comp147-test

December 13, 2023

[ ]:
```python
# this function can convert
def graph_to_cnf(graph):
    clauses = [] # this one is the matrix
    num_nodes = len(graph) #check how many rows in this matrix

    # Each node must be colored with exactly one color
    for i in range(1, num_nodes + 1):
        clause = [f"{i * 3 - 2}", f"{i * 3 - 1}", f"{i * 3}"]
        clauses.append(" ".join(clause) + " 0")

    # two adjacent nodes can't have the same color
    for node, neighbors in enumerate(graph, start=1):
        for neighbor in neighbors:
            clause = [f"-{node * 3 - 2}", f"-{neighbor * 3 - 2}", "0"]
            clauses.append(" ".join(clause))

            clause = [f"-{node * 3 - 1}", f"-{neighbor * 3 - 1}", "0"]
            clauses.append(" ".join(clause))

            clause = [f"-{node * 3}", f"-{neighbor * 3}", "0"]
            clauses.append(" ".join(clause))

    return clauses

#input the number of rows of the matrix and the data in every row
def input_graph():
    list=[]
    n= int(input())
    for i in range(0,n) :
      #num = input()
      #list.append([int(i) for i in input().split()])
      list.append(input().split())
    return list

def print_cnf(clauses):
    for clause in clauses:
        print(clause)
```

```python


# Example usage
print("The row number of the matrix")
input_graph = input_graph()
print ("please enter the data in row")

    #Example graph representation
    # intput_graph = [
    #     [2,3],
    #     [1,3,4],
    #     [1,2],
    #     [4]
    # ]


def main():


    #convert string into integer
    input_graph1 = []
    for i in range(len(input_graph)):
      input_graph2 = []
      for j in range(len(input_graph[i])):
        a = ord(input_graph[i][j]) - 96
        input_graph2.insert(j,a)
      input_graph1.insert(i, input_graph2)


    #Convert graph to CNF
    cnf_clauses = graph_to_cnf(input_graph1)

   # Print the CNF clauses
    print_cnf(cnf_clauses)


if __name__ == "__main__":
    main()
```

```
The row number of the matrix
4
b c
a c d
a b
b
please enter the data in row
1 2 3 0
```

```
4 5 6 0
7 8 9 0
10 11 12 0
-1 -4 0
-2 -5 0
-3 -6 0
-1 -7 0
-2 -8 0
-3 -9 0
-4 -1 0
-5 -2 0
-6 -3 0
-4 -7 0
-5 -8 0
-6 -9 0
-4 -10 0
-5 -11 0
-6 -12 0
-7 -1 0
-8 -2 0
-9 -3 0
-7 -4 0
-8 -5 0
-9 -6 0
-10 -4 0
-11 -5 0
-12 -6 0
```

[ ]:
```
#Copy the content of cnf_clauses and paste it into the online MiniSAT solver:␣
 ↪https://msoos.github.io/cryptominisat_web/
```