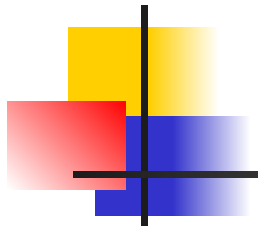


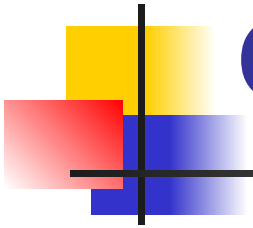
Programming Languages and Compilers (CS 421)

Talia Ringer (they/them)

4218 SC, UIUC

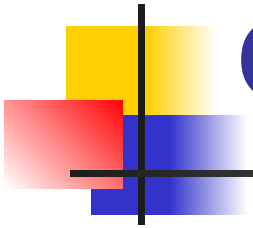
https://courses.grainger.illinois.edu/cs421/fa2023/

Based heavily on slides by Elsa Gunter, which were based in part on slides by Mattox Beckman, as updated by Vikram Adve and Gul Agha



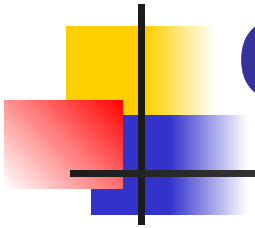
Operational Semantics

- **What it is:**
  - Describe how to execute (implement) programs of language on a virtual machine, by describing **how to execute each program statement** (i.e., following the **structure** of the program)
  - Meaning of program is how its execution **changes the state** of the machine
- **Tradeoffs:**
  - Easy to **implement**
  - Hard to **reason about abstractly** (without thinking about implementation details)



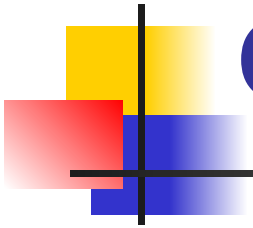
Operational Semantics

- **What it is:**
  - Describe how to execute (implement) programs of language on a virtual machine, by describing **how to execute each program statement** (i.e., following the **structure** of the program)
  - Meaning of program is how its execution **changes the state** of the machine
- **Tradeoffs:**
  - Easy to **implement**
  - Hard to **reason about abstractly** (without thinking about implementation details)



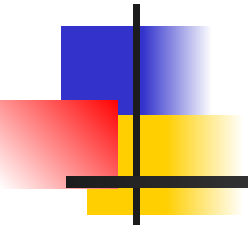
Operational Semantics

- Can be **small step** or **big step**
  - **Small step:** define meaning of one step of execution of a program statement at a time
  - **Big step:** define meaning in terms of value of execution of whole program statement
- **Common to have both** and **relate** them

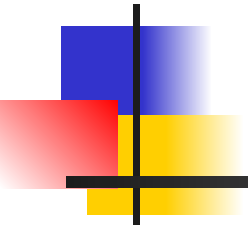


Operational Semantics

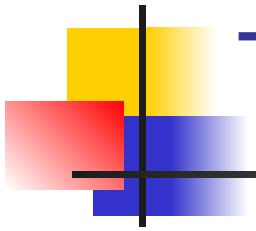
- Can be **small step** or **big step**
  - **Small step:** define meaning of one step of execution of a program statement at a time
  - **Big step:** define meaning in terms of value of execution of whole program statement
- **Common to have both** and **relate** them



Questions before we start?



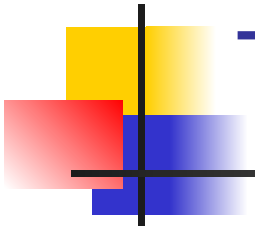
Transition (Small Step) Semantics



Transition Semantics

- Also known as **small-step operational semantics**
- Describes how each program construct **transforms** machine state by **transitions**
- **Rules** look like

  (C, m) --> (C′, m′) or (C, m) --> m′
  - C, C′ is code remaining to be executed
  - m, m′ represent the state/memory/environment
  - Sometimes m (or C) not needed
- Indicates exactly **one step** of computation

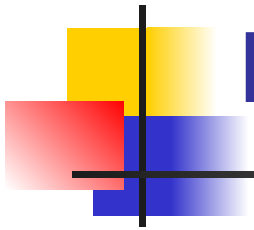


Transition Semantics

- Also known as **small-step operational semantics**
- Describes how each program construct **transforms** machine state by **transitions**
- **Rules** look like

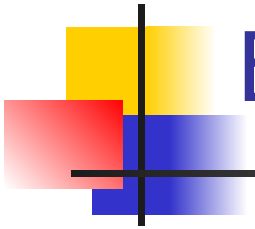
$$(C, m) \rightarrow (C', m') \quad \text{or} \quad (C, m) \rightarrow m'$$

  - C, C′ is code remaining to be executed
  - m, m′ represent the state/memory/environment
  - Sometimes m (or C) not needed
- Indicates exactly **one step** of computation



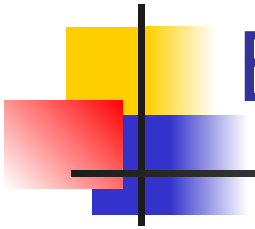
Evaluation Semantics

- Transitions successfully **stop** when E/C is a value/memory
- Evaluation **fails** if **no transition possible**, but **not** at value/memory
- Value/memory is the final **meaning** of original expression/command (in the given state)
  - Coarse semantics: final value/memory
  - More fine grained: whole transition sequence



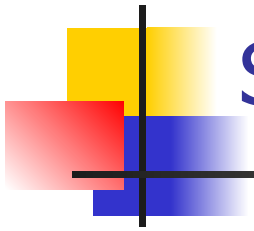
Evaluation Semantics

- Transitions successfully **stop** when E/C is a value/memory
- Evaluation **fails** if **no transition possible**, but **not** at value/memory
- Value/memory is the final **meaning** of original expression/command (in the given state)
  - Coarse semantics: final value/memory
  - More fine grained: whole transition sequence



Evaluation Semantics

- Transitions successfully **stop** when E/C is a value/memory
- Evaluation **fails** if **no transition possible**, but **not** at value/memory
- Value/memory is the final **meaning** of original expression/command (in the given state)
  - Coarse semantics: final value/memory
  - More fine grained: whole transition sequence



Simple Imperative Language Syntax

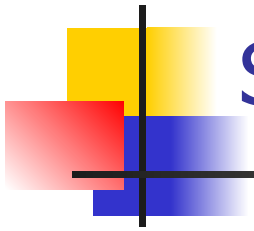
I ∈ Identifiers

N ∈ Numerals

B ::= true | false | **B** & **B** | **B** or **B** |
not **B** | **E** < **E** | **E** = **E**

