Fix the First, Ignore the Rest: Dealing with Multiple Compiler Error Messages

BRETT A BECKER*, CORMAC MURRAY — UNIVERSITY COLLEGE DUBLIN

TIANYI TAO, CHANGHENG SONG — FUDAN UNIVERSITY

ROBERT MCCARTNEY — UNIVERSITY OF CONNECTICUT

KATE SANDERS — RHODE ISLAND COLLEGE

*BRETT DOT BECKER AT UCD DOT IE

Background

- Compiler error messages pose substantial problems for novices (both technical and affective)

- In order to help students learning to program, many computing education researchers have analyzed the compiler error messages generated by them

Background

- Almost all prior studies look at **all** error messages generated by students (a single failed compile often generates multiple messages)
  - Some of these messages are *spurious* – a result of compiler confusion, not a genuine error committed by a student
- Few studies have specifically investigated **first** error messages in isolation
  - First messages are important
    - some tools only show the first (BlueJ, etc.)
    - some teachers instruct students to ignore all but the first
    - some students only consider the first either way

Motivation

```java
public class PrintNums {
    public static void main (String[] args) {
        for(int x = 0: x < 10; x++)
            System.out.println(x);
    }
}
```

- One syntax error: regular colon instead of semicolon

Motivation

- Output from above code

  - Only first error message represents actual error

  - The rest are spurious – they are resolved when the original error is resolved

```
PrintNums.java:3: error: ';' expected
for(int x = 0: x < 10; x++)
                ^
PrintNums.java:3: error: ';' expected
for(int x = 0: x < 10; x++)
                   ^
PrintNums.java:3: error: illegal start of type
for(int x = 0: x < 10; x++)
                        ^
PrintNums.java:3: error: illegal start of expression
for(int x = 0: x < 10; x++)
                           ^
PrintNums.java:3: error: ')' expected
for(int x = 0: x < 10; x++)
                            ^
PrintNums.java:3: error: illegal start of expression
for(int x = 0: x < 10; x++)
                             ^
6 errors
```
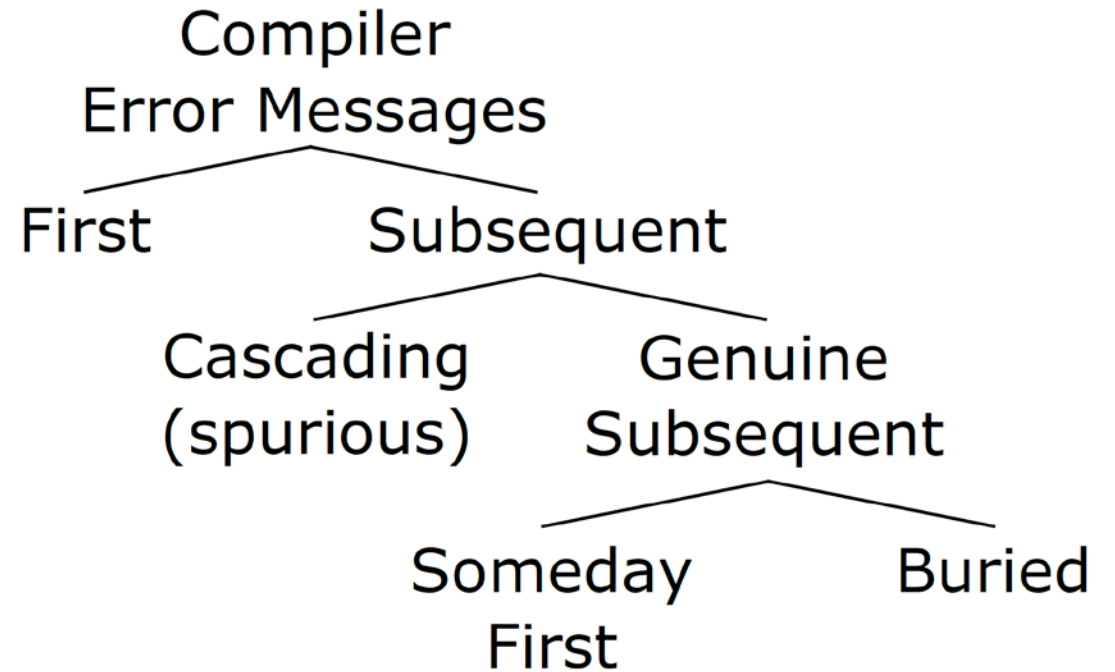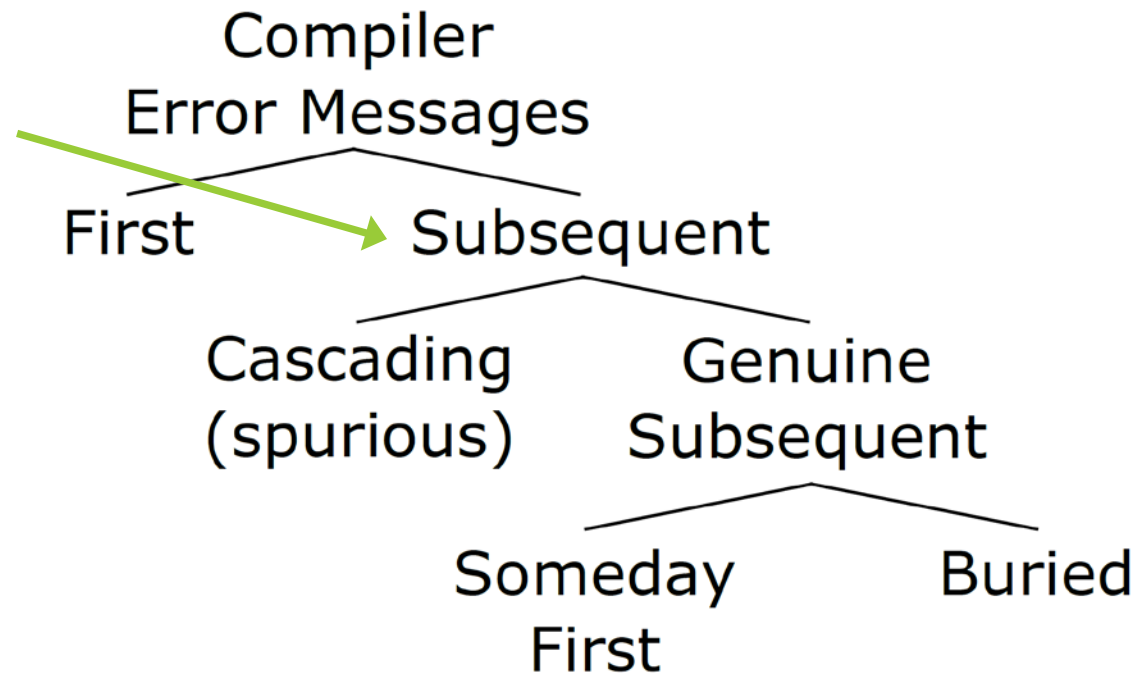
A Compiler Error Message Taxonomy

