Regression Testing Techniques for Relational Database Applications

Gregory M. Kapfhammer†

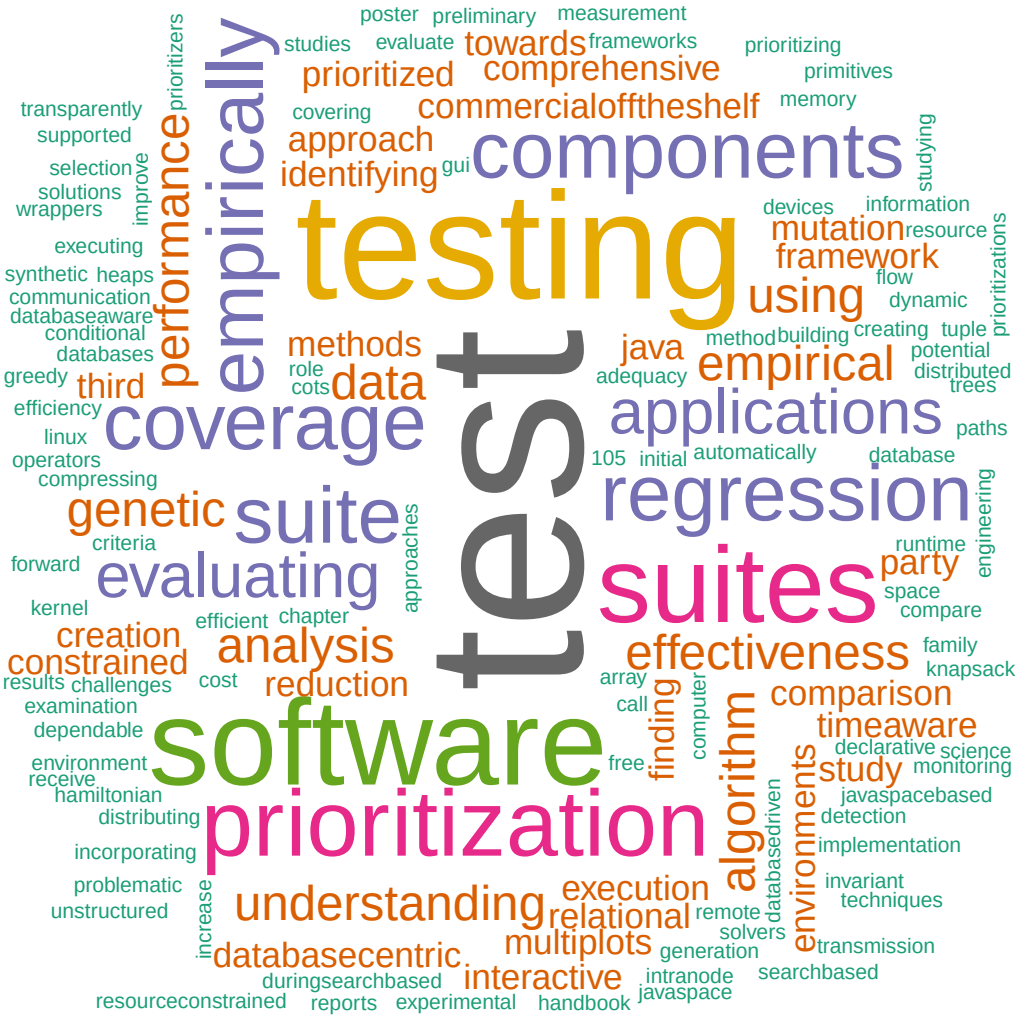
Department of Computer Science
Allegheny College
http://www.cs.allegheny.edu/~gkapfham/

University of Ulm – January 23, 2012

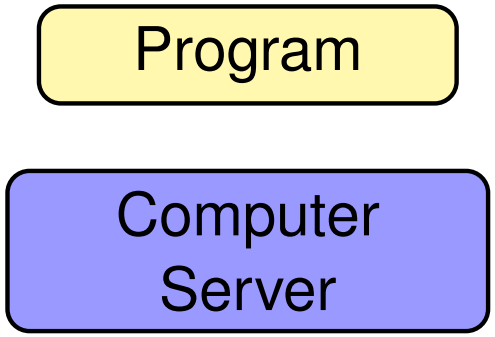
† Joint with Mary Lou Soffa (University of Virginia) and Jonathan Miller Kauffman (Allegheny College)

ALLEGHENY COLLEGE

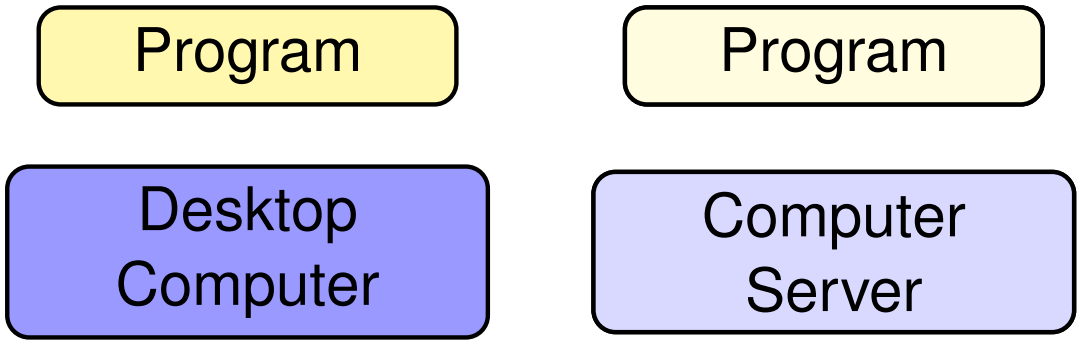
Presenter Introduction: Gregory M. Kapfhammer