E ::= N | I | **E** + **E** | **E** * **E** | **E** - **E** | - **E** | (**E**)

C ::= skip | **C**; **C** | I := **E** |
if **B** then **C** else **C** fi | while **B** do **C** od



Simple Imperative Language Syntax

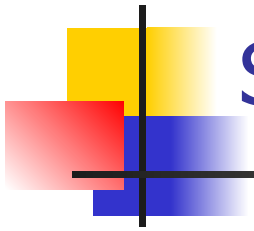
I ∈ Identifiers

N ∈ Numerals

B ::= true | false | **B** & **B** | **B** or **B** |
not **B** | **E** < **E** | **E** = **E**

E ::= N | I | **E** + **E** | **E** * **E** | **E** - **E** | - **E** | (**E**)

C ::= skip | **C**; **C** | I := **E** |
if **B** then **C** else **C** fi | while **B** do **C** od



Simple Imperative Language Syntax

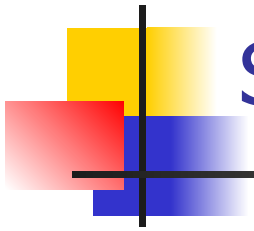
I ∈ Identifiers

N ∈ Numerals

B ::= true | false | **B** & **B** | **B** or **B** |
not **B** | **E** < **E** | **E** = **E**

E ::= N | I | **E** + **E** | **E** * **E** | **E** - **E** | - **E** | (**E**)

C ::= skip | **C**; **C** | I := **E** |
if **B** then **C** else **C** fi | while **B** do **C** od



Simple Imperative Language Syntax

I ∈ Identifiers

N ∈ Numerals

B ::= true | false | **B** & **B** | **B** or **B** |
not **B** | **E** < **E** | **E** = **E**

E ::= N | I | **E** + **E** | **E** * **E** | **E** - **E** | - **E** | (**E**)

C ::= skip | **C**; **C** | I := **E** |
if **B** then **C** else **C** fi | while **B** do **C** od

Simple Imperative Language Semantics

$(E, m) \rightarrow (V, m')$

- Values V:
  - Numerals N
  - Boolean atoms {true, false}

$$\frac{}{(I, m) \rightarrow (m(I), m)}\ \text{Id}$$

Transition Semantics

Simple Imperative Language Semantics

$(E, m) \rightarrow (V, m')$

- Values V:
  - Numerals N
  - Boolean atoms {true, false}

Look up identifiers

$$\frac{}{(I, m) \rightarrow (m(I), m)}\ \textbf{Id}$$

Simple Imperative Language Semantics

$(B, m) \rightarrow (B', m')$

$$\frac{}{(\text{false} \,\&\, B, m) \rightarrow (\text{false}, m)}\ \textbf{And-F}$$

Short-circuit

$$\frac{}{(\text{true} \,\&\, B, m) \rightarrow (B, m)}\ \textbf{And-T}$$

$$\frac{(B, m) \rightarrow (B'', m)}{(B \,\&\, B', m) \rightarrow (B'' \,\&\, B', m)}\ \textbf{And}$$

Simple Imperative Language Semantics

$(B, m) \rightarrow (B', m')$

$$\frac{}{(\textbf{false} \;\&\; B, m) \rightarrow (\textbf{false}, m)}\;\textbf{And-F}$$

Short-circuit

$$\frac{}{(\text{true} \;\&\; B, m) \rightarrow (B, m)}\;\textbf{And-T}$$

$$\frac{(B, m) \rightarrow (B'', m)}{(B \;\&\; B', m) \rightarrow (B'' \;\&\; B', m)}\;\textbf{And}$$

Simple Imperative Language Semantics

$(B, m) \rightarrow (B', m')$

$$\frac{}{(\textbf{false} \;\&\; B, m) \rightarrow (\textbf{false}, m)}\;\textbf{And-F}$$

Short-circuit

$$\frac{}{(\text{true} \;\&\; \textbf{B}, m) \rightarrow (\textbf{B}, m)}\;\textbf{And-T}$$

$$\frac{(B, m) \rightarrow (B'', m)}{(B \;\&\; B', m) \rightarrow (B'' \;\&\; B', m)}\;\textbf{And}$$

Simple Imperative Language Semantics

$(B, m) \rightarrow (B', m')$

$$\frac{}{(\textbf{false} \;\&\; B, m) \rightarrow (\textbf{false}, m)}\;\textbf{And-F}$$

$$\frac{}{(\text{true} \;\&\; \textbf{B}, m) \rightarrow (\textbf{B}, m)}\;\textbf{And-T}$$

Evaluate first clause one step

$$\frac{(B, m) \rightarrow (B'', m)}{(B \;\&\; B', m) \rightarrow (B'' \;\&\; B', m)}\;\textbf{And}$$

Simple Imperative Language Semantics

$(B, m) \rightarrow (B', m')$

$$\frac{}{(\textbf{false} \;\&\; B, m) \rightarrow (\textbf{false}, m)}\;\textbf{And-F}$$

$$\frac{}{(\text{true} \;\&\; \textbf{B}, m) \rightarrow (\textbf{B}, m)}\;\textbf{And-T}$$

Evaluate first clause one step

$$\frac{(\textbf{B}, m) \rightarrow (\textbf{B''}, m)}{(B \;\&\; B', m) \rightarrow (B'' \;\&\; B', m)}\;\textbf{And}$$

Simple Imperative Language Semantics

$(B, m) \rightarrow (B', m')$

$$\frac{}{(\textbf{false} \;\&\; B, m) \rightarrow (\textbf{false}, m)}\;\textbf{And-F}$$

$$\frac{}{(\text{true} \;\&\; \textbf{B}, m) \rightarrow (\textbf{B}, m)}\;\textbf{And-T}$$

Evaluate first clause one step

$$\frac{(\textbf{B}, m) \rightarrow (\textbf{B''}, m)}{(\textbf{B} \;\&\; B', m) \rightarrow (\textbf{B''} \;\&\; B', m)}\;\textbf{And}$$

Simple Imperative Language Semantics

$(B, m) \rightarrow (B', m')$

$$\frac{}{(\textbf{true} \text{ or } B, m) \rightarrow (\textbf{true}, m)}\ \textbf{Or-T}$$

