

# heuristic decision

[/Reference manual](#)/[Z-related commands](#)/[Decision procedure commands](#)

The *heuristic decision* command solves some predicates in non-linear arithmetic. These predicates can include multiplication and division of two variables in addition to the notations of linear arithmetic (see [linear decision](#)); they can also include predicates over enumerated free types, and predicates over empty schemas. Non-linear arithmetic predicates cannot in general be decided algorithmically. Instead, the *heuristic decision* command takes into account what constraints it can, then invokes a technique known as simulated annealing to search for a solution. The search commences at a random point, so a *heuristic decision* command that fails to find a solution on one occasion may succeed in finding a solution on another occasion. If it succeeds, it replaces each by either *true* or *false*.

See also the [heuristic solution](#) command.

## 1. Tactic example

*“heuristic decision” “number” p<sub>1</sub> p<sub>2</sub>*

This example applies the *heuristic decision* command to predicates  $p_1$  and  $p_2$ , using the number in the string literal to seed a random number generator to ensure repeatable behaviour.

*“heuristic decision”*  $p_1$   $p_2$

Page 1 of 2



Go Back

Full Screen

Close

Quit

This example applies the *heuristic decision* command to predicates  $p_1$  and  $p_2$ , using a random seed.

*IT 28-Sep-2001*