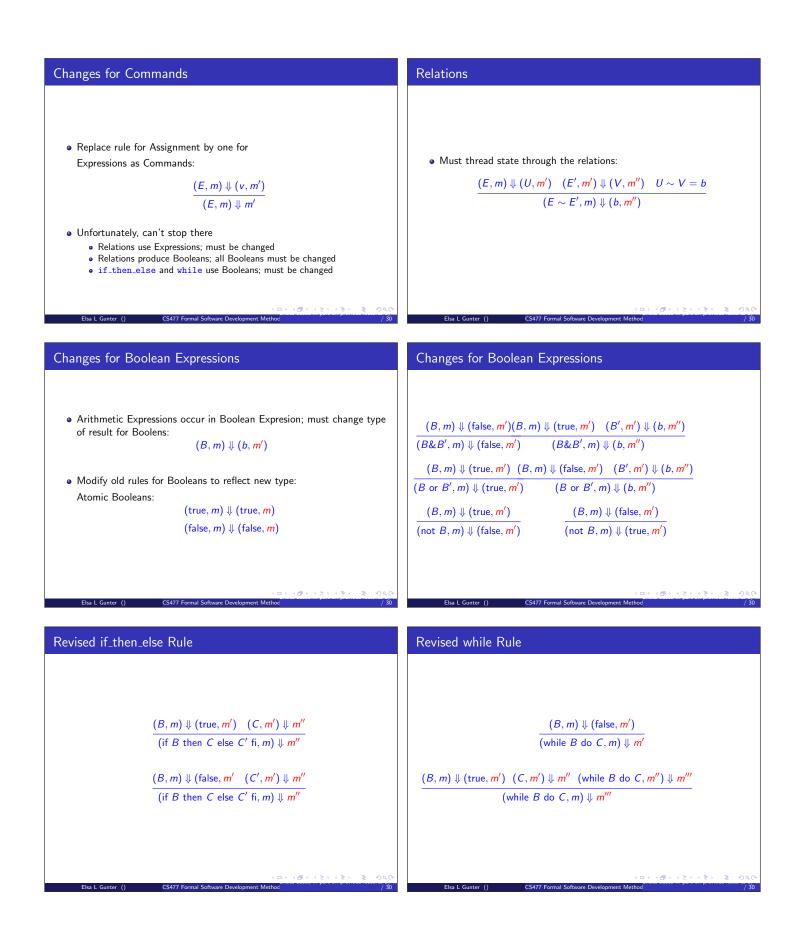


 $(E \oplus E', m) \Downarrow (N, m'')$

(間) (目) (目) (日) (日)



Termination and Errors in SOS	Transition Semantics
 (C,m), (E,m), (B,m) called configurations A configuration <i>c</i> evaluates to a result <i>r</i> if <i>c</i> ↓ <i>r</i>. If a configuration <i>c</i> evaluates to a result <i>r</i>, then <i>c</i> terminates without error Problem: Can not distinguish between untermination (<i>e.g.</i> a while loop that runs forever), versus and error (<i>e.g.</i> referencing an unassigned value Can be (partially) remedied by adding error result 	 Aka "small step structured operational semantics" Defingstarmation and for stepping from the semantics of complete evaluation Typically have two kinds of "result": configurations and final values Written (C, m) → (C', m') or (C, m) → m'
Elsa L Gunter () CS477 Formal Software Development Method () Simple Imperative Programming Language #1 (SIMPL1)	Elsa L Gunter () CS477 Formal Software Development Method /30
$I \in Identifiers$ $N \in Numerals$ $E ::= N I E + E E * E E - E$ $B ::= true false B \& B B \text{ or } B \text{ not } B$ $ E < E E = E$ $C ::= skip C; C \{C\} I ::= E$ $ if B \text{ then } C \text{ else } C \text{ fi}$ $ while B \text{ do } C$	Identifiers: $(I, m) \longrightarrow m(I)$ Numerals are values: $(N, m) \longrightarrow N$ Booleans: $(true, m) \longrightarrow true$ $(false, m) \longrightarrow false$
Elsa L Gunter () CS477 Formal Software Development Method /3	ে (০০০৫০০০০০০০০০০০০০০০০০০০০০০০০০০০০০০০০০

- Values = {true, false}
- Operators: (short-circuit)

$$\begin{array}{ll} (\mathsf{false}\&B,m) \longrightarrow \mathsf{false} & (B,m) \longrightarrow (B'',m) \\ (\mathsf{true}\&B,m) \longrightarrow (B,m) & \overline{(B\&B',m)} \longrightarrow (B''\&B,m) \end{array}$$

$$(\mathsf{true}\&B,m) \longrightarrow (B,m) \qquad \overline{(B\&a)}$$

(true or
$$B, m$$
) \longrightarrow true
(false or B, m) \longrightarrow (B, m)

$$(\text{not true}, m) \longrightarrow \text{false}$$
$$(\text{not false}, m) \longrightarrow \text{true}$$

