16.35

Aerospace Software Engineering

Reliability, Availability, and Maintainability
Software Fault Tolerance

Prof. Kristina Lundqvist
Dept. of Aero/Astro, MIT

Definitions

- Software reliability
  - The probability that a system will operate without failure under given conditions for a given time interval
  - Expressed on a scale 0 to 1
- Software availability
  - Probability that a system is functioning completely at a given instant in time, assuming that the required external resources are also available.
  - A system completely up and running has availability 1; on that is unusable has availability 0.

Definitions

- Software maintainability
  - Probability that, for a given condition of use, a maintenance activity can be carried out within a stated time interval and using stated procedures and resources.
  - Ranges from 0 to 1.

MIL-STD-1629A

- Catastrophic: A failure that may cause death or system loss
- Critical: a failure that may cause severe injury or major system damage that results in mission loss
- Marginal: a failure that may cause minor injury or minor system damage that results in delay, loss of availability, or mission degradation
- Minor: a failure not serious enough to cause injury or system damage, but that results in unscheduled maintenance or repair.

Failure Data

Interfailure Times (Read left to right, in rows)

3	30	113	81	115	9	2	91	112	15
138	50	77	24	108	88	670	120	26	114
325	55	242	68	422	180	10	1146	600	15
36	4	0	8	227	65	176	58	457	…
10	1071	6150	3321	1045	648	5485	1160	1864	4116

Type-1 uncertainty

Type-2 uncertainty

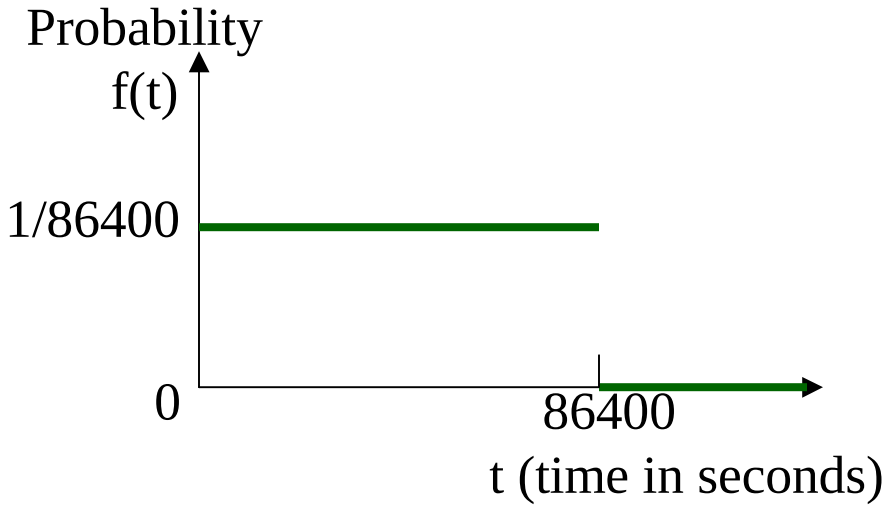
Measuring Reliability, Availability, and Maintainability

- MTTF
  - Average of the interfailure times
- MTTR
- MTBF
  - MTBF = MTTF + MTTR
- R = MTTF/(1 + MTTF)
- A = MTBF/(1 + MTBF)
- M = 1/(1 + MTTR)

