

COMPSCI/SFWRENG 2FA3
Discrete Mathematics with Applications II
Winter 2020

Assignment 4

Dr. William M. Farmer
McMaster University

Revised: February 6, 2020

Assignment 4 consists of two problems. You must write your solutions to the problems using LaTeX.

Please submit Assignment 4 as two files, `Assignment_4_YourMacID.tex` and `Assignment_4_YourMacID.pdf`, to the Assignment 4 folder on Avenue under Assessments/Assignments. *YourMacID* must be your personal MacID (written without capitalization). The `Assignment_4_YourMacID.tex` file is a copy of the LaTeX source file for this assignment (`Assignment_4.tex` found on Avenue under Contents/Assignments) with your solution entered after each problem. The `Assignment_4_YourMacID.pdf` is the PDF output produced by executing

`pdflatex Assignment_4_YourMacID`

This assignment is due **Sunday, February 16, 2019 before midnight**. You are allowed to submit the assignment multiple times, but only the last submission will be marked. **Late submissions and files that are not named exactly as specified above will not be accepted!** It is suggested that you submit your preliminary `Assignment_4_YourMacID.tex` and `Assignment_4_YourMacID.pdf` files well before the deadline so that your mark is not zero if, e.g., your computer fails at 11:50 PM on February 16.

Although you are allowed to receive help from the instructional staff and other students, your submission must be your own work. Copying will be treated as academic dishonesty! If any of the ideas used in your submission were obtained from other students or sources outside of the lectures and tutorials, you must acknowledge where or from whom these ideas were obtained.

Background

1. Let $(S, <)$ be a strict partial order. $(S, <)$ is *dense* if, for all $x, y \in S$ with $x < y$, there is some $z \in S$ such that $x < z < y$.

2. A *queue* is a finite sequence of elements for which elements are added (“enqueued”) to the back of the sequence and removed (“dequeued”) from the front of the sequence. An *empty queue* is a queue with no members. A *singleton queue* is a queue with a single element that is obtain by enqueueing an element to an empty queue.

Problems

1. [10 points] Construct in MSFOL a theory T of strict total orders that are dense and have minimum and maximum elements. Give two models for T .

Jatin Chowdhary — chowdhaj — Feb. 16th, 2020

Put your solution here.

BEGIN SOLUTION

The Strict Total Order $(S, <)$ is dense. So for all $x, y \in S$ with $x < y$, there is some $z \in S$ such that $x < z < y$.

Let \mathbb{U} be the universe set that contains all objects including itself. This is the notation I will be using to denote the universe set.

Let $\Sigma_T = (\mathcal{B}, \mathcal{C}, \mathcal{F}, \mathcal{P}, \tau)$ where:

- a. $\mathcal{B} = \{\mathbb{U}\}$
- b. $\mathcal{C} = \emptyset$
- c. $\mathcal{F} = \{\text{min}, \text{max}\}$
 $\tau(\text{min}) = \mathbb{U}$.
 $\tau(\text{max}) = \mathbb{U}$.
- d. $\mathcal{P} = \{<\}$
 $\tau(<) = \mathbb{U} \times \mathbb{U} \rightarrow \mathbb{B}$.

Let Σ_T be a strict total order and has the following properties:

The property Dense has already been defined above. The other properties are defined in Lecture 3: Predicate Logic, slide 9.

- a. Irreflexivity
- b. Transitivity
- c. Asymmetric
- d. Trichotomous
- e. Dense

Two examples of Σ_T are:

- $(\mathbb{N} : [0, 1000], <)$
- $(\mathbb{Q} : [69, 420], <)$

END SOLUTION

2. [10 points] Let $\Sigma_{\text{queue}} = (\mathcal{B}, \mathcal{C}, \mathcal{F}, \mathcal{P}, \tau)$ where:

- a. $\mathcal{B} = \{\text{Element}, \text{Queue}\}$.
- b. $\mathcal{C} = \{\text{error}, \text{empty}\}$.
- c. $\mathcal{F} = \{\text{front}, \text{back}, \text{enqueue}, \text{dequeue}\}$.
- d. $\mathcal{P} = \emptyset$.
- e. $\tau(\text{error}) = \text{Element}$.
- f. $\tau(\text{empty}) = \text{Queue}$.
- g. $\tau(\text{front}) = \text{Queue} \rightarrow \text{Element}$.
- h. $\tau(\text{back}) = \text{Queue} \rightarrow \text{Element}$.
- i. $\tau(\text{enqueue}) = \text{Element} \times \text{Queue} \rightarrow \text{Queue}$.
- j. $\tau(\text{dequeue}) = \text{Queue} \rightarrow \text{Queue}$.

Construct in MSFOL a theory $T = (\Sigma_{\text{queue}}, \Gamma_{\text{queue}})$ of queues. Γ_{queue} should contain axioms that say:

- a. The front of an empty queue is the error element.
- b. The front of a singleton queue is the single element in the queue.
- c. Let q be a queue obtain by enqueueing e to a nonempty queue q' . The front of q is the front of q' .
- d. The back of an empty queue is the error element.
- e. Let q be a queue obtain by enqueueing e to a queue q' . The back of q is e .
- f. The dequeue of an empty queue is the empty queue.
- g. The dequeue of a singleton queue is the empty queue.
- h. Let q be a queue obtain by enqueueing e to a nonempty queue q' . The dequeue of q is the enqueue of e to the dequeue of q' .

Jatin Chowdhary — chowdhaj — Feb. 16th, 2020

Put your solution here.

BEGIN SOLUTION

Σ_{queue} is defined above.

Γ_{queue} contains the following:

Let *singleton_queue* represent a singleton queue, that only contains one element. In other words, *singleton_queue* is:

$$\text{enqueue}(\text{Element}, \text{empty}) = \text{Queue}$$

- a. $\text{front}(\text{empty}) = \text{error}$
- b. $\text{front}(\text{enqueue}(\text{Element}, \text{empty})) = \text{Element}$

I can also write: $\text{front}(\text{singleton_queue}) = \text{Element}$

- c. $\forall e : \text{Element}, q' : \text{Queue} . \text{front}(q') = \text{front}(\text{enqueue}(e, q'))$
Can be written as: $\text{front}(q) = \text{front}(q')$
- d. $\text{back}(\text{empty}) = \text{error}$
- e. $\forall e : \text{Element}, q' : \text{Queue} . \text{back}(\text{enqueue}(e, q')) = e$
Can be written as: $\text{back}(q) = e$
- f. $\text{dequeue}(\text{empty}) = \text{empty}$
- g. $\text{dequeue}(\text{enqueue}(\text{Element}, \text{empty})) = \text{empty}$
- h. $\forall e : \text{Element}, q' : \text{Queue} . \text{dequeue}(\text{enqueue}(e, q')) = \text{enqueue}(e, \text{dequeue}(q'))$
Can be written as: $\text{dequeue}(q) = \text{enqueue}(e, \text{dequeue}(q'))$

END SOLUTION