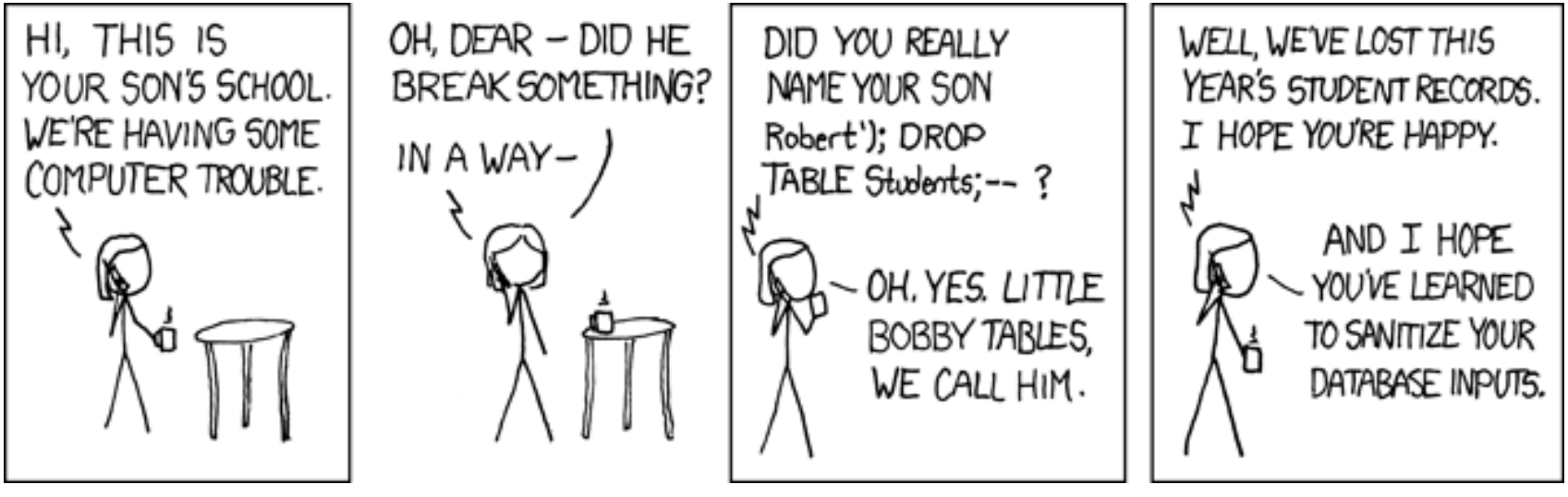
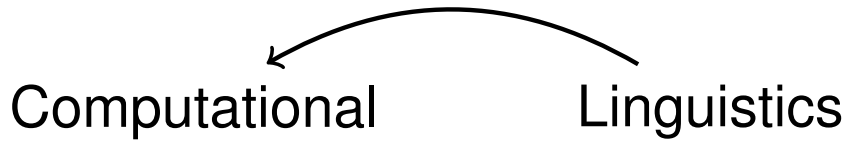
Quoting side effects

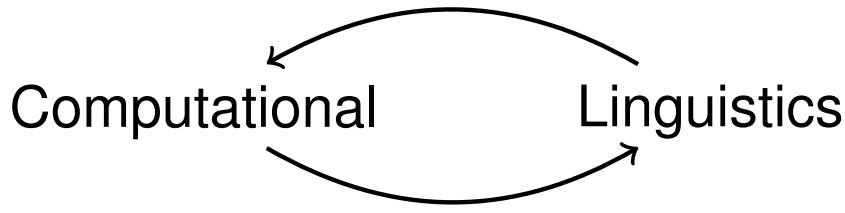
Chung-chieh Shan
Rutgers University
2007-10-13



http://xkcd.com/327/

Computational Linguistics


Computational
Linguistics


Computational
Linguistics

?

Outline

- **Natural vs programming languages**

Side effects
- State in programming languages
- Control in programming languages
- State in natural languages
- Control in natural languages

Quotation
- Code generation
- Mixed quotation

Together

How do natural languages work?

How should programming languages work?

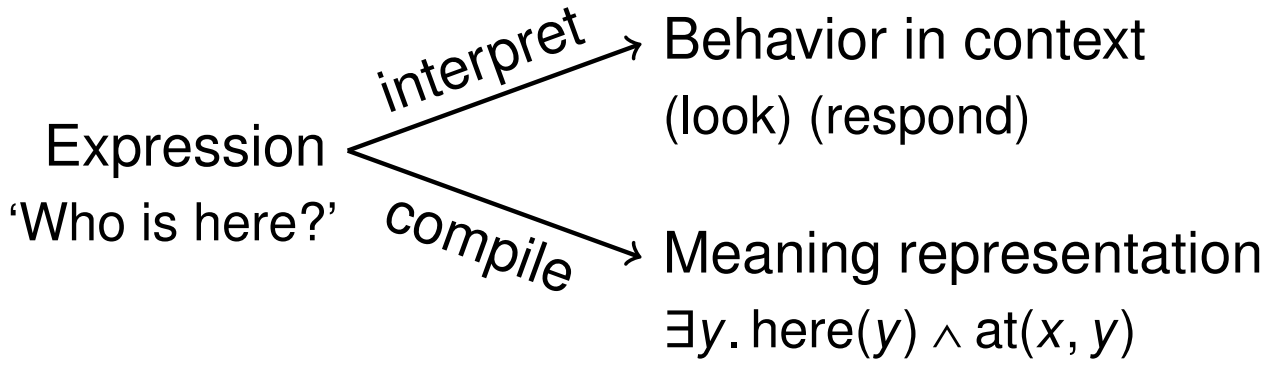
How do natural languages work?
How do people learn to speak?