Or is similar

$$\frac{}{(\text{false or } \textbf{B}, m) \rightarrow (\textbf{B}, m)}\ \textbf{Or-F}$$

$$\frac{(\textbf{B}, m) \rightarrow (\textbf{B''}, m)}{(\textbf{B} \text{ or } B', m) \rightarrow (\textbf{B''} \text{ or } B', m)}\ \textbf{Or}$$

Simple Imperative Language Semantics

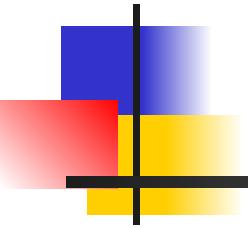
$(B, m) \rightarrow (B', m')$

$$\frac{}{(\textbf{not true}, m) \rightarrow (\textbf{false}, m)}\ \textbf{Not-T}$$

Not is not too different

$$\frac{}{(\textbf{not false}, m) \rightarrow (\textbf{true}, m)}\ \textbf{Not-F}$$

$$\frac{(\textbf{B}, m) \rightarrow (\textbf{B'}, m)}{(\textbf{not B}, m) \rightarrow (\textbf{not B'}, m)}\ \textbf{Not}$$



Questions so far?

Simple Imperative Language Semantics

$(E, m) \rightarrow (E', m')$

Evaluate LHS first ...

$$\frac{(E, m) \rightarrow (E'', m)}{(E \sim E', m) \rightarrow (E'' \sim E', m)} \textbf{ Rel-E}$$

$$\frac{(E, m) \rightarrow (E', m)}{(V \sim E, m) \rightarrow (V \sim E', m)} \textbf{ Rel-V}$$

(**U ~ V**, m) → (**true**, m) or (**false**, m) depending on whether **U ~ V** holds or not

Simple Imperative Language Semantics

$(E, m) \rightarrow (E', m')$

Evaluate LHS first ...

$$\frac{(\mathbf{E}, m) \rightarrow (\mathbf{E''}, m)}{(\mathbf{E} \sim E', m) \rightarrow (\mathbf{E''} \sim E', m)} \text{ Rel-E}$$

$$\frac{(E, m) \rightarrow (E', m)}{(V \sim E, m) \rightarrow (V \sim E', m)} \text{ Rel-V}$$

(**U ~ V**, m) → (**true**, m) or (**false**, m) depending on whether **U ~ V** holds or not

Simple Imperative Language Semantics

$(E, m) \rightarrow (E', m')$

Evaluate LHS first ...

$$\frac{(\mathbf{E}, m) \rightarrow (\mathbf{E''}, m)}{(\mathbf{E} \sim E', m) \rightarrow (\mathbf{E''} \sim E', m)} \text{ Rel-E}$$

... until you get a value

$$\frac{(E, m) \rightarrow (E', m)}{(\mathbf{V} \sim E, m) \rightarrow (\mathbf{V} \sim E', m)} \text{ Rel-V}$$

($\mathbf{U} \sim \mathbf{V}$, m) → (**true**, m) or (**false**, m) depending on whether $\mathbf{U} \sim \mathbf{V}$ holds or not

Simple Imperative Language Semantics

$(E, m) \rightarrow (E', m')$

Evaluate LHS first ...

$$\frac{(\mathbf{E}, m) \rightarrow (\mathbf{E''}, m)}{(\mathbf{E} \sim E', m) \rightarrow (\mathbf{E''} \sim E', m)} \text{ Rel-E}$$

... until you get a value

$$\frac{(\mathbf{E}, m) \rightarrow (\mathbf{E'}, m)}{(\mathbf{V} \sim \mathbf{E}, m) \rightarrow (\mathbf{V} \sim \mathbf{E'}, m)} \text{ Rel-V}$$

($\mathbf{U} \sim \mathbf{V}$, m) → (**true**, m) or (**false**, m) depending on whether $\mathbf{U} \sim \mathbf{V}$ holds or not

Simple Imperative Language Semantics

$(E, m) \rightarrow (E', m')$

Evaluate LHS first ...

$$\frac{(\mathbf{E}, m) \rightarrow (\mathbf{E''}, m)}{(\mathbf{E} \sim E', m) \rightarrow (\mathbf{E''} \sim E', m)} \text{ Rel-E}$$

... until you get a value

$$\frac{(\mathbf{E}, m) \rightarrow (\mathbf{E'}, m)}{(\mathbf{V} \sim \mathbf{E}, m) \rightarrow (\mathbf{V} \sim \mathbf{E'}, m)} \text{ Rel-V}$$

$(\mathbf{U} \sim \mathbf{V}, m) \rightarrow (\mathbf{true}, m)$ or $(\mathbf{false}, m)$ depending on whether $\mathbf{U} \sim \mathbf{V}$ holds or not

Simple Imperative Language Semantics

$(E, m) \rightarrow (E', m')$

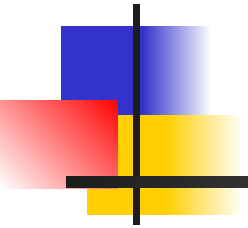
Evaluate LHS first …

$$\frac{(\mathbf{E}, m) \rightarrow (\mathbf{E''}, m)}{(\mathbf{E}\ \text{op}\ E', m) \rightarrow (\mathbf{E''}\ \text{op}\ E', m)}\ \textbf{Arith-E}$$

… until you get a value

$$\frac{(\mathbf{E}, m) \rightarrow (\mathbf{E'}, m)}{(\mathbf{V}\ \text{op}\ \mathbf{E}, m) \rightarrow (\mathbf{V}\ \text{op}\ \mathbf{E'}, m)}\ \textbf{Arith-V}$$

$(\mathbf{U}\ \text{op}\ \mathbf{V}, m) \rightarrow (\mathbf{N}, m)$
where **N** is the specified value for **U op V**



Questions so far?