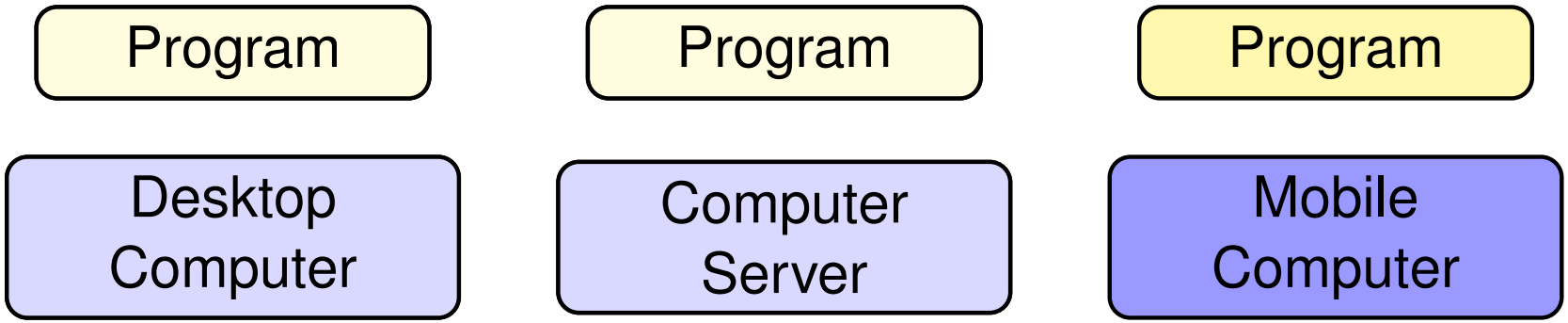
Software and Data are Everywhere



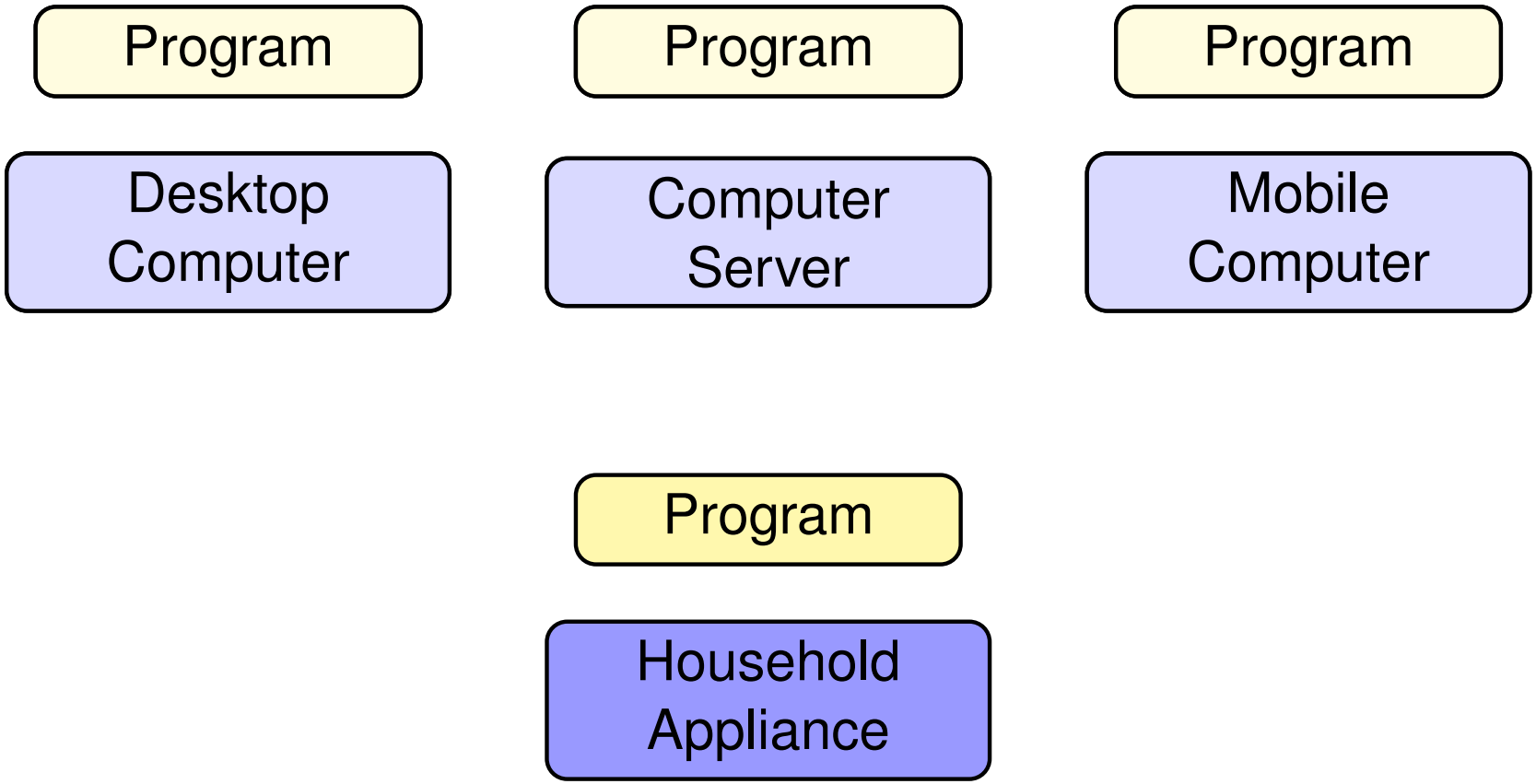
Software and Data are Everywhere



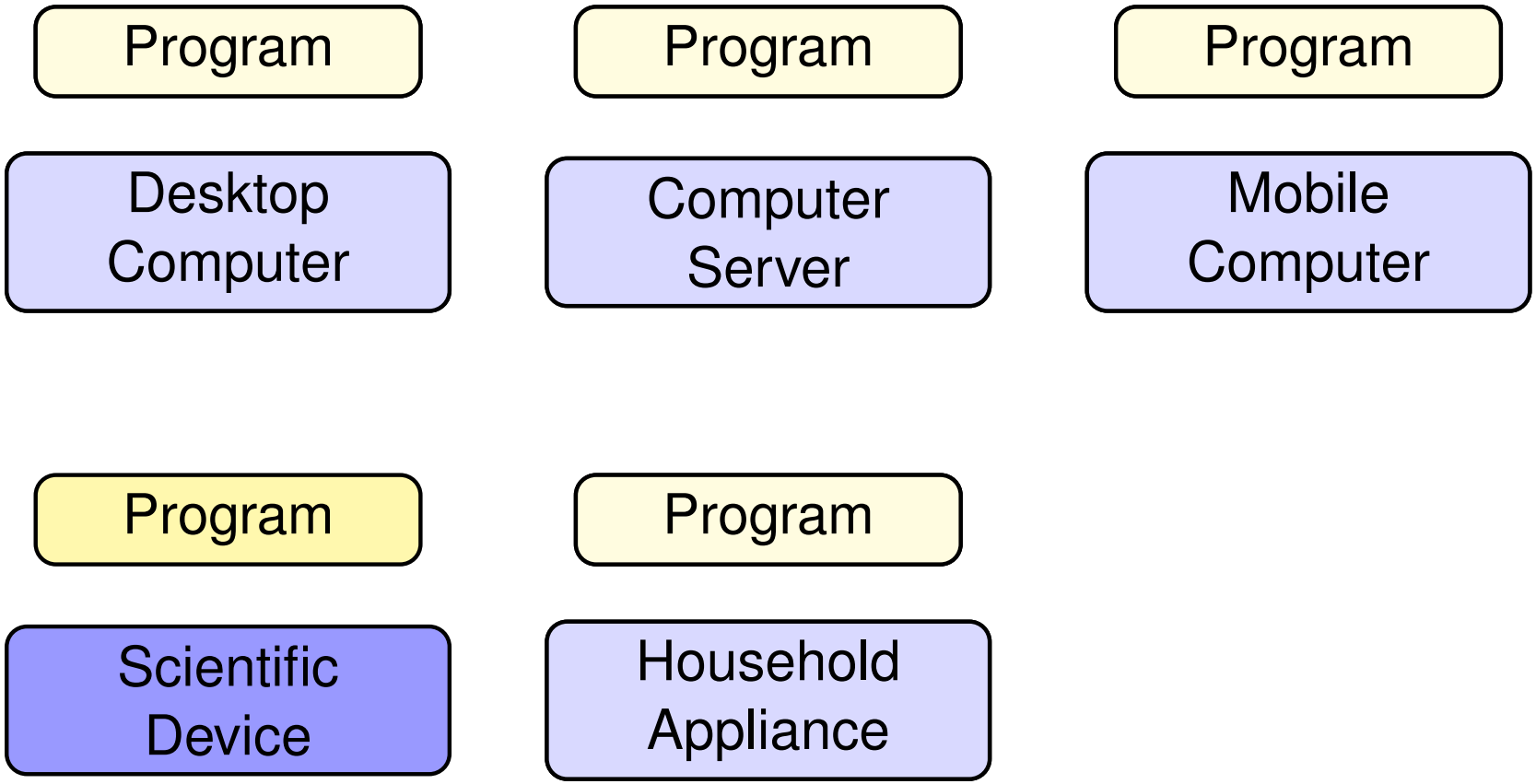
Software and Data are Everywhere



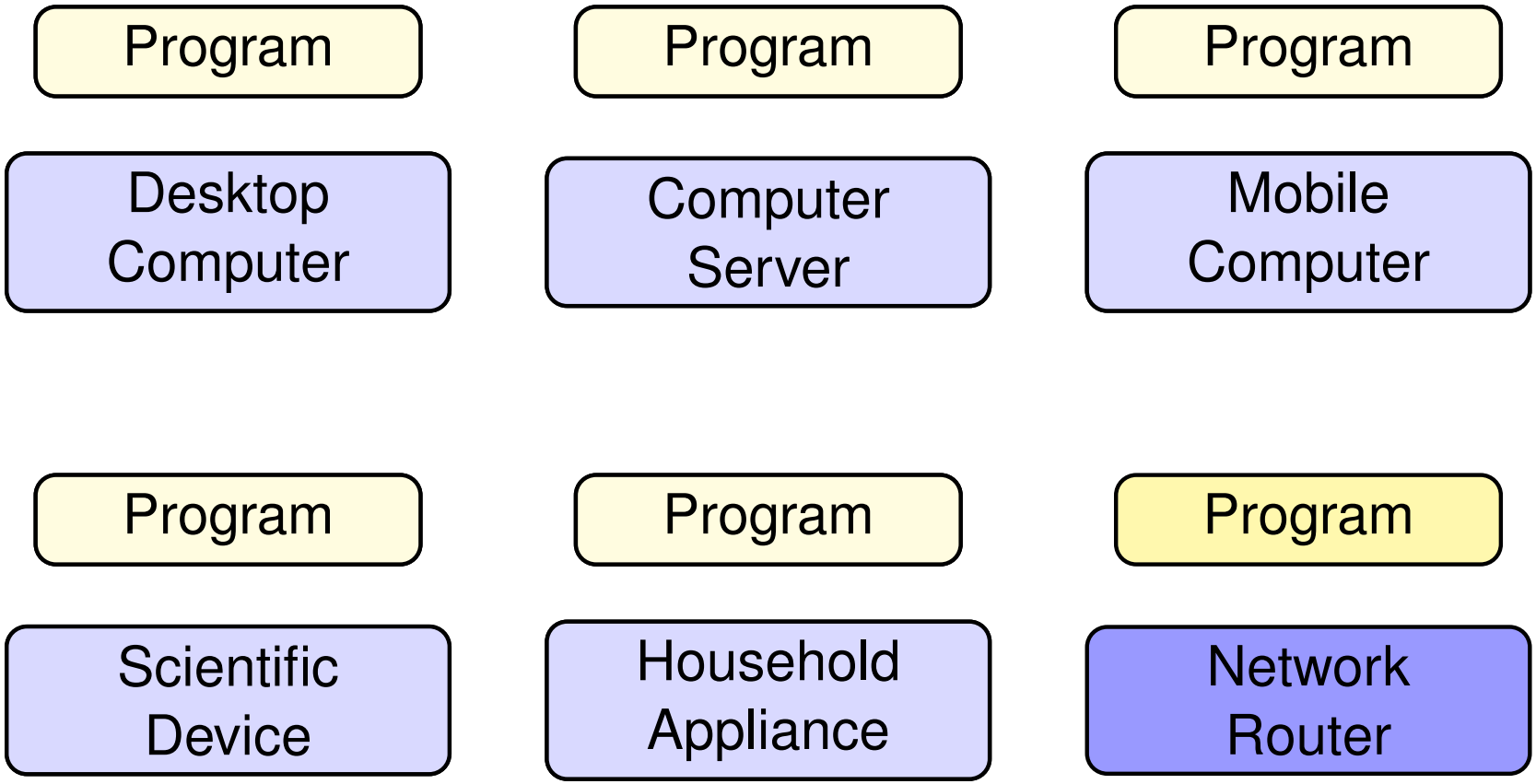
Software and Data are Everywhere



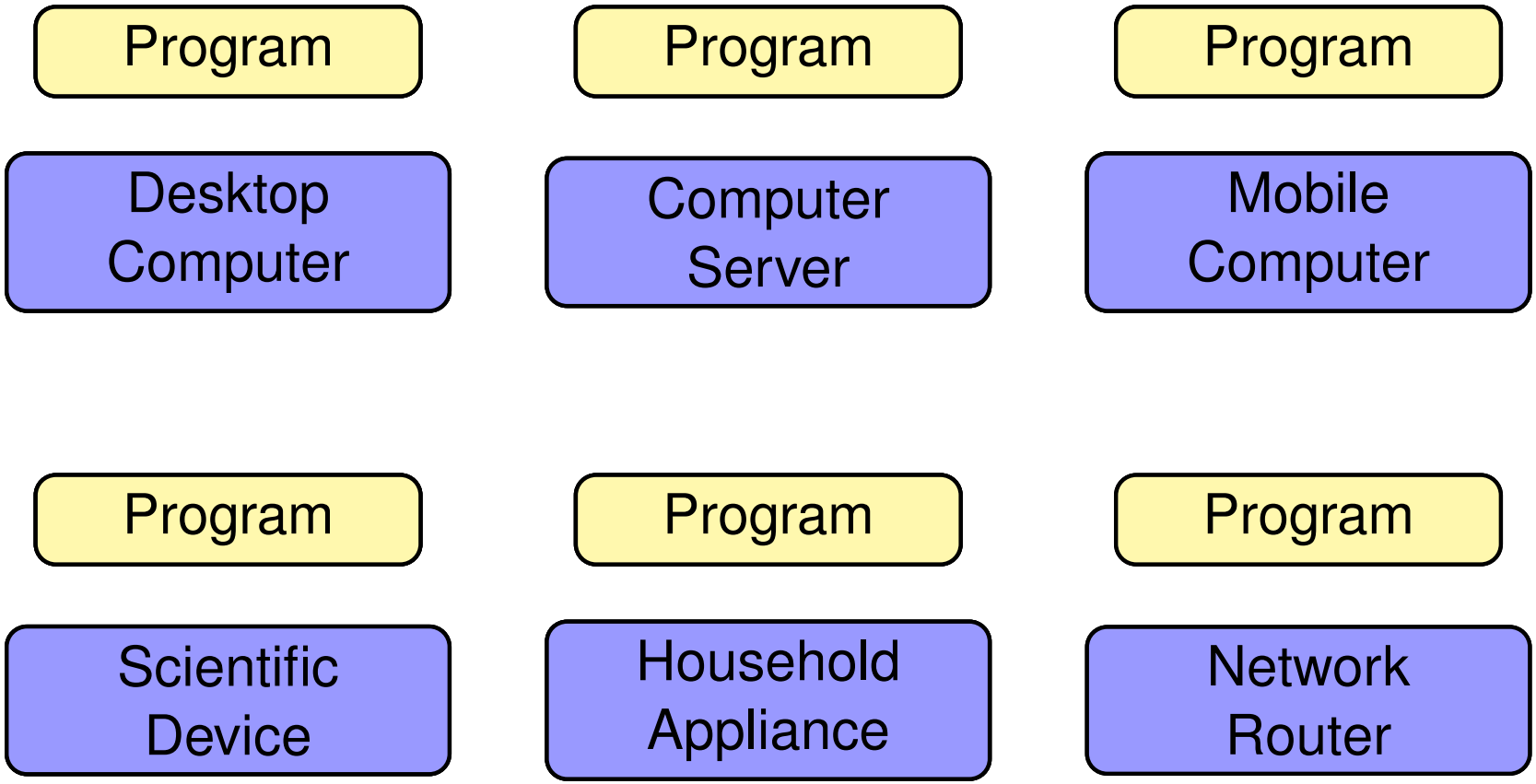
Software and Data are Everywhere



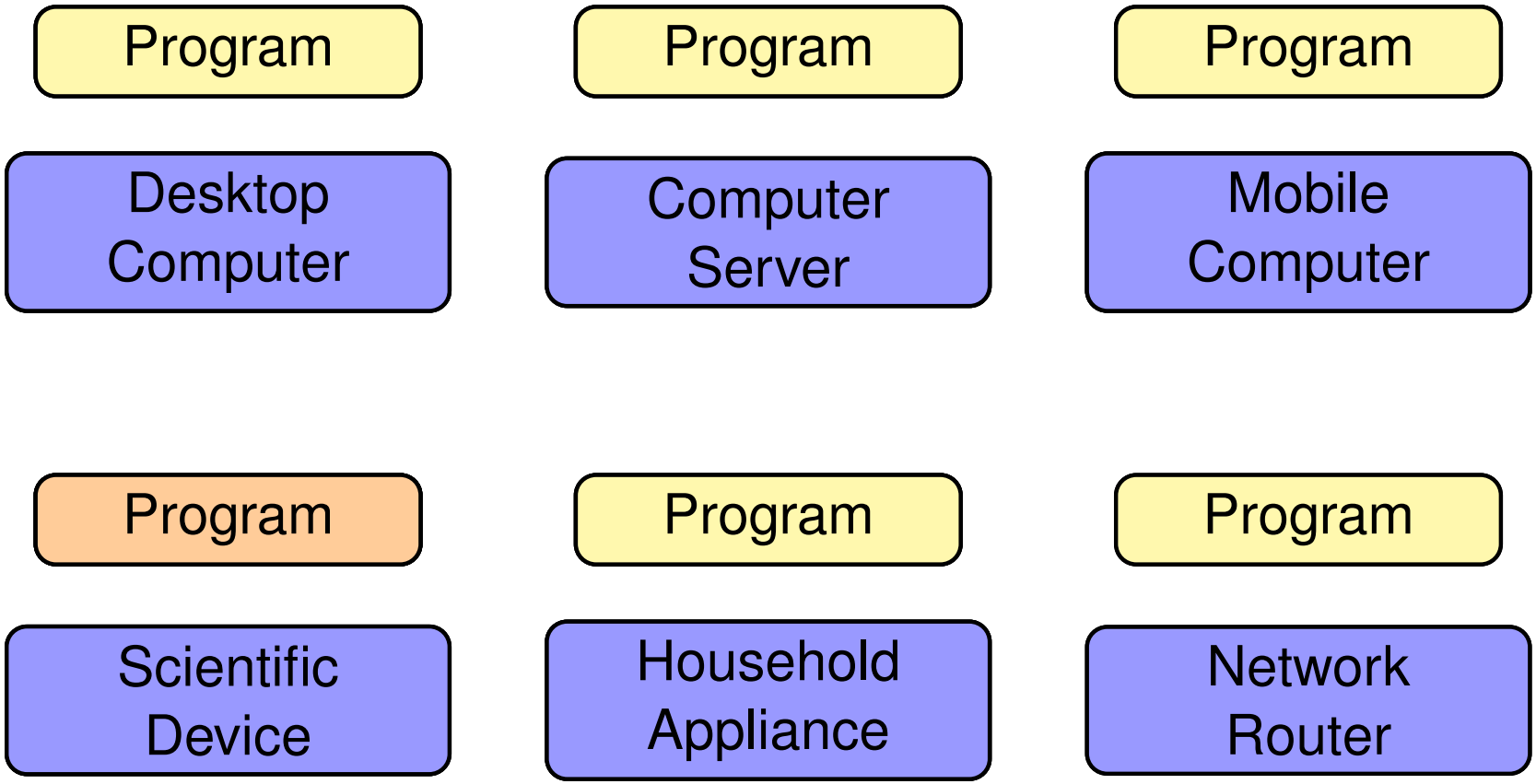
Software and Data are Everywhere



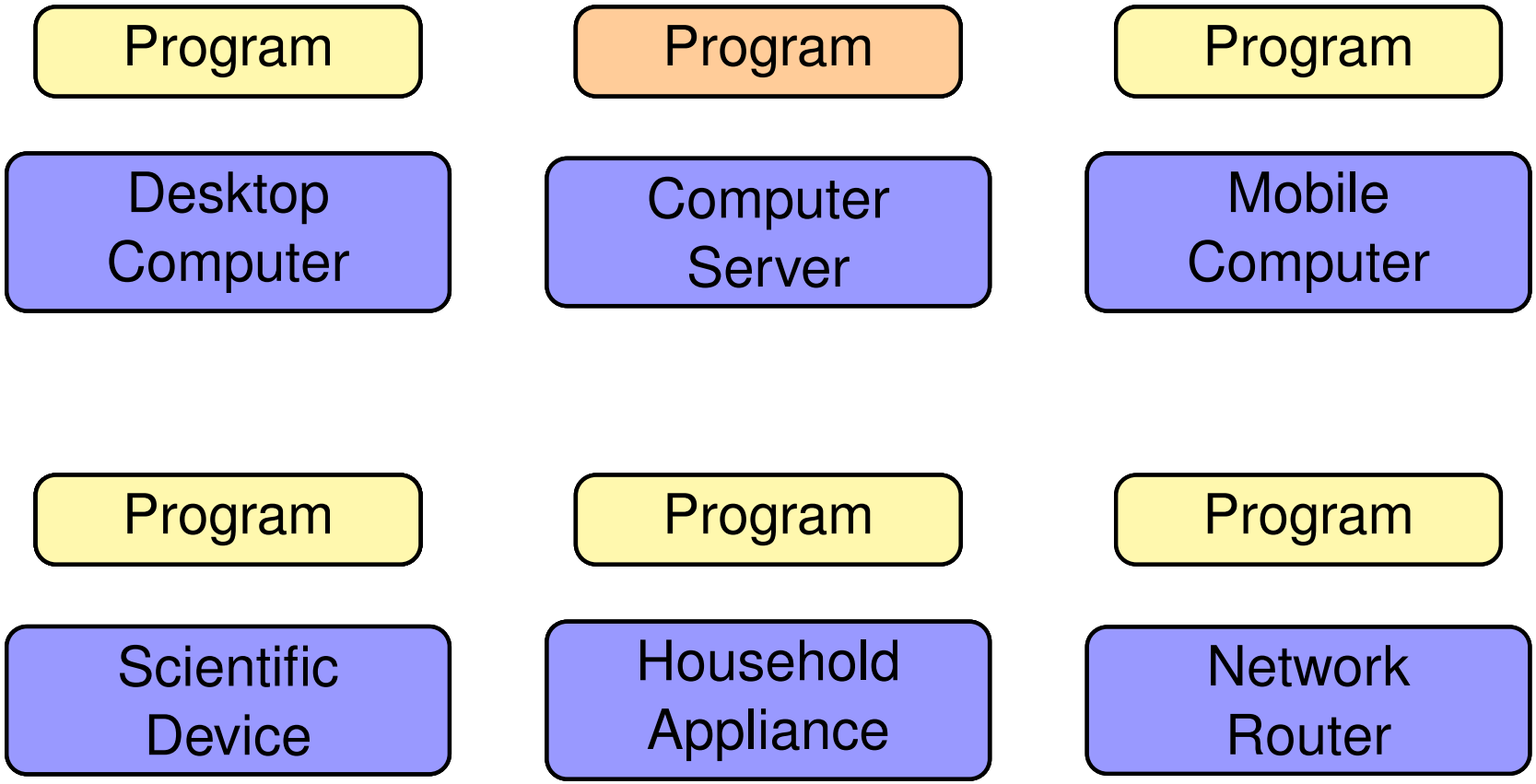
Software and Data are Everywhere



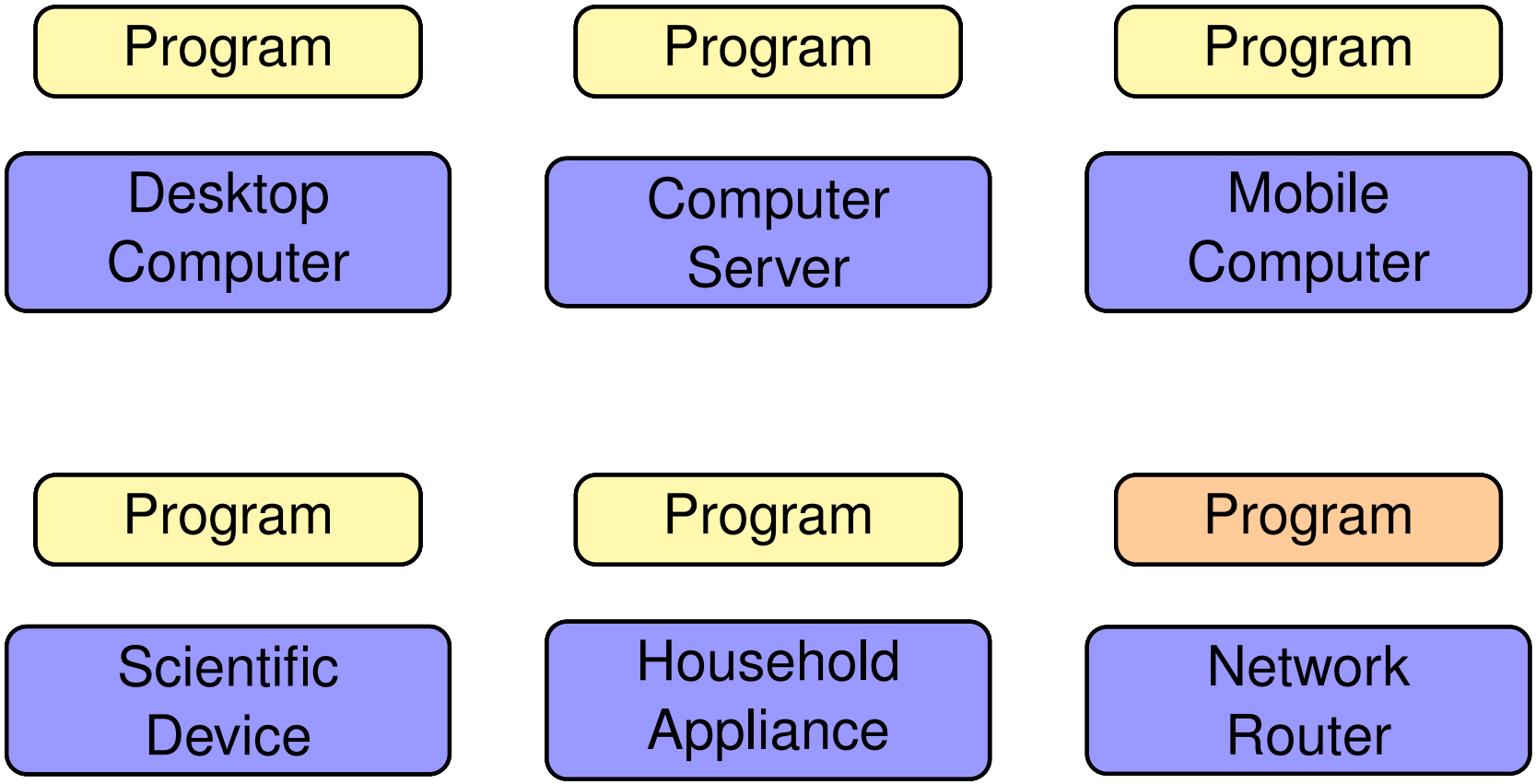
Software and Data are Everywhere



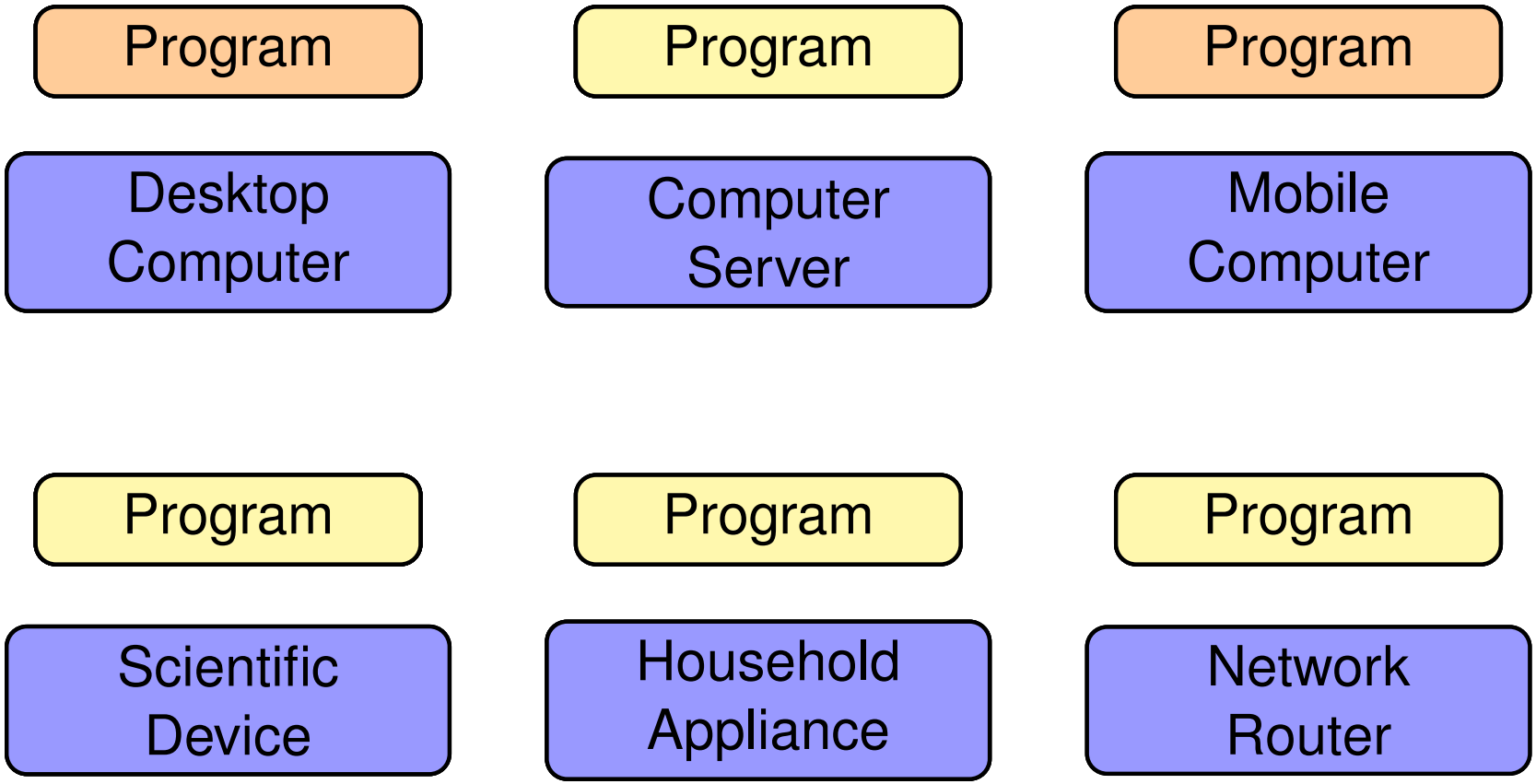
Software and Data are Everywhere



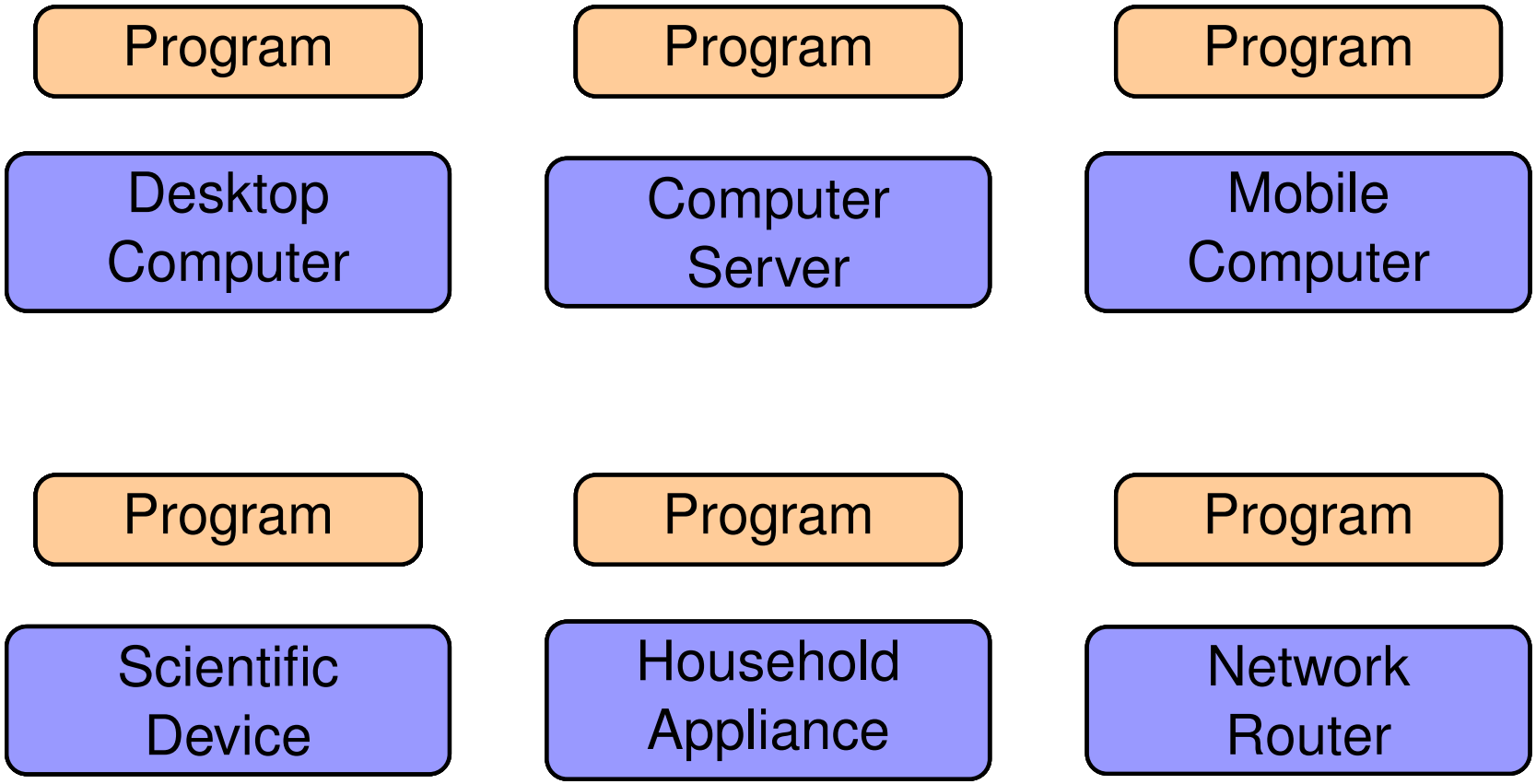
Software and Data are Everywhere



Software and Data are Everywhere



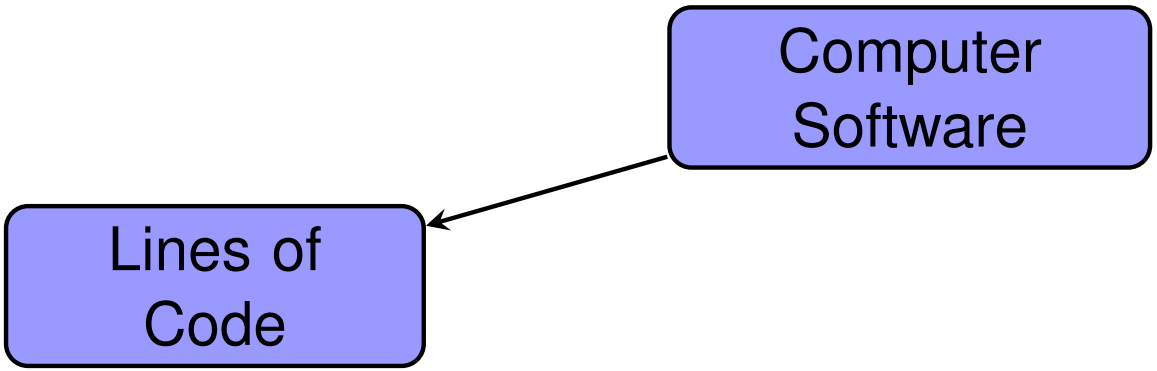
Software and Data are Everywhere



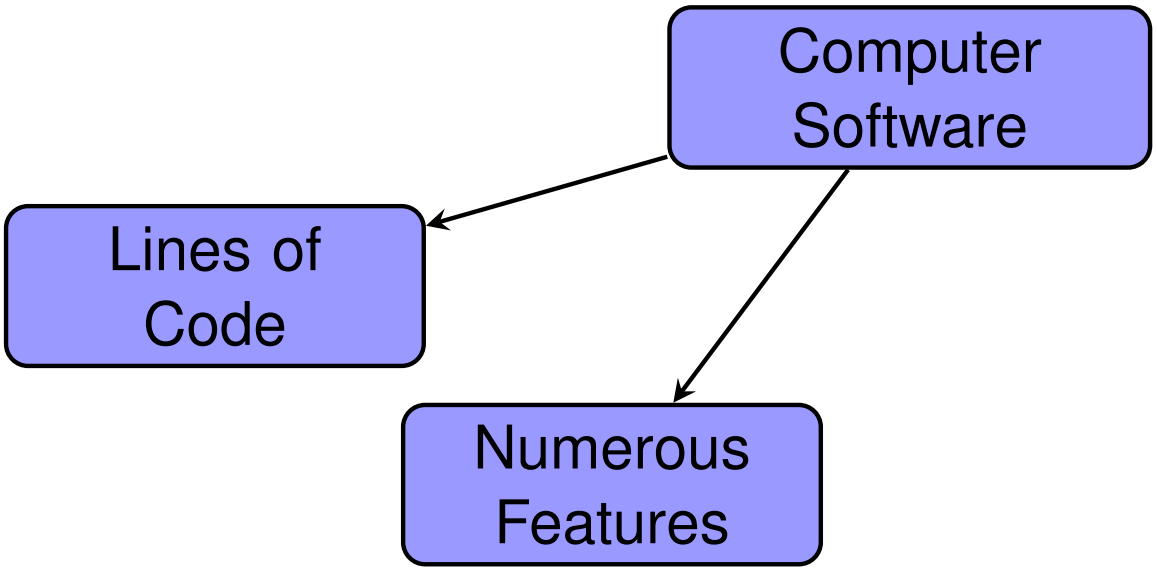
Software Complexity and Data Enormity

Computer Software

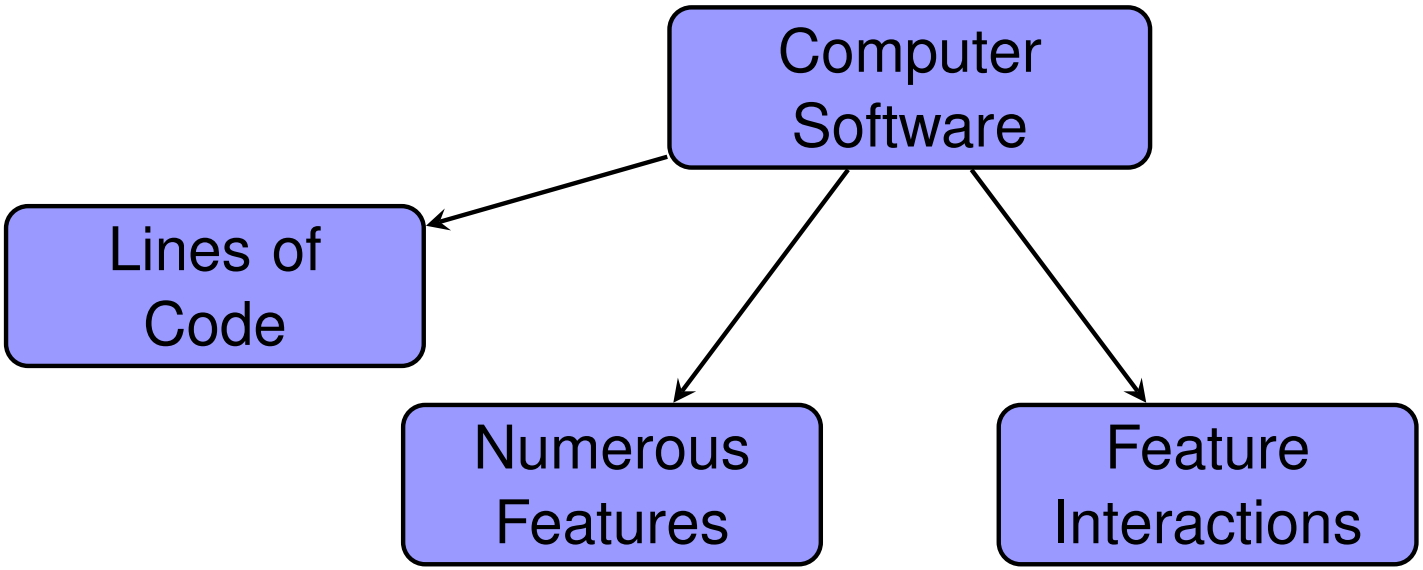
Software Complexity and Data Enormity



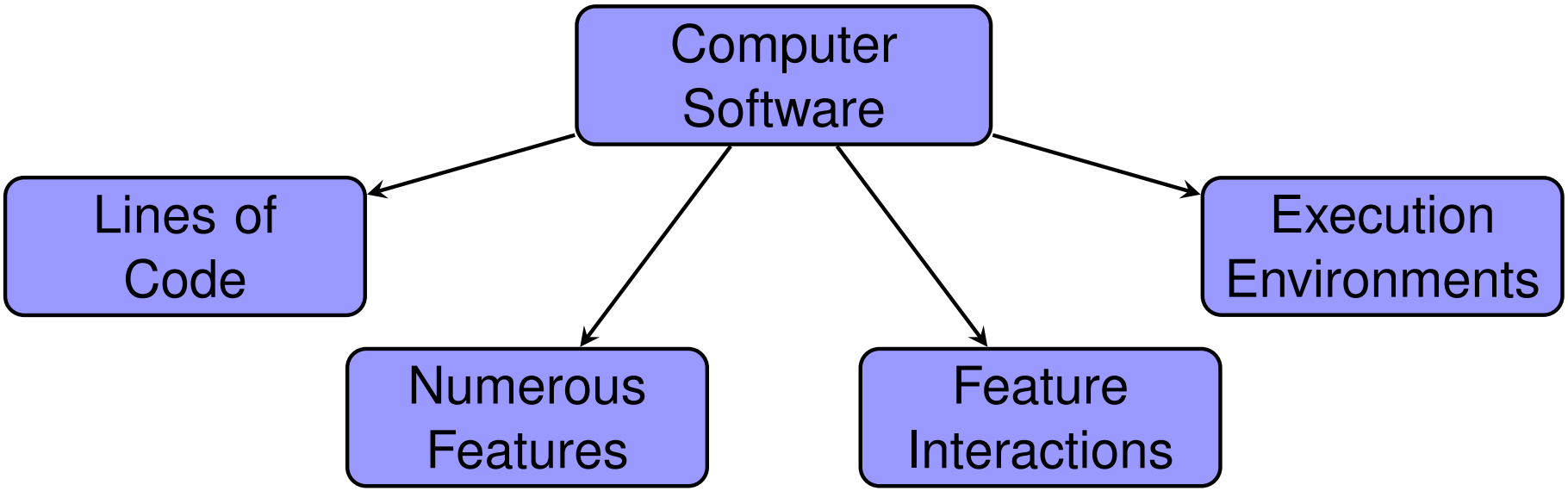
Software Complexity and Data Enormity



Software Complexity and Data Enormity

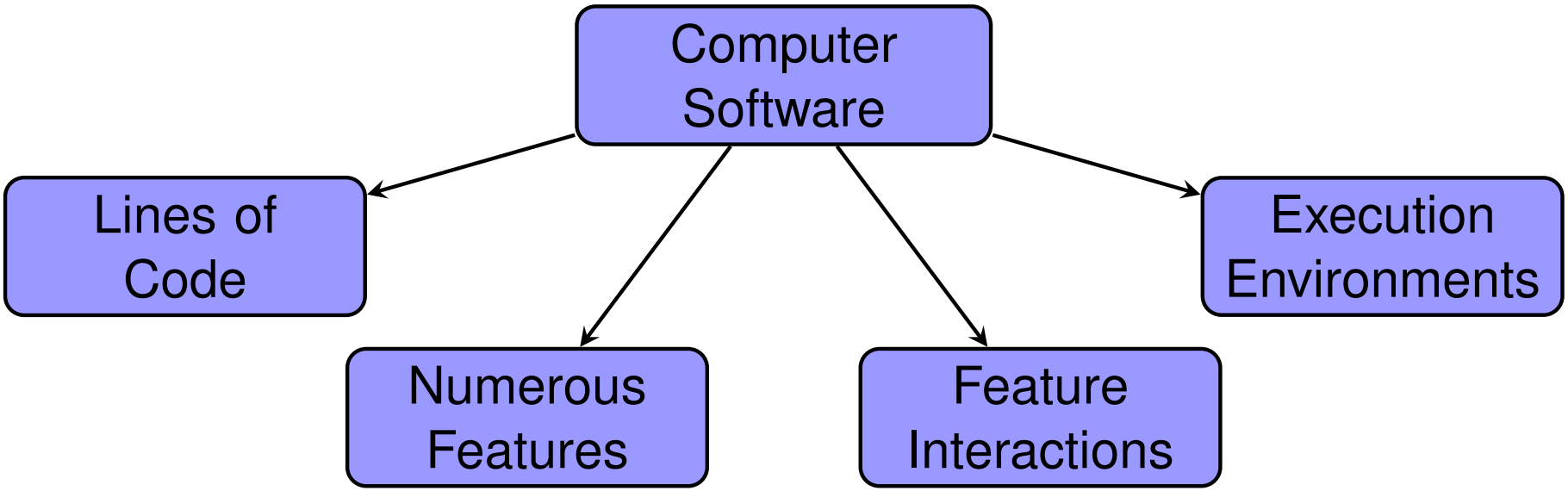


Software Complexity and Data Enormity



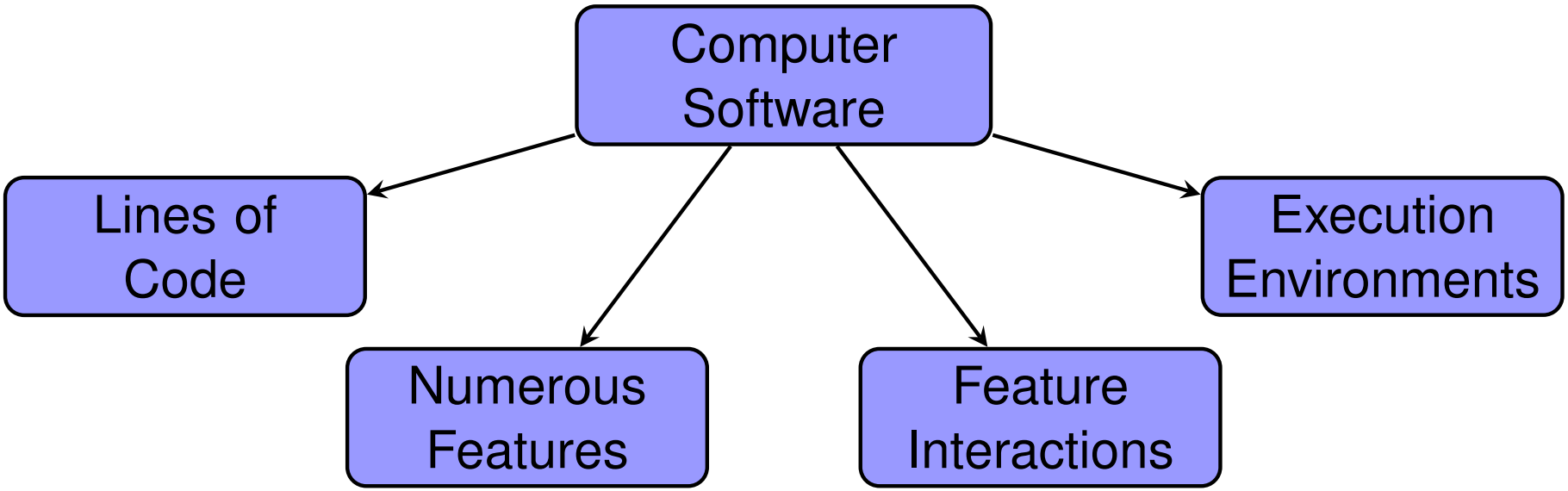
Software Complexity and Data Enormity

Software entities are more complex for their size than perhaps any other human construct - Frederick P. Brooks, Jr.

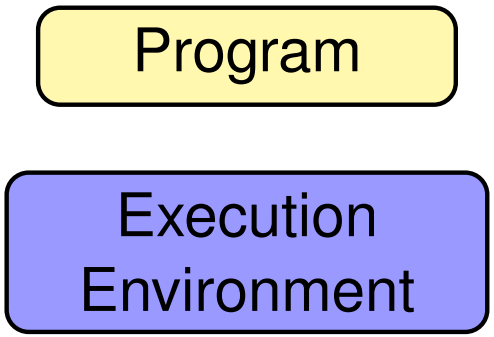


Software Complexity and Data Enormity

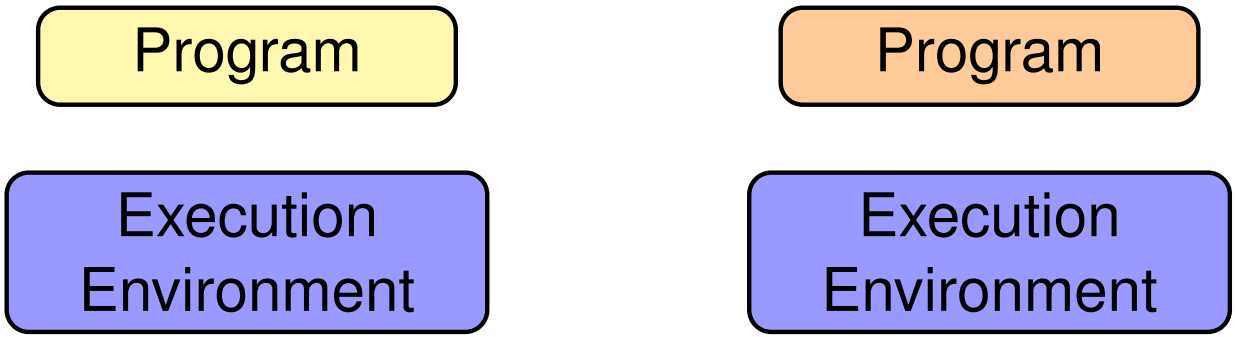
Prediction: in 2011, 1.8 zettabytes (i.e., 1.8 trillion gigabytes) of data will be created - IDC Digital Universe Study



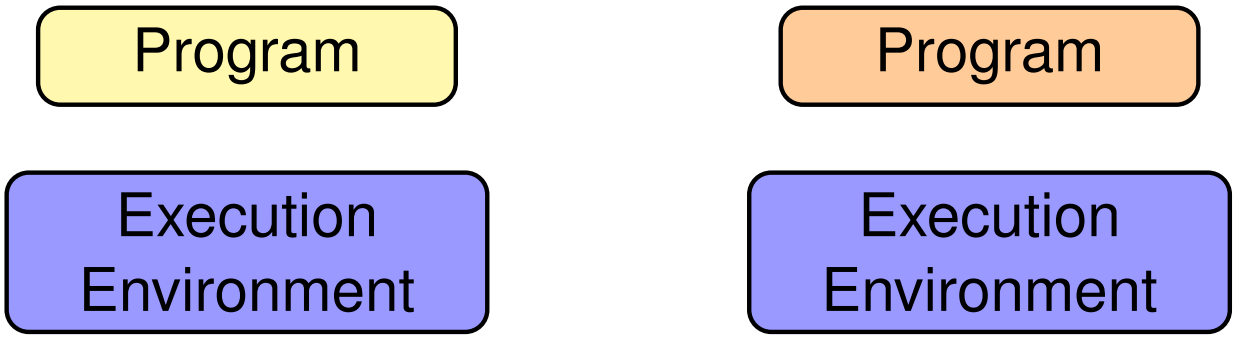
Software and Data are Evolving



Software and Data are Evolving

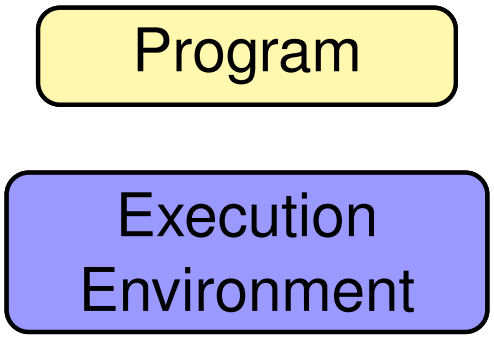


Software and Data are Evolving

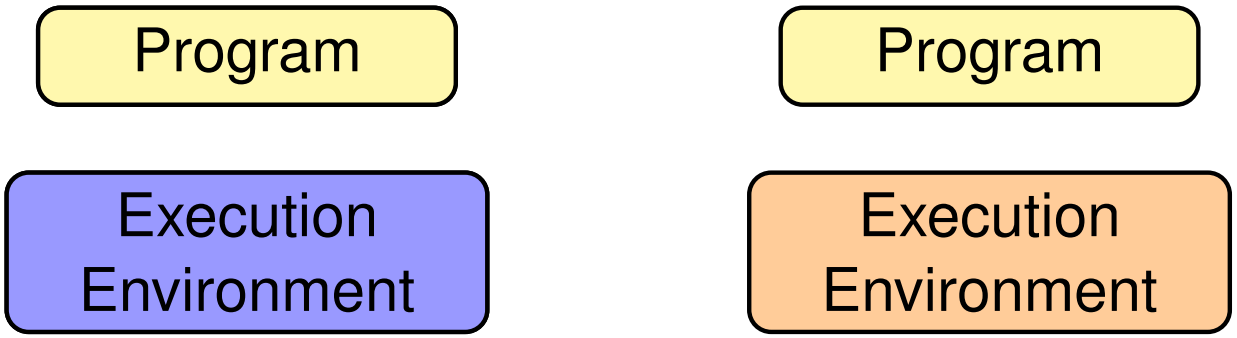


Program Changed because of the addition of a new feature or the correction of a defect

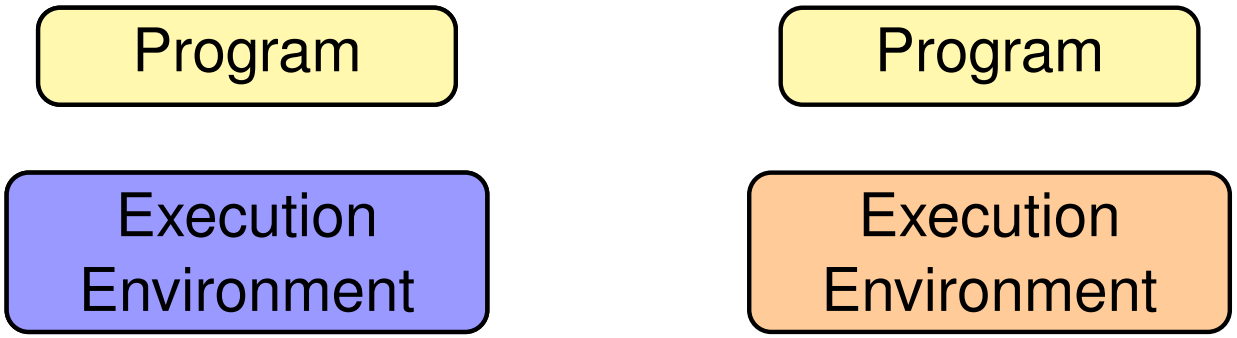
Software and Data are Evolving



Software and Data are Evolving



Software and Data are Evolving



Execution Environment Changed due to modification of a kernel, device driver, or relational database

An Interesting Defect Report

Database
Server Crashes

An Interesting Defect Report

Database
Server Crashes

When you run a complex query against Microsoft SQL Server 2000, the SQL Server scheduler may stop responding. Additionally, you receive an error message that resembles the following: **Date Time server Error: 17883 Severity: 1, State: 0 Date Time server Process 52:0 (94c) ...**

An Interesting Defect Report

Input-Dependent Defect

An Interesting Defect Report

Input-Dependent Defect

This problem occurs when one or more of the following conditions are true: The query contains a `UNION` clause or a `UNIONALL` clause that affects many columns. The query contains several `JOIN` statements. The query has a large estimated cost. **BUG 473858 (SQL Server 8.0)**

Real-World Defective Database Application

The Risks Digest, Volume 22, Issue 64, 2003

Jeppesen reports airspace boundary problems

About 350 airspace boundaries contained in Jeppesen NavData are incorrect, the FAA has warned. The error occurred at Jeppesen after a software upgrade when information was pulled from a database containing 20,000 airspace boundaries worldwide for the March NavData update, which takes effect March 20.

Real-World Defective Database Application

The Risks Digest, Volume 22, Issue 64, 2003
Jeppesen reports airspace boundary problems

About 350 airspace boundaries contained in Jeppesen NavData are incorrect, the FAA has warned. The error occurred at Jeppesen after a software upgrade when information was pulled from a database containing 20,000 airspace boundaries worldwide for the March NavData update, which takes effect March 20.

Practically all use of databases occurs from within application programs [Silberschatz et al., 2006, pg. 311]

Structured Query Language

The *structured query language* (SQL) is an established standard and a query and manipulation language for *relational database management systems* (RDBMS)

Structured Query Language

The *structured query language* (SQL) is an established standard and a query and manipulation language for *relational database management systems* (RDBMS)

A *schema* is a collection of table definitions:

```
CREATE TABLE person (
id INT,
name VARCHAR(100) NOT NULL,
age INT(3),
PRIMARY KEY (id)
)
```

Structured Query Language

The *structured query language* (SQL) is an established standard and a query and manipulation language for *relational database management systems* (RDBMS)

The *data manipulation language* supports several operations:

```
SELECT name FROM person WHERE age >= 30 AND age <= 40
```

Structured Query Language

The *structured query language* (SQL) is an established standard and a query and manipulation language for *relational database management systems* (RDBMS)

The *data manipulation language* supports several operations:

```
UPDATE person SET name = Jan WHERE id = 2
```

Structured Query Language

The *structured query language* (SQL) is an established standard and a query and manipulation language for *relational database management systems* (RDBMS)

The *data manipulation language* supports several operations:

```
INSERT INTO person (id, name, age) VALUES
                   (1, John, 38)
```

Structured Query Language

The *structured query language* (SQL) is an established standard and a query and manipulation language for *relational database management systems* (RDBMS)

The *data manipulation language* supports several operations:

```
DELETE FROM person WHERE id = 2
```

Relational Database Tables

id	name	age
1	Chalker Conrad	12
2	Abby Clulow	14
3	David Rogan	18
4	Stacie Reckling	32
5	Megan Hartnup	29

Relational Database Tables

id	name	age
1	Chalker Conrad	12
2	Abby Clulow	14
3	David Rogan	18
4	Stacie Reckling	32
5	Megan Hartnup	29

Relational Database Tables

id	name	age
1	Chalker Conrad	12
2	Abby Clulow	14
3	David Rogan	18
4	Stacie Reckling	32
5	Megan Hartnup	29

Relational Database Tables

id	name	age
1	Chalker Conrad	12
2	Abby Clulow	14
3	David Rogan	18
4	Stacie Reckling	32
5	Megan Hartnup	29

Relational Database Tables

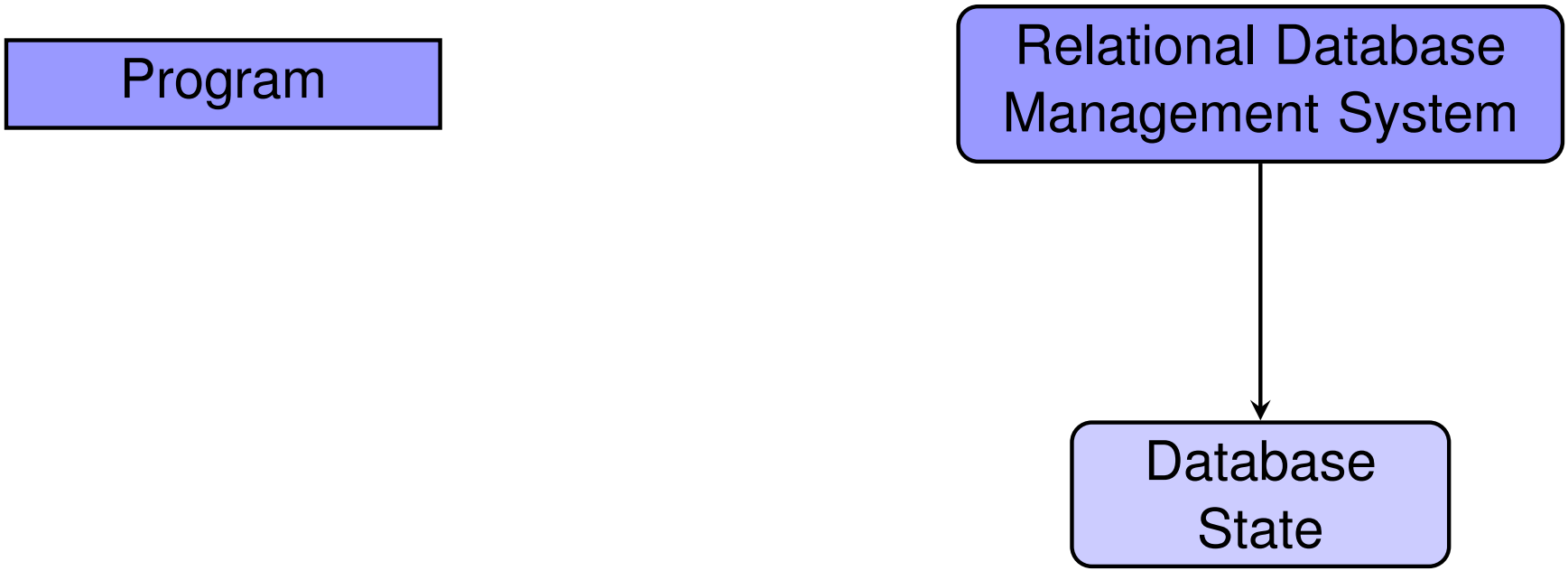
id	name	age
1	Chalker Conrad	12
2	Abby Clulow	14
3	David Rogan	18
4	Stacie Reckling	32
5	Megan Hartnup	29

Relational Database Tables

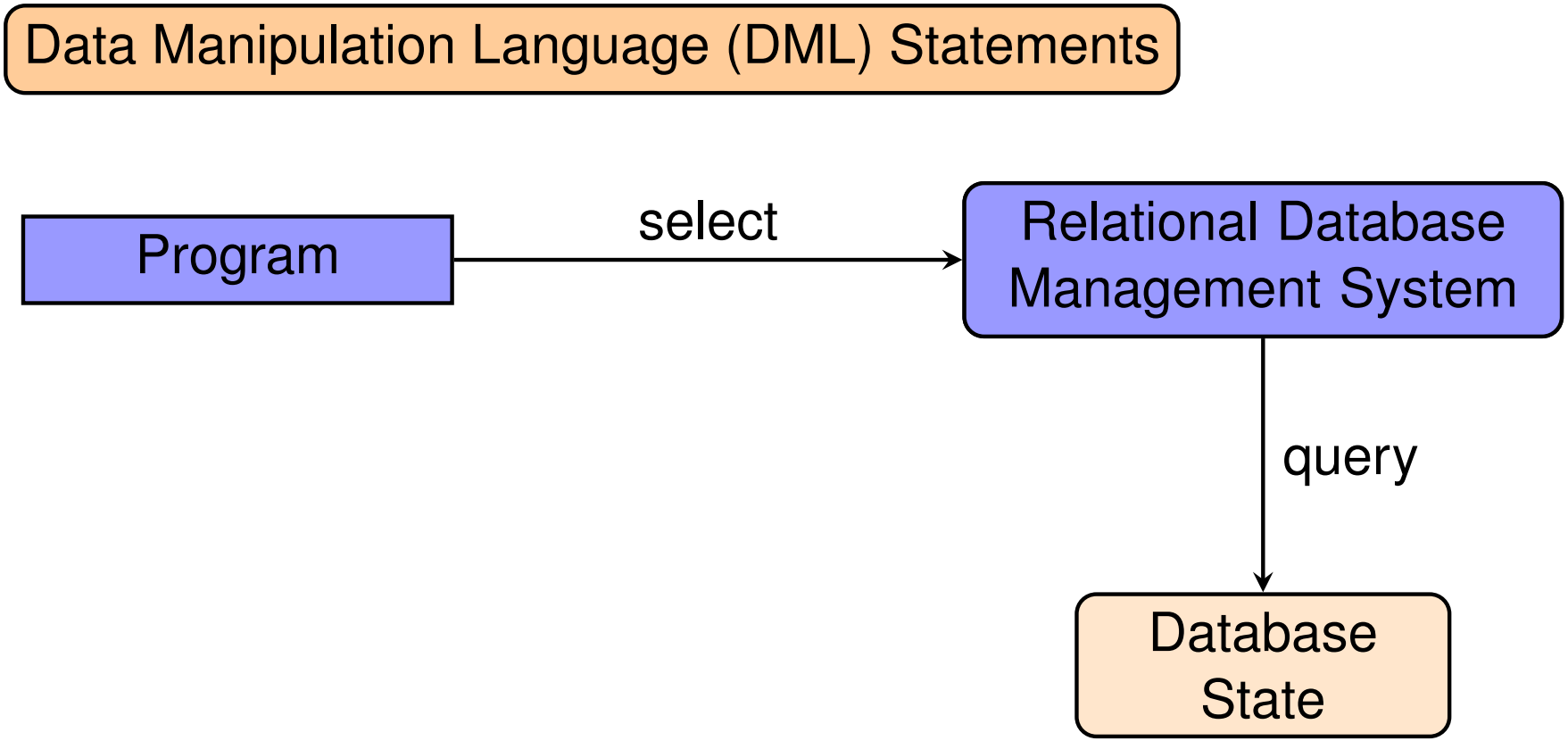
id	name	age
1	Chalker Conrad	12
2	Abby Clulow	14
3	David Rogan	18
4	Stacie Reckling	32
5	Megan Hartnup	29

