

MSc in Software Engineering and Database Technologies

CT621 Artificial Intelligence

**Student Name:** Cristina Borges

**NUIG ID Number:** 20234955

**Course:** CT621 Artificial Intelligence

**Workshop No:** Week 4

**Assignments:** Week 4

**Date of Submission:** 13/06/2021

Contents

1.0 Assignment 2

1.1 Question 1 3

Predicates 3

Events 3

1.2 Question 2 5

Logical description of initial situation 5

Logical query to give paths 5

Go Instruction 6

Appendix 1 7

# Assignment

**Question 1.**

Use Event Calculus to state the relation that holds between every pair of the following real-world events:

BT: The birth of Alan Turing (1912)

DT: Death of Alan Turing (1954)

WS: Start of World War II

WE: End of World War II

EM: Turing helps break enigma codes

**Question 2.**

In this assignment, you will consider the problem of planning a route for a robot to take from one city to another. The basic action taken by the robot is *Go (x,y),* which takes it from city x to city y if there is a direct route between those cities. *DirectRoute(x,y)* is true if an only if there is a direct route from x to y; you can assume that all such facts are already in the knowledge base. The robot begins in New York and must reach Los Angeles.

1. Write a suitable logical description of the initial situation of the Robot.
2. Write a suitable logical query whose solutions will provide possible paths to the goal.
3. Write a sentence describing the *Go* action. (Hint: Use a successor-state axiom

## 1.1 Question 1

## Predicates

The following are a list of predicates functions and their meaning that we have looked at for the purpose of this problem

|  |  |
| --- | --- |
| Predicate/function | Meaning |
| Holds(p) | p holds |
| Start (p, e) | e starts p |
| End (p, e) | e ends p |
| Initiates (e, f) | e initiates f |
| Initiates (e, f, t) | If e occurs at t, f is true after |
| Terminate (e, f) | e terminates f |
| Terminate (e, f, t) | If e occurs at time t, then F is false after t |
| e1<e2 | e1 precedes e2 |
| Broken(e1, f, e2) | f is broken between e1 and e2 |
| Incompatible (f1, f2) | f1 and f2 incompatible |
| After (e, f) | Time period after e in which f holds |
| Before (e, f) | Time period before e in which f holds |
| T(f, e) | f is true at time e |
| Happens (e, t) | Event e happens over period t |
| Clipped (e, t) | Ceases to be true during period t |
| Restored (e, t) | Becomes true during period t |
| Initially(f) | f is true at timepoint 0 |
| HoldsAt (e, t) | e is true at time t |
| Happens(e, t) | e occurs at t |
| StoppedIn (t1, f, t2) | f is stopped between t1 and t2 |

## Events

I have used the predicates above to express each of the events referred to above

**Event BT**: Birth of Alan Turing (1912)

Happens (BT, 1912)

Uses the predicate Happens to state the event BT occurs in 1912

**Event DT:** Death of Alan Turing (1954)

This is a discreet event so should use discrete event calculus

Terminates(DT, death,1954)

Uses the predicate Terminates to state the event death occurs in 1954

**Event WS:** Start of WWII

Initiates (WWII, at war, 1939)

Uses the predicate Initiates to state the event WWII and the function “at war” commenced in 1939

**Event WE:** End of WWII

Terminate (WWII, at war, 1945)

Uses the predicate Terminate to state that the event WWI and the function “at war” ceased in 1945

**Event EM:** Break Enigma – during WWII

StoppedIn(1939, EM, 1945)

Uses the predicate StoppedIn to state that the event EM ceased between 1939 and 1945

