properties

/Reference manual/Z-related commands/Proof rule commands

The *properties* command is applicable to a name in a goal, where the name refers to a top-level declaration in the specification about which the specification provides some constraints (excepting names declared in the specification using the == notation, and also the names of schema paragraphs, for which use the *expansion* command). It makes the specification's constraints on that name available as antecedents in the sub-goal. Constraints that are newline-separated in the specification become comma-separated in the sub-goal. If the name is declared generically, the new antecedents are the specification's constraints instantiated according to the instantiation of the name in the goal.

In the case of a name that is declared in an axiomatic paragraph, *properties* makes available the constraint implicit in the declaration, along with the explicit constraints from the axiomatic paragraph.

$$\begin{array}{|l}
i : e \\
\hline
ps
\end{array}$$

$$\frac{\mid i \in e, ps \vdash ?}{\vdash ?}$$

Global constraints on the name can be made available using the separate *constraint* command.

In the case of a name that is declared in a free type paragraph (as the name of either a free type, an element, or an injection), *properties* makes available the constraints from the paragraphs that the Z standard defines the free type paragraph as abbreviating.

$$
\begin{array}{l}
f_1 ::= h_{1,1} ... h_{1,m_1} \mid g_{1,1} \langle\!\langle e_{1,1} \rangle\!\rangle ... g_{1,n} \langle\!\langle e_{1,n} \rangle\!\rangle \\
\& ... \& \\
f_r ::= h_{r,1} ... h_{r,m_r} \mid g_{r,1} \langle\!\langle e_{r,1} \rangle\!\rangle ... g_{r,n_r} \langle\!\langle e_{r,n_r} \rangle\!\rangle
\end{array}
$$

$-- \, membership$

$h_{i,j} \in f_i,$

$g_{i,k} \in \mathbb{P}(e_{i,k} \times f_i),$

$-- \, totality$

$\forall\, x : e_{i,k} \bullet \exists_1\, y : g_{i,k} \bullet y.1 = x,$

$-- \, injectivity$

$\forall\, x, y : e_{i,k} \mid g_{i,k}\ x = g_{i,k}\ y \bullet x = y,$

$-- \, disjointness$

$$
\begin{array}{l}
\forall\, b_1, b_2 : \mathbb{N} \bullet \\
\quad \forall\, e : f_i \mid \\
\qquad (b_1 = 1 \land e = h_{i,1} \lor \ldots \lor b_1 = m_i \land e = h_{i,m_i} \lor \\
\qquad b_1 = m_i + 1 \land e \in i : g_{i,1} \bullet i.2 \lor \ldots \lor b_1 = m_i + n_i \land e \in i : g_{i,n_i} \bullet i.2) \land \\
\qquad (b_2 = 1 \land e = h_{i,1} \lor \ldots \lor b_2 = m_i \land e = h_{i,m_i} \lor \\
\qquad b_2 = m_i + 1 \land e \in i : g_{i,1} \bullet i.2 \lor \ldots \lor b_2 = m_i + n_i \land e \in i : g_{i,n_i} \bullet i.2) \bullet \\
\qquad\quad b_1 = b_2
\end{array}
$$

$-- \, induction$

$$
\begin{array}{l}
\forall\, f_1' : \mathbb{P}\, f_1;\ \ldots;\ f_r' : \mathbb{P}\, f_r \mid \\
\quad h_{1,1} \in f_1' \land \ldots \land h_{1,m_1} \in f_1' \land \\
\quad \ldots \land \\
\quad h_{r,1} \in f_r' \land \ldots \land h_{r,m_r} \in f_r' \land \\
\quad (\forall\, y_{1,1} : \mu\, f_1 == f_1';\ \ldots;\ f_r == f_r' \bullet e_{1,1} \bullet g_{1,1}\ \ y1, 1 \in f_1') \land \\
\quad \ldots \land \\
\quad (\forall\, y_{1,n_1} : \mu\, f_1 == f_1';\ \ldots;\ f_r == f_r' \bullet e_{1,n_1} \bullet g_{1,n_1}\ \ y_{1,n_1} \in f_1') \land \\
\quad \ldots \land \\
\quad (\forall\, y_{r,1} : \mu\, f_1 == f_1';\ \ldots;\ f_r == f_r' \bullet e_{r,1} \bullet g_{r,1}\ \ y_{r,1} \in f_r') \land \\
\quad \ldots \land \\
\quad (\forall\, y_{r,n_r} : \mu\, f_1 == f_1';\ \ldots;\ f_r == f_r' \bullet e_{r,n_r} \bullet g_{r,n_r}\ \ yr, n_r \in f_r') \bullet \\
\qquad f_1' = f_1 \land \ldots \land f_r' = f_r
\end{array}
$$

$\vdash ?$

1. Tactic example

$$\textit{“properties”}\ e_1\ \ e_2$$

This example applies the *properties* command to expressions e_1 and e_2.

IT 27-Sep-1999