Reliability Stability and Growth

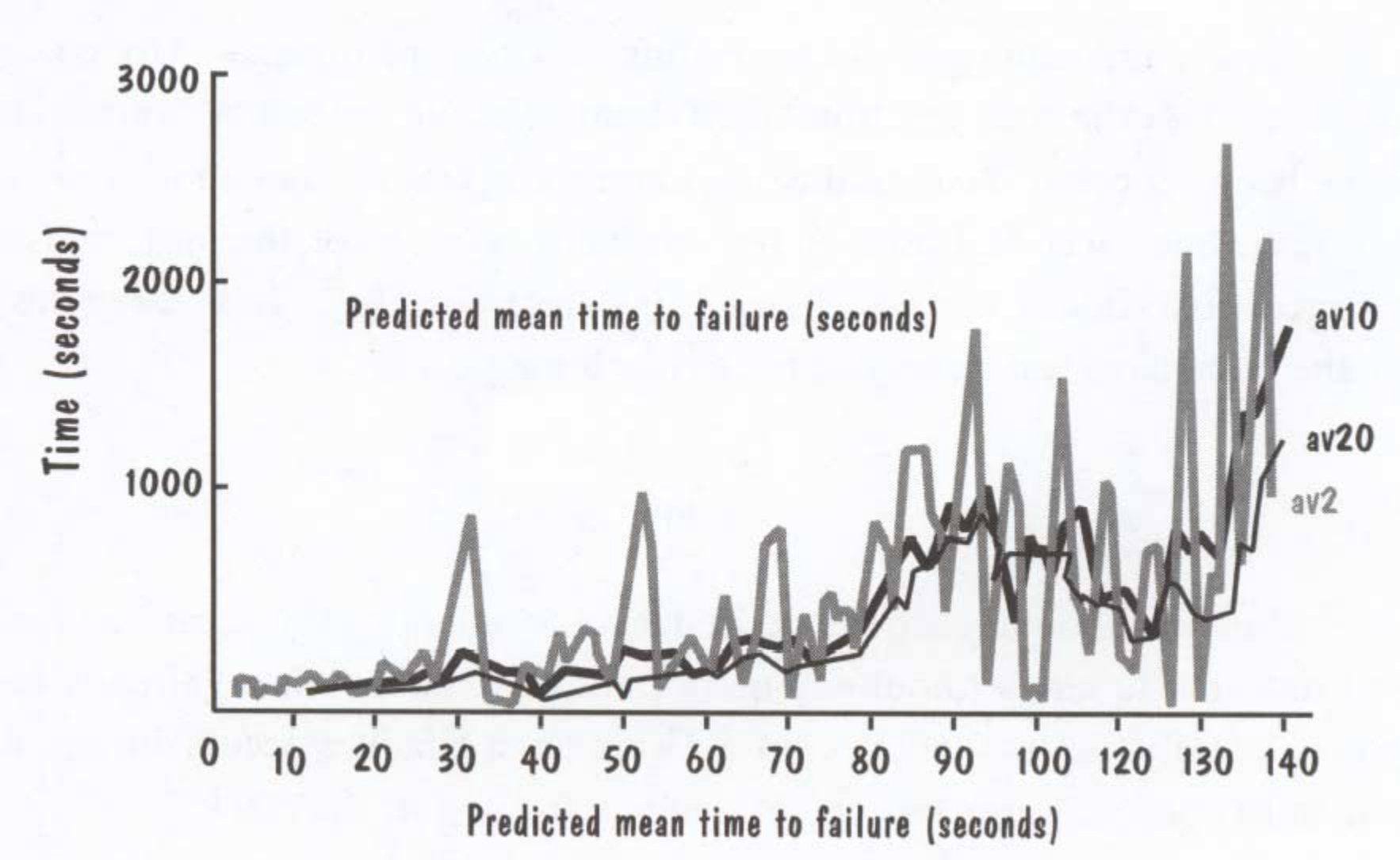
- Reliability stability
  - If the interfailure times stay the same
- Reliability growth
  - If they increase
- Probability density function
  - f(t)

$$\int_{t_2}^{t_1} f(t)\,dt$$



Reliability function R(t) = 1-F(t)

Predicting next Failure Times from Past History



Reliability Prediction

- The Jelinsky-Moranda Model (1972)
  - Assumes: no type-2 uncertainty
  - Assumes: fixing any fault contributes equally to improving the reliability

i	Mean Time to ith Failure	Simulated Time to ith Failure
1	22	11
2	24	41
3	26	13
4	28	4
5	30	30
6	33	77
7	37	11
8	42	64
9	48	54
10	56	34
11	67	183
12	83	83
13	111	17
14	167	190
15	333	436

Software Fault Tolerance

Fault vs. Failure

- How do faults occur?
- What is a failure?

- Faults represent problems that developers see
- Failures are problems that users or customers see

Handling Design Faults

- Prevention
- Removal
- Fault Tolerance
- Input Sequence Work Arounds

Measures of Software Quality

- Reliability is *"the probability of failure free operation of a computer program in a specified environment for a specified period of time"*, where failure free operation in the context of software is interpreted as adherence to its requirements

  -[Pressman 97].

