

# Software Productivity

Paulo Borba  
Informatics Center  
Federal University of Pernambuco

[phmb@cin.ufpe.br](mailto:phmb@cin.ufpe.br) ♦ [twitter.com/pauloborba](https://twitter.com/pauloborba)

# Basic transformation concepts

Paulo Borba  
Informatics Center  
Federal University of Pernambuco

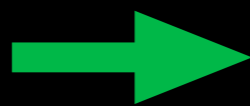
[phmb@cin.ufpe.br](mailto:phmb@cin.ufpe.br) ♦ [twitter.com/pauloborba](https://twitter.com/pauloborba)

# Transformations define DSL semantics

```
applicationCode
entity User {
  username :: String (id.name)
  tasks    -> List<Task>
  log      -> List<Task>
}
entity Task {
  description :: String
  done        :: Bool
}

define page home() {
  section()
  header("Users")
  list()
  for(user :: User) {
    listTasks()
    navigate(tasks(user))
    {output(user.username)}
    (" output(user.tasks.length) ")
  }
  navigate(newUser()) ("Add new user")
}

define page newUser() {
  var user :: User := User ()
  form()
  group("Add new user") {
    derive addNew from user for ( username )
    action "add" { add() }
    action "save" { save() }
    return tasks(user)
  }
  navigate(home()) ("Home")
}
```



```
build.xml
1 <project name="todo" default="compile">
2
3   <property name="hibernate.c3p0.acquire_increment" value="1"/>
4   <property name="hibernate.c3p0.min_size" value="0"/>
5   <property name="hibernate.c3p0.max_size" value="0"/>
6   <property name="hibernate.c3p0.timeout" value="0"/>
7   <property name="hibernate.c3p0.max_statements" value="0"/>
8   <property name="hibernate.c3p0.idle_test_period" value="0"/>
9   <property name="hibernate.c3p0.preferred_connection_test_period" value="0"/>
10  </project>

hibernate.properties
hibernate.c3p0.acquire_increment=1
hibernate.c3p0.min_size=0
hibernate.c3p0.max_size=0
hibernate.c3p0.timeout=0
hibernate.c3p0.max_statements=0
hibernate.c3p0.idle_test_period=0
hibernate.c3p0.preferred_connection_test_period=0

User.java
package todo.domain;

import java.util.*;
import javax.persistence.*;
import todo.domain.*;
import java.io.Serializable;
import utils.*;
import java.io.PrintWriter;
import org.webdsl.lang.Environment;

@Entity
@Table(name = "_User", uniqueConstraints = {
    @UniqueConstraint(columnNames = { "username" })
})
public class User {
    private String username;
    private List<Task> tasks;
    private List<Task> log;

    public User() {}
    public User(String username) {
        this.username = username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getUsername() {
        return username;
    }
    public void addTask(Task task) {
        tasks.add(task);
    }
    public List<Task> getTasks() {
        return tasks;
    }
    public void setTasks(List<Task> tasks) {
        this.tasks = tasks;
    }
    public void addLog(Task task) {
        log.add(task);
    }
    public List<Task> getLog() {
        return log;
    }
    public void setLog(List<Task> log) {
        this.log = log;
    }
}

addtask1.java
package todo.actions;

import todo.domain.*;

public interface addtask1 {
    public void addTask(User user, Task task);
}

HomePage.java
package todo.actions;

import java.io.PrintWriter;
import org.webdsl.lang.Environment;
import org.hibernate.Session;
import org.webdsl.lang.Environment;
import utils.TemplateCall;

public class HomePage {
    public void render(PrintWriter s, Environment env) {
        s.println("<!DOCTYPE HTML PUBLIC");
        s.println("<html>");
        s.println("<head>");
        s.println("<title>" + getPageTitle());
        s.println("<link href=\"" + "/todo/" + ">");
        s.println("<link href=\"" + "/todo/" + ">");
        s.println("<script src=\"" + "/todo/" + ">");
        s.println("<script src=\"" + "/todo/" + ">");
        s.println("<script type=\"text/javascript\">");
        s.println("</head>");
        s.println("<body>");
        renderIncomingSuccessMessages(s);
        s.flush();
    }
}

EntityType.java
package utils;

public class EntityType {
    public static Object loadEntity(org.hibernate.Session hibSession, Class c, String s) {
        try {
            return hibSession.load(c, s);
        } catch (Exception e) {
            return null;
        }
    }
}
```

