14주차 예비보고서

전공: 아트&테크놀로지학과 학년: 4학년 학번: 20191048 이름: 김도솔

**1.**

**FSM(Finite-State Machine)**은 추상적인 모델링 도구로, 시스템이나 프로세스의 동작을 나타내는 데 사용된다. 이는 유한한 상태와 전이만을 다루기 때문에 유한 상태 기계라고 불린다. 간단한 형태의 FSM은 상태 다이어그램으로 나타낼 수 있으며, 복잡한 시스템의 모델링에도 적용할 수 있다. 주로 이벤트 기반 시스템, 자동 제어 시스템, 프로토콜 디자인 등 다양한 분야에서 사용된다. FSM은 여러 상태(state)들 간의 전이(transition)와 각 상태에서 수행되는 동작(action)으로 정의된다.

* 상태 (State): 시스템이나 프로세스가 가질 수 있는 여러 상태들을 나타낸다. 각 상태는 시스템이나 프로세스의 특정 상황을 나타낸다.

초기 상태 (Initial State): FSM의 시작 상태를 나타낸다. 시스템이나 프로세스가 처음 시작될 때 어떤 상태에 있는지를 나타낸다.

종료 상태 (Final State): 시스템이나 프로세스가 작업을 완료하거나 특정 조건이 충족되었을 때 도달하는 상태를 나타낸다.

* 입력 (Input): 외부에서 시스템에 주어지는 입력을 나타낸다. 이 입력은 상태 전이를 일으킬 수 있다.
* 전이 (Transition): 상태 간의 전이는 특정 조건이 충족되었을 때 일어나는 상태의 변경을 나타낸다. 예를 들어, 특정 입력이 들어오면 시스템은 현재 상태에서 다른 상태로 전이된다.
* 동작 (Action): 각 상태에서 수행되는 동작을 나타낸다. 이는 특정 상태로의 전이 시에 수행되는 작업이며, 이를 통해 시스템의 동작이 결정된다.

FSM의 한 **예시**로 자판기의 동작 시스템을 다음과 같이 나타낼 수 있다.

* 상태(State):

상태 1: 대기 상태 (Idle)

상태 2: 동전 투입 대기 상태 (Waiting for Coin)

상태 3: 음료 선택 대기 상태 (Waiting for Selection)

상태 4: 음료 출고 상태 (Dispensing)

* 상태 전이(Transition):

Transition 1: 대기 상태에서 동전이 투입되면 동전 투입 대기 상태로 이동

Transition 2: 동전 투입 대기 상태에서 음료를 선택하면 음료 선택 대기 상태로 이동

Transition 3: 음료 선택 대기 상태에서 구매를 완료하면 음료 출고 상태로 이동

Transition 4: 음료 출고 상태에서 구매가 완료되면 대기 상태로 이동

* 초기 상태(Initial State):

대기 상태 (Idle)

* 종료 상태(Final State):

없음 (자판기는 계속해서 동작 가능)

이러한 FSM 모델을 통해 자판기의 작동 흐름을 시각적으로 표현할 수 있다. 각 상태에서 특정 이벤트(동전 투입, 음료 선택 등)가 발생하면 해당 상태로 전이되고, 최종적으로 구매가 완료되면 다시 초기 상태로 돌아가는 것을 확인할 수 있다.

**2.**

**Mealy machine**은 FSM의 한 유형으로, 상태 전이와 출력이 입력 신호에 따라 결정되는 모델이다. Mealy machine은 Moore machine과는 다르게, 상태의 출력이 입력 신호와 현재 상태에 따라 결정된다. Mealy machine는 주로 상태의 수를 줄이는 데 사용되며, 무어 모델과는 달리 중간 상태가 없다. 엘리베이터 문을 예로 들면, 이 밀리 모델에서는 문이 열리거나 닫히는 중의 상태가 없다. 이는 현재 상태와 입력값 두 가지에 모두 영향을 받는 입력 행위를 갖게 된다. 예를 들면, "만약 문을 열어라"라는 입력값이 들어오면 현재 상태와 함께 모터를 작동하여 문을 열게 되고, 반대로 "만약 문을 닫아라"라는 입력값이 들어오면 현재 상태와 함께 모터를 반대 방향으로 작동시켜 문을 닫게 된다.

밀리 모델은 수학적으로 (S, S0, Σ, Λ, T, G)으로 구성된 6-튜플로 정의된다.