  $$MTBF = MTTF + MTTR$$

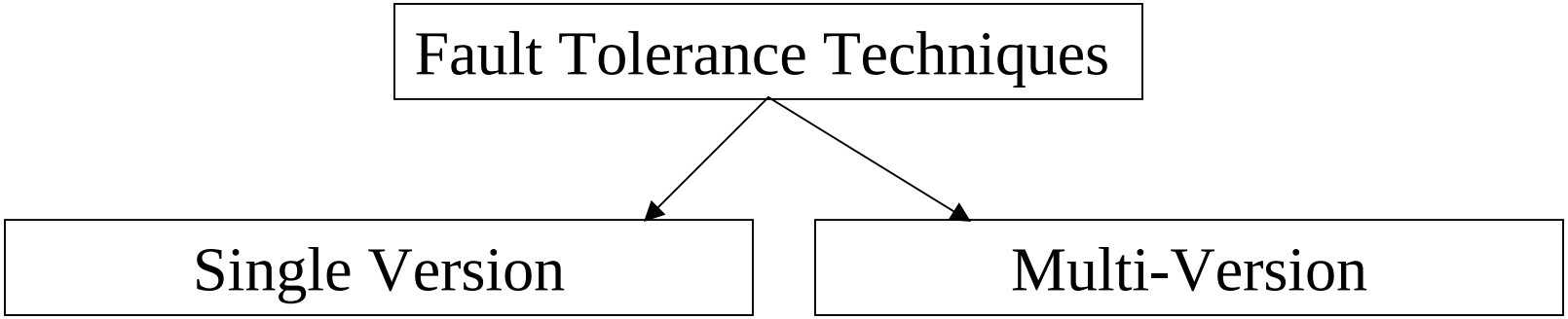
- Safety

DO-178B

"The goal of [software] fault tolerance methods is to include safety features in the software design or source Code to ensure that the software will respond correctly to input data errors and prevent output and control errors. The need for error prevention or fault tolerance methods is determined by the system requirements and the system safety assessment process."

Software Fault Tolerance

The function of software fault tolerance is to prevent system accidents (or undesirable events, in general), and mask out faults if possible.



Single Version Techniques

- Program structure and actions
- Error detection
- Exception handling
- Checkpoint and restart
- Process pairs
- Data diversity

Program Structure and Actions

- Modular Decomposition – Partitioning
  - Horizontal Partitioning
  - Vertical Partitioning
- Visibility & Connectivity
- System Closure
- Temporal Structuring

Error Detection

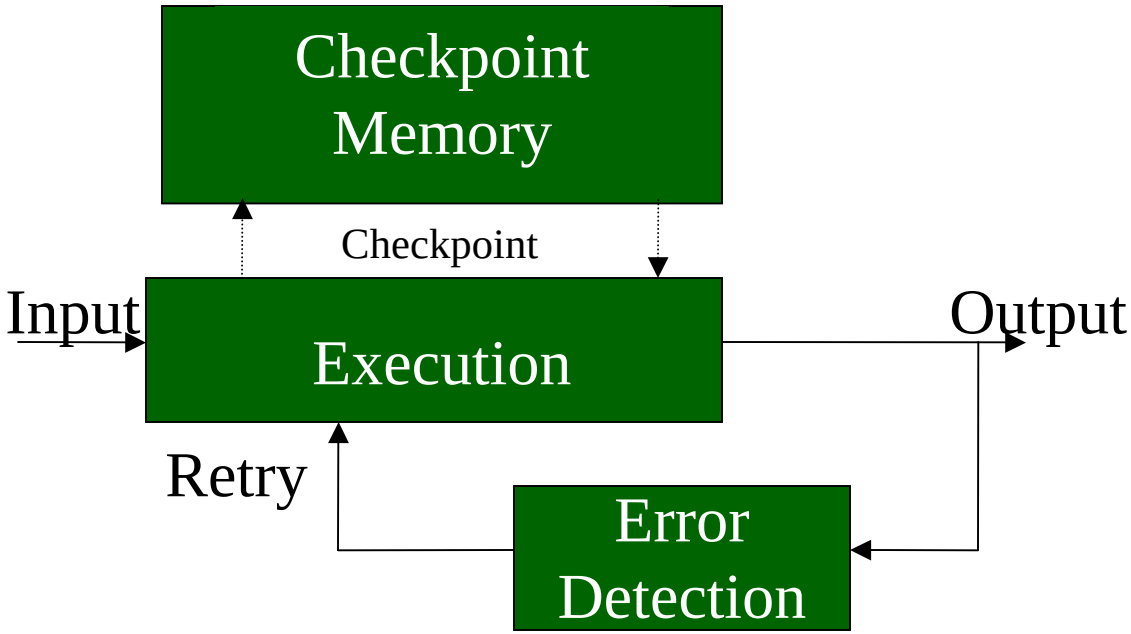
- Structured
  - Replication
  - Timing
  - Reversal
  - Coding
  - Reasonableness
  - Structural checks
- Ad-Hoc
  - Fault Trees

