**Derivation of Logic Proofs**

**Purpose**

This assignment is an exercise for using natural deduction to derive logic proofs. This exercise not only tests students’ ability to apply natural deduction but also familiarizes students with logic proofs that are derived in computers.

**Objectives**

Students will be able to:

* Apply natural deduction to derive logic proofs.

**Technology Requirements**

* N/A

**Assignment Description**

#### *You may add space as needed to provide your complete answers.*

#### **Part 1: Proving the Validity of Sequents**

Assuming binding priority (¬, ∧, ∨, →), how would you prove the validity of the sequents?

1. p → (q ∨ r), ¬q, ¬r ⊢ ¬p using MT rule

|  |  |  |
| --- | --- | --- |
| 1 | p → (q ∨ r) | premise |
| 2 | ¬q | premise |
| 3 | ¬r | premise |
| 4 | p | assumption |
| 5 | q |  |
| 6 | ¬q |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 | ¬p |  |

1. ⊢(p → q) ∨(q → r) without using LEM

|  |  |  |
| --- | --- | --- |
| 1 | (p → q) ∨(q → r) | assumption |
| 2 | p | assumption |
| 3 | q | assumption |
| 4 | r | assumption |
| 5 | p → q | i 2-3 |
| 6 | q → r | i 3-4 |
| 7 | (p → q) ∨(q → r) | i 5-6 |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |

1. (p → q) ∧ (q → r) ⊢ (p → r)

|  |  |  |
| --- | --- | --- |
| 1 | (p → q) ∧ (q → r) | premise |
| 2 | (p → q) | assumption |
| 3 | (q → r) | assumption |
| 4 | p | assumption |
| 5 | q | E 2, 4 |
| 6 | r | E 3, 5 |
| 10 | (p → r) | 4, 6 I |

1. (p → r) ∧ (q → ¬r) ⊢ (q → ¬p)

|  |  |  |
| --- | --- | --- |
| 1 | (p → r) ∧ (q → ¬r) | premise |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 | (q → ¬p) |  |

1. (p → r) ∨ (q → r) ⊢ (p ∧ q) → r

|  |  |  |
| --- | --- | --- |
| 1 | (p → r) ∨ (q → r) | premise |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 | (p ∧ q) → r |  |

1. ¬(p ∨ q) is equivalent to (¬p ∧ ¬q) (prove → in both directions)

|  |  |  |
| --- | --- | --- |
| 1 | ¬(p ∨ q) | premise |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 | (¬p ∧ ¬q) |  |

|  |  |  |
| --- | --- | --- |
| 1 | (¬p ∧ ¬q) | premise |
| 2 | ¬p |  |
| 3 | ¬q |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 | ¬(p ∨ q) |  |

1. (p → ¬q) is equivalent to ¬(p ∧ q)

|  |  |  |
| --- | --- | --- |
| 1 | (p → ¬q) | premise |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 | ¬(p ∧ q) |  |

#### **Part 2: Determining the Validity of Sequents, Proofs, and Truth Tables**

Assuming binding priority (¬, ∧, ∨, →), are these sequents valid or not? If they are valid, how do you prove it? If they are not valid, what would the truth table be?

1. A → B, C → D ⊢ A ∨ C → B ∧ D

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | A → B | C → D | A ∨ C | B ∧ D | A → B, C → D | A ∨ C → B ∧ D | A → B, C → D ⊢ A ∨ C → B ∧ D |
| T | T | T | T | T | T | T | T | T | T | T |
| T | T | T | F | T | F | T | F | F | F | F |
| T | T | F | T | T | T | T | T | T | T | T |
| T | T | F | F | T | T | T | F | T | F | F |
| T | F | T | T | F | T | T | F | F | F | F |
| T | F | T | F | F | F | T | F | F | F | F |
| T | F | F | T | F | T | T | F | F | F | F |
| T | F | F | F | F | T | T | F | F | F | F |
| F | T | T | T | T | T | T | T | T | T | T |
| F | T | T | F | T | F | T | F | F | F | F |
| F | T | F | T | T | T | F | T | T | T | T |
| F | T | F | F | T | T | F | F | T | T | T |
| F | F | T | T | T | T | T | F | T | F | F |
| F | F | T | F | T | F | T | F | F | F | F |
| F | F | F | T | T | T | F | F | T | T | T |
| F | F | F | F | T | T | F | F | T | T | T |

1. A ∧ ¬A ⊢ ¬(B → C) ∧ (B → C)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | ¬A | A ∧ ¬A | ¬(B → C) | (B → C) | ¬(B → C) ∧ (B → C) | A ∧ ¬A ⊢ ¬(B → C) ∧ (B → C) |
| T | T | T | F | F | F | T | F | F |
| T | T | F | F | F | F | T | F | F |
| T | F | T | F | F | T | F | F | F |
| T | F | F | F | F | T | F | F | F |
| F | T | T | T | F | F | T | F | F |
| F | T | F | T | F | F | T | F | F |
| F | F | T | T | F | F | T | F | F |
| F | F | F | T | F | F | T | F | F |