Simple Imperative Language Semantics

$(C, m) \rightarrow m'$

$$\frac{}{(\text{skip}, m) \rightarrow m}\ \text{Skip}$$

$(C, m) \rightarrow (C', m')$

$$\frac{(E, m) \rightarrow (E', m)}{(I ::= E, m) \rightarrow (I ::= E', m)}\ \text{Assign-E}$$

$$\frac{}{(I ::= V, m) \rightarrow m[I \leftarrow V\ ]}\ \text{Assign-V}$$

Transition Semantics

Simple Imperative Language Semantics

Skip means done evaluating

$$\frac{}{(\text{skip}, m) \rightarrow m}\ \textbf{Skip}$$

$$\frac{(C, m) \rightarrow m'}{(C, m) \rightarrow (C', m')}$$

$$\frac{(E, m) \rightarrow (E', m)}{(I ::= E, m) \rightarrow (I ::= E', m)}\ \textbf{Assign-E}$$

$$\frac{}{(I ::= V, m) \rightarrow m[I \leftarrow V]}\ \textbf{Assign-V}$$

Transition Semantics

Simple Imperative Language Semantics

Skip means done evaluating

$$\frac{}{(\text{skip}, \mathbf{m}) \rightarrow \mathbf{m}}\ \textbf{Skip}$$

$$(C, m) \rightarrow m'$$

$$(C, m) \rightarrow (C', m')$$

$$\frac{(E, m) \rightarrow (E', m)}{(I ::= E, m) \rightarrow (I ::= E', m)}\ \textbf{Assign-E}$$

$$\frac{}{(I ::= V, m) \rightarrow m[I \leftarrow V\ ]}\ \textbf{Assign-V}$$

Transition Semantics

Simple Imperative Language Semantics

$$\frac{}{(\text{skip}, \mathbf{m}) \rightarrow \mathbf{m}}\ \textbf{Skip}$$

$$(C, m) \rightarrow m'$$

$$(C, m) \rightarrow (C', m')$$

Evaluate RHS of assignment …

$$\frac{(E, m) \rightarrow (E', m)}{(I ::= E, m) \rightarrow (I ::= E', m)}\ \textbf{Assign-E}$$

$$\frac{}{(I ::= V, m) \rightarrow m[I \leftarrow V\ ]}\ \textbf{Assign-V}$$

Transition Semantics

Simple Imperative Language Semantics

$$\frac{}{(\text{skip}, \mathbf{m}) \rightarrow \mathbf{m}}\ \textbf{Skip}$$

$$(C, m) \rightarrow m'$$

$$(C, m) \rightarrow (C', m')$$

Evaluate RHS of assignment …

$$\frac{(\mathbf{E}, m) \rightarrow (\mathbf{E'}, m)}{(I ::= \mathbf{E}, m) \rightarrow (I ::= \mathbf{E'}, m)}\ \textbf{Assign-E}$$

$$\frac{}{(I ::= V, m) \rightarrow m[I \leftarrow V\ ]}\ \textbf{Assign-V}$$

Transition Semantics

Simple Imperative Language Semantics

$$\frac{(C, m) \rightarrow m'}{(C, m) \rightarrow (C', m')}$$

$$\frac{}{(\text{skip}, \mathbf{m}) \rightarrow \mathbf{m}}\ \textbf{Skip}$$

Evaluate RHS of assignment …

$$\frac{(\mathbf{E}, m) \rightarrow (\mathbf{E'}, m)}{(I ::= \mathbf{E}, m) \rightarrow (I ::= \mathbf{E'}, m)}\ \textbf{Assign-E}$$

… until you get a value

$$\frac{}{(I ::= \mathbf{V}, m) \rightarrow m[I \leftarrow V\ ]}\ \textbf{Assign-V}$$

Simple Imperative Language Semantics

$$\frac{}{(\text{skip}, \mathbf{m}) \rightarrow \mathbf{m}}\ \textbf{Skip}$$

$(C, m) \rightarrow m'$

$(C, m) \rightarrow (C', m')$

Evaluate RHS of assignment …

$$\frac{(\mathbf{E}, m) \rightarrow (\mathbf{E'}, m)}{(I ::= \mathbf{E}, m) \rightarrow (I ::= \mathbf{E'}, m)}\ \textbf{Assign-E}$$

… until you get a value

$$\frac{}{(I ::= \mathbf{V}, m) \rightarrow \mathbf{m}[\mathbf{I} \leftarrow \mathbf{V}]}\ \textbf{Assign-V}$$

Simple Imperative Language Semantics

$(C, m) \rightarrow m'$

$(C, m) \rightarrow (C', m')$

Evaluate first command of sequence first …

$$\frac{(C, m) \rightarrow (C'', m')}{(C; C', m) \rightarrow (C''; C', m')} \text{ Seq-L}$$

$$\frac{(C, m) \rightarrow m'}{(C; C', m) \rightarrow (C', m')} \text{ Seq-R}$$

Simple Imperative Language Semantics

$(C, m) \rightarrow m'$

$(C, m) \rightarrow (C', m')$

Evaluate first command of sequence first …

$$\frac{(\mathbf{C}, m) \rightarrow (\mathbf{C''}, m')}{(\mathbf{C}; C', m) \rightarrow (\mathbf{C''}; C', m')} \text{ Seq-L}$$

$$\frac{(C, m) \rightarrow m'}{(C; C', m) \rightarrow (C', m')} \text{ Seq-R}$$

Simple Imperative Language Semantics

$(C, m) \rightarrow m'$

$(C, m) \rightarrow (C', m')$

Evaluate first command of sequence first …

$$\frac{(\mathbf{C}, m) \rightarrow (\mathbf{C''}, m')}{(\mathbf{C}; C', m) \rightarrow (\mathbf{C''}; C', m')} \text{ Seq-L}$$

… until it gives back a memory

$$\frac{(\mathbf{C}, m) \rightarrow \mathbf{m'}}{(\mathbf{C}; C', m) \rightarrow (C', \mathbf{m'})} \text{ Seq-R}$$

Simple Imperative Language Semantics

$$\frac{(C, m) \rightarrow m'}{(C, m) \rightarrow (C', m')}$$

Evaluate first command of sequence first …

$$\frac{(\mathbf{C}, m) \rightarrow (\mathbf{C''}, m') \quad \textbf{Seq-L}}{(\mathbf{C}; C', m) \rightarrow (\mathbf{C''}; C', m')}$$

… until it gives back a memory

$$\frac{(\mathbf{C}, m) \rightarrow \mathbf{m'} \quad \textbf{Seq-R}}{(\mathbf{C}; \mathbf{C'}, m) \rightarrow \mathbf{(C', m')}}$$

Simple Imperative Language Semantics

$(C, m) \rightarrow m'$

$(C, m) \rightarrow (C', m')$

$$\frac{}{(\text{if } \textbf{true} \text{ then } C \text{ else } C' \text{ fi}, m) \rightarrow (C, m)}\ \textbf{If-T}$$

$$\frac{}{(\text{if } \textbf{false} \text{ then } C \text{ else } C' \text{ fi}, m) \rightarrow (C', m)}\ \textbf{If-F}$$

$$\frac{(B, m) \rightarrow (B', m)}{(\text{if } B \text{ then } C \text{ else } C' \text{ fi}, m) \rightarrow (\text{if } B' \text{ then } C \text{ else } C' \text{ fi}, m)}\ \textbf{If}$$