Exception Handling

- Interface Exceptions
- Local Exceptions
- Failure Exceptions

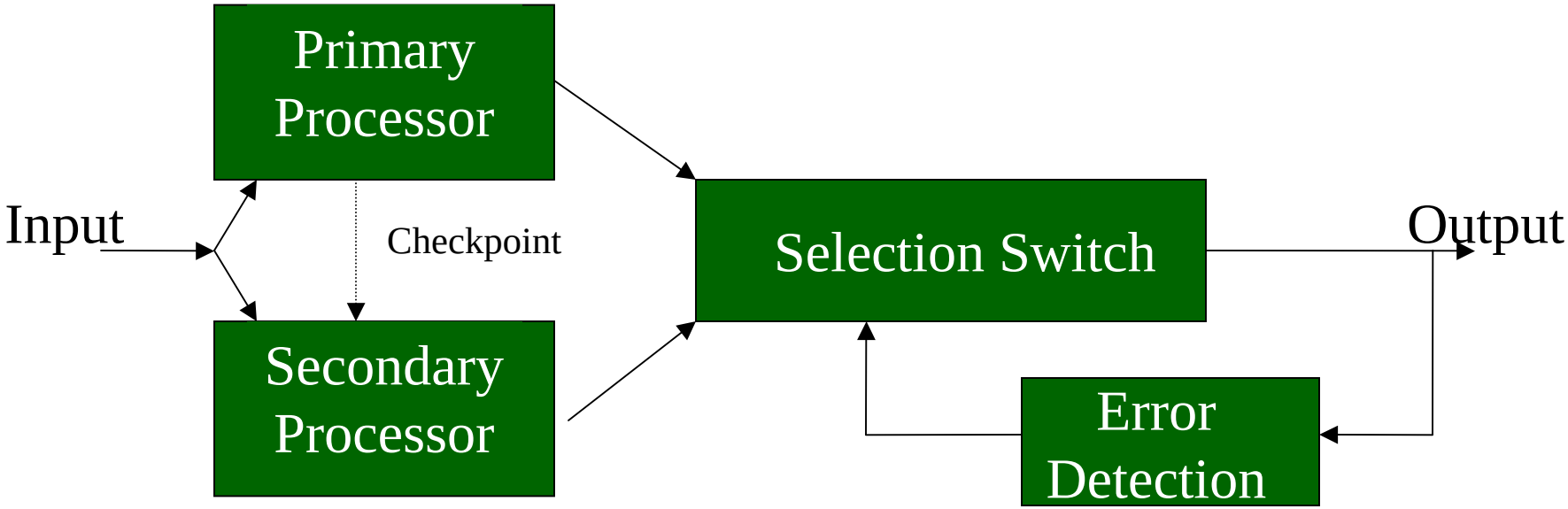
Checkpoint and Restart

- Checkpoint
- Restart
  - Static
  - Dynamic



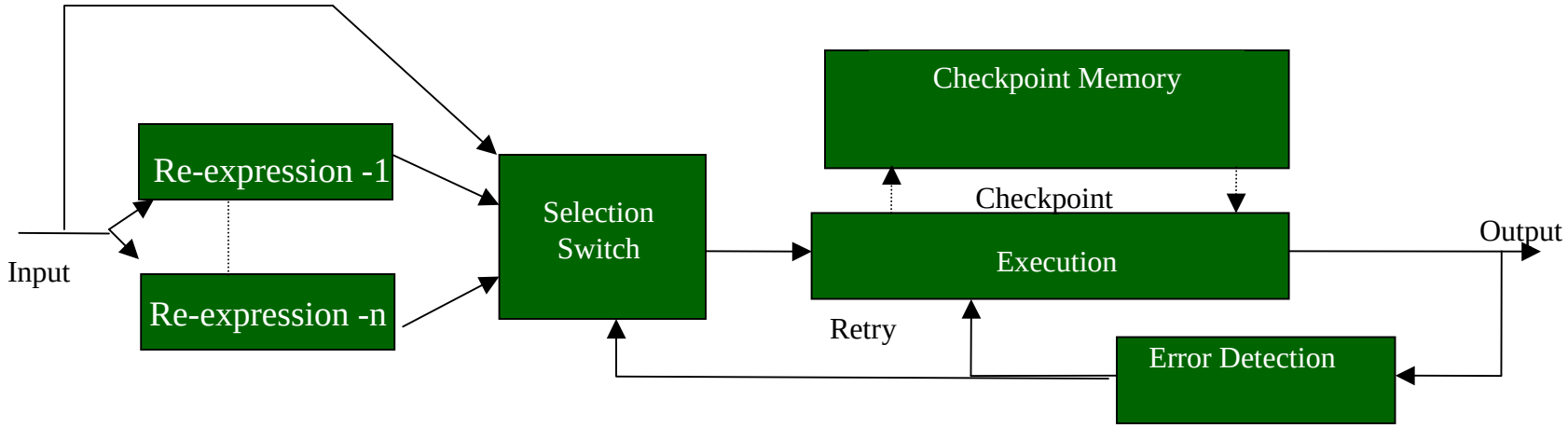
Process Pairs

