## Semantics of machine operations To be able to check formally that operations work as they supposed to, we have to assign precise mathematical meaning (semantics) to them Operations in general are concerned with changing the local state and setting output variables

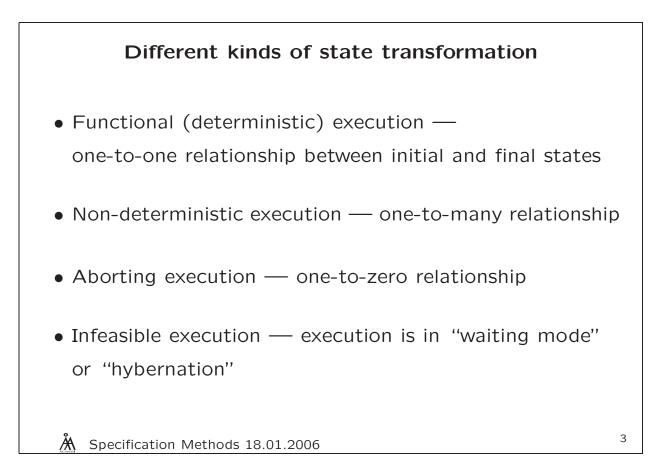
• A specification of an operation basically describes the relationship between initial (before) and final (after) states

A Specification Methods 18.01.2006

## Machine states

- Machine states include the combinations of all possible values of (input, output, local) machine variables
- Machine states can be modelled as values of cartesian product on machine variable types, for example,  $NAT \times NAT$  for two variables of natural numbers
- An operation is then state transformation described using Abstract Machine Notation

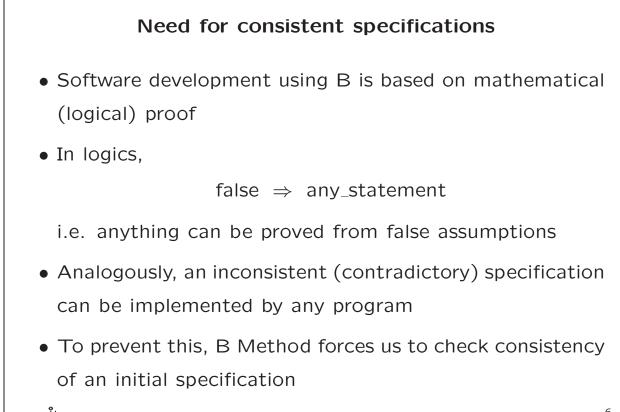
1

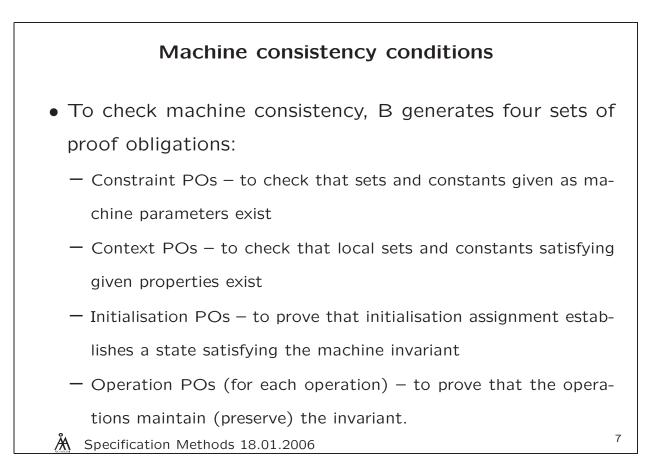


## Weakest preconditions

- We are often interested only in certain "expected" or "acceptable" final states
- A predicate *P* describing a set of "acceptable" final states is called a *postcondition*
- [S]P denotes all initial states from which execution of S is guaranteed to achieve P. [S]P weakest precondition
- The B Method provides rules for calculating weakest preconditions for different statements

Weakest precondition rules for some B Statements		
$\begin{array}{l} [x := e]  P \\ [x,y := e1,e2]  P \\ [skip]  P \\ [PRE E THEN S END]  \mathsf{P \\ [IF E THEN S1 ELSE S2 END]  \mathsf{P \end{array}$		$E \land [S]P$
[CASE E OF EITHER e1 THEN S1 OR e2 THEN S2 OR OR en THEN Sn ELSE T END] P		$(E=e1 \Rightarrow [S1]P) \land$ $(E=e2 \Rightarrow [S2]P) \land$
Specification Methods 18.01.2006		5





## Inconsistency of operations

Operations can be inconsistent because

- Operation precondition is too weak
- Operation body is not correct
- Invariant is too strong
- Invariant is too weak
- Invariant is simply wrong