How should programming languages work?
How should computers be designed?

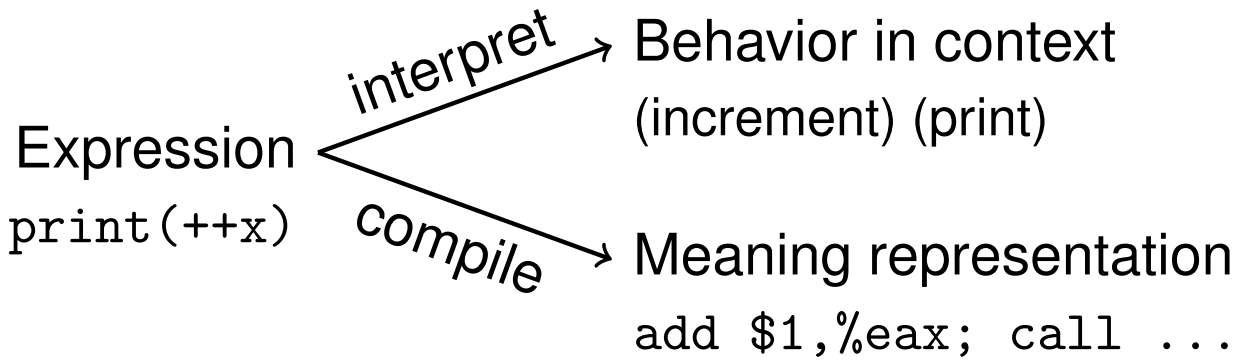
How do natural languages work?
How do people learn to speak?
How do people understand utterances?

How should programming languages work?
How should computers be designed?
How should computers run programs?

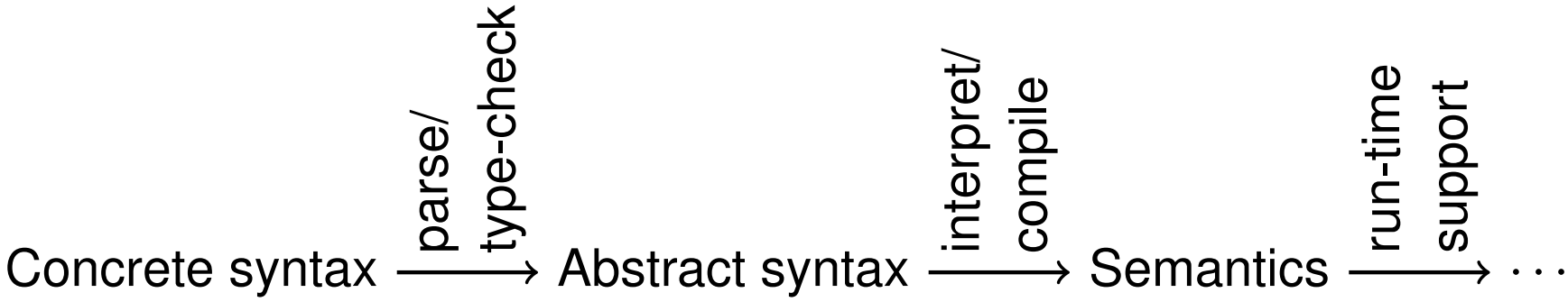
How do natural languages work?
How do people learn to speak?
How do people understand utterances?



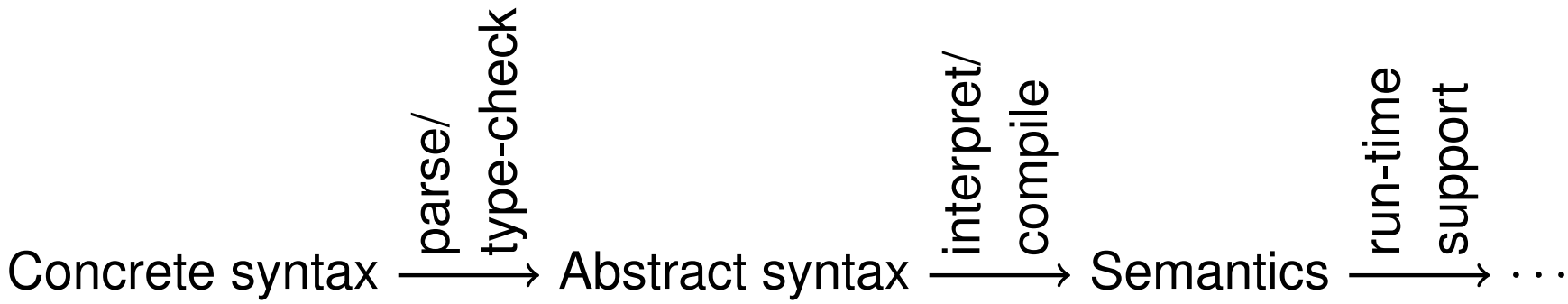
How should programming languages work?
How should computers be designed?
How should computers run programs?