Simple Imperative Language Semantics

$(C, m) \rightarrow m'$

$(C, m) \rightarrow (C', m')$

If guard is true, evaluate first branch

$$\frac{}{(\text{if } \mathbf{true} \text{ then } \mathbf{C} \text{ else } C' \text{ fi}, m) \rightarrow (\mathbf{C}, m)}\ \textbf{If-T}$$

$$\frac{}{(\text{if } \mathbf{false} \text{ then } C \text{ else } C' \text{ fi}, m) \rightarrow (C', m)}\ \textbf{If-F}$$

$$\frac{(B, m) \rightarrow (B', m)}{(\text{if } B \text{ then } C \text{ else } C' \text{ fi}, m) \rightarrow (\text{if } B' \text{ then } C \text{ else } C' \text{ fi}, m)}\ \textbf{If}$$

Simple Imperative Language Semantics

$(C, m) \rightarrow m'$

$(C, m) \rightarrow (C', m')$

If guard is false, evaluate second branch

$$\frac{}{(\text{if } \mathbf{true} \text{ then } \mathbf{C} \text{ else } C' \text{ fi}, m) \rightarrow (\mathbf{C}, m)}\ \textbf{If-T}$$

$$\frac{}{(\text{if } \mathbf{false} \text{ then } C \text{ else } \mathbf{C'} \text{ fi}, m) \rightarrow (\mathbf{C'}, m)}\ \textbf{If-F}$$

$$\frac{(B, m) \rightarrow (B', m)}{(\text{if } B \text{ then } C \text{ else } C' \text{ fi}, m) \rightarrow (\text{if } B' \text{ then } C \text{ else } C' \text{ fi}, m)}\ \textbf{If}$$

Simple Imperative Language Semantics

$(C, m) \rightarrow m'$

$(C, m) \rightarrow (C', m')$

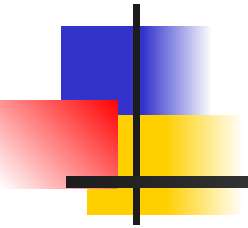
$$\frac{}{(\text{if } \mathbf{true} \text{ then } \mathbf{C} \text{ else } C' \text{ fi}, m) \rightarrow (\mathbf{C}, m)}\ \textbf{If-T}$$

$$\frac{}{(\text{if } \mathbf{false} \text{ then } C \text{ else } \mathbf{C'} \text{ fi}, m) \rightarrow (\mathbf{C'}, m)}\ \textbf{If-F}$$

$$\frac{(\mathbf{B}, m) \rightarrow (\mathbf{B'}, m)}{(\text{if } \mathbf{B} \text{ then } C \text{ else } C' \text{ fi}, m) \rightarrow (\text{if } \mathbf{B'} \text{ then } C \text{ else } C' \text{ fi}, m)}\ \textbf{If}$$

Evaluate guard until it's a value, so one of the above applies

Transition Semantics



Questions so far?

Simple Imperative Language Semantics

$(C, m) \rightarrow m'$

$(C, m) \rightarrow (C', m')$

$$\frac{??}{(\text{while } B \text{ do } C \text{ od}, m) \rightarrow ??}$$

Question: What should while do?

Simple Imperative Language Semantics

$(C, m) \rightarrow m'$

$(C, m) \rightarrow (C', m')$

$$\frac{(B, m) \rightarrow (B', m)}{(\text{while } B \text{ do } C \text{ od}, m) \rightarrow (\text{while } B' \text{ do } C \text{ od}, m)}$$

This is tempting, but it is wrong!

Simple Imperative Language Semantics

$(C, m) \rightarrow m'$

$(C, m) \rightarrow (C', m')$

Rather, we unroll the loop once

$$\frac{}{(\text{while } B \text{ do } C \text{ od}, m) \rightarrow (\text{if } B \text{ then } C; \text{ while } B \text{ do } C \text{ od else skip fi}, m)}$$

Simple Imperative Language Semantics

$(C, m) \rightarrow m'$

$(C, m) \rightarrow (C', m')$

Rather, we unroll the loop once …

$$\frac{}{(\text{while } \mathbf{B} \text{ do } C \text{ od}, m) \rightarrow (\text{if } \mathbf{B} \text{ then } C; \text{while } B \text{ do } C \text{ od else skip fi}, m)}$$

Simple Imperative Language Semantics

$(C, m) \rightarrow m'$

$(C, m) \rightarrow (C', m')$

… continue while guard is true …

(**while B do C od**, m) →
(if B then **C; while B do C od** else skip fi, m)

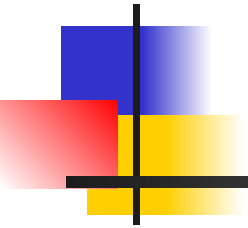
Simple Imperative Language Semantics

$(C, m) \rightarrow m'$

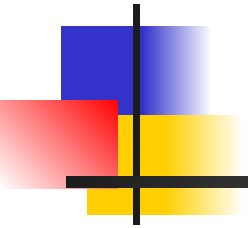
$(C, m) \rightarrow (C', m')$

… and stop when guard is false.

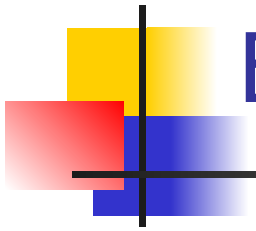
$$(\text{while } B \text{ do } C \text{ od}, m) \rightarrow$$
$$(\text{if } B \text{ then } C; \text{ while } B \text{ do } C \text{ od else } \textbf{skip} \text{ fi}, m)$$



Questions so far?



Example



Example Evaluation

First step:

$$\frac{}{(\text{if } x > 5 \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \to 7\}) \rightarrow \; ??}$$

Example Evaluation

First step:

$$\frac{}{(\text{if } \mathbf{x > 5} \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \to 7\}) \rightarrow ??}\ \textbf{If}$$

Evaluate guard until it's a value

Example Evaluation

First step:

$$\frac{(\textbf{x > 5}, \{x \to 7\}) \to \textbf{??}}{(\text{if } \textbf{x > 5} \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \to 7\}) \to \textbf{??}}\ \textbf{If}$$

Evaluate guard until it's a value

Example Evaluation

First step:

$$\frac{(\textbf{x > 5}, \{x \text{ -> } 7\}) \rightarrow \textbf{??}}{(\text{if } x > 5 \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \text{ -> } 7\}) \rightarrow \textbf{??}}\ \textbf{If}$$

How to evaluate guard?