Database Applications

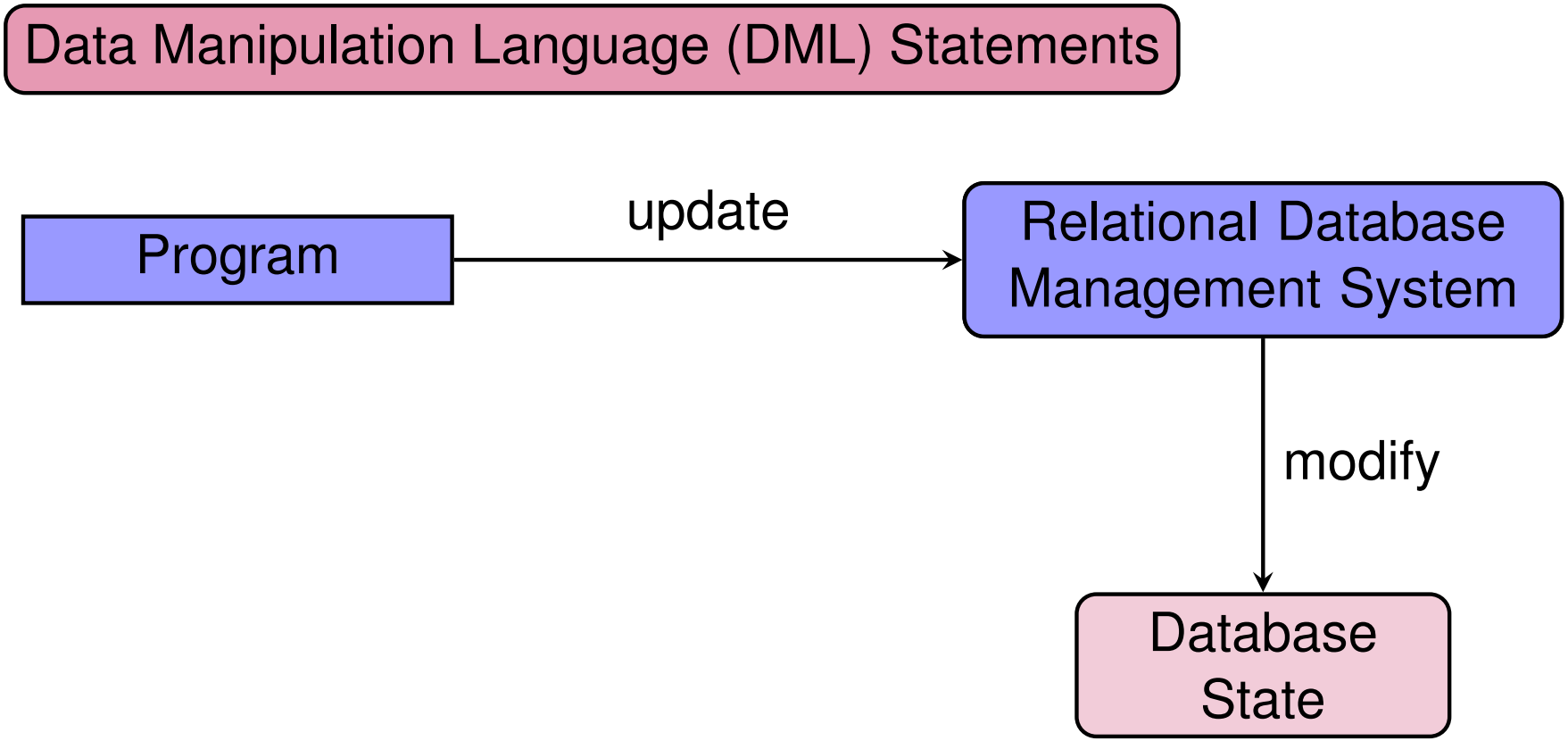
Database Applications



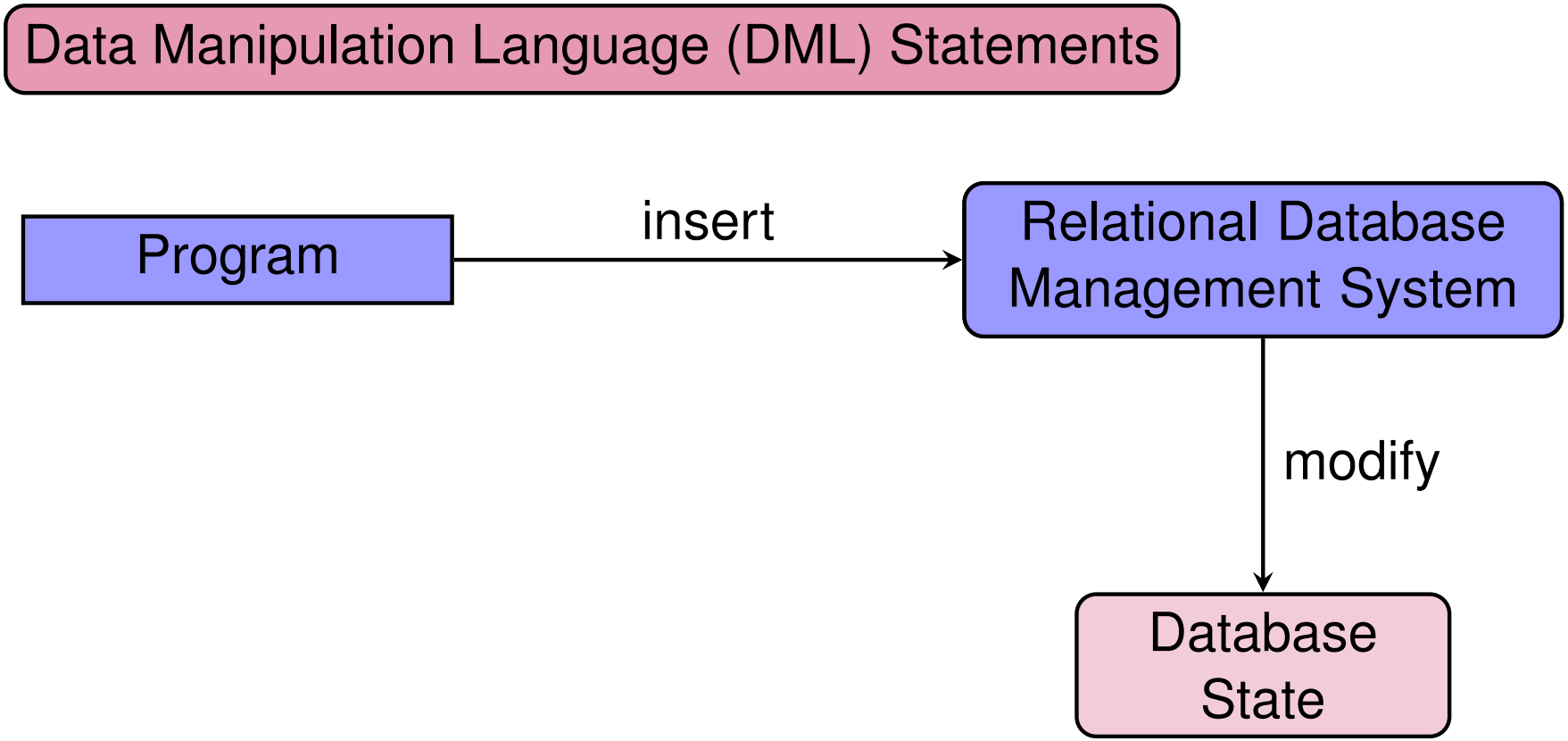
Database Applications



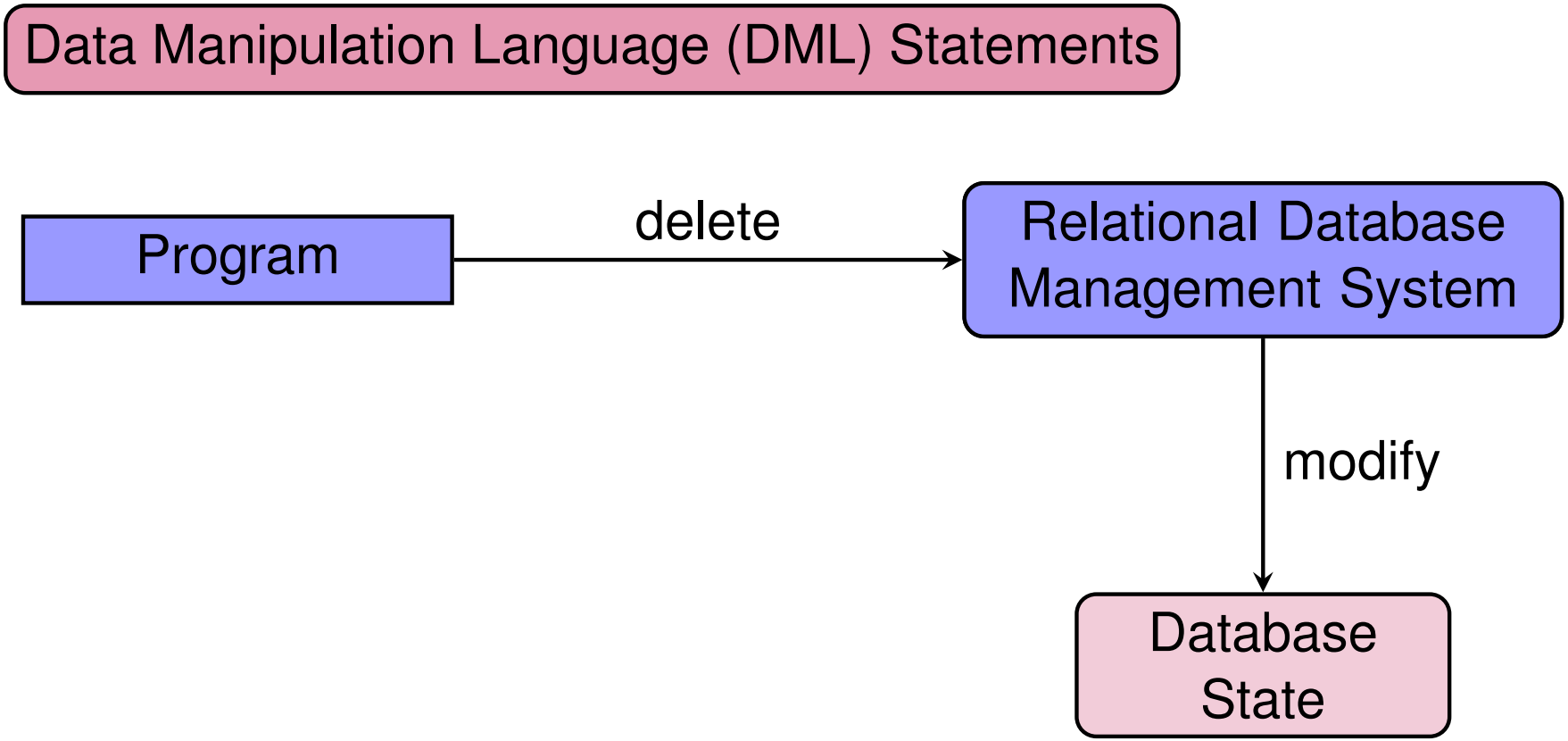
Database Applications



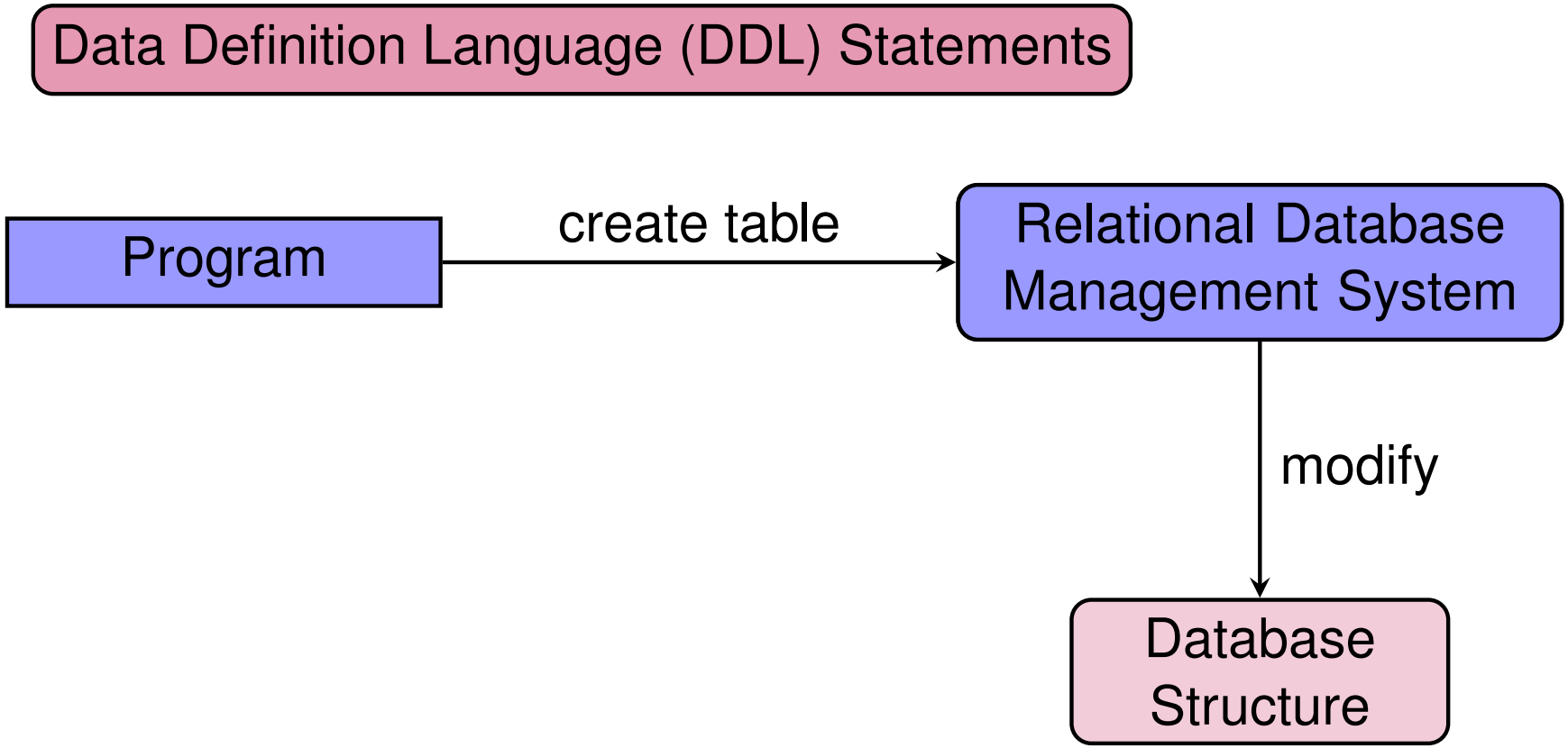
Database Applications



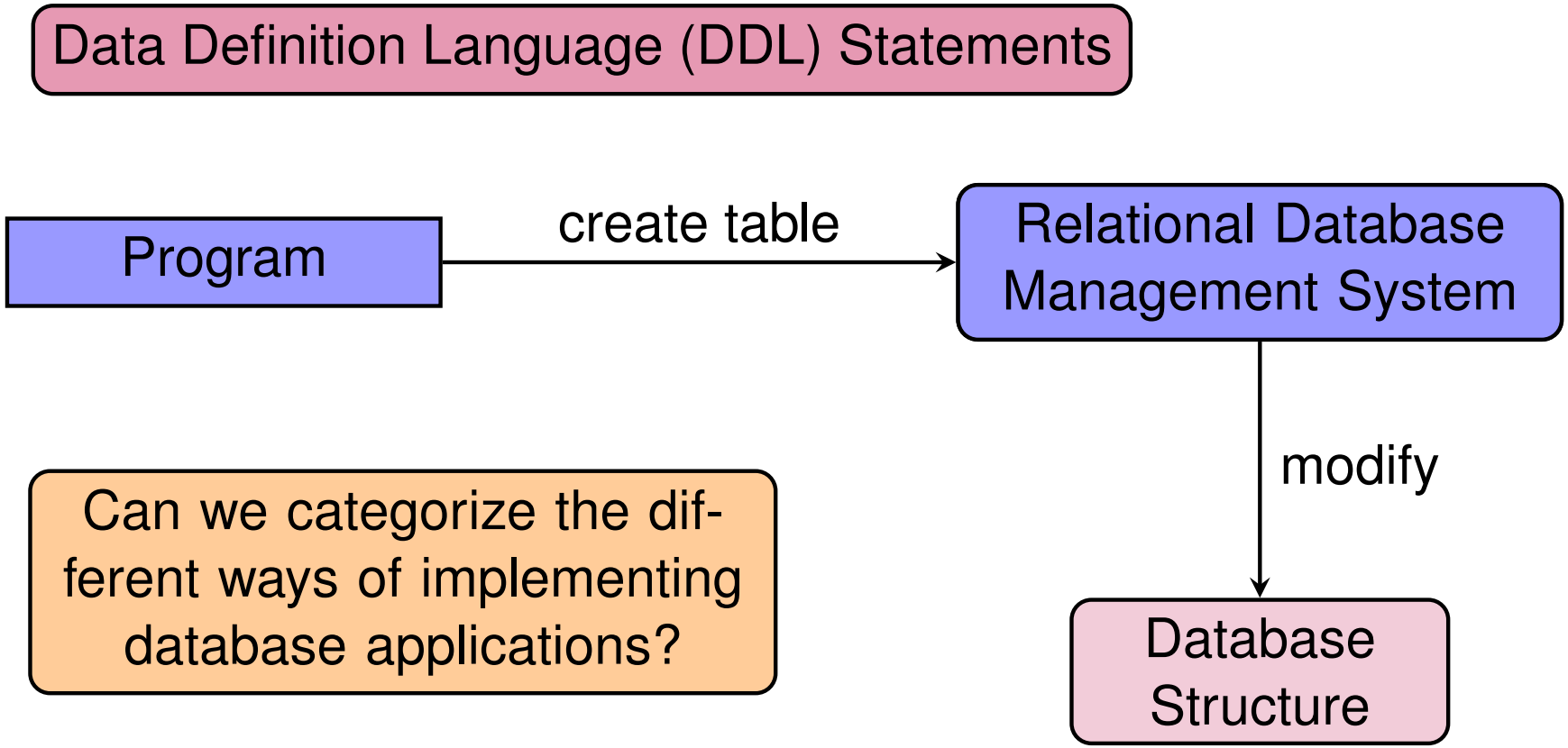
Database Applications



Database Applications

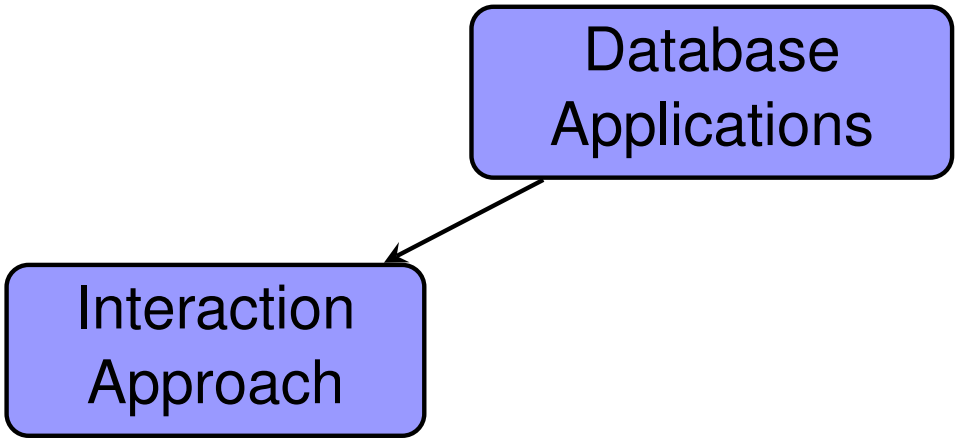


Database Applications

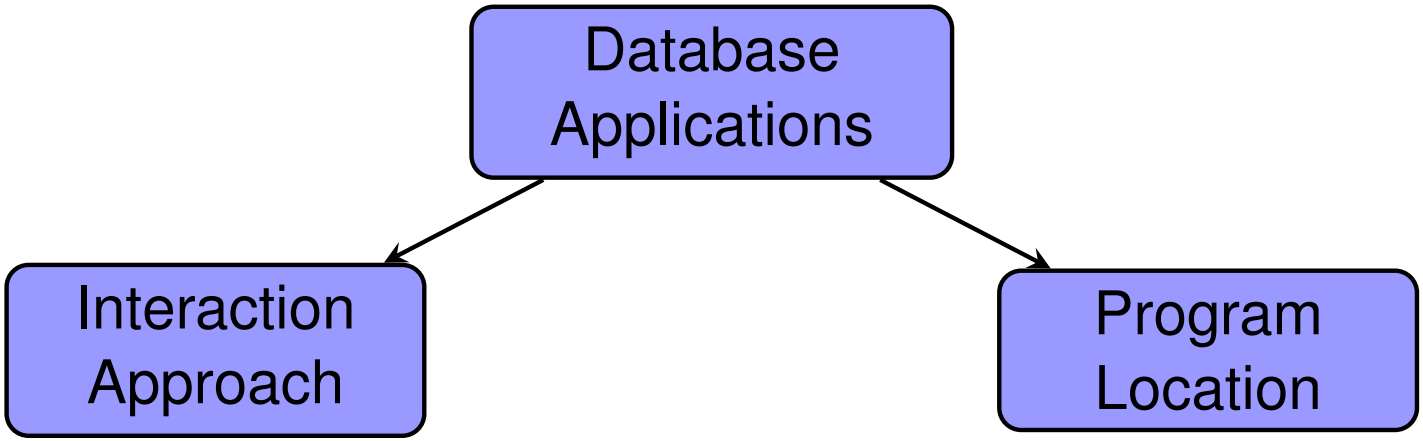


Categorizing Database Applications

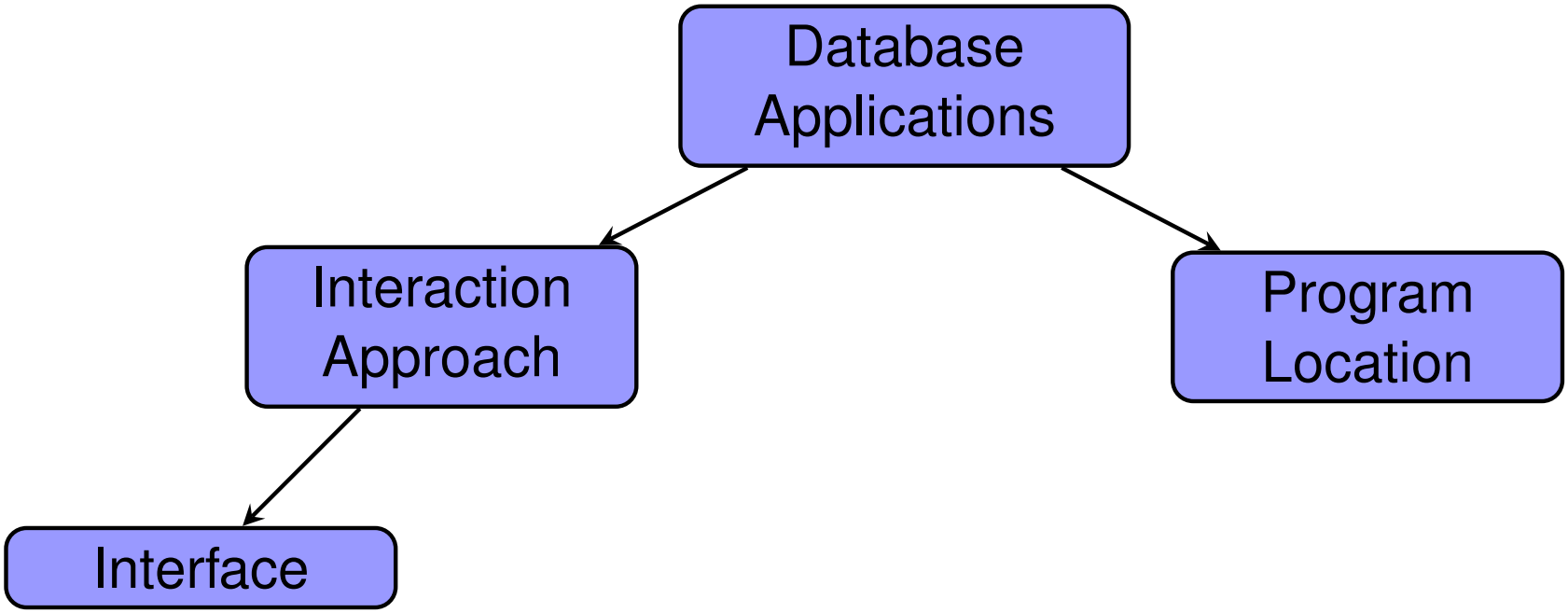
Categorizing Database Applications



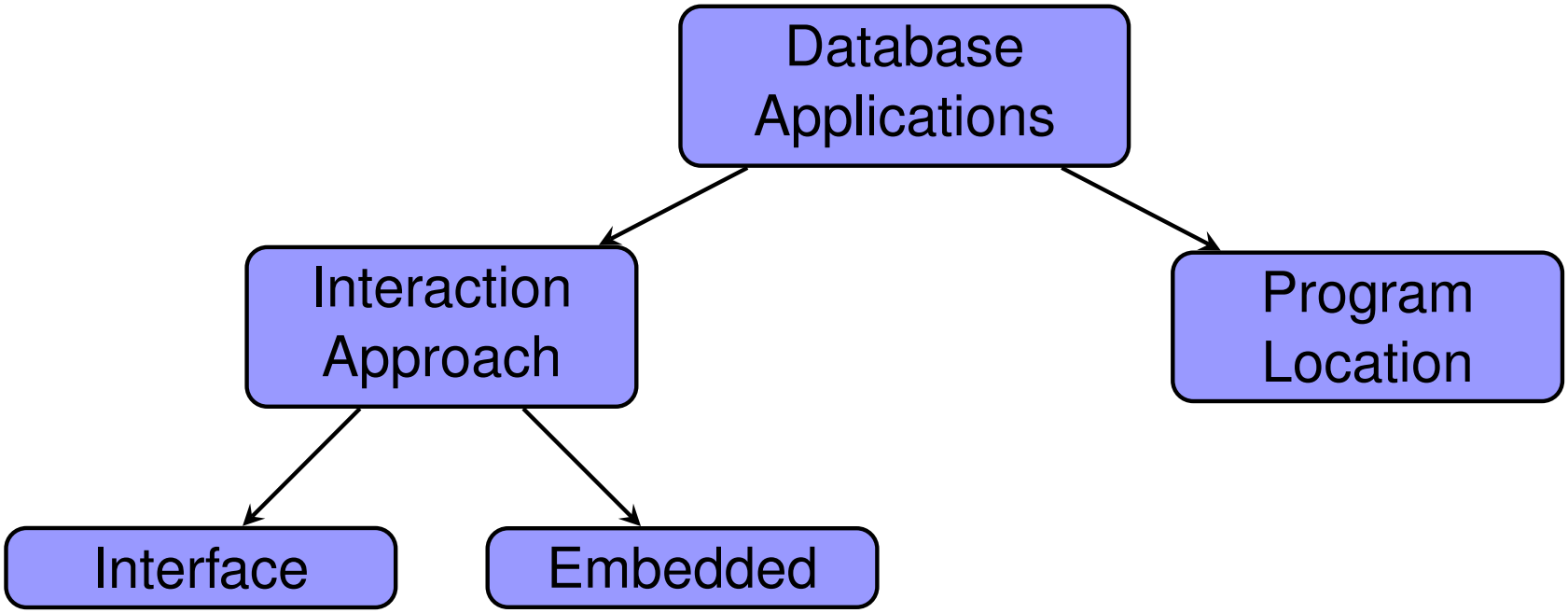
Categorizing Database Applications



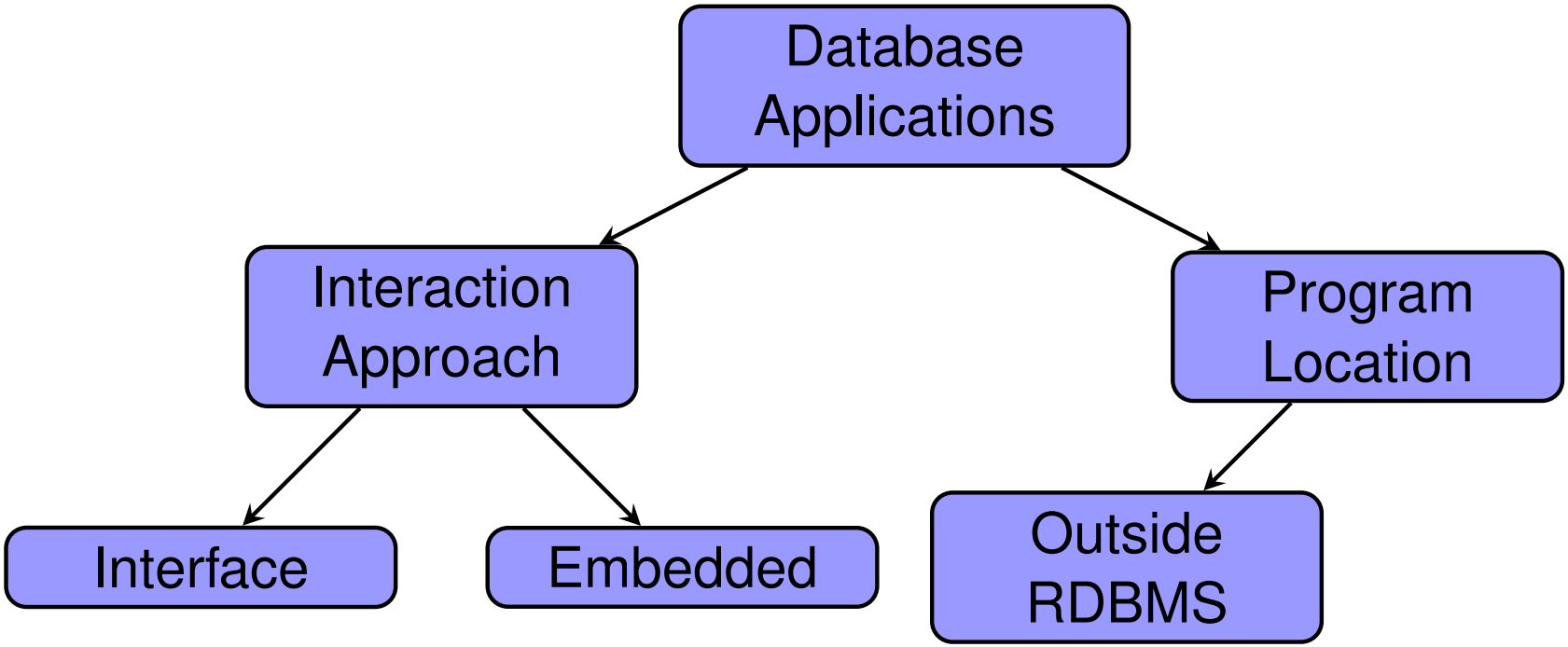
Categorizing Database Applications



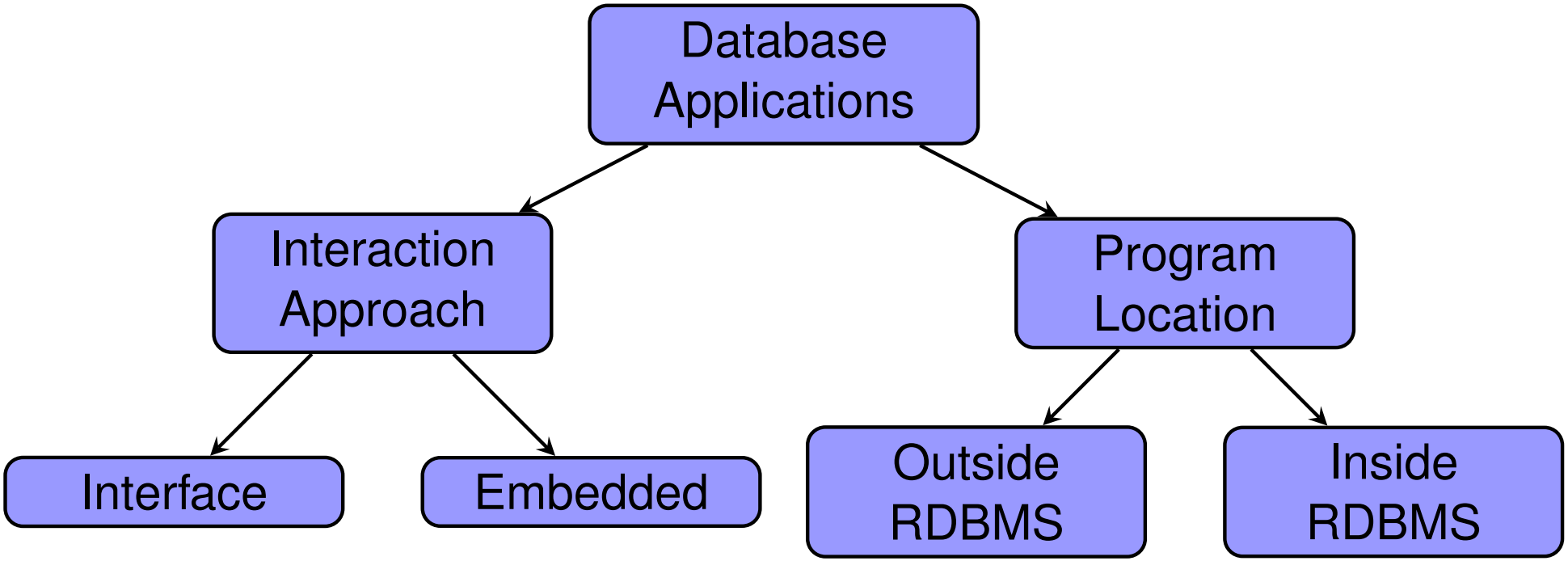
Categorizing Database Applications



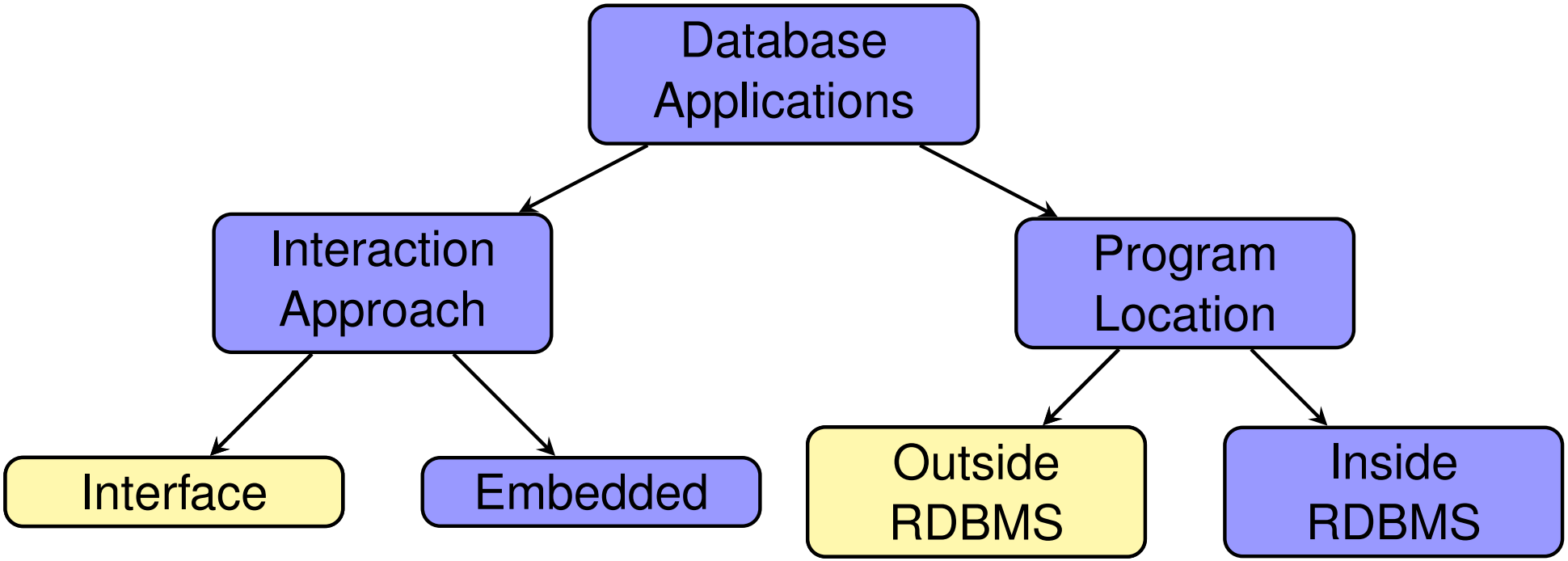
Categorizing Database Applications



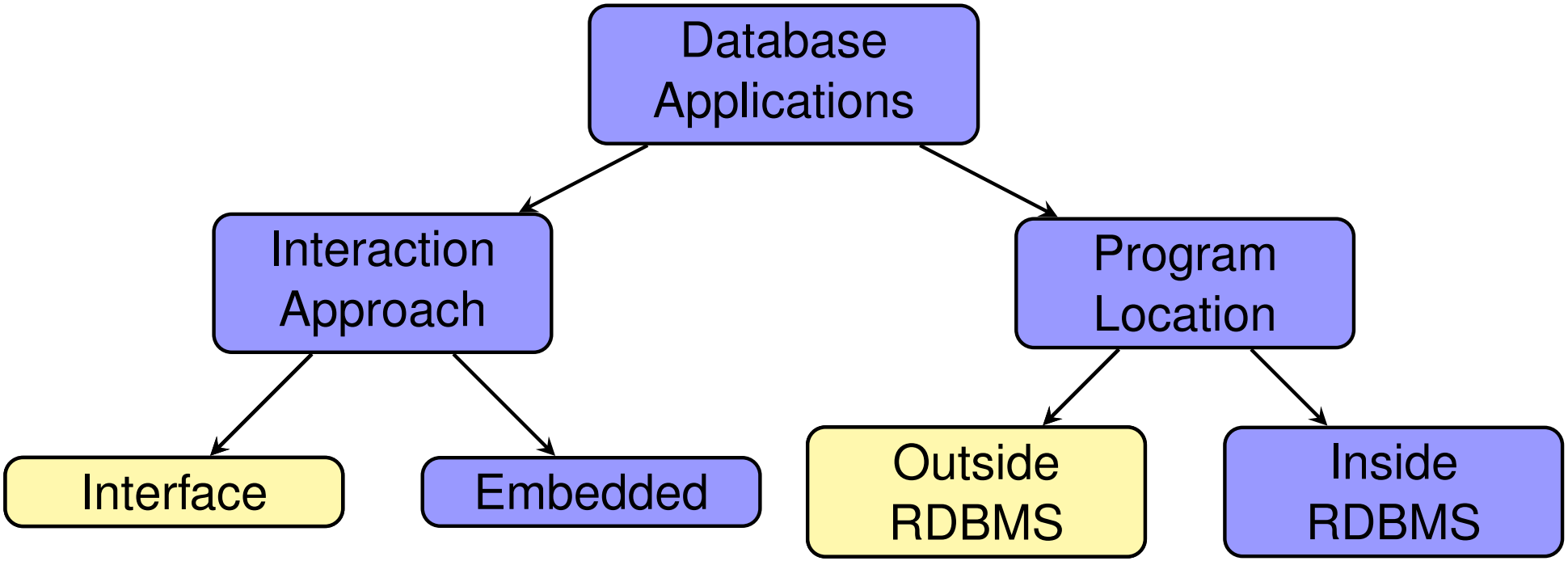
Categorizing Database Applications



Categorizing Database Applications

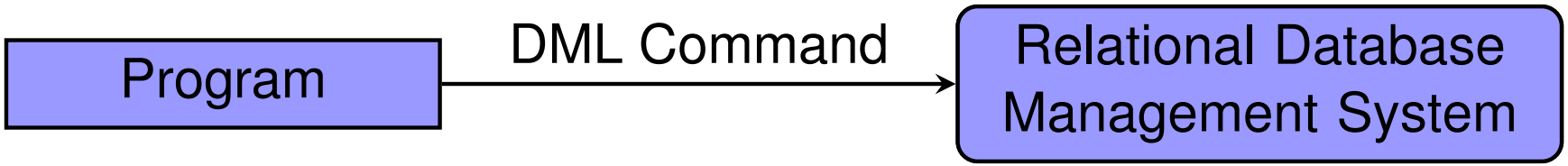


Categorizing Database Applications

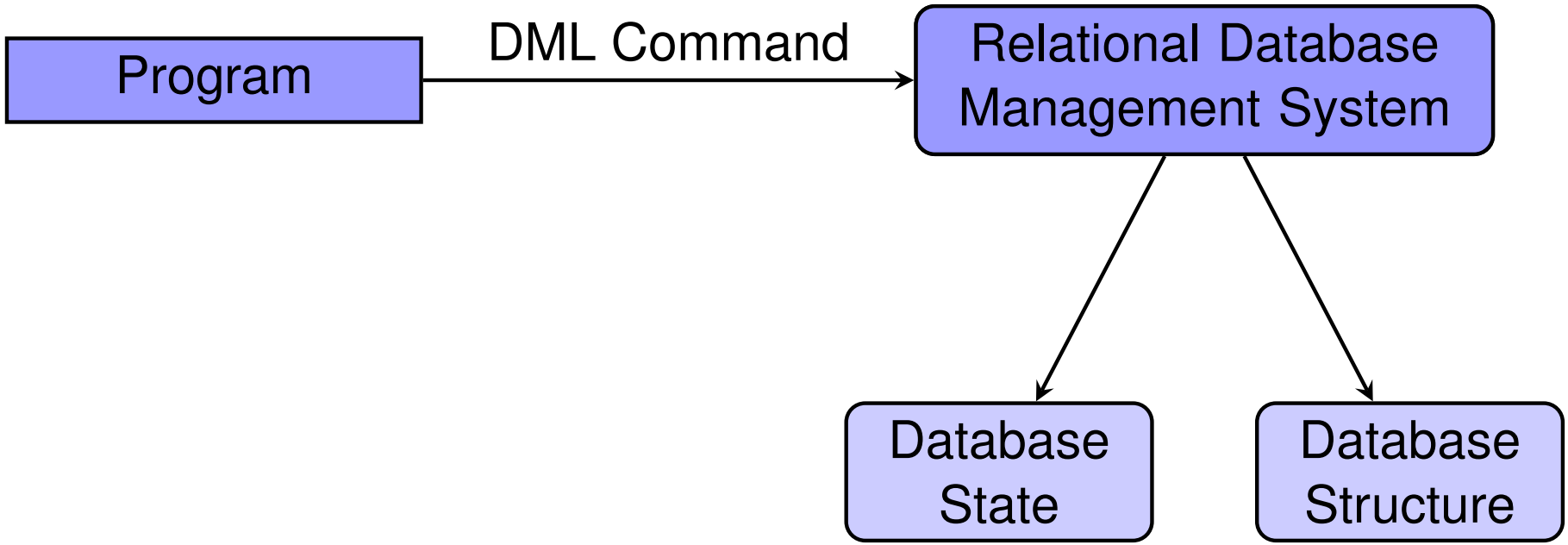


Java application that submits SQL strings to HSQLDB using JDBC

Evolution of Database Applications

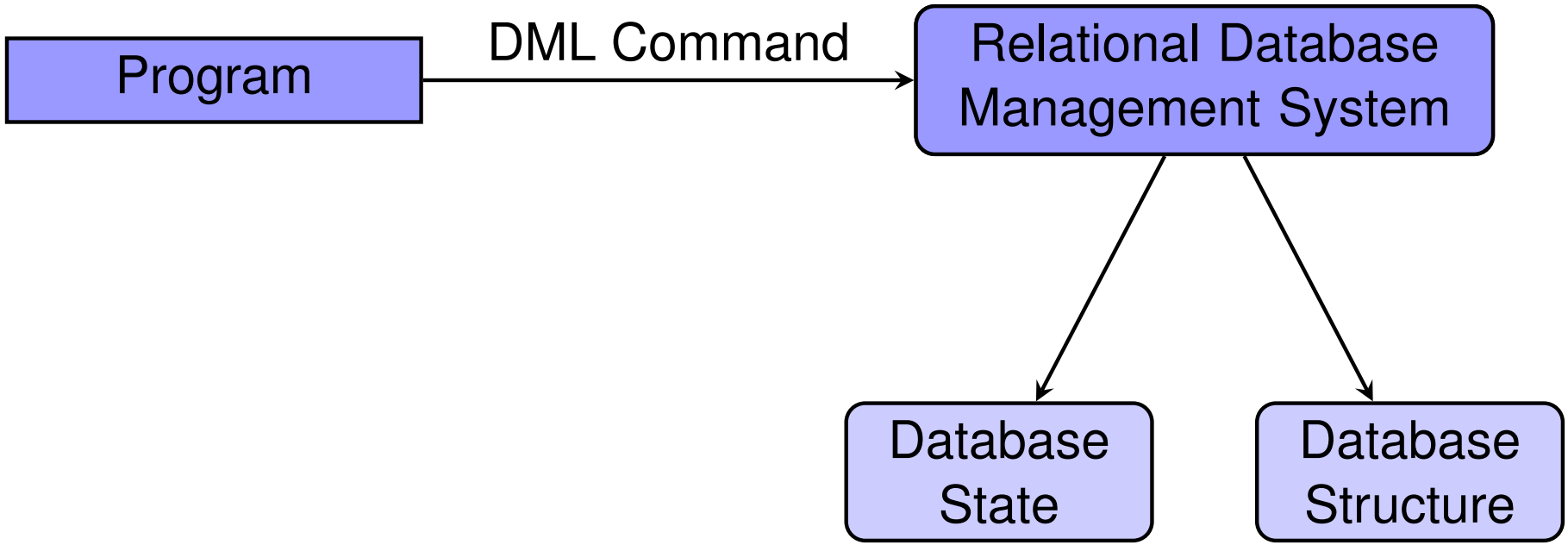


Evolution of Database Applications



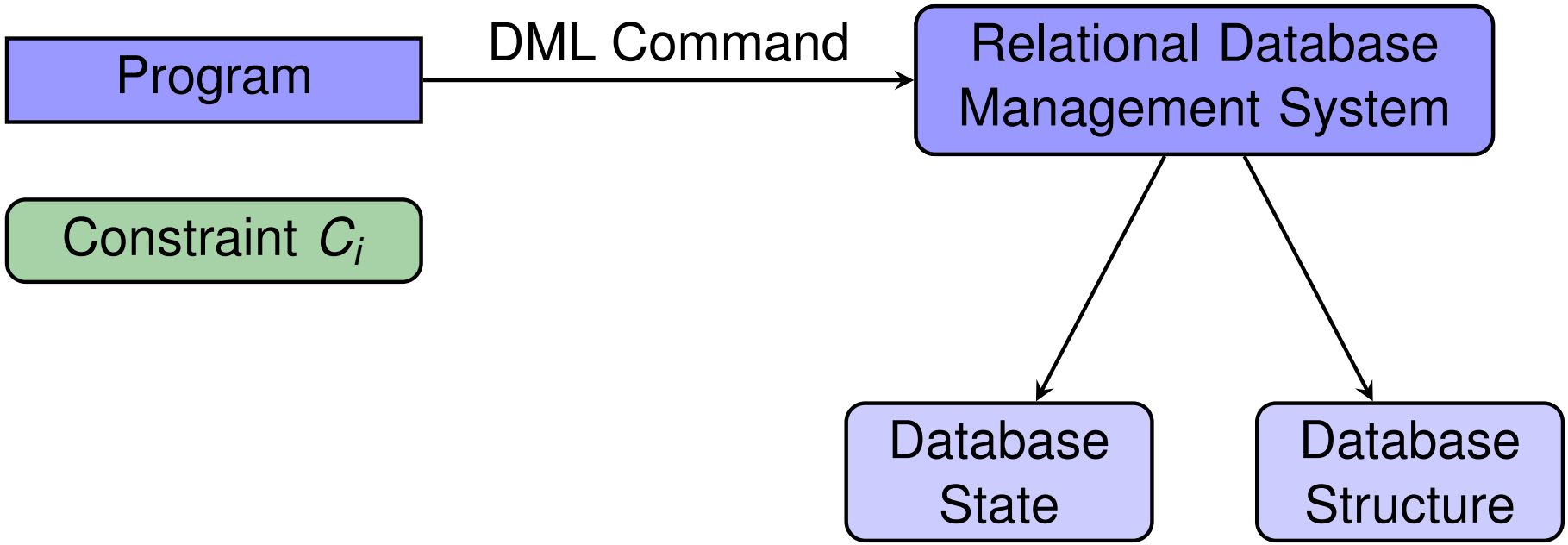
Evolution of Database Applications

Only the database administrator can add new constraints to the schema!



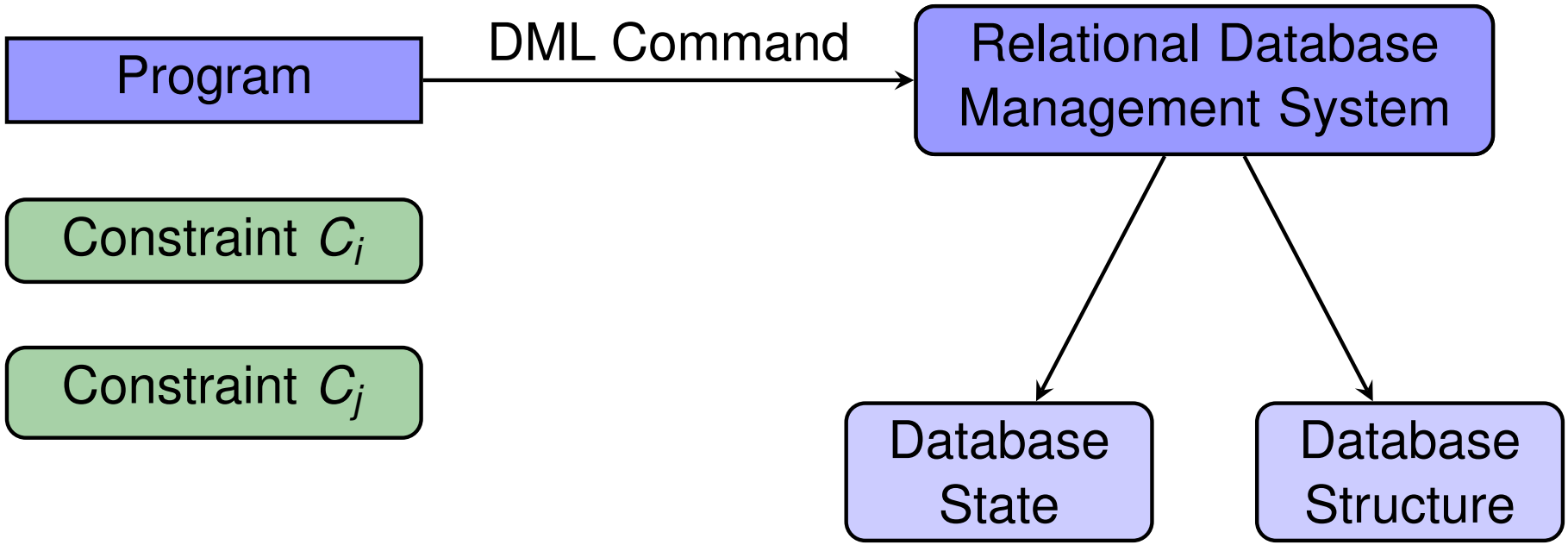
Evolution of Database Applications

The programmers encode the constraints in the program's source code!



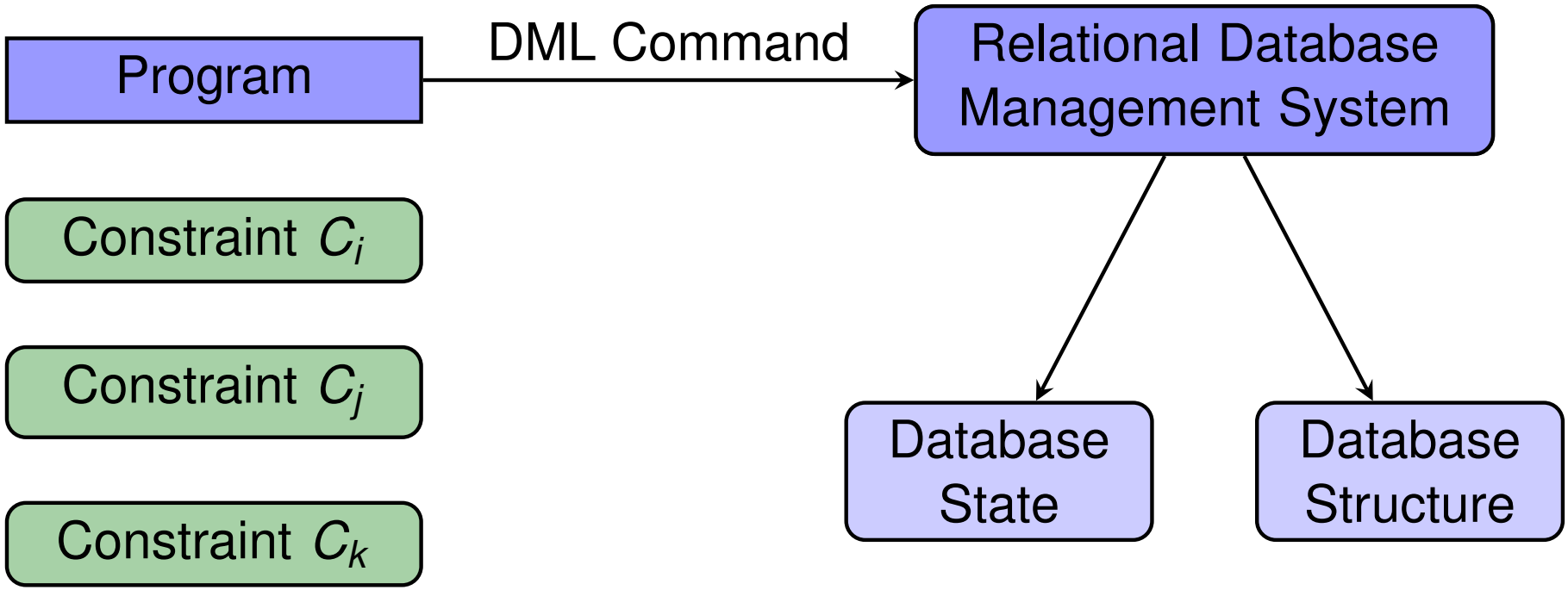
Evolution of Database Applications

The programmers encode the constraints in the program's source code!

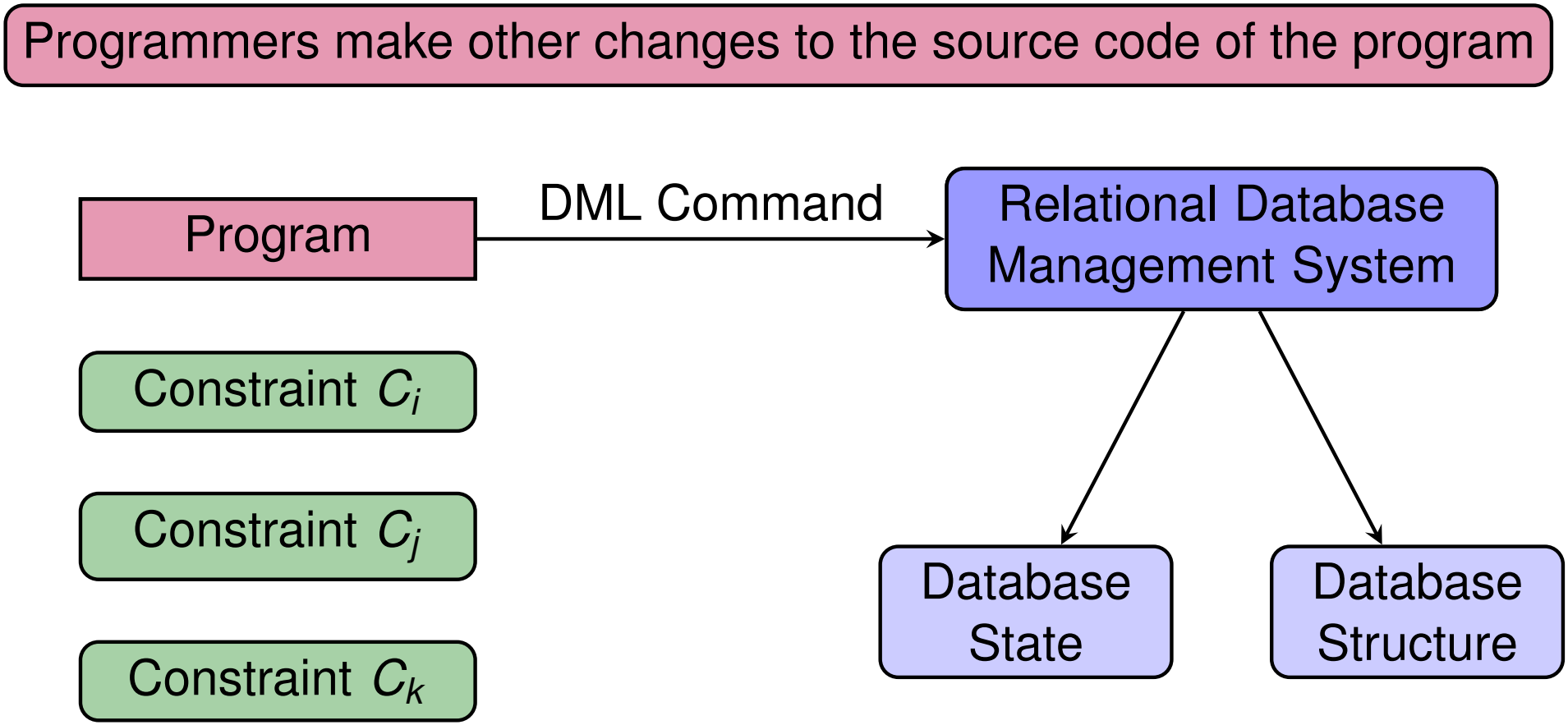


Evolution of Database Applications

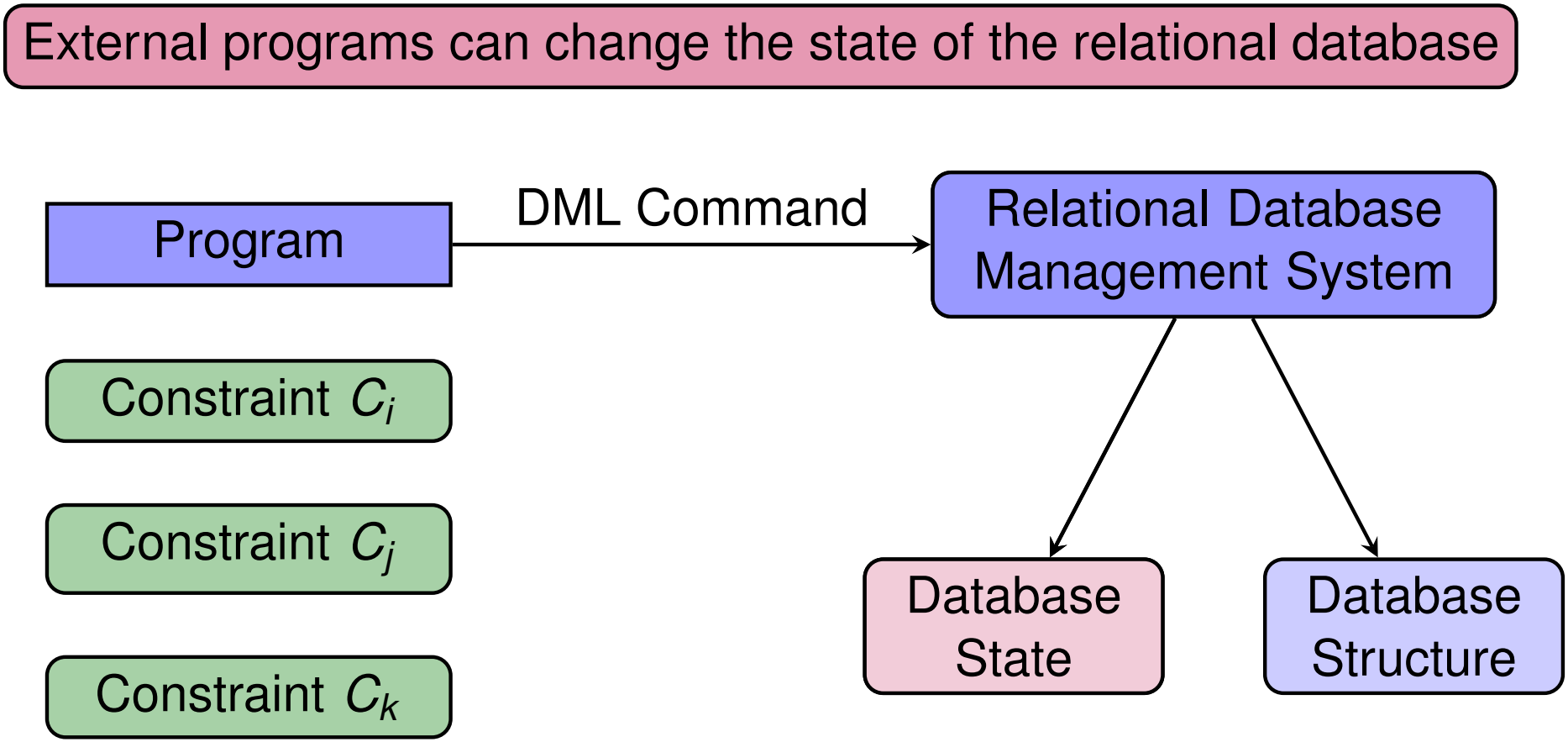
The programmers encode the constraints in the program's source code!



