

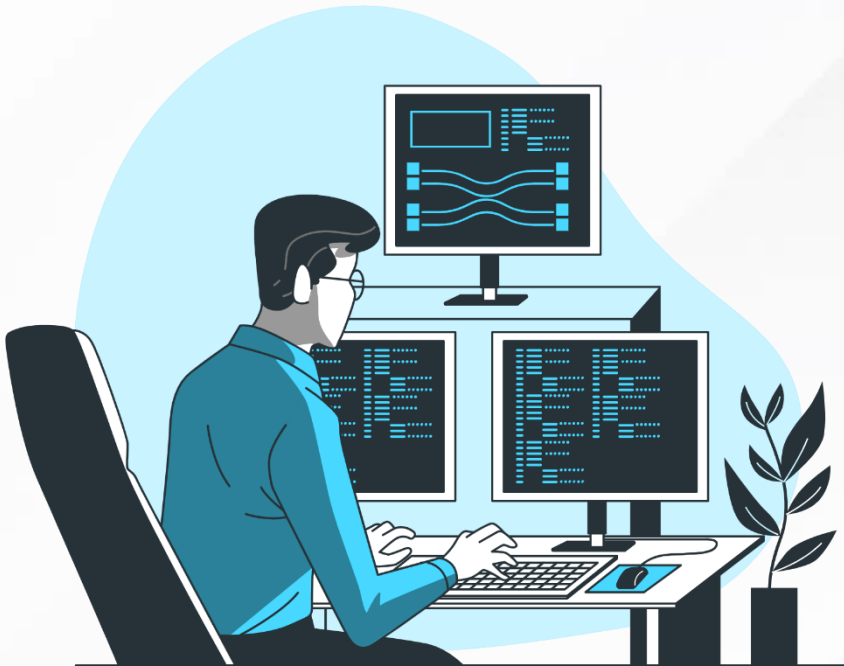
SQA

MANUAL TESTING



গতক্লাস পর্যন্ত আমরা শিখে এসেছি Boundary Value Analysis সম্পর্কে। এবার আমরা Decision Table Testing নিয়ে আলোচনা করবো।

Decision Table Testing



Decision table হলো এক ধরনের সফটওয়্যার এর টেস্টিং টুল যার মাধ্যমে টেস্ট এক্সপ্লোরেশনের হাত থেকে বাচাতে পারি টেস্ট ডিজাইন কে। এর মাধ্যমে অনেকগুলো শর্তের কন্সট্রিকশন করে ক্রিটিকাল ডিসিশন নেয়া যায়।

Isolated input condition এর সাথে **Equivalence class partitioning** এবং **Boundary analysis** কাজ করে থাকে। একটি **input condition** শুধুমাত্র **combination** করার সময়েই প্রভাব ফেলে থাকে সাধারণত। আগের উল্লেখ করা পদ্ধতিগুলো **dependencies and combinations** এর প্রভাবগুলো বিবেচনা করে থাকে না।

Equivalence class এর সকল **Combination** সেট ব্যবহার করলে অনেক বেশি সংখ্যক টেস্ট কেস তৈরী হয়। (**Test case explosion**)

Cause-and-effect graph এবং **decision table** ব্যবহার করে সম্ভাব্য আসা কন্ট্রিনেশনসংখ্যা সিস্টেমাটিক্যালি সাবসেটে কমিয়ে আনা যায়।

Business Case

Online Banking

একজন ইউজার নিজেকে একাউন্ট নাম্বার ও পিন দিয়ে পোর্টালে লগ ইন করে। তার কাছে যদি পর্যাপ্ত কভারেজ (টাকা) থাকে তাহলে সেটা থেকে অন্যজনের একাউন্টে ট্রান্সফার করতে পারে। এটি করার জন্য তাকে প্রাপকের সঠিক তথ্য ও ভ্যালিড TAN (OTP) দিতে হবে।

	T01	T02	T03	T04	T05	T06	T07	T08
Enough coverage	Yes	Yes	Yes	Yes	No	No	No	No
Correct recipient	Yes	Yes	No	No	Yes	Yes	No	No
Valid TAN	Yes	No	Yes	No	Yes	No	Yes	No
Do transferal	X							
Deny transferal	x		x	x	x	x	x	x
Request again TAN		x						

	T01	T02	T03	T04	T05	T06	T07	T08
Enough coverage	Yes	Yes	Yes	Yes	No	No	No	No
Correct recipient	Yes	Yes	No	No	Yes	Yes	No	No
Valid TAN	Yes	No	Yes	No	Yes	No	Yes	No
Do transferal	X							
Deny transferal	x		x	x	x	x	x	x
Request again TAN		x						

T05 –T08 পর্যন্ত সবগুলো কেসই পর্যাপ্ত ব্যালেন্স নেই। তাই এই কয়টা কেসকে আমরা একটি কেস হিসেবে ধরে নিয়ে টেস্ট কেসের সংখ্যা কমিয়ে আনতে পারি।

	T01	T02	T03	T04	T05
Enough coverage	Yes	Yes	Yes	Yes	No
Correct recipient	Yes	Yes	No	No	Yes
Valid TAN	Yes	No	Yes	No	Yes
Do transferal	X				
Deny transferal	x		x	x	x
Request again TAN		x			