- Two identical versions of software
- Separate processors

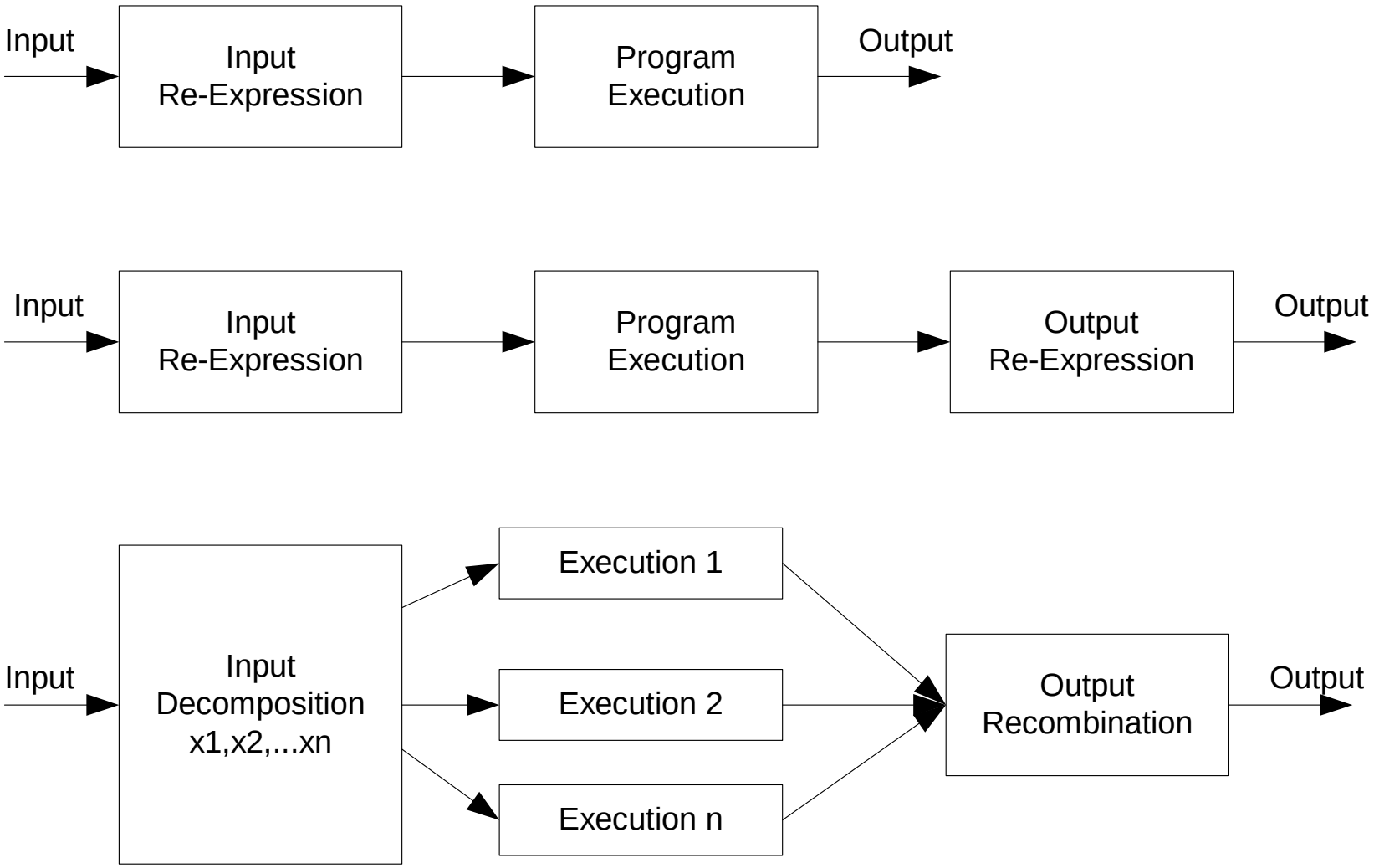


Data Diversity I

- Input Data Re-Expression
- Input Re-Expression with Post-Execution Adjustment
- Re-Expression via Decomposition and Recombination