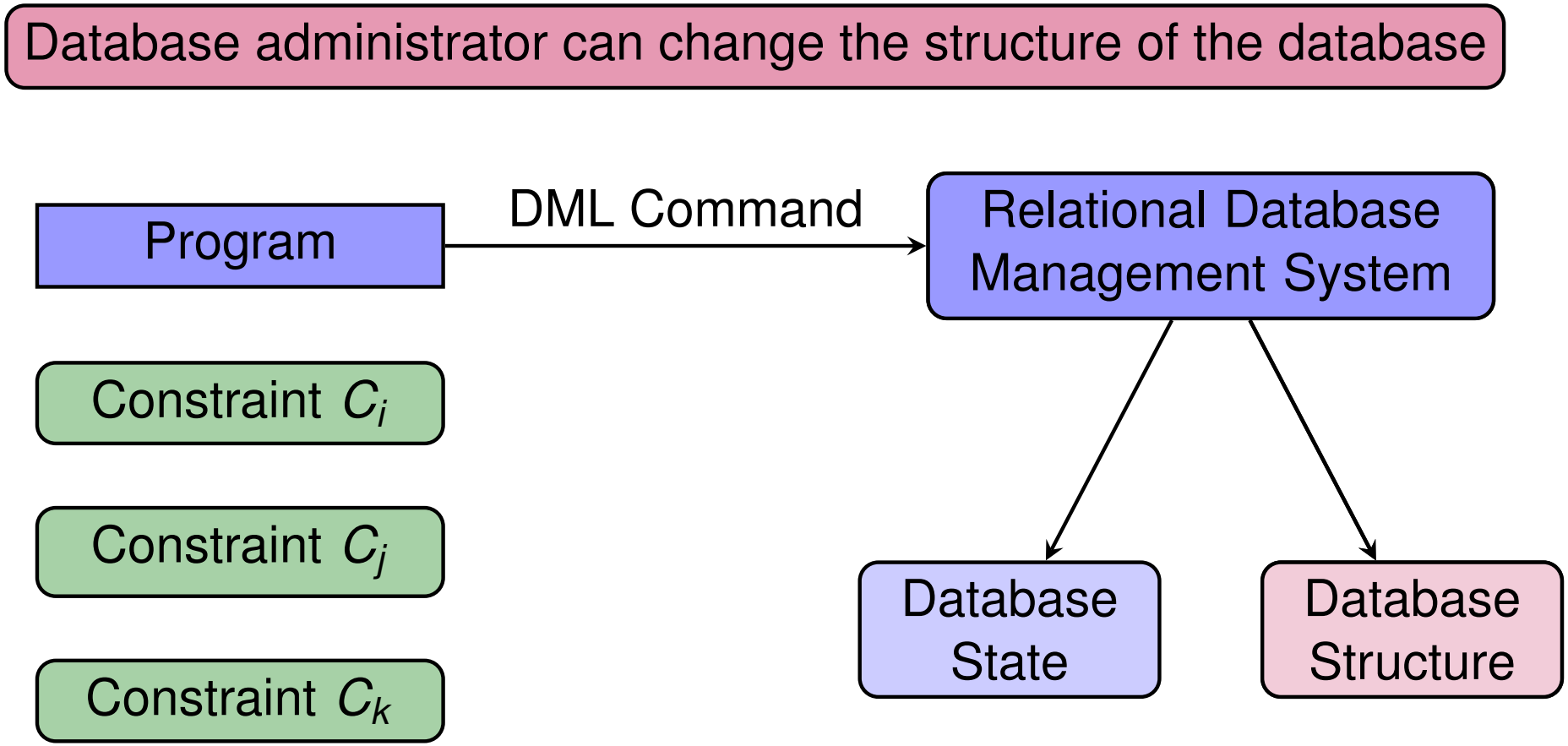
Evolution of Database Applications



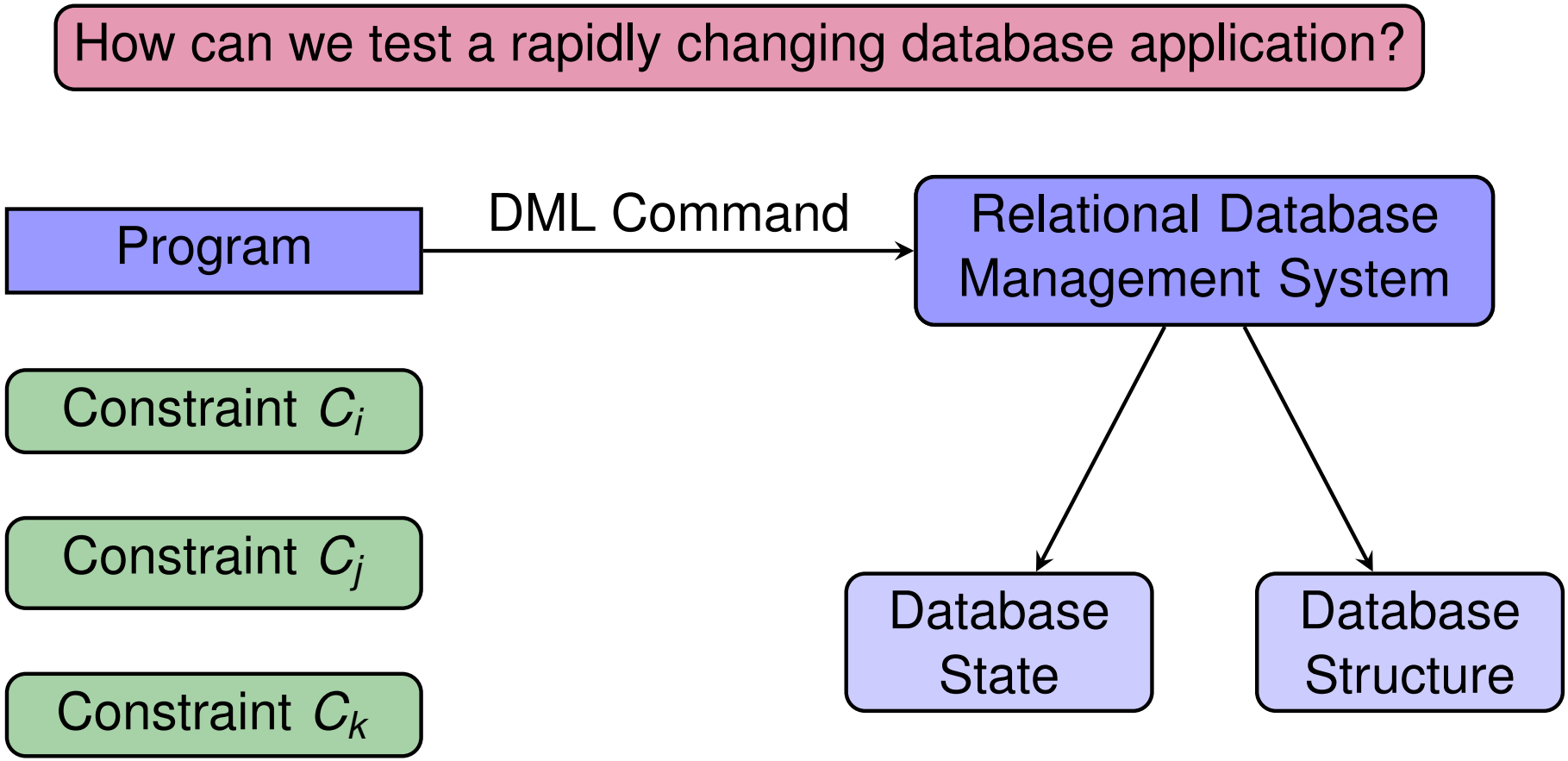
Evolution of Database Applications



Evolution of Database Applications

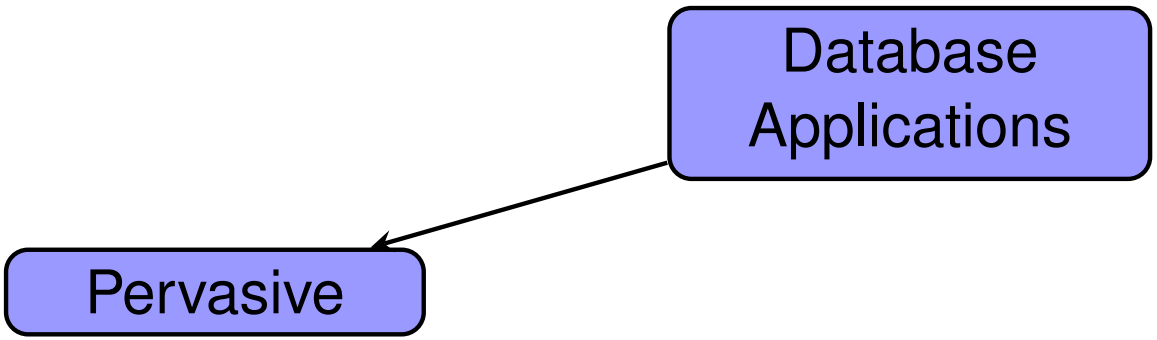


Evolution of Database Applications

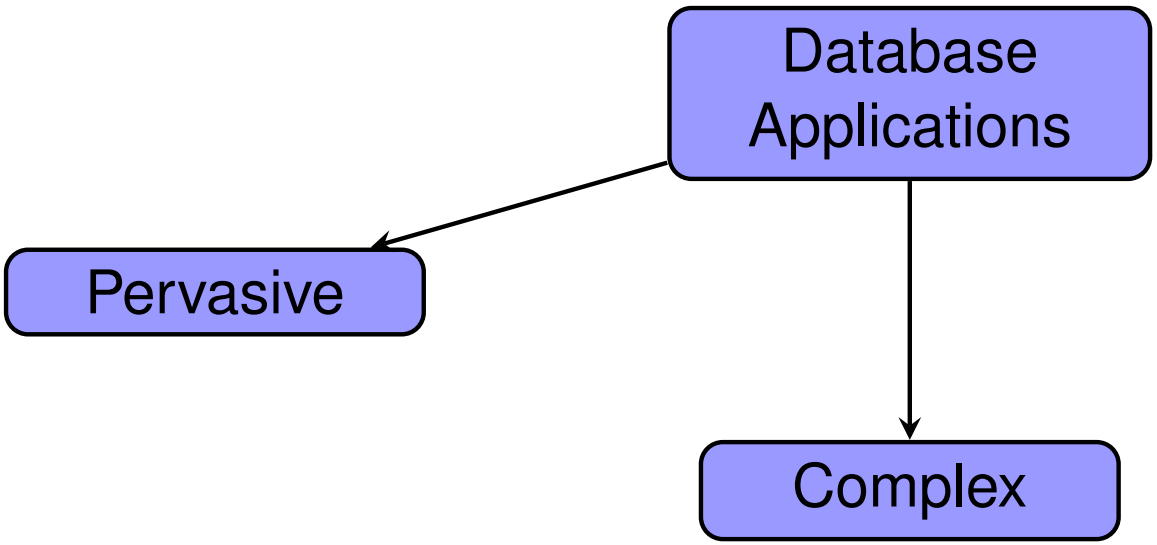


Regression Testing to the Rescue

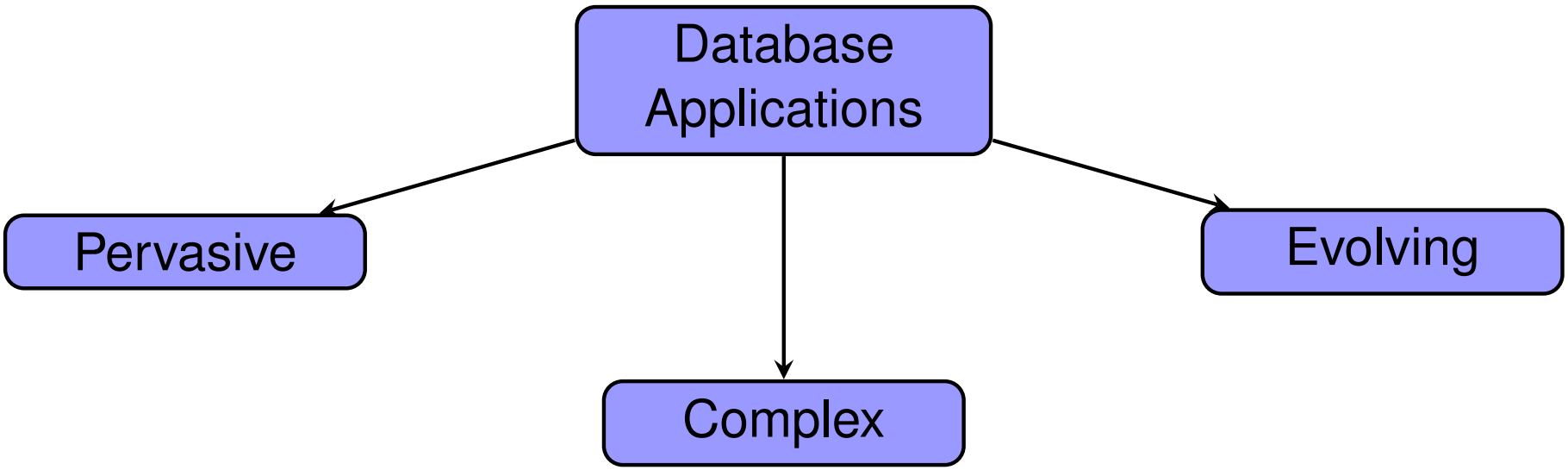
Regression Testing to the Rescue



Regression Testing to the Rescue

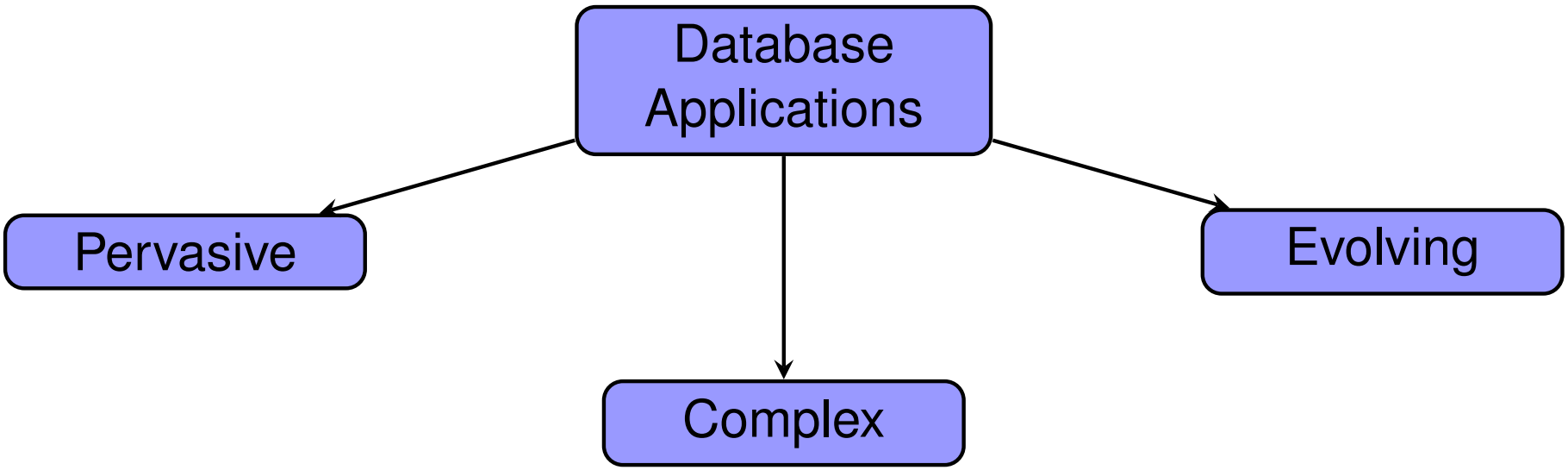


Regression Testing to the Rescue

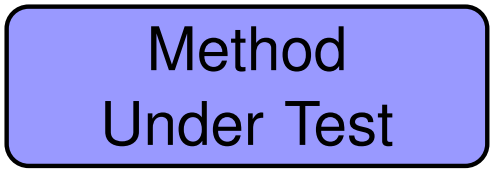


Regression Testing to the Rescue

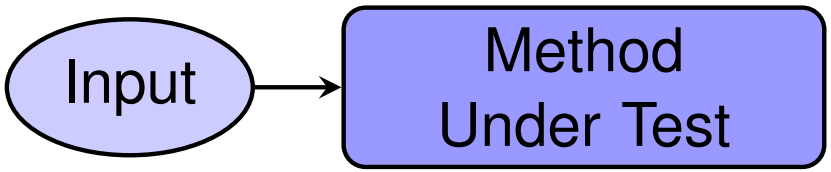
Regression Testing supports the efficient construction of database software that is complex and rapidly evolving



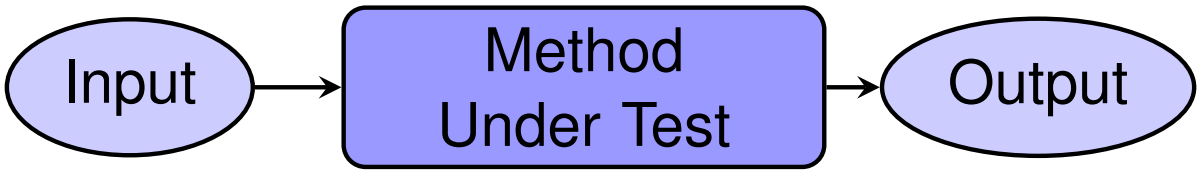
What is a Test Case?



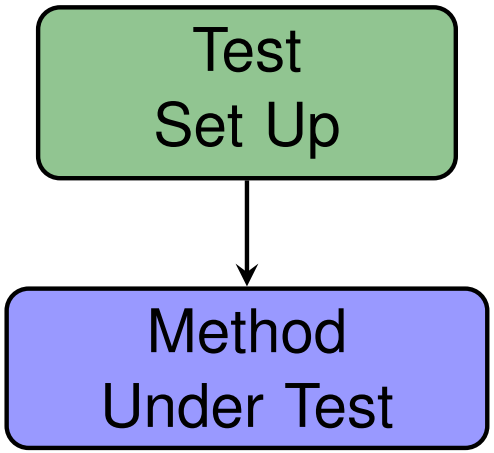
What is a Test Case?



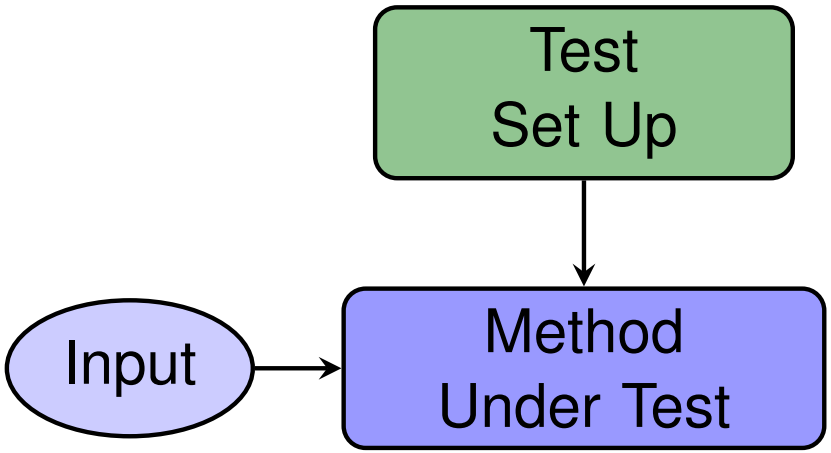
What is a Test Case?



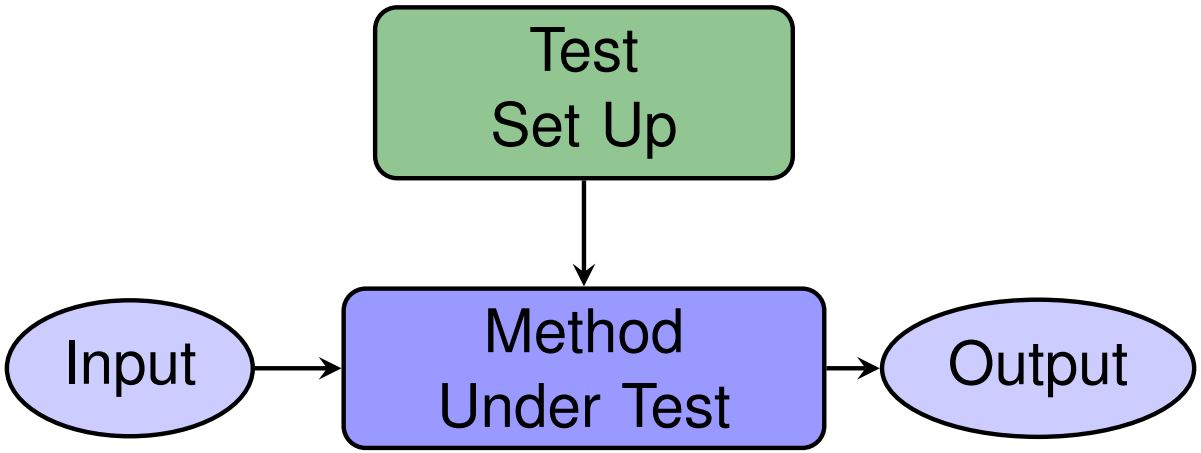
What is a Test Case?



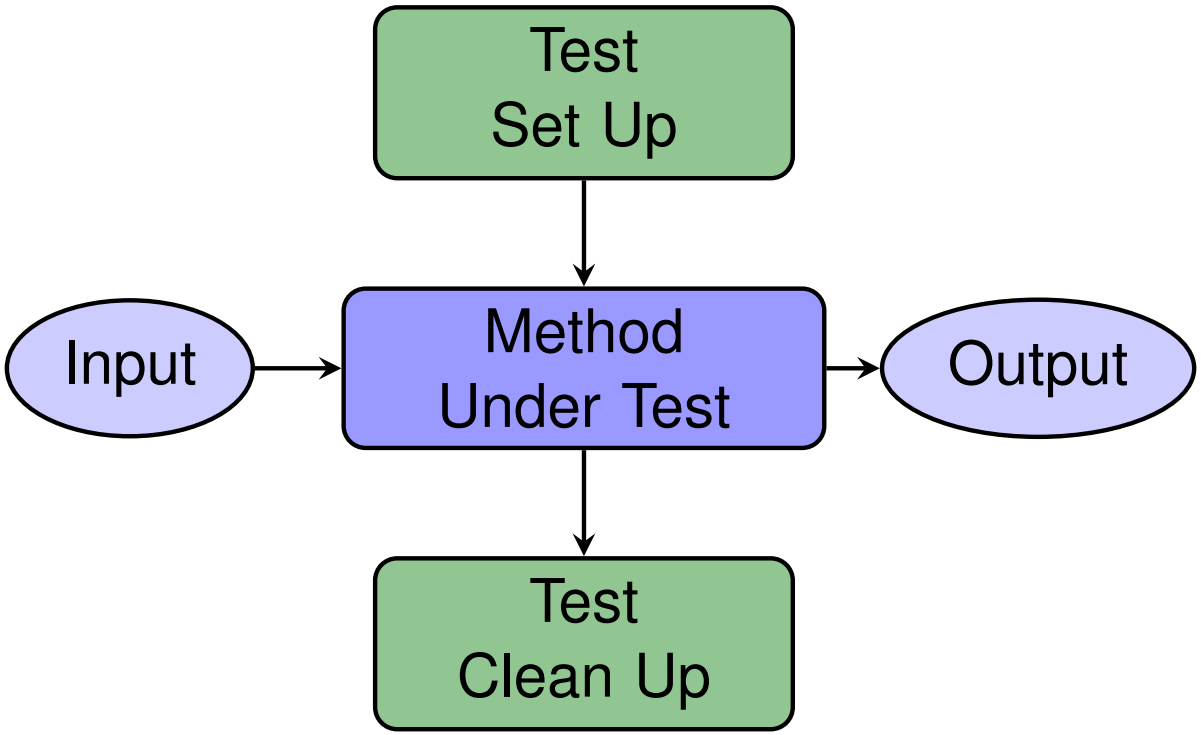
What is a Test Case?



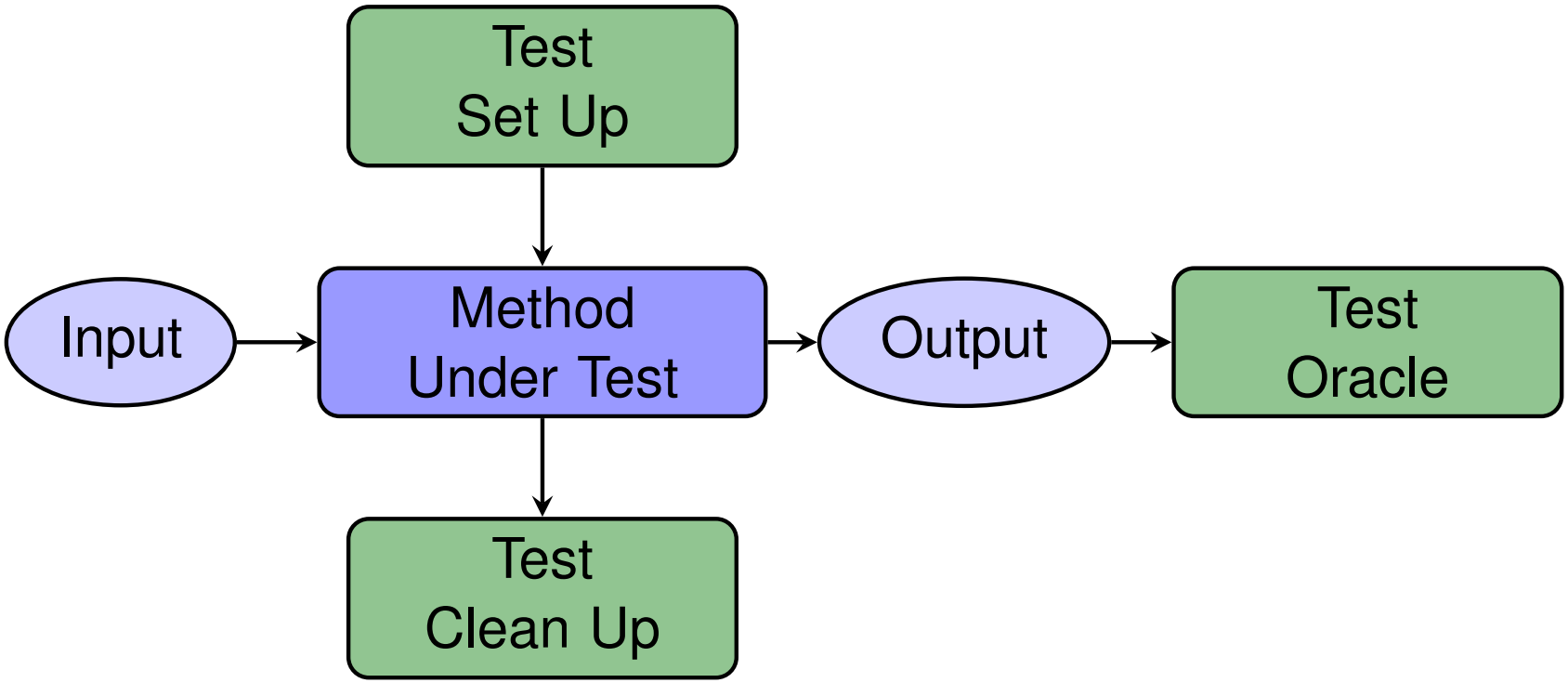
What is a Test Case?



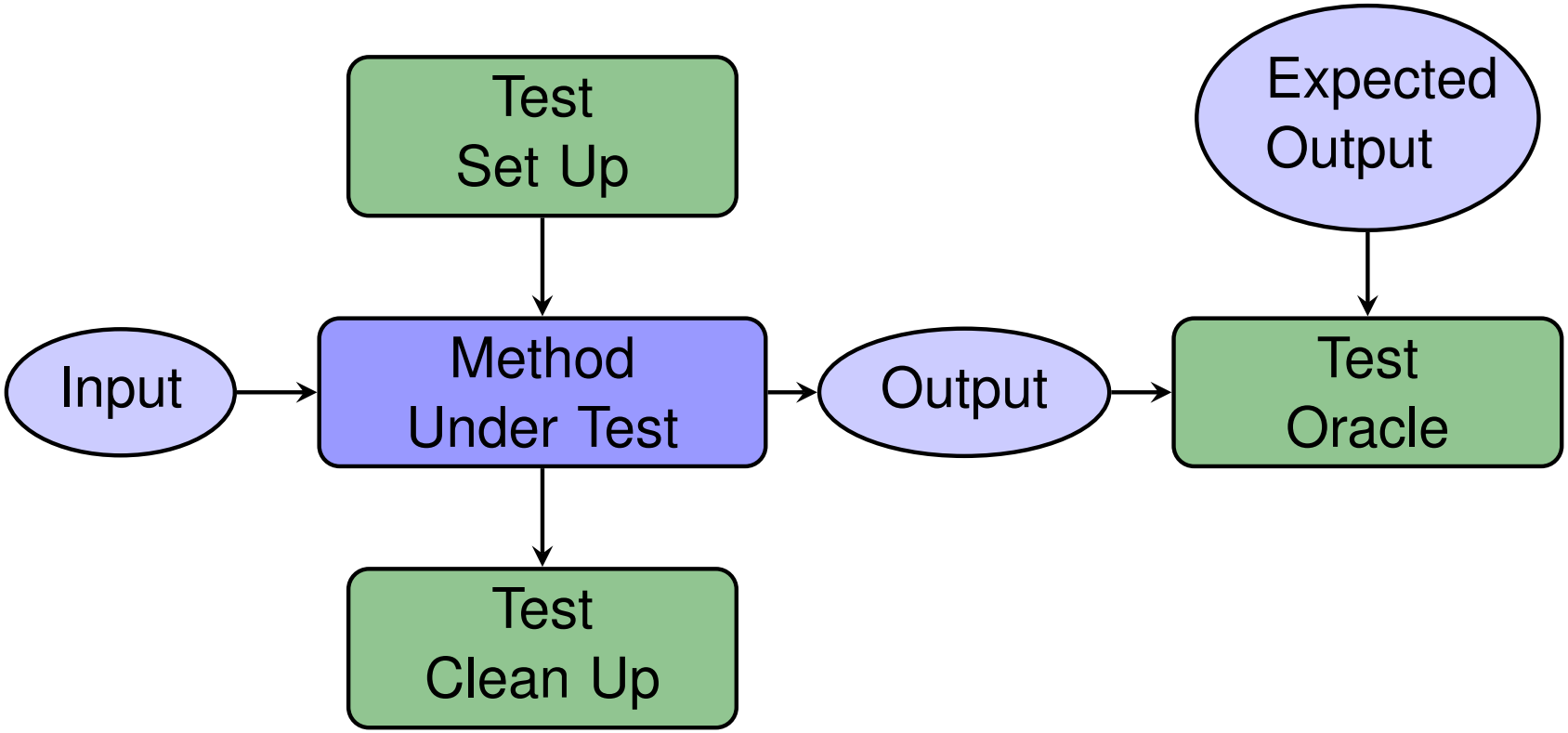
What is a Test Case?



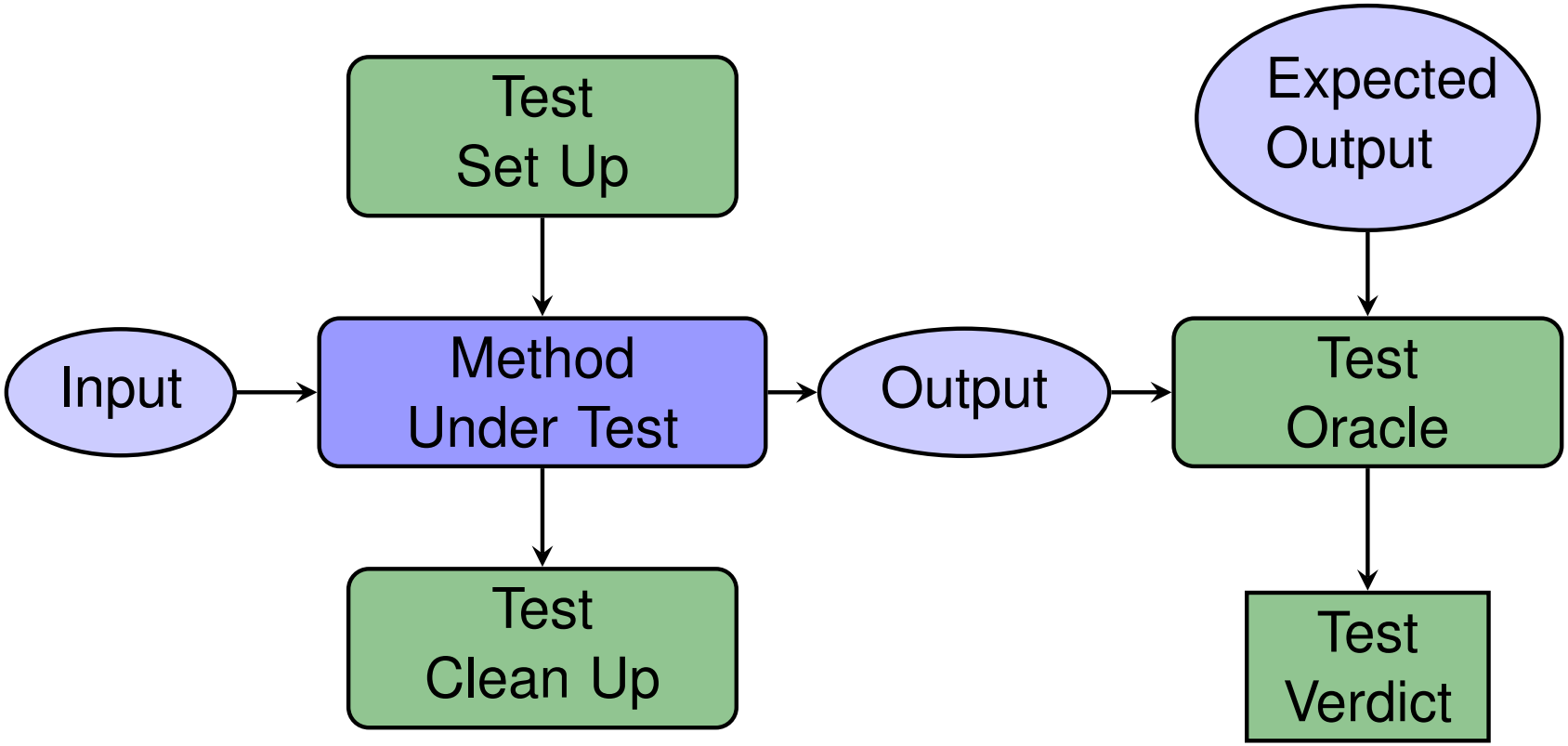
What is a Test Case?



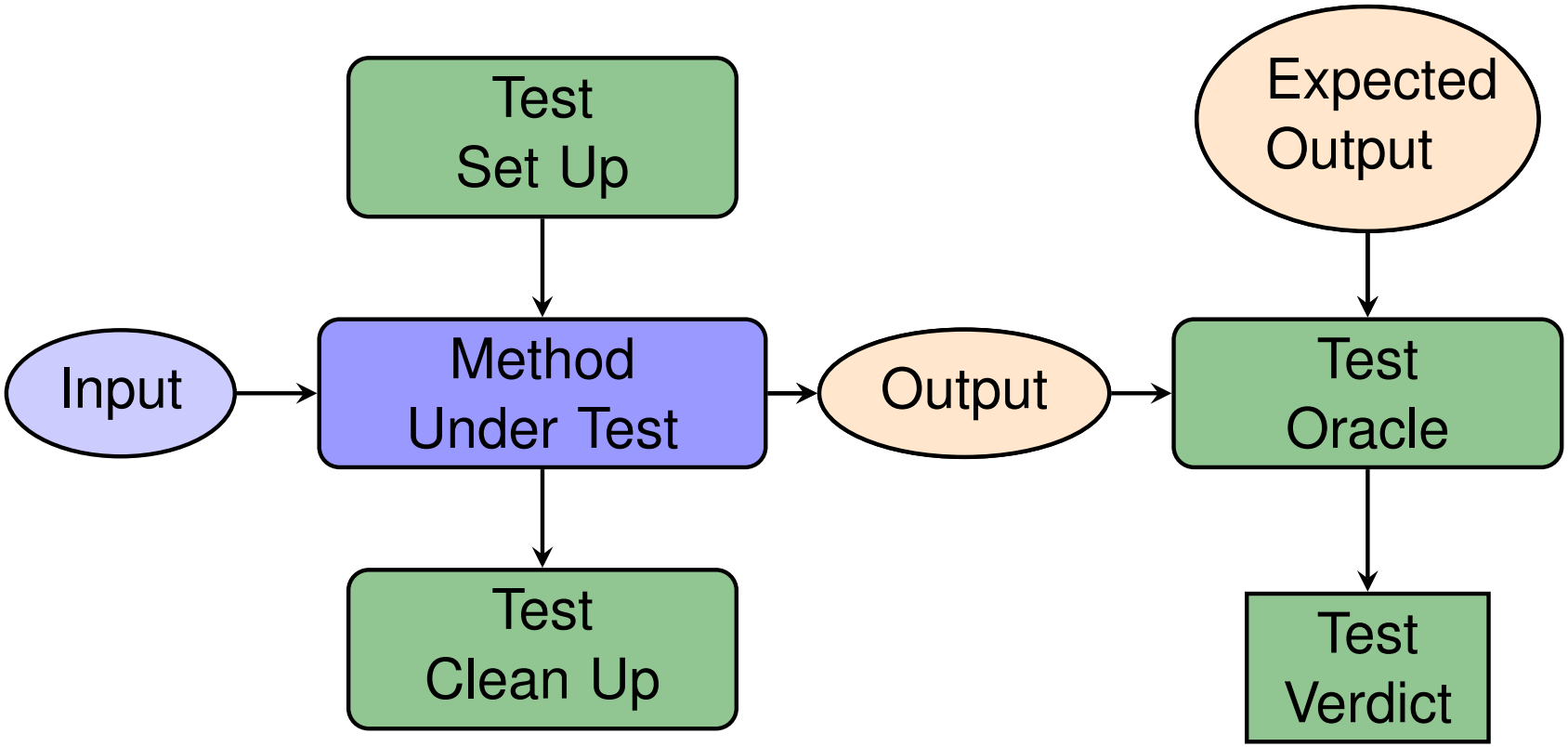
What is a Test Case?



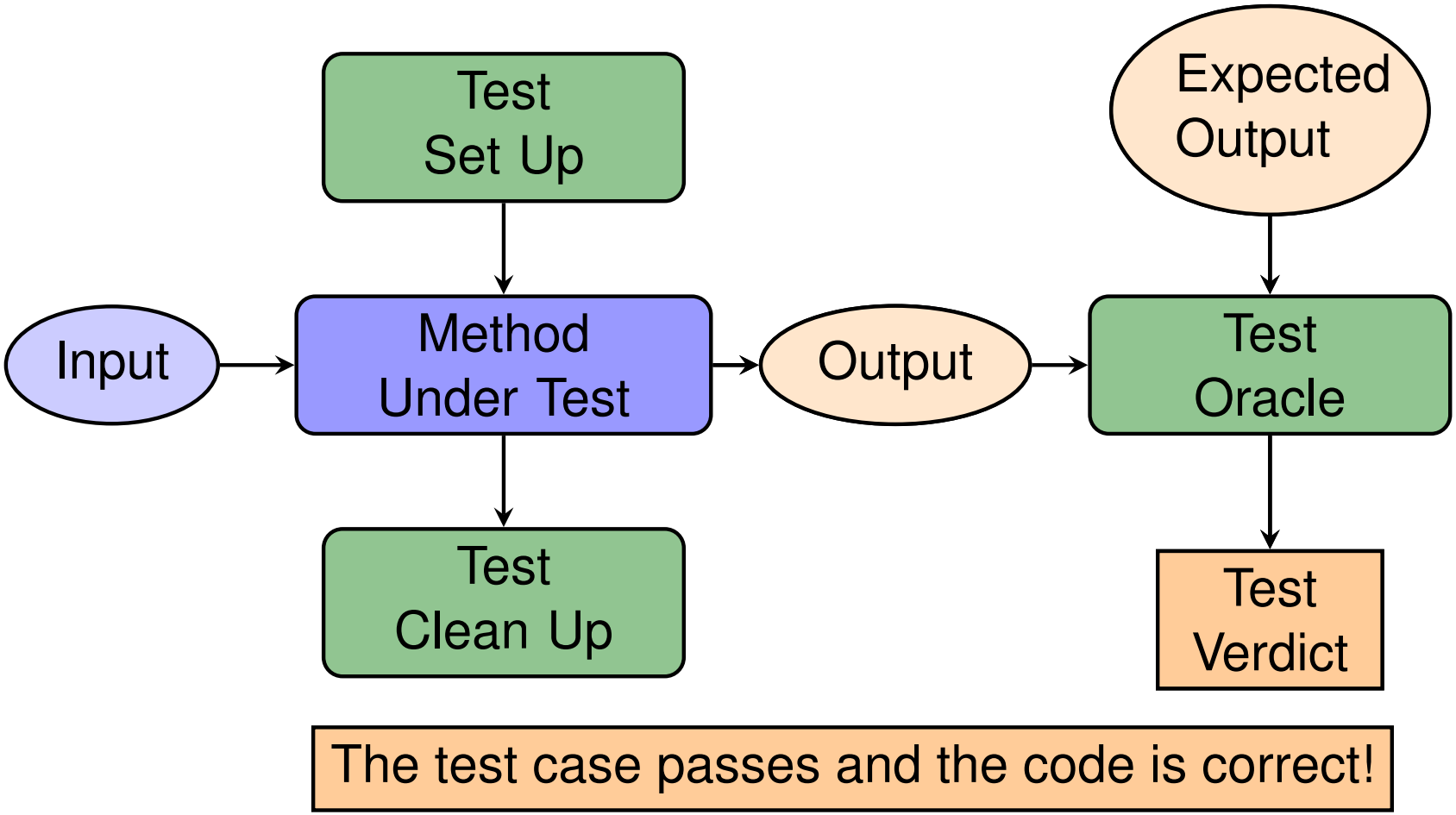
What is a Test Case?



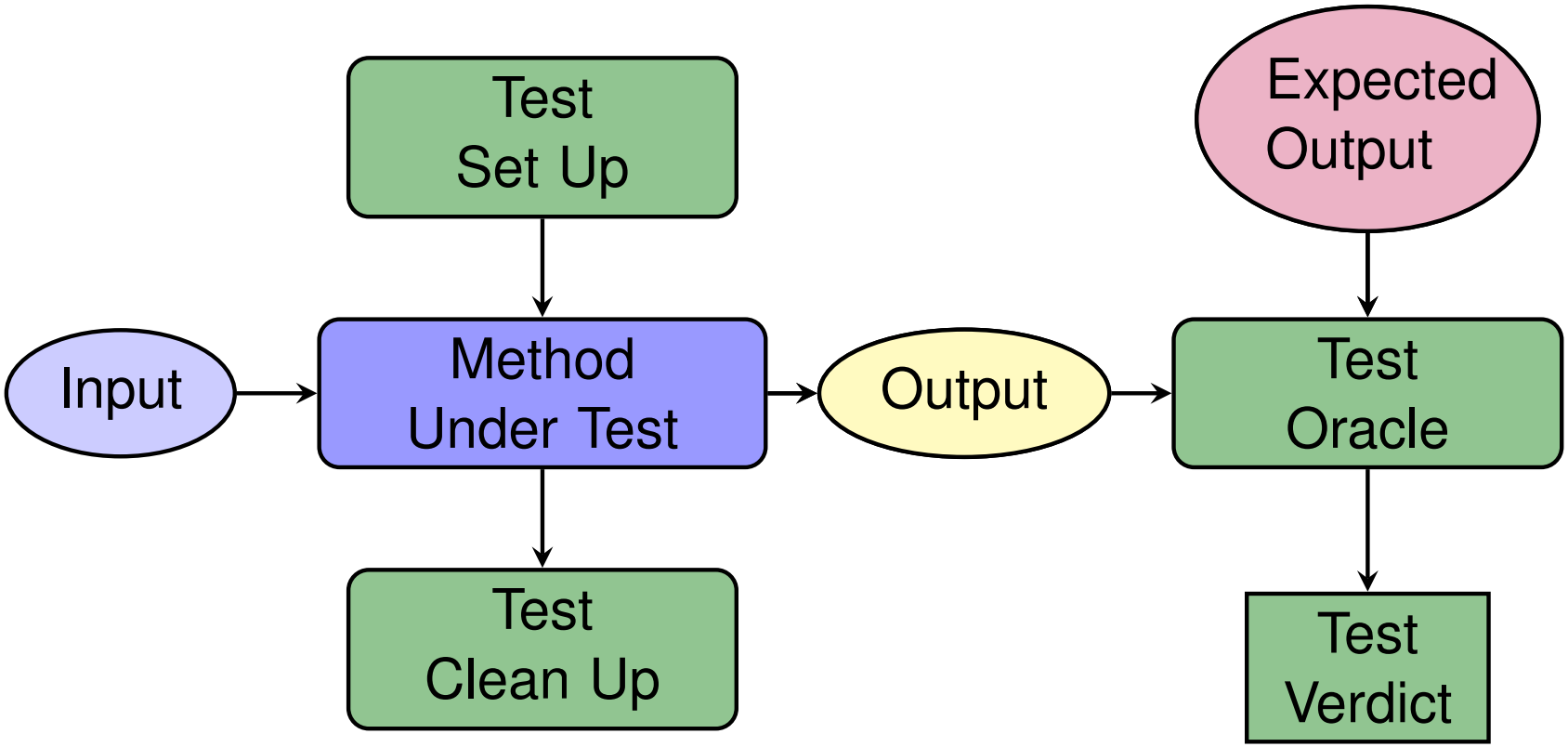
What is a Test Case?