Elsa L Gunter ()

$$\frac{(B,m) \longrightarrow (B'',m)}{(B \text{ or } B',m) \longrightarrow (B'' \text{ or } B,m)}$$
$$\frac{(B,m) \longrightarrow (B'' \text{ or } B,m)}{(\text{not } B,m) \longrightarrow (\text{not } B',m)}$$

nt Me

0

• Let U, V be arithmetic values

$$\frac{(E,m)\longrightarrow (E'',m)}{(E\sim E',m)\longrightarrow (E''\sim E',m)}$$

$$\frac{(E,m)\longrightarrow (E',m)}{(V\sim E,m)\longrightarrow (V\sim E',m)}$$

are Development Met

 $(U \sim V, m) \longrightarrow b$

CS477 Formal Soft

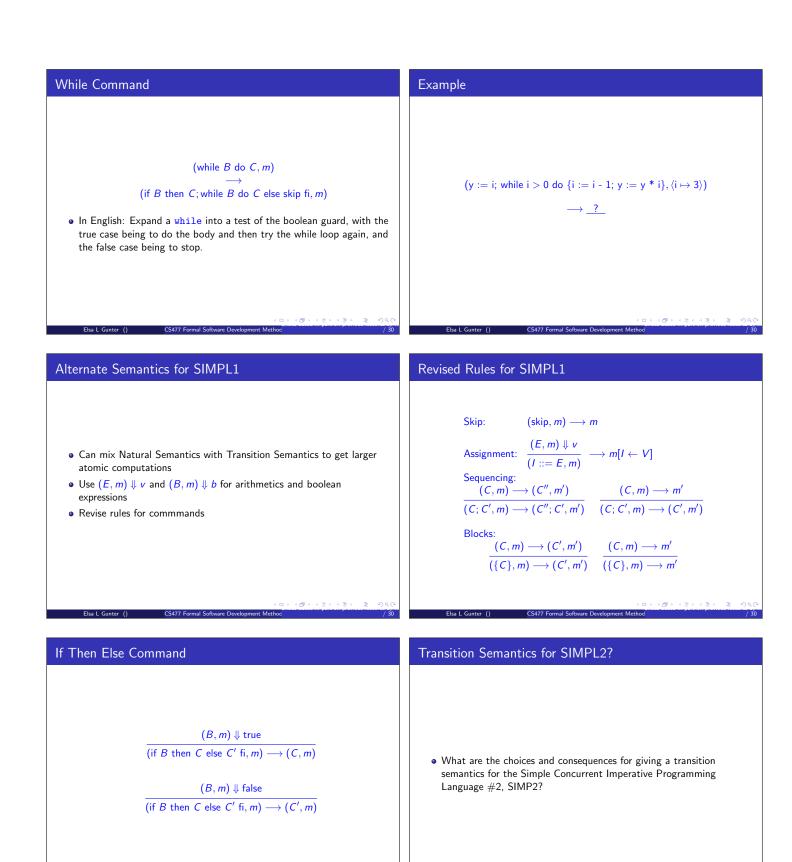
where $U \sim V = b$

Elsa L Gunter ()

10 < a> 3

Arithmetic ExpressionsCommands - in English
$$(E, m) \rightarrow (E', m)$$

 $(E \in E', m) \rightarrow (E', m)$
 $(V \in E, m) \rightarrow (V \in E', m)$
 $(V \in E, m) \rightarrow (V \in E', m)$
 $(U \in V, m) \rightarrow N$
where N is the specified values for $U \oplus V$ • all p means done evaluating an assignment, coaluate expression first
• If the organized size actuar, update the memory with
the new value for the distribution.
• When evaluating a sequence, work on the first command in the
sequence with new memory
• When evaluating a sequence, work on the first command in the
sequence with new memoryCommandsStip:
 $(E = V, m) \rightarrow m$
 $(E = E, m) \rightarrow (E', m)$
 $(E = E, m) \rightarrow (E', m) \rightarrow (C', m)$
 $(E = E, m) \rightarrow (E', m) \rightarrow (C', m)$
 $(E = E, m) \rightarrow (E', m) \rightarrow (C', m)$
 $(E = E, m) \rightarrow (E', m) \rightarrow (C', m)$
 $(E = E, m) \rightarrow (E', m) \rightarrow (C', m)$
 $(E = E = Command - in EnglishIf Then Else Command - in EnglishIf the boolean guard in an if, sheat, all as is true, then evaluate thefirst banch.If the boolean guard is not a value, then start by coaluate thefirst banch.If the boolean guard is not a value, then start by coaluate thefirst banch.If the boolean guard is not a value, then start by coaluate thefirst banch and is the start by coaluate the start E' the first banch and E' the coaluate the start E' the start by coaluate the start during the start E' the start by coaluate the start by coaluate the start$



□ > < ⊕ > < ≧ > < ≧ > < ≧ > / 30

