Problem Set 3

COSC 290 Spring 2018

Caio Brighenti

March 2, 2018

1 Problem 1: Translating natural language into logic

1.1 DLN 3.145

$\forall t \in T : [\forall j_1, j_2 \in J : [At(j_1, t) \wedge At(j_2, t) \implies j_1 = j_2]]$

1.2 DLN 3.146

$\forall j \in J : [\exists t \in T : scheduledAt(j, t)]$

1.3 DLN 3.147

$\forall t \in T : [At(A, t) \implies \neg(At(A, t+1) \vee At(A, t+2))]$

1.4 DLN 3.148

$\exists t_1, t_2, t_3 \in T : [At(B, t_1) \wedge At(B, t_2) \wedge At(B, t_3) \wedge (t_1 \neq t_2 \neq t_3)]$

1.5 DLN 3.149

$\forall t_1, t_2, t_3 \in T : [(At(B, t_1) \wedge At(B, t_2) \wedge (t_1 \neq t_2 \neq t_3)) \implies \neg At(B, t_3)]$

1.6 DLN 3.150

$\exists t_1, t_2 \in T : [At(E, t_1) \wedge (\forall t_3 \in T : [At(E, t_3) \implies t_3 <= t_1]) \wedge At(D, t_2) \wedge t_2 > t_1]$

1.7 DLN 3.151

$\exists t_1, t_2, t_3 \in T : [At(G, t_1) \wedge At(F, t_2) \wedge At(G, t_3) \wedge (t_1 < t_3) \wedge (\forall t_4 \in T : [At(G, t_4) \implies ((t_1 > t_4) \vee (t3 < t_4))]) \wedge (t_3 > t_2 > t_1)]$

1.8 DLN 3.152

$\exists t_1, t_2, t_3 \in T : [At(I, t_1) \wedge At(H, t_2) \wedge At(I, t_3) \wedge (t_1 < t_3) \wedge (\forall t_4 \in T : [At(I, t_4) \implies ((t_1 >= t_4) \vee (t3 =< t_4))]) \wedge (t_3 > t_2 > t_1) \wedge \forall t_5 \in T : [(At(H, t_5) \wedge t_5 \neq t_2) \implies \neg(t_3 > t_5 > t_1)]]$

2 Problem 2: Translating code snippets into logic

2.1 DLN 3.191

y_n represents the last value in S that the interior loop considers, where n is the amount of items in S.

$\exists x \in S : [P(x)] \vee P(y_n)$

Since the inner loop always sets the flag to false initially, and the flag is only checked upon exiting the inner loop, the check will only return true when the last check the inner loop makes results in the flag being set to true.

2.2 DLN 3.192

$\exists x, y \in S : \neg P(x, y)$

2.3 DLN 3.193

y_1 represents the first element in S the inner loop evaluates $P(x, y)$ for.

$\exists x \in S : [P(x, y_1)]$

As the flag is initially set to true in the outer loop, if the flag is ever set to false in the inner loop, it will remain false until the next execution of the outer loop. Since the flag check is inside the inner loop, then the check will only return true if the first evaluation of $P(x, y)$ does not set the flag to false.

2.4 DLN 3.194

y_1 and x_1 represent the first elements $P(x, y)$ is evaluated for.

$\neg P(x_1, y_1)$

As the flag check is in the inner loop, and returns false if the flag is false and true otherwise, a value will be returned in the first iteration of the inner loop. This value will be false if $P(x, y)$ is evaluated to true, as the flag will remain false, and will return true if $P(x, y)$ evaluates to false, setting the flag to true.

2.5 DLN 3.195

$\forall x, y \in S : \neg P(x, y)$

2.6 DLN 3.196

$\exists x, y \in S : P(x, y)$

3 Problem 3: Is it a theorem?

The proposition $\exists x \in S : (Q(x) \implies (\forall y \in S : Q(y)))$ is a theorem as it is always true. The proposition is true if for any element x in S, $Q(x) \implies (\forall y \in S : Q(y))$ evaluates to true. The $\implies$ operator always results true when the left hand side is false. Therefore, if $Q(x)$ ever evaluates to false, the overall proposition evaluates to true. This means that if set S contains any element x where $Q(x)$ evaluates to false, the proposition will be true.

The other possible scenario is then that S does not contain any elements x where $Q(x)$ evaluates to false. This is synonymous to saying $Q(x)$ is true for all elements x in S. This is precisely what the right hand side of the implication, $\forall y \in S : Q(y)$, states. The $\implies$ operator always results in true when the right hand side is true, thus making the overall proposition true in the scenario where $Q(x)$ is true for all elements of S.

Succinctly, if S contains any elements x such that $Q(x)$ is false, the overall proposition will be true as the implication will be true for that element. Otherwise, if $Q(x)$ is true for all elements x of S, implication will also be true, rendering the overall proposition true. These are the only two ways in which S can exist, and as the proposition evaluates to true in both cases, the proposition is a theorem.