Example Evaluation

First step:

$$\dfrac{\dfrac{??}{(\mathbf{x > 5}, \{x \text{ -> } 7\}) \rightarrow ??}\ \textbf{Rel-E}}{(\text{if } x > 5 \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \text{ -> } 7\}) \rightarrow ??}\ \textbf{If}$$

The guard is a relation

Example Evaluation

First step:

$$\dfrac{\dfrac{(\mathbf{x}, \{x \text{ -> } 7\}) \rightarrow \textbf{??}}{(\mathbf{x} > 5, \{x \text{ -> } 7\}) \rightarrow \textbf{??}}\ \textbf{Rel-E}}{(\text{if } x > 5 \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \text{ -> } 7\}) \rightarrow \textbf{??}}\ \textbf{If}$$

And x is not yet a value, so we evaluate LHS first

Example Evaluation

First step:

$$\dfrac{\dfrac{\dfrac{}{(\mathbf{x}, \{x \text{ -> } 7\}) \rightarrow \text{??}}\ \textbf{Id}}{(\mathbf{x} > 5, \{x \text{ -> } 7\}) \rightarrow \text{??}}\ \textbf{Rel-E}}{(\text{if } x > 5 \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \text{ -> } 7\}) \rightarrow \text{??}}\ \textbf{If}$$

x is an identifier ...

Example Evaluation

First step:

$$\dfrac{\dfrac{\dfrac{}{(\mathbf{x}, \{x \text{ -> } 7\}) \rightarrow (\mathbf{7}, \{x \text{ -> } 7\})}\ \textbf{Id}}{(\mathbf{x} > 5, \{x \text{ -> } 7\}) \rightarrow \mathbf{??}}\ \textbf{Rel-E}}{(\text{if } x > 5 \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \text{ -> } 7\}) \rightarrow \mathbf{??}}\ \textbf{If}$$

x is an identifier so we look it up

Example Evaluation

First step:

$$\dfrac{\dfrac{\dfrac{}{(\mathbf{x}, \{x \text{ -> } 7\}) \rightarrow (\mathbf{7}, \{x \text{ -> } 7\})}\ \textbf{Id}}{(\mathbf{x} > 5, \{x \text{ -> } 7\}) \rightarrow (\mathbf{7} > 5, \{x \text{ -> } 7\})}\ \textbf{Rel-E}}{(\text{if } x > 5 \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \text{ -> } 7\}) \rightarrow ??}\ \textbf{If}$$

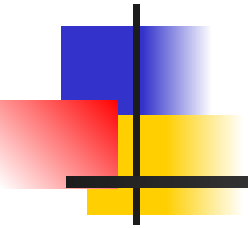
Propagate downward

Example Evaluation

First step:

$$\dfrac{\dfrac{\dfrac{}{(\mathbf{x}, \{x \text{ -> } 7\}) \rightarrow (\mathbf{7}, \{x \text{ -> } 7\})}\ \textbf{Id}}{(\mathbf{x} > 5, \{x \text{ -> } 7\}) \rightarrow (\mathbf{7} > 5, \{x \text{ -> } 7\})}\ \textbf{Rel-E}}{(\text{if } x > 5 \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \text{ -> } 7\}) \rightarrow (\text{if } \mathbf{7} > 5 \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \text{ -> } 7\})}\ \textbf{If}$$

Propagate downward

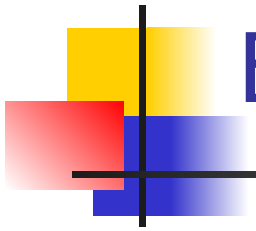


Questions so far?

Example Evaluation

First step:

$$\dfrac{\dfrac{\dfrac{}{(\mathbf{x}, \{x \text{ -> } 7\}) \rightarrow (\mathbf{7}, \{x \text{ -> } 7\})}\,\textbf{Id}}{(\mathbf{x} > 5, \{x \text{ -> } 7\}) \rightarrow (\mathbf{7} > 5, \{x \text{ -> } 7\})}\,\textbf{Rel-E}}{(\text{if } x > 5 \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \text{ -> } 7\}) \rightarrow (\text{if } \mathbf{7} > 5 \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \text{ -> } 7\})}\,\textbf{If}$$



Example Evaluation

Second step:

$$\frac{}{(\text{if } \mathbf{7} > 5 \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \to 7\}) \rightarrow \; ??}$$

Example Evaluation

Second step:

$$\frac{(\textbf{7 > 5}, \{x \to 7\}) \to \textbf{??}}{(\text{if } \textbf{7 > 5} \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \to 7\}) \to \textbf{??}}\ \textbf{If}$$

Evaluate guard until it's a value (still not a value)

Example Evaluation

Second step:

By semantics of >

$$\frac{(\mathbf{7 > 5}, \{x \text{ -> } 7\}) \rightarrow (\mathbf{true}, \{x \text{ -> } 7\})}{(\text{if } \mathbf{7 > 5} \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \text{ -> } 7\}) \rightarrow \text{??}}\ \text{If}$$

Evaluate guard until it's a value (still not a value)

Example Evaluation

Second step:

By semantics of >

$$\frac{(\mathbf{7 > 5}, \{x \text{ -> } 7\}) \rightarrow (\mathbf{true}, \{x \text{ -> } 7\})}{(\text{if } \mathbf{7 > 5} \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \text{ -> } 7\}) \rightarrow (\text{if } \mathbf{true} \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \text{ -> } 7\})}\ \textbf{If}$$

Evaluate guard until it's a value (still not a value)

Example Evaluation

Second step:

By semantics of >

$$\frac{(\mathbf{7 > 5}, \{x \text{ -> } 7\}) \rightarrow (\mathbf{true}, \{x \text{ -> } 7\})}{(\text{if } \mathbf{7 > 5} \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \text{ -> } 7\}) \rightarrow (\text{if } \mathbf{true} \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \text{ -> } 7\})}\ \textbf{If}$$

Evaluate guard until it's a value (now it's a value!)