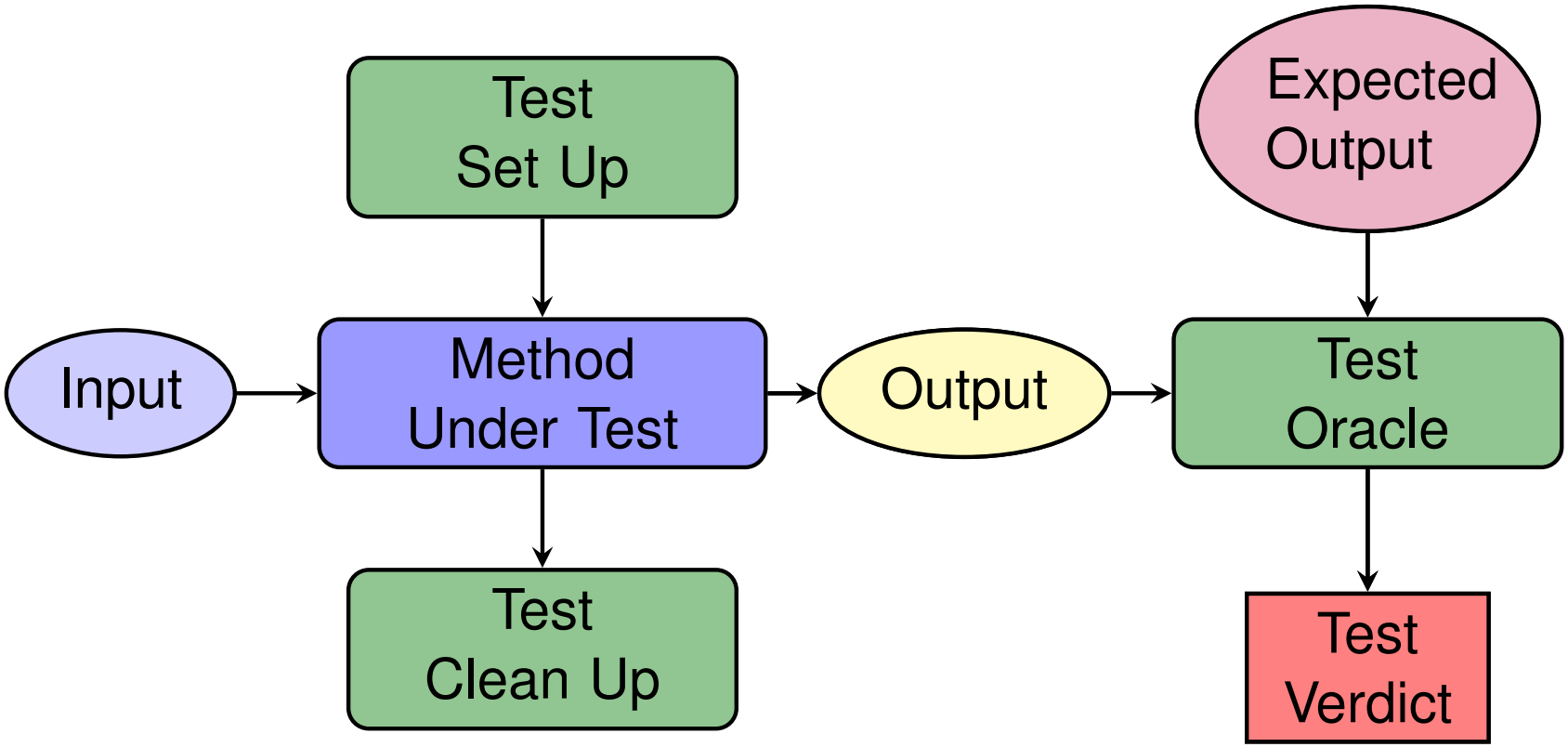
What is a Test Case?



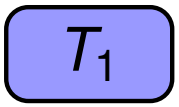
What is a Test Case?



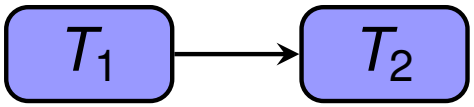
What is a Test Case?



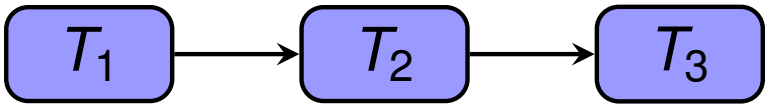
What is a Test Suite?



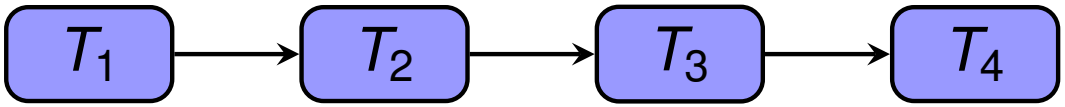
What is a Test Suite?



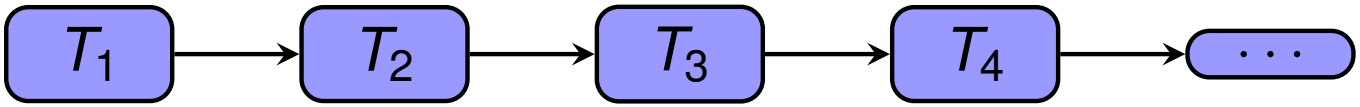
What is a Test Suite?



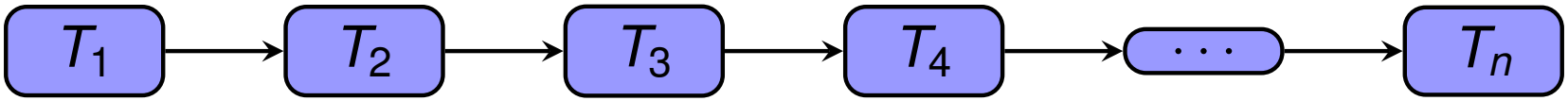
What is a Test Suite?



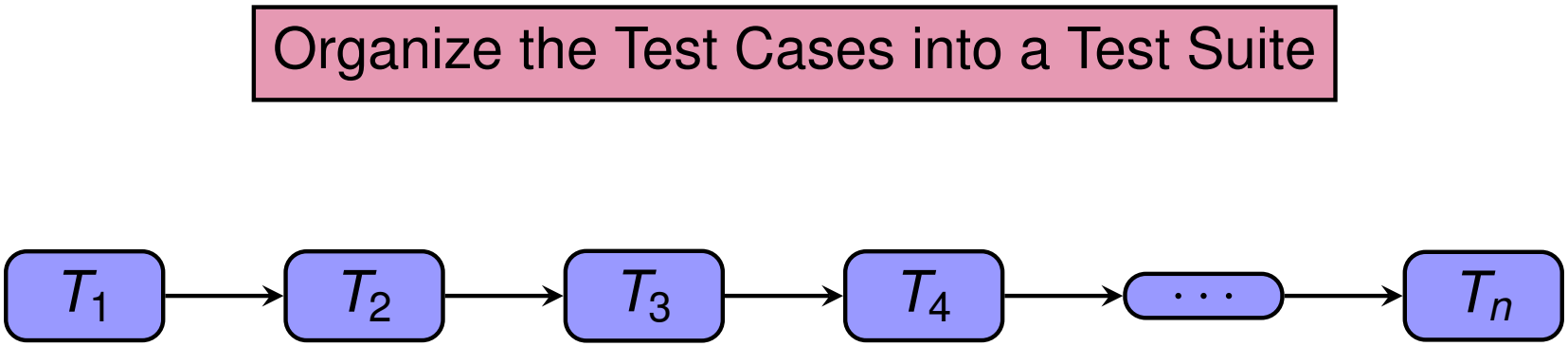
What is a Test Suite?



What is a Test Suite?

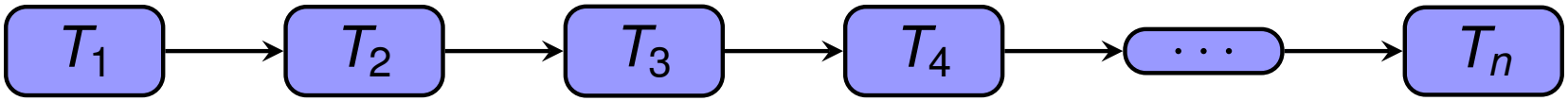


What is a Test Suite?



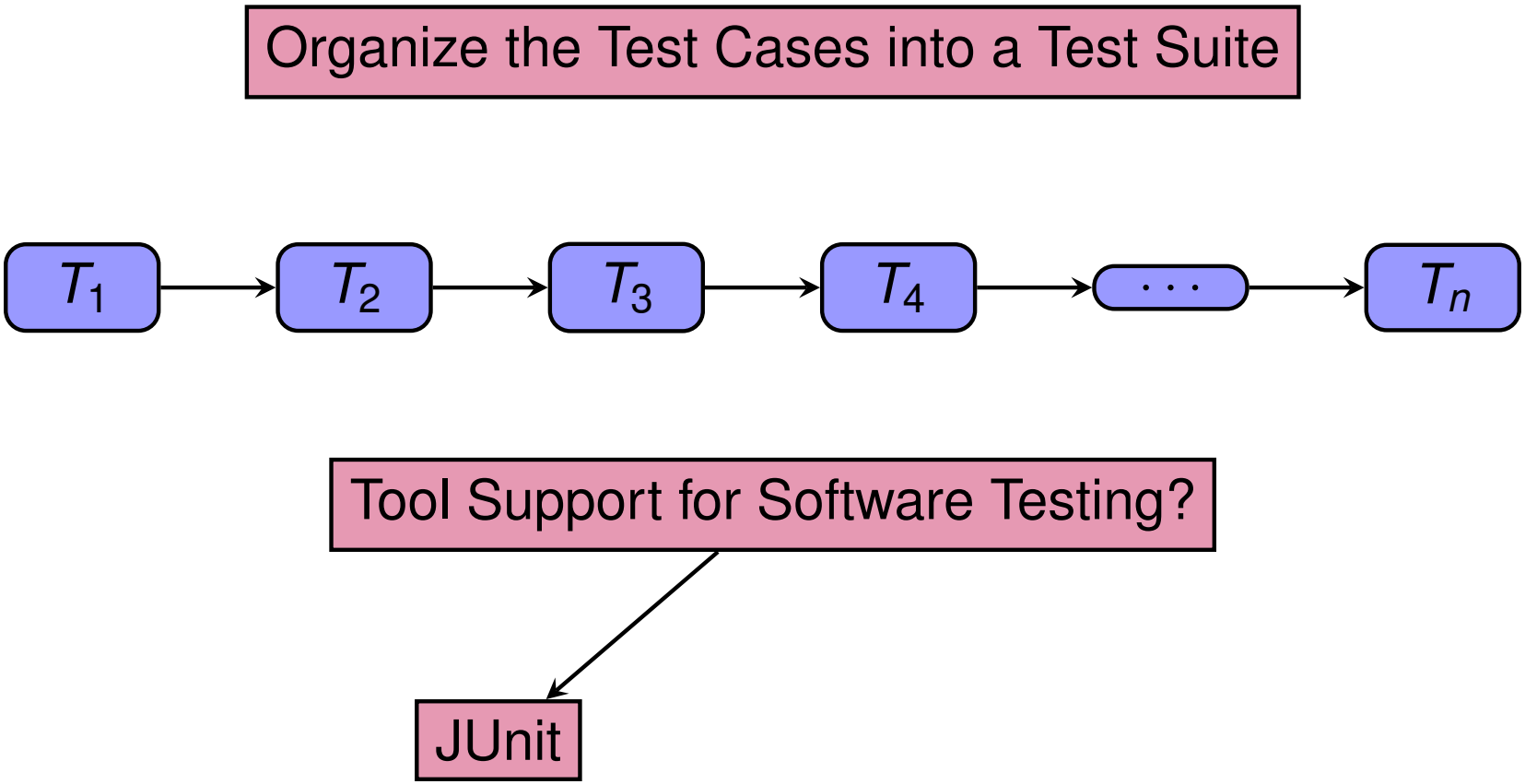
What is a Test Suite?

Organize the Test Cases into a Test Suite

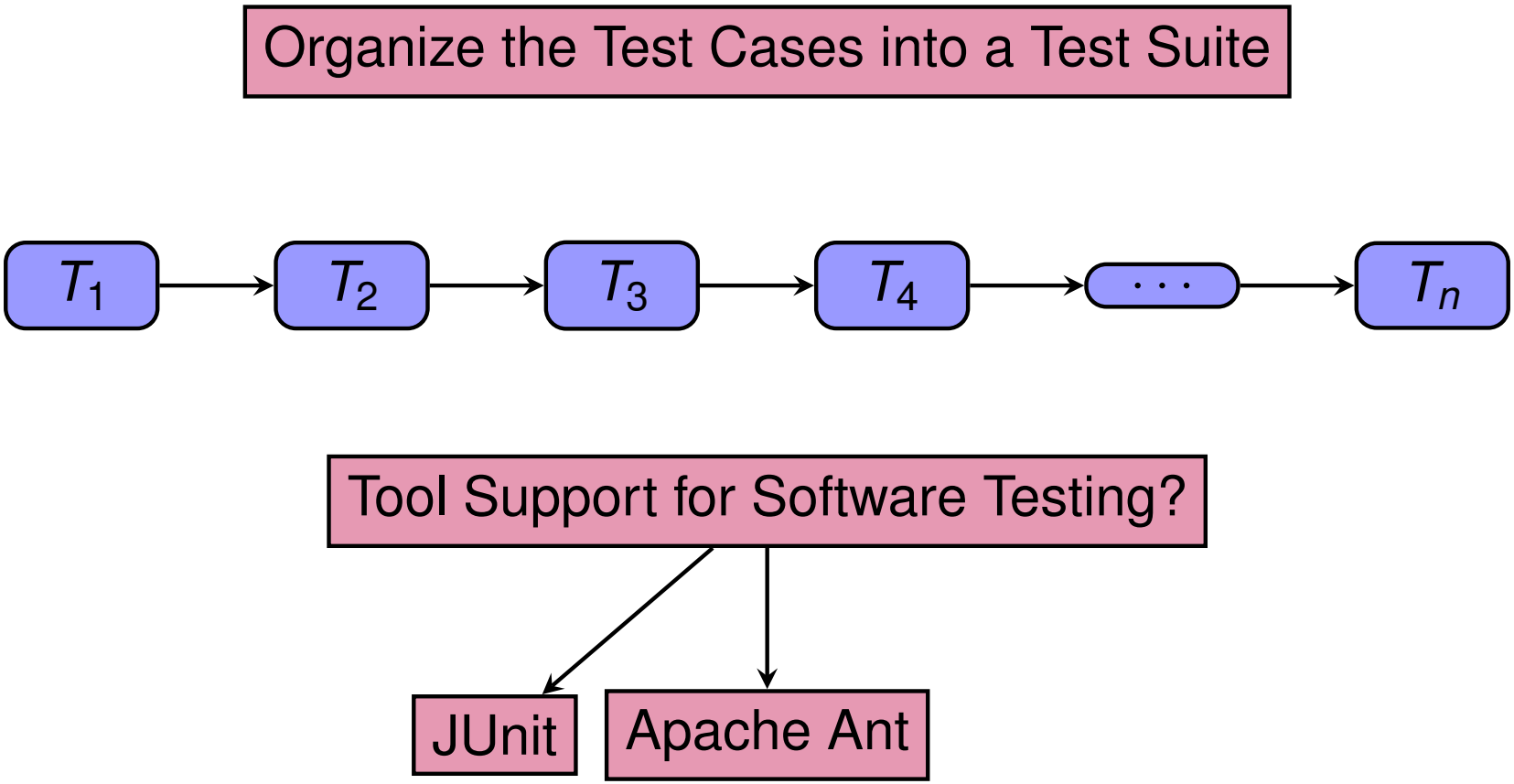


Tool Support for Software Testing?

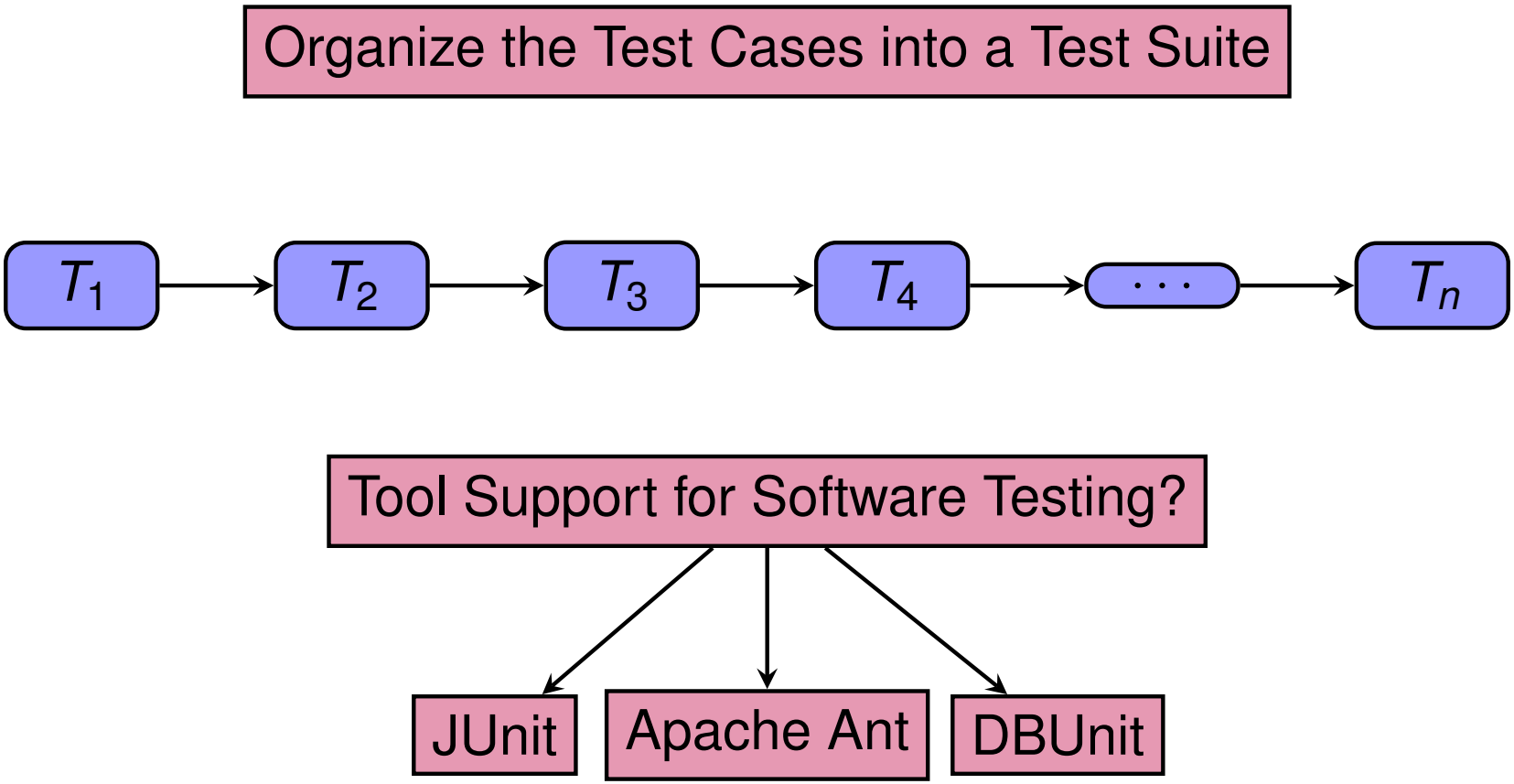
What is a Test Suite?



What is a Test Suite?

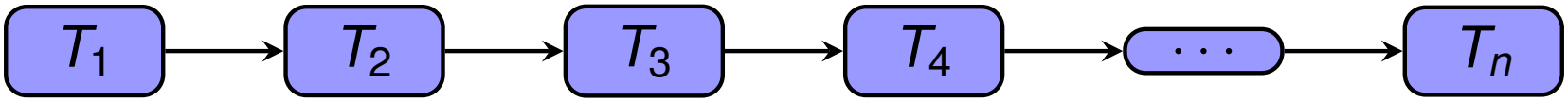


What is a Test Suite?

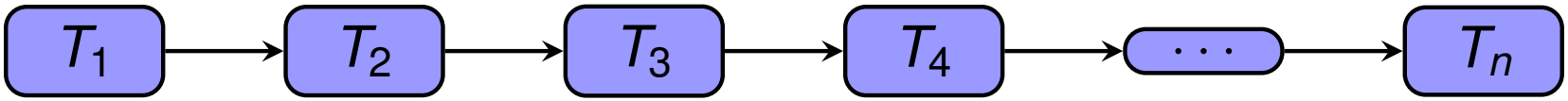


Test Suite Management

Organize the Test Cases into a Test Suite



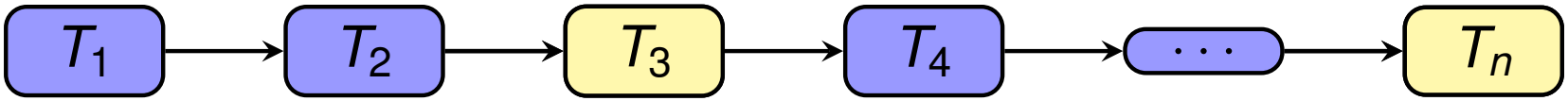
Test Suite Management



Regression Testing Technique

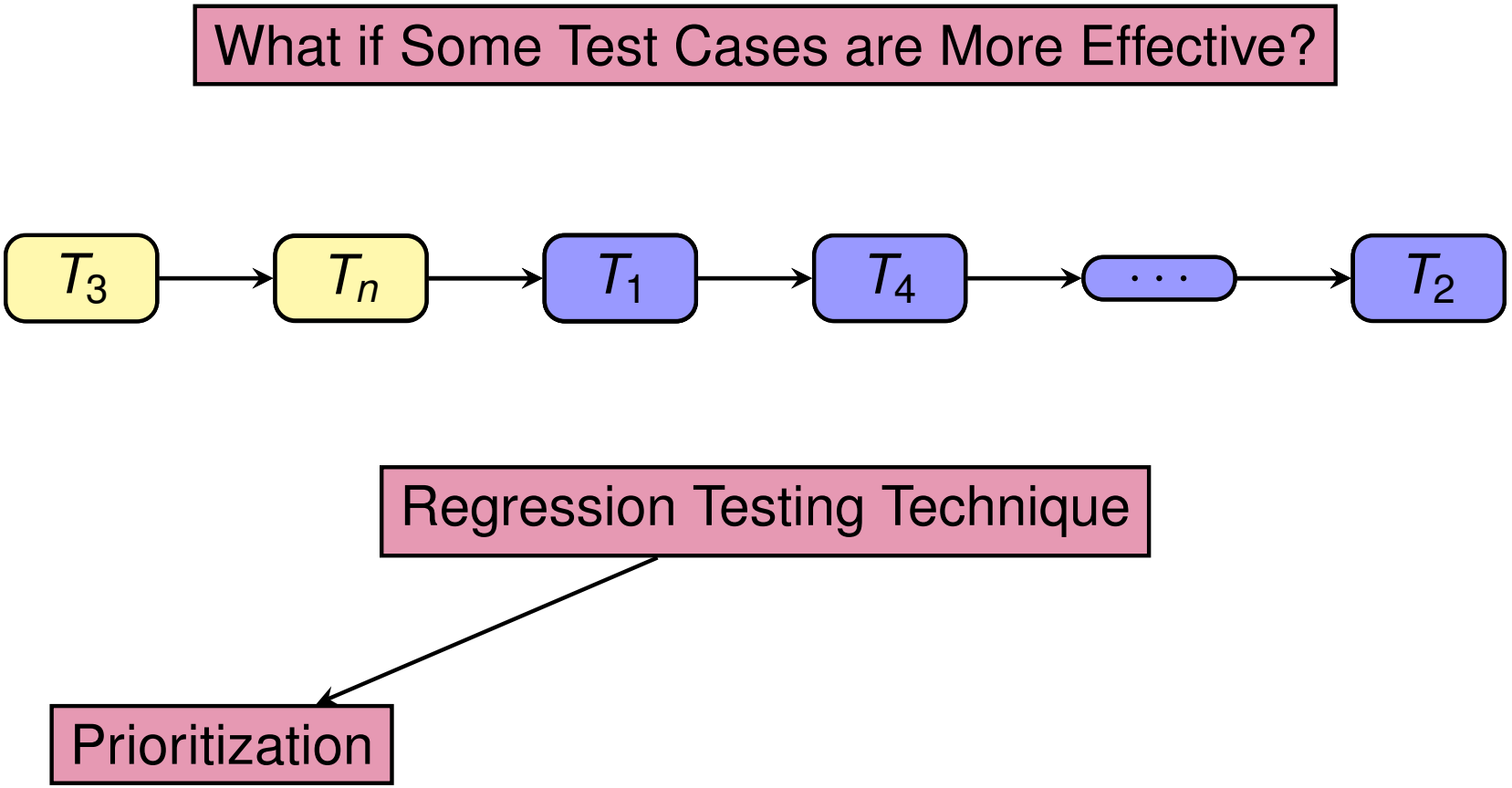
Test Suite Management

What if Some Test Cases are More Effective?