**I then have the following pairs of events:**

|  |  |
| --- | --- |
| Pair | Predicate |
| BT DT | ∃ BTDT, t (Happens(BT, 1912) ∧ 1912 < t < 1954 ∧ Terminates(DT, death,1954)) |
| BT WS | ∃ BTWS, t (Happens(BT, 1912) ∧ 1912 < t < 1939 ∧ Happens(WS, 1939)) |
| BT WE | ∃ BTWE, t (Happens(BT, 1912) ∧ 1912 < t < 1945 ∧ ¬Clipped(at war, 1939, 1945) |
| BT EM | ∃ BTEM, t (Happens(BT, 1912) ∧ 1939 < t < 1945 ∧ ¬StoppedIn(1939, EM, 1945) |
| WS DT | ∃ WSDT, t (Happens(WS, 1939) ∧ Initiates(WWII, at war, 1939) ∧ 1939 < t < 1954 ∧ Terminates(DT, death,1954)) |
| WE DT | ∃ WEDT, t (Happens(WE, 1945) ∧ 1945 < t < 1954 ∧ Terminates(DT, death,1954)) |
| EM DT | ∃ EMDT, t (StoppedIn(1939, EM, 1945) ∧ 1939 < t < 1954 ∧ Terminates(DT, death,1954)) |
| WS WE | ∃ WWII, 1939 Happens(WWII, 1939) ∧ Initiates(WWII, at war, 1939) ∧ (1939 < t) ¬Clipped(at war, 1939, 1945) |
| WS EM | ∃ WSWE, 1939 Happens(WWII, 1939) ∧ Initiates(WWII, at war, 1939) ∧ (1939 < t) ¬StoppedIn(1939, EM, 1945) |
| EM WE | ∃ EMDT, t (StoppedIn(1939, EM, 1945) ∧ 1939 < t < 1954 ∧ ¬Clipped(at war, 1939, 1945)) |

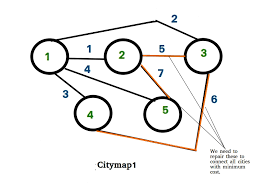
## 1.2 Question 2

## Logical description of initial situation

Firstly, I am trying to express the situation of the robot in words.

Robot starts in location X (New York), Objective is to find a direct route to location Y (Los Angeles).

The following diagram might illustrate the different nodes (cities) and paths to be explored. If we say that Node 1 is New York and Node 3 is Los Angeles, we can investigate each path and see the paths that provide a route from our starting point to the proposed destination.



A logical representation of moving from location New York (NY) to Los Angeles (LA) could be articulated as:

initiates(go\_to\_node(NodeNY, NodeLA), in(NodeLA), t)  
holdsat(in(NodeLA), T)

The instruction is saying that if the robot is at NodeNY initially it will move the robot to NodeLA and it is true that it remains there at time T.

## Logical query to give paths

First the starting location of the robot needs to be defined – starting position is Node1

Next is to set that robot will follow path for the goal destination

1. This query should define the starting point at Node1 and denotes the location of the robot.
2. Robot then moves to the next node by moving forward – in reality the robot would need to move in all possible directions (forward/backward etc – have assumed forward only for simplicity).
3. Programme terminates after program reaches Node Y

initiates(follow\_path, loc (Node1, ahead),T)

hold\_at(loc(Node1,behind),T),

next\_node(Node1,forward,T).

terminate(follow\_path, loc(NodeY, behind,T).

## Go Instruction

happens(go\_to\_node(Node i, Node j),T i,T j)

towards( Node k, Node j, Node i), connects (pathX, Node i, Node k)

holds\_at(next\_node(NodeX), Tx)

happens(go\_to\_node(Node x, Node j),T j,T x)

Restored(Ty,in(NodeY), Tj)

Go instruction tells robot to

1. The move to Node j from Node i occurs over the period Ti to Tj.
2. Move to Node k given the move from Node i to Node j. Node K can be reached from Node i by pathX
3. Robot remains at Node X at time Tx
4. Repeat step 4 to go from Node X to Node j
5. During the period to Tj, Ty robot arrives at NodeY

## Appendix 1

**Assignment 1**

**References**

Kowalski R, Sergot M (1986) A Logic-based Calculus of Events

Available at: <https://www.doc.ic.ac.uk/~rak/papers/event%20calculus.pdf>

Accessed 12th June 2021

Shanahan M (undated) The Event Calculus Explained

Available at: <https://www.doc.ic.ac.uk/~mpsha/ECExplained.pdf>

Accessed 12th June 2021

CT621 Artificial Intelligence (2021) Knowledge Representation Part 2

Available at: NUIG Blackboard

Accessed 12th June 2021

Mueller E T. (2008) Handbook of Knowledge Representation

Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.406.6366&rep=rep1&type=pdf>

Accessed 12th June 2021

Shanahan M, Witkowski M () Robot Navigation and Map Building with the Event Calculus

Available at: <http://www.ee.ic.ac.uk/mark/papers/ijcai99.pdf>

Accessed 12th June 2021

Geeksforgeeks (2021) Minimum cost to connect all cities

Available at: <https://www.geeksforgeeks.org/minimum-cost-connect-cities/>

Accessed 12th July 2021