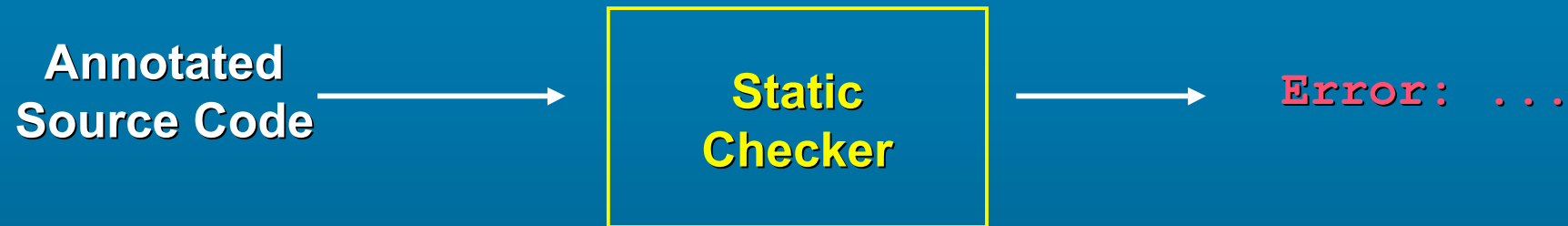


Extended Static Checking for Java

Cormac Flanagan

**Joint work with: Rustan Leino,
Mark Lillibridge, Greg Nelson,
Jim Saxe, and Raymie Stata**

What is “Static Checking”?