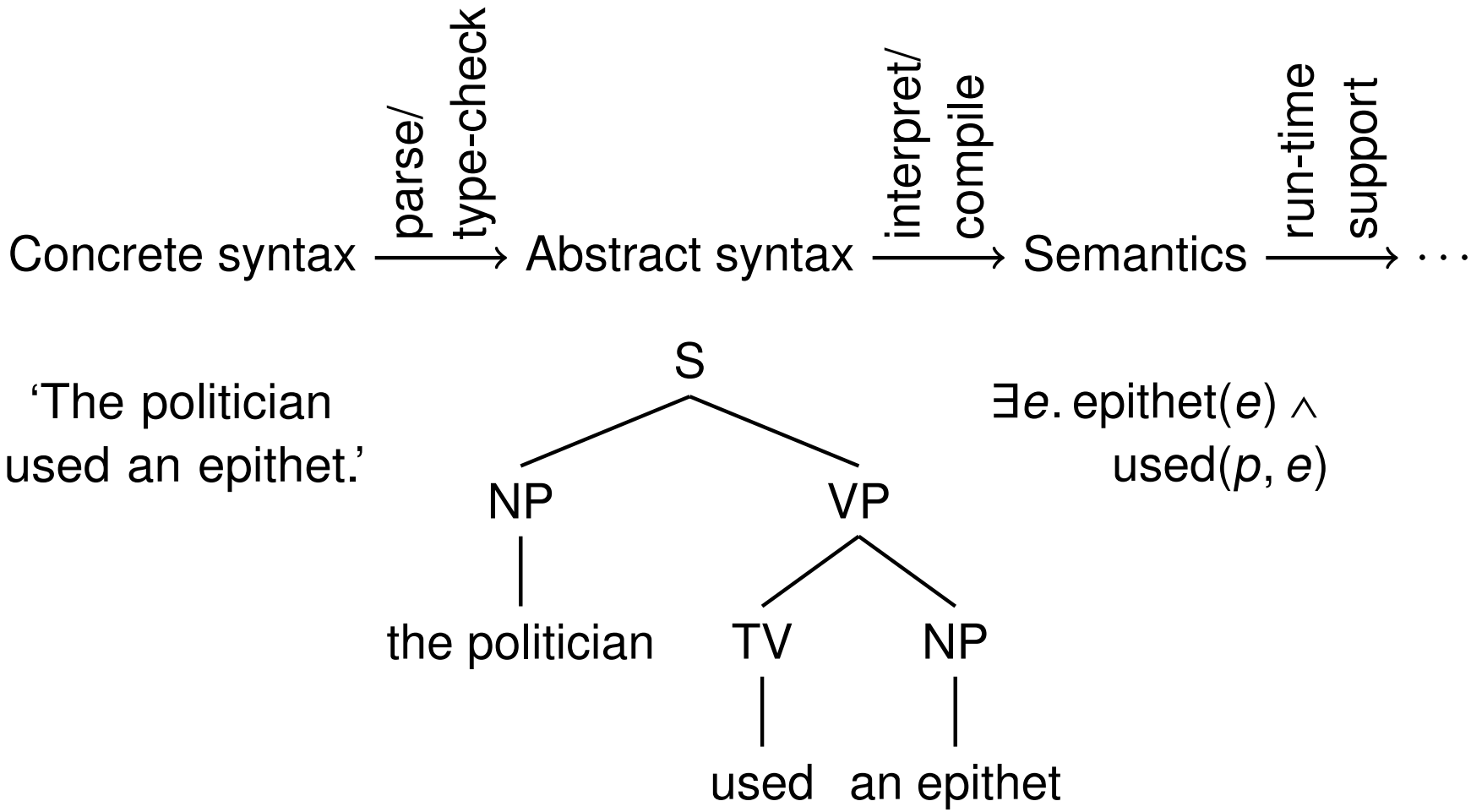
Fact checking



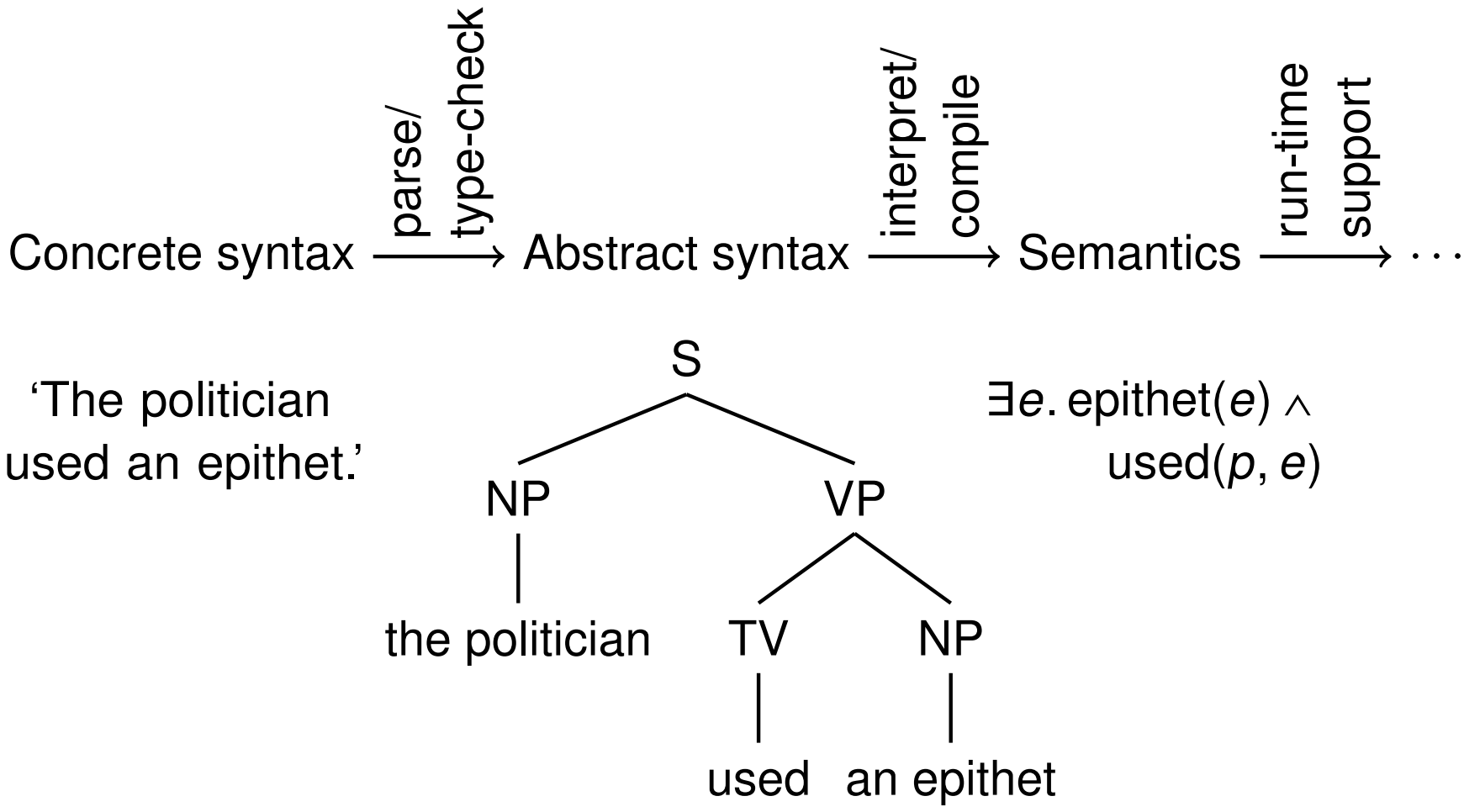
Fact checking



Fact checking



Fact checking



Challenges: language and the world are ambiguous and complex.
Never mind the query language—what is the database schema?
‘The journalist knows which politician used an epithet.’

Outline

Natural vs programming languages

- **Side effects**
  - State in programming languages
  - Control in programming languages
  - State in natural languages
  - Control in natural languages

Quotation
  - Code generation
  - Mixed quotation

Together

Unsound reasoning

A is better than B. B is better than C.

Therefore, A is better than C.

Unsound reasoning

The devil is better than nobody. Nobody is better than God.

Therefore, the devil is better than God.

Unsound reasoning

The devil is better than nobody. Nobody is better than God.

Therefore, the devil is better than God.

A is shorter than B. B is shorter than C.

Therefore, A is shorter than C.

Unsound reasoning

The devil is better than nobody. Nobody is better than God.

Therefore, the devil is better than God.

Alicia is shorter than her mom. Her mom is shorter than Alicia.

Therefore, Alicia is shorter than Alicia.

Unsound reasoning

The devil is better than nobody. Nobody is better than God.

Therefore, the devil is better than God.

Alicia is shorter than her mom. Her mom is shorter than Alicia.

Therefore, Alicia is shorter than Alicia.

$$\frac{A < B \qquad B < C}{A < C}$$

Unsound reasoning

The devil is better than nobody. Nobody is better than God.

Therefore, the devil is better than God.

Alicia is shorter than her mom. Her mom is shorter than Alicia.

Therefore, Alicia is shorter than Alicia.

```
x < --y    --y < x
-------------------
       x < x
```

Unsound reasoning

The devil is better than nobody. Nobody is better than God.

Therefore, the devil is better than God.

Alicia is shorter than her mom. Her mom is shorter than Alicia.

Therefore, Alicia is shorter than Alicia.

$$\frac{\texttt{x < --y} \qquad \texttt{--y < x}}{\texttt{x < x}}$$

Side effects make substitution unsound.

In natural language:

nobody her who anyone know the king of France ...

