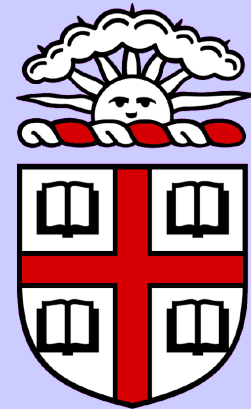


Consistent Updates from Atomic Composition

**C. Gordon, L. Meyerovich, J. Weinberger,
S. Krishnamurthi**

14th Intl. ASM Workshop
06.07.2007



**Brown
University**

Consistent Updates from Atomic Composition

Motivation

CC 05: Paper Assignments

http://continue.local/servlets/continue/continue.ss/137538/?id11*k1-759779008

ContinueCon 2005

Login: gonzo (reviewer, admin)

[All Papers](#) [Review](#) [Bidding](#) [Assign](#) [Decide](#) [Meeting](#) [Admin](#)

4 unassigned papers

[Spreadsheet View](#)

← Previous Next →
Page: 1

27 of 27 Papers
Show: 5 10 20 30 all

| # | Authors | Title | Reviewers |
|---|---|--|-----------|
| 128 | John Hadden, Gaffer Winer, and Art McFadden | Greedy Model-Checkification in ML | 0 |
| 0/0: No reviews assigned | | | |
| 136 | Pippin Gaiman | Fearful Rock Bands in Logo | 3 |
| 1/3: Assistant Beaker, Bunsan Honeydew, Dr. Teeth (AOK) | | | |
| 144 | Myra Vandelay | Dominating Model-Verifying in JavaScript | 4 |
| 2/4: Assistant Beaker, The Great Gonzo, Ms. Piggy (reviewed), Roph the Dog (reviewed) | | | |
| 147 | Douglas Gamage | Dominating Verification in ML | 4 |
| 3/4: Assistant Beaker (COK), The Great Gonzo (COK), Ms. Piggy (BIZ), Rizzo | | | |
| 151 | Jimmy Gamage | Persuasive Testing in Perl | 3 |

Motivation: Dynamic Access Control

The screenshot shows a web browser window titled "CC 05: Paper Assignments" with the URL `http://continue.local/servlets/continue/continue.ss/137538/?id11*k1-759779008`. The page header displays "ContinueCon 2005" and the user is logged in as "gonzo (reviewer, admin)". A navigation bar contains buttons for "All Papers", "Review", "Bidding", "Assign", "Decide", "Meeting", and "Admin". The "Assign" button is currently selected.

Below the navigation bar, it indicates "4 unassigned papers" and "Spreadsheet View". Navigation controls show "← Previous Next →" and "Page: 1". On the right, it says "27 of 27 Papers" and "Show: 5 10 20 30 all".

| # | Authors | Title | Reviewers |
|---|---|--|-----------|
| 128 | John Hadden, Gaffer Winer, and Art McFadden | Greedy Model-Checkification in ML | 0 |
| 0/0: No reviewers assigned | | | |
| 136 | Pippin Gaiman | Fearful Rock Bands in Logo | 3 |
| 1/3: Assistant Beaker, Bunsan Honeydew, Dr. Teeth (A/X) | | | |
| 144 | Myra Vandelay | Dominating Model-Verifying in JavaScript | 4 |
| 2/4: Assistant Beaker, The Great Gonzo, Ms. Piggy (reviewed), Roph the Dog (reviewed) | | | |
| 147 | Douglas Gamage | Dominating Verification in ML | 4 |
| 3/4: Assistant Beaker (C/X), The Great Gonzo (C/X), Ms. Piggy (B/Z), Rizzo | | | |
| 151 | Jimmy Gamage | Persuasive Testing in Perl | 3 |

Example: Change Role

IF `ChangeJobToAdmin(user)` AND `isReviewer(user)`

Example: Change Role

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

Example: Change Role

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

ChangeJobToAdmin(fred) \wedge isReviewer(fred)

Example: Change Role

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
isAdmin(*user*)
NOT isReviewer(*user*)

ChangeJobToAdmin(fred) \wedge isReviewer(fred)
→
{ +isAdmin(fred), -isReviewer(fred) }

Example: Add Reviewer

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

Example: Add Reviewer

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*)

Example: Add Reviewer

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)

Example: Add Reviewer

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*)

Example: Add Reviewer

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*)

Example: Add Reviewer

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

Example: Add Reviewer

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

AddPaperReviewer(*bob*, *iliad*) \wedge ...
→

Example: Add Reviewer

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

AddPaperReviewer(*bob*, *iliad*) \wedge ...

→

Example: Add Reviewer

IF $\text{ChangeJobToAdmin}(user)$ AND $\text{isReviewer}(user)$ THEN
 $\text{isAdmin}(user)$
 NOT $\text{isReviewer}(user)$

IF $\text{AddPaperReviewer}(u1, p)$ AND $\text{isReviewer}(u1)$
AND $\text{isAuthor}(u2, p)$ AND NOT $\text{Conflict}(u1, u2)$ THEN
 $\text{isPaperReviewer}(u1, p)$

{ $+\text{isPaperReviewer}(bob, iliad)$ }

$\text{AddPaperReviewer}(bob, iliad) \wedge \dots$
→
{ $+\text{isPaperReviewer}(bob, iliad)$ }

Example: Remove Admin

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

Example: Remove Admin

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

IF RemoveAdmin(*u*)

Example: Remove Admin

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

IF RemoveAdmin(*u*) THEN
 NOT isAdmin(*u*)

