0[15] The notation \textbf{do }P\textbf{ while }b\textbf{ od } has been used as a loop construct that is executed as follows. First, \(P\) is executed; then \(b\) is evaluated, and if its value is \(\top\) then execution is repeated, and if its value is \(\bot\) then execution is finished. Let the program variable be integer variable \(x\). Prove that the specification
\[x': x \times 2^{\text{n}at}\]
is refined by the loop
\[
\text{do } x := x \times 2 \text{ while } x < 100 \text{ od}
\]
To prove \(S\) is refined by \textbf{do }P\textbf{ while }b\textbf{ od }, prove instead
\[S \iff P. \text{ if } b \text{ then } S \text{ else } ok \text{ fi}\]
So we prove
\[
(x': x \times 2^{\text{n}at} \iff x := x \times 2. \text{ if } x < 100 \text{ then } x': x \times 2^{\text{n}at} \text{ else } ok \text{ fi}) \quad \text{replace } ok
\]
\[
= (x': x \times 2^{\text{n}at} \iff x := x \times 2. \text{ if } x < 100 \text{ then } x': x \times 2^{\text{n}at} \text{ else } x' = x \text{ fi}) \quad \text{substitution}
\]
\[
= x': x \times 2^{\text{n}at} \iff \text{if } x \times 2 < 100 \text{ then } x': x \times 2^{\text{n}at} \text{ else } x' = x \times 2 \text{ fi} \quad \text{by cases}
\]
\[
= (x': x \times 2^{\text{n}at} \iff x \times 2 < 100 \land x' = x \times 2) \quad \text{specialization and}
\]
\[
\quad \land (x': x \times 2^{\text{n}at} \iff x \times 2 \geq 100 \land x' = x \times 2) \quad \text{specialization again}
\]
\[
\iff (x': x \times 2^{\text{n}at} \iff x': x \times 2 \times 2^{\text{n}at}) \land (x': x \times 2^{\text{n}at} \iff x' = x \times 2) \quad \text{nat1: } nat \text{ and } 1: nat
\]
\[
\iff (x': x \times 2^{\text{n}at+1} \iff x': x \times 2 \times 2^{\text{n}at}) \land (x': x \times 2^{\text{n}at} \iff x' = x \times 2) \quad \text{nat+1: } nat \text{ and } 1: nat
\]
\[
= T \land T
\]
\[
= T
\]

1 From the construction axiom \(0, 1\text{--few: few}\)
(a)[3] what elements are constructed?
§ \(0, 1\)
(b)[3] give three solutions (considering \textit{few} as the unknown).
§ solution \((0, 1)\) and solution \((0, 1/2, 1)\) and solution \((0, 1/3, 2/3, 1)\) and solution \textit{int} and solution \textit{rat} and solution \textit{real}
(c)[3] give the corresponding induction axiom.
§ \(0, 1\text{--B: } B \Rightarrow \text{few: } B\)
(d)[3] state which solution is specified by construction and induction.
§ solution \(0, 1\)

2 Let \(x\) and \(y\) be rational variables. Define program \textit{zot} by the fixed-point equation
\[\textit{zot} \iff \text{if } x = 0 \text{ then } y := 0 \text{ else } x := x / 2. \textit{zot} \text{ fi}\]
where / is rational division.
(a)[3] Add recursive time.
§ \textit{zot} \iff \text{if } x = 0 \text{ then } y := 0 \text{ else } x := x / 2. \text{ t := t+1. } \textit{zot} \text{ fi}
(b)[9] Give two solutions to this equation (with recursive time added) (considering \textit{zot} as the unknown). (No proof needed.)
§ \(x = 0 \Rightarrow x' = y' = 0 \land t' = t\)
\(\text{if } x = 0 \text{ then } x' = y' = 0 \land t' = t\text{ else } x' = y' = 17 \land t' = \infty \text{ fi}\)
(c)[3] The definition of \textit{zot} makes it a solution (fixed-point) of an equation. What axiom is needed to make \textit{zot} the strongest solution (strongest fixed-point)?
§ \((\forall x, y, t, x', y', t'. Z = \text{if } x = 0 \text{ then } y := 0 \text{ else } x := x / 2. \text{ t := t+1. } Z \text{ fi})\)
\(\Rightarrow (\forall x, y, t, x', y', t'. Z \Leftarrow \textit{zot})\)