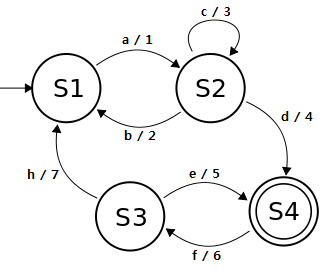
14주차 예비보고서

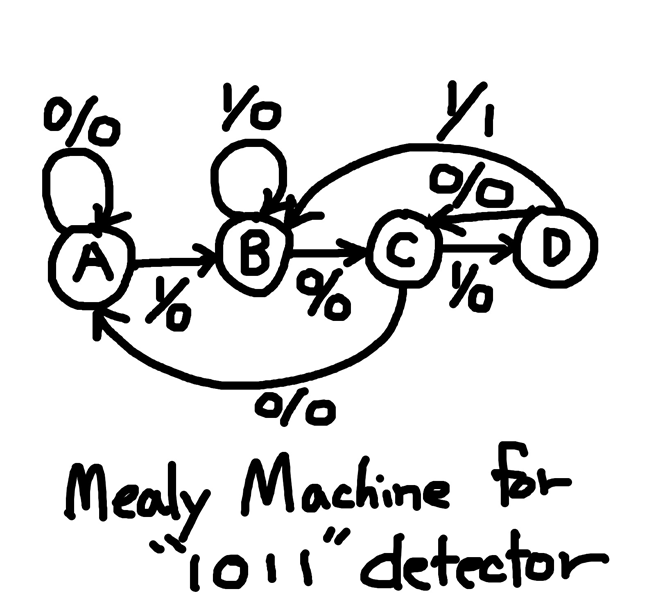
전공: 컴퓨터공학과 학년: 3학년 학번: 20191612 이름: 윤기웅

1.

FSM이란 Finite State Machine의 줄임말로 컴퓨터 혹은 논리 회로를 만드는데 사용되는 수학적 모델이다. 또한 상태기계라고 불리기도 하고 이 기계는 한 가지의 상태를 갖고 이는 현재 상태라고 불린다. 현재의 상태에서 다른 상태로 이동하려면 사건에 의해서 변할 수 있고 이 경우를 전이라고 한다. AI 혹은 게임 개발에도 많이 사용되면서 개발자가 아닌 다른 3자가 쉽게 작동을 이해할 수 있고 직관적이다.

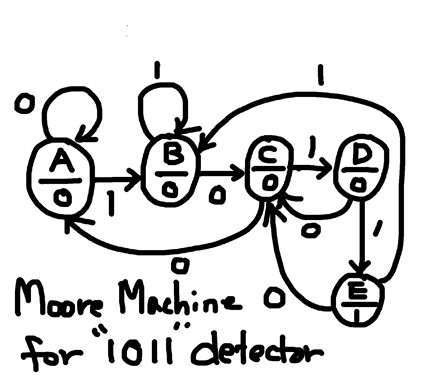


2. Mealy machine은 아웃풋(다음의 상태)이 현재의 상태와 입력에 의해 변하는 모델이다. 밀리머신은 무어 머신보다 상태의 개수가 항상 작거나 같다. 아래의 1011을 감지하는 밀리 머신을 예시로 보게 되면 처음 상태인 A에서 1이 감지되지 아니하면 계속 A에 머무르게 된다. 그림에서 알 수 있듯이 각 상태는 0과 1의 감지에 대해서 모두 경우를 나타내고 있다. 첫 번째 숫자는 감지하는 숫자를 의미하고 뒤의 숫자는 결과값을 의미하면서 0또는 1을 갖는데 0은 감지하고자 하는 수를 찾지 못 했다는 뜻이고 1이면 감지하고자 하는 수를 발견했다는 뜻이다. 이어서 B에 대해서도 보면, 두 번째 숫자인 0을 감지하면 다음 상태인 C로 넘어가게 되고 만약 1을 감지하면 그 상태에 계속 머물러 있다. A에서 이미 1을 한 번 발견했기 때문에 B에서 다시 1을 발견하면 뒤에 발견한 1부터 다시 1011을 찾아보면 되기 때문이다. C에서는 1을 발견하면 최종 상태인 D로 넘어가게 되고 만약 0을 발견하면 다시 처음의 상태인 A로 돌아간다. 왜냐하면 이전에 한 번도 실패 없이 C까지 왔어도 0을 발견하면 100이 되니까 가장 앞의 1을 다시 찾아야 하기 때문이다. D에서는 1이 나오면 1011을 찾게 되니 1을 반환하고 상태는 B로 돌아간다. 왜냐하면 가장 마지막의 1이 앞으로 다시 나올 1011의 첫 번째 자리의 1이 될 수 있기 때문이다.



3. Moore machine은 밀리 머신과 달리 출력이 현재의 상태에 의해서만 결정되는 모델이다. 각각의

상태에서 실패에 대한 입력에 대해서 어느 상태로 이동할 지 명시해줘야 한다. 무어 머신은 밀리머신보다 상태의 개수가 항상 크거나 같다. 아래의 무어 머신을 보게 되면 1011을 찾아내는 머신인데 첫 번째 상태인 A에서 0을 찾으면 다음상태로 넘어가지 못 하고 계속 머무르게 된다. B에서도 0을 찾지 못하면 즉, 1만 계속 들어온다면 다음 상태로 넘어가지 못하고 막히게 된다. C에서는 1을 찾으면 다음 상태로 넘어가지만 그러지 못 하면 다시 A로 돌아간다. 0이 나왔기 때문에 처음부터 다시 1을 찾아야 하기 때문이다. D에서는 1이 나오면 끝나는 상태로 가지만 만약 0이 나오게 되면 다시 C로 돌아가서 찾으려는 뒷 부분인 11을 다시 찾는다.



4.

sequence detector란 앞서 살펴본 밀리 머신이나 무어 머신처럼 입력에서 특정한 규칙을 찾아내는 회로이다. 비트의 문자열을(0 또는 1) 받아들이고 원래 찾으려고 한 규칙이 나오게 되면 1을 반환하는 머신이 일반적이다. 이를 설계하려면 회로에 대한 상태 다이어그램과 테이블을 만들고 상태의 수를 결정한다. 그리고 예상 시퀀스에 대해서 상태 전이를 만든다.

5.

게임에서 FSM을 자주 사용하는데 주로 적이나 몬스터의 이동을 구현하려는 경우에 많이 사용된다. 게임에서 인공지능적인 부분을 구현하려고 사용되는데 예를 들어 사냥해야 하는 몬스터의 감정을 상태로 표시하고 그 상태간의 이동에 대해서 몬스터 이동을 구현하는 방식이다. 몬스터가 안정(rest)된 경우 이동이 불규칙적이며 공격을 안한다. 그리고 화가 난 경우(angry), 이동속도가 빨라진다. 그리고 마지막 RAGE(분노)의 상태에서는 주변 플레이어를 공격하거나 주변에 플레이어가 없으면 플레이어를 향해 이동한다. 이 상태 전이 과정을 도표로 그리면 아래와 같다.