# Boolean abstract syntax

```
module prop
signature
  sorts Prop
  constructors
    False  : Prop
    True   : Prop
    Not    : Prop -> Prop
    And    : Prop * Prop -> Prop
```

# Defining a transformation

**Not (False)  $\rightarrow$  True**

before

after

single language, abstract syntax

# Applying a transformation

`Not (False) -> True`

`Not (False)`

transformation tool

`True`

# Applying a transformation, failure

`Not (False) -> True`

`Not (False) And True`

transformation tool

**failure!**

# Applying a transformation (exhaustively)

Not (False) -> True

Not (False) And True

transformation tool

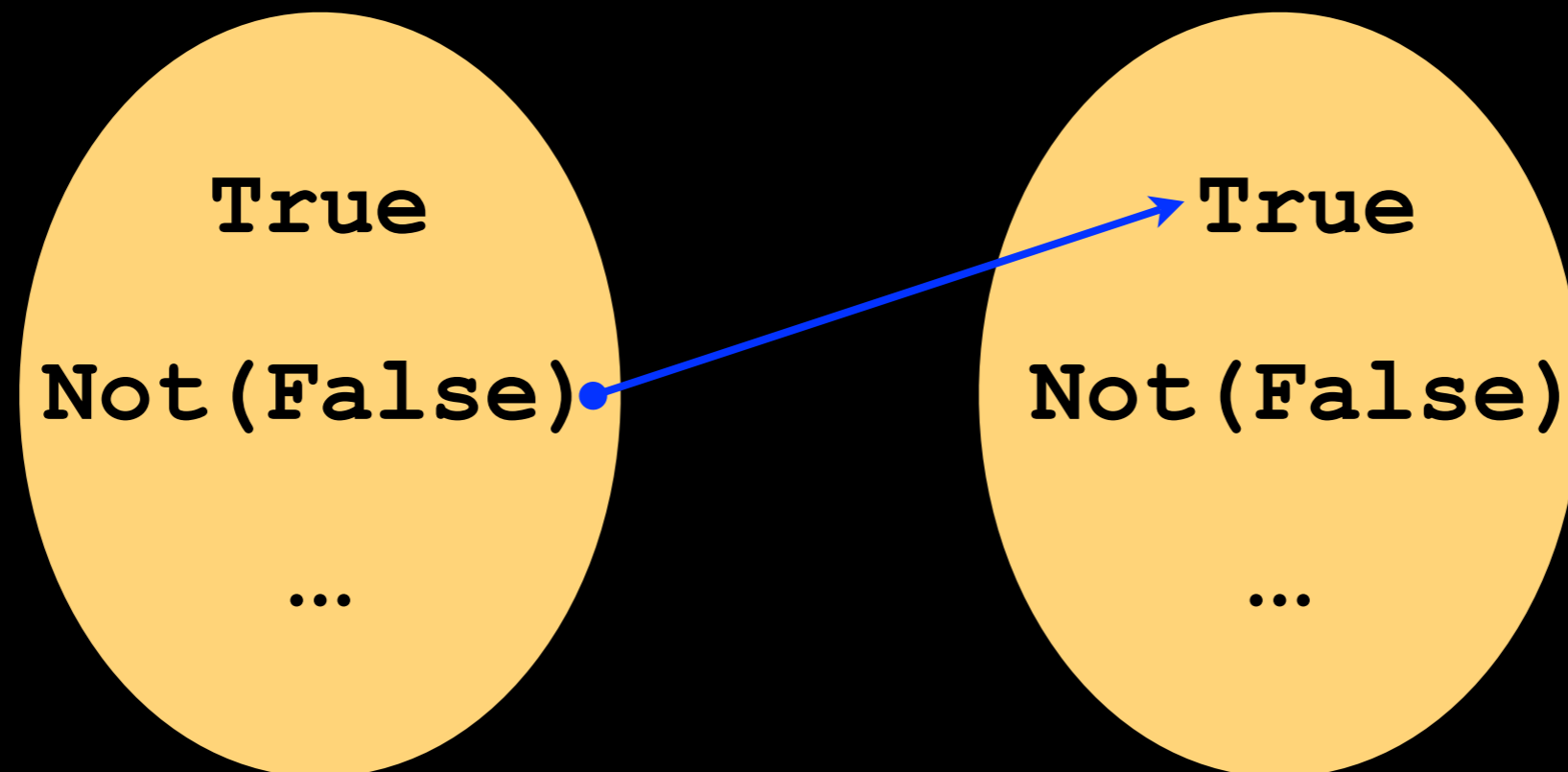


True And True

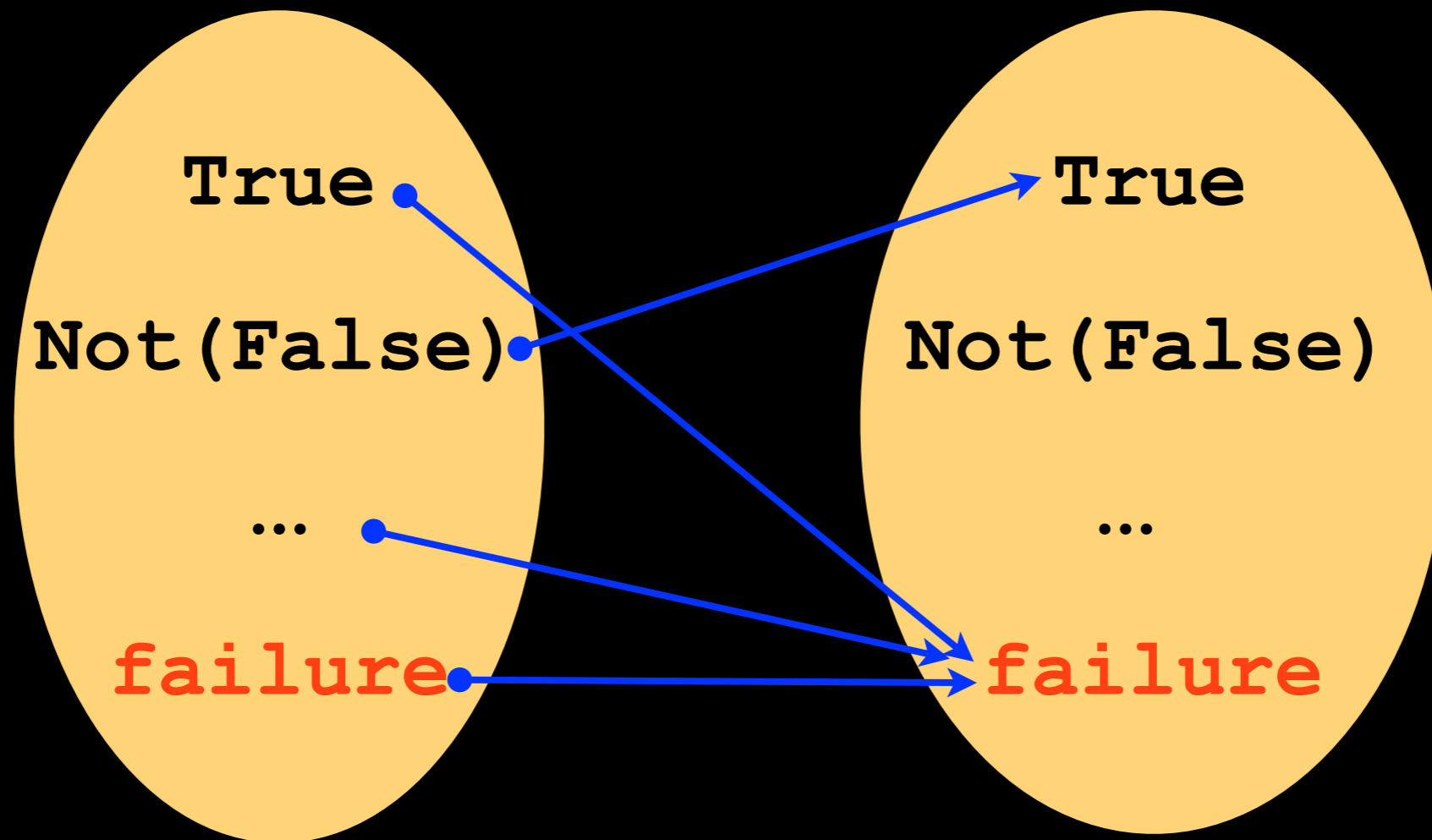


# A transformation is...

a partial function from  
**terms** to terms



# Alternatively... **failure!**



# Meta-variables in transformations

**True And x -> x**

meta-variable

# Applying a transformation, success

**True And x  $\rightarrow$  x**

**True And True**

transformation tool



**True**

# Pattern-matching and replacement

**True And x -> x**

Term	Match
True And True	x = True
False And True	no match
True And (True Or False)	x = (True Or False)
Not (True And False)	no match