Example: Remove Admin

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
isAdmin(*user*)
NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
isPaperReviewer(*u1*, *p*)

IF RemoveAdmin(*u*) THEN
NOT isAdmin(*u*)

RemoveAdmin(*alice*)

→

Example: Remove Admin

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
isAdmin(*user*)
NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
isPaperReviewer(*u1*, *p*)

IF RemoveAdmin(*u*) THEN
NOT isAdmin(*u*)

{ -isAdmin(*alice*) }

RemoveAdmin(*alice*)

→

Example: Remove Admin

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
isAdmin(*user*)
NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
isPaperReviewer(*u1*, *p*)

IF RemoveAdmin(*u*) THEN
NOT isAdmin(*u*)

{ -isAdmin(*alice*) }

RemoveAdmin(*alice*)

→

{ -isAdmin(*alice*) }

Example: Concurrent Requests

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

IF RemoveAdmin(*u*) THEN
 NOT isAdmin(*u*)

Example: Concurrent Requests

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

IF RemoveAdmin(*u*) THEN
 NOT isAdmin(*u*)

Example: Concurrent Requests

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

IF RemoveAdmin(*u*) THEN
 NOT isAdmin(*u*)

Example: Concurrent Requests

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

IF RemoveAdmin(*u*) THEN
 NOT isAdmin(*u*)

{ -isAdmin(*alice*) }

Example: Concurrent Requests

{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **+isPaperReviewer**(bob, *iliad*) }

{ **-isAdmin**(alice) }

Example: Concurrent Requests

{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **+isPaperReviewer**(bob, iliad) }

{ **-isAdmin**(alice) }

Example: Concurrent Requests

{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **+isPaperReviewer**(bob, iliad) }

{ **-isAdmin**(alice) }

→

{ **+isAdmin**(fred), **-isReviewer**(fred), **+isPaperReviewer**(bob, iliad),
-isAdmin(alice) }

Example: Concurrent Requests

{ +isAdmin(*fred*), -isReviewer(*fred*) }

{ +isPaperReviewer(*bob*, *iliad*) }

{ ~~-isAdmin(*alice*)~~ }

→

{ +isAdmin(*fred*), -isReviewer(*fred*), +isPaperReviewer(*bob*, *iliad*),
-isAdmin(*alice*) }

Example: Concurrent Requests

{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **+isPaperReviewer**(bob, iliad) }

{ ~~**-isAdmin**(alice)~~ **-isAdmin**(fred) }

→

{ **+isAdmin**(fred), **-isReviewer**(fred), **+isPaperReviewer**(bob, iliad),
-isAdmin(alice) }

Example: Concurrent Requests

{ +isAdmin(*fred*), -isReviewer(*fred*) }

{ +isPaperReviewer(*bob*, *iliad*) }

{ ~~-isAdmin(*alice*)~~ -isAdmin(*fred*) }

→

{ +isAdmin(*fred*), -isReviewer(*fred*), +isPaperReviewer(*bob*, *iliad*),
~~-isAdmin(*alice*)~~ }

Example: Concurrent Requests

{ +isAdmin(*fred*), -isReviewer(*fred*) }

{ +isPaperReviewer(*bob*, *iliad*) }

{ ~~-isAdmin(*alice*)~~ -isAdmin(*fred*) }

→

{ +isAdmin(*fred*), -isReviewer(*fred*), +isPaperReviewer(*bob*, *iliad*),
~~-isAdmin(*alice*)~~ -isAdmin(*fred*) }

Example: Concurrent Requests

{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **+isPaperReviewer**(bob, iliad) }

{ ~~**-isAdmin**(alice)~~ **-isAdmin**(fred) }

→

{ **+isAdmin**(fred), ~~-isReviewer~~(fred), ~~+isPaperReviewer~~(bob, iliad),
~~-isAdmin~~(alice) **-isAdmin**(fred) }

Options: Total Failure


{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **+isPaperReviewer**(bob, iliad) }

{ ~~**-isAdmin**(alice)~~ **-isAdmin**(fred) }

→

{ **+isAdmin**(fred), ~~**-isReviewer**(fred)~~, **+isPaperReviewer**(bob, iliad),
~~**-isAdmin**(alice)~~ **-isAdmin**(fred) }



Options:

{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **+isPaperReviewer**(bob, iliad) }

{ ~~**-isAdmin**(alice)~~ **-isAdmin**(fred) }

→

{ **+isAdmin**(fred), **-isReviewer**(fred), **+isPaperReviewer**(bob, iliad),
-isAdmin(fred) }

Options: Silent Failure

```
{ +isAdmin(fred), -isReviewer(fred) }
```

```
{ +isPaperReviewer(bob, iliad) }
```

```
{ -isAdmin(alice) -isAdmin(fred) }
```

→

```
{  
    -isReviewer(fred), +isPaperReviewer(bob, iliad)  
}
```

Under Either Approach:

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

IF RemoveAdmin(*u*) THEN
 NOT isAdmin(*u*)

Under Either Approach:

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
isAdmin(*user*)
NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

IF RemoveAdmin(*u*) THEN
NOT isAdmin(*u*)

{ -isAdmin(*fred*) }

Under Either Approach:

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
isAdmin(*user*)
NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

IF RemoveAdmin(*u*) THEN
NOT isAdmin(*u*)

{ -isAdmin(*fred*) }

Under Either Approach:

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
isAdmin(*user*)
NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

IF RemoveAdmin(*u*) AND NOT ChangeJobToAdmin(*u*) THEN
NOT isAdmin(*u*)

Under Either Approach:

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
isAdmin(*user*)
NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