1. (A ∧ B) → C, C → D, B ∧ ¬D ⊢ ¬A

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | ¬A | ¬D | (A ∧ B) | C → D | B ∧ ¬D | (A ∧ B) → C | B ∧ ¬D ⊢ ¬A | (A ∧ B) → C, C → D, B ∧ ¬D ⊢ ¬A |
| T | T | T | T | F | F | T | T | F | T | F | F |
| T | T | T | F | F | T | T | F | T | T | F | F |
| T | T | F | T | F | F | T | T | F | F | F | F |
| T | T | F | F | F | T | T | T | T | F | F | F |
| T | F | T | T | F | F | F | T | F | T | F | F |
| T | F | T | F | F | T | F | F | F | T | F | F |
| T | F | F | T | F | F | F | T | F | T | F | F |
| T | F | F | F | F | T | F | T | F | T | F | F |
| F | T | T | T | T | F | F | T | F | T | F | F |
| F | T | T | F | T | T | F | F | T | T | T | F |
| F | T | F | T | T | F | F | T | F | T | F | F |
| F | T | F | F | T | T | F | T | T | T | T | T |
| F | F | T | T | T | F | F | T | F | T | F | F |
| F | F | T | F | T | T | F | F | F | T | F | F |
| F | F | F | T | T | F | F | T | F | T | F | F |
| F | F | F | F | T | T | F | T | F | T | F | F |

#### **Part 3: Drawing Parse Trees**

Assuming binding priority (¬, ∧, ∨, →), what are **all** the possible parse trees for each formula? Draw the trees for each formula.

1. (p → r) ∨ (q → r) → (p ∧ q) → r
2. (p → r) ∧ (q → ¬r) ∧ (q → ¬p)

**Submission Directions for Deliverables**

Drafting your answers in Microsoft Word is strongly encouraged. Office 365 is Microsoft’s productivity suite, and includes Word, Excel, PowerPoint, Access, OneNote and more. It is available for offline and online use. Currently enrolled students can use this software for free. Refer to the article ["ASU providing Microsoft Office 365 to all Students, Faculty & Staff"](https://uto.asu.edu/asu-providing-microsoft-office-365-all-students-faculty-staff) (you must be logged into [my.asu.edu](http://my.asu.edu/) to view the article) in the “Welcome and Start Here” section of the course.

Type your answers on the “**CSE 571\_U2\_ Assignment 1\_Derivation of Logic Proofs\_Submission Template**”. Submit the template as a single PDF titled “**Last Name\_First Name\_Derivation of Logic Proofs**”.

**Rubric**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **No Attempt** | **Undeveloped** | **Developing** | **Approaching** | **Meets** |
| **Part 1: Proving the Validity of Sequents** | Provides no proof. | Provides incorrect steps.  Fully incorrect answer with no merit in approach showing how proof rules should be applied to solve a problem. | Provides some steps that are correct.  Answer may demonstrate surface-level understanding of logic proof rules, but lacks the overall view to develop a proof toward the conclusion.  Application of proof rules has  clear imperfections. | Provides a mostly correct answer.  Answer may demonstrate mid-level understanding of proof rules, with a reasonable approach to developing a proof toward the conclusion. Some inconsistency with specificity may be present, such as providing incorrect rule names.  Application of proof rules may have slight imperfections. | Provides a clear, fully correct answer.  Answer demonstrates high-level understanding of proof rules, with a logical approach to developing the proof.  All proof names are provided correctly, and a correct proof is present.  Application of proof rules has no imperfections. |
| **Part 2: Determining the Validity of Sequents, Proofs, and Truth Tables** | **For valid sequents:**  Provides no proof.  **For invalid sequents:**  Provides no truth table. | **For valid sequents:**  Provides incorrect proof.  Fully incorrect answer with no merit in approach to show how proof rules should be applied to solve a problem.  **For invalid sequents:**  Fully incorrect answer with no merit in approach to show how truth tables should be developed. | **For valid sequents:**  Provides some steps that are correct.  Answer may demonstrate surface-level understanding of logic proof rules, but lacks the overall view to develop a proof toward the conclusion.  Application of proof rules has  clear imperfections.  **For invalid sequents:**  Answer demonstrates a general understanding of how to develop a truth table, but entries may be missing or inaccurate. | **For valid sequents:**  Provides a mostly correct answer.  Answer may demonstrate mid-level understanding of logic proof rules, with a reasonable approach to developing a proof toward the conclusion. Some inconsistency with specificity may be present, such as providing incorrect rule names.  Application of proof rules may have slight imperfections.  **For invalid sequents:**  Answer provides a fully developed truth table, though a few entries may be inaccurate. | **For valid sequents:**  Provides a clear, fully correct answer.  Answer demonstrates high-level understanding of proof rules, with a logical approach to developing the proof.  All proof names are provided correctly, and a correct proof is present.  Application of proof rules has no imperfections.  **For invalid sequents:**  Answer provides a fully developed and correct truth table. |
| **Part 3: Drawing Parse Trees** | Provides no parse trees. | Provides incorrect parse trees.  Fully incorrect answer with no merit in approach to show how parse trees should be developed for a given formula. | Provides some subtrees that are correct.  Answer may demonstrate surface-level understanding of parse trees, but lacks the overall view to develop a tree for a given formula.  Drawings of parse trees have clear imperfections. | Provides a mostly correct answer.  Answer may demonstrate mid-level understanding of the parse trees, with a reasonable approach to developing the correct tree for a given formula. Some nodes may be missing, some incorrect rule names may be provided.  Drawings of parse trees may have slight imperfections. | Provides a clear, fully correct answer.  A high-level of understanding of the parse tree is provided. All nodes in the tree are correctly given.  Drawings of parse trees have no imperfections. |