# Applying a transformation, failure

True And x  $\rightarrow$  x

False And True

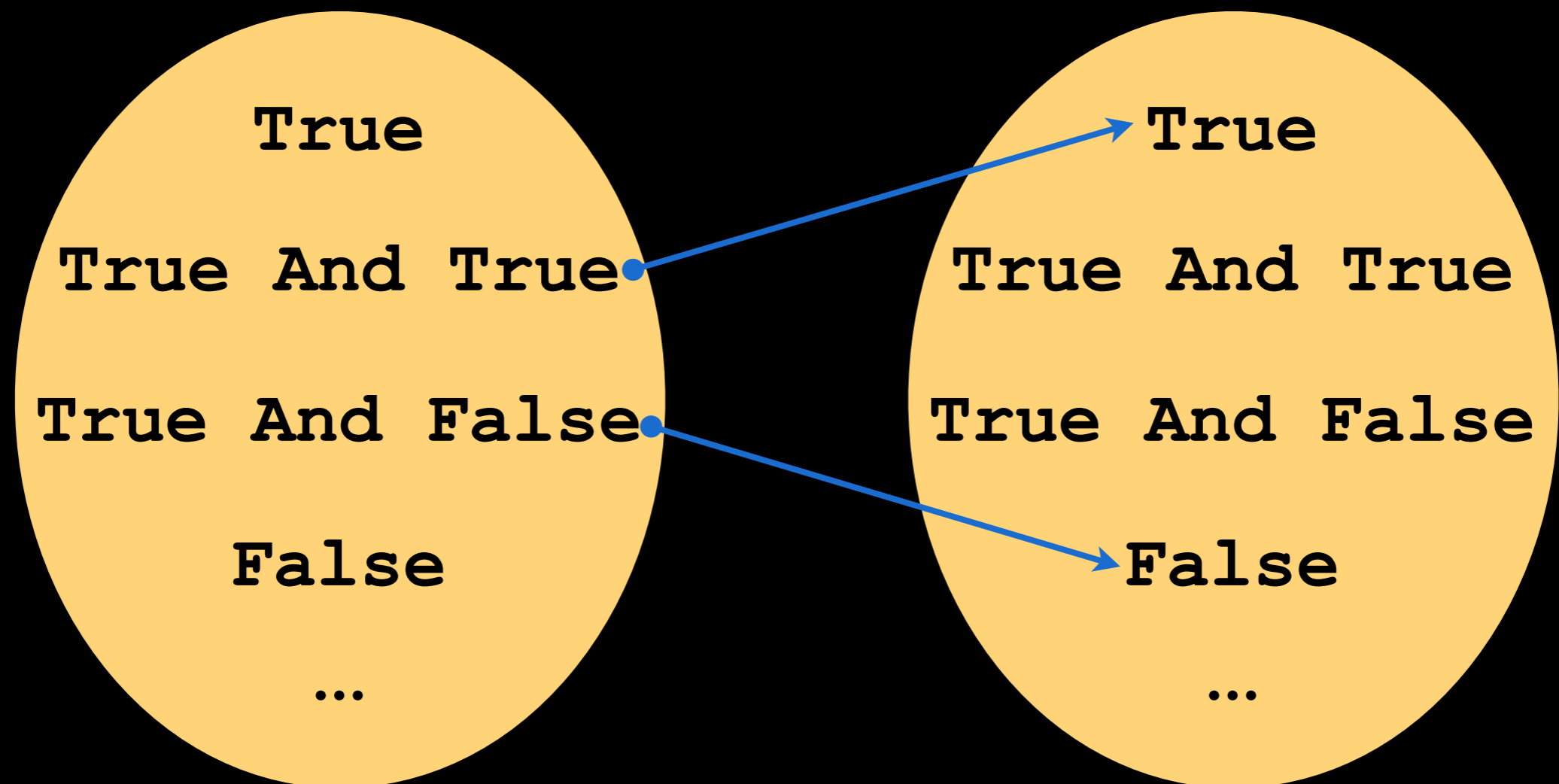
transformation tool



**failure!**

# One transformation mapping several terms

`True And x -> x`



# Java transformations in Stratego/XT

$\rightarrow$  `| [ if (e) stm ] |`  
`| [ if (e) {stm} ] |`

`stm (stm1 , ...)` and `e (e1, ...)`  
are predefined meta-variables



# Which variables to use?

## variables

```
"stm" [0-9\']* -> Stm {prefer}
"bstm" [0-9\']* "*" -> BlockStm* {prefer}
```

## variables

```
"ent" [0-9]* -> Entity {prefer}
"prop" [0-9]* -> Property {prefer}
"prop" [0-9]* "*" -> Property* {prefer}
"srt" [0-9]* -> Sort {prefer}
[xyz][0-9]* -> Id {prefer}
[xyz]"_" [A-Za-z0-9]+ -> Id {prefer}
```

# Concrete syntax on transformations

## quotation and anti-quotation

### context-free syntax

```
"|[" ClassBodyDec "]" -> E {cons("ToMetaExpr")}  
"|[" ClassBodyDec* "]" -> E {cons("ToMetaLExpr")}  
"~" E -> ClassBodyDec {cons("FromMetaExpr")}  