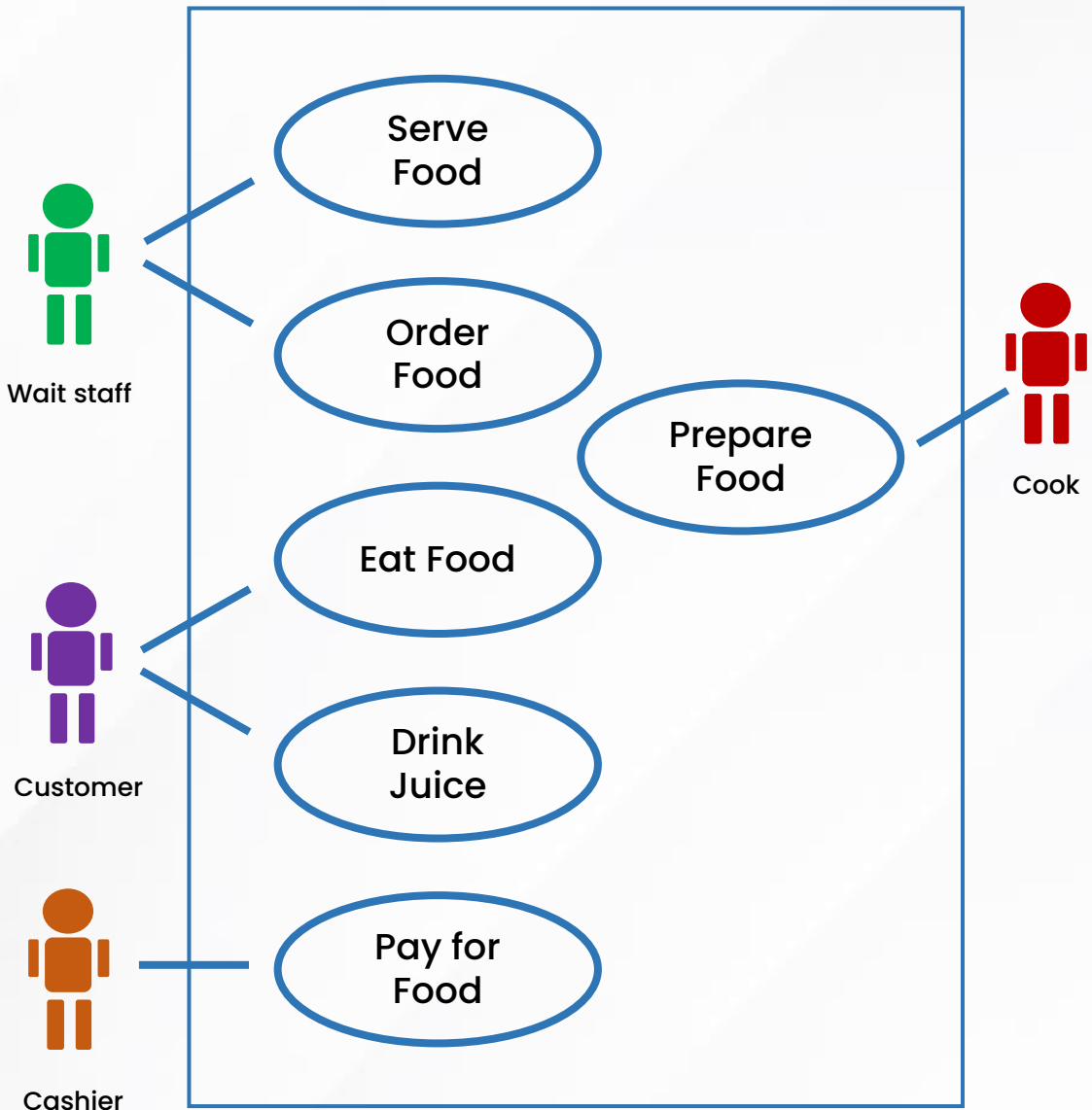
	T01	T02	T03	T04	T05
Enough coverage	Yes	Yes	Yes	Yes	No
Correct recipient	Yes	Yes	No	No	Yes
Valid TAN	Yes	No	Yes	No	Yes
Do transferal	X				
Deny transferal	x		x	x	x
Request again TAN		x			

T03 ও T04 টেস্ট কেসে দেখা যাচ্ছে যে রেসিপিয়ার সঠিক তথ্য নেই তাই এই দুটি নেই কেসকে আমরা একটি কেস হিসেবে ধরে নিয়ে টেস্ট কেসের সংখ্যা কমিয়ে আনতে পারি।

	T01	T02	T03	T05
Enough coverage	Yes	Yes	Yes	No
Correct recipient	Yes	Yes	No	Yes
Valid TAN	Yes	No	Yes	Yes
Do transferal	X			
Deny transferal	x		x	x
Request again TAN		x		

এই উদাহরনে,

- প্রতিটি টেবিল কলাম দিয়ে একেকটি টেস্ট কেস বোঝায়।
- একটি একশন বা ইফেক্ট নির্ধারন করা হয়।
- প্রতিটি কন্ডিশনের কন্ডিশনেশন একেকটি ডিসিশন টেবিলের কলাম (**Test case**) বুঝায়।
- ভিন্ন প্রভাব ফেলে এমন একই রকম কারনগুলোকে একত্র করে একটা একক টেস্ট কেস বানানো যায়।



Use Case Design

পাশের ডায়াগ্রামটি একটি সাধারণ রেস্টুরেন্ট সিস্টেমের কার্যকারিতাকে বর্ণনা করে।

Use Case গুলো এখানে বৃত্ত দিয়ে এবং এক্টরদের স্টিক ফিগার দিয়ে বোঝানো হয়েছে।

- এখানে কাস্টমার খাবার খেতে পারে। খাবার আর পানীয় এর জন্য টাকা দিতে পারে।
- শুধুমাত্র বাবুচি খাবার রান্না করতে পারে। আরো
- দেখা যাচ্ছে যে এখানে কাস্টমার আর কেশিয়ার দুইজন ই খাবারের পেমেন্টের **Use Case** এর সাথে সম্পর্কিত।
- এখানে বক্সটি দিয়ে রেস্টুরেন্ট সিস্টেম টা মডেল কেবল এক্টররা না, এটা বোঝায়।

টেস্ট অবজেক্টের **Use Case** থেকে সরাসরিভাবে টেস্ট কেস বানানো হয়।

- টেস্ট অবজেক্টকে দেখা হয় একটি সিস্টেম হিসেবে যেটি এক্টরদের সাথে কাজ করছে।
- একটা **Use Case** সকল এক্টরদের ইন্টারেকশনের কারনে সিস্টেমের যে শেষ ফলাফল আসে তা ব্যখ্যা করে।
- প্রত্যেকটি **Use Case** এর প্রিকন্ডিশন থাকে যেগুলো পূরন করা প্রয়োজন **Use Case (Test Case)** গুলো সফলভাবে পরিচালনা করার জন্য।
- প্রত্যেকটি **Use Case** এর পোস্টকন্ডিশন থাকে যা **Use Case (Test Case)** পরিচালনা করার পর সিস্টেমকে ব্যখ্যা করে।

Use Case হলো **Unified Modeling Language (UML)** এর উপাদান **Use Case** ডায়াগ্রাম হলো **UML** এ ব্যবহৃত ১৩ টি ভিন্ন ডায়াগ্রামের একটি। সকল **Use Case** ডায়াগ্রাম বিহেভিয়ার ব্যখ্যা করে। এটি কোনো ঘটনার ক্রম ব্যখ্যা করে না। এটি ইউজারের দিক থেকে সিস্টেমের প্রতিক্রিয়াকে দেখিয়ে থাকে।

UC 001	Use Case Name	
Dependencies		
Description		
Precondition		
Ordinary Sequence	Step	Action
	1	
	2	
	3	
Post Condition		
Alternative Sequence	Step	Action
	1	
	2	
Comments		

