

My Courses All Courses About Contact



Help

vitorpbarbosa7



Previous			Ē		Ø		Ē		B		Ø		Next	>
Questions														
Bookmark th	is page													
lot														
/1 point (graded)														
lyssa and Ben are	studying	the recursi	ive datatyp	e for Bool	ean formul	as.								
en suggests chan ounterexamples?	-	Not varia	nt from N	ot(formu	la:Formu	la) to N	lot(var:V	ariable)	. Which of	these for	nulas shou	ld Alyssa o	ffer as a	
	$\neg P \lor R)$													
$ (P \vee Q) \wedge ($	$P \vee \neg R$)													
	$\neg P \vee \neg R)$													
$(P \lor Q) \land \neg$														
(F V Q) X =	(F V K)													
$\checkmark \neg ((P \lor Q))$	$\wedge (P \vee R)$)												
✓														
Explanation														
Vith this change, I	Ben can on	nly represe	nt negated	variables,	not negate	ed (sub-)fo	rmulas.							
Vith this change, I	3en can on	nly represe	nt negated	variables,	not negate	ed (sub-)fo	rmulas.						Show A	
Submit				variables,	not negate	ed (sub-)fo	rmulas.							
Submit				variables,	not negate	ed (sub-)fo	rmulas.							
Submit 3 Answers are				variables,	not negate	ed (sub-)fo	rmulas.							
Submit Answers are Pair /1 point (graded)	displayed	within the	problem										Show A	nsw
Submit Answers are Pair /1 point (graded) When we write Er	displayed	within the	problem ist <varia< td=""><td>able x B</td><td>oolean> ,</td><td>we mean</td><td>that an Er</td><td>nvironme</td><td>nt is imple</td><td>emented a</td><td>san ImLi</td><td>st oftwo</td><td>Show A</td><td>nsw</td></varia<>	able x B	oolean> ,	we mean	that an Er	nvironme	nt is imple	emented a	san ImLi	st oftwo	Show A	nsw
Submit Answers are Pair /1 point (graded) When we write Erak.a. pairs) where	displayed nvironmen e the first e	within the	problem ist <varia< td=""><td>able x B</td><td>oolean> ,</td><td>we mean</td><td>that an Er</td><td>nvironmen</td><td>nt is imple</td><td>emented a</td><td>san ImLi</td><td>st oftwo</td><td>Show A</td><td>nsw</td></varia<>	able x B	oolean> ,	we mean	that an Er	nvironmen	nt is imple	emented a	san ImLi	st oftwo	Show A	nsw
Answers are Pair /1 point (graded) /then we write Erak.a. pairs) where then we wrote sta	displayed nvironmen e the first e tements lil	within the nt = ImL element is ke:	problem ist <varia a Variab</varia 	able x B le and th	oolean> , ne second i	we mean sa Boole	that an Er		nt is imple	emented a	san ImLi	st of two	Show A	nsw
Submit Answers are Pair /1 point (graded) When we write Era.k.a. pairs) where then we wrote state lookup (Cons (() which says: when we	displayed nvironmene e the first e tements lil var, val we look up	within the nt = ImL element is ke: L), rest)	problem ist <varia a="" b,="" nvironme<="" td="" variab="" x)="i"><td>able x B le and th .f var = nt whose</td><td>oolean>, ne second i x then v</td><td>we mean sa Boole ral else</td><td>thatan Erean .</td><td>est, x)</td><td></td><td></td><td></td><td></td><td>Show A</td><td>nsw</td></varia>	able x B le and th .f var = nt whose	oolean>, ne second i x then v	we mean sa Boole ral else	thatan Erean .	est, x)					Show A	nsw
Answers are Pair /1 point (graded) /hen we write Era.k.a. pairs) where hen we wrote sta lookup (Cons ((which says: when very consent (or consent) (or co	displayed nvironmen e the first e tements lil var, val we look up which is ju	within the nt = ImL element is ke: L), rest; D xin an E stan ImL	problem ist <variab td="" variab="" variab<=""><td>able x B le and th .f var = nt whose ese pairs).</td><td>oolean>, ne second i x then v e first pair i</td><td>we mean sa Boole ral else s (var, val),</td><td>that an Er ean . lookup(r if var is x, t</td><td>est, x)</td><td></td><td></td><td></td><td></td><td>Show A</td><td>nsw</td></variab>	able x B le and th .f var = nt whose ese pairs).	oolean>, ne second i x then v e first pair i	we mean sa Boole ral else s (var, val),	that an Er ean . lookup(r if var is x, t	est, x)					Show A	nsw
