DOCUMENTATION BY
Adriel Kobe Nepomuceno
Dania Ashraf Mohammed

Java Documentation

Row Class

FIELDS	
Type	Name and Description
String	**minterm** -contains all minterm values that were combined to produce the current binary value (minterm values are comma-separated)
String	**binary** -contains the combined binary values of selected minterms
int	**group** -initially the group number based on the amount of 1s in the binary value, after comparison with another row, it takes the group of the row with the smaller group number
boolean	**used** -tells whether the row was combined with another during the current round of comparisons

CONSTRUCTOR
Row(String minterm, String binary, int group, boolean used) -constructs a row with the specified values of minterm, binary, group, and used **Parameters:** **minterm** – minterm value to initialize new row **binary** - binary value to initialize new row **group** - group value to initialize new row **used** - used value to initialize new row

METHODS	
Type	Name and Description
String	**getMinterm()** -returns minterm value of chosen row
void	**setMinterm(String minterm)** -sets chosen row's minterm value to the specified minterm **Parameters:** **minterm** – minterm value to change the minterm of the chosen row
String	**getBinary()** -returns binary value of chosen row
void	**setBinary(String binary)** -sets chosen row's binary value to the specified binary **Parameters:** **binary** – binary value to change the binary of the chosen row
int	**getGroup()** -returns group value of chosen row
void	**setGroup(int group)** -sets chosen row's group value to the specified group **Parameters:** **group** – group value to change the group of the chosen row
boolean	**getUsed()** -returns minterm value of chosen row
void	**setUsed(boolean used)** -sets chosen row's used value to the specified boolean value of used **Parameters:** **used** – used value to change the used value of the chosen row

Main Class

FIELDS	
Type	Name and Description
int	**numVars** -stores the number of variables for the given function
boolean	**noDuplicates** -stores the boolean value for whether or not there are duplicates in the list of rows for the current comparison of binary values
List<Row>	**rowContainer** -stores the resulting rows for the current comparison of binary values
String	**mintermResult** -stores the resulting binary value of comparing two binary values

METHODS	
Type	Name and Description
String[]	**isValidVars(Scanner sc)** -recursive function that checks whether input string of space-separated variables is valid, ends when the input is valid **Parameters:** **sc** – Scanner used to get user input **Returns:** -Array of strings containing the variables used for the function
boolean	**isNumber(String str)** -checks whether the given string is convertible to a double **Parameters:** **str** – String to be checked **Returns:** -true if the string can be converted to a double, false otherwise

String[]	**isValidMinterms(Scanner sc)** -recursive function that checks whether input string of space-separated minterms is valid, ends when the input is valid **Parameters:** **sc** – Scanner used to get user input **Returns:** -Array of strings containing the minterms used for the function
String	**getBinary(String s)** -converts a decimal digit to its binary value **Parameters:** **s** – String containing decimal you want to convert **Returns:** -String containing the binary value of the input
String	**normalize(String s, int length)** -left pads the binary value with 0s until it reaches the desired length **Parameters:** **s** – String containing binary value to be left padded **length** – desired length of the adjusted binary value **Returns:** -String containing binary value with the desired length
boolean	**isPair(String s1, String s2)** -compares two strings of binary values to check whether they only vary by a single bit **Parameters:** **s1** – first String containing a binary value to be compared **s2** – second String containing a binary value to be compared **Returns:** -true if the two Strings contain binary values that only vary by a single bit, and false otherwise
boolean	**containsPair(List<Row> r)** -compares all combinations of two rows and checks whether there are pairs whose binary values only vary by a single bit

	Parameters: r – list of all rows to be used for the current round of comparisons Returns: -true if the list contains pairs whose binary values only vary by a single bit, and false otherwise
void	getUnused(List<Row> r) -adds all rows whose used value is false to the list rowContainer Parameters: r – list of all rows to be used for the current round of comparisons
void	comparison(List<Row> r) - changes used values of rows that are part of a pair whose binary values only vary by a single bit to true -adds a new row to the list rowContainer with its minterm, binary, and group is taken from the pair and with the used value false Parameters: r – list of all rows to be used for the current round of comparisons
String	sortMinterms(String minterms) - sort the string containing the minterms so that they are in increasing order Parameters: minterms – String containing comma separated values of minterms Returns: - String containing comma separated values of minterms arranged in increasing order
boolean	hasDuplicates(List<Row> r) - checks whether the list of rows contains duplicates Parameters: r – list of all rows to be used for the current round of comparisons Returns: -true if the list contains duplicates, and false otherwise
void	removeDuplicate(List<Row> r) - removes duplicated values in the list of rows Parameters:

	r – list of all rows to be used for the current round of comparisons
boolean	**isIn(String input, String value)** -converts the string input to an int array of minterms, and checks whether the value of the string value is in the array **Parameters:** **input** – String of minterms to be checked **value**- String containing a single value to be checked **Returns:** -true if input contains value, false otherwise
String	**toTerm(String binary, String[] vars)** -converts the string binary to a term by changing all 0s to its primed variable counterpart, all 1s to its unprimed variable counterpart, and removing all "-" **Parameters:** **binary** – String containg a binary value **vars**- String array containing the variables used in the function **Returns:** -String of the term equivalent of a binary value