Structure based or White Box Technique

White box

1. Statement Coverage
2. Branch Coverage
3. Condition Coverage
4. Path Coverage



Main type of Coverage

Statement Coverage :

স্টেটমেন্ট কভারেজ হলো মূলত টেস্ট কেসে সোর্স কোডে থাকা স্টেটমেন্টের মধ্যে কতগুলো কভার করা হলো তার পারসেন্টেজ। এটি মডিউল, ক্লাস, মেনু পয়েন্ট ইত্যাদিতে ব্যবহার করা যেতে পারে।

Decision/ Branch Coverage :

টেস্ট কেসে ডিসিশনগুলোর কতগুলো ব্রাঞ্চ এক্সিকিউট করা হয়েছে তার পারসেন্টেজ হলো ডিসিশন বা ব্রাঞ্চ কভারেজ।

Condition Coverage :

টেস্ট কেসের সকল একক কন্ডিশনের আউটকাম যা ডিসিশনের আউটকাম কে প্রভাবিত করছে তার পারসেন্টেজ হলো কন্ডিশন কভারেজ।

কন্ডিশন কভারেজ অনেক মাত্রায় প্রকাশিত হতে পারে যেমনঃ সিঙ্গেল, ডাবল, মাল্টিপল আর মিনিমাল মাল্টিপল কন্ডিশন কভারেজ।

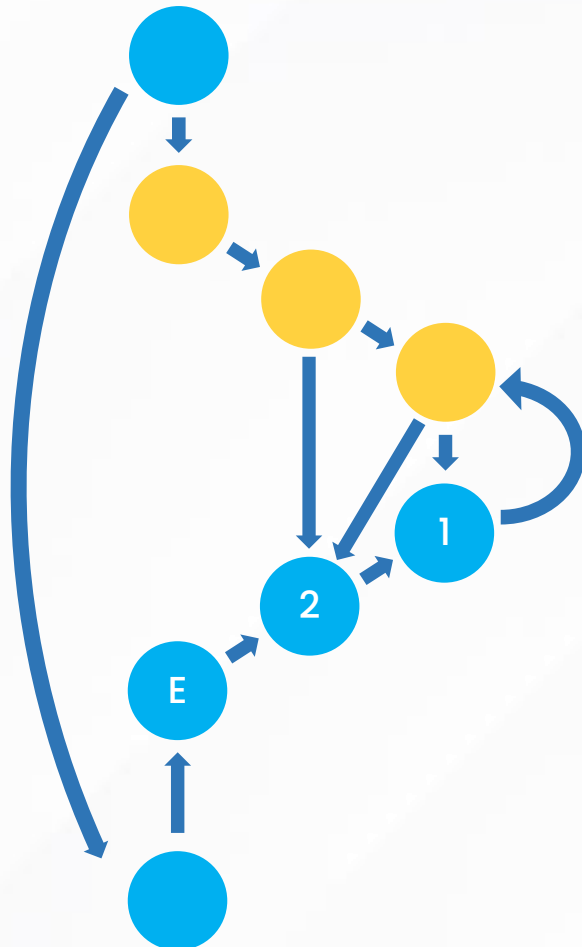
Path Coverage :

টেস্ট কেসের এক্সিকিউশন পাথের পারসেন্টেজ।

Statement Coverage :

Example 01 :

```
If (i>0) {  
  If (j>10) {  
    While (k>10) {  
      Do 1st Task ...  
    }  
    Do 2nd Task  
  }  
  Do End Task  
}
```



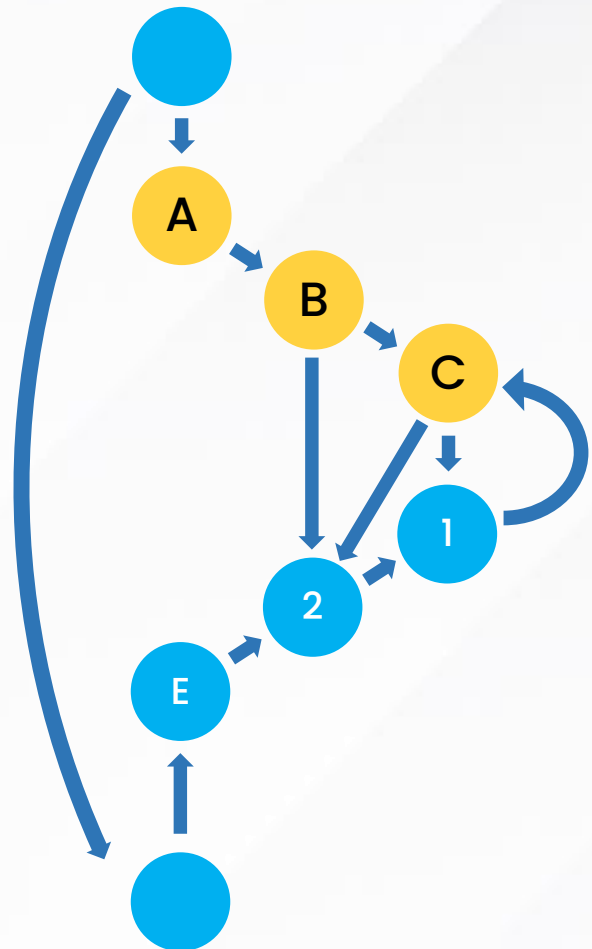
Statement Coverage :

Example 01:

প্রোগ্রামটি কন্ট্রোল ফ্লো দ্বারা প্রকাশিত হয়েছে বলে ধরা যাক। এতে দুইটি **if** স্টেটমেন্ট আছে এবং একটি লুপ (**do while**) আছে। দ্বিতীয় **if** স্টেটমেন্টের ভিতর।