- We quickly realized that talking about compiler error messages was problematic
  - We could find no complete, consistent taxonomy to describe them precisely

- So, we devised our own

Compiler
Error Messages

First   Subsequent

Cascading   Genuine
(spurious)   Subsequent

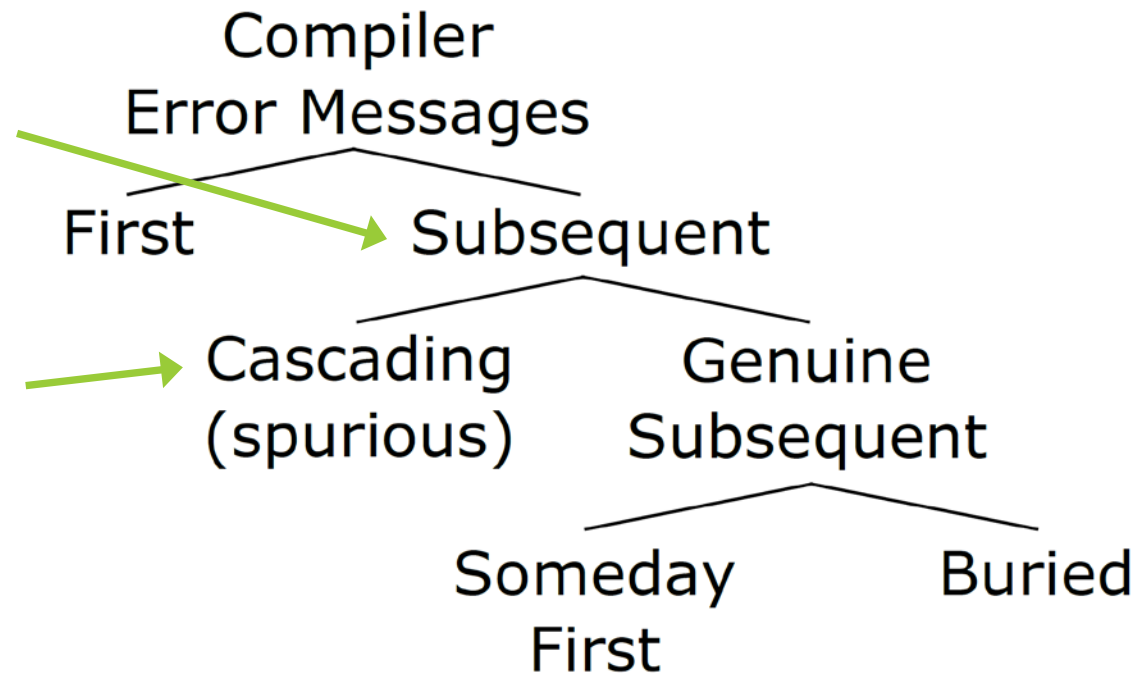Someday   Buried
First

A Compiler Error Message Taxonomy

- All messages after first (it is possible there is only a first)

A Compiler Error Message Taxonomy

- All messages after first (it is possible there is only a first)

- Not due to genuine errors in code

Compiler Error Messages

First   Subsequent

Cascading (spurious)   Genuine Subsequent

Someday First   Buried

A Compiler Error Message Taxonomy

- All messages after first (it is possible there is only a first)