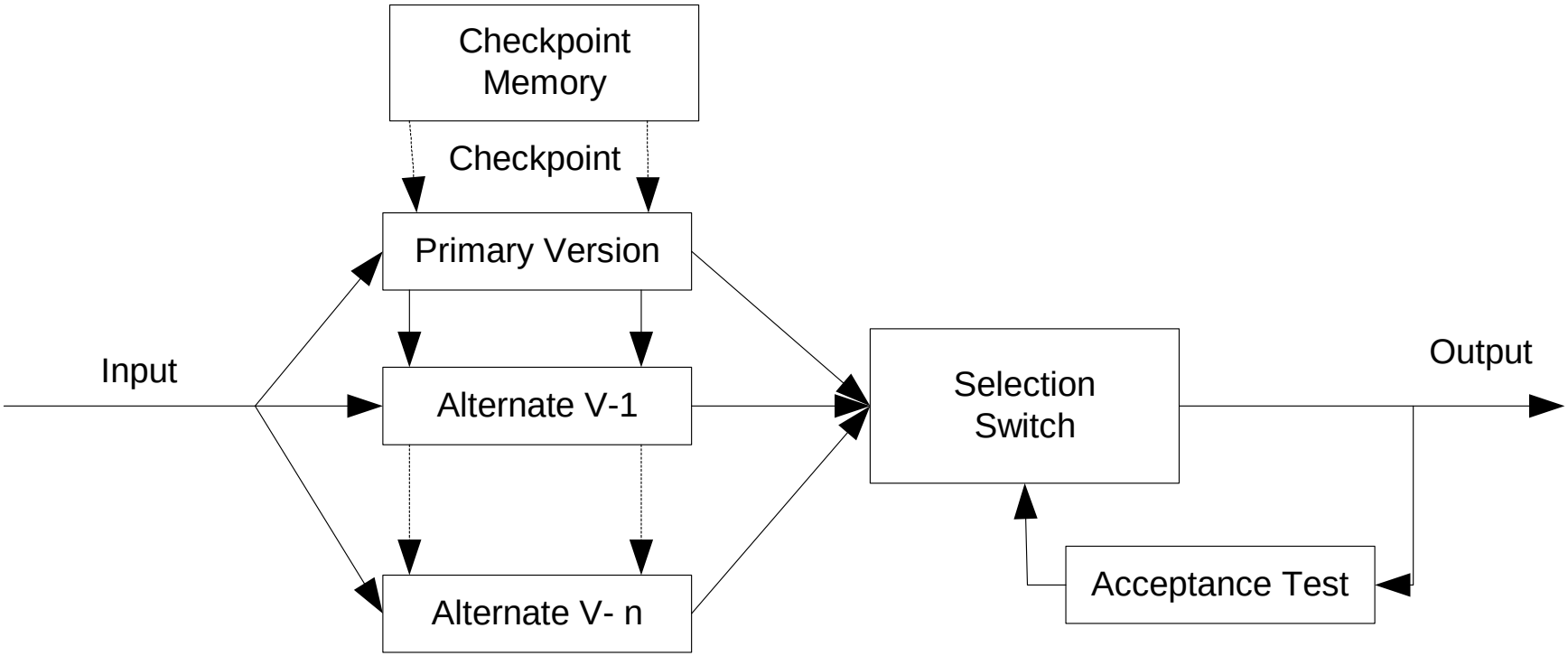
Data Diversity II



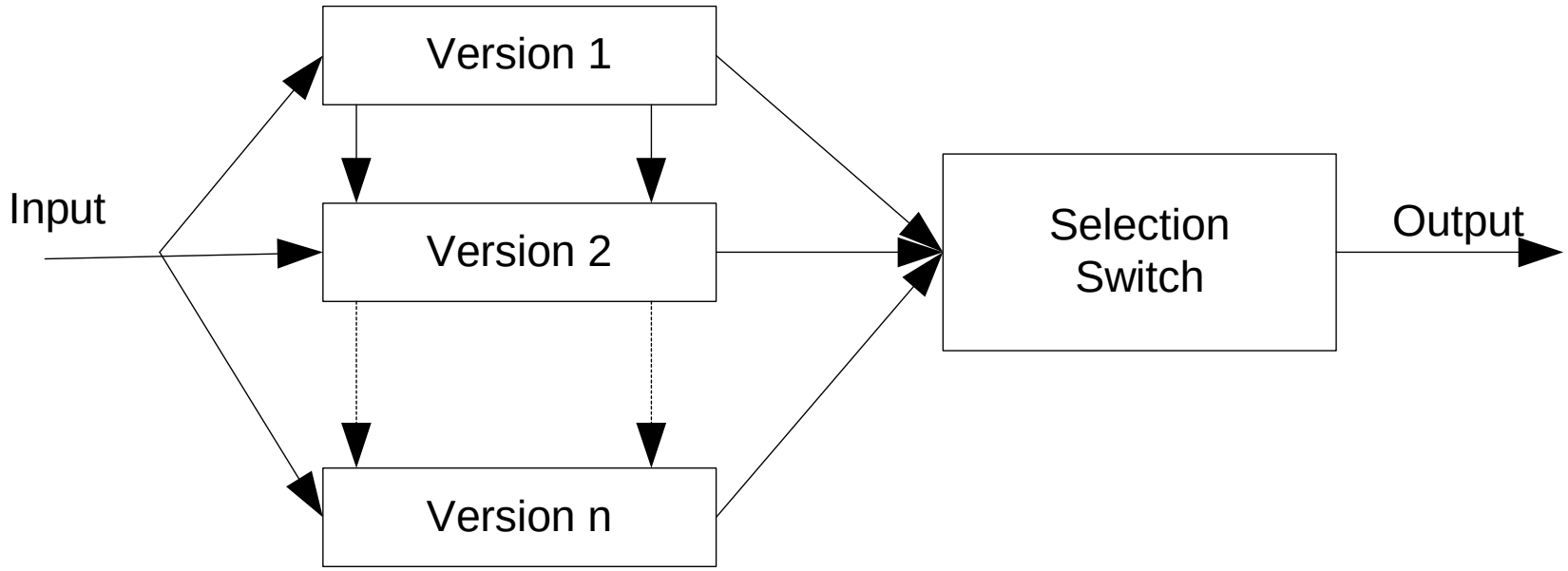
Multi-Version Techniques

- Recovery Blocks
- N-Version Programming
- N Self-Checking Programming
- Consensus Recovery Blocks
- t/(n-1) Variant Programming