এখানে তিনটি ভিন্ন **route** আছে।

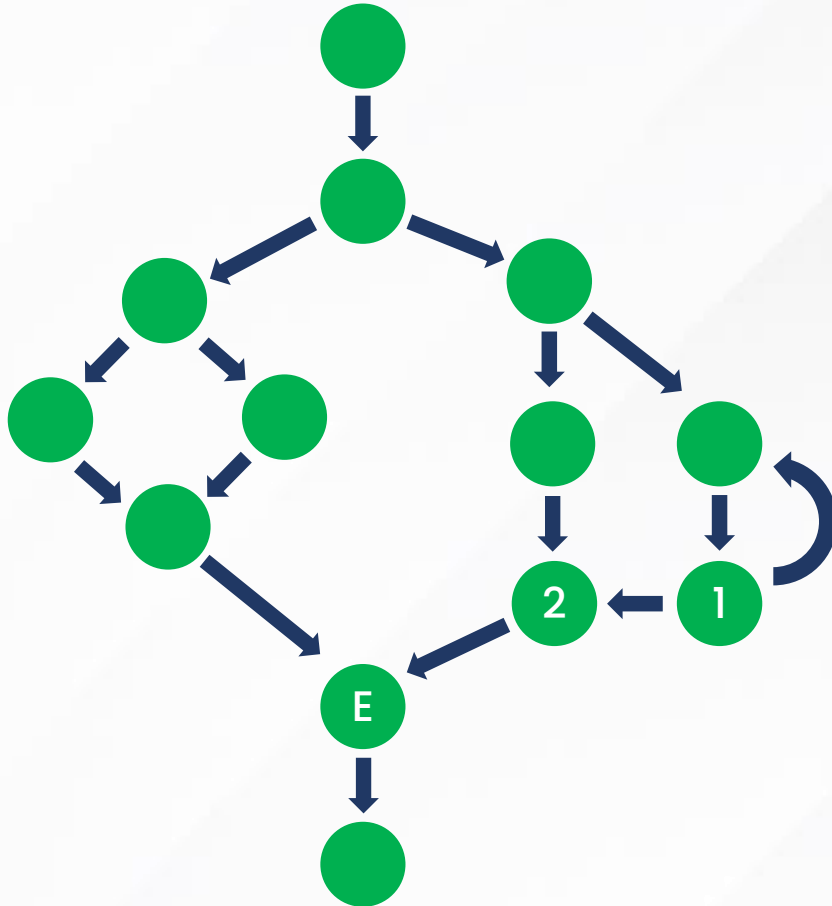
- ১ম **if** স্টেটমেন্ট দেখায় দুইটি দিক।
- ডান পাশের ১ম স্টেটমেন্টটি আরো দুইটি ভাগে বিভক্ত। দ্বিতীয় **if** স্টেটমেন্ট ব্যবহার করে।
- এই প্রোগ্রামের সকল স্টেটমেন্ট এ পৌঁছানো যাবে ডান পাশের **route** ব্যবহার করে।



1>A>B>C>1>C>2>E>Z

Statement Coverage :

Example 02 :

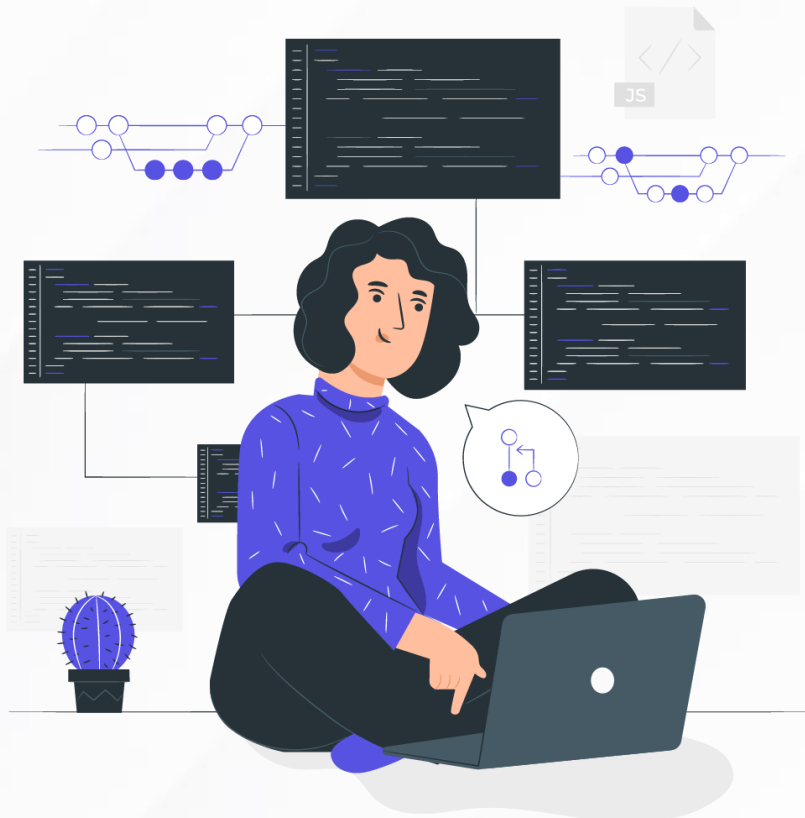


- এই গ্রাফটি আগেরটির তুলনায় কঠিন
- এই প্রোগ্রামে একটি if স্টেটমেন্টের মধ্যে আরেকটি if স্টেটমেন্ট ও লুপ আছে।
- চারটি ভিন্ন route আছে এই প্রোগ্রাম অংশে।
- প্রথম ব্রাঞ্চ টা ২ টা দিক দেয় যাওয়ার জন্য।
- দুইটি ব্রাঞ্চেই if স্টেটমেন্ট ও আরেকটি if স্টেটমেন্ট আরো দুইটি ভিন্ন দিক দিচ্ছে যাওয়ার জন্য।

Benefits and Drawbacks of this method

1. ডেড কোড যেগুলো কখনো এক্সিকিউট হয়না তা বের করা যায়।
2. প্রোগ্রামে ডেড কোড থাকলে ১০০% কভারেজ পাওয়া যায় না।
3. মিসিং ইন্সট্রাকশন অর্থাৎ যেসব কোড প্রয়োজন স্পেসিফিকেশন পূরন করতে তা পাওয়া যায় না।
4. শুধুমাত্র এক্সিকিউট করা স্টেটমেন্ট এর আলোকেই টেস্ট করা হয়। সব কোড কি পাওয়া যায় বা এক্সিকিউট করা হয়ে থাকে?
5. হোয়াট বক্স টেকনিক ব্যবহার করে মিসিং ইন্সট্রাকশন শনাক্ত করা যায় না।