- Not due to genuine errors in code

Compiler Error Messages

First — Subsequent

Cascading (spurious) — Genuine Subsequent

Someday First — Buried

- Due to genuine error(s) in code

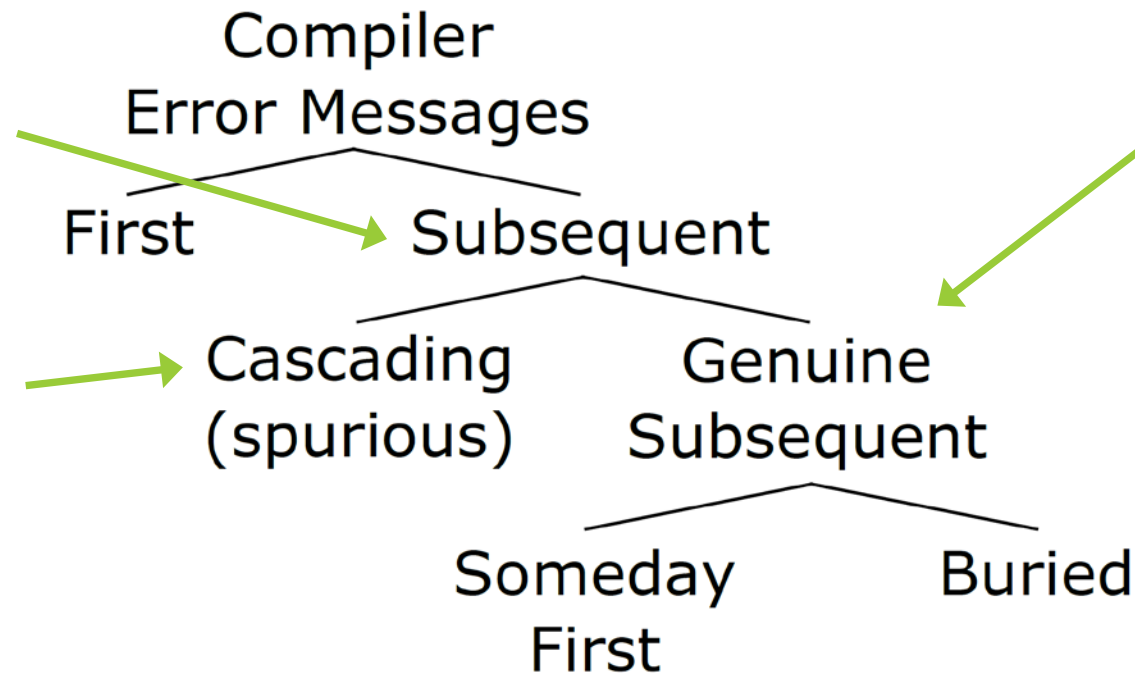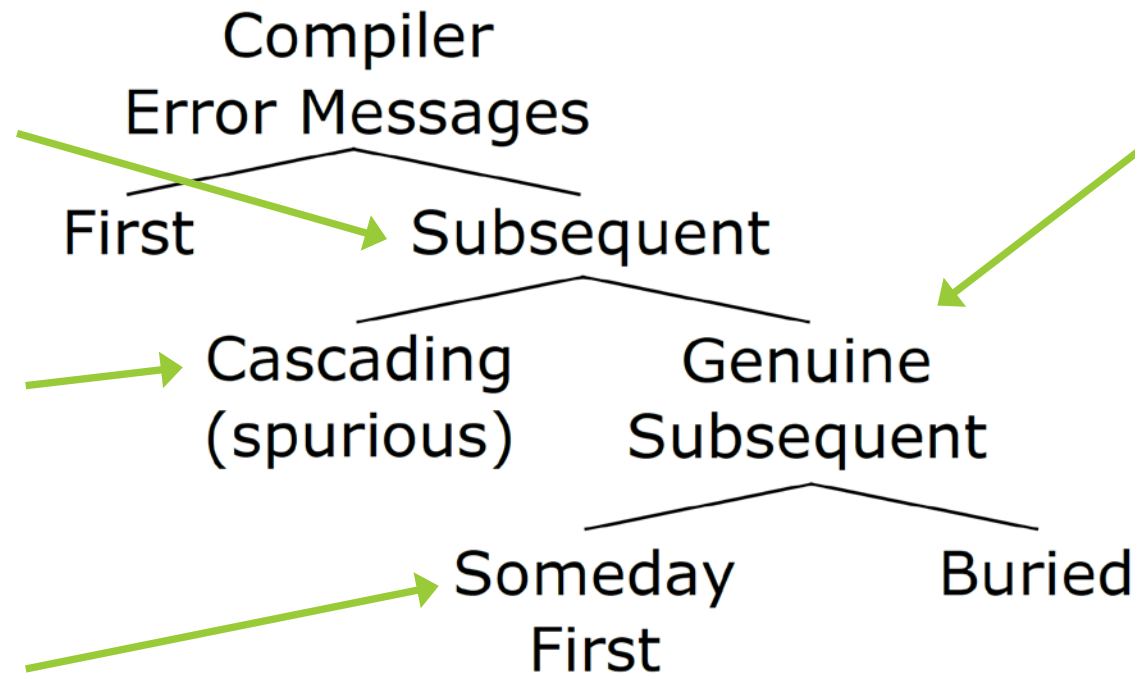# A Compiler Error Message Taxonomy

- All messages after first (it is possible there is only a first)