* S: 상태의 유한 집합
* S0: 초기 상태, S의 원소
* Σ: 입력값의 유한 집합
* Λ: 출력값의 유한 집합
* T: 상태 전이 함수, 현재 상태와 입력값을 기반으로 다음 상태값을 반환하는 함수 (S × Σ → S)
* G: 출력 함수, 현재 상태와 입력값을 기반으로 출력값을 반환하는 함수 (S × Σ → Λ)

Mealy machine은 간단한 상태 다이어그램으로 나타낼 수 있다. 각 상태는 원으로 표시되고, 화살표는 상태 전이를 나타내며 각 화살표에는 해당 전이를 일으키는 입력과 출력이 표시된다.

|  |  |
| --- | --- |
| State table | State diagram |
| 테이블이(가) 표시된 사진  자동 생성된 설명 |  |

**3.**

**Moore machine**은 FSM의 다른 유형으로, 상태의 출력이 입력에만 의존하는 모델이다. Mealy machine과는 달리, Moore machine에서는 출력은 현재 상태에만 종속되며 입력과 현재 상태에 따라 다음 상태 값이 결정된다. 무어 기계의 장점은 행동을 단순화할 수 있다는 것이다. 다시 엘리베이터 문을 예시로 들면, 유한 상태 기계는 “문을 열어라”와 “문을 닫아라”라는 상태를 변경하는 두 명령어를 인식할 수 있다. "열리는 중"의 상태에 있는 진입 동작은 모터를 돌려 문을 열기 시작하는 것이고, "닫히는 중"의 상태에 있는 진입 동작은 모터를 반대로 돌려 문을 닫는 것이다. “열림”과 “닫힘”의 상태는 완전히 열리거나 닫힌 상태에서 모터를 정지시킨다.

무어 모델은 수학적으로 (S, S0, Σ, Λ, T, G)으로 구성된 6-튜플로 정의된다.

* S: 상태의 유한 집합
* S0: 초기 상태, S의 원소
* Σ: 입력 값의 유한 집합
* Λ: 출력 값의 유한 집합
* T: 상태 전이 함수, 현재 상태와 입력 값을 기반으로 다음 상태값을 반환하는 함수 (S × Σ → S)
* G: 출력 함수, 현재 상태를 기반으로 출력 값을 반환하는 함수 (S → Λ)

Moore machine도 간단한 상태 다이어그램으로 나타낼 수 있다. 각 상태는 원으로 표시되고, 화살표는 상태 전이를 나타내며 각 화살표에는 해당 전이를 일으키는 입력이 표시된다. 출력은 상태에 대한 레이블로 나타낼 수 있다.

|  |  |
| --- | --- |
| State table | State diagram |
| 테이블이(가) 표시된 사진  자동 생성된 설명 |  |

**4.**

**Sequence Detector**는 입력된 디지털 시퀀스를 감지하고 인식하는 디지털 논리 회로를 말한다. 특히, 시퀀스 디텍터는 특정한 비트 또는 신호 패턴의 시퀀스를 찾아내는 데 사용된다. 이러한 시퀀스는 주어진 시스템에서 원하는 동작을 트리거하거나 특정 조건을 만족시킬 때 사용된다. 시퀀스 디텍터는 주로 유한 상태 기계(Finite State Machine, FSM)의 형태로 구현되며, 입력 신호의 시퀀스에 따라 다양한 상태로 전이하고 출력을 생성한다. 이를 통해 특정 시퀀스의 감지 및 처리가 가능하다.

예를 들어, 3비트의 입력 시퀀스 "101"을 감지하는 3-비트 이진 시퀀스 디텍터를 생각해보자. 이 시퀀스 디텍터는 다음과 같은 동작을 수행할 수 있다:

* 입력이 "1"일 때 상태를 전이하고 기다린다.
* 두 번째 비트가 "0"이면 상태를 전이하고 기다린다.
* 세 번째 비트가 "1"이면 시퀀스를 감지하고 특정 동작을 트리거한다.

이러한 시퀀스 디텍터는 다양한 응용 분야에서 사용된다. 예를 들어, 통신 시스템에서는 특정 패턴의 비트 시퀀스를 감지하여 데이터 프레임의 시작이나 종료를 식별할 수 있다.

**5.**

밀리 머신(Mealy Machine)과 무어 머신(Moore Machine) 이외에도 다양한 유형의 유한 상태 기계(Finite-State Machine, FSM)이 존재한다.

* 메플(Meaple) 모델: 메플 모델은 밀리 머신과 무어 머신의 특징을 혼합한 모델로, 상태 전이마다 출력이 변하고 상태에 따라도 고유한 출력을 가질 수 있다.
* PCM (Parameterized Concurrent Mealy) 모델: PCM 모델은 병렬적으로 동작하는 밀리 머신으로, 다양한 상태와 이벤트 조합에 대한 출력을 생성하는 데 특화되어 있다.
* UML 상태 다이어그램: Unified Modeling Language(UML)에서 정의된 상태 다이어그램은 상태, 이벤트, 전이, 동작 등을 시각적으로 나타내어 유한 상태 기계를 모델링하는 데 사용된다.
* 혼합 상태 머신: 혼합 상태 머신은 계층적인 상태 구조를 지원하여 복잡한 시스템을 모델링하는데 유용하게 쓰인다. UML에서는 이러한 개념을 Statechart 다이어그램을 사용하여 나타낼 수 있다.
* 투스테이트(Statecharts): 투스테이트(Statecharts)는 David Harel이 제안한 확장된 상태 다이어그램으로, 계층적인 상태와 동시에 여러 상태의 동작을 표현할 수 있는 기능을 제공한다.