CS135 Sample Test 1 —with solution guide—

Closed book: no textbook, no electronic devices. You may have one sheet of paper with notes —which must be handed in with the test.

Question 1 *(15 points)* Suppose the predicate $Fool(x, y, t)$ means person x can fool person y at time t. For example, $\forall x\ \exists t\ Fool(x, x, t)$ means that everyone can fool themself sometimes. Using quantifiers, write formulas that express these ideas:

There's always someone who can fool Dave.

There's someone who can fool everyone all the time.

Everyone can be fooled sometimes.

SOLUTION

- There's always someone who can fool Dave. $\forall t\ \exists x\ Fool(x, Dave, t)$.

  Wrong: $\exists x\ \forall t\ Fool(x, Dave, t)$ means there's someone who can fool Dave all the time.

- There's someone who can fool everyone all the time. $\exists x\ \forall y\ \forall t\ Fool(x, y, t)$

  Or equivalently $\exists x\ \forall t\ \forall y\ Fool(x, y, t)$, which can be abbreviated $\exists x\ \forall t, y\ Fool(x, y, t)$.

- Everyone can be fooled sometimes. $\forall x\ \exists y\ \exists t\ Fool(y, x, t)$ (Think of "x can be fooled" as meaning "There is someone who can fool x.")

Question 2 *(10 points)* Let f be a function $f : \mathbf{Z} \to \mathbf{N}$. Write a logical formula that expresses that "f is surjective" (i.e., onto).

SOLUTION Here is one solution: $\forall x\ \exists y\ f(y) = x$.
To be very clear about the domains for the quantified variables, we can write $\forall x : \mathbf{N}\ \exists y : \mathbf{Z}\ f(y) = x$, or use the $\in$ symbol in place of :.

Question 3 *(5 points)* What is the value of this Scheme expression:
`(caddr '(solar (decathalon (ribbon)) cutting))`

SOLUTION `cutting`
(that is, the literal atom `cutting`, which you may also write as `'cutting`)

Question 4 *(10 points)* Trace the execution of this Scheme expression: `(append '(3 2 1) '(4 5))` where `append` is defined as follows.

```
(define (append xs ys)
 (cond [(null? xs) ys]
       [else (cons (car xs) (append (cdr xs) ys))]))
```

SOLUTION

```
  (append '(3 2 1) '(4 5))
= (cons 3 (append '(2 1) '(4 5)))
= (cons 3 (cons 2 (append '(1) '(4 5))))
= (cons 3 (cons 2 (cons 1 (append '() '(4 5)))))
= (cons 3 (cons 2 (cons 1 '(4 5))))
= (cons 3 (cons 2 '(1 4 5)))
= (cons 3 '(2 1 4 5)))
= '(3 2 1 4 5)
```

Please use this format, with the = signs!!! It tells the reader what's the connection between the formulas.

Notice that I immediately simplified car/cdr expressions above. In full detail the trace would look like:

```
  (append '(3 2 1) '(4 5))
= (cons (car '(3 2 1)) (append (cdr '(3 2 1)) '(4 5)))   by def append
= (cons 3 (append '(2 1) '(4 5)))                         simplify car/cdr
...
```

Question 5 *(10 points)* The following is a tautology: $(p \wedge (p \rightarrow q)) \rightarrow q$.
Prove that it is a tautology, using a truth table.

p	q	$p \rightarrow q$	$p \wedge (p \rightarrow q)$	$(p \wedge (p \rightarrow q)) \rightarrow q$
F	F	T	F	T
F	T	T	F	T
T	F	F	F	T
T	T	T	T	T

Question 6 *(20 points)* Prove that $(p \wedge (p \rightarrow q)) \rightarrow q$ is a tautology, by calculating using the laws of propositional logic. That is, simplify the formula to *true*. Give a hint to justify each step. If you want to use laws not listed at the back of the exam, that's ok, but prove them too.

