Konzepte objektorientierter Programmierung
– Lecture 12 –

Prof. Dr. Peter Müller

Software Component Technology

Wintersemester 06/07

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

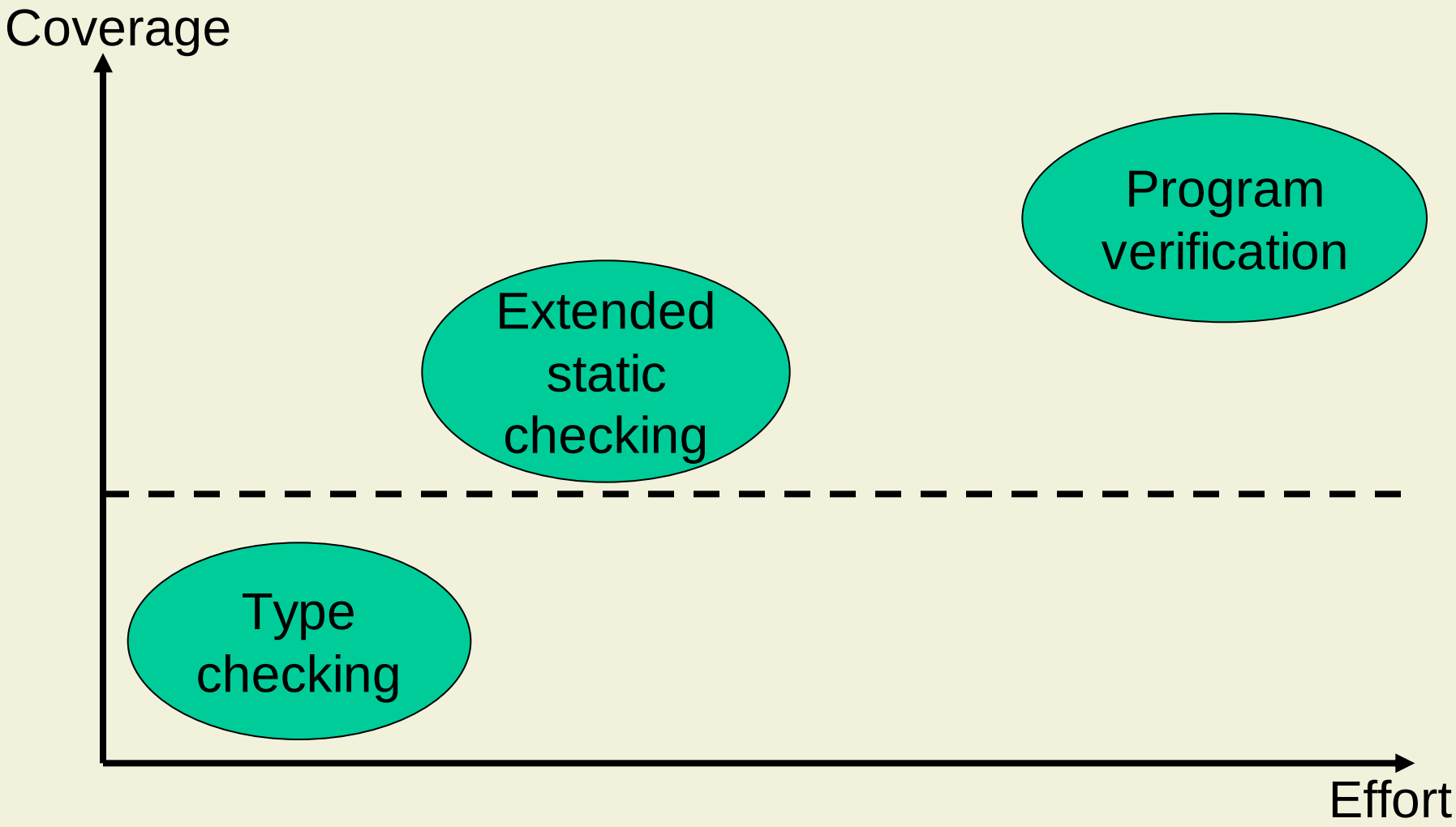
Agenda for Today

12. Extended Static Checking

Objectives

- Excitement

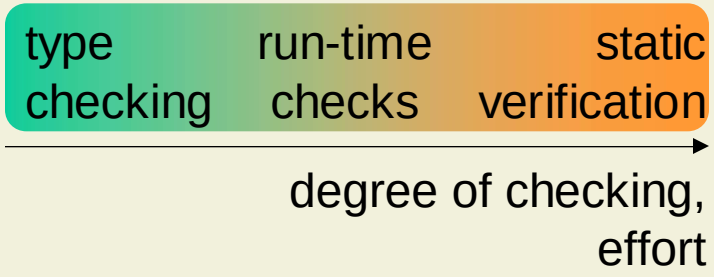
Extended Static Checking



Spec#

- Experimental mix of contracts and tools

- Superset of C#
  - non-null types
  - pre- and postconditions
  - object invariants
- Tool support
  - more type checking
  - compiler-emitted run-time checks
  - static program verification



