quantification tac

The *quantification tac* command introduces extra constraints on the values of quantified declarations, equating their values to those of particular expressions. To make this be an equivalence transformation, the original quantified predicate is logically combined with the new one.

$$\forall\, i : e_0;\ e;\ \ldots \mid p_1 \bullet p_2 \quad \Longrightarrow \quad (\forall\, i : e_0;\ e;\ \ldots \mid p_1 \bullet p_2) \land (\forall\, i : e_0;\ e;\ \ldots \mid i = e_1 \land \theta e =$$
$$\exists\, i : e_0;\ e;\ \ldots \mid p_1 \bullet p_2 \quad \Longrightarrow \quad (\exists\, i : e_0;\ e;\ \ldots \mid p_1 \bullet p_2) \lor (\exists\, i : e_0;\ e;\ \ldots \mid i = e_1 \land \theta e =$$

There are two ways of supplying the expressions e_1, e_2 etc for the declarations to be quantified. In the first, cadiz prompts the user to enter values for each declaration using a sequence of dialogue boxes. In the case of a basic declaration, the prompt is the name of the declaration. In the case of a schema inclusion declaration, the prompt is θ of that schema. The expressions are requested in the same order as the declarations appear. The alternative is to have crossed suitable expressions, in that same order as they would be prompted for in the interactive way, before articulating the *quantification tac* command. The expressions must be crossed in the window where the *quantification tac* is requested. Their text is used, so the bindings of references are irrelevant. (There is no "half-way-house" where some expressions would be crossed and others prompted for by dialogue boxes.) In the interactive dialogue, entering just $ as the response is interpreted as omit quantifying the corresponding declaration: no constraint on the value of that declaration will be introduced. If the name in the prompt is the same as one

prompted for before in the same cadiz session, then the dialogue box is primed with the previous response.

The *quantification tac* command can be applied to only one quantified predicate at a time, because of the different interpretation of crossed selections.

The *quantification tac* command is not fundamental to the logical inference system: the same effect can be achieved by a composition of *cut apart* and *distribution*, and *absorption* commands.

1. Tactic example

"quantification tac" "{1,2}" "(x,y)" p_3

This example applies the *quantification tac* rule using expression x as the value for the first declaration and the expression y as the value for the second declaration to the predicate p_3. The first argument must be a set extension of numbers. The second argument must be a tuple, unless there is only one expression. If the string arguments are omitted from the tactic, dialogue boxes will be used. The joker bound to the quantified predicate remains bound to the entire result.

IT 21-Dec-1998