- type systems

`Error: wrong number of arguments in method call`

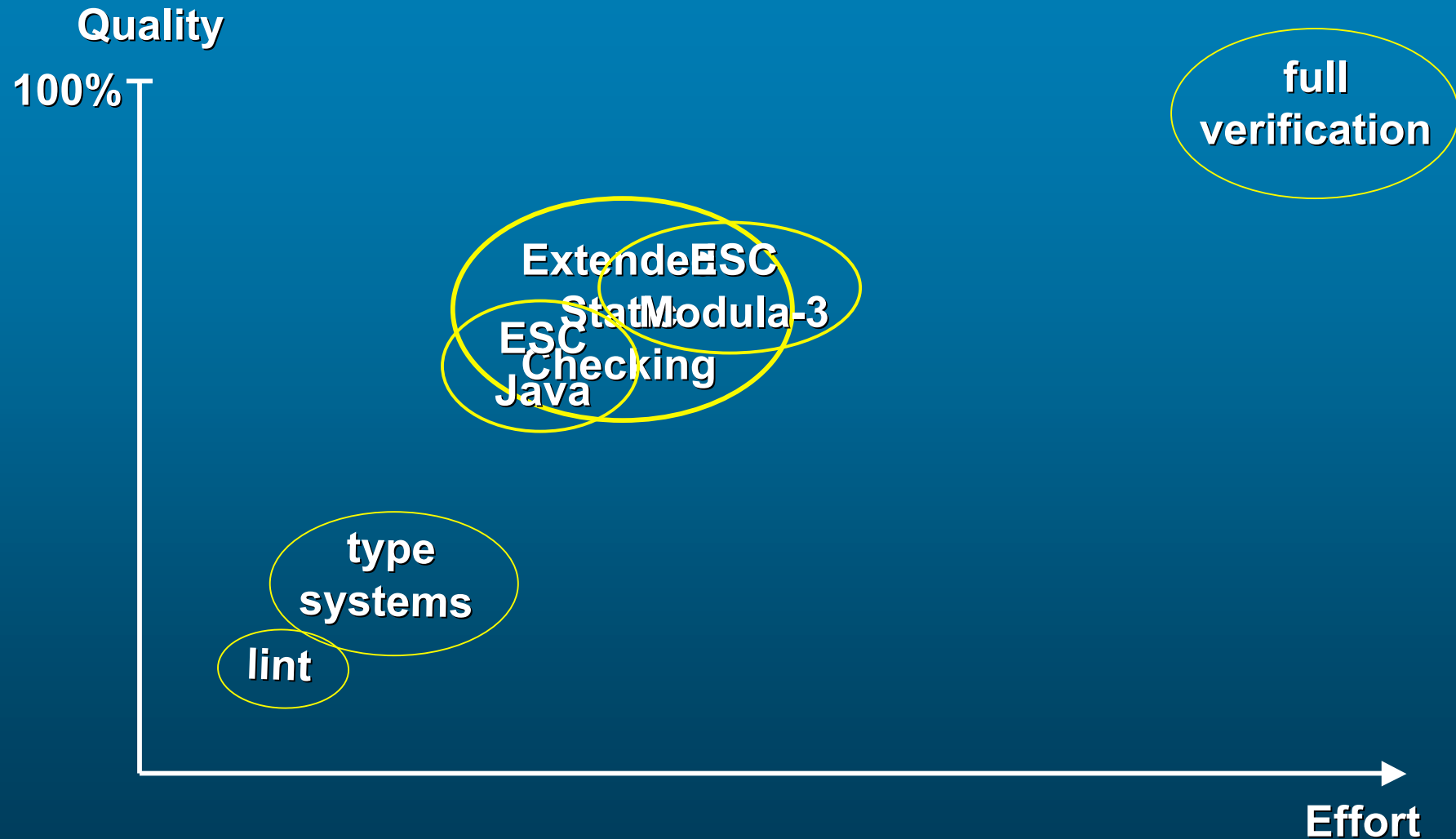
- lint

`Error: unreachable code`

- full program verification

`Error: qsort does not yield a sorted array`

Comparison of Static Checkers



Note: Graph is not to scale

ESC/Java

□ Detect common run-time errors

- null dereferences
- array bounds
- type casts
- race conditions
- deadlocks
- ...

□ Check other properties

- object invariants
- method specifications
- ...

Architecture of ESC/Java

Annotated program



Verification condition generator



Verification condition



***Automatic*
theorem prover**



Counter-example

$\forall x. \forall y. (x > y \Rightarrow \dots)$

Error: index out of bounds on line 218 when $x = 4$ and $y = 3$

Not Full Verification

- Prove absence of run-time errors,
not full functional correctness
 - Simpler method specifications
 - Verification conditions easier to prove

Example

```
void sort2(int[] a)
  //@ requires a != null && a.length == 2
  //@ ensures a[0] <= a[1]
{
  if( a[0] > a[1] )
  {
    int t = a[0];
    a[0] = a[1];
    a[1] = t;
  }
}
```

No more errors detected (found 1st at line 8)

Under the Hood

□ Verification condition large but “dumb”

```
(FORALL (t1) (FORALL (t2) (IMPLIES (AND (NEQ a null) (EQ (arrayLength a) 2))
(AND (NEQ a null) (AND (AND (<= 0 0) (< 0 (arrayLength a))) (AND (NEQ a null)
(AND (AND (<= 0 1) (< 1 (arrayLength a))) (AND (IMPLIES (> (select (select
elem a) 0) (select (select elem a) 1)) (FORALL (t3) (AND (NEQ a null) (AND
(AND (<= 0 0) (< 0 (arrayLength a))) (FORALL (t) (IMPLIES (EQ t (select
(select elem a) 0)) (FORALL (t1) (IMPLIES (EQ t1 a) (AND (NEQ a null) (AND
(AND (<= 0 1) (< 1 (arrayLength a))) (AND (NEQ t1 null) (AND (AND (<= 0 0) (<
0 (arrayLength t1))) (FORALL (t2) (IMPLIES (EQ t2 a) (AND (NEQ t2 null) (AND
(AND (<= 0 1) (< 1 (arrayLength t2))) (AND (<= (select (select (store (store
elem t1 (store (select elem t1) 0 (select (select elem a) 1))) t2 (store
(select (select elem t1 (store (select elem t1) 0 (select (select elem a) 1)))
t2) 1 t)) a) 0) (select (select (store (store elem t1 (store (select elem t1)
0 (select (select elem a) 1))) t2 (store (select (store elem t1 (store
(select elem t1 0 (select (select elem a) 1))) t2 1 t)) a) 1)) (EQ true
true)))))))))))))) (IMPLIES (NOT (> (select (select elem a) 0) (select
(select elem a) 1))) (AND (<= (select (select elem a) 0) (select (select elem
a) 1)) (EQ true true))))))))))
```