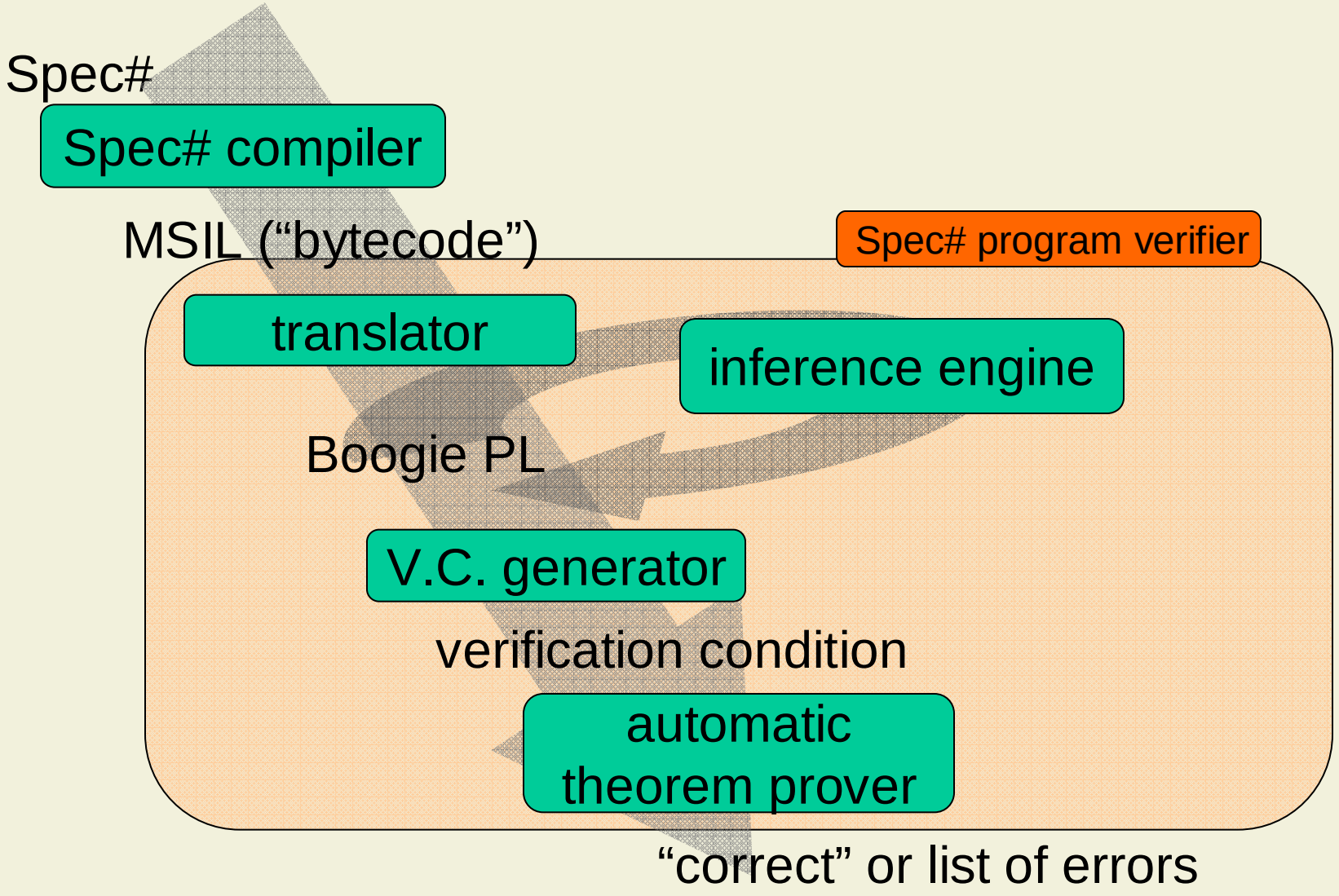
Boogie

- Boogie checks programs for coding errors …
  - Null-dereferences
  - Array bounds errors
  - Illegal casts
- … and specification violations
  - Simple pre-post specifications
  - Simple invariants

Program Checker Design Tradeoffs

- Objectives
  - Fully automated reasoning
  - As little annotation overhead as possible
  - Performance
- Boogie is sound
  - No errors are missed
- Boogie is not complete
  - Warnings do not always report errors (false alarms)

Boogie Architecture



Boogie Example

```
public static int f( int n )
  ensures result == (( n > 100 ) ? n – 10 : 91);
{
  if ( n > 100)   return n – 10;
  else            return f( f( n + 11 ) );
}
```