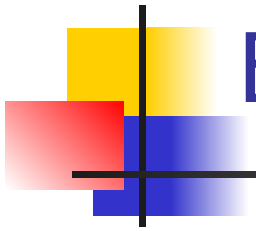
Example Evaluation

Second step:

$$\frac{(\mathbf{7 > 5}, \{x \to 7\}) \to (\mathbf{true}, \{x \to 7\})}{(\text{if } \mathbf{7 > 5} \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \to 7\}) \to (\text{if } \mathbf{true} \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \to 7\})}\ \textbf{If}$$

Third step:

$$\frac{}{(\text{if } \mathbf{true} \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \to 7\}) \to \mathbf{??}}\ \textbf{If-T}$$



Example Evaluation

Second step:

$$\frac{(\textbf{7 > 5}, \{x \to 7\}) \to (\textbf{true}, \{x \to 7\})}{(\text{if } \textbf{7 > 5} \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \to 7\}) \to (\text{if } \textbf{true} \text{ then } y := 2 + 3 \text{ else } y := 3 + 4 \text{ fi}, \{x \to 7\})}\ \textbf{If}$$

Third step:

Step to then case

$$\frac{}{(\text{if } \textbf{true} \text{ then } \textbf{y := 2 + 3} \text{ else } y := 3 + 4 \text{ fi}, \{x \to 7\}) \to (\textbf{y := 2 + 3}, \{x \to 7\})}\ \textbf{If-T}$$

Transition Semantics Example

Example Evaluation

Fourth Step:

$$\frac{(\mathbf{2 + 3}, \{x \text{ -> } 7\}) \rightarrow (\mathbf{5}, \{x \text{ -> } 7\})\ \textbf{Assign-E}}{(y := \mathbf{2 + 3}, \{x \text{ -> } 7\}) \rightarrow (y := \mathbf{5}, \{x \text{ -> } 7\})}$$

Example Evaluation

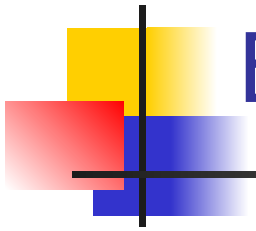
Fourth Step:

$$\frac{(\mathbf{2 + 3}, \{x \text{ -> } 7\}) \rightarrow (\mathbf{5}, \{x \text{ -> } 7\})\ \textbf{Assign-E}}{(y := \mathbf{2 + 3}, \{x \text{ -> } 7\}) \rightarrow (y := \mathbf{5}, \{x \text{ -> } 7\})}$$

Fifth Step:

$$\frac{}{(y := \mathbf{5}, \{x \text{ -> } 7\}) \rightarrow \{\mathbf{y \text{ -> } 5}, x \text{ -> } 7\}}\ \textbf{Assign-V}$$

Transition Semantics Example



Example Evaluation

Bottom line:

(if x > 5 then y := 2 + 3 else y := 3 + 4 fi, {x -> 7}) →

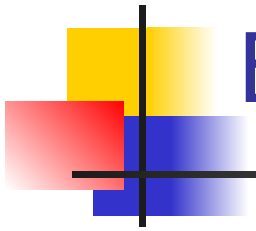
(if 7 > 5 then y := 2 + 3 else y := 3 + 4 fi, {x -> 7}) →

(if true then y := 2 + 3 else y := 3 + 4 fi, {x -> 7}) →

(y := 2 + 3, {x -> 7}) →

(y := 5, {x -> 7}) →

{y -> 5, x -> 7}



Example Evaluation

Bottom line:

(if x > 5 then y := 2 + 3 else y := 3 + 4 fi, {x -> 7}) →

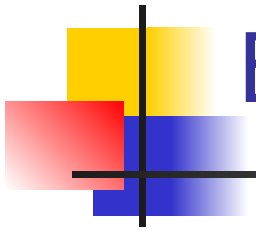
(if 7 > 5 then y := 2 + 3 else y := 3 + 4 fi, {x -> 7}) →

(if true then y := 2 + 3 else y := 3 + 4 fi, {x -> 7}) →

(y := 2 + 3, {x -> 7}) →

(y := 5, {x -> 7}) →

{y -> 5, x -> 7}



Example Evaluation

Bottom line:

(if x > 5 then y := 2 + 3 else y := 3 + 4 fi, {x -> 7}) →

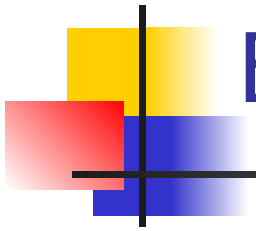
(if 7 > 5 then y := 2 + 3 else y := 3 + 4 fi, {x -> 7}) →

(if true then y := 2 + 3 else y := 3 + 4 fi, {x -> 7}) →

(y := 2 + 3, {x -> 7}) →

(y := 5, {x -> 7}) →

{y -> 5, x -> 7}



Example Evaluation

Bottom line:

(if x > 5 then y := 2 + 3 else y := 3 + 4 fi, {x -> 7}) →

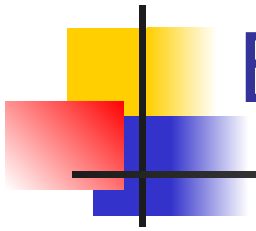
(if 7 > 5 then y := 2 + 3 else y := 3 + 4 fi, {x -> 7}) →

(if true then y := 2 + 3 else y := 3 + 4 fi, {x -> 7}) →

(y := 2 + 3, {x -> 7}) →

(y := 5, {x -> 7}) →

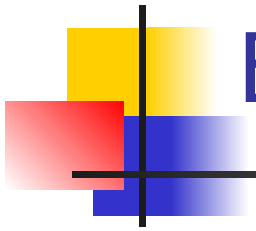
{y -> 5, x -> 7}



Example Evaluation

Bottom line:

(if x > 5 then y := 2 + 3 else y := 3 + 4 fi, {x -> 7}) →
(if 7 > 5 then y := 2 + 3 else y := 3 + 4 fi, {x -> 7}) →
(if true then y := 2 + 3 else y := 3 + 4 fi, {x -> 7}) →
(y := 2 + 3, {x -> 7}) →
(y := 5, {x -> 7}) →
{y -> 5, x -> 7}



