## Student Information

Full Name : Beyazıt Yalçınkaya
Id Number : 2172138

# 1 Classical Semantics

Let's first translate each statement into propositional logic expressions by assigning meanings to some atomic propositions, and then construct a truth table to check the semantics.

- $p :=$ The statements on the left sign is correct.
- $q :=$ The statements on the middle sign is correct.
- $r :=$ The statements on the right sign is correct.

Then, each sign can be expressed in terms of $p, q, r$ as follows.

- Left button: $(q \supset r) \wedge (r \supset q)$
- Middle button: $\neg p \wedge ((\neg q \wedge r) \vee (q \wedge \neg r))$
- Right button: $\neg q$

The truth table for the above statements is given below.

| $p$ | $q$ | $r$ | $(q \supset r) \wedge (r \supset q)$ | $\neg p \wedge ((\neg q \wedge r) \vee (q \wedge \neg r))$ | $\neg q$ |
|---|---|---|---|---|---|
| T | T | T | T | F | F |
| T | T | F | F | F | F |
| T | F | T | F | F | T |
| T | F | F | T | F | T |
| F | T | T | T | F | F |
| **F** | **T** | **F** | **F** | **T** | **F** |
| F | F | T | F | T | T |
| F | F | F | T | F | T |

In the above table, the truth value of $p$ and $(q \supset r) \wedge (r \supset q)$ must be the same, the truth value of $q$ and $\neg p \wedge ((q \wedge \neg r) \vee (\neg q \wedge r))$ must be the same, and the truth value of $r$ and $\neg q$ must be the same since each of these pairs are basically expressing the same sign. The row with the bold font is the only row in the truth table satisfying the equivalence of these pairs of expressions. By observing this row, we concluded that the proposition $q$ and $\neg p \wedge ((\neg q \wedge r) \vee (q \wedge \neg r))$ are true whereas others are false. Hence, the sign of the middle button is truthful to open the door, i.e., I would press the middle button to go through the door.

# 2 Modeling in Predicate Logic

## (a) Family Relations

- Everybody has a mother.

$$\forall x \exists y (M(y, x))$$

- Everybody has a father and a mother.

$$\forall x \exists y \exists z (M(y, x) \wedge F(z, x))$$

- Whoever has a mother has a father.

$$\forall x ((\exists y M(y, x)) \supset (\exists z F(z, x)))$$

- Ali is a grandfather.

$$\exists x \exists y (F(Ali, x) \wedge F(x, y))$$

- All fathers are parents.

$$\forall x \exists y (F(x, y) \supset F(x, y))$$

- All husbands are spouses.

$$\forall x \exists y (H(x, y) \supset H(x, y))$$

- No uncle is an aunt.

$$\forall x \exists y \exists z ((B(x, y) \wedge (M(y, z) \vee F(y, z))) \supset (\neg S(x, y)))$$

- All brothers are siblings.

$$\forall x \exists y (B(x, y) \supset (B(y, x) \vee S(y, x)))$$

- Nobody's grandmother is anybody's father.

$$\forall x ((\neg(\exists y \exists z M(x, y) \wedge M(y, z))) \supset (\exists w F(x, w)))$$

- Ali and Veli are husband and wife.

$$H(Ali, Veli) \vee H(Veli, Ali)$$

- Ahmet is Fatma's brother-in-law.

$$\exists x (H(x, Fatma) \wedge B(Ahmet, x))$$

## (b) A Security Protocol

- An attacker can persuade a server that a successful login has occurred, even if it hasn't.
  $Attacker(x)$: $x$ is an attacker.
  $SuccessfulLogin(x)$: $x$ is a server that a successful login has occurred.
  $LoggedIn(x, y)$: $x$ logged into server $y$.
  $CanPersuade(x, y)$: $x$ can persuade server $y$.

$$\exists x \exists y (Attacker(x) \land SuccessfulLogin(y) \land \neg LoggedIn(x, y) \land CanPersuade(x, y))$$

- An attacker can overwrite someone else's credentials on the server.
  $Attacker(x)$: $x$ is an attacker.
  $Credential(x)$: $x$ is a credential.
  $CanOverwrite(x, y)$: $x$ can overwrite $y$ on the server.
  $CredentialOf(x, y)$: $y$ is credential of $x$.

$$\exists x \exists y (Attacker(x) \land Credential(y) \land \neg CredentialOf(x, y) \land CanOverwrite(x, y))$$

- All users enter passwords instead of names.
  $User(x)$: $x$ is a user.
  $EntersPassword(x)$: $x$ enters password.
  $EntersName(x)$: $x$ enters name.

$$\forall x((User(x)) \supset (EntersPassword(x) \land \neg EntersName(x)))$$

- Credential transfer both to and from a device MUST be supported.
  $Device(x)$: $x$ is a device.
  $TransferToSupported(x)$: Credential transfer to $x$ is supported.
  $TransferFromSupported(x)$: Credential transfer from $x$ is supported.

$$\forall x((Device(x)) \supset (TransferToSupported(x) \land TransferFromSupported(x)))$$

- Credentials MUST NOT be forced by the protocol to be present in cleartext at any device other than the end user's.
  $Device(x)$: $x$ is a device.
  $EndUser(x)$: $x$ is a end user.
  $EndUsersDevice(x, y)$: $x$ is end user y's device.
  $Forced(x)$: Credentials of $x$ is forced to be present in cleartext.

$$\forall x \forall y (\neg (Device(x) \land EndUser(y) \land \neg EndUsersDevice(x, y) \supset Forced(y)))$$