IF RemoveAdmin(*u*) AND NOT ChangeJobToAdmin(*u*) THEN
NOT isAdmin(*u*)

Under Either Approach:

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
isAdmin(*user*)
NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

IF RemoveAdmin(*u*) AND NOT ChangeJobToAdmin(*u*) THEN
NOT isAdmin(*u*)

Problem:

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

IF RemoveAdmin(*u*) AND NOT ChangeJobToAdmin(*u*) THEN
 NOT isAdmin(*u*)

Problem: Removing Spec

IF AddPaperReviewer($u1, p$) AND isReviewer($u1$)
AND isAuthor($u2, p$) AND NOT Conflict($u1, u2$) THEN
isPaperReviewer($u1, p$)

IF RemoveAdmin(u) AND NOT ChangeJobToAdmin(u) THEN
NOT isAdmin(u)

Problem:

IF AddPaperReviewer($u1, p$) AND isReviewer($u1$)
AND isAuthor($u2, p$) AND NOT Conflict($u1, u2$) THEN
isPaperReviewer($u1, p$)

IF RemoveAdmin(u) AND NOT ChangeJobToAdmin(u) THEN
NOT isAdmin(u)

Problem: Adding Spec

**IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
isAdmin(*user*)
NOT isReviewer(*user*)**

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
isPaperReviewer(*u1*, *p*)

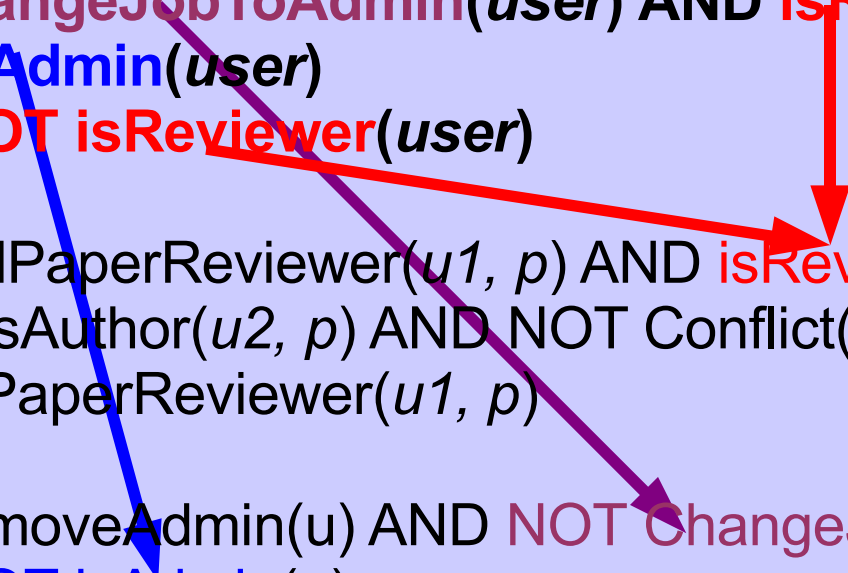
IF RemoveAdmin(*u*) AND NOT ChangeJobToAdmin(*u*) THEN
NOT isAdmin(*u*)

Problem: Adding Spec

IF **ChangeJobToAdmin**(*user*) AND **isReviewer**(*user*) THEN
isAdmin(*user*)
NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*) AND **isReviewer**(*u1*)
AND **isAuthor**(*u2*, *p*) AND NOT **Conflict**(*u1*, *u2*) THEN
isPaperReviewer(*u1*, *p*)

IF **RemoveAdmin**(*u*) AND NOT **ChangeJobToAdmin**(*u*) THEN
NOT isAdmin(*u*)



Problem: Adding Spec

IF **ChangeJobToAdmin**(*user*) AND isReviewer(*user*) THEN
isAdmin(*user*)
NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
isPaperReviewer(*u1*, *p*)

IF RemoveAdmin(*u*) AND NOT **ChangeJobToAdmin**(*u*) THEN
NOT isAdmin(*u*)



Problem: Adding Spec

IF **ChangeJobToAdmin**(*user*) AND isReviewer(*user*) THEN
isAdmin(*user*)
NOT isReviewer(*user*)

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
isPaperReviewer(*u1*, *p*)

IF RemoveAdmin(*u*) AND NOT **ChangeJobToAdmin**(*u*) THEN
NOT isAdmin(*u*)



Properties?

{ **+isAdmin**(fred), **-isReviewer**(fred) }

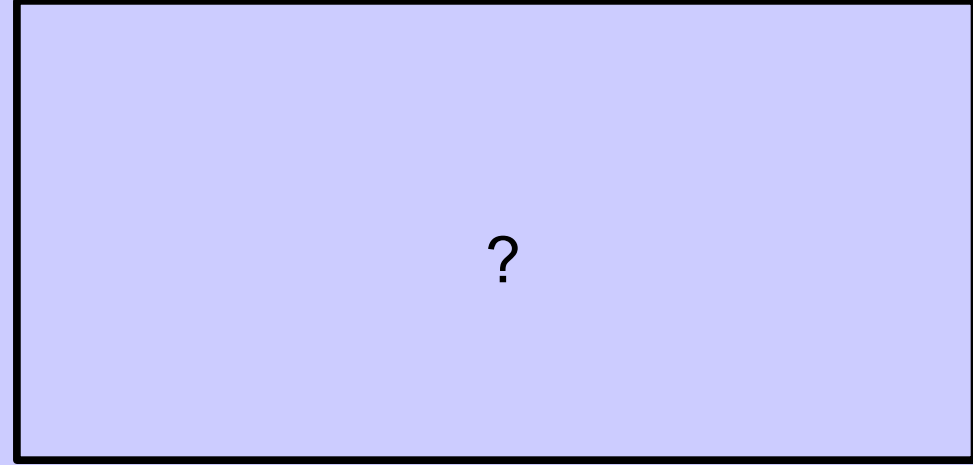
{ **+isPaperReviewer**(bob, iliad) }

{ **-isAdmin**(fred) }



?