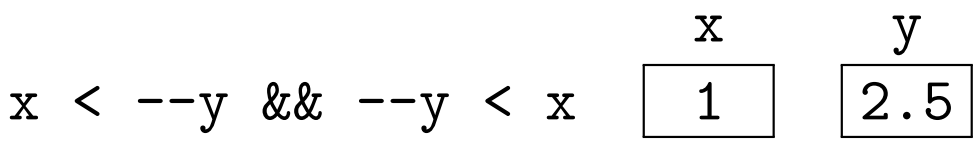
In programming languages:

`--` `throw` `print` `open` `amb` `...`

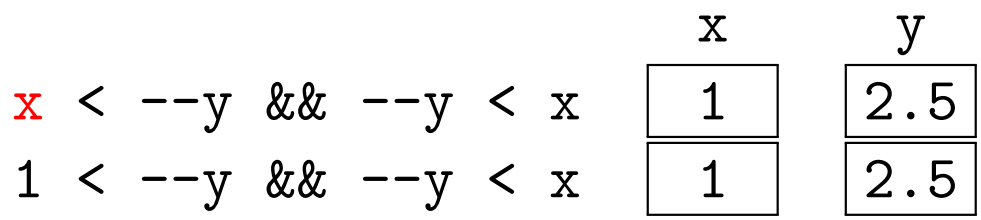
Operational semantics

```
x < --y && --y < x
```

Operational semantics



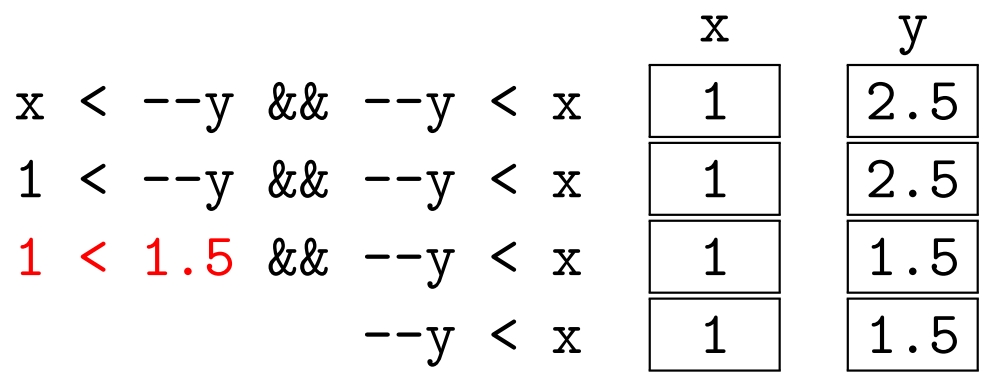
Operational semantics



Operational semantics

	x	y
`x < --y && --y < x`	1	2.5
`1 < --y && --y < x`	1	2.5
`1 < 1.5 && --y < x`	1	1.5

Operational semantics



Operational semantics

	x	y
`x < --y && --y < x`	1	2.5
`1 < --y && --y < x`	1	2.5
`1 < 1.5 && --y < x`	1	1.5
`--y < x`	1	1.5
`0.5 < x`	1	0.5

Operational semantics

	x	y
`x < --y && --y < x`	1	2.5
`1 < --y && --y < x`	1	2.5
`1 < 1.5 && --y < x`	1	1.5
`--y < x`	1	1.5
`0.5 < x`	1	0.5
`0.5 < 1`	1	0.5

Operational semantics

	x	y
`x < --y && --y < x`	1	2.5
`1 < --y && --y < x`	1	2.5
`1 < 1.5 && --y < x`	1	1.5
`--y < x`	1	1.5
`0.5 < x`	1	0.5
`0.5 < 1`	1	0.5
▶ `true`	1	0.5

Why is state useful?

Manipulating the context

```
every(4,6,9) % some(2,3) == 0
```

Manipulating the context

```
every(4,6,9) % some(2,3) == 0
```

```
[4 % some(2,3) == 0] &&
[6 % some(2,3) == 0] &&
[9 % some(2,3) == 0]
```

Manipulating the context

```
every(4,6,9) % some(2,3) == 0

[4 % some(2,3) == 0] &&
[6 % some(2,3) == 0] &&
[9 % some(2,3) == 0]

[4 % 2 == 0 || 4 % 3 == 0] &&
[6 % some(2,3) == 0] &&
[9 % some(2,3) == 0]
```

Manipulating the context

```
every(4,6,9) % some(2,3) == 0

[4 % some(2,3) == 0] &&
[6 % some(2,3) == 0] &&
[9 % some(2,3) == 0]

[4 % 2 == 0 || 4 % 3 == 0] &&
[6 % some(2,3) == 0] &&
[9 % some(2,3) == 0]

[6 % some(2,3) == 0] &&
[9 % some(2,3) == 0]
```

Manipulating the context

```
every(4,6,9) % some(2,3) == 0

[4 % some(2,3) == 0] &&
[6 % some(2,3) == 0] &&
[9 % some(2,3) == 0]

[4 % 2 == 0 || 4 % 3 == 0] &&
[6 % some(2,3) == 0] &&
[9 % some(2,3) == 0]

[6 % some(2,3) == 0] &&
[9 % some(2,3) == 0]

[6 % 2 == 0 || 6 % 3 == 0] &&
[9 % some(2,3) == 0]
```

Manipulating the context

```
every(4,6,9) % some(2,3) == 0

[4 % some(2,3) == 0] &&
[6 % some(2,3) == 0] &&
[9 % some(2,3) == 0]

[4 % 2 == 0 || 4 % 3 == 0] &&
[6 % some(2,3) == 0] &&
[9 % some(2,3) == 0]

[6 % some(2,3) == 0] &&
[9 % some(2,3) == 0]

[6 % 2 == 0 || 6 % 3 == 0] &&
[9 % some(2,3) == 0]

...
```

