G54FOP: Lecture 6

Operational Semantics III: State

Henrik Nilsson

University of Nottingham, UK

Introducing State: Key Ideas

- μ: Store (state, memory); can be read/updated.
- Evaluation relation relates terms and stores to new terms and stores;

$$t \mid \mu \longrightarrow t' \mid \mu'$$

- l ∈ L: Uninterpreted set of locations (addresses) with equality.
- $\mu \in \mathcal{L} \to v$: Store: map from location to value.
- $\mu(l)$: Lookup value at location l.
- $[l \mapsto v]\mu$: Update; like μ except $([l \mapsto v]\mu)(l) = v$

Homework Lecture 6 (1)

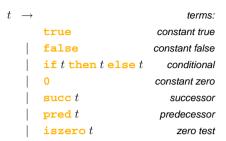
Consider the Small Imperative Language.
 Add a loop construct:

$$\begin{array}{cccc} t & \rightarrow & \textit{terms:} \\ & \dots & \\ & & | & \textit{while } t \mathrel{\mbox{do}} t & \textit{while loop} \end{array}$$

Provide evaluation rule(s) for this construct, assuming the usual semantics of a while loop: repetition of loop body **zero** or more times as long as loop condition is true.

Hint: Make use of what you have!

Small Expression Language: Terms



Small Imperative Language (1)

New terms; extends the terms of Small Expression Language:

```
t \rightarrow terms:
...

| unit constant unit | new t allocation | ! t dereferencing | t := t assignment | l store location | t; t sequencing
```

o o o G54FOP: Lecture 6 – p.5/8

Homework Lecture 6 (2)

- As mentioned, our language is still an expression language where expressions may have side effects. Design a new language (syntax and op. sem.) by separating the terms into
 - · expressions: do not have side effects
 - · commands: have side effects

and making any other changes you see fit.
Don't worry too much if the resulting language
isn't "useful" (the Small Imperative Language
isn't really). Can the evaluation rules for
expressions somehow be simplified by exploiting
that expressions do not have side effects?

Small Expression Language: Values

Small Imperative Language (2)

New values; extends the values of Small Expression Language:



Note: Still an expression language in that every term is an expression that evaluates to a value, even if some expressions have side effects. No separate category of commands.