Properties:



Properties:



{ **+isPaperReviewer**(*bob*, *iliad*) }

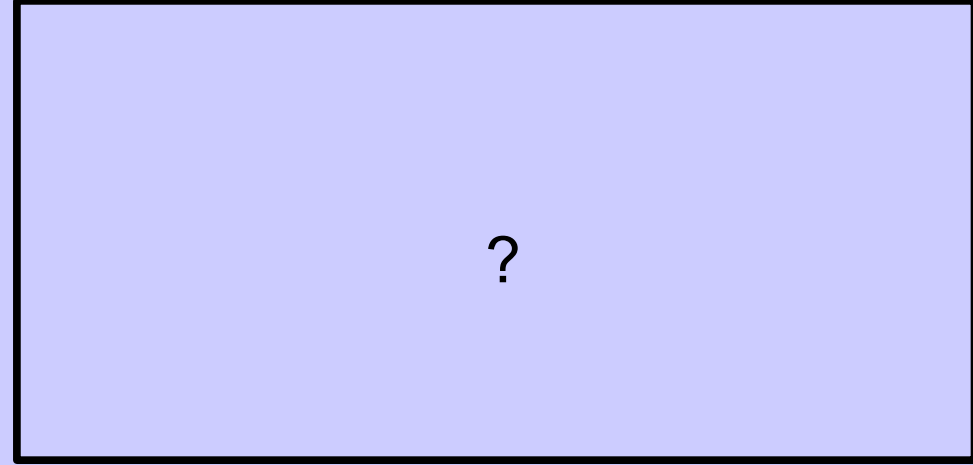
Properties: Consistency

{ **+isPaperReviewer**(*bob*, *iliad*) }



{ **+isPaperReviewer**(*bob*, *iliad*) }

Properties:



Properties:



{ **+isAdmin**(fred), **-isAdmin**(fred) }

Properties: Consistent Inconsistency

{ **+isAdmin**(fred), **-isAdmin**(fred) }

→

{ **+isAdmin**(fred), **-isAdmin**(fred) }

Properties: Consistent Inconsistency

{ **+isAdmin**(fred) }

{ **-isAdmin**(fred) }



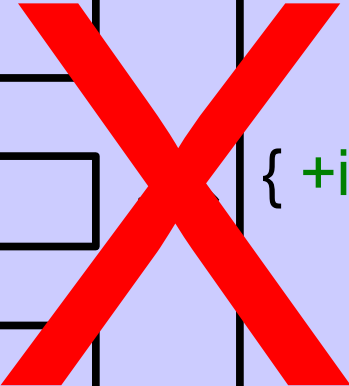
{ **+isAdmin**(fred), **-isAdmin**(fred) }

Properties: Consistent Inconsistency

{ **+isAdmin**(fred) }

{ **-isAdmin**(fred) }

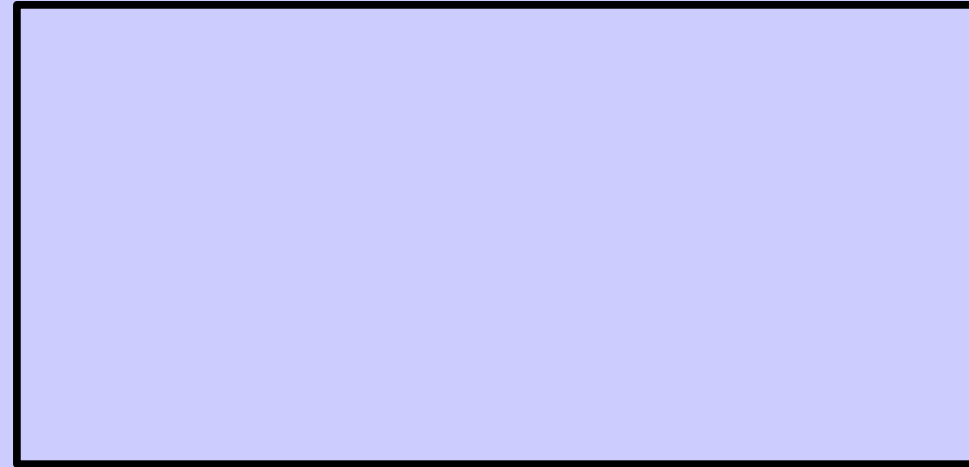
{ **+isAdmin**(fred), **-isAdmin**(fred) }