Backtracking search; back button

Anaphora

Alicia is shorter than her mom
and her mom is shorter than Alicia

Anaphora

Alicia is shorter than her mom
and her mom is shorter than Alicia

Alicia is shorter than her mom
and her mom is shorter than Alicia | Alicia

Anaphora

Alicia is shorter than her mom
and her mom is shorter than Alicia

Alicia is shorter than her mom
and her mom is shorter than Alicia | Alicia

Alicia is shorter than Alicia's mom
and her mom is shorter than Alicia | Alicia

Anaphora

Alicia is shorter than her mom
and her mom is shorter than Alicia

Alicia is shorter than her mom
and her mom is shorter than Alicia | Alicia |

Alicia is shorter than Alicia's mom
and her mom is shorter than Alicia | Alicia |

Alicia is shorter than Beatrice
and her mom is shorter than Alicia | Beatrice | Alicia |

Anaphora

Alicia is shorter than her mom
and her mom is shorter than Alicia

Alicia is shorter than her mom
and her mom is shorter than Alicia | Alicia

Alicia is shorter than Alicia's mom
and her mom is shorter than Alicia | Alicia

Alicia is shorter than Beatrice
and her mom is shorter than Alicia | Beatrice | Alicia

her mom is shorter than Alicia | Beatrice | Alicia

Anaphora

Alicia is shorter than her mom and her mom is shorter than Alicia	
Alicia is shorter than her mom and her mom is shorter than Alicia	Alicia
Alicia is shorter than Alicia's mom and her mom is shorter than Alicia	Alicia
Alicia is shorter than Beatrice and her mom is shorter than Alicia	Beatrice, Alicia
her mom is shorter than Alicia	Beatrice, Alicia
Beatrice's mom is shorter than Alicia	Beatrice, Alicia

Anaphora

Alicia is shorter than her mom and her mom is shorter than Alicia	
Alicia is shorter than her mom and her mom is shorter than Alicia	Alicia
Alicia is shorter than Alicia's mom and her mom is shorter than Alicia	Alicia
Alicia is shorter than Beatrice and her mom is shorter than Alicia	Beatrice Alicia
her mom is shorter than Alicia	Beatrice Alicia
Beatrice's mom is shorter than Alicia	Beatrice Alicia
Clara is shorter than Alicia	Clara Beatrice Alicia

Anaphora

Alicia is shorter than her mom and her mom is shorter than Alicia	
Alicia is shorter than her mom and her mom is shorter than Alicia	Alicia
Alicia is shorter than Alicia's mom and her mom is shorter than Alicia	Alicia
Alicia is shorter than Beatrice and her mom is shorter than Alicia	Beatrice Alicia
her mom is shorter than Alicia	Beatrice Alicia
Beatrice's mom is shorter than Alicia	Beatrice Alicia
Clara is shorter than Alicia	Clara Beatrice Alicia
▶ true	Clara Beatrice Alicia

More sophisticated theory of discourse referents

Quantification

the devil is better than nobody

Quantification

the devil is better than nobody

the devil is not better than Alicia
and the devil is not better than Beatrice
and the devil is not better than Clara

Quantification

the devil is better than nobody

the devil is not better than Alicia
and the devil is not better than Beatrice
and the devil is not better than Clara

the devil is not better than Beatrice
and the devil is not better than Clara

Quantification

the devil is better than nobody

the devil is not better than Alicia
and the devil is not better than Beatrice
and the devil is not better than Clara

the devil is not better than Beatrice
and the devil is not better than Clara

the devil is not better than Clara

Quantification

the devil is better than nobody

the devil is not better than Alicia
and the devil is not better than Beatrice
and the devil is not better than Clara

the devil is not better than Beatrice
and the devil is not better than Clara

the devil is not better than Clara

- true

In-situ quantifiers

Outline

Natural vs programming languages

Side effects

- State in programming languages
- Control in programming languages
- State in natural languages
- Control in natural languages