SOLUTION
Here's one solution. I've underlined the part that I rewrote, in each step.

$$
\begin{array}{lll}
 & \underline{(p \wedge (p \rightarrow q)) \rightarrow q} & \\
\equiv & \neg(p \wedge \underline{(p \rightarrow q)}) \vee q & \text{definition of } \rightarrow \\
\equiv & \underline{\neg(p \wedge (\neg p \vee q))} \vee q & \text{definition of } \rightarrow \\
\equiv & \neg p \vee \underline{\neg(\neg p \wedge q)} \vee q & \text{De Morgan } (\neg \text{ over } \wedge)\text{, and } \vee \text{ assoc.} \\
\equiv & \neg p \vee (\underline{\neg\neg p} \wedge \neg q) \vee q & \text{De Morgan } (\neg \text{ over } \vee) \\
\equiv & \underline{\neg p \vee (p \wedge \neg q)} \vee q & \text{double neg.} \\
\equiv & \neg p \vee \neg q \vee q & \text{lemma: } \neg x \vee (x \wedge y) \equiv \neg x \vee y\text{, see below} \\
\equiv & \neg p \vee T & \text{excluded middle} \\
\equiv & T & T \text{ is the ``zero element'' of } \vee
\end{array}
$$

Where I used associativity, it justifies omitting parentheses.

Now I need to prove the LEMMA: $\neg x \vee (x \wedge y) \equiv \neg x \vee y$; here goes:

$$\begin{array}{lll} & \neg x \vee (x \wedge y) & \\ \equiv & (\neg x \vee x) \wedge (\neg x \vee y) & \text{distribute} \\ \equiv & T \wedge (\neg x \vee y) & \text{excluded middle} \\ \equiv & \neg x \vee y & \text{identity of } \wedge \end{array}$$

I could have just done that reasoning inside the main proof, but the lemma seemed like an interesting law I might want to remember.

Here's another solution, from Amanda Kowalski, Christopher Kelley and others, which I like better than mine.

$$\begin{array}{lll} & (p \wedge (p \rightarrow q)) \rightarrow q & \\ \equiv & (p \wedge (\neg p \vee q)) \rightarrow q & \text{definition of } \rightarrow \\ \equiv & ((p \wedge \neg p) \vee (p \wedge q)) \rightarrow q & \text{distribution (of } \wedge \text{ over } \vee) \\ \equiv & (F \vee (p \wedge q)) \rightarrow q & \text{contradiction (negation law)} \\ \equiv & (p \wedge q) \rightarrow q & \vee \text{ commutative, } \vee \text{ identity element} \end{array}$$

(Some of the parentheses are omittable since $\wedge$ binds more tightly than $\rightarrow$.) It's reasonable to stop right here, because $p \wedge q \rightarrow q$ is a well-known tautology. Strictly speaking, though it's not in the list at the back of the test so we need to prove it. Here's one proof:

$$\begin{array}{lll} & p \wedge q \rightarrow q & \\ \equiv & \neg(p \wedge q) \vee q & \text{definition of } \rightarrow \\ \equiv & \neg p \vee \neg q \vee q & \text{de Morgan (and } \vee \text{ assoc.)} \\ \equiv & \neg p \vee T & \text{excluded middle (negation law)} \\ \equiv & T & \text{“zero element” of } \vee \text{ (Rosen says: domination law)} \end{array}$$

Question 7 *(5 points)*

Write out the elements of the powerset of $\{1, 2, 3\}$, in mathematical notation.

SOLUTION $\{ \{1, 2, 3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1\}, \{2\}, \{3\}, \emptyset \}$ (or in another order, or even with duplicates although that seems pointless).

Wrong: this set does not include $\{\emptyset\}$. But it's ok to write { } instead of $\emptyset$.

This is not a solution: `((1 2 3) (1 2) (1 3) (2 3) (1) (2) (3) () )` since the question asks for math notation, not the Scheme representation.

Question 8 *(5 points)*

Write out the elements of the set $\{1, 2\} \times \{"fish", "blue"\}$.

SOLUTION

$\{(1, "fish"), (2, "fish"), (1, "blue"), (2, "blue")\}$

Note that it's as set (notation $\{\ldots\}$) but the elements are ordered pairs.