Example Evaluation

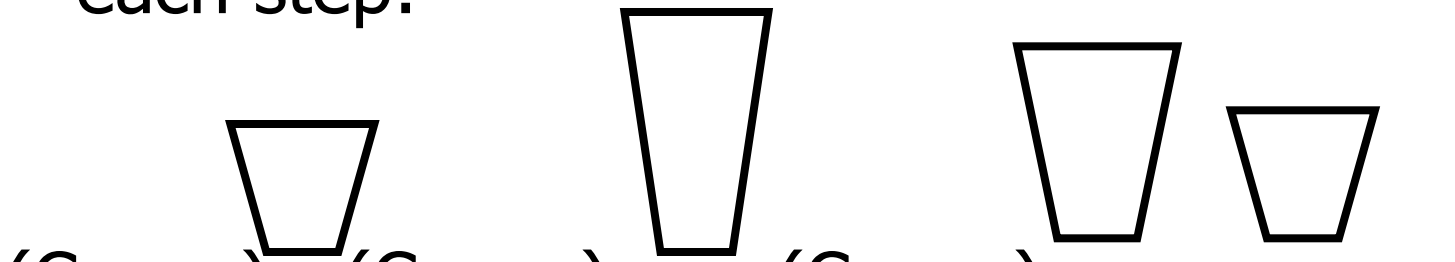
Bottom line:

(if x > 5 then y := 2 + 3 else y := 3 + 4 fi, {x -> 7}) →
(if 7 > 5 then y := 2 + 3 else y := 3 + 4 fi, {x -> 7}) →
(if true then y := 2 + 3 else y := 3 + 4 fi, {x -> 7}) →
(y := 2 + 3, {x -> 7}) →
(y := 5, {x -> 7}) →
{y -> 5, x -> 7}

(if x > 5 then y := 2 + 3 else y := 3 + 4 fi, {x -> 7})
→* {y -> 5, x -> 7}

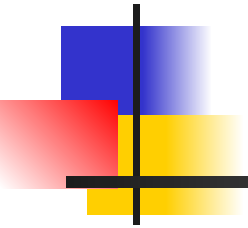
Transition Semantics Evaluation

- A sequence of steps with trees of justification for each step:

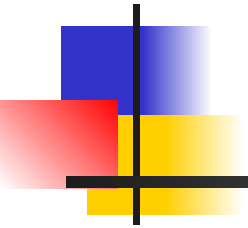


- Let $\rightarrow^*$ be the transitive closure of $\rightarrow$
- i.e., the smallest transitive relation containing $\rightarrow$

Transition Semantics Example



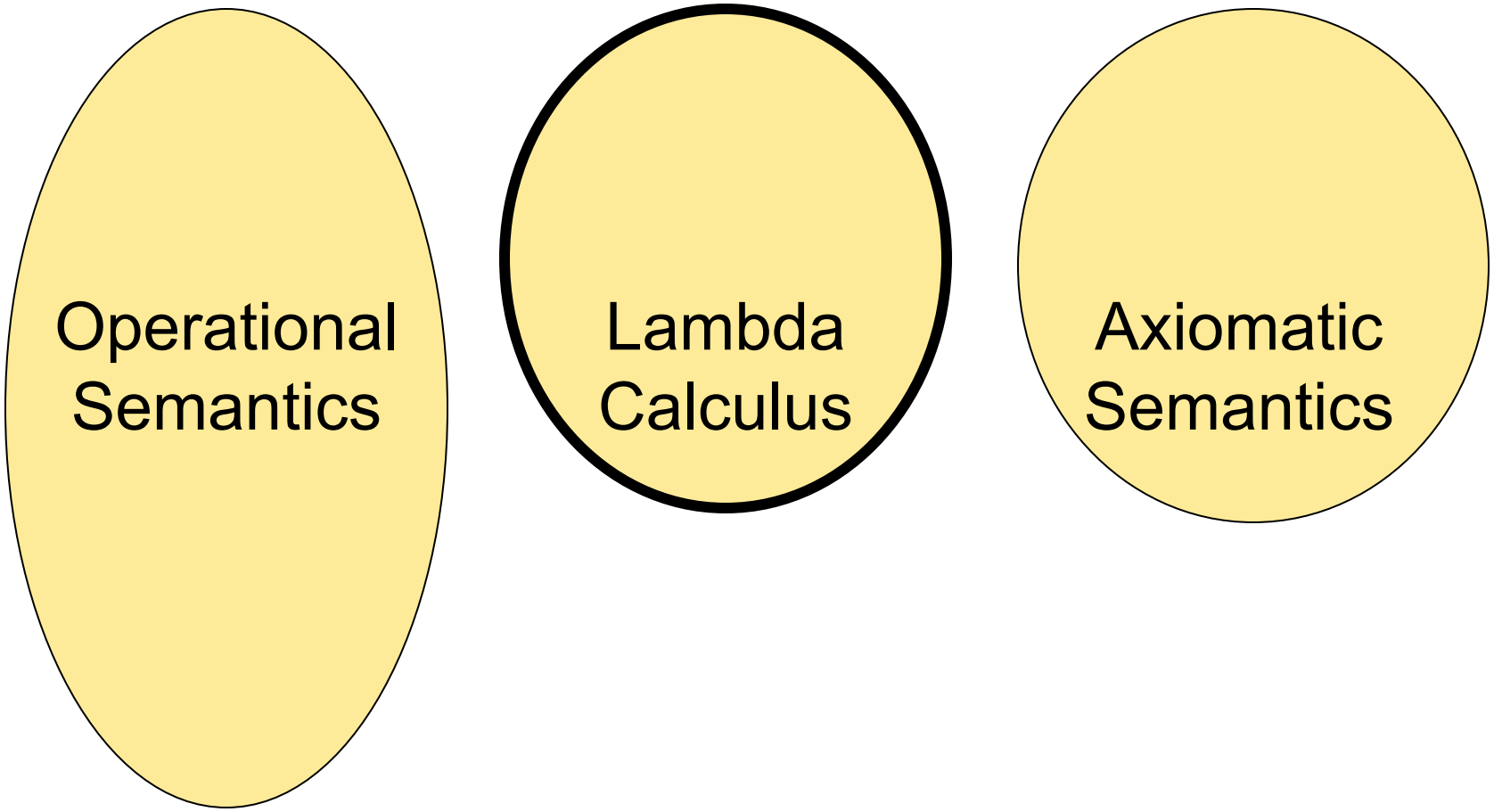
Questions?



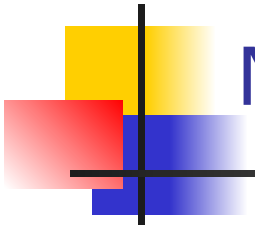
Next Class: Lambda Calculus

Next Class

III : Language Semantics



Lamda Calculus



Next Class

- **Just started** grading **EC2**
- **WA9** is due **this Thursday**
- **MP10** is not, in fact, for credit (it is for **Q5**)
- **Q5** is immediately upon returning from break
- May be in touch about extra **makeup** opportunities
- All deadlines can be found on **course website**
- Use **office hours** and **class forums** for help