

## Notes for Lecture 3

---

### Forward- and backward semantics for sequential programs

#### Forward

Let the state space be some set  $S$ .

- *Deterministic, terminating* —  $S \rightarrow S$
- *Deterministic, possibly terminating* —  $S \rightarrow S_{\perp}$
- *Possibly nondeterministic, terminating* —  $S \rightarrow \mathbb{P}S$  (or  $S \leftrightarrow S$ )
- *Possibly nondeterministic, possibly non-terminating* —  $S \rightarrow \mathbb{P}S_{\perp}$   
(or  $S \leftrightarrow S_{\perp}$ )

#### Backward

Note that the predicates over  $S$  denote elements of  $\mathbb{P}S$ ; a predicate transformer is then of type  $\mathbb{P}S \leftarrow \mathbb{P}S$ , written “backwards” to remind us that it’s from post-condition to (weakest) precondition. We assume the transformers are all conjunctive and strict (two of Diskstra’s “healthiness conditions”).

In this case, the types of program are characterised by healthiness conditions within the single “type”  $\mathbb{P}S \leftarrow \mathbb{P}S$ , rather than varying the type as above.

- *Deterministic, terminating* — Co-strict and disjunctive.
- *Deterministic, possibly terminating* — Disjunctive.
- *Possibly nondeterministic, terminating* — Co-strict.
- *Possibly nondeterministic, possibly non-terminating* —  
(no extra healthiness conditions)

#### Galois connection

The two Galois-connected functions are those that convert a forward description to a backward one, and vice versa.