"~*" E -> ClassBodyDec* {cons("FromMetaLExpr")}
```

### context-free syntax

```
"web" "|[" Entity "]" -> E {cons("ToMetaExpr")}  
"|[" Entity "]" -> E {cons("ToMetaExpr")}
```

# Transformations = rewrite rules

## rules

### AddBlockToIf:

```
| [ if (e) stm ] |  
-> | [ if (e) {stm} ] |
```

### AddBlockToIfElse:

```
| [ if (e) stm else stm1 ] |  
-> | [ if (e) {stm}  
      else {stm1} ] |
```

# Referring twice to a meta-variable

rules

**AddBlockToIfElse:**

```
|| [ if (e) stm else stm ] |  
-> || [ if (e) {stm}  
      else {stm} ] |
```

# Applying a transformation, bug!

`| [ if (e) stm ] |`  
→ `| [ if (e) {stm} ] |`

`if (a<5) {a = a+1;}`

transformation tool

`if (a<5) {{a = a+1;}}`

# Conditional transformations

**rules**

**AddBlockToIf:**

| [ if (e) stm ] |  
-> | [ if (e) {stm} ] |

**where**

<notBlock> stm

# Applying a conditional transformation, failure

```
|| [ if (e) stm ] ||  
-> || [ if (e) {stm} ] ||  
where <notBlock> stm
```

```
if (a<5) {a = a+1;}
```