- Not due to genuine errors in code

- Will eventually be a first error

- Due to genuine error(s) in code

Compiler Error Messages

First   Subsequent

Cascading (spurious)   Genuine Subsequent
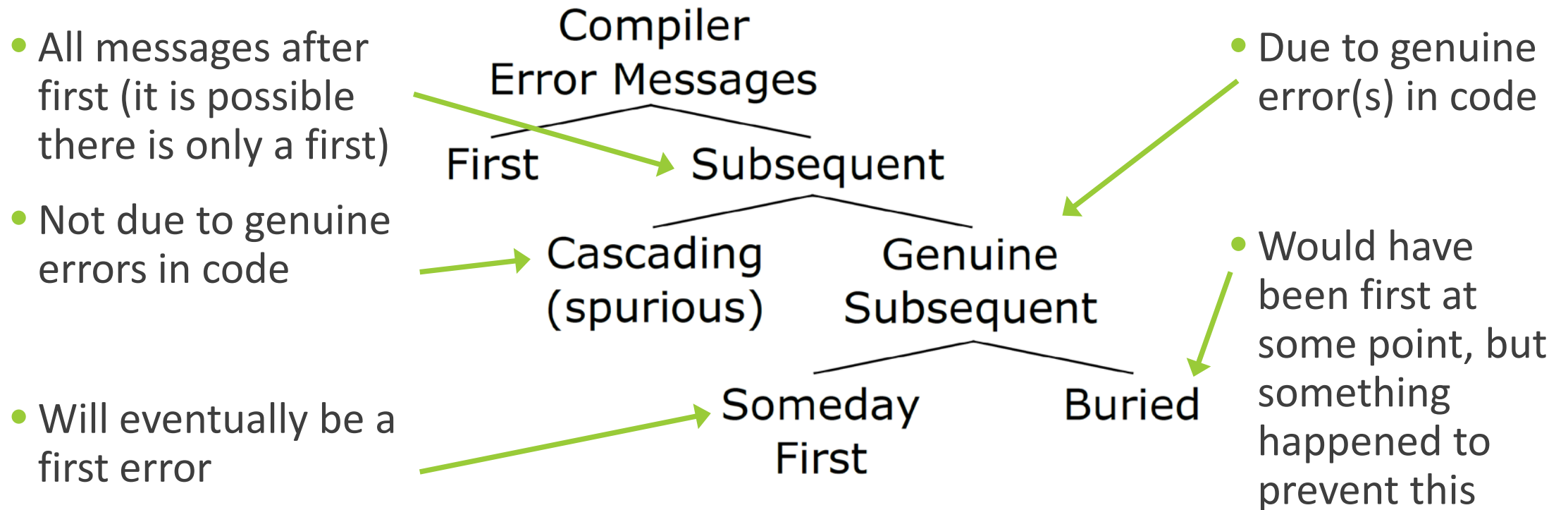
Someday First   Buried

A Compiler Error Message Taxonomy

- All messages after first (it is possible there is only a first)

- Not due to genuine errors in code

- Will eventually be a first error

Compiler Error Messages
First — Subsequent
Cascading (spurious) — Genuine Subsequent
Someday First — Buried

- Due to genuine error(s) in code

- Would have been first at some point, but something happened to prevent this

Student Perspectives

- In 1984 Du Boulay and Matthew noted: *novice programmers find that their initial programs are rejected by the compiler in a flurry of incomprehensible error messages*

- We conducted a student survey indicating that this is still the case:
  - 106 Software Engineering students in second semester of CS1
  - C / Notepad++ / gcc @ command line
  - We sought to explore the **frustration, confusion** and **misconception** surrounding compiler error messages (particularly subsequent)

Student Perspectives

- **Frustration & confusion**
  - 85% of students reported compiler error messages to be frustrating
  - 92% reported compiler error messages to be a barrier to progress
  - 64% reported subsequent compiler error messages to be confusing

Student Perspectives

- **Misconceptions**
  - 67% believe the first error message may not represent a true error
  - 21% believe that subsequent error messages always represent true errors
    - 11% believe that these never represent true errors
  - When confronted with subsequent error messages:
    - 43% try to fix one error (only 22% focus on the first only)
    - 34% try to fix more than one error simultaneously

Student Perspectives

- **Requests for help**

  - 100% reported that having additional help specifically on dealing with compiler error messages would help them in learning to program

  - 99% reported that having the details of subsequent error messages better explained to them would help them deal with subsequent error messages more effectively

Research Questions

1. Out of all compiler error messages, what proportion are first messages?

When first error messages are considered separately:

2. How does the most-frequent list of messages change?

3. How do the frequencies of specific error messages change?

Methodology

- 21 million BlueJ (Java) error messages from the Blackbox dataset
  - Six months (US users only)
- Analyzed and compared:
  - All first error messages
  - All subsequent error messages
  - All error messages

RQ1

Out of all compiler error messages, what proportion are first messages?

- Out of the 21 million error messages in our dataset, 28.5% were first messages

- The average failed compile generated 3.50 error messages, 2.50 of which were subsequent
 - This is in line with prior studies, but data is hard to come by

RQ2

When first error messages are considered separately, how does the most-frequent list of messages change?