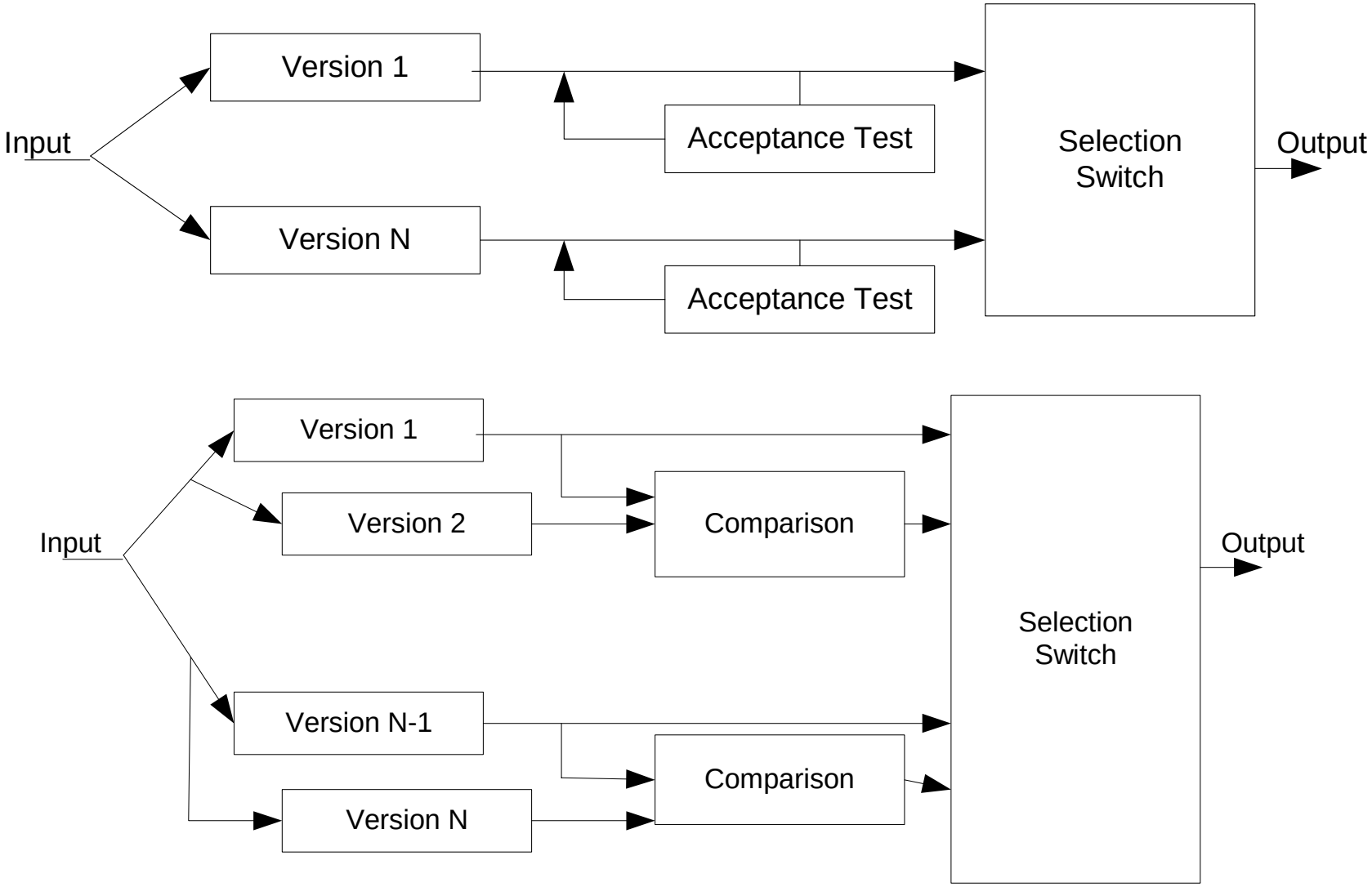
Recovery Blocks



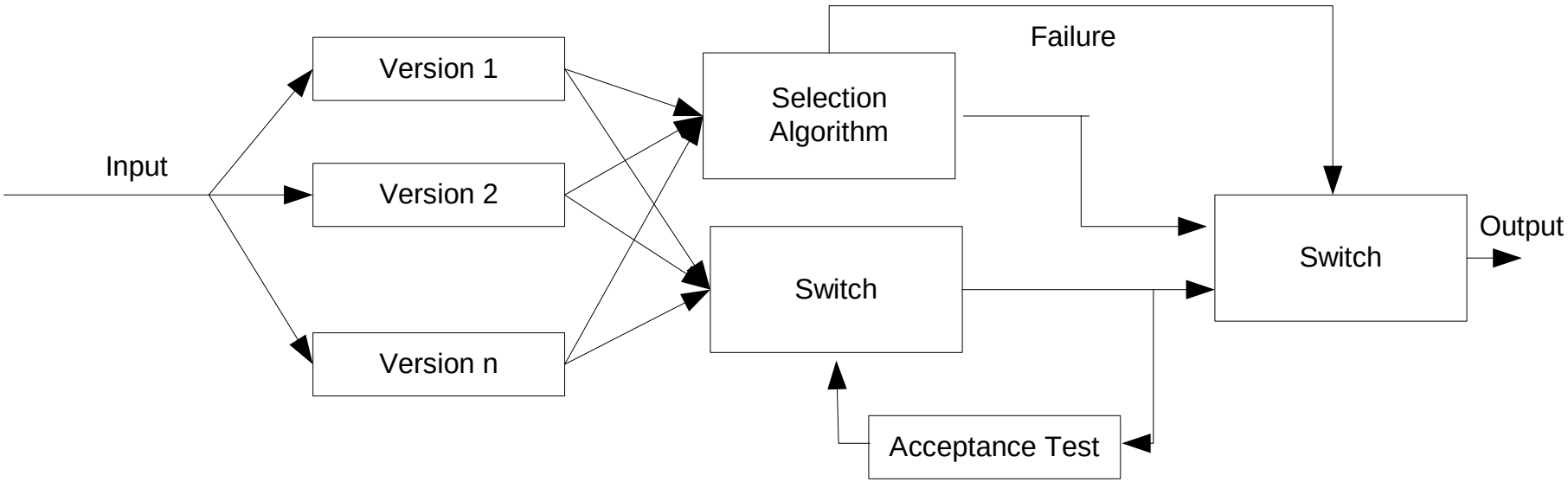
N-Version Programming



N Self-Checking Program



Consensus Recovery Blocks



t/(n-1) Variant Programming:
diagnosability measure to isolate the faulty units to a subset of size at most (n- 1) assuming there are at most t faulty units

- Web-based subject evaluations available, please fill them out

- Wednesday: practice questions for the final exam