Properties: Consistent Inconsistency

{ **+isAdmin**(fred) }

{ **-isAdmin**(fred) }



Properties: Consistent Inconsistency

{ **+isAdmin**(fred) }

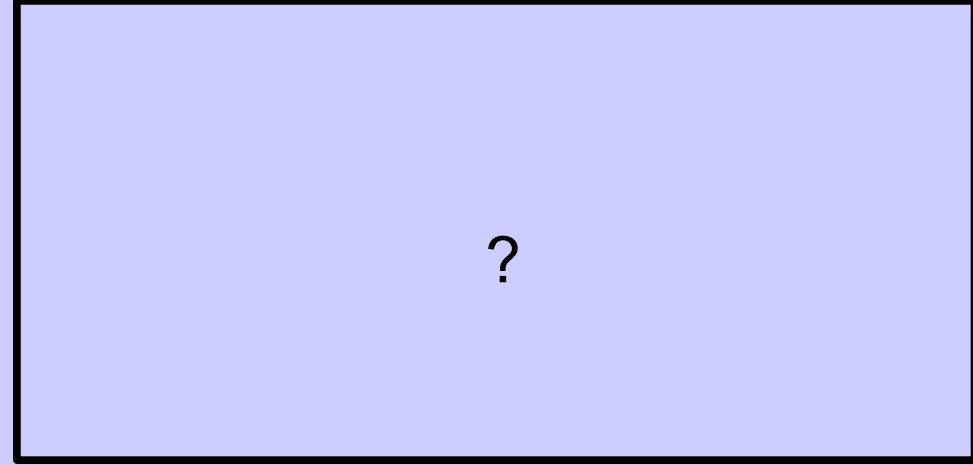
{ **-isAdmin**(fred) }

{ **+isAdmin**(fred), **-isAdmin**(fred) }



{ **+isAdmin**(fred), **-isAdmin**(fred) }

Properties:



Properties:

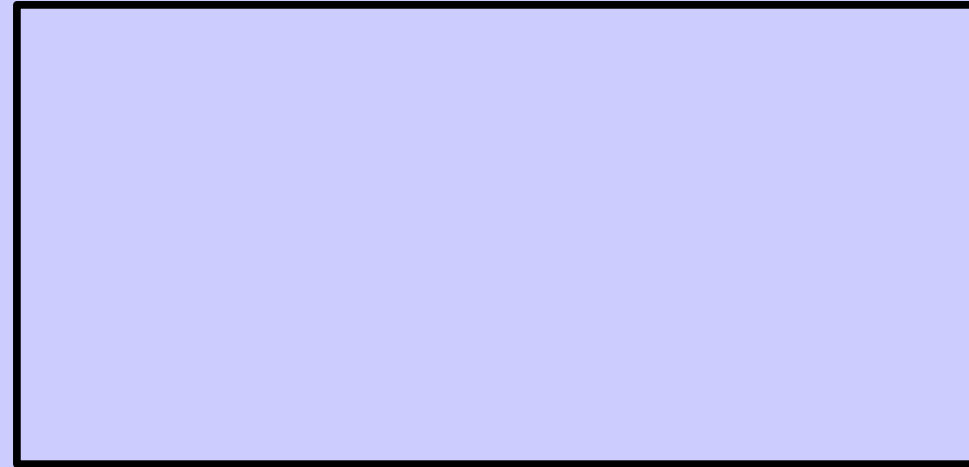
{ **+isAdmin**(fred), **-isReviewer**(fred) }



Properties:

{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **-isAdmin**(fred) }



Properties:

{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **-isAdmin**(fred) }



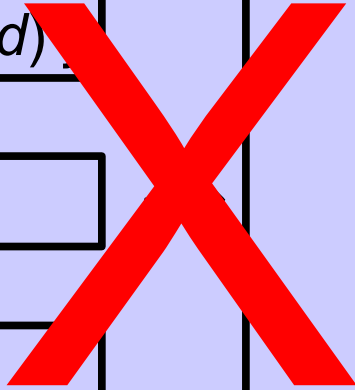
{ **-isReviewer**(fred) }

Properties:

{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **-isAdmin**(fred) }

{ **-isReviewer**(fred) }



Properties: Atomicity

{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **-isAdmin**(fred) }



Properties: Atomicity

{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **+isPaperReviewer**(bob, iliad) }

{ **-isAdmin**(fred) }



Properties: Atomicity

{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **+isPaperReviewer**(bob, iliad) }

{ **-Admin**(fred) }



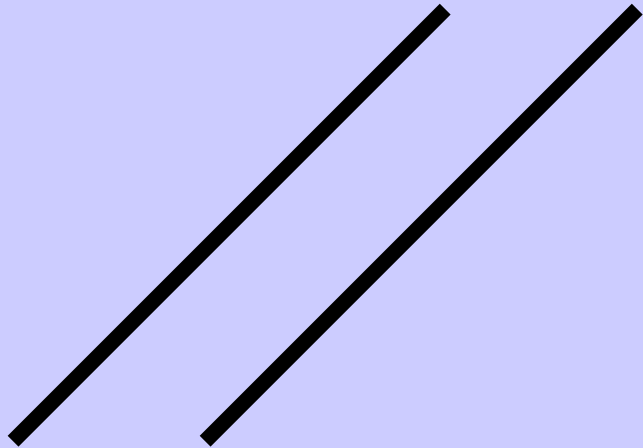
{ **+isPaperReviewer**(bob, iliad) }

Properties: Quadratic Complexity

with respect to number of rules

Approaches: Feature Composition

Marianna Nicolosi-Asmundo and Elvinia Riccobene, 2003



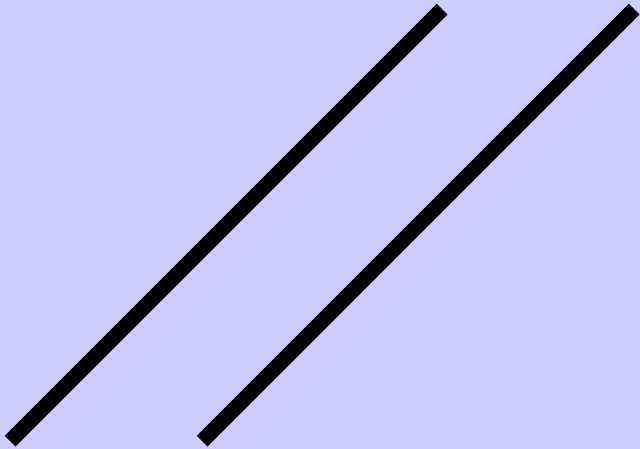
{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **+isPaperReviewer**(bob, *iliad*) }