Answers are Pair /1 point (graded) When we write Era.k.a. pairs) where then we wrote state lookup (Cons ((which says: when ye Environment ((when ye Enviro	displayed nvironmele e the first e tements lil var, val we look up which is ju	within the nt = ImL element is ke: L), rest) o xin an E st an ImL t have tupl	problem ist <variab td="" variab="" variab<=""><td>able x B le and th of var = nt whose ese pairs). ould we do</td><td>oolean>, ne second i x then v e first pair i</td><td>we mean sa Boole ral else s (var, val),</td><td>that an Er ean . lookup(r if var is x, t</td><td>est, x)</td><td></td><td></td><td></td><td></td><td>Show A</td><td>nsw</td></variab>	able x B le and th of var = nt whose ese pairs). ould we do	oolean>, ne second i x then v e first pair i	we mean sa Boole ral else s (var, val),	that an Er ean . lookup(r if var is x, t	est, x)					Show A	nsw
Answers are Pair /1 point (graded) /hen we write Era.k.a. pairs) where then we wrote sta lookup (Cons ((which says: when we renvironment) (which says: whe	displayed nvironmele e the first e tements lil var, val we look up which is ju va does no	within the nt = ImL element is ke: L), rest) o xin an E st an ImL t have tupl	problem ist <variab td="" variab="" variab<=""><td>able x B le and th of var = nt whose ese pairs). ould we do an>></td><td>oolean>, ne second i x then v e first pair i</td><td>we mean sa Boole ral else s (var, val),</td><td>that an Er ean . lookup(r if var is x, t</td><td>est, x)</td><td></td><td></td><td></td><td></td><td>Show A</td><td>nsw</td></variab>	able x B le and th of var = nt whose ese pairs). ould we do an>>	oolean>, ne second i x then v e first pair i	we mean sa Boole ral else s (var, val),	that an Er ean . lookup(r if var is x, t	est, x)					Show A	nsw
Answers are Pair /1 point (graded) When we write Era.k.a. pairs) where then we wrote state lookup (Cons ((which says: when we conserved to the conserved	displayed nvironmene e the first e tements lil var, val we look up which is ju va does no t = ImList< t = ImList<	within the nt = ImL element is ke: L), rest) o xin an E st an ImL t have tupl c List< Varia	problem ist <variab td="" variab="" variab<=""><td>able x B le and th of var = nt whose ese pairs). ould we do an>></td><td>oolean>, ne second i x then v e first pair i</td><td>we mean sa Boole ral else s (var, val),</td><td>that an Er ean . lookup(r if var is x, t</td><td>est, x)</td><td></td><td></td><td></td><td></td><td>Show A</td><td>nsw</td></variab>	able x B le and th of var = nt whose ese pairs). ould we do an>>	oolean>, ne second i x then v e first pair i	we mean sa Boole ral else s (var, val),	that an Er ean . lookup(r if var is x, t	est, x)					Show A	nsw
Submit Answers are Pair /I point (graded) /then we write Era.k.a. pairs) where then we wrote state lookup (Cons () /thich says: when we Environment () Inlike Python, Jav Environment Environment Environment	displayed nvironmene e the first e tements lil var, val we look up which is ju va does no t = ImList< t = ImList<	within the nt = ImL element is. ke: L), rest) xin an E st an ImL t have tupl List< Varia List< Strin	problem ist <variable (ist)="" a="" able,="" able,<="" boolean="" case="" of="" td="" the="" variable=""><td>able x B le and th of var = nt whose ese pairs). ould we do an>></td><td>oolean>, ne second i x then v e first pair i</td><td>we mean sa Boole ral else s (var, val),</td><td>that an Er ean . lookup(r if var is x, t</td><td>est, x)</td><td></td><td></td><td></td><td></td><td>Show A</td><td>nsw</td></variable>	able x B le and th of var = nt whose ese pairs). ould we do an>>	oolean>, ne second i x then v e first pair i	we mean sa Boole ral else s (var, val),	that an Er ean . lookup(r if var is x, t	est, x)					Show A	nsw
Answers are Pair 1/1 point (graded) When we write Er (a.k.a. pairs) where Then we wrote sta lookup(Cons)((which says: when very the says: whe	displayed nvironmene e the first e tements lil var, val we look up which is ju va does no t = ImList< t = ImList< t = ImList<	within the nt = ImL element is. ke: L), rest) o xin an E st an ImL t have tupl c List< Varia c List< Object Variable[]	problem ist <variable and="" td="" varia<="" variable=""><td>able x B le and th of var = nt whose ese pairs). ould we do an>></td><td>oolean>, ne second i x then v e first pair i</td><td>we mean sa Boole ral else s (var, val),</td><td>that an Er ean . lookup(r if var is x, t</td><td>est, x)</td><td></td><td></td><td></td><td></td><td>Show A</td><td>nsw</td></variable>	able x B le and th of var = nt whose ese pairs). ould we do an>>	oolean>, ne second i x then v e first pair i	we mean sa Boole ral else s (var, val),	that an Er ean . lookup(r if var is x, t	est, x)					Show A	nsw

