Assignment 4

Question #1

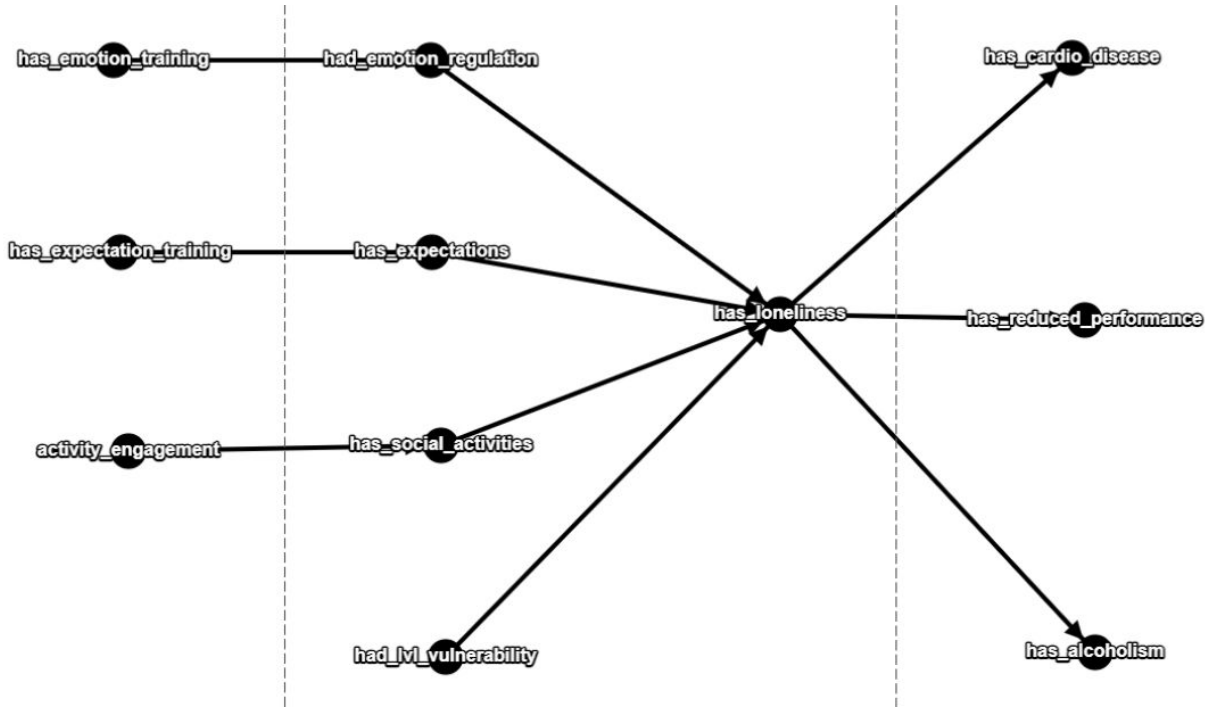
Formalisation of the concept	*Informal explanation*	*Influenced by*
has_loneliness(A:AGENT; X:SCALE)	‘agent A has X feeling of loneliness’	vulnerability regulation expectation
had_lvl_vulnerability(A:AGENT; X:SCALE)	‘agent A has a X level of vulnerability’	genes
had_emotion_regulation(A:AGENT; X:REAL)	‘agent A has X emotion regulation skills’	emotion training
has_expectations(A:AGENT; X:SCALE)	‘agent A has X expectations about others’	expectation training
has_social_activities(A:AGENT; X:SCALE)	‘agent A has a X amount of social activities’	engaging in more activities
has_alcoholism(A:AGENT; X:BOOLEAN)	‘agent A has alcoholism, which is X’	feeling of loneliness
has_cardio_disease(A:AGENT; X:BOOLEAN)	‘agent A has cardiovascular diseases, which is X’	feeling of loneliness
has_reduced_performance (A:AGENT; X:BOOLEAN)	‘agent A has reduced performance at work, which is X ’	feeling of loneliness
activity_engagement(A:AGENT; X:BOOLEAN)	‘agent A has activity engagement, which is X’	/
has_emotion_training(A:AGENT; X:BOOLEAN)	'agent A has emotion training, which is X'	/
has_expectation_training(A:AGENT; X:BOOLEAN)	'agent A has expectation training, which is X'	/