- The protocol MUST support a range of cryptographic algorithms, including symmetric and asymmetric algorithms, hash algorithms, and MAC algorithms.
  $P(x)$: The protocol supports algorithm $x$.
  $C(x)$: $x$ is a cryptographic algorithm.
  $S(x)$: $x$ is a symmetric algorithm.
  $A(x)$: $x$ is a asymmetric algorithm.
  $H(x)$: $x$ is a hash algorithm.
  $M(x)$: $x$ is a MAC algorithm.

$$\exists x(C(x) \land P(x)) \land \exists x(S(x) \land P(x)) \land \exists x(A(x) \land P(x)) \land \exists x(H(x) \land P(x)) \land \exists x(M(x) \land P(x))$$

- Credentials MUST only be downloadable following user authentication or else only downloadable in a format that requires completion of user authentication for deciphering.
  $Credential(x)$: $x$ is a credential.
  $A(x)$: Credential $x$ is downloadable following user authentication.
  $B(x)$: Credential $x$ is downloadable in a format that requires completion of user authentication for deciphering.

$$\forall x((Credential(x)) \supset (A(x) \vee (\neg A(x) \wedge B(x))))$$

- Different end user devices MAY be used to download, upload, or manage the same set of credentials.
  $Device(x)$: $x$ is an end user device.
  $CredentialSet(x)$: $x$ is a set of credentials.
  $A(x, y)$: $x$ can be used to download, upload, or manage $y$.
  $x \neq y$: $x$ and $y$ are not the same.

$$\exists x \exists y \exists z(Device(x) \wedge Device(y) \wedge x \neq y \wedge CredentialSet(z) \wedge A(x, z) \wedge A(y, z))$$

# 3 Structural Induction

We will use structural induction for this proof. Let $m$ be the height of the parse tree of the expression $\psi \supset (\phi_1 \wedge (\phi_2 \wedge (... \wedge (\phi_{n-1} \wedge \phi_n)...)))$.

**Base Case.** For $m = 1$, we need to prove $\psi \supset (\phi_1 \wedge \phi_2) \vdash (\psi \supset \phi_1) \wedge (\psi \supset \phi_2)$. We use natural deduction for this proof, it is given below.

| | | |
|---|---|---|
| 1. | $\psi \supset (\phi_1 \wedge \phi_2)$ | *premise* |
| 2. | $\psi$ | *assumption* |
| 3. | $\phi_1 \wedge \phi_2$ | $\supset e$ 1, 2 |
| 4. | $\phi_1$ | $\wedge e_1$ 3 |
| 5. | $\psi \supset \phi_1$ | $\supset i$ 2-4 |
| 6. | $\psi$ | *assumption* |
| 7. | $\phi_1 \wedge \phi_2$ | $\supset e$ 1, 6 |
| 8. | $\phi_2$ | $\wedge e_2$ 7 |
| 9. | $\psi \supset \phi_2$ | $\supset i$ 6-8 |
| 10. | $(\psi \supset \phi_1) \wedge (\psi \supset \phi_2)$ | $\wedge i$ 5, 9 |

This concludes the base case.

**Inductive Step.** For $m > 1$, we need to prove $\psi \supset (\phi_1 \wedge (\phi_2 \wedge (... \wedge (\phi_{n-1} \wedge \phi_n)...))) \vdash (\psi \supset \phi_1) \wedge ((\psi \supset \phi_2) \wedge (... \wedge (\psi \supset \phi_n)...))$ where $n$ is the proper number for a parse tree with height $m$. Say $\phi' = (\phi_2 \wedge (... \wedge (\phi_{n-1} \wedge \phi_n)...))$, then we have $\psi \supset (\phi_1 \wedge \phi') \vdash (\psi \supset \phi_1) \wedge (\psi \wedge \phi')$ which trivially follows from the base case. Inductive application of this argument completes the proof.

# 4 Normal Forms

Horn formulas are basically formulas in CNF. Given a Horn formula it can be trivially converted to CNF by converting implications into disjunctions (i.e., $P \supset Q \equiv \neg P \vee Q$). However, not every formula in CNF can be directly converted into a Horn formula, e.g., $P \wedge Q$. A subset of formulas in CNF that can be converted to Horn formulas are be identified with the following BNF grammar.

$$\begin{aligned} L &::= \bot \mid \top \mid p \\ A &::= \neg L \mid \neg L \vee A \\ D &::= A \vee L \\ C &::= D \mid D \wedge C \end{aligned} \tag{1}$$

Given formulas from this grammar, we can easily convert them to Horn formulas. Let's informally give the steps of the procedure.

1. Apply double negation to the negated $L$s coming from $A$ and move one negation in.
2. Use $P \supset Q \equiv \neg P \vee Q$ equivalence to convert left disjunctions to implications.

As an example, consider the following Horn formula $(p \wedge q \wedge s) \supset p$. Now, we give a derivation for this formula in our new grammar.

$$\begin{aligned} C &\to D \\ &\to A \vee L \\ &\to (\neg L \vee A) \vee L \\ &\to (\neg L \vee \neg L \vee A) \vee L \\ &\to (\neg L \vee \neg L \vee \neg L) \vee L \\ &\to (\neg p \vee \neg L \vee \neg L) \vee L \\ &\to (\neg p \vee \neg q \vee \neg L) \vee L \\ &\to (\neg p \vee \neg q \vee \neg s) \vee L \\ &\to (\neg p \vee \neg q \vee \neg s) \vee p \end{aligned} \tag{2}$$

Now, we apply 1. and we get $\neg(p \wedge q \wedge s) \vee p$. Then, we apply 2. and we get: $(p \wedge q \wedge s) \supset p$. To convert a Horn formula into CNF, apply the same procedure in the reverse order, i.e., convert implications to disjunctions and distribute the negations obtained from this conversion.