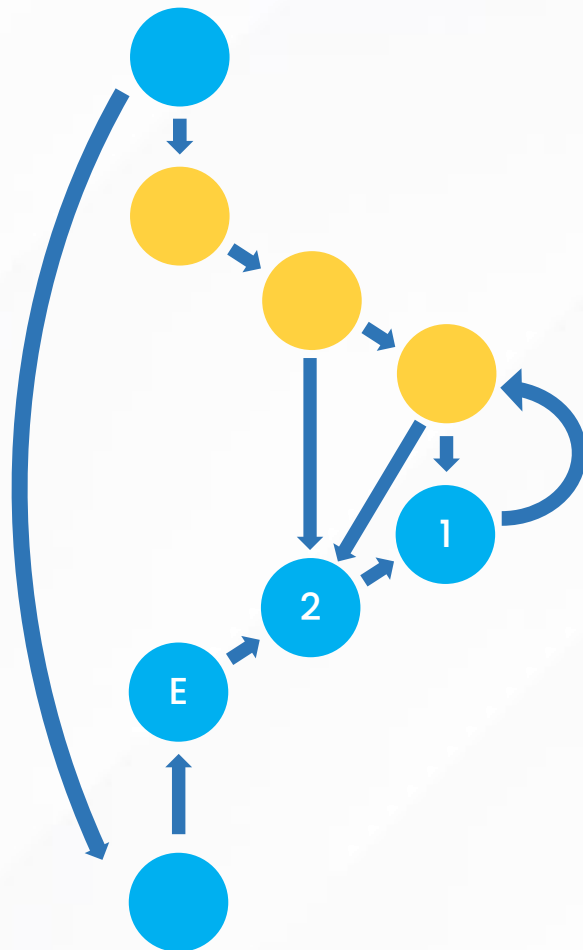
$$\text{Statement coverage} = \frac{\text{number of executed statement}}{\text{Total number of statement}} \times 100$$



Decision Coverage

Example 01 :

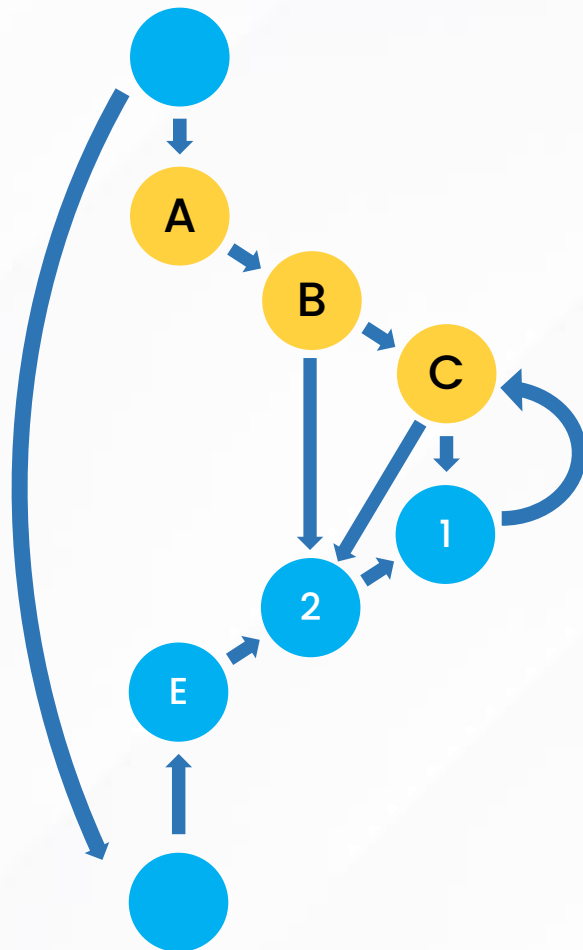
```
If (i>0) {  
  If (j>10) {  
    While (k>10) {  
      Do 1st Task ...  
    }  
    Do 2nd Task  
  }  
Do End Task
```



Decision Coverage

Example 01 :

```
If (i>0) {  
  If (j>10) {  
    While (k>10) {  
      Do 1st Task ...  
    }  
    Do 2nd Task  
  }  
Do End Task
```



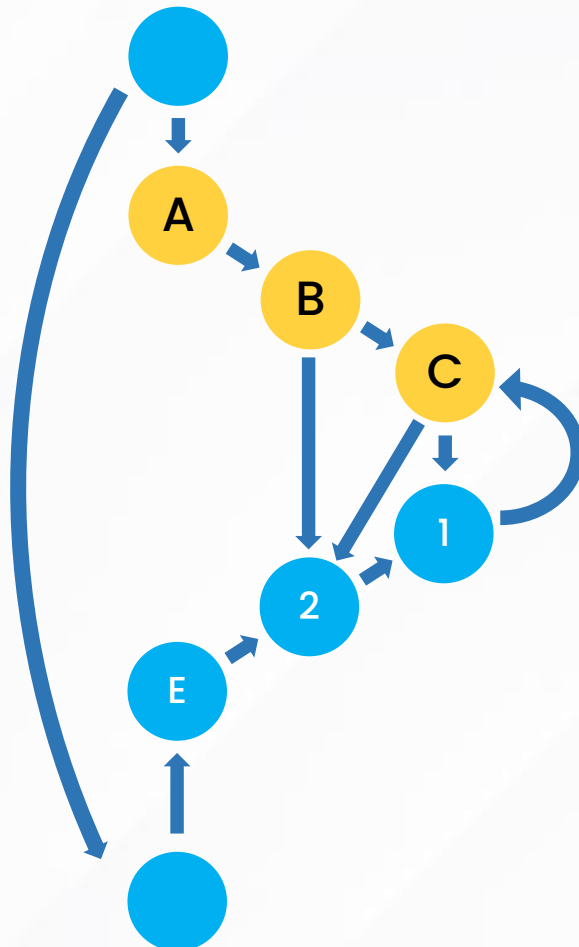
Decision Coverage

Example 01 :

