* replacements. The inverses of some of the inferences are performed by the *contraction* command.

0.0.0.1. Predicates

$$
\begin{array}{rcl}
p_1 \Rightarrow p_2 & \Longrightarrow & \neg\, p_1 \vee p_2 \\
p_1 \Leftrightarrow p_2 & \Longrightarrow & (p_1 \Rightarrow p_2) \wedge (p_2 \Rightarrow p_1) \\
p_1 \veebar p_2 & \Longrightarrow & \neg\, (p_1 \Leftrightarrow p_2) \\
p_1 ;\ p_2 & \Longrightarrow & p_1 \wedge p_2 \\
p_1\ NL\ p_2 & \Longrightarrow & p_1 \wedge p_2 \\
\exists_1\, ds \mid p_1 \bullet p_2 & \Longrightarrow & \exists\, ds \mid p_1 \bullet (p_2 \wedge (\forall\, ds{+} \mid p_1{+} \bullet p_2{+} \Rightarrow \mathit{charac}\ ds \\
e \in \{\} & \Longrightarrow & \mathit{false} \\
e \in \{e_1, \ldots, e_n\} & \Longrightarrow & e = e_1 \vee \ldots \vee e = e_n \\
e_1 \in \{s \bullet e_2\} & \Longrightarrow & \exists\, s \bullet e_1 = e_2 \\
e_1 \in \lambda\, t \bullet e_2 & \Longrightarrow & \exists\, t \bullet e_1 = (\mathit{chartuple}\ t, e_2) \\
e_1 \in \mathbb{P}\, e_2 & \Longrightarrow & \forall\, z : e_1 \bullet z \in e_2 \\
e \in (e_1 \times \ldots \times e_n) & \Longrightarrow & e.1 \in e_1 \wedge \ldots \wedge e.n \in e_n \\
e_1 \in e_2 & \Longrightarrow & \exists\, e_2 \bullet e_1 = \theta\, e_2 \text{ where } e_1 \text{ is of the type of a bindi} \\
e = (e_1, \ldots, e_n) & \Longrightarrow & e.1 = e_1 \wedge \ldots \wedge e.n = e_n \\
e = \langle\!|\ i_1 == e_1, \ldots, i_n = e_n\ |\!\rangle & \Longrightarrow & e.i_1 = e_1 \wedge \ldots \wedge e.i_n = e_n \\
e_1 = e_2 & \Longrightarrow & \forall\, s : \tau \bullet s \in e_1 \Leftrightarrow s \in e_2 \text{ where } e_1 \text{ and } e_2 \text{ are of s} \\
e & \Longrightarrow & \theta\, e \in e
\end{array}
$$

0.0.0.2. Expressions

$$i \quad \Longrightarrow \quad e_1$$

where i is bound either to the abbreviation declaration $i == e_1$ or to the generic abbreviation declaration $i == e_2$ where e_1 is the appropriate instantiation of e_2.

$$i \quad \Longrightarrow \quad [ds \mid p]$$

where i is bound to a schema definition paragraph.

$$i \quad \Longrightarrow \quad (i.1, ..., i.n)$$

where i is bound to a colon declaration of cartesian product type.

$$i \quad \Longrightarrow \quad \langle\!| \; i_1 == i.i_1, ..., i_n == i.i_n \; |\!\rangle$$

where i is bound to a colon declaration of schema type.

$$\theta \, e+ \quad \Longrightarrow \quad \langle\!| \; i_1 == i_1+, ..., i_n == i_n+ \; |\!\rangle$$

where $i_1 ... i_n$ are the variables declared in schema e.

$$e_1 \;\; e_2 \quad \Longrightarrow \quad (\mu \, x : \tau \mid (e_2, x) \in e_1)$$

where $e_1 \;\; e_2$ is either a juxtaposed application or a function operator application.

$$
\begin{array}{rcl}
2 & \Longrightarrow & 1+1 \\
3 & \Longrightarrow & 2+1 \\
4 & \Longrightarrow & 3+1 \\
5 & \Longrightarrow & 4+1 \\
6 & \Longrightarrow & 5+1 \\
7 & \Longrightarrow & 6+1 \\
8 & \Longrightarrow & 7+1 \\
9 & \Longrightarrow & 8+1 \\
bc & \Longrightarrow & b+b+b+b+b+b+b+b+b+b+c
\end{array}
$$

where bc is a number literal in which c is the last digit and b is the preceding digits.

$$
\begin{array}{rcl}
e_1 \times \ldots \times e_n & \Longrightarrow & \{ i_1 : e_1;\ \ldots;\ i_n : e_n \} \\
"c_1 \ldots c_n" & \Longrightarrow & (1, c_1), \ldots, (n, c_n) \\
\mathrm{pre}\ e & \Longrightarrow & \exists\, nextstatevars;\ outputvars \bullet e \\
\mathrm{post}\ e & \Longrightarrow & e \\
e_1 \Rightarrow e_2 & \Longrightarrow & \neg\, e_1 \lor e_2 \\
e_1 \Leftrightarrow e_2 & \Longrightarrow & (e_1 \Rightarrow e_2) \land (e_2 \Rightarrow e_1) \\
e_1 \veebar e_2 & \Longrightarrow & \neg\, (e_1 \Leftrightarrow e_2) \\
e_1 \mathbin{;} e_2 & \Longrightarrow & (e_1[common\ \ names\ \ renaming\ \ matching\ \ next\ \ state\ \ nam \\
 & & e_2[common\ names\ renaming\ matching\ current\ state\ names]) \\
e_1 \gg e_2 & \Longrightarrow & (e_1[common\ names\ renaming\ matching\ output\ and\ next\ state\ na \\
 & & e_2[common\ names\ renaming\ matching\ input\ and\ current\ state\ n \\
e_1 \upharpoonright e_2 & \Longrightarrow & \exists\, sig(e_1) \setminus sig(e_2) \bullet e_1 \land e_2 \\
e_1 \oplus e_2 & \Longrightarrow & e_1 \land \neg\ \mathrm{pre}\ e_2 \lor e_2 \\
e \setminus (i_1, \ldots, i_n) & \Longrightarrow & \exists\, i_1 : \tau\ i_1;\ \ldots;\ i_n : \tau\ i_n \bullet e
\end{array}
$$

1. Tactic example

"expansion" p e

This example applies the *expansion* command to predicate p and expression e.

IT 5-Apr-2000