□ Proved in < 1 second

ESC/Java vs. Testing

- Testing essential but

- Expensive
- Finds errors late
- Misses errors

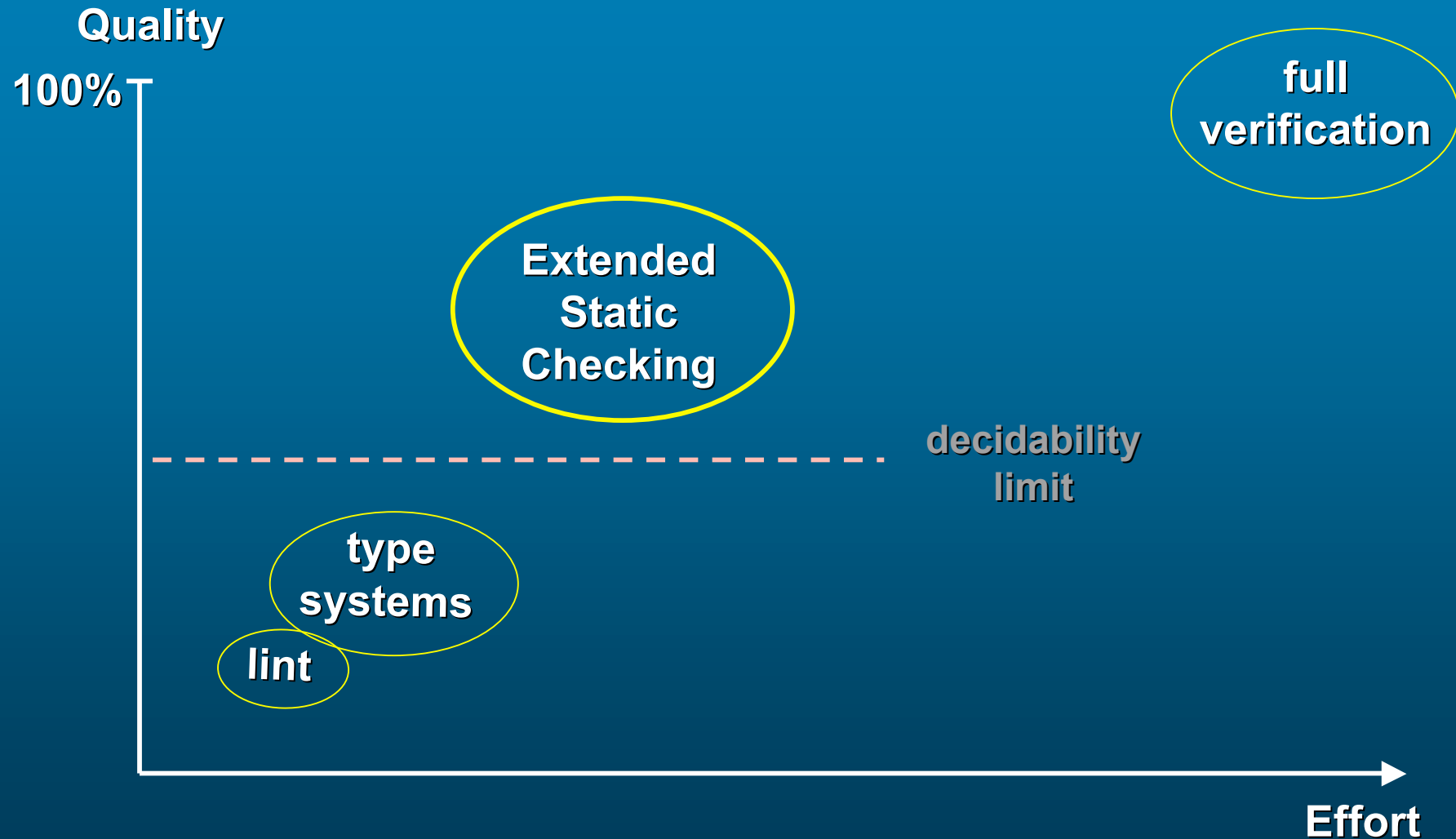
- ESC/Java ... ?

ESC/Java Summary

- ❑ Finds more errors than type checking
- ❑ Costs less than full verification
- ❑ Currently working on small test programs

www.research.digital.com/SRC/esc/Esc.html

Comparison of Static Checkers



Note: Graph is not to scale

Metrics for Static Checkers

- ❑ **Cost**

 - of using the tool

- ❑ **Quality**

 - Does it miss errors?

 - Does it give spurious warnings?