{ **-isAdmin**(fred) }

Approaches: Feature Composition

Marianna Nicolosi-Asmundo and Elvinia Riccobene, 2003



{ *+isAdmin(fred)*, *-isReviewer(fred)* }

+isPaperReviewer(bob, iliad)

{ *-isAdmin(fred)* }

Approaches: Feature Composition

Marianna Nicolosi-Asmundo and Elvinia Riccobene, 2003

inconsistent
updates

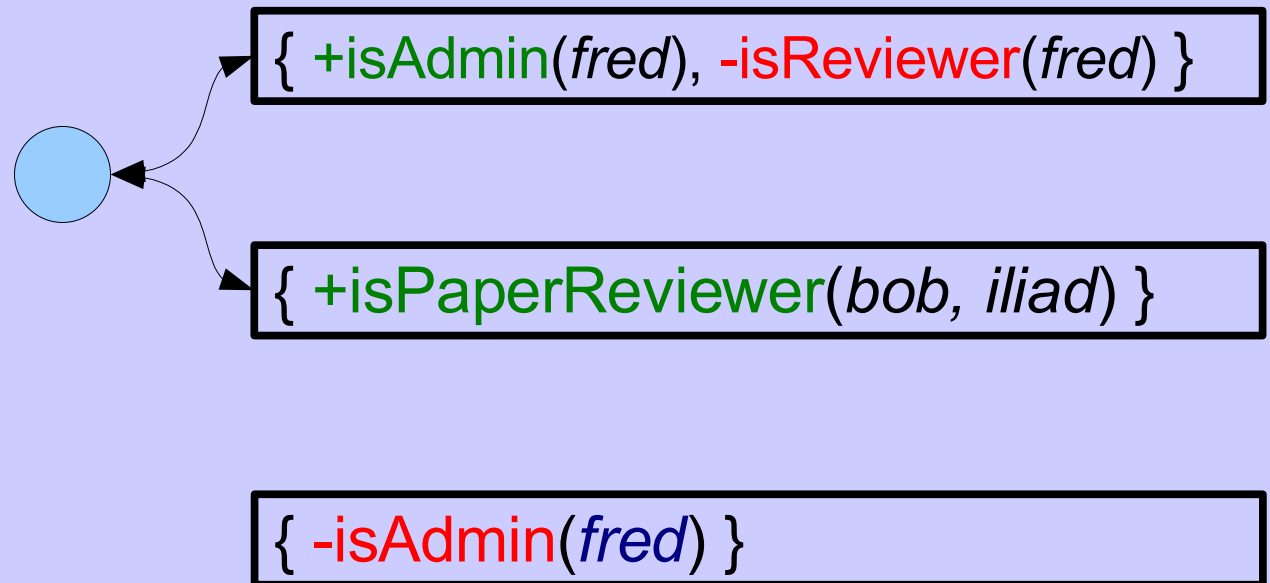
```
{ +isAdmin(fred), -isReviewer(fred)
```

```
+isPaperReviewer(bob, iliad)
```

```
{ -isAdmin(fred) }
```

