```
class Queue:
    def __init__(self, capacity):
        self.capacity = capacity
        self.queue = []
        self.front = 0
        self.rear = -1
        self.size = 0

    def is_empty(self):
        """Check if the queue is empty."""
        return self.size == 0

    def is_full(self):
        """Check if the queue is full."""
        return self.size == self.capacity

   def enqueue(self, car):
    """Add a car to the end of the queue."""
      if self.is_full():
            print("Queue is full. Cannot add more cars.")
            return


     self.rear = self.rear + 1
     self.queue[self.rear] = car
     self.size += 1
     print(f"Car '{car}' added to the queue.")

    def dequeue(self):
        """Remove and return the car from the front of the queue."""
        if self.is_empty():
            print("Queue is empty. No car to remove.")
            return None

        car = self.queue[self.front]
        self.queue[self.front]=None
        self.front = self.front + 1
        self.size -= 1
        print(f"Car '{car}' removed from the queue.")
        return car

    def display_queue(self):
        """Display all cars in the queue."""
        if self.is_empty():
            print("Queue is empty.")
            return
```

```
        print("Current Queue of Cars:")
        for i in range(self.size):
            index = (self.front + i) % self.capacity
            print(f"- {self.queue[index]}")

# Example Usage
if __name__ == "__main__":
    # Create a queue with a maximum capacity of 5
    parking_queue = Queue(capacity=5)

    # Enqueue cars
    parking_queue.enqueue("Car A")
    parking_queue.enqueue("Car B")
    parking_queue.enqueue("Car C")

    # Display the queue
    parking_queue.display_queue()

    # Dequeue cars
    parking_queue.dequeue()
    parking_queue.dequeue()

    # Display the queue after dequeuing
    parking_queue.display_queue()

    # Enqueue more cars
    parking_queue.enqueue("Car D")
    parking_queue.enqueue("Car E")
    parking_queue.enqueue("Car F")

    # Display the queue after adding more cars
    parking_queue.display_queue()

    # Attempt to enqueue when the queue is full
    parking_queue.enqueue("Car G")  # Should indicate queue is full
```