Environment = ImList< String[]> Environment = ImList< Object[]> Explanation The first two options are not valid Java: List s have only one type of thing in them. And we need to store two different types, so the Object list and array are the only options that work. 0 Submit Show Answer Answers are displayed within the problem Pair problems 0/1 point (graded) The solutions for storing pairs in the previous question are pretty bad. What do they *not* provide? ✓ static checking of the types of objects in the pair dynamic checking of the types of objects in the pair ✓ static checking of the number of objects in the pair dynamic checking of the number of objects in the pair static checking that we only access valid indices in the pair dynamic checking that we only access valid indices in the pair Lists and arrays provide dynamic out-of-bounds checking, but that's it. Storing these pairs as sequences of <code>Object</code> s is not a good design. Submit Pair progress 4/4 points (graded) Let's define an abstract datatype | Pair<T, U> to store pairs of values. This is another generic type: the first element of the pair is of some unknown type we'll $call \ \ T \ , and \ the second is of unknown \ type \ \ U \ . For Boolean formula environments, we will use \ \ Pair < Variable, \ Boolean > \ .$ This ADT will be characterized by two operations: first, to retrieve the first element in the pair, and second, to retrieve the second element. To work on the concrete implementation, we'll write a datatype definition: Pair<T, U> = Pair(first:T, second:U) Pair has only one concrete variant. Fill in the blanks to define operations: first : FIRST_INPUTS -> FIRST_OUTPUTS second : SECOND_INPUTS -> SECOND_OUTPUTS FIRST_INPUTS Pair< T, U> \bigcirc T $\bigcup \mathsf{U}$ first \bigcirc t

○ u	
•	
FIRST_OUTPUTS	
Pair< T, U>	
● Т	
Ou	
first	
○t	
Ou	
•	
SECOND_INPUTS	
Pair< T, U>	
От	
Ou	
second	
○t	
Ou	
~	
SECOND_OUTPUTS	
Pair< T, U>	
От	
● U	
second	
○t	
○ u	
~	
Explanation These function definitions are in terms of types, so names like first, t, etc. are out. The input is a Pair (probably these wil will be this), and the output is the first or second element, which are of arbitrary, unknown types T and U.	ll be instance methods, and it
Submit	Show Answer
Answers are displayed within the problem	
<pre>Previous Next ></pre>	

△ ③ Some Rights Reserved

Open Learning Library
About
Contact
Accessibility
Twitter
All Courses
Why Support MIT Open Learning?
Help

Privacy Policy Terms of Service

© Massachusetts Institute of Technology, 2024