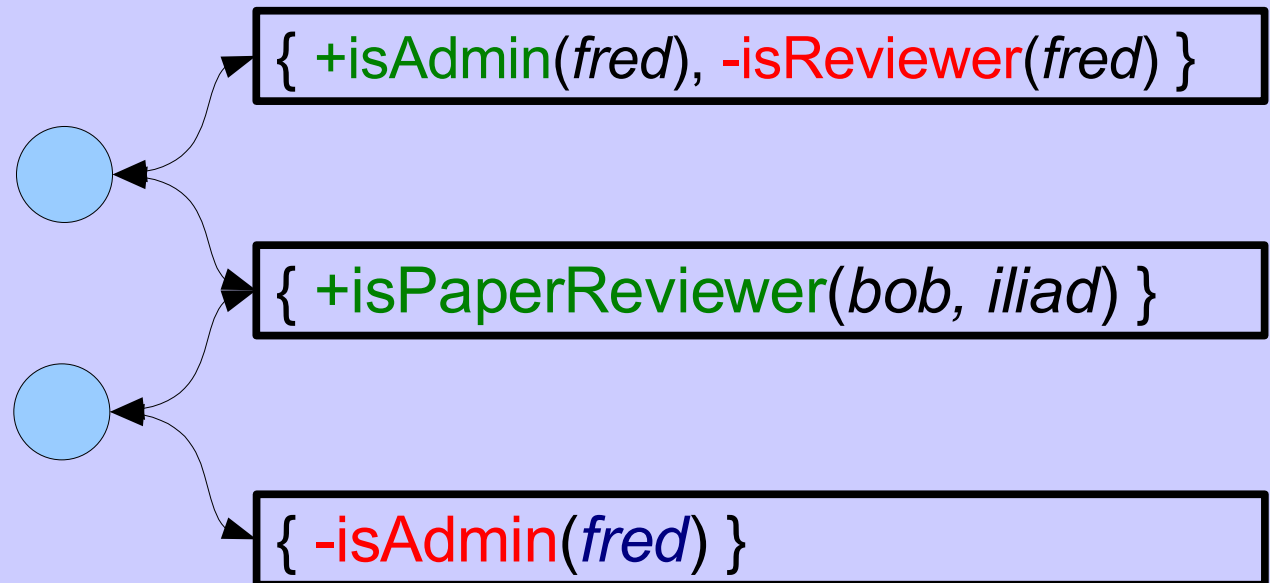
Approaches: Component Composition

Marianna Nicolosi-Asmundo and Elvinia Riccobene, 2003



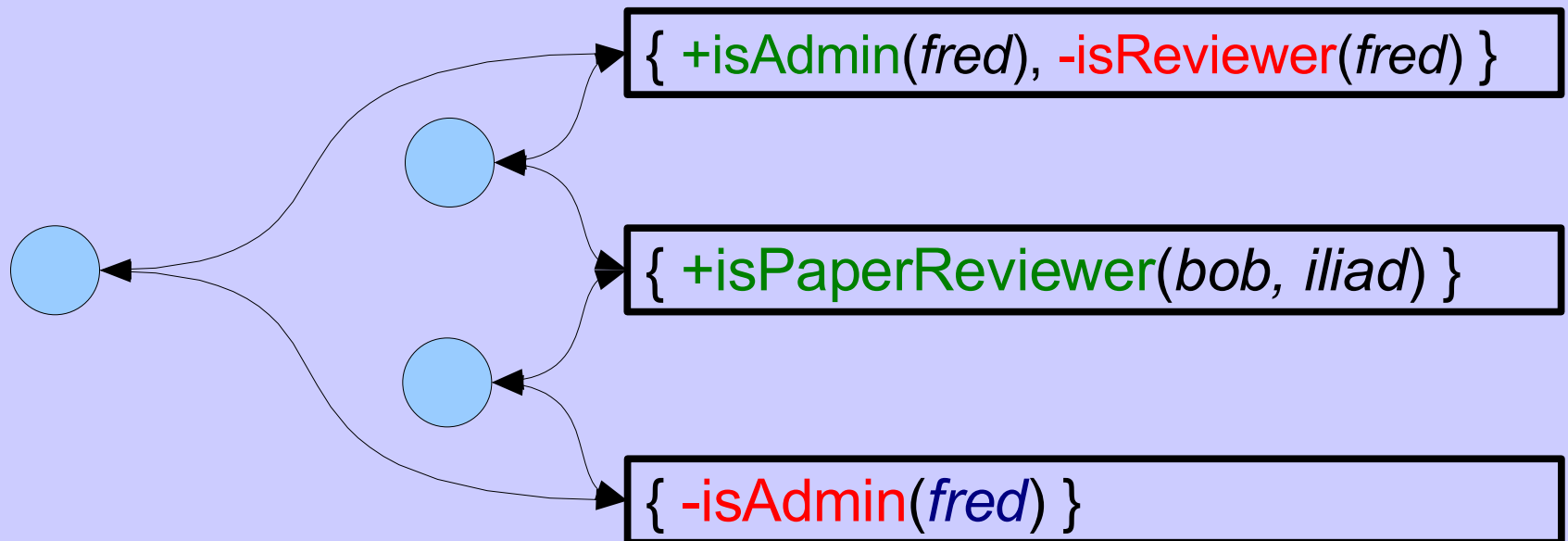
Approaches: Component Composition

Marianna Nicolosi-Asmundo and Elvinia Riccobene, 2003



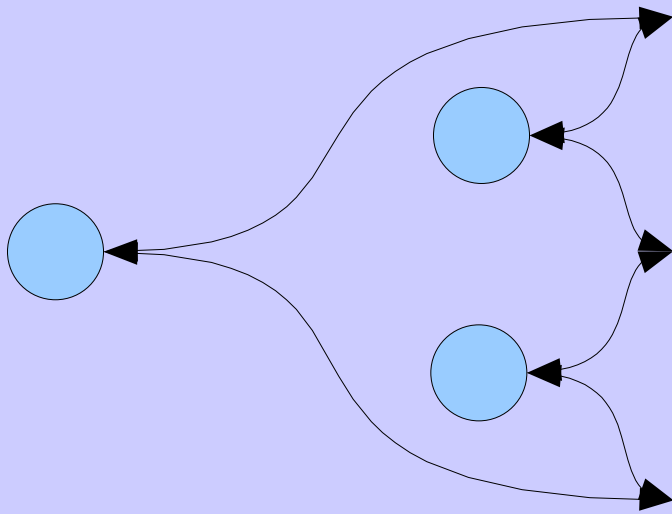
Approaches: Component Composition

Marianna Nicolosi-Asmundo and Elvinia Riccobene, 2003



Approaches: Component Composition

Marianna Nicolosi-Asmundo and Elvinia Riccobene, 2003



over-specification
(handwriting)

Approaches: Partially Ordered Runs

Y. Gurevich and D. Rosenzweig, 2000



{ +isPaperReviewer(*bob*, *iliad*) }

{ +isAdmin(*fred*), -isReviewer(*fred*) }

{ -isAdmin(*fred*) }

time



Approaches: Partially Ordered Runs

Y. Gurevich and D. Rosenzweig, 2000



{ +isPaperReviewer(*bob*, *iliad*) }

{ +isAdmin(*fred*), -isReviewer(*fred*) }

{ -isAdmin(*fred*) }

explicit / less concurrency

What Do We Want?

// Chunk 1

```
IF ChangeJobToAdmin(user) AND isReviewer(user) THEN  
  isAdmin(user)  
  NOT isReviewer(user)
```

What Do We Want?

// Chunk 1

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

What Do We Want?

// Chunk 1

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

// Chunk 2

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

What Do We Want?