Error Message Description	R_F	R_S	R_A
cannot find symbol - variable	1	1	1
cannot find symbol - method	2	3	3
';' expected	3	2	2
incompatible types	4	9	7
cannot find symbol - class	5	5	4
')' expected	6	12	11
illegal start of expression	7	6	6
missing return statement	8	22	14
<identifier> expected	9	7	8
not a statement	10	8	9

- Top 10 (top 67%) error messages only presented here – we looked at top 23 (top 85%)

- R_F is rank (amongst all first messages), R_S is rank amongst all subsequent, R_A is rank amongst all

- Shifts of up to 19 ranks observed (this message has a rank in one list that is 19 positions from its rank in another list)

RQ3

When first error messages are considered separately, how do the frequencies of specific error messages change?
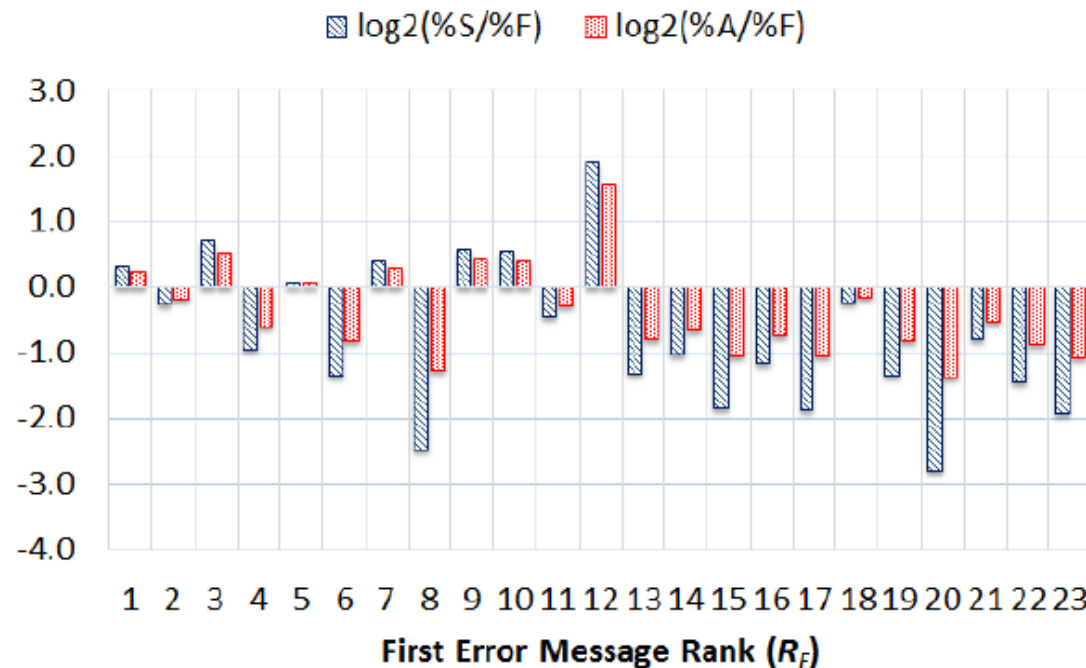
- Top 23 error messages making up 85% of all
- Some error messages are 7 times more likely to be first than subsequent
- Some are ~3 times more likely to be subsequent than first

Discussion

- In languages such as Java, only the first error message is guaranteed to be due to a real error in code

- Some novice editors only show the first error message to students, and some teachers teach students to ignore all subsequent messages

- Most research has focused on ALL error messages

- We have shown that when first messages are compared to subsequent, and compared to all error messages:
  - Some messages change rank in non-trivial ways
  - The chance of seeing particular errors changes in non-trivial ways

Discussion

- These changes need to be accounted for when teaching or researching novices

  - Our most-frequent first messages include all with at least 1% frequency

  - Students that only see (or only attempt to fix) first messages would see each of these messages between 1 and 19 times in an average session*

    - These students **would see** nine different messages per session that **do not** feature in most lists of frequent error messages reported by the literature (that are based on overall frequencies)

    - They **would not see** three messages that **do** feature in most lists

*given 100 failed compiles in a programming session, which is realistic based on our data

Thank you!

Thanks to Neil Brown and the BlueJ / Blackbox teams for providing access to their data

Questions?