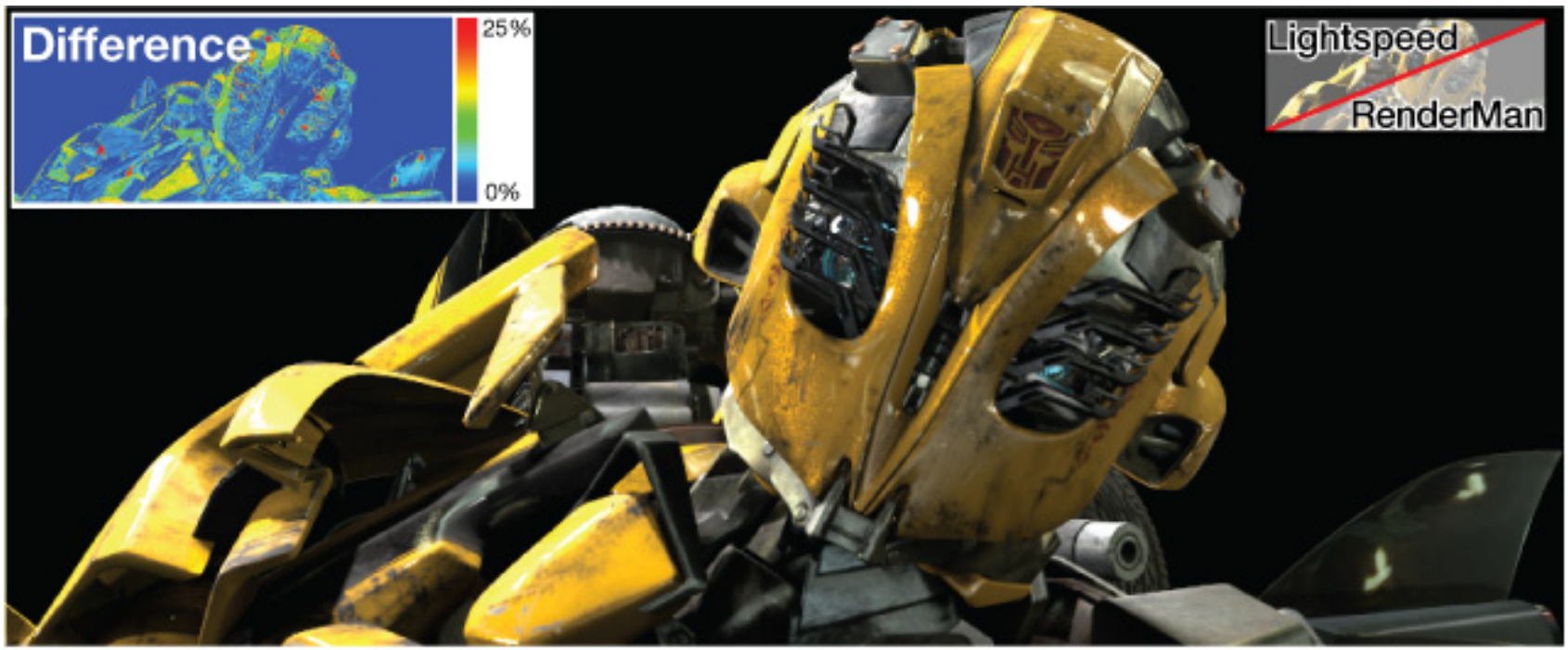
▶ **Quotation**

- Code generation
- Mixed quotation

Together

A recurring situation

render (scene, lighting)



Ragan-Kelley et al., SIGGRAPH 2007

A recurring situation

Program	Static input	Dynamic input
render	(scene,	lighting)
power	(exponent,	base)
parse	(grammar,	string)
invert	(size,	matrix)
compile	(headers,	source)
	⋮	

A recurring situation

Specialized program	Dynamic input
render_robot	(lighting)
power_12	(base)
parse_java	(string)
invert_16	(matrix)
compile_gui	(source)
⋮	

A recurring situation

Cogen	Static input	Dynamic input
renderGen	(scene)	(lighting)
powerGen	(exponent)	(base)
parseGen	(grammar)	(string)
invertGen	(size)	(matrix)
compileGen	(headers)	(source)
	⋮	

Quotation helps write cogens

```
power (0, x) = 1
power (n, x) = x * power (n-1, x)

power 12 2
```

▶ 4096

Quotation helps write cogens

```
power (0, x) = 1
power (n, x) = x * power (n-1, x)

power 12 2
▶ 4096


powerGen (0, x) = ⟨1⟩
powerGen (n, x) = ⟨~x * ~(powerGen (n-1, x))⟩
```

Quotation helps write cogens

```
power (0, x) = 1
power (n, x) = x * power (n-1, x)

power 12 2
▶ 4096
```

```
powerGen (0, x) = ⟨1⟩
powerGen (n, x) = ⟨~x * ~(powerGen (n-1, x))⟩

⟨fun x -> ~(powerGen (12, ⟨x⟩))⟩
▶ ⟨fun x -> x*x*x*x*x*x*x*x*x*x*x*x*1⟩
```

Mixed quotation

I am sorry to have used an ‘epithet’.

Mixed quotation

I am sorry to have used an ‘epithet’.

Quine said that quotation ‘has a certain anomalous feature’.
(Davidson 1979)

Bush also said his administration would ‘achieve our objectives’ in Iraq. (*New York Times*, 2004-11-04)

Outline

Natural vs programming languages

Side effects
- State in programming languages
- Control in programming languages
- State in natural languages
- Control in natural languages

Quotation
- Code generation
- Mixed quotation

- **Together**

Code generation with side effects

State for counting operations

```
count = 0

powerGen (0, x) = ⟨1⟩
powerGen (n, x) = ++count; ⟨~x * ~(powerGen (n-1, x))⟩
```

Code generation with side effects

State for counting operations

```
count = 0

powerGen (0, x) = ⟨1⟩
powerGen (n, x) = ++count; ⟨~x * ~(powerGen (n-1, x))⟩
```

Control for generating variable bindings

```
⟨fun x -> ~(powerGen (4, powerGen (3, ⟨x⟩)))⟩
```