প্রোগ্রামটি কন্ট্রোল ফ্লো দ্বারা প্রকাশিত হয়েছে বলে ধরা যাক।

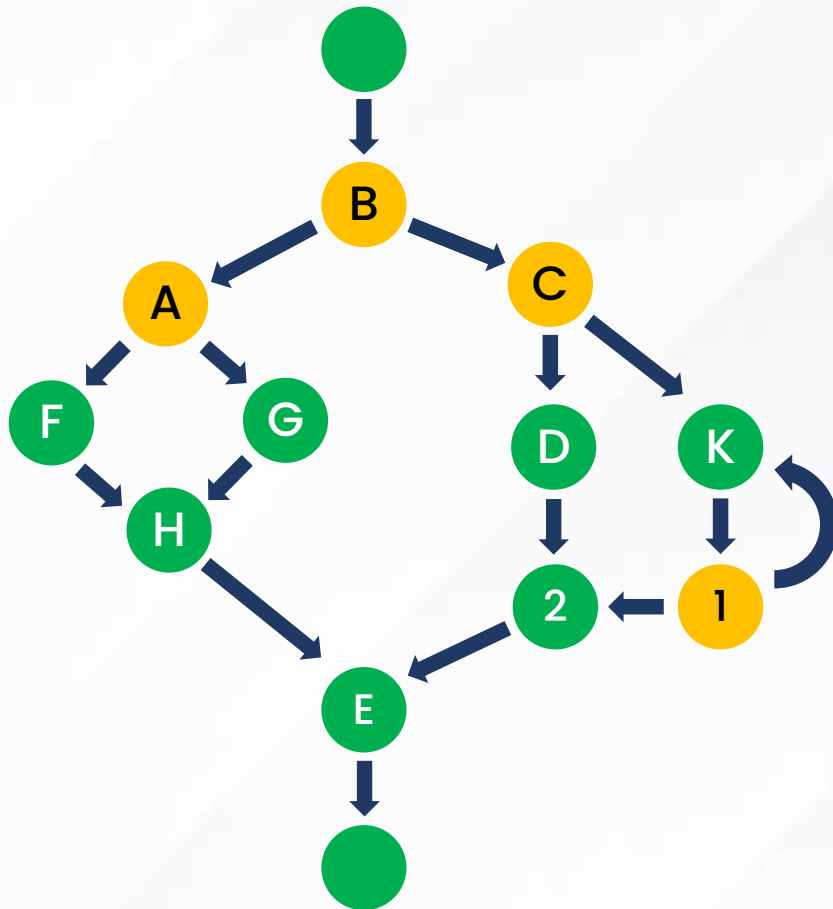
এখানে তিনটি ভিন্ন **route** আছে -

- প্রথম if স্টেটমেন্ট টি দুইটি দিক দেয়।
- প্রথম if স্টেটমেন্ট আবারো দুইটি দিক দেয় যার একটিতে লুপ আছে।
- সকল প্রান্তে পৌঁছানো সম্ভব এই তিনটি পথের কম্বিনেশন দ্বারা।



Decision Coverage

Example 02 :



- এই গ্রাফটি আগেরটির তুলনায় কঠিন
- চারটি ভিন্ন **route** আছে এই প্রোগ্রাম অংশে।
- প্রথম **if** স্টেটমেন্ট টি দুইটি দিক দেয়।
- **if** স্টেটমেন্টের দুটি ব্রাঞ্চেই আবারো দুইটি দিক দেয়।
- এই উদাহরনে লুপটিকে আলাদা ডিসিশন হিসেবে গণনা করা হয় নি।
- এখানে একই টেস্ট কেসের সেটের জন্য ১০০% স্টেটমেন্ট কভারেজ প্রয়োজন।

Condition Coverage

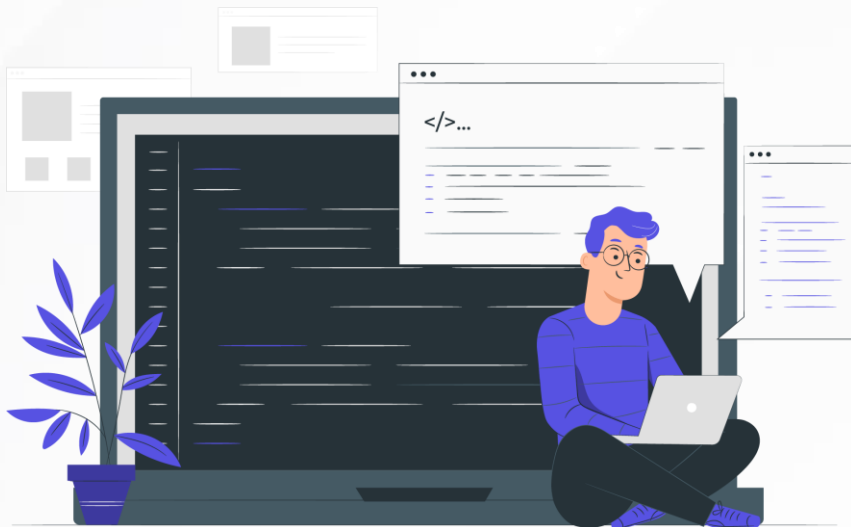
অনেকগুলো এটমিক কন্ডিশনে তৈরী একটি কন্ডিশনের জটিলতাকে গ্রাহ্য করা হয়ে থাকে। একটি এটমিক কন্ডিশনকে আরো ক্ষুদ্র কন্ডিশন স্টেটমেন্ট এ ভাগ করা যায়।

এই পদ্ধতিটি মাল্টিপল কন্ডিশনের প্রয়োগ করার সময়ে ঘটা ডিফেক্টগুলোকে খুঁজে বের করে। অনেকগুলো এটমিক কন্ডিশনের মাধ্যমে মাল্টিপল কন্ডিশন তৈরী হয় যা লজিকাল অপারেটর যেমন **OR, AND, XOR** ইত্যাদি ব্যবহারে হয়ে থাকে।

এটমিক কন্ডিশনে লজিকাল অপারেটর থাকেনা শুধু রিলেশনাল অপারেটর ও **NOT** অপারেটর থাকে। (**=, >, < etc.**)

তিন ধরনের কন্ডিশন কভারেজ হয়

- Simple Condition Coverage
- Multiple Condition Coverage
- Minimal multiple Condition Coverage



Simple Condition Coverage

কম্পাইন্ড কন্ডিশনের এটমিক সাব কন্ডিশনগুলোর অন্তত একটি লজিকাল ভ্যালু (True/False) থাকা লাগবে।

মাল্টিপল কন্ডিশন এক্সপ্রেশন ব্যবহার করে কন্ডিশন কভারেজের এই উদাহরণটি ব্যাখ্যা করা হচ্ছে।

মাত্র দুইটি টেস্টের মাধ্যমেই একটা সিম্পল কন্ডিশন কভারেজ পাওয়া যেতে পারে

প্রতিটি সাব কন্ডিশনেই দুইটি মান **True** এবং **False** নেয়া হয়েছে।

যদিও কম্পাইন্ড ফলাফলে দুইটি কেসেই **True** এসেছে -

- True OR False = True
- False OR True = True

Consider the following condition

$a > 2$ or $b < 6$

Test cases for simple condition coverage could be for example

$a = 3$ (true)	$b = 7$ (true)	$a > 2$ OR $b < 6$ (true)
$a = 1$ (false)	$b = 5$ (false)	$a > 2$ OR $b < 6$ (true)

Multiple Condition Coverage

এটমিক কন্ডিশনের পারমুটেশনে মাধ্যমে তৈরী হওয়া কন্ডিশনগুলো টেস্ট কেসের অংশ হবে। এটি একটি কন্ডিশন কভারেজের উদাহরন যেটায় মাল্টিপল কন্ডিশন এক্সপ্রেশন ব্যবহৃত হয়েছে।