// Chunk 1

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

// Chunk 2

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

// Chunk 3

IF RemoveAdmin(*u*) THEN
 NOT isAdmin(*u*)

{ -isAdmin(*fred*) }

Modules!

Module

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

Module

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

Module

IF RemoveAdmin(*u*) THEN
 NOT isAdmin(*u*)

{ -isAdmin(*fred*) }

Modules!

{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **+isPaperReviewer**(bob, *iliad*) }

{ **-isAdmin**(fred) }

Modules!

{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **+isPaperReviewer**(bob, iliad) }

{ **-isAdmin**(fred) }

Modules!

{ +isAdmin(*fred*), -isReviewer(*fred*) }

{ +isPaperReviewer(*bob*, *iliad*) }

{ -isAdmin(*fred*) }

→

{ +isAdmin(*fred*), -isReviewer(*fred*) }

{ +isPaperReviewer(*bob*, *iliad*) }

{ -isAdmin(*fred*) }

Modules!

{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **+isPaperReviewer**(bob, iliad) }

{ **-isAdmin**(fred) }

→

{ **+isAdmin**(fred), ~~**-isReviewer**~~(fred) }

{ **+isPaperReviewer**(bob, iliad) }

{ ~~**-isAdmin**~~(fred) }

Modules!

{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **+isPaperReviewer**(bob, iliad) }

{ **-isAdmin**(fred) }

→

{ **+isPaperReviewer**(bob, iliad) }

Modules!

{ **+isAdmin**(fred), **-isReviewer**(fred) }

{ **+isPaperReviewer**(bob, iliad) }

{ **-isAdmin**(fred) }



{ **+isPaperReviewer**(bob, iliad) }

(ACID) Transactions

Module

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

Module

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

Module

IF RemoveAdmin(*u*) THEN
 NOT isAdmin(*u*)

{ -isAdmin(*fred*) }

Properties Revisted

- Consistency
- Consistent Inconsistency
- Atomicity
- Quadratic Complexity

Properties Revisted

- ✓ Consistency
- ✓ Consistent Inconsistency
- ✓ Atomicity
- ✓ Quadratic Complexity

Extensions: Prioritization

Module

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

Module

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

Module

IF RemoveAdmin(*u*) THEN
 NOT isAdmin(*u*)

{ -isAdmin(*fred*) }

Extensions: Prioritization

Module

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

Module

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

Module

IF RemoveAdmin(*u*) THEN
 NOT isAdmin(*u*)

{ -isAdmin(*fred*) }

Extensions: Prioritization

Module **High**

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

Module

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

Module

IF RemoveAdmin(*u*) THEN
 NOT isAdmin(*u*)

{ -isAdmin(*fred*) }

Extensions: Prioritization

Module **High**

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

Module **Mid**

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

Module

IF RemoveAdmin(*u*) THEN
 NOT isAdmin(*u*)

{ -isAdmin(*fred*) }

Extensions: Prioritization

Module **High**

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

Module **Mid**

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

Module **Low**

IF RemoveAdmin(*u*) THEN
 NOT isAdmin(*u*)

{ -isAdmin(*fred*) }

Extensions: Prioritization

Module High

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

Module Mid

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

Module Low

IF RemoveAdmin(*u*) THEN
 NOT isAdmin(*u*)

{ -isAdmin(*fred*) }

Extensions: Prioritization

Module High

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

{ +isAdmin(*fred*), -isReviewer(*fred*) }

Module Mid

IF AddPaperReviewer(*u1*, *p*) AND isReviewer(*u1*)
AND isAuthor(*u2*, *p*) AND NOT Conflict(*u1*, *u2*) THEN
 isPaperReviewer(*u1*, *p*)

{ +isPaperReviewer(*bob*, *iliad*) }

Module Low

IF RemoveAdmin(*u*) THEN
 NOT isAdmin(*u*)



Extensions: Prioritization

Module High

IF ChangeJobToAdmin(*user*) AND isReviewer(*user*) THEN
 isAdmin(*user*)
 NOT isReviewer(*user*)

Module Mid

IF AddPaperReviewer(*u1*, *p*) AN
AND isAuthor(*u2*, *p*) AND NOT
 isPaperReviewer(*u1*, *p*)

Module Low

IF RemoveAdmin(*u*) THEN
 NOT isAdmin(*u*)

{ +Admin(*fred*), -Reviewer(*fred*)

+PaperReviewer(*bob*, *iliad*) }

Extensions: Error Reporting

```
{ +isAdmin(fred), -isReviewer(fred) }
```

```
{ +isPaperReviewer(bob, iliad) }
```

```
{ -isAdmin(fred) }
```

Extensions: Error Reporting

{ +isAdmin(*fred*), -isReviewer(*fred*) }

{ +isPaperReviewer(*bob*, *iliad*) }

{ -isAdmin(*fred*) }

| { +inconsistent(admin, fred) } |

Extensions: Error Reporting



```
{ +isPaperReviewer(bob, iliad) }
```



```
| { +inconsistent(admin, fred) } |
```

Extensions: Error Reporting

```
{ +isPaperReviewer(bob, iliad),
```

```
+inconsistent(admin, fred) }
```

Future Work

- Comparing Concurrency

Future Work

- Comparing Concurrency
- Predicating Composition

Don't Cross the Streams



Thanks!

Dan Dougherty

Kathi Fisler

The Brown University CS296-1 Class

Egon Boerger

Marianna Nicolosi-Asmundo

Wolfram Schulte

Wolfgang Schoenfeld

Yuri Gurevich

Don Batory