▶ `⟨fun x -> (x*x*x*1)*(x*x*x*1)*(x*x*x*1)*(x*x*x*1)*1⟩`

Code generation with side effects

State for counting operations

```
count = 0

powerGen (0, x) = ⟨1⟩
powerGen (n, x) = ++count; ⟨~x * ~(powerGen (n-1, x))⟩
```

Control for generating variable bindings

```
⟨fun x -> ~(powerGen (4, powerGen (3, ⟨x⟩)))⟩
⟨fun x -> let y = x*x*x*1 in ~(powerGen (4, ⟨y⟩))⟩
```

Code generation with side effects

State for counting operations

```
count = 0

powerGen (0, x) = ⟨1⟩
powerGen (n, x) = ++count; ⟨~x * ~(powerGen (n-1, x))⟩
```

Control for generating variable bindings

```
⟨fun x -> ~(powerGen (4, powerGen (3, ⟨x⟩)))⟩
⟨fun x -> let y = x*x*x*1 in ~(powerGen (4, ⟨y⟩))⟩
⟨fun x -> let y = x*x*x*1 in let z = y*y*y*y*1 in ~⟨z⟩⟩
```

Code generation with side effects

State for counting operations

```
count = 0

powerGen (0, x) = ⟨1⟩
powerGen (n, x) = ++count; ⟨~x * ~(powerGen (n-1, x))⟩
```

Control for generating variable bindings

```
⟨fun x -> ~(powerGen (4, powerGen (3, ⟨x⟩)))⟩
⟨fun x -> let y = x*x*x*1 in ~(powerGen (4, ⟨y⟩))⟩
⟨fun x -> let y = x*x*x*1 in let z = y*y*y*y*1 in ~⟨z⟩⟩
▶ ⟨fun x -> let y = x*x*x*1 in let z = y*y*y*y*1 in z⟩
```

Code generation with side effects

State for counting operations

```
count = 0

powerGen (0, x) = ⟨1⟩
powerGen (n, x) = ++count; ⟨~x * ~(powerGen (n-1, x))⟩
```

Control for generating variable bindings

```
⟨fun x -> ~(powerGen (4, powerGen (3, ⟨x⟩)))⟩
⟨fun x -> let y = x*x*x*1 in ~(powerGen (4, ⟨y⟩))⟩
⟨fun x -> let y = x*x*x*1 in let z = y*y*y*y*1 in ~⟨z⟩⟩
▶ ⟨fun x -> let y = x*x*x*1 in let z = y*y*y*y*1 in z⟩
```

Also, generating code with side effects

Mixed quotation with side effects

Anaphora and mixed quotation

Bush also said his administration would ‘achieve our objectives’ in Iraq, but what are they precisely?

Mixed quotation with side effects

Anaphora and mixed quotation

Bush also said his administration would ‘achieve our objectives’ in Iraq, but what are they precisely?

Quantification and mixed quotation

Someone is shorter than everyone. (scope ambiguity)

Mixed quotation with side effects

Anaphora and mixed quotation

Bush also said his administration would ‘achieve our objectives’ in Iraq, but what are they precisely?

Quantification and mixed quotation

Someone is shorter than everyone. (scope ambiguity)
‘Someone is shorter than [everyone].’ (code generation)

Mixed quotation with side effects

Anaphora and mixed quotation

Bush also said his administration would ‘achieve our objectives’ in Iraq, but what are they precisely?

Quantification and mixed quotation

Someone is shorter than everyone. (scope ambiguity)
‘Someone is shorter than [everyone].’ (code generation)

Nobody is shorter than anybody. (polarity sensitivity)

Mixed quotation with side effects

Anaphora and mixed quotation

Bush also said his administration would ‘achieve our objectives’ in Iraq, but what are they precisely?

Quantification and mixed quotation

Someone is shorter than everyone. (scope ambiguity)
‘Someone is shorter than [everyone].’ (code generation)

Nobody is shorter than anybody. (polarity sensitivity)

× Anybody is shorter than nobody. (no inverse scope)

Mixed quotation with side effects

Anaphora and mixed quotation

Bush also said his administration would ‘achieve our objectives’ in Iraq, but what are they precisely?

Quantification and mixed quotation

Someone is shorter than everyone.	(scope ambiguity)
‘Someone is shorter than [everyone].’	(code generation)
Nobody is shorter than anybody.	(polarity sensitivity)
× Anybody is shorter than nobody.	(no inverse scope)
× ‘Anybody is shorter than [nobody].’	(quotation failure)

Conclusion

Natural languages Programming languages

Conclusion

	Natural languages	Programming languages
Side effects • State • Control		
Quotation		

Conclusion

	Natural languages	Programming languages
Side effects		
• State	her	`--y`
• Control	nobody	`some(2,3)`
Quotation	use an ‘epithet’	⟨`fun x -> x*x*x*1`⟩

Conclusion

	Natural languages		Programming languages
Side effects			
• State	her	$\longleftrightarrow$	`--y`
• Control	nobody		`some(2,3)`
Quotation	use an ‘epithet’	$\longleftrightarrow$	⟨`fun x -> x*x*x*1`⟩

Conclusion

	Natural languages	Programming languages
Side effects • State • Control	her nobody	`--y` `some(2,3)`
Quotation	use an ‘epithet’	⟨`fun x -> x*x*x*1`⟩