Figure 1: Table of Concepts Used in Loneliness Model

Sorts
SCALE = {low; normal; high; very high}
AGENT = {lieve}

Figure 2: Table of Sorts Used in Loneliness Model

Question #2



Question 3

AT EACH TIME STEP t
Agent A has X level of vulnerability

AT EACH TIME STEP t
IF Agent A had emotion training is TRUE AND emotion regulation is BAD
    THEN emotion regulation increases to POOR
ELSEIF Agent A had emotion training is TRUE AND emotion regulation is POOR
    THEN emotion regulation increases to FAIR
ELSEIF Agent A had emotion training is TRUE AND emotion regulation is FAIR
    THEN emotion regulation increases to GOOD
ELSEIF Agent A had emotion training is TRUE AND emotion regulation is GOOD
    THEN emotion regulation increases to GREAT

AT EACH TIME STEP t
IF Agent A had expectation training is TRUE AND expectations is LOW
    THEN expectations increases to NORMAL
ELSEIF Agent A had expectation training is TRUE AND expectations is NORMAL
    THEN expectations increases to HIGH
ELSEIF Agent A had expectation training is TRUE AND expectations is HIGH

THEN expectations increases to VERY HIGH

AT EACH TIME STEP t
Agent A has X amount of social activities

AT EACH TIME STEP t
IF Agent A has emotion training is TRUE
THEN emotion regulation increases by 1
ELSEIF Agent A has emotion training is FALSE
THEN emotion regulation increases by 0

AT EACH TIME STEP t
IF Agent A has lvl vulnerability AND has emotion regulation AND has expectations AND has social activities
THEN Agent A has X amount of loneliness

AT EACH TIME STEP t
IF Agent A has loneliness is TRUE
THEN Agent A has alcoholism OR has cardiac disease OR has reduced performance

Question #4

rules.m

```
function [ fncs ] = rules()
  % DO NOT EDIT
  fncs = l2.getRules();
  for i=1:length(fncs)
  fncs{i} = str2func(fncs{i});
  end
end
function result = ddr3( trace, params, t)
  result = {};
  for had_emotion_regulation = trace(t).had_emotion_regulation
    agent = had_emotion_regulation.arg{1};
    emotion_regulation = had_emotion_regulation.arg{2};
    value = 0;
    for has_emotion_training = trace(t).has_emotion_training
      agent = has_emotion_training.arg{1};
      train = has_emotion_training.arg{2};
```

```
        if train == true
            value = value + 1;
        elseif train == false
            value = value + 0;
        end
      end
   emotion_regulation = emotion_regulation + value;
   result = {result{:} {t+1, 'had_emotion_regulation', {agent, emotion_regulation}} };
   end
end
```

sorts.l2

```
SCALE; {low, normal, high, very high}
AGENT; {lieve}
```

scenarios.l2

```
default(
had_emotion_regulation{lieve, 0.2}; [1]
had_emotion_regulation{lieve, 0.8}; [1]
has_emotion_training{lieve, true};    [1:25]
has_emotion_training{lieve, false};    [26:50]
)
```

predicates.l2

```
has_loneliness; {AGENT, SCALE}
had_lvl_vulnerability;  {AGENT, SCALE}
had_emotion_regulation; {AGENT, REAL}
has_expectations;   {AGENT, SCALE}
has_social_activities;  {AGENT, SCALE}
has_alcoholism; {AGENT, BOOLEAN}
has_cardio_disease; {AGENT, BOOLEAN}
has_reduced_performance;    {AGENT, BOOLEAN}
has_emotion_training;   {AGENT, BOOLEAN}
has_expectation_training;   {AGENT, BOOLEAN}
```

parameters.l2

```
default(
step_size; 0.1
)
```