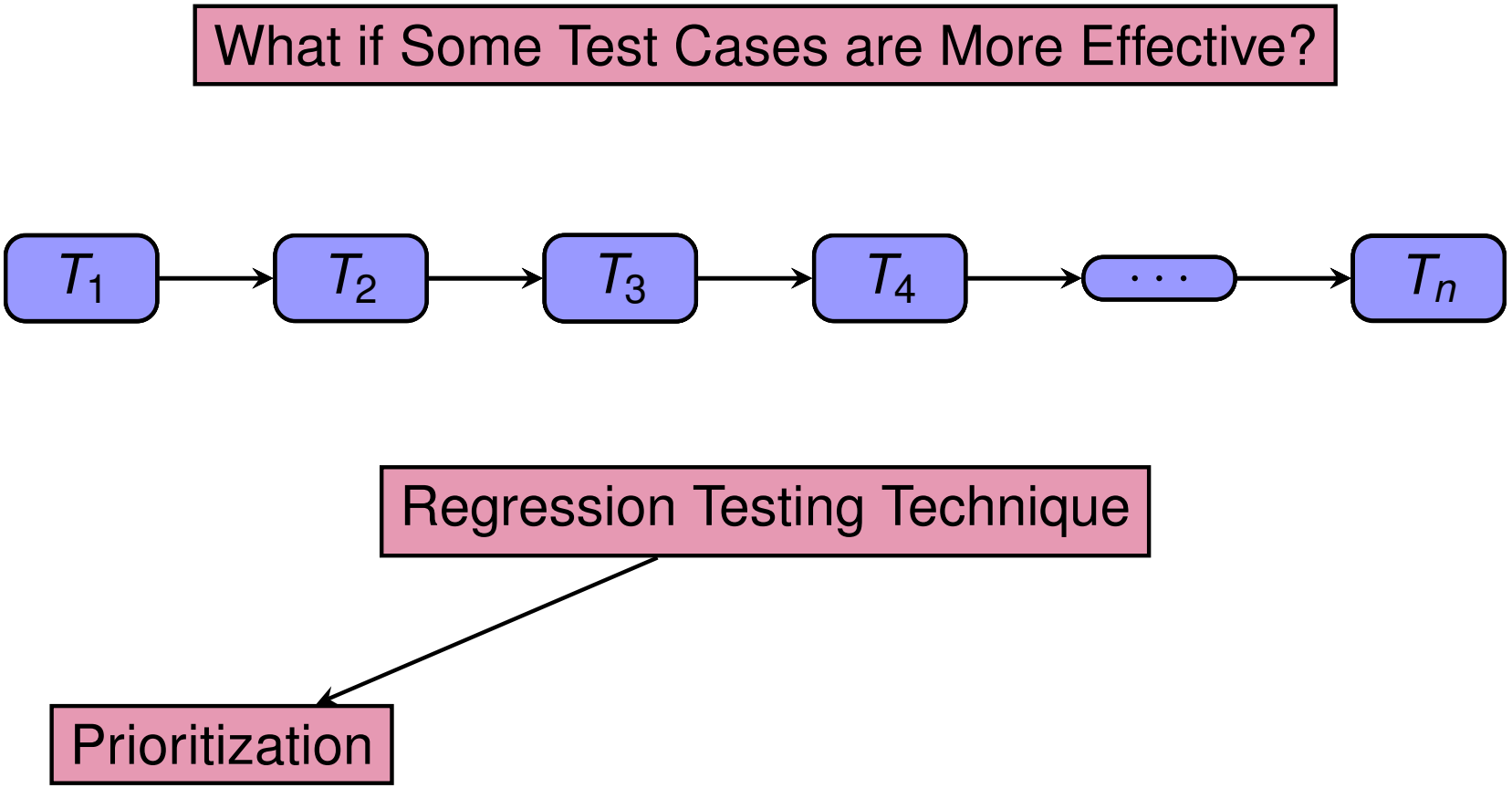


Regression Testing Technique

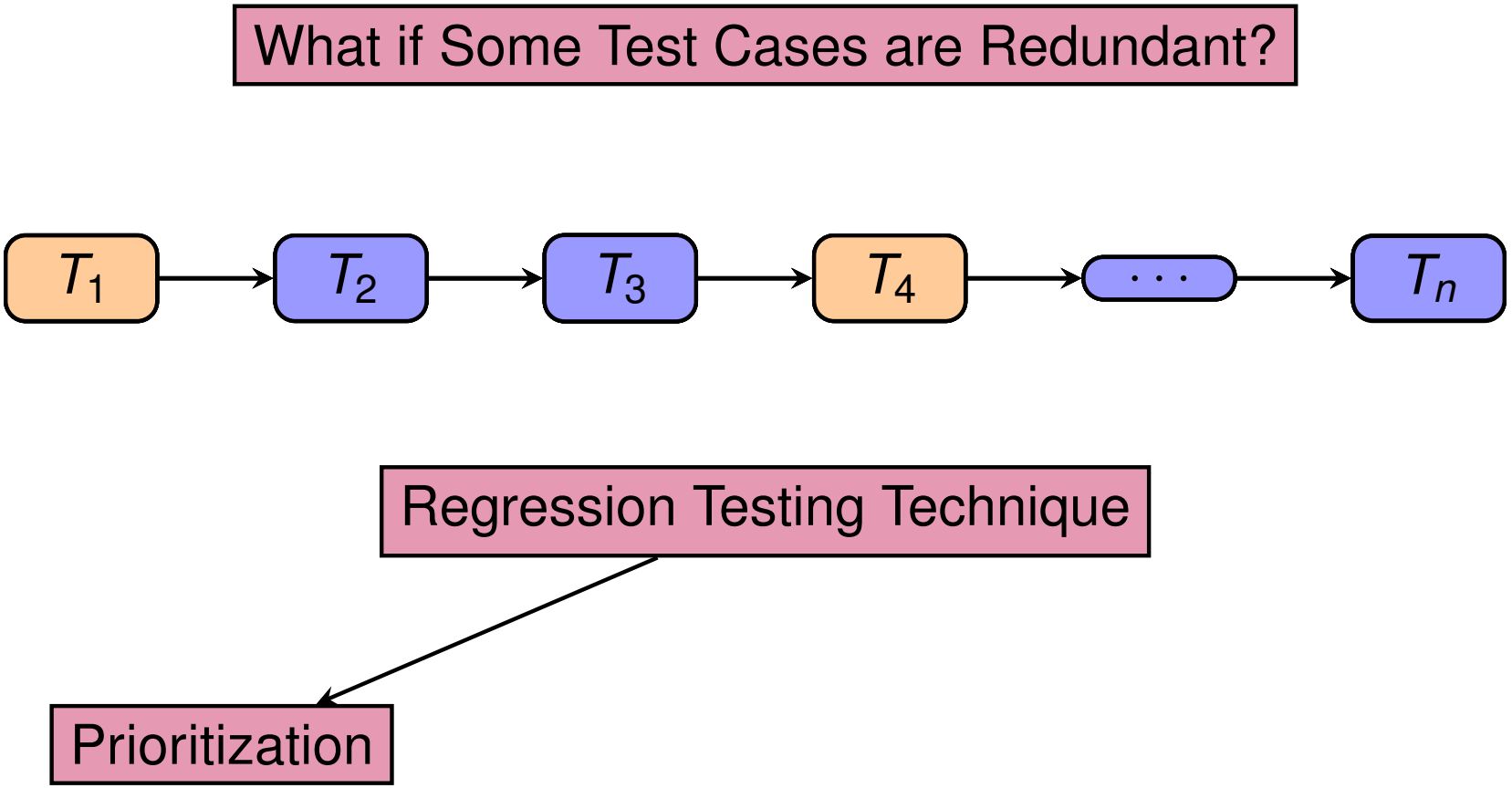
Test Suite Management



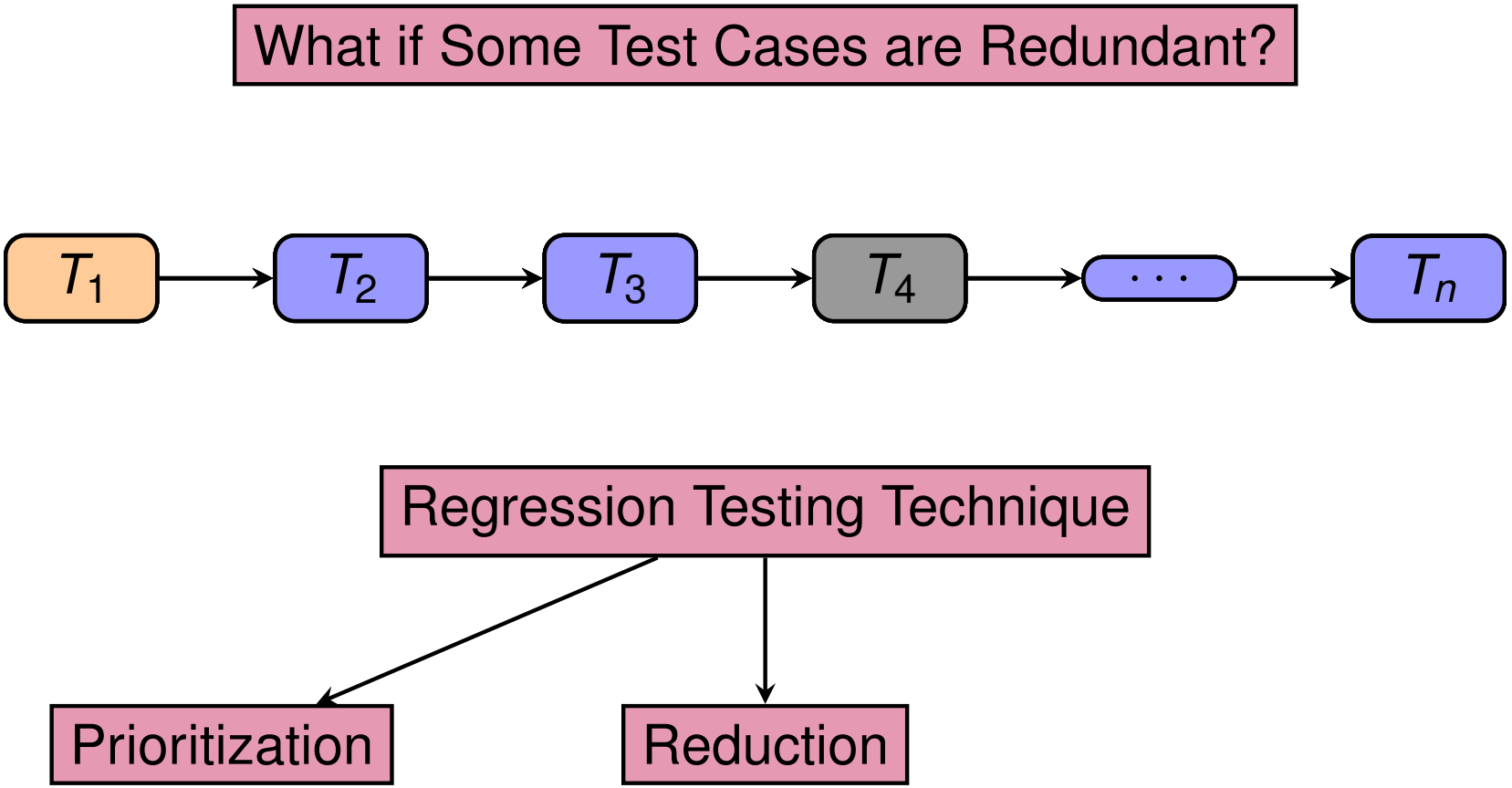
Test Suite Management



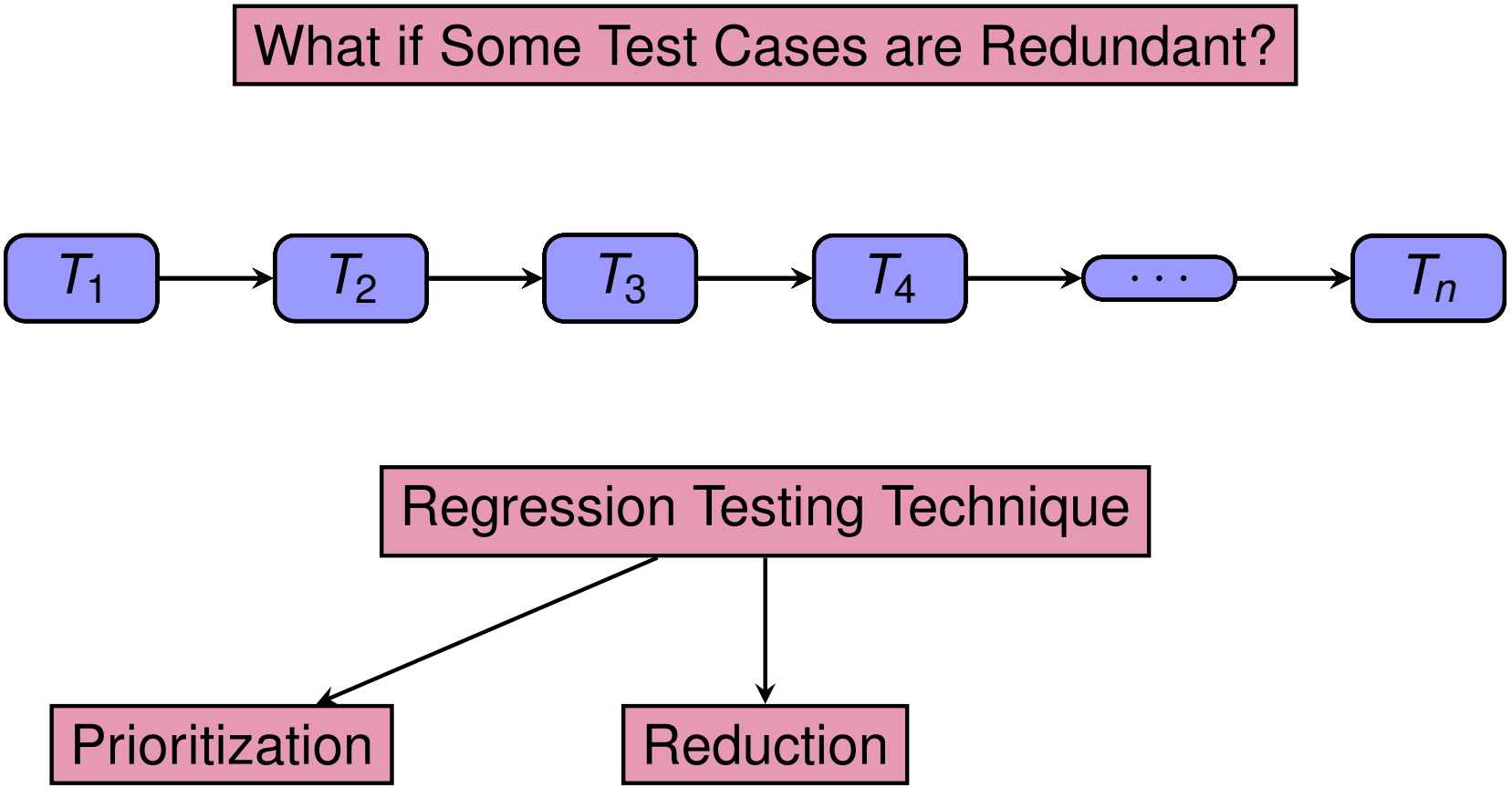
Test Suite Management



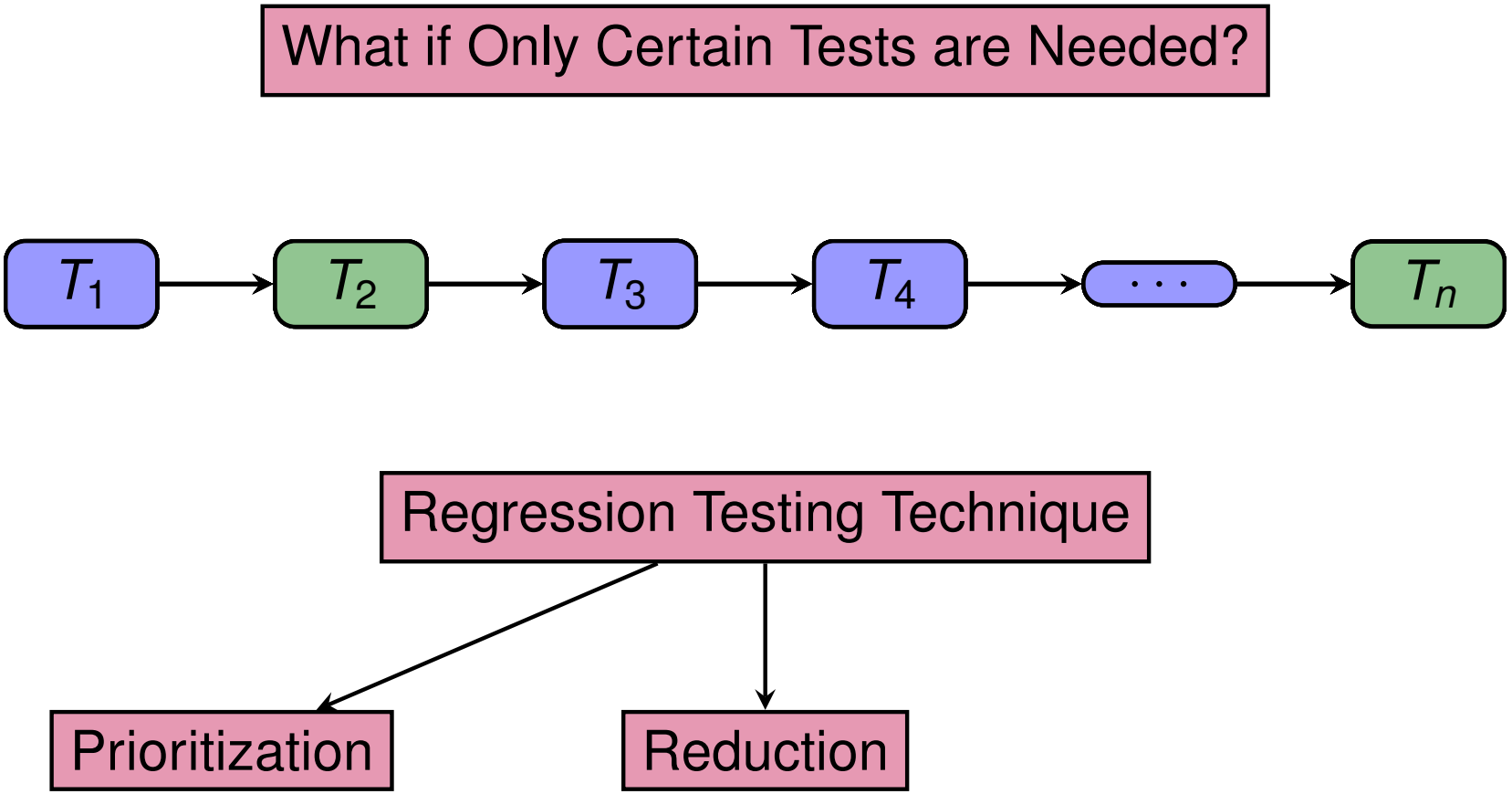
Test Suite Management



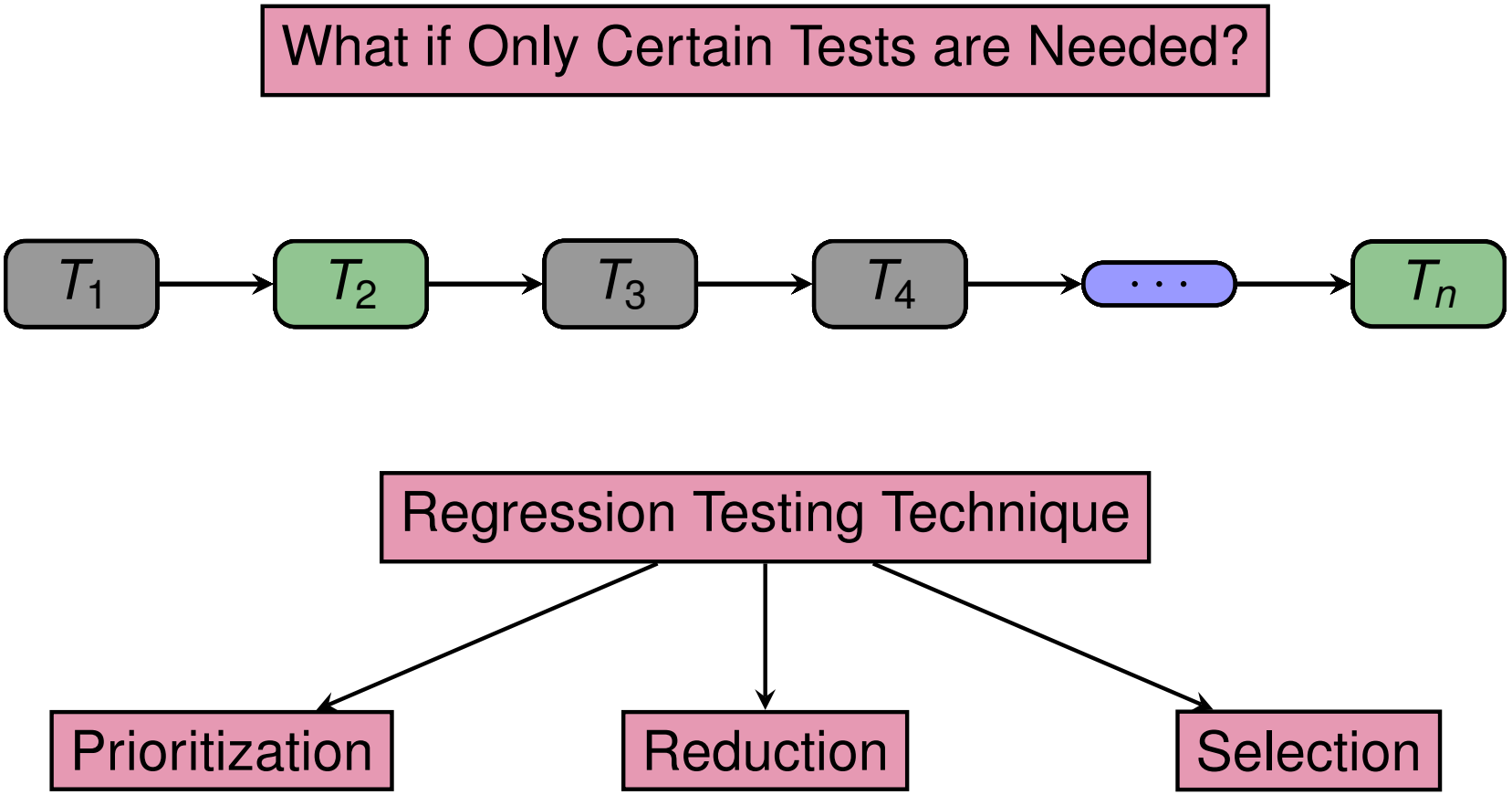
Test Suite Management



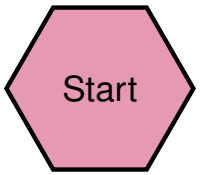
Test Suite Management



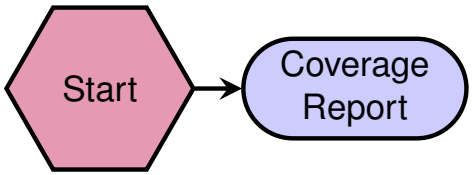
Test Suite Management



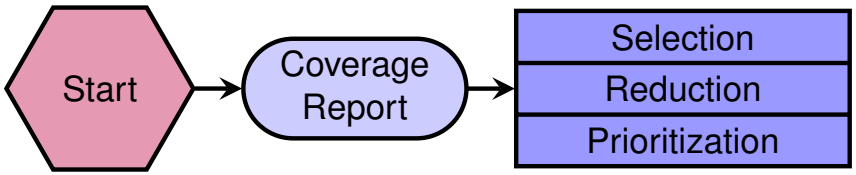
Model of Regression Testing



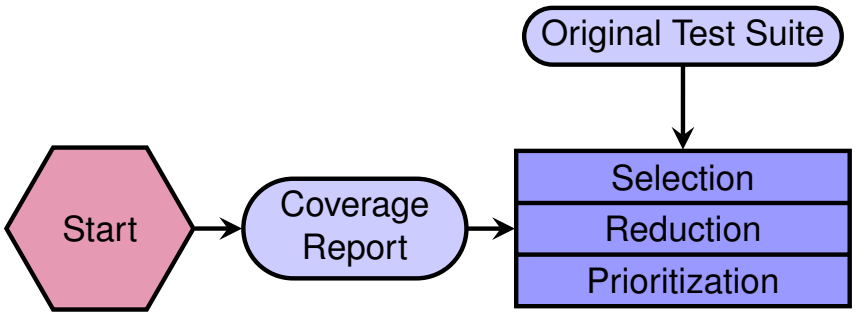
Model of Regression Testing



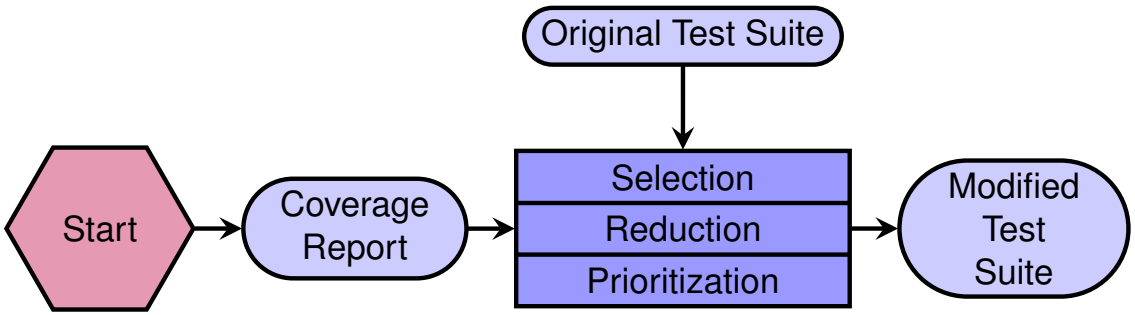
Model of Regression Testing



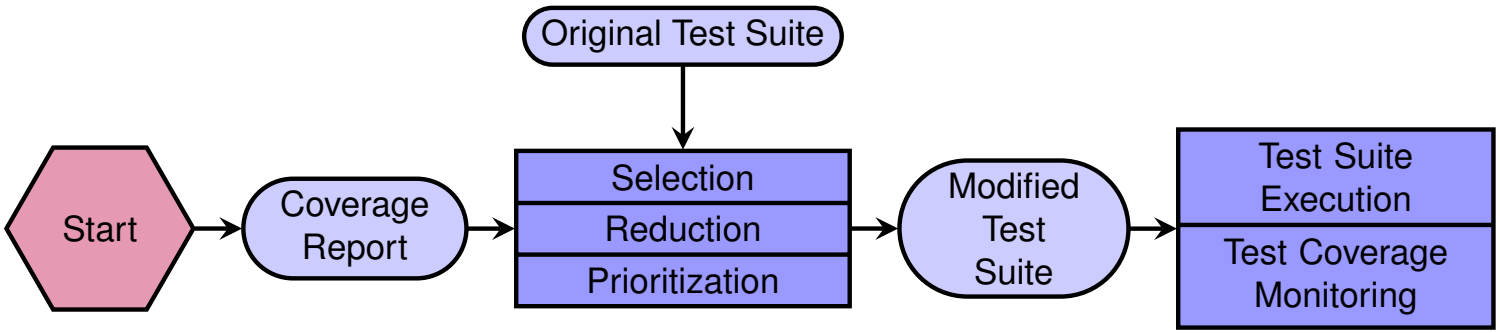
Model of Regression Testing



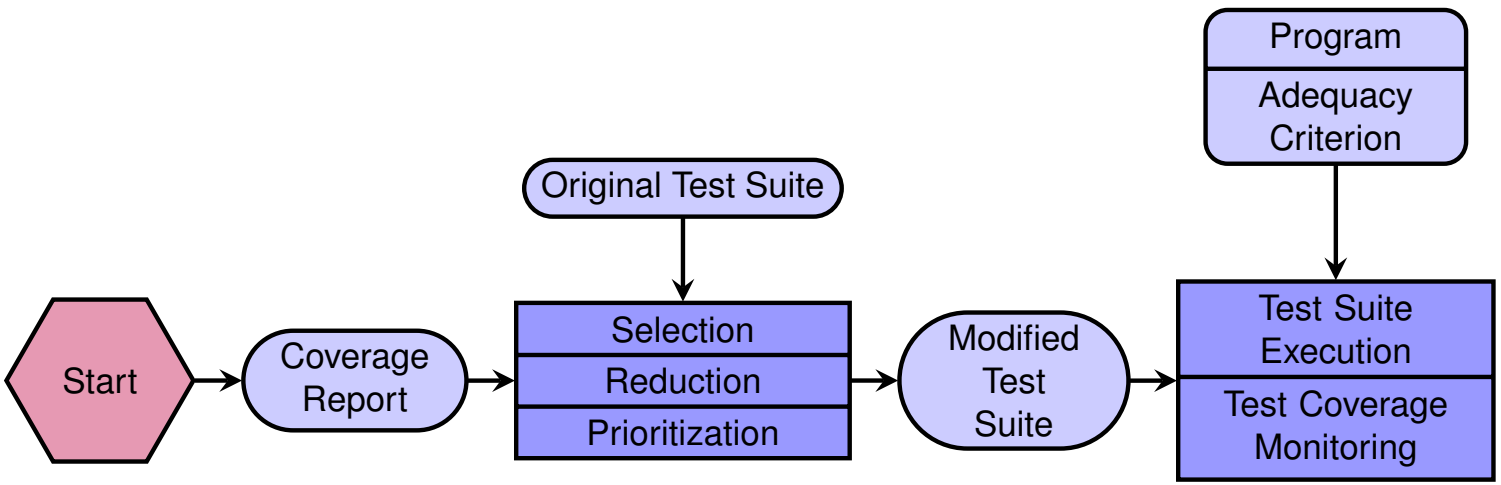
Model of Regression Testing



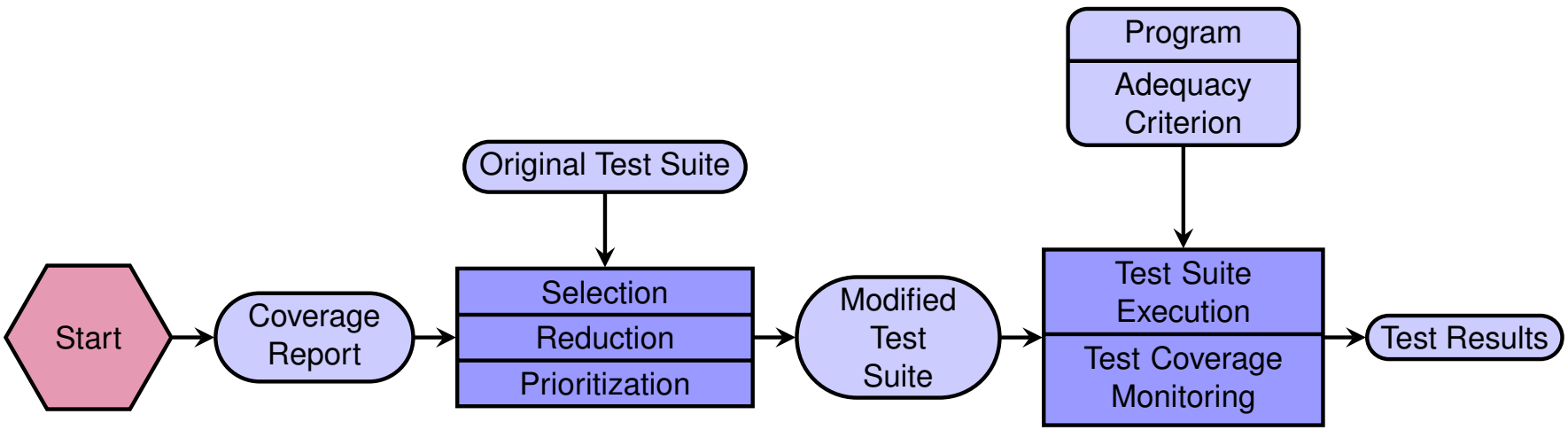
Model of Regression Testing



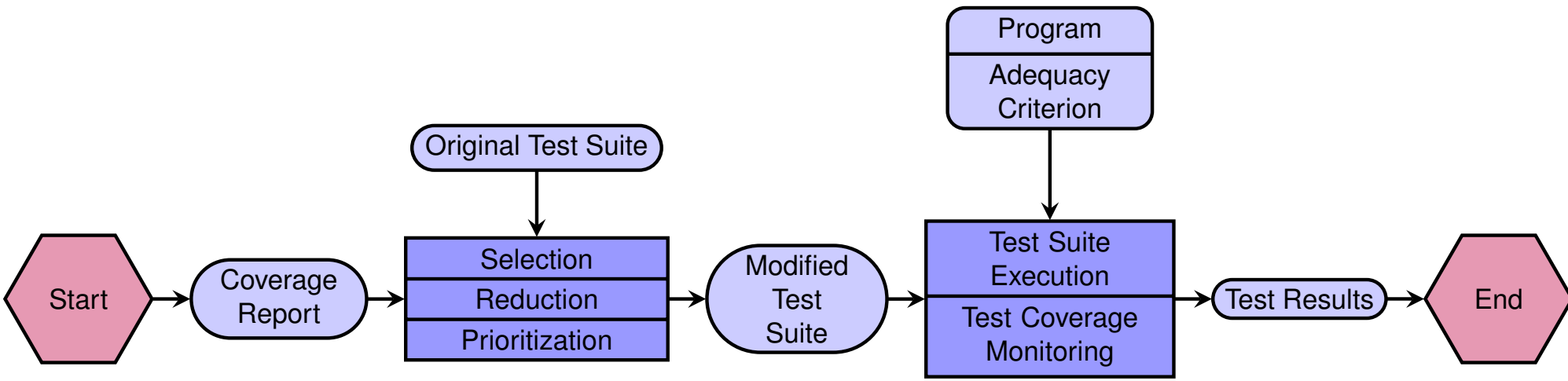
Model of Regression Testing



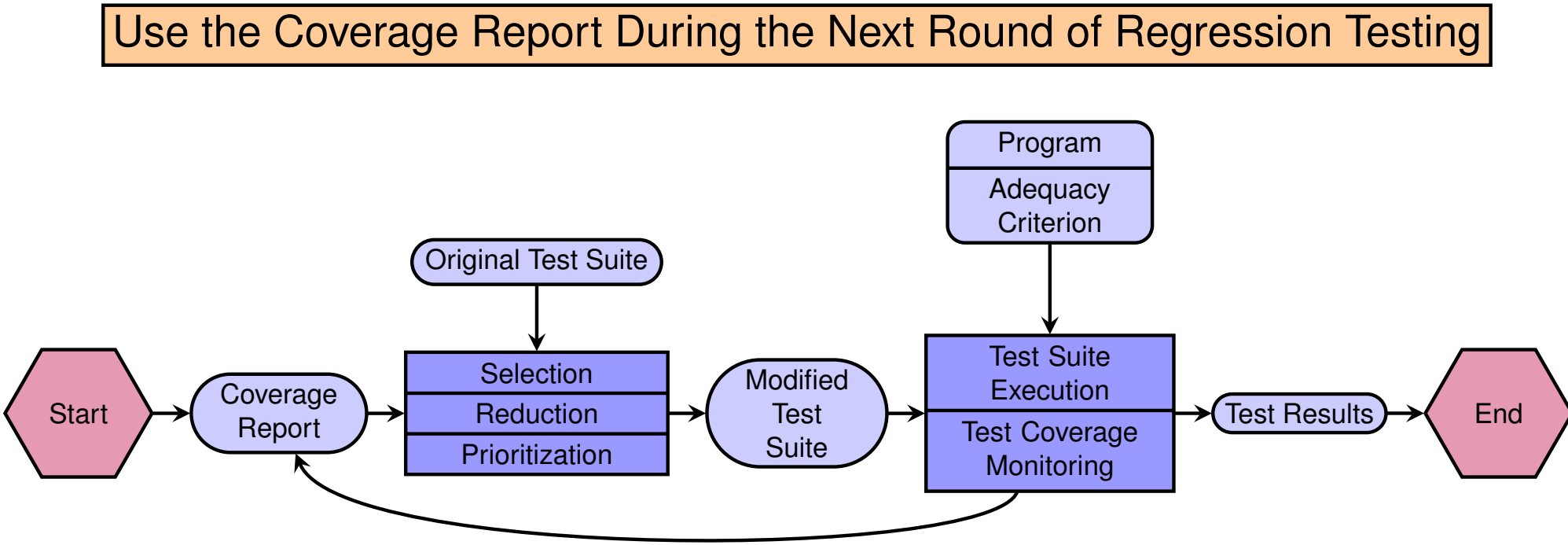
Model of Regression Testing



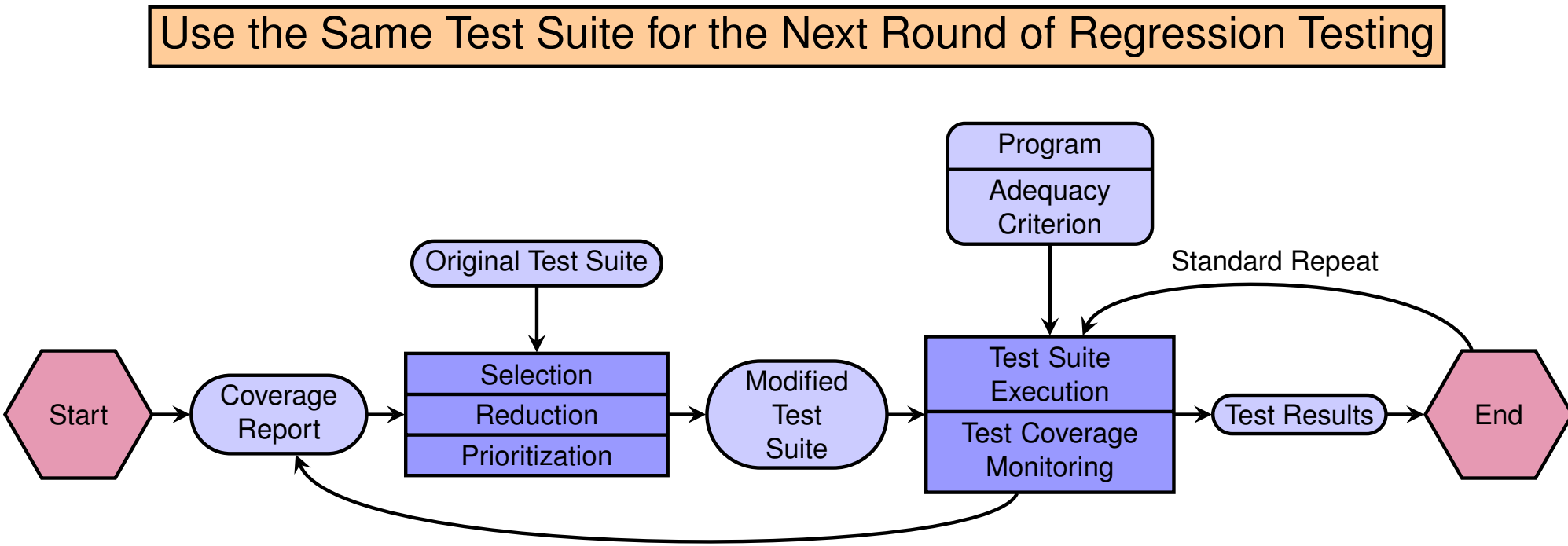
Model of Regression Testing



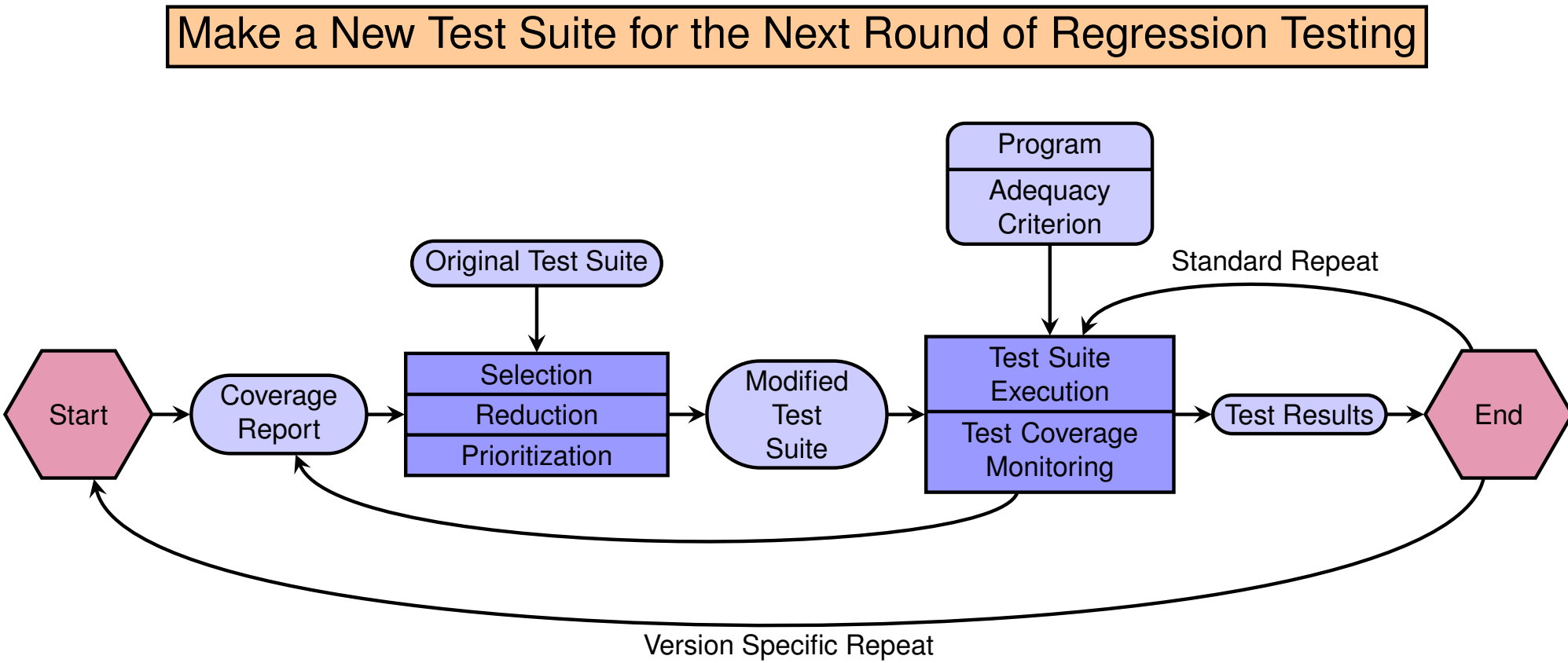
Model of Regression Testing



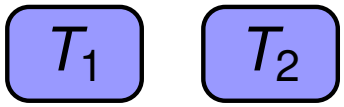
Model of Regression Testing



Model of Regression Testing



Test Suite Adequacy



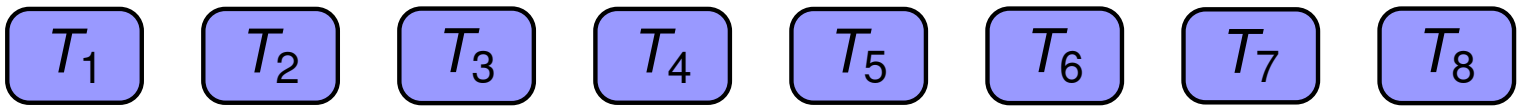
Test Suite Adequacy

T_1 T_2 T_3 T_4

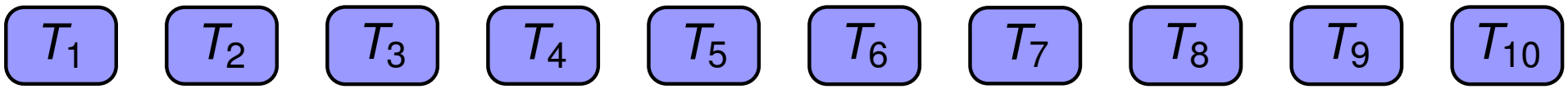
Test Suite Adequacy

T_1 T_2 T_3 T_4 T_5 T_6

Test Suite Adequacy

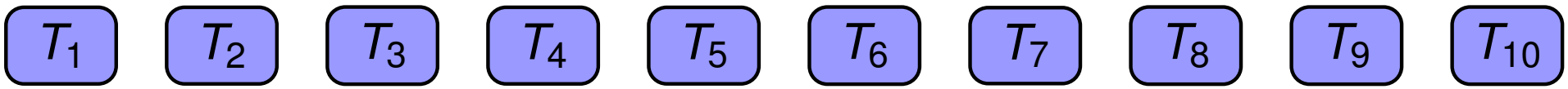


Test Suite Adequacy



Test Suite Adequacy

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$



Test Suite Adequacy

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

R_1 R_2

Test Suite Adequacy

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

R_1 R_2 R_3 R_4

Test Suite Adequacy

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

R_1 R_2 R_3 R_4 R_5 R_6

Test Suite Adequacy

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8

Test Suite Adequacy

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8 R_9 R_{10}

Test Suite Adequacy

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8 R_9 R_{10} R_{11} R_{12}

Test Suite Adequacy

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

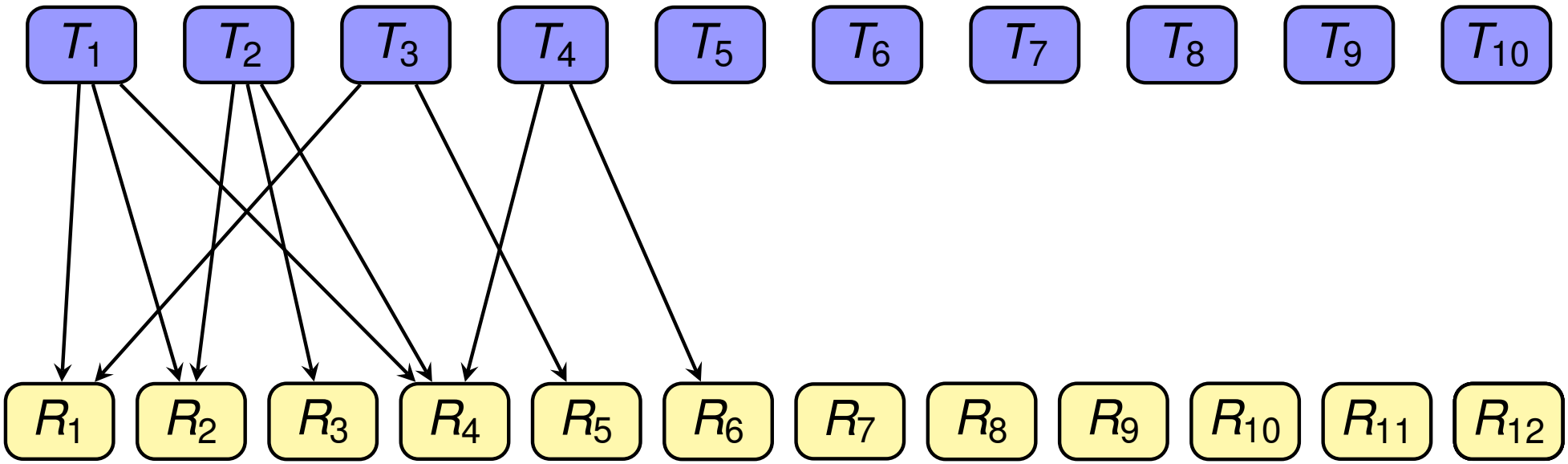
T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8 R_9 R_{10} R_{11} R_{12}

Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Test Suite Adequacy

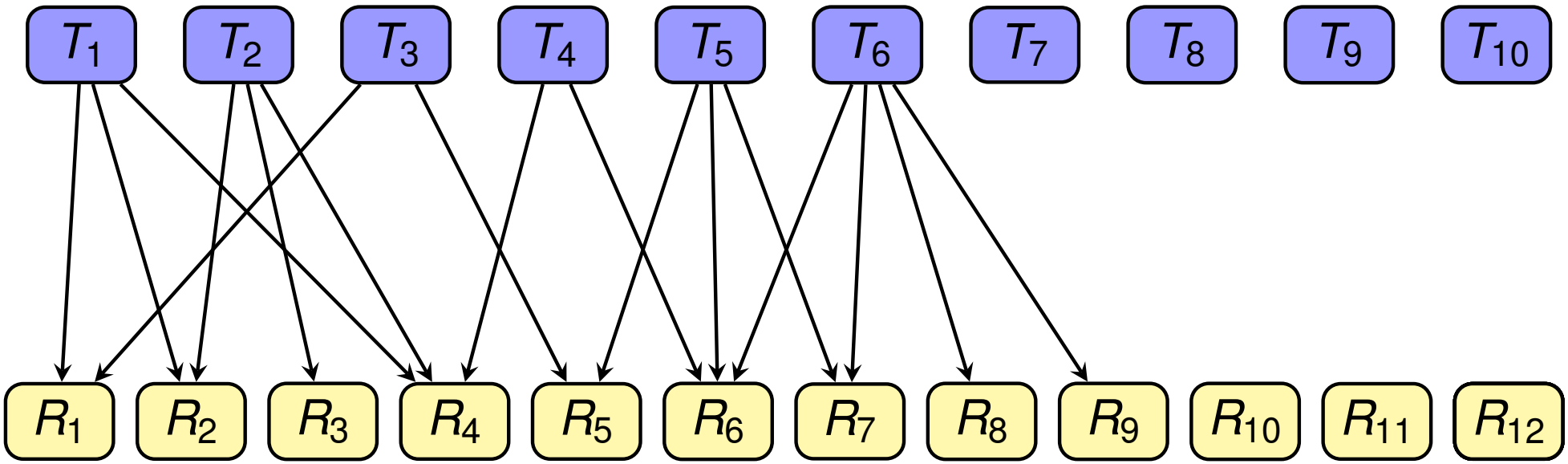
Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$



Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Test Suite Adequacy

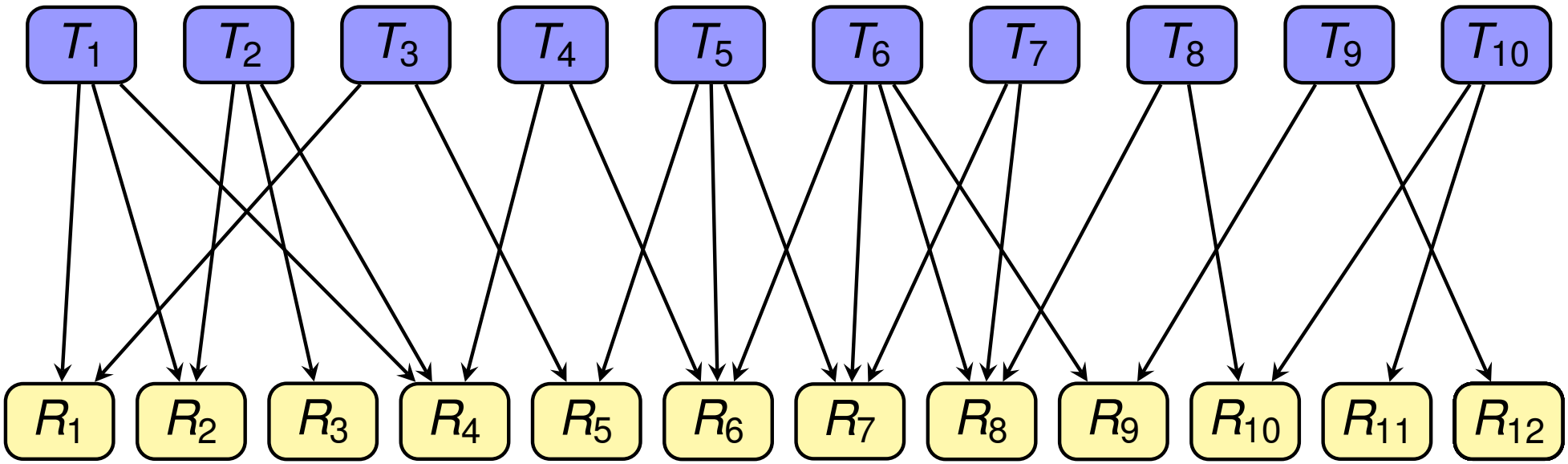
Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$



Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

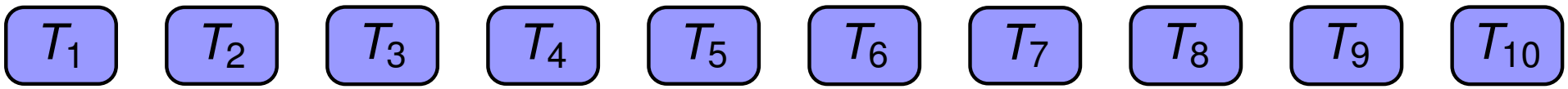
Test Suite Adequacy

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$



Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Test Suite Execution



Test Suite Execution

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

Test Suite Execution

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

JUnit and DBUnit Test Automation Frameworks

Test Suite Execution

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

JUnit and DBUnit Test Automation Frameworks

Run Test Case

Test Suite Execution

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

JUnit and DBUnit Test Automation Frameworks

Passing Test Case: $O_A = O_E$

Test Suite Execution

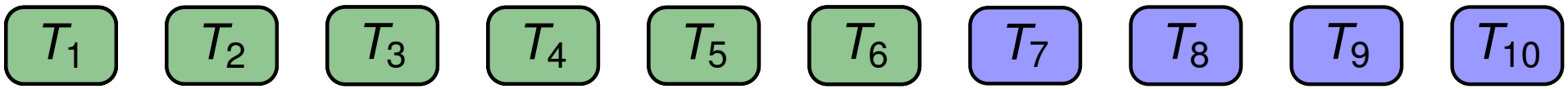
Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

JUnit and DBUnit Test Automation Frameworks

Test Suite Execution

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$



JUnit and DBUnit Test Automation Frameworks

Test Suite Execution

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

JUnit and DBUnit Test Automation Frameworks

Run Test Case

Test Suite Execution

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

JUnit and DBUnit Test Automation Frameworks

Failing Test Case: $O_A \neq O_E$

Test Suite Execution

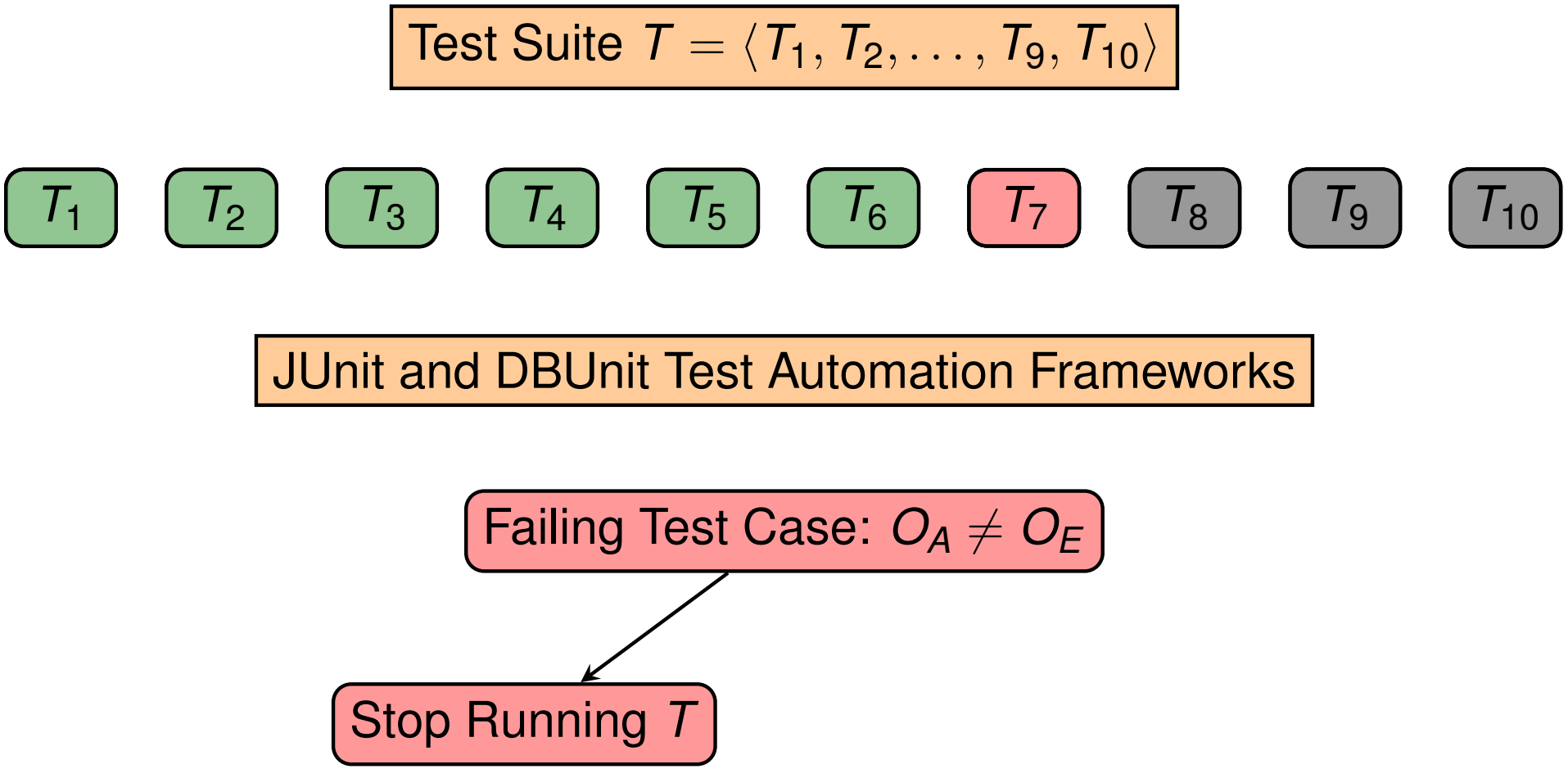
Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

JUnit and DBUnit Test Automation Frameworks

Failing Test Case: $O_A \neq O_E$

Test Suite Execution



Test Suite Execution

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

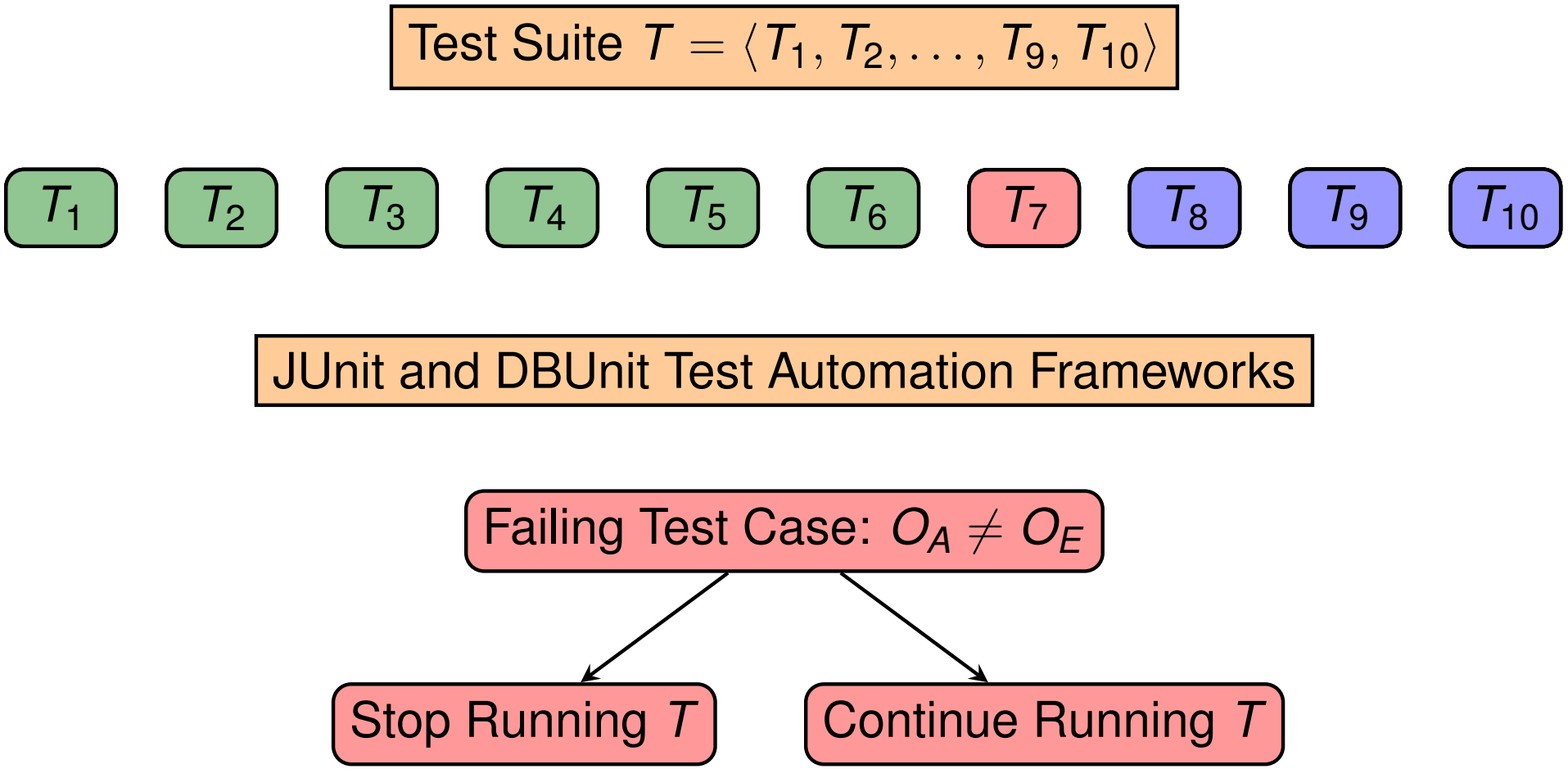
T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

JUnit and DBUnit Test Automation Frameworks

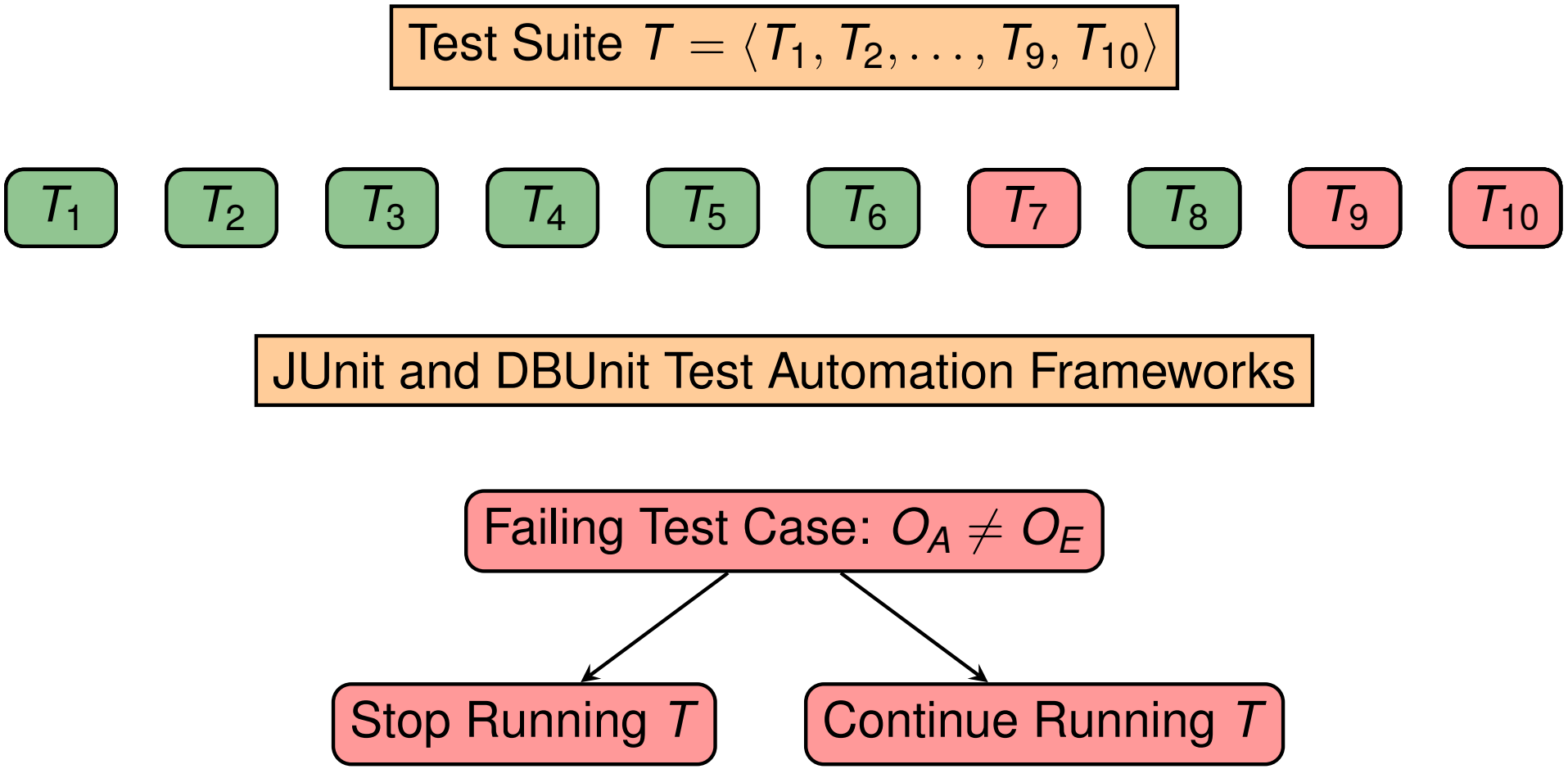
Failing Test Case: $O_A \neq O_E$

Stop Running T

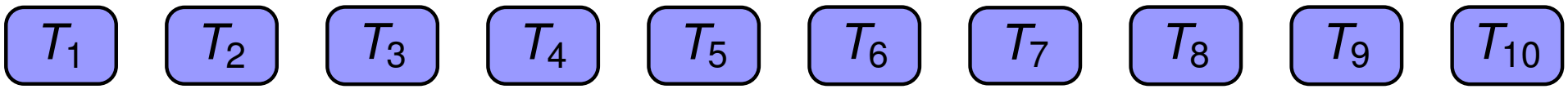
Test Suite Execution



Test Suite Execution



Test Coverage Monitoring



Test Coverage Monitoring

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

Test Coverage Monitoring

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

JUnit and DBUnit Test Automation Frameworks
Database-Aware Test Coverage Monitor
Proteja Test Suite Manager