transformation tool



**failure!**

# Applying a conditional transformation, success

```
|| [ if (e) stm ] |  
-> || [ if (e) {stm} ] |  
where <notBlock> stm
```

```
if (a<5) a = a+1;
```

transformation tool



```
if (a<5) {a = a+1;}
```



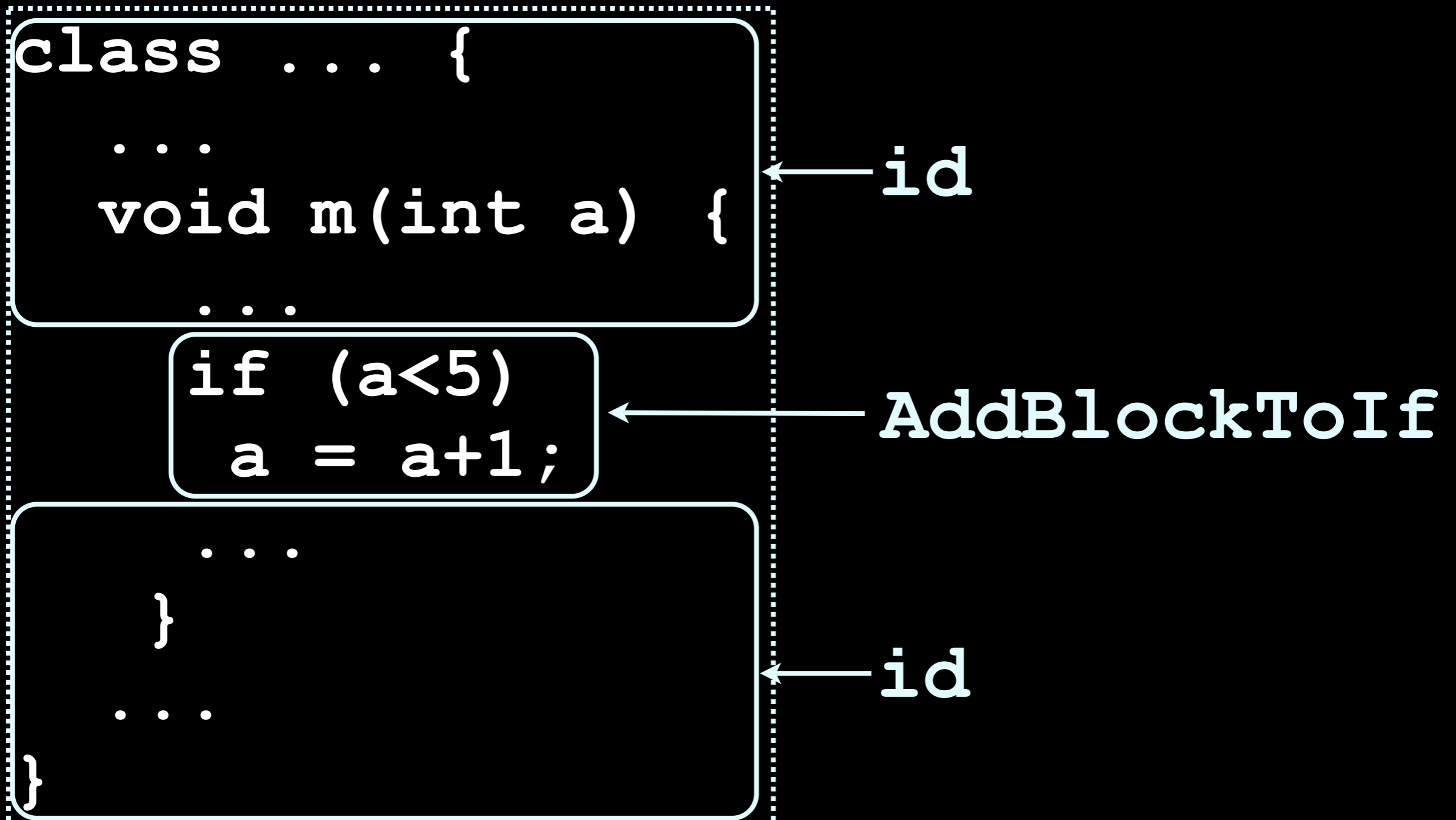
# Applying to a program

```
    |[ if (e) stm ]|  
-> |[ if (e) {stm} ]|  
where <notBlock> stm  
...if (a<5) a = a+1;...
```