চারটি টেস্ট কেসের মাধ্যমে মাল্টিপল কন্ডিশন কভারেজ পাওয়া যেতে পারে।

True, False এর সকল সম্ভাব্য কন্ডিশন বানানো হয়েছে।

মাল্টিপল কন্ডিশনের সকল ফলাফল পাওয়া গেছে।

টেস্ট কেসের সংখ্যা এক্সপোনেনশিয়ালভাবে বৃদ্ধি পাচ্ছে।

n = number of atomic conditions

$2n$ = number of test cases

Consider the following condition

$a > 2$ or $b < 6$

Test cases for simple condition coverage could be for example

$a = 3$ (true)	$b = 7$ (false)	$a > 2$ OR $b < 6$ (true)
$a = 3$ (true)	$b = 5$ (true)	$a > 2$ OR $b < 6$ (true)
$a = 1$ (false)	$b = 5$ (true)	$a > 2$ OR $b < 6$ (true)
$a = 1$ (false)	$b = 7$ (false)	$a > 2$ OR $b < 6$ (false)

Minimal Multiple Condition Coverage

সাব-কন্ডিশনের লজিক্যাল ফলাফল ব্যবহার করে যেসব কন্ডিশন পাওয়া যায় সবগুলোই টেস্ট এর অংশ হবে কেবল যদি সাব কন্ডিশনের ফলাফলের কারণে কন্ডিশনের ফলাফলের মধ্যে কোন পরিবর্তন আসে।

মাল্টিপল কন্ডিশন এক্সপ্রেশন ব্যবহার করে কন্ডিশন কভারেজের এই উদাহরণটি ব্যাখ্যা করা হচ্ছে।

চারটির মধ্যে তিনটি টেস্ট কেসেই সাব কন্ডিশনের কারণে পুরো ফলাফলটি পরিবর্তিত হয়।

শুধুমাত্র দুই নম্বার কেসেই (**true OR true=true**) সাব কন্ডিশনের কারণে পুরো ফলাফল পরিবর্তিত হয় না। এই টেস্ট কেসটি বাতিল করা যেতে পারে।

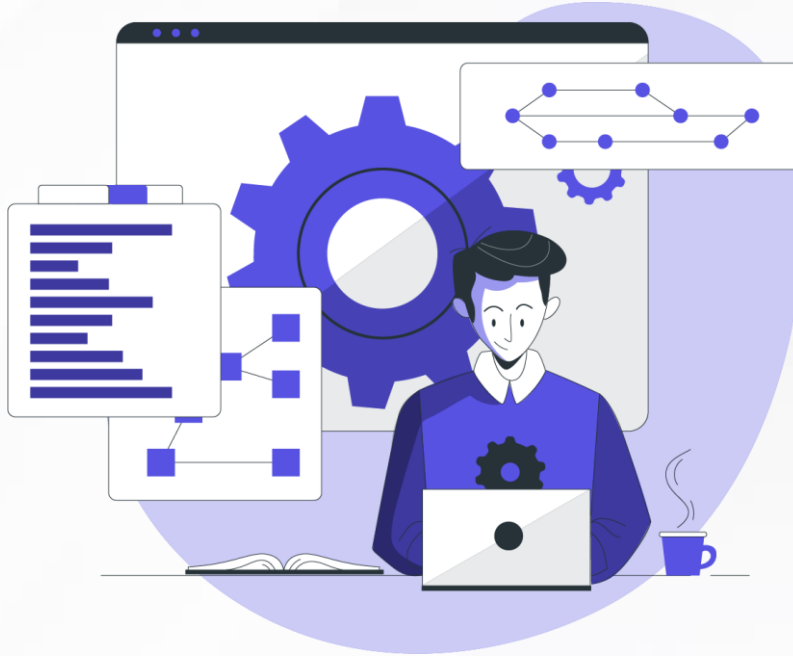
Consider the following condition

$a > 2$ or $b < 6$

Test cases for simple condition coverage could be for example

$a = 3$ (true)	$b = 7$ (false)	$a > 2$ OR $b < 6$ (true)
$a = 3$ (true)	$b = 5$ (true)	$a > 2$ OR $b < 6$ (true)
$a = 1$ (false)	$b = 5$ (true)	$a > 2$ OR $b < 6$ (true)
$a = 1$ (false)	$b = 7$ (false)	$a > 2$ OR $b < 6$ (false)

Condition Coverage – General Conclusion



সিম্পল কন্ডিশন কভারেজ মাল্টিপল কন্ডিশন পরীক্ষার জন্য উপযোগী না।

মাল্টিপল কন্ডিশন কভারেজ অধিক ভালো একটি পদ্ধতি -

- এটি স্টেটমেন্ট ও ডিসিশন কভারেজ কে নিশ্চিত করে
- এটির কারণে অনেক সংখ্যক টেস্ট কেস তৈরী হয়।

মিনিমাম মাল্টিপল কন্ডিশন কভারেজ আরো ভালো কারন -

- এটি টেস্ট কেসের সংখ্যা কমায়।
- স্টেটমেন্ট ও ডিসিশন কভারেজ ও ভালোভাবে করে।
- ডিসিশন স্টেটমেন্টের জটিলতা কে গ্রাহ্য করে।

সকল জটিল ডিসিশন পরীক্ষা করতে হবে আর এজন্য মিনিমাম মাল্টিপল কন্ডিশন কভারেজ সবচেয়ে ভালো পদ্ধতি।

Path Coverage

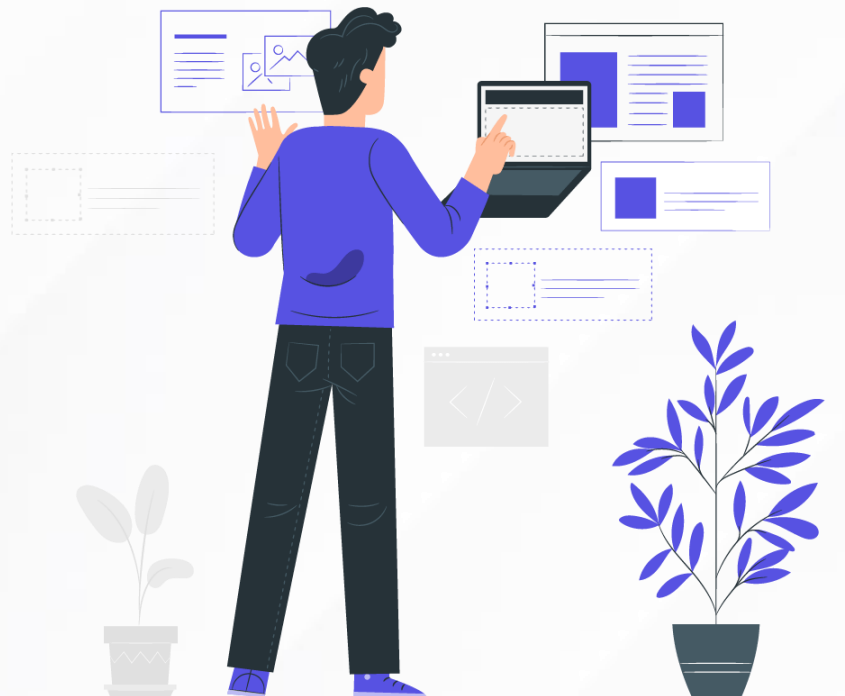
সকল সম্ভাব্য পথের এক্সিকিউশন হলো পাথ কভারেজের ফোকাস।

