# *Deadlock*
# *(not the metal band,*
# *not the game)*

When it absolutely, positively, isn't going anywhere, anytime, ever

# *Deadlock*

## Process 1

1. Reserve radar
2. Reserve launcher
3. Release radar
4. Release launcher

## Process 