transformation tool

**failure!**

# Combining transformations



# Transformation strategies = complex transformations

```
strategies  
  add-block =  
    topdown (try (AddBlockToIf) )
```

# Applying strategies

add-block

```
...if (a<5) a = a+1;...
```

transformation tool



```
...if (a<5) {a = a+1;}...
```

# Strategies and rules

```
strategies...
```

```
  add-block =
```

```
    topdown (try (AddBlockToIf))
```

```
rules
```

```
  AddBlockToIf:
```

```
    |[ if (e) stm ]|  
-> |[ if (e) {stm} ]|
```

```
  where
```

```
    <notBlock> stm
```

# Conditions as strategy applications

## strategies

```
...  
notBlock =  
  not(? (| [ {bstm*} ] |))
```

## rules

**AddBlockToIf:**

```
| [ if (e) stm ] |  
-> | [ if (e) {stm} ] |
```

**where**

```
<notBlock> stm
```

**bstm** is a predefined meta-variable for block statements

# No ambiguity now!

**strategies**

```
...  
notBlock =  
  not (? (stm | [ {bstm*} ] |))
```

**rules**

**AddBlockToIf:**

```
| [ if (e) stm ] |  
-> | [ if (e) {stm} ] |
```

**where**

```
<notBlock> stm
```

# Modules

```
module add-block-if
imports libstratego-lib libjava-front
strategies
  main =
    io-java2java-wrap (add-block)
  add-block =
    topdown (try (AddBlockToIf))
  notBlock = not (? (stm | [ {bstm*} ] |))
rules
  AddBlockToIf:
    | [ if (e) stm ] |
  -> | [ if (e) {stm} ] |
  where
    <notBlock> stm
```



# Predefined strategy operators so far

- `id`
- `? ( _ )`
- `not ( _ )`
- `topdown ( _ )`
- `try ( _ )`
- `io-java2java-wrap ( _ )`

# Applying more than one rule...

nondeterministic  
choice operator



## strategies

```
...  
add-block =  
  topdown (try (AddBlockToIf +  
               AddBlockToIfElse))
```

## rules

```
AddBlockToIf: ...  
AddBlockToIfElse: ...
```

# with auxiliary strategy...

**strategies**

...

```
if-rules = AddBlockToIf +  
           AddBlockToIfElse
```

```
add-block =  
  topdown (try (if-rules))
```

**rules**

```
AddBlockToIf: ...
```

```
AddBlockToIfElse: ...
```

# or with same rule name...

**strategies**

...

add-block =

topdown (try (AddBlockToIf) )

**rules**

AddBlockToIf: ...

AddBlockToIf: ...

rules are tried  
(undefined order) until  
one succeeds, or all fail

same holds  
for strategies

# Basic transformation concepts

Paulo Borba  
Informatics Center  
Federal University of Pernambuco

[phmb@cin.ufpe.br](mailto:phmb@cin.ufpe.br) ♦ [twitter.com/pauloborba](https://twitter.com/pauloborba)

# Software Productivity

Paulo Borba  
Informatics Center  
Federal University of Pernambuco

[phmb@cin.ufpe.br](mailto:phmb@cin.ufpe.br) ♦ [twitter.com/pauloborba](https://twitter.com/pauloborba)