পাথ হলো প্রোগ্রাম অংশের কন্ট্রোল ফ্লো গ্রাফে **nodes** ও **edge** এর একের পর এক ক্রমই হলো পাথ)

ডিসিশন কভারেজের জন্য একটি লুপের ভিতর দিয়ে একটি একক পাথই যথেষ্ট। পাথ কভারেজের আরো যেসব টেস্ট কেস আছে তা হলো -

- একটি টেস্ট কেস হলো লুপের মধ্যে না যাওয়া
- প্রতিটি এক্সিকিউট করা লুপের জন্য একটি টেস্ট কেস

এটি খুব সহজেই অনেক সংখ্যক টেস্ট কেস বানাতে পারে।

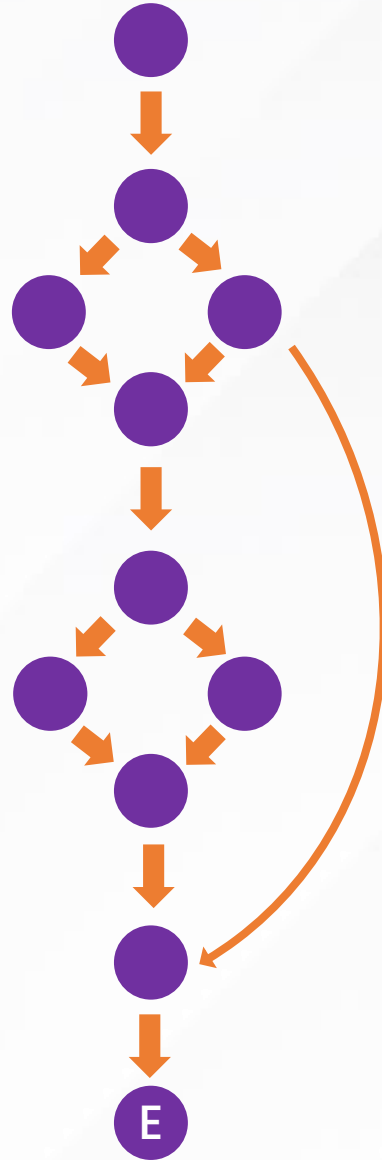


Path Coverage

Example 01 :

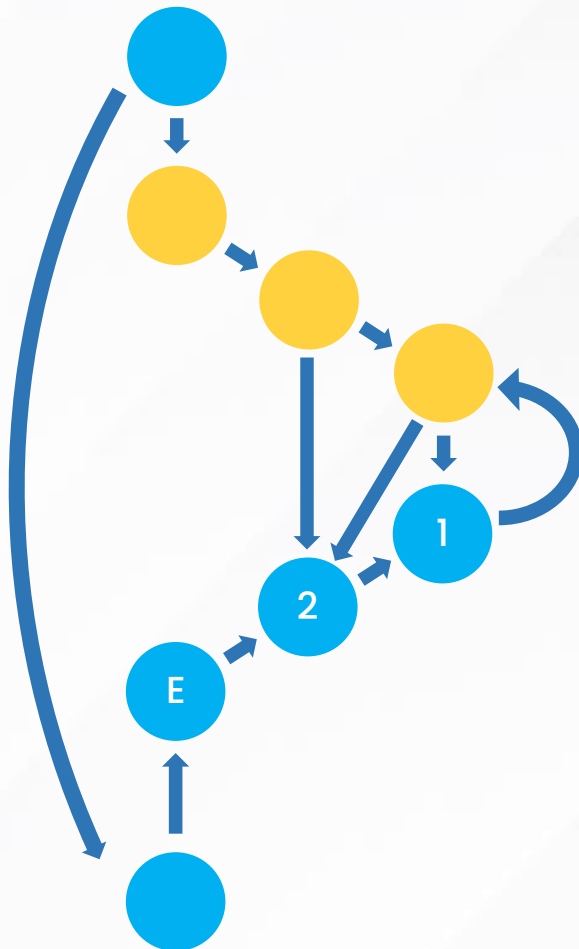
তিনটি ভিন্ন পথ এই প্রোগ্রাম
অংশের পুরো ডিসিশন কভারেজ
পাওয়ায় কাজ করছে।

যদিও পাঁচটি ভিন্ন সম্ভাব্য পথ
এখানে হয়ত এক্সিকিউট করা
হয়েছে।



Path Coverage

Example 02 :



- প্রোগ্রামটির এই অংশটি কন্ট্রোল ফ্লো দ্বারা প্রকাশিত হয়েছে বলে ধরা যাক। এখানে দুইটি if স্টেটমেন্ট ও একটি লুপ আছে দ্বিতীয় if স্টেটমেন্টের ভিতর।
- তিনটি ভিন্ন পথ এখানে পুরো ডিসিশন কভারেজ পেতে কাজ করছে।
- লুপটি দুইবার এক্সিকিউট করলে চারটি পথ পাওয়া সম্ভব।
- লুপ কাউন্টার প্রতিবার বৃদ্ধিতে একটি করে টেস্ট কেস বাড়াবে।

Path Coverage – General Conclusion

- খুবই সহজ প্রোগ্রামের জন্যই ১০০% পাথ কভারেজ পাওয়া যেতে পারে
- একটিমাত্র লুপ ও টেস্ট কেস এক্সপ্লোরেশন ঘটাতে পারে যেহেতু একটি লুপের এক্সিকিউশনের সাথে টেস্ট কেস ও বাড়ে প্রতিবারে।
- অসীমসংখ্যক পাথ হওয়া থিওরিটিক্যালি অসম্ভব।
- প্রোগ্রামের প্রতিটি সম্ভাব্য পাথ ই এক্সিকিউট করা হয়ে থাকে।
- ১০০% পাথ কভারেজের মধ্যে থাকে ১০০% ডিসিশন কভারেজ, আর ১০০% ডিসিশন কভারেজের মধ্যে থাকে ১০০% স্টেটমেন্ট কভারেজ।