Test Coverage Monitoring

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

JUnit and DBUnit Test Automation Frameworks
Database-Aware Test Coverage Monitor
Proteja Test Suite Manager

Run Test Case

Collect Per-Test Case Coverage

Test Coverage Monitoring

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8 R_9 R_{10} R_{11} R_{12}

Test Coverage Monitoring

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

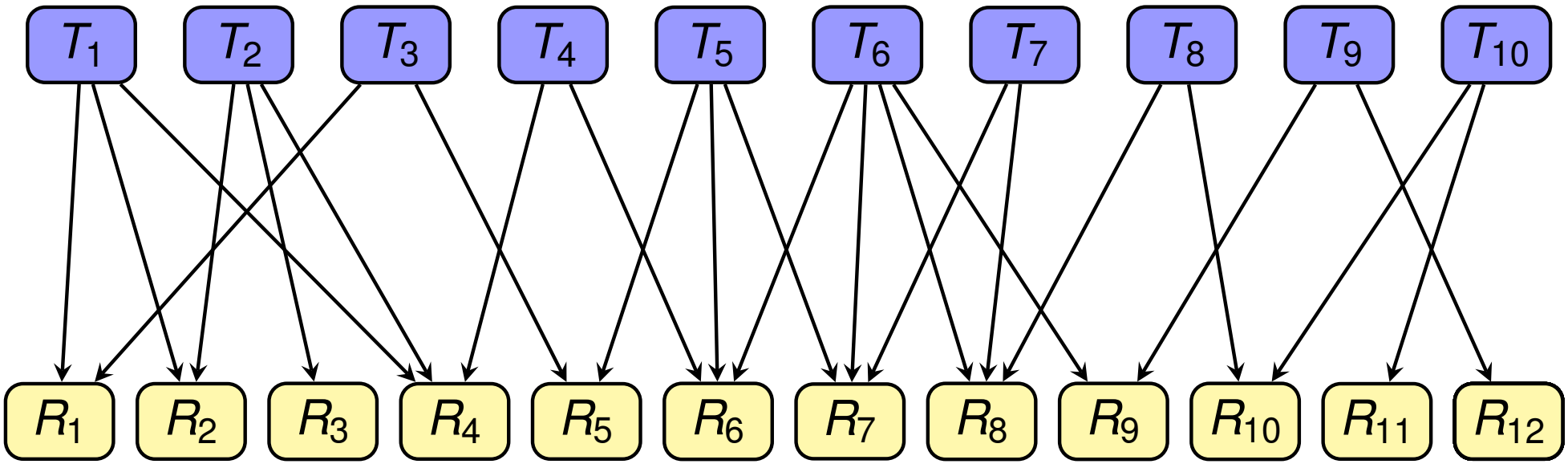
T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8 R_9 R_{10} R_{11} R_{12}

Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Test Coverage Monitoring

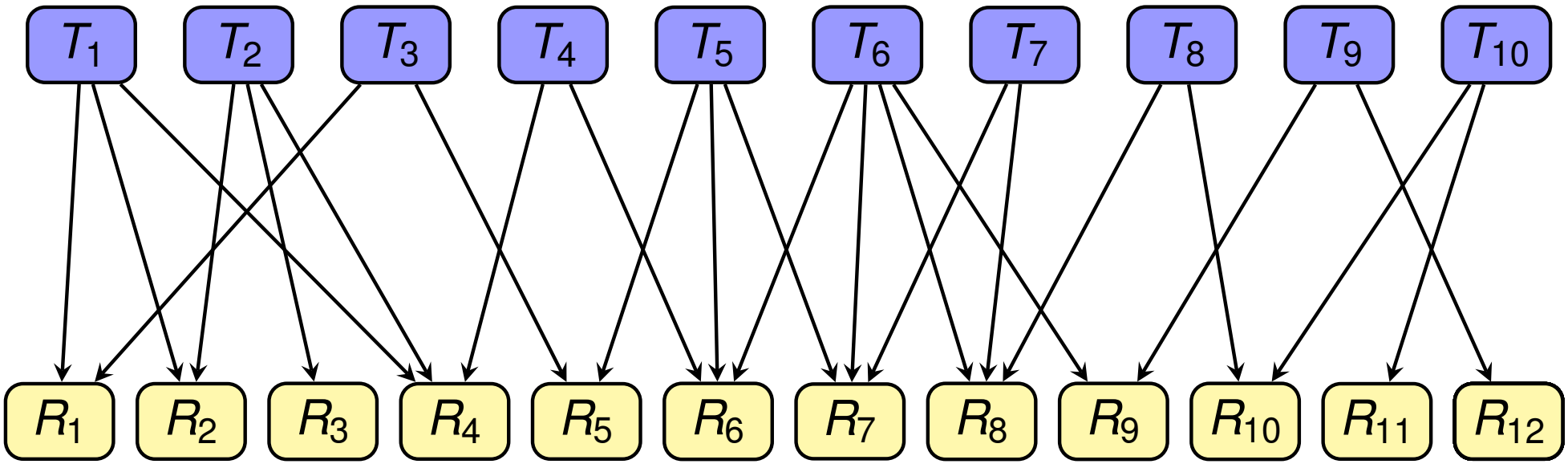
Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$



Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Test Coverage Monitoring

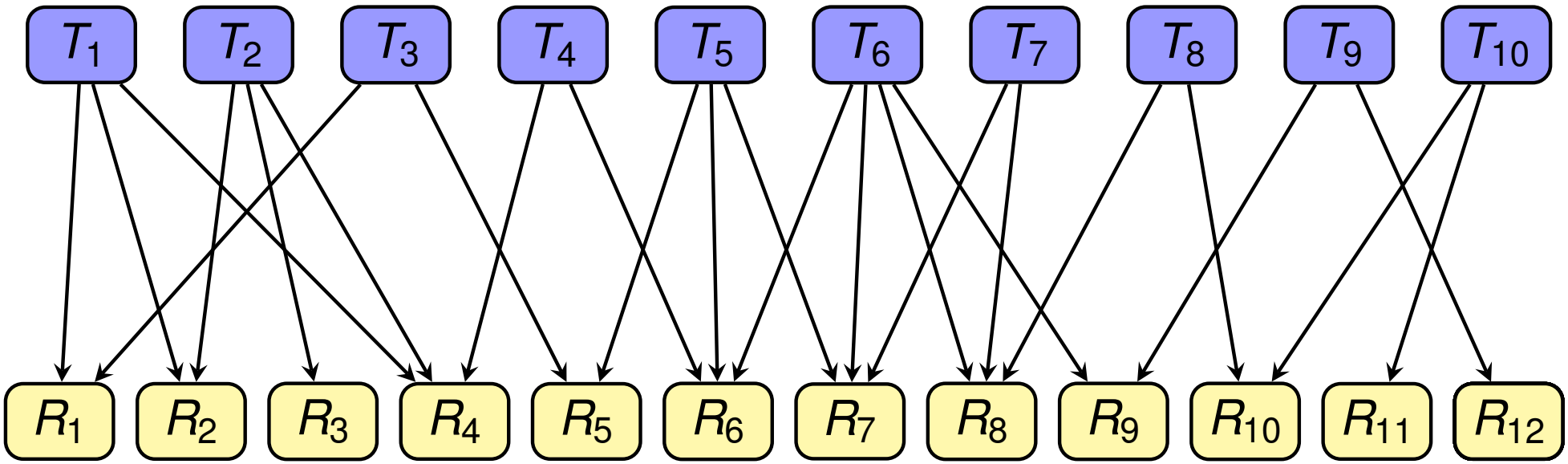
Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$



Requirements Set R for ... Statement Coverage

Test Coverage Monitoring

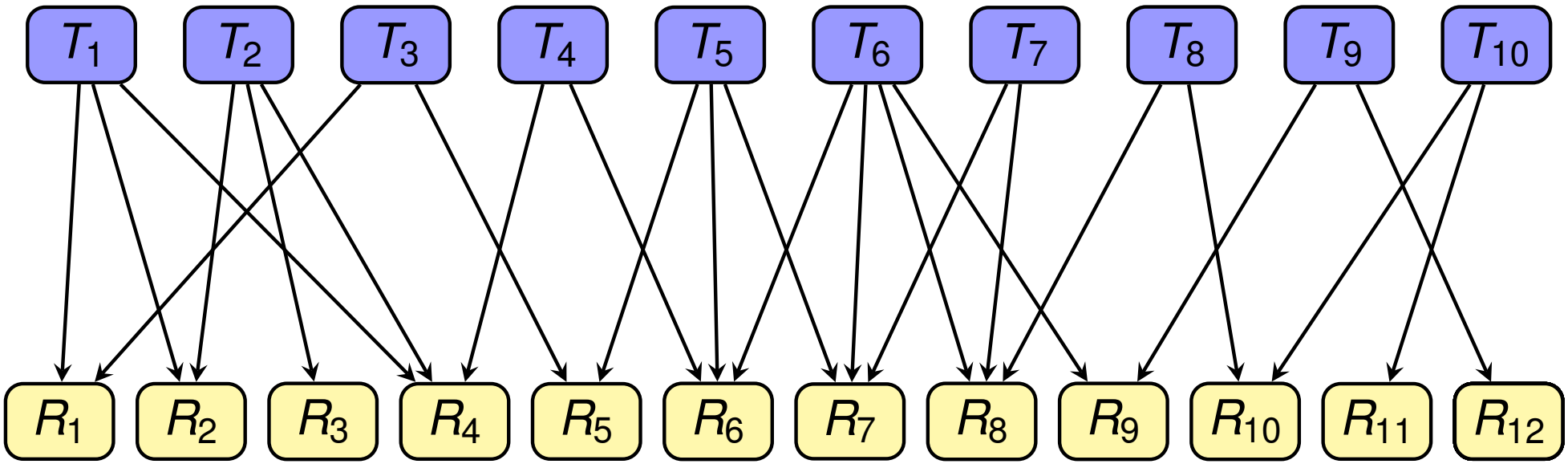
Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$



Requirements Set R for ... Database Interaction Coverage

Test Coverage Monitoring

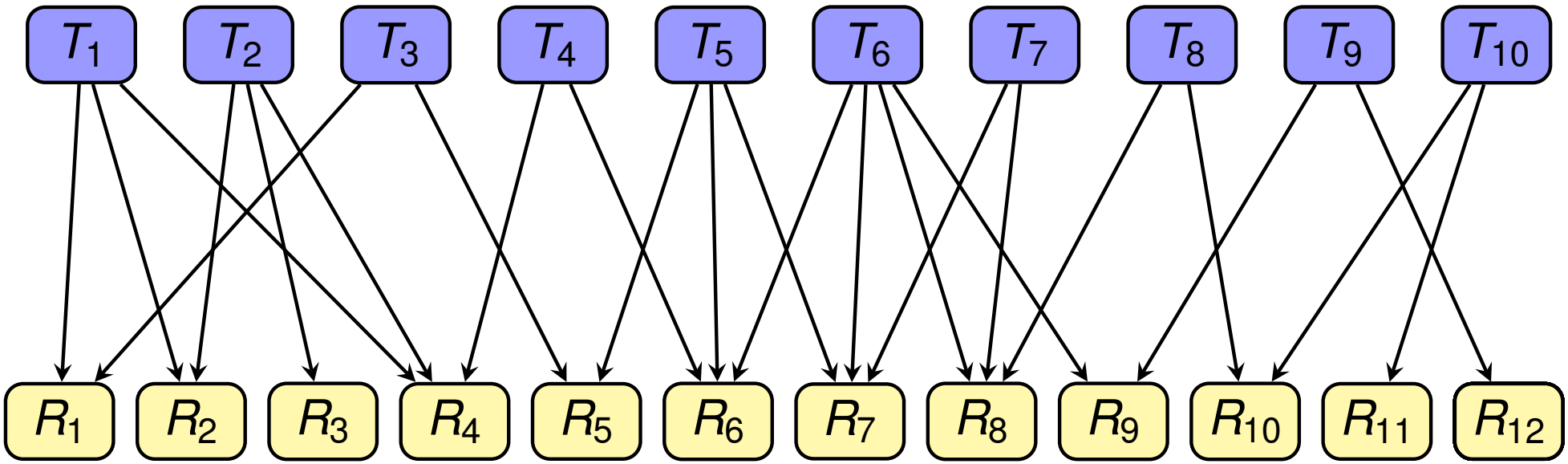
Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$



Requirements Set R for ... Database Table Coverage

Test Coverage Monitoring

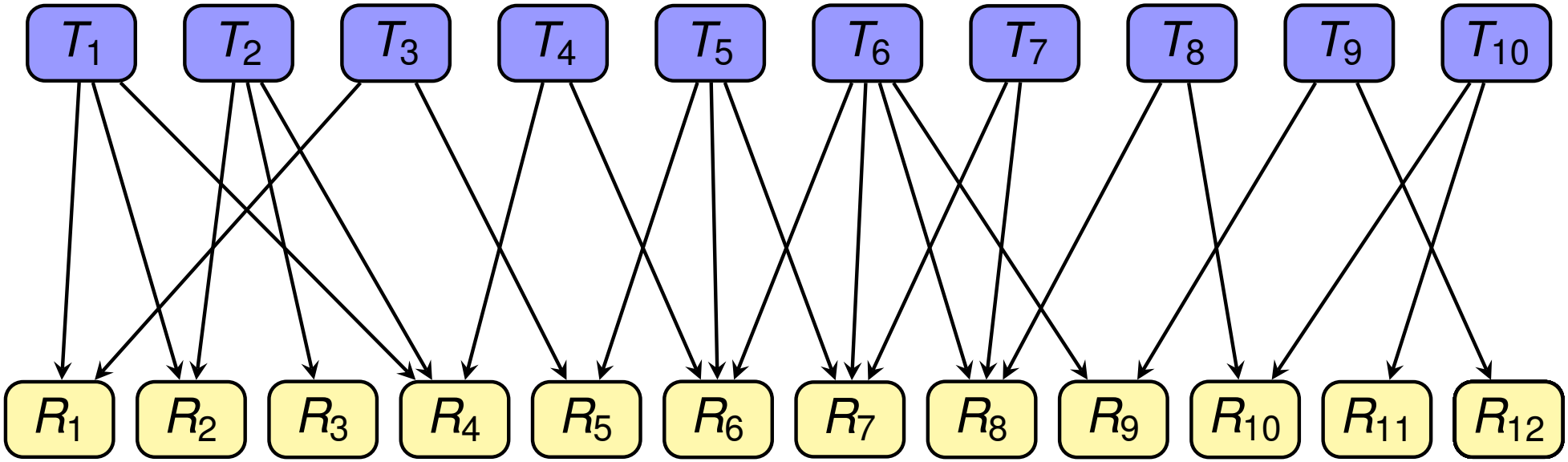
Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$



Requirements Set R for ... Database Record Coverage

Test Coverage Monitoring

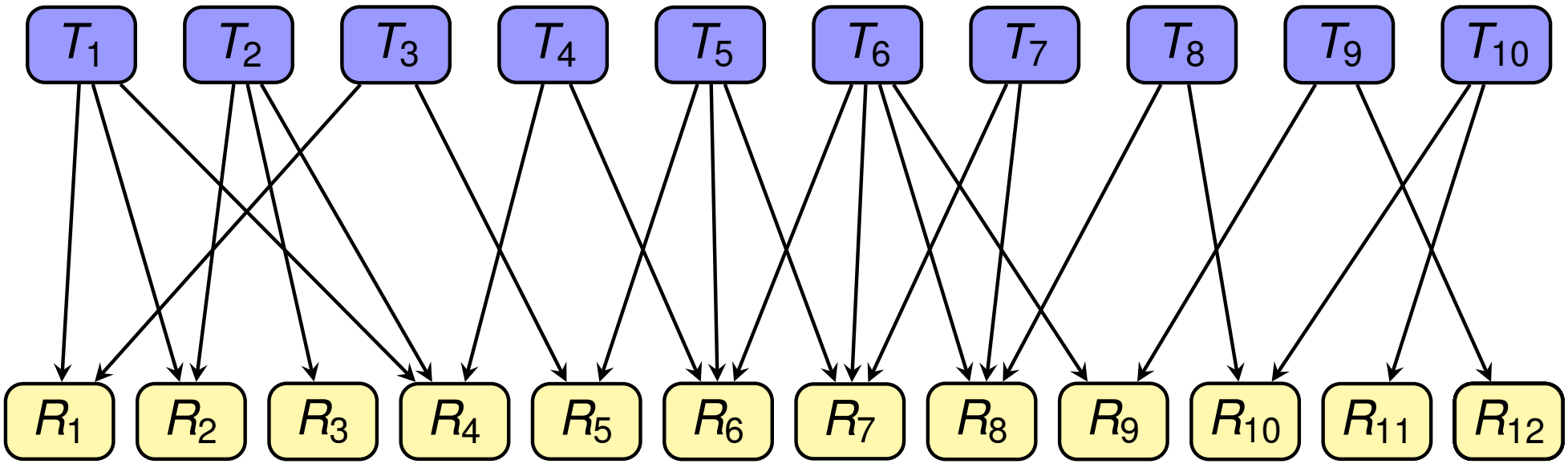
Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$



Requirements Set R for ... Database Attribute Coverage

Test Coverage Monitoring

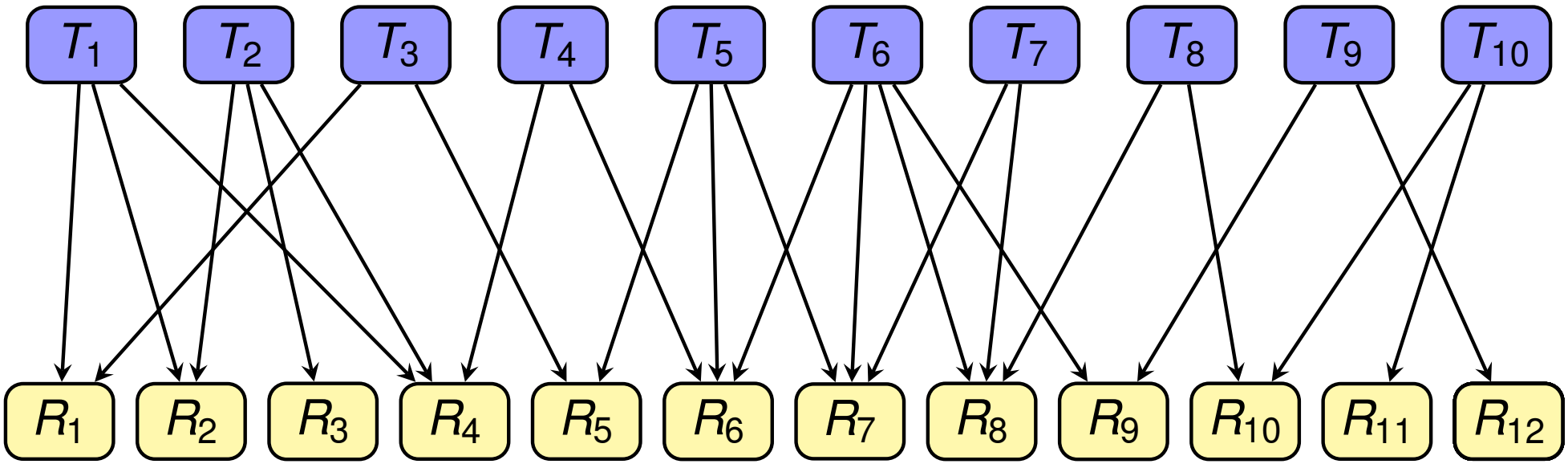
Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$



Requirements Set R for ... Database Attribute-Value Coverage

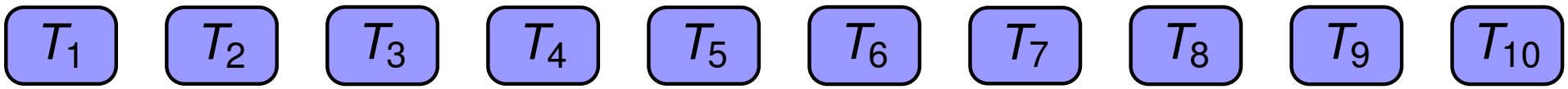
Test Coverage Monitoring

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$



See [Kapfhammer and Soffa, ISEC 2008] for more details

Greedy Algorithms



Greedy Algorithms

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

Greedy Algorithms

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8 R_9 R_{10} R_{11} R_{12}

Greedy Algorithms

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

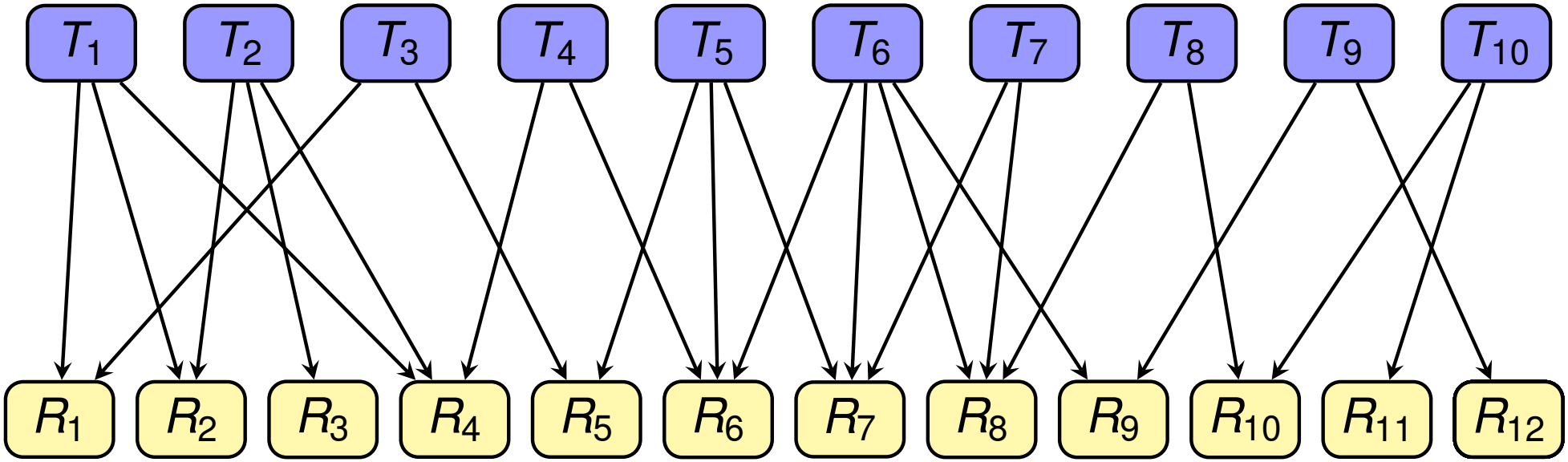
T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8 R_9 R_{10} R_{11} R_{12}

Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

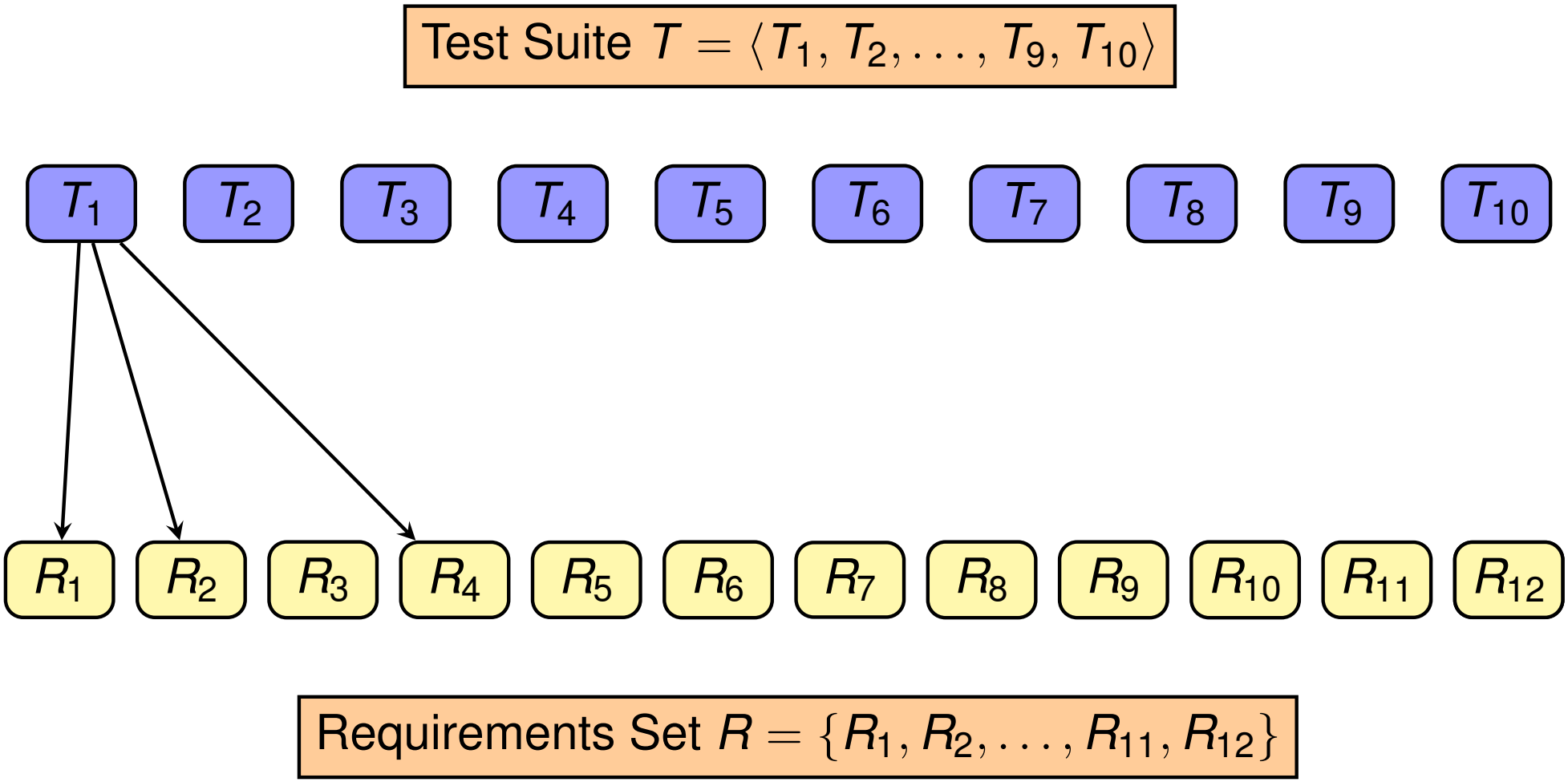
Greedy Algorithms

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$



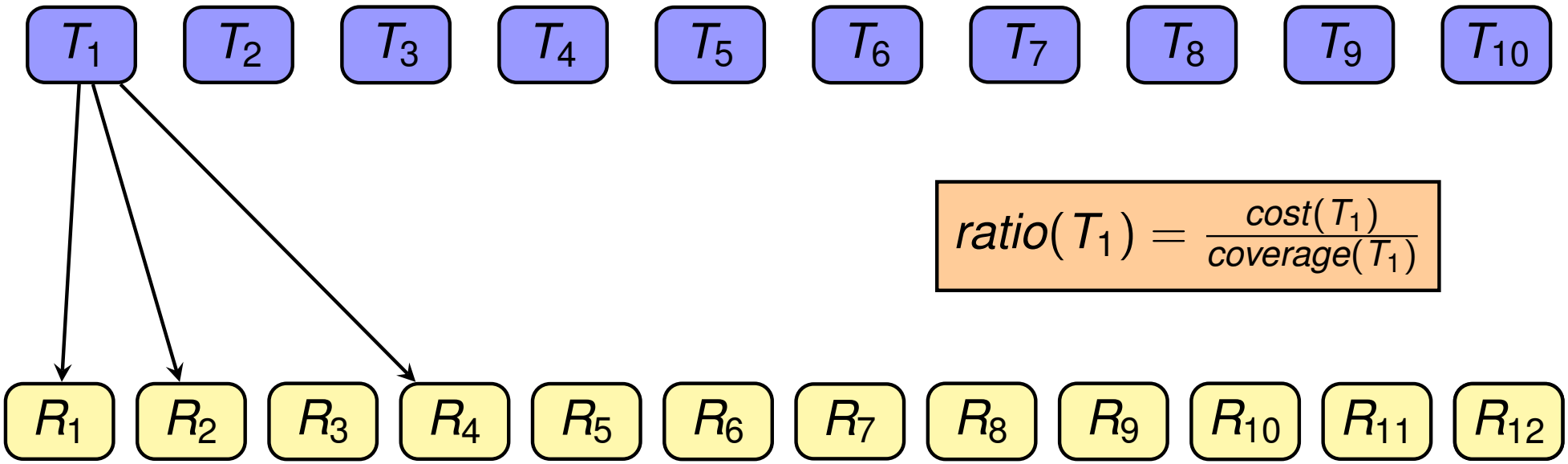
Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Greedy Algorithms



Greedy Algorithms

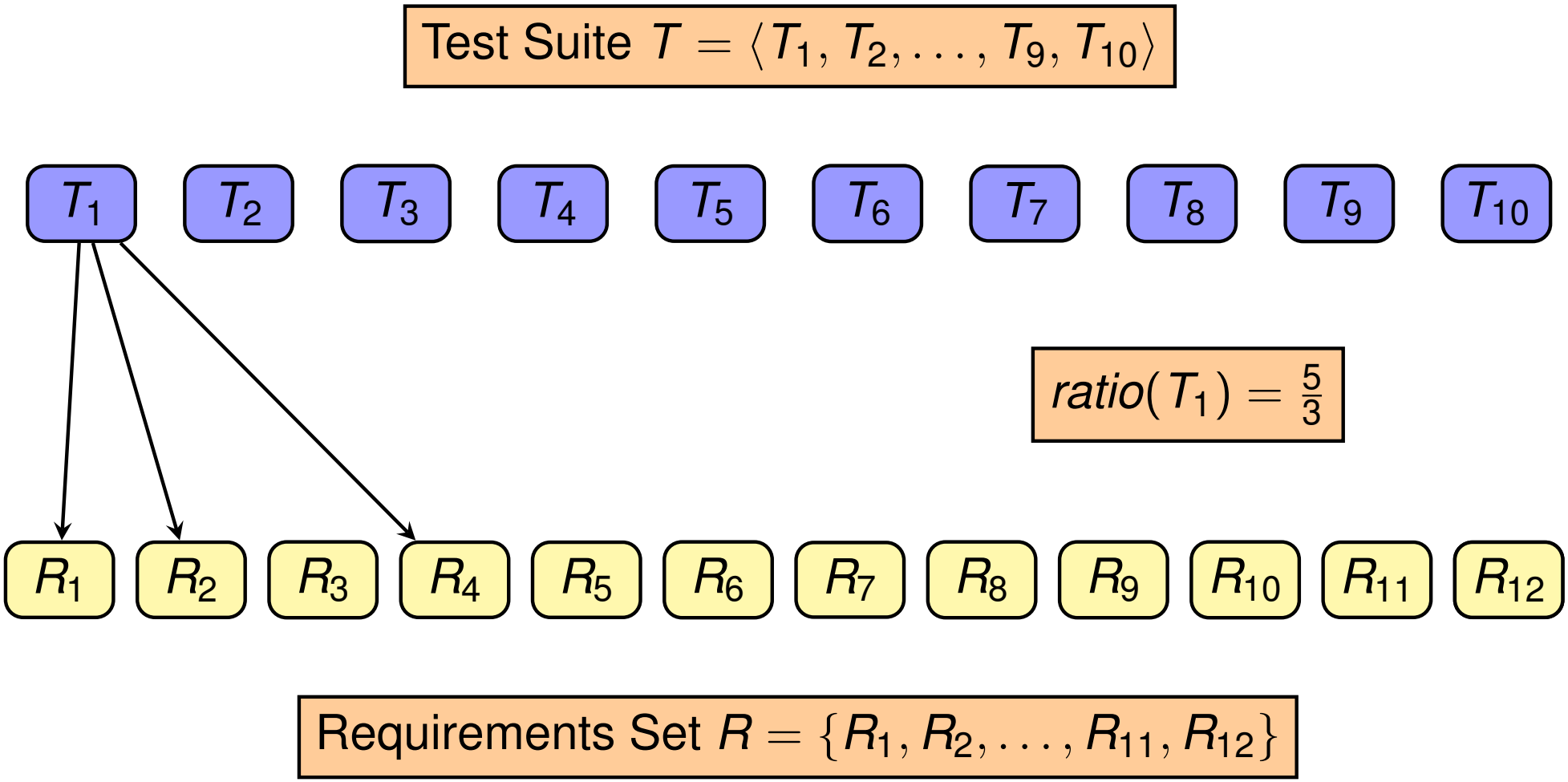
Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$



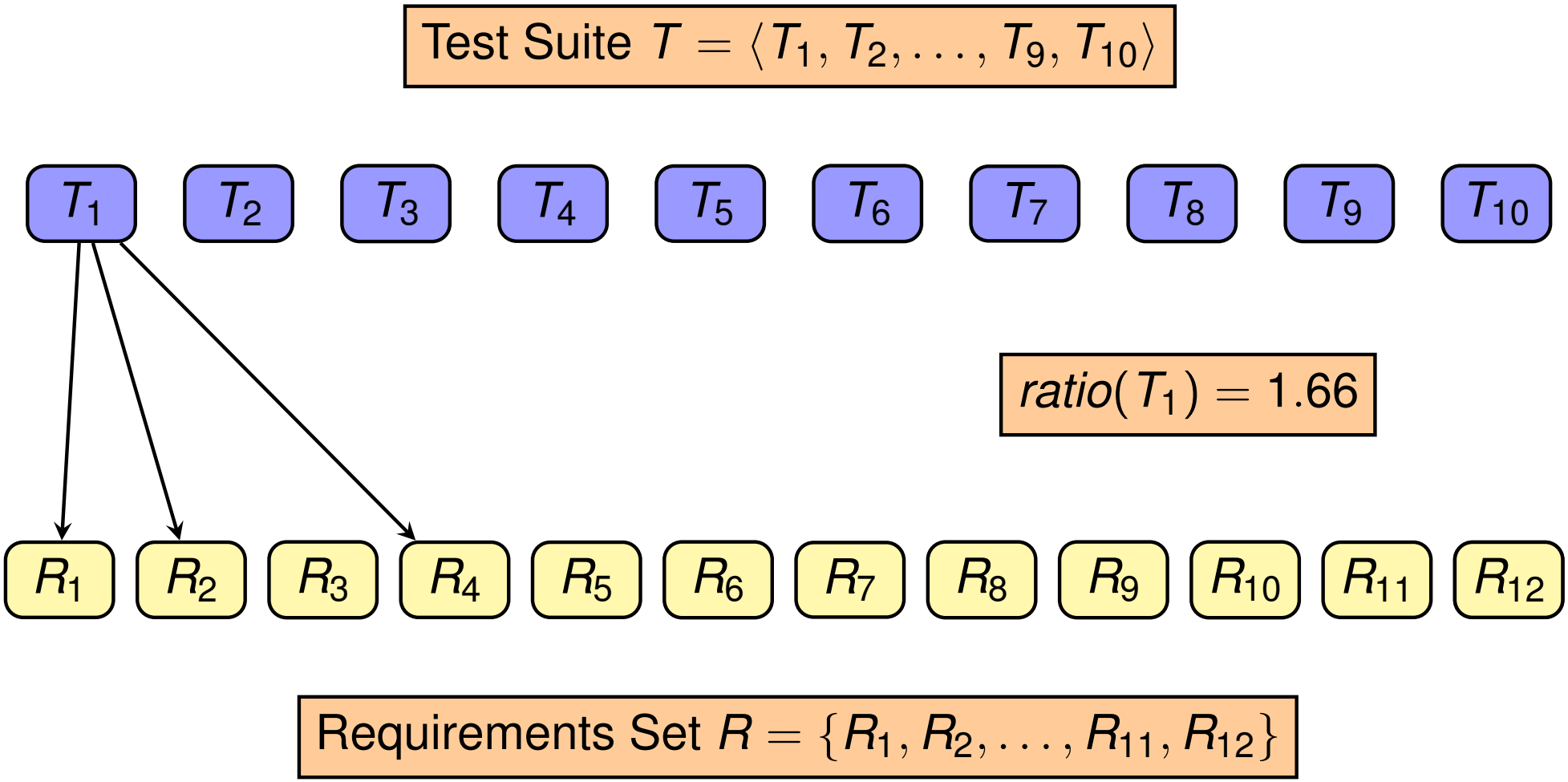
$$ratio(T_1) = \frac{cost(T_1)}{coverage(T_1)}$$

Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Greedy Algorithms



Greedy Algorithms



Greedy Algorithms

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

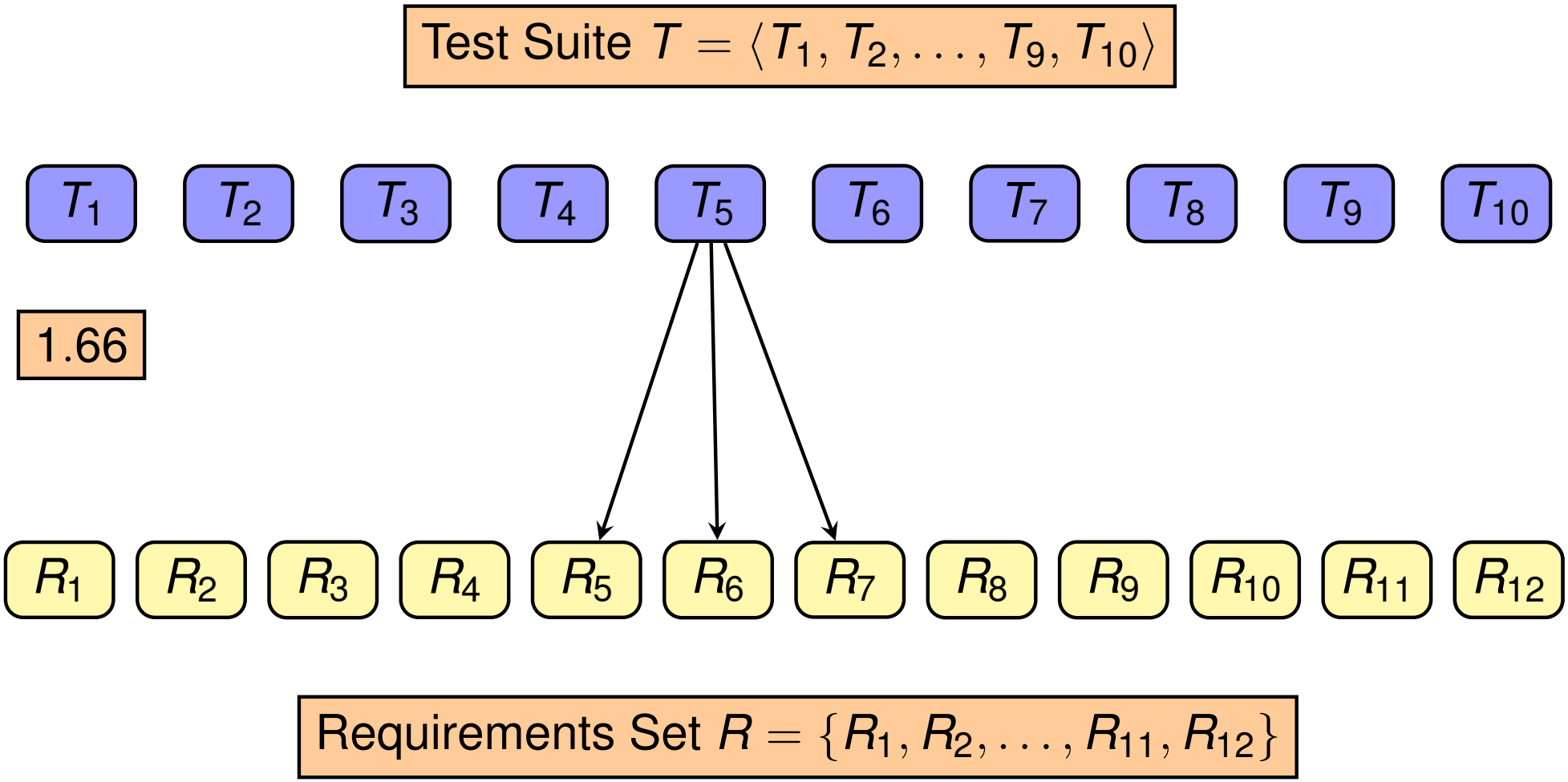
T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

1.66

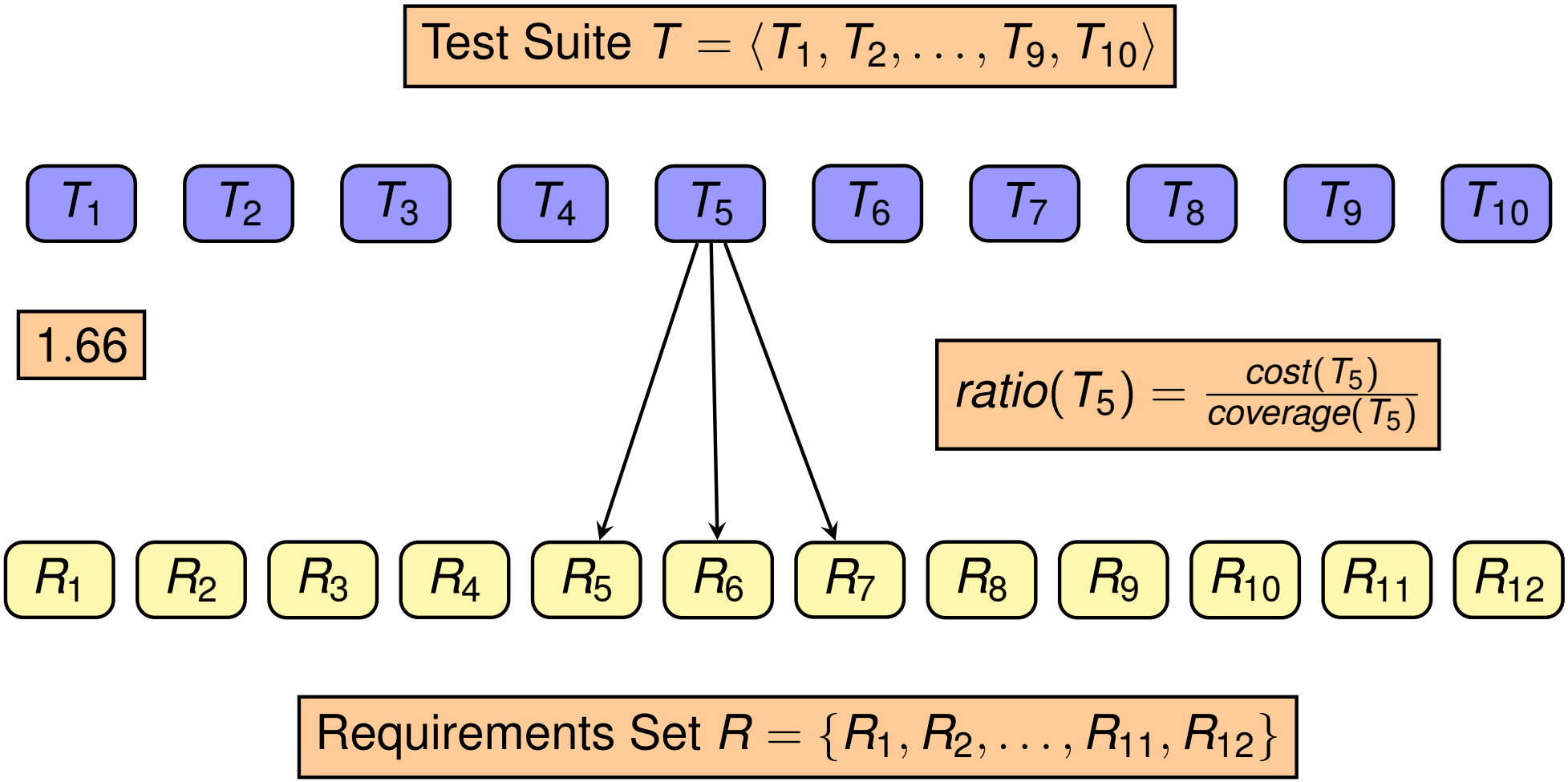
R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8 R_9 R_{10} R_{11} R_{12}

Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

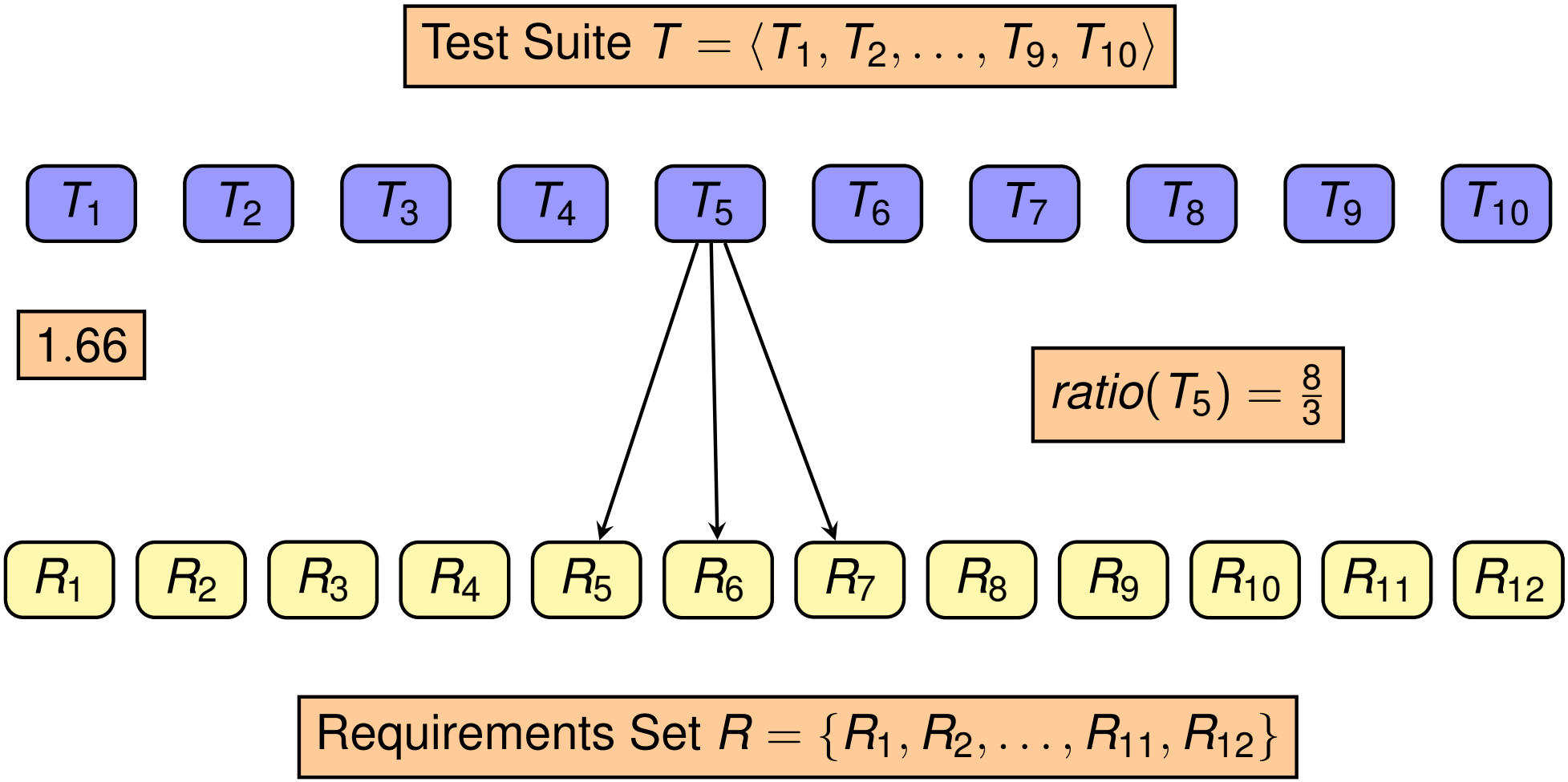
Greedy Algorithms



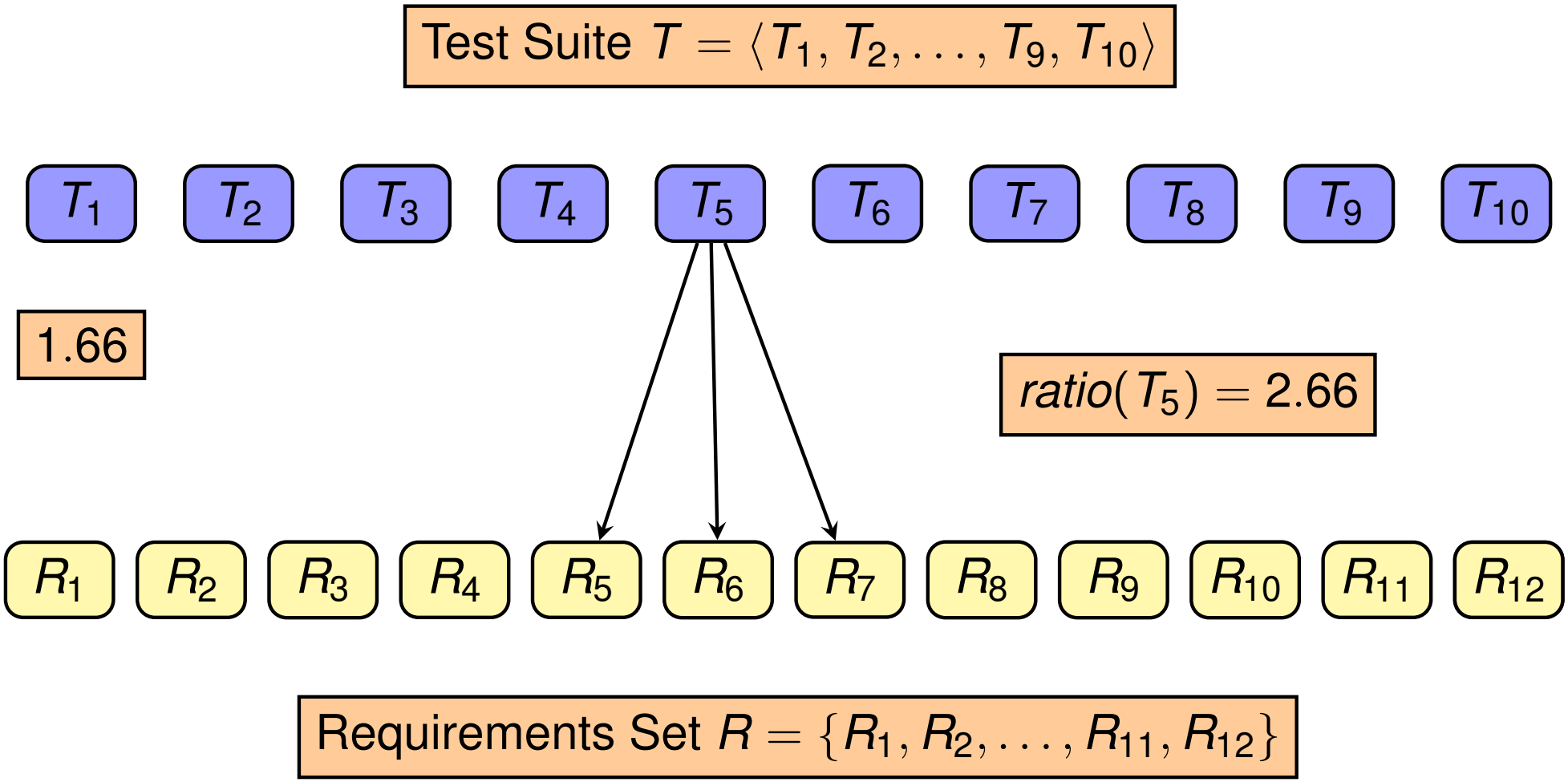
Greedy Algorithms



Greedy Algorithms



Greedy Algorithms



Greedy Algorithms

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

1.66 2.66

R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8 R_9 R_{10} R_{11} R_{12}

Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Greedy Algorithms

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

1.66

2.66

$ratio(T_1) < ratio(T_5)$
Prefer T_1 over T_5

R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8 R_9 R_{10} R_{11} R_{12}

Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Greedy Algorithms

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

1.66

2.66

Proceed incrementally, picking the test case with the lowest *ratio* value for the uncovered requirements

R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8 R_9 R_{10} R_{11} R_{12}

Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Greedy Algorithms

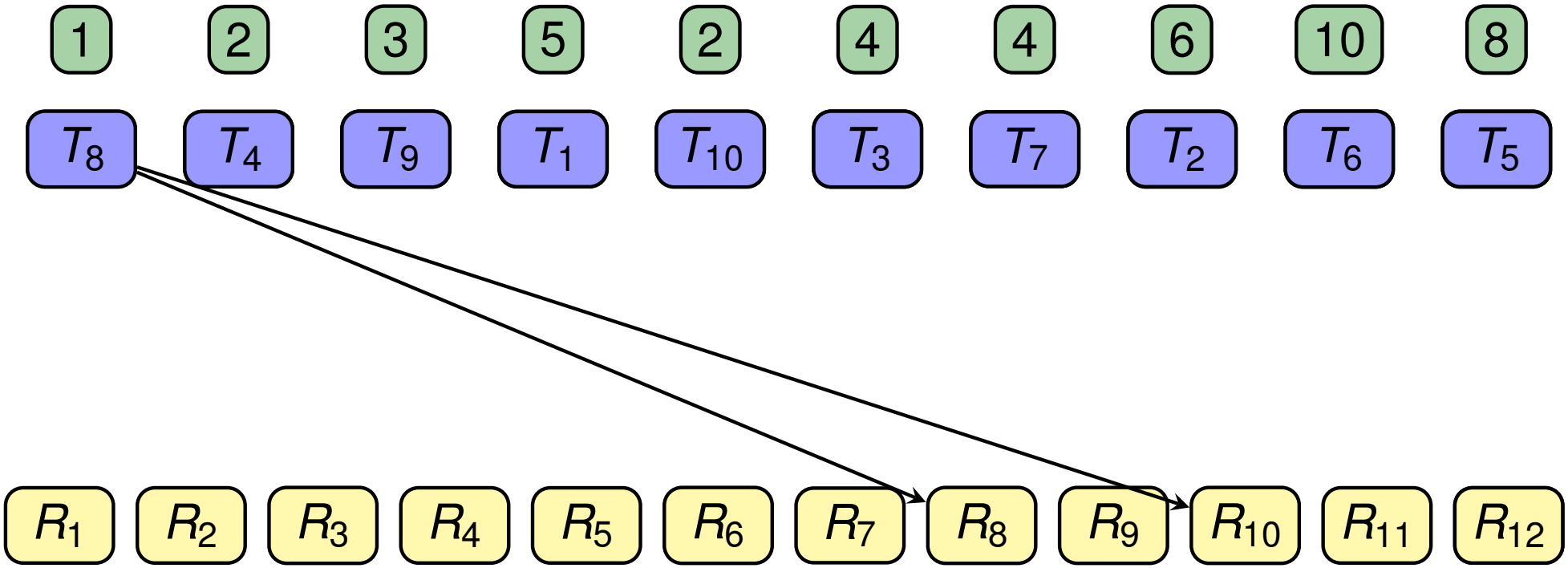
Test Suite $T = \langle T_8, T_4, T_9, T_1, T_{10}, T_3, T_7, T_2, T_6, T_5 \rangle$

T_8 T_4 T_9 T_1 T_{10} T_3 T_7 T_2 T_6 T_5

R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8 R_9 R_{10} R_{11} R_{12}

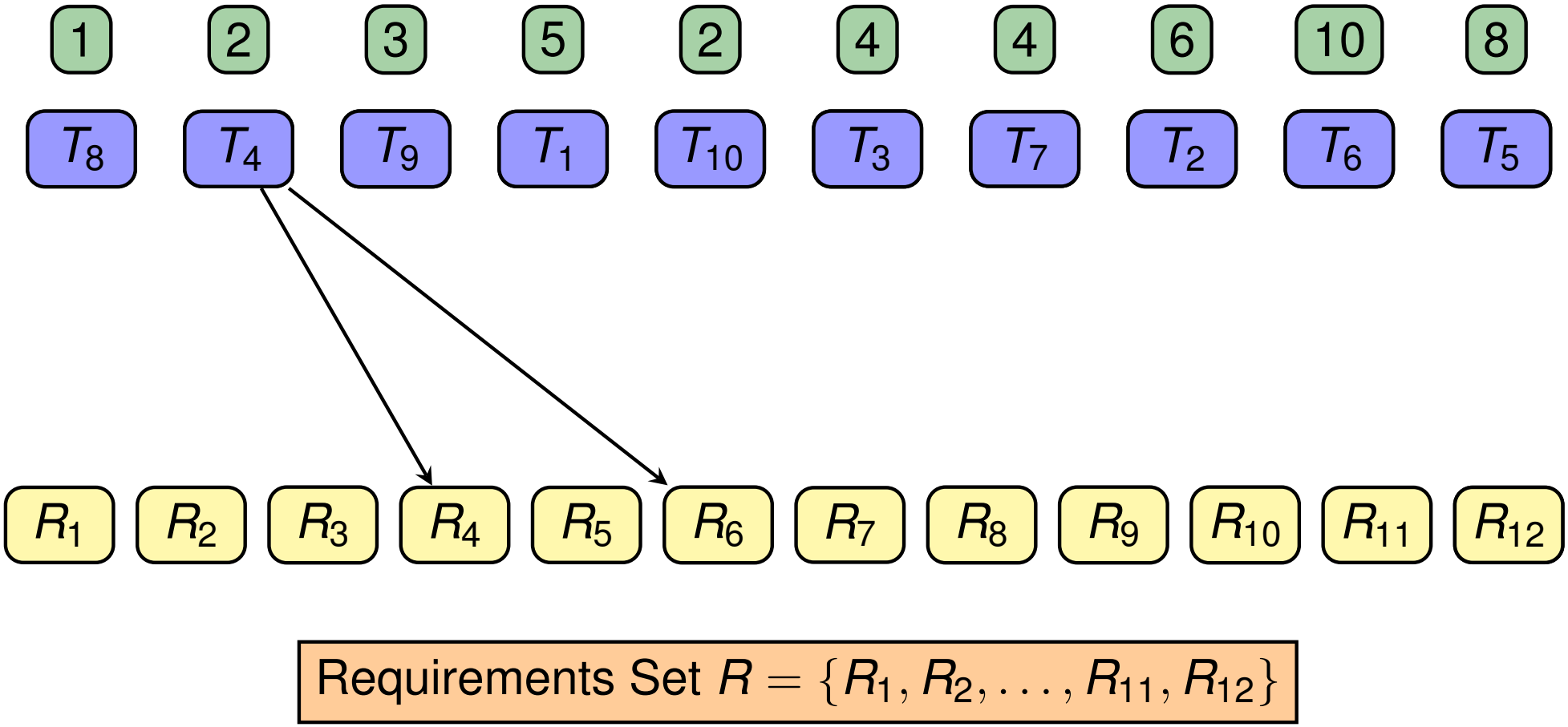
Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Greedy Algorithms



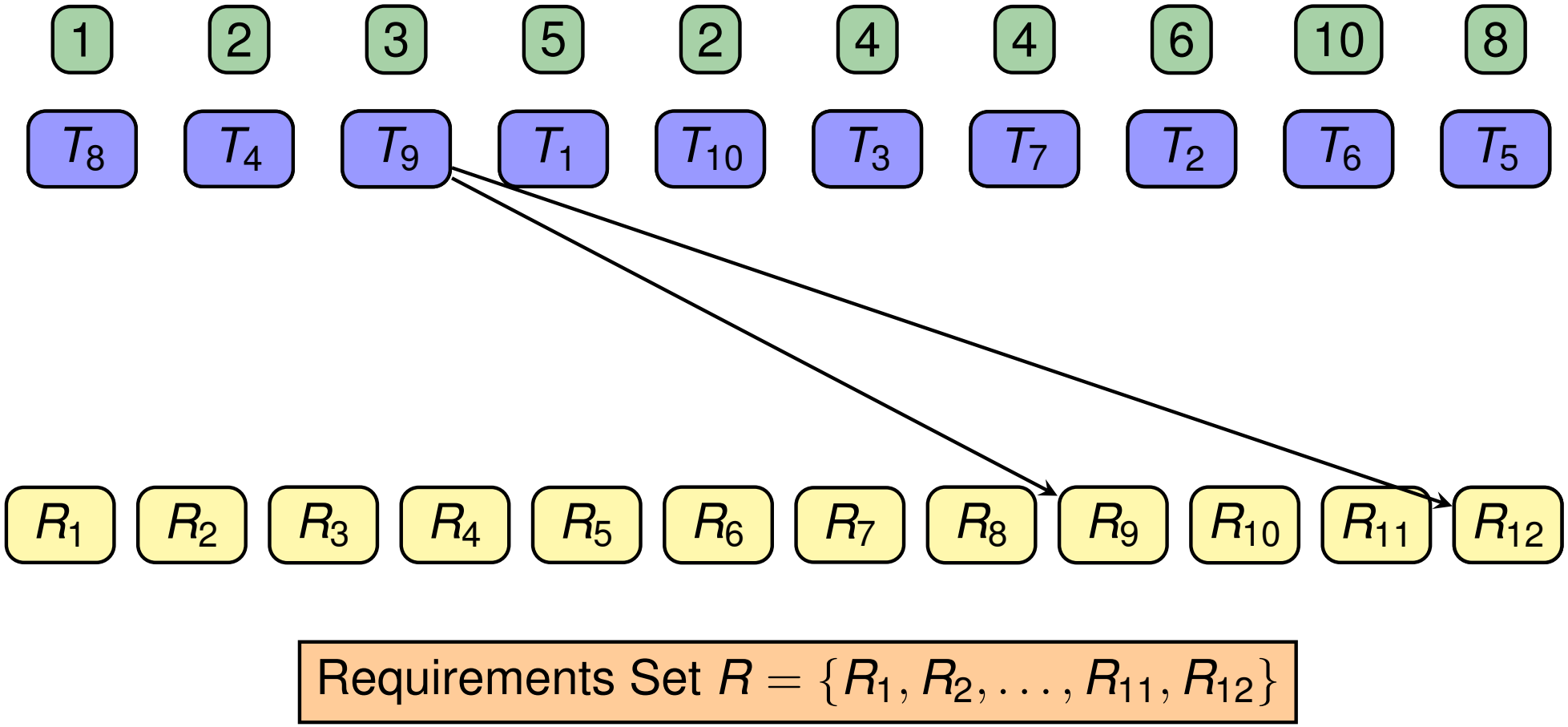
Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Greedy Algorithms



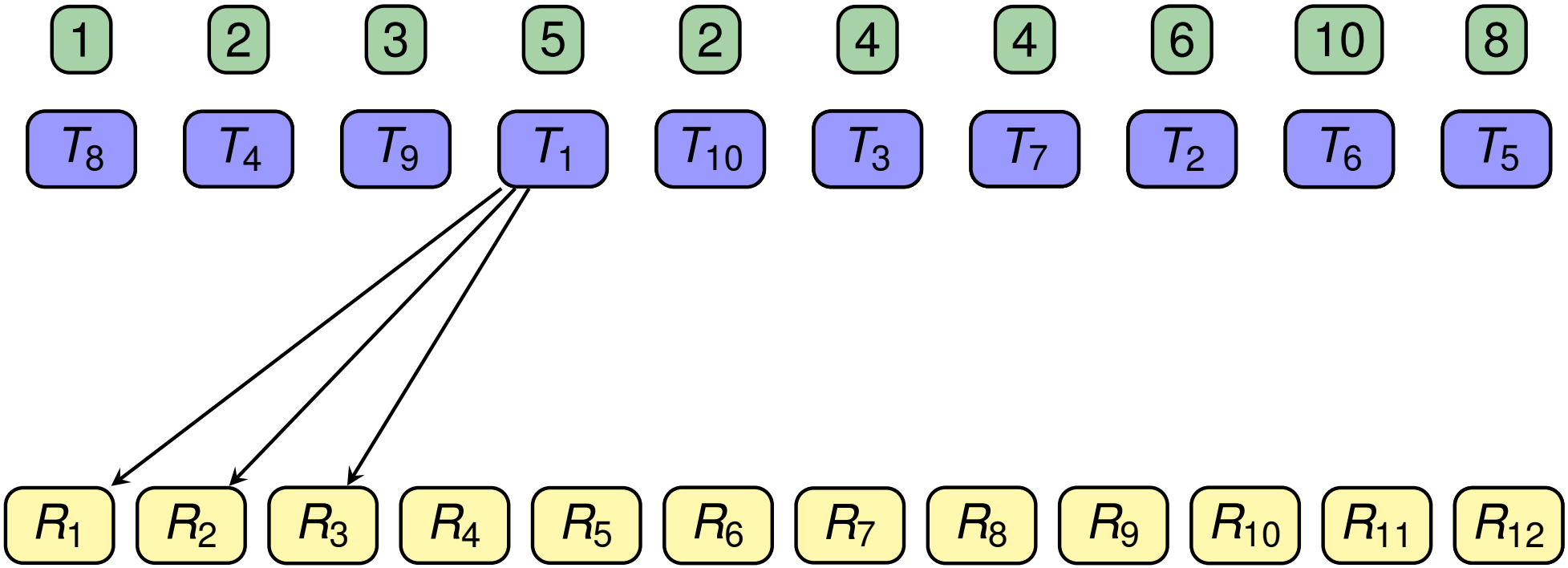
Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Greedy Algorithms



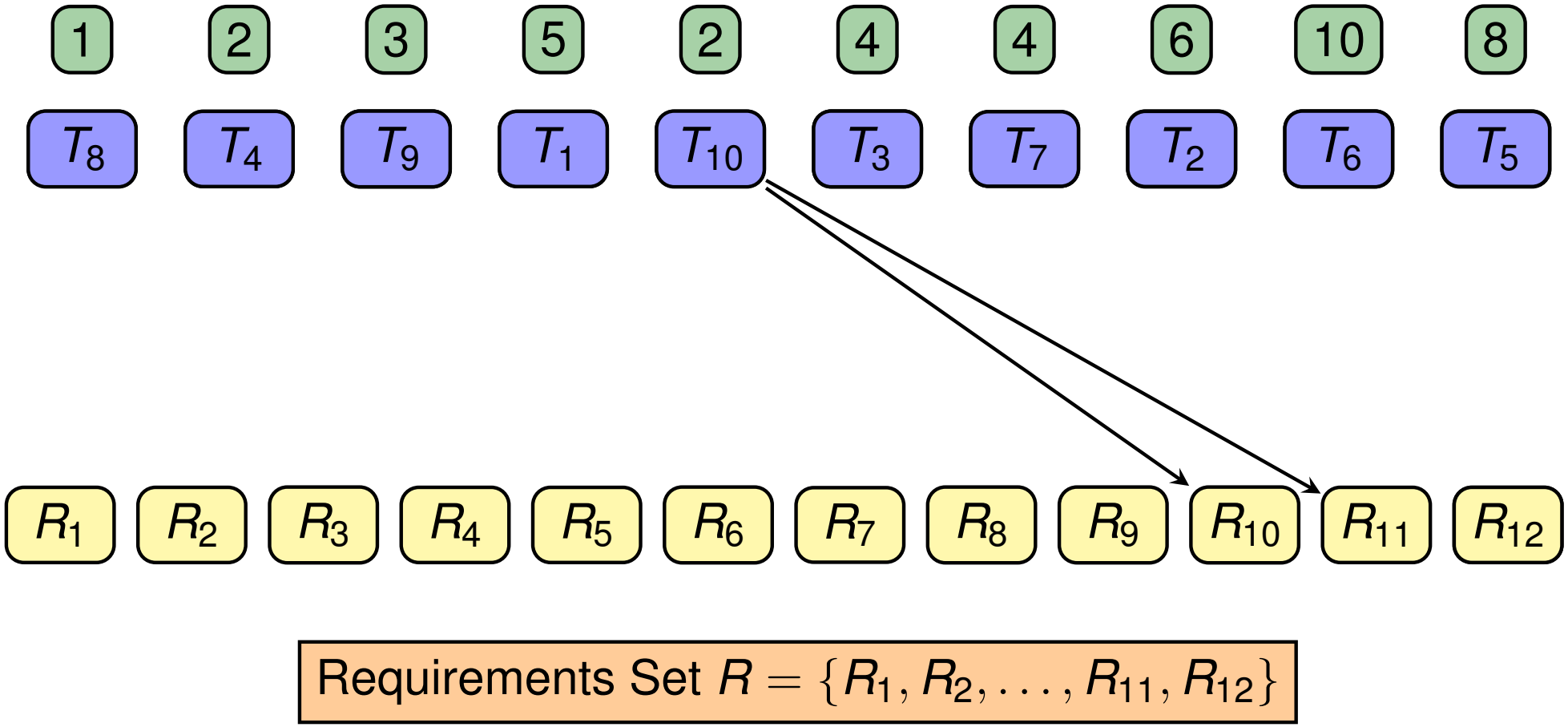
Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Greedy Algorithms



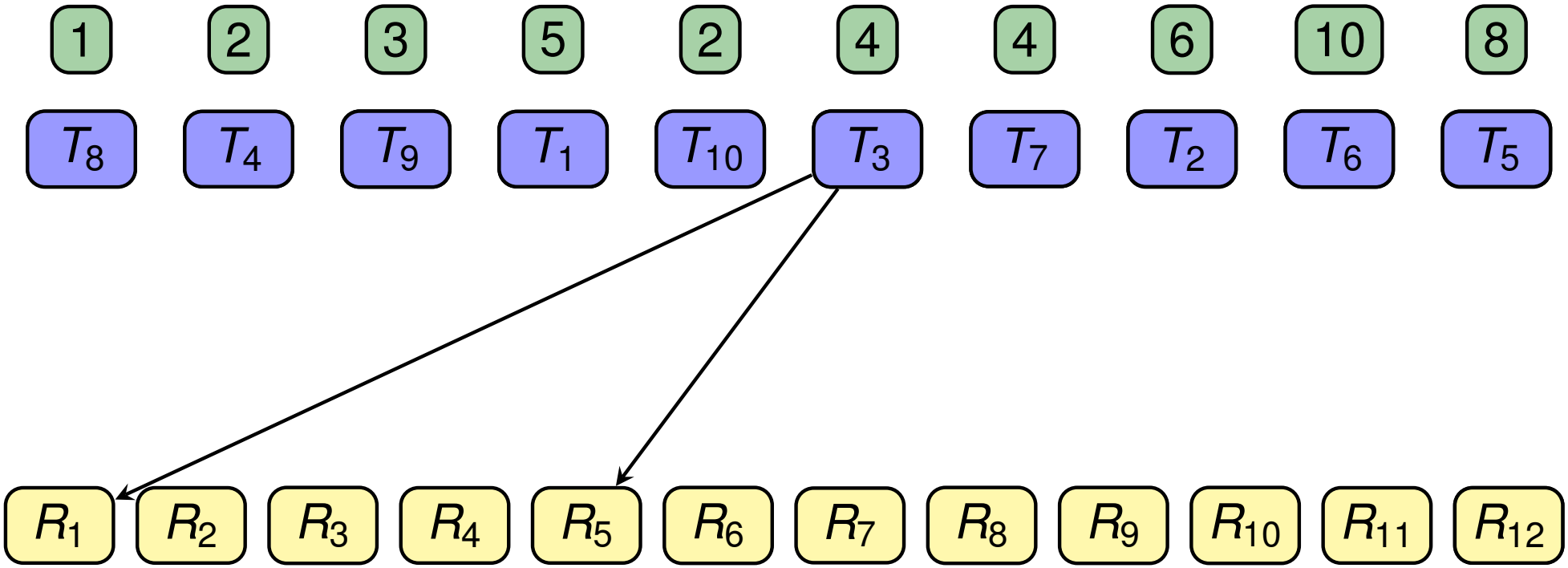
Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Greedy Algorithms



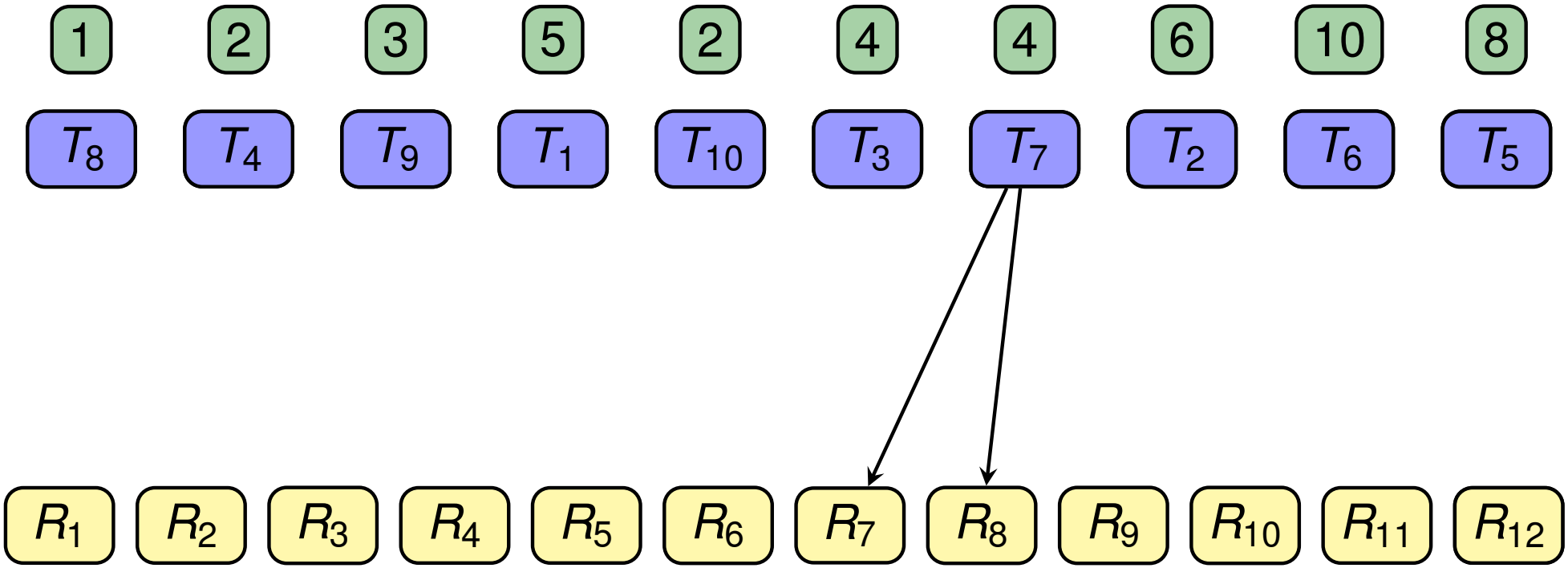
Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Greedy Algorithms



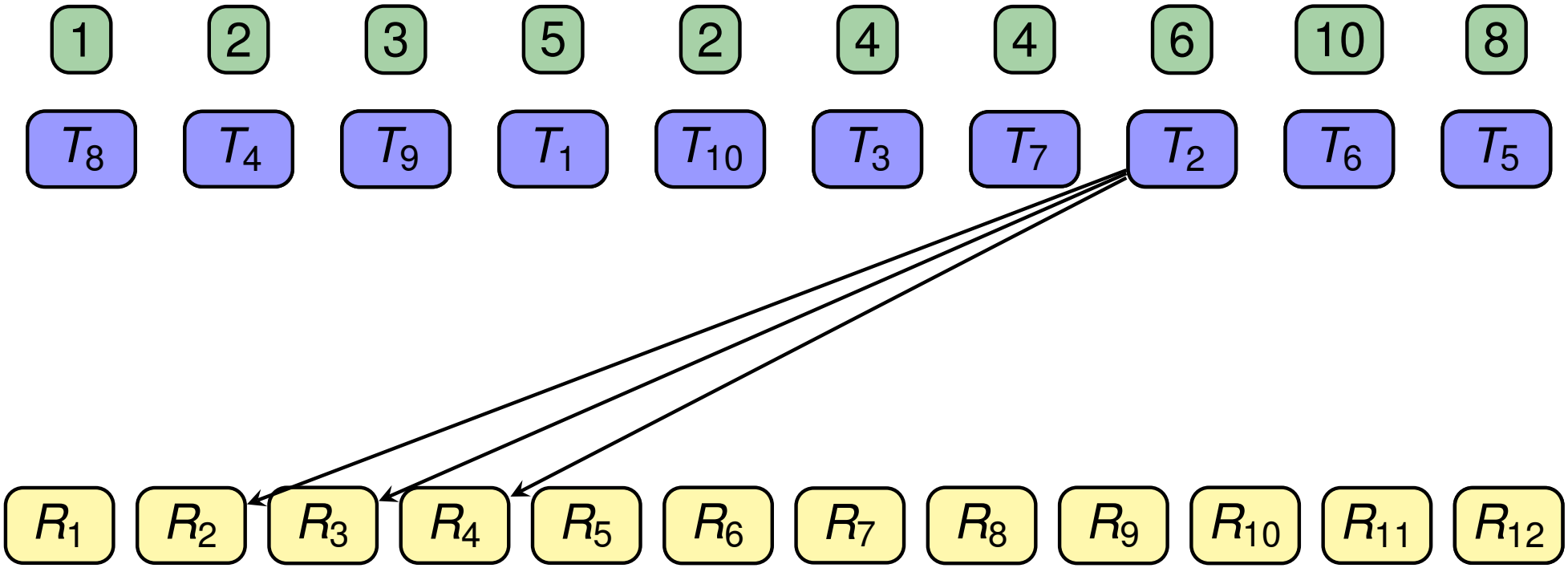
Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Greedy Algorithms



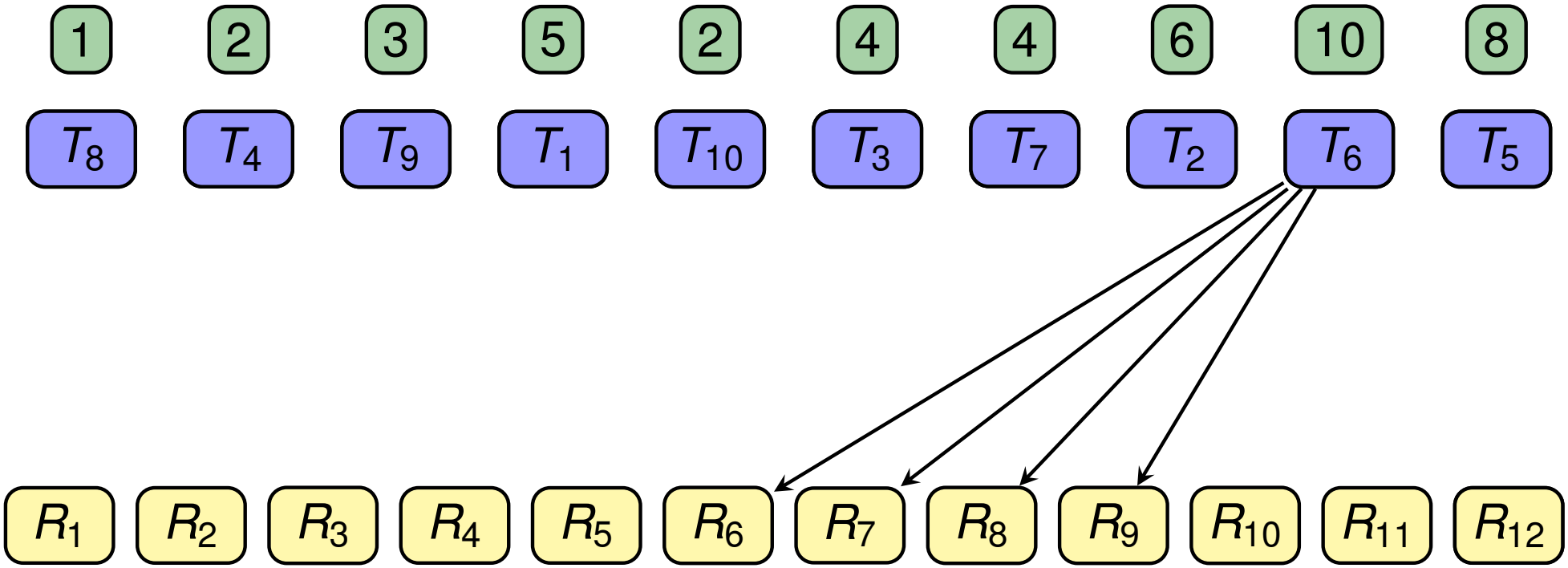
Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Greedy Algorithms



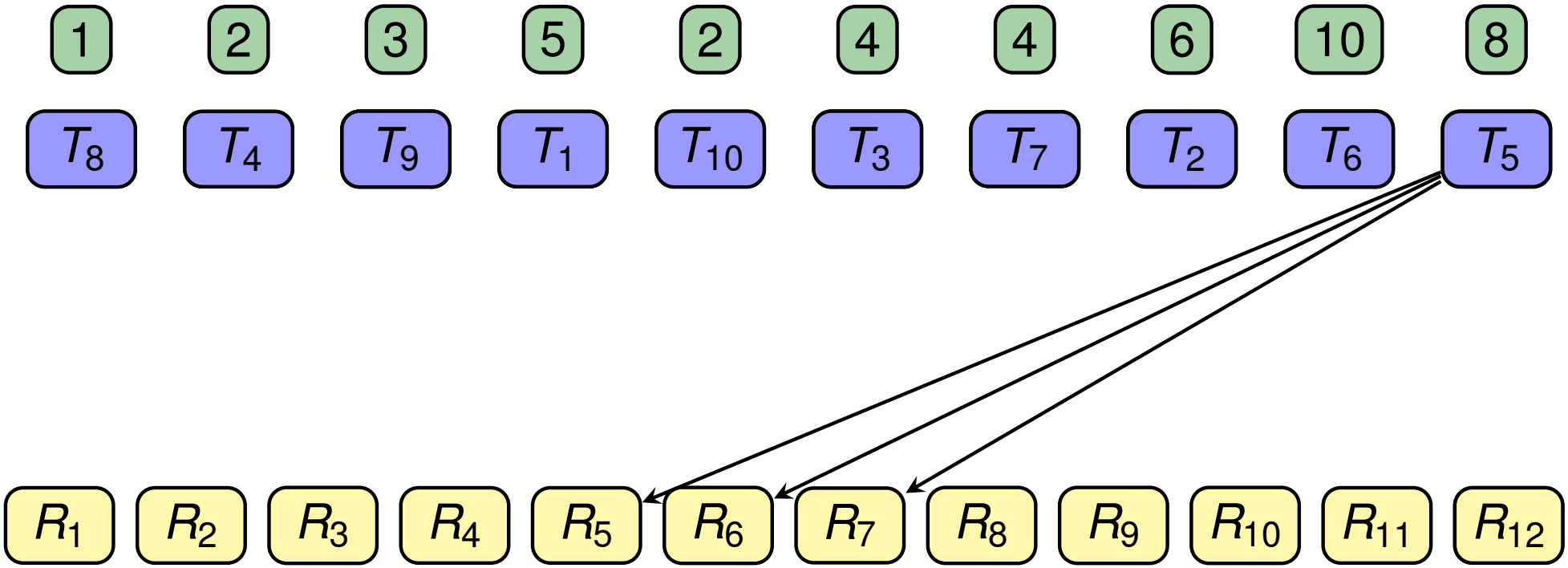
Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Greedy Algorithms



Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Greedy Algorithms



Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Greedy Algorithms

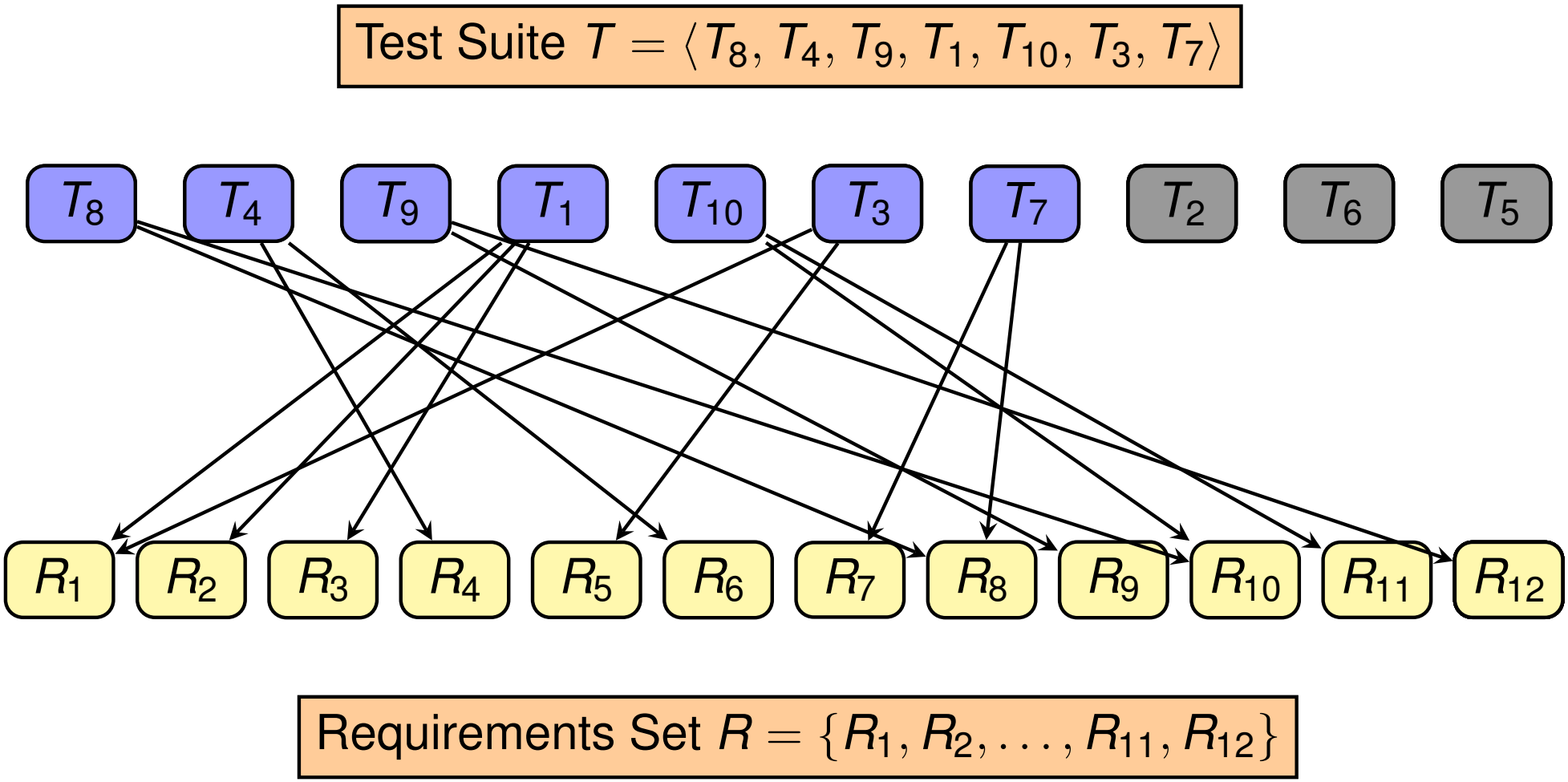
Test Suite $T = \langle T_8, T_4, T_9, T_1, T_{10}, T_3, T_7 \rangle$

T_8 T_4 T_9 T_1 T_{10} T_3 T_7 T_2 T_6 T_5

R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8 R_9 R_{10} R_{11} R_{12}

Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$

Greedy Algorithms



Empirical Results – Test Suite Reduction

Program	Rel.	Attr.	Rec.	Attr. Val.	All
RM (13)	(7, .46)	(7, .46)	(10, .30)	(9, .31)	(8.25, .37)
FF (16)	(7, .56)	(7, .56)	(11, .31)	(11, .31)	(9, .44)
PI (15)	(6, .60)	(6, .60)	(8, .70)	(7, .53)	(6.75, .55)
ST (25)	(5, .80)	(5, .76)	(11, .56)	(10, .60)	(7.75, .690)
TM (27)	(14, .48)	(14, .48)	(15, .45)	(14, .48)	(14.25, .47)
GB (51)	(33, .35)	(33, .35)	(33, .35)	(32, .37)	(32.75, .36)
All (24.5)	(12, .51)	(12.17, .50)	(14.67, .40)	(13.83, .44)	

- Reduction values range from .30 to .80
- Reduction level varies depending on interaction granularity
- How will the reduction of a test suite impact defect detection?

Empirical Results – Test Suite Reduction

Program	Rel.	Attr.	Rec.	Attr. Val.	All
RM (13)	(7, .46)	(7, .46)	(10, .30)	(9, .31)	(8.25, .37)
FF (16)	(7, .56)	(7, .56)	(11, .31)	(11, .31)	(9, .44)
PI (15)	(6, .60)	(6, .60)	(8, .70)	(7, .53)	(6.75, .55)
ST (25)	(5, .80)	(5, .76)	(11, .56)	(10, .60)	(7.75, .690)
TM (27)	(14, .48)	(14, .48)	(15, .45)	(14, .48)	(14.25, .47)
GB (51)	(33, .35)	(33, .35)	(33, .35)	(32, .37)	(32.75, .36)
All (24.5)	(12, .51)	(12.17, .50)	(14.67, .40)	(13.83, .44)	

- Reduction values range from .30 to .80
- Reduction level varies depending on interaction granularity
- How will the reduction of a test suite impact defect detection?

Conclusion

Conclusion

- Databases are widely used in real-world applications
- Database applications have complex state and structure
- Source code, database state, and relational schema evolve
- Prioritization techniques can increase effectiveness
- Reduction methods can improve the efficiency of testing

Future Work

- New empirical studies of database-aware regression testing
- Implement and release free and open source testing tools

Conclusion

Conclusion

- Databases are widely used in real-world applications
- Database applications have complex state and structure
- Source code, database state, and relational schema evolve
- Prioritization techniques can increase effectiveness
- Reduction methods can improve the efficiency of testing

Future Work

- New empirical studies of database-aware regression testing
- Implement and release free and open source testing tools

Regression Testing Techniques for Relational Database Applications

Gregory M. Kapfhammer

Department of Computer Science
Allegheny College
http://www.cs.allegheny.edu/~gkapfham/

Thank you for your attention!
I welcome your questions and comments.

ALLEGHENY COLLEGE