It's fine to rearrange the pairs, e.g.,: $\{(2, "blue"), (1, "fish"), (2, "fish"), (1, "fish")\}$ but wrong to reorder the pairs —WRONG: $\{("fish", 1), ("fish", 2) \ldots$ and wrong to write pairs as sets: $\{\{"fish", 1\}\{"fish", 2\} \ldots$

Question 9 *(10 points)* Consider this code:

```
(define (make-ones n)          ; list of n ones, assuming n>=0
  (cond [(eqv? n 0) '()]
        [else (cons 1 (make-ones (pred n)))]
        ))
(define (product lon)          ; product of a list of numbers
  (cond [(null? lon) 1]
        [else (* (car lon) (product (cdr lon)))]))
```

Here is a fact: $\boxed{\texttt{(product (make-ones n)) = 1} \text{ for any natural number } \texttt{n}}$

As part of a proof by induction, *state and prove the base case.* SOLUTION

Base case is `(product (make-ones 0)) = 1`, i.e., the instance with `n = 0`.

Proof of base case:

```
  (product (make-ones 0))
= (product '())            by def make-ones
= 1                        by def product
```

Question 10 *(20 points)* Following on from the previous question, you will complete the proof by induction. Here is the induction step and induction hypothesis, for any `k`>0.

Ind. Hyp.: `(product (make-ones (- k 1))) = 1`
Ind. Case: `(product (make-ones k)) = 1`

Prove the inductive case, by transforming `(product (make-ones k))` to `1`. Say what justifies each step. You may use this fact about the `product` function.

P-Fact: $\boxed{\texttt{(product (cons n lon)) = (* n (product lon))} \text{ for any } \texttt{n} \text{ and any } \texttt{lon}.}$

SOLUTION

```
  (product (make-ones k))
= (product (cons 1 (make-ones (- k 1)))   by def make-ones
= (* 1 (product (make-ones (- k 1)))      by P-fact
= (* 1 1)                                 by ind hyp
= 1                                       by arith
```

I don't know why, but some people profer the following format, which is also ok. (I'll use == to mean logical equivalence. Do you see why?)

```
   (product (make-ones k))                = 1
== (product (cons 1 (make-ones (- k 1))) = 1  by def make-ones
== (* 1 (product (make-ones (- k 1)))    = 1  by P-fact
== (* 1 1)                               = 1  by ind hyp
== 1                                     = 1  by arith
== T                                          by logic
```

So we showed the equation is equivalent to True. The first calculation took less writing and did the same thing.

Some laws of propositional logic

Binding power: $\wedge, \vee$ bind more tightly than $\rightarrow$, less tightly than $\neg$.

$\neg(\neg p)$	$\equiv$	p	double negation
$p \wedge p$	$\equiv$	p	idempotent laws
$p \vee p$	$\equiv$	p	
$p \wedge T$	$\equiv$	p	identity elements
$p \vee F$	$\equiv$	p	
$p \wedge F$	$\equiv$	F	zero elements ('domination laws')
$p \vee T$	$\equiv$	T	
$p \wedge q$	$\equiv$	$q \wedge p$	commutativity
$p \vee q$	$\equiv$	$q \vee p$	
$p \vee \neg p$	$\equiv$	T	negation laws (excluded middle and contradiction)
$p \wedge \neg p$	$\equiv$	F	
$p \wedge (q \wedge z)$	$\equiv$	$(p \wedge q) \wedge z$	associativity
$p \vee (q \vee z)$	$\equiv$	$(p \vee q) \vee z$	
$p \vee (p \wedge q)$	$\equiv$	p	absorption
$p \wedge (p \vee q)$	$\equiv$	p	
$p \vee (q \wedge z)$	$\equiv$	$(p \vee q) \wedge (p \vee z)$	distributive laws
$p \wedge (q \vee z)$	$\equiv$	$(p \wedge q) \vee (p \wedge z)$	
$\neg(p \wedge q)$	$\equiv$	$(\neg p) \vee (\neg q)$	De Morgan's laws
$\neg(p \vee q)$	$\equiv$	$(\neg p) \wedge (\neg q)$	
$p \rightarrow q$	$\equiv$	$\neg p \vee q$	definition of $\rightarrow$
$p \leftrightarrow q$	$\equiv$	$(p \rightarrow q) \wedge (q \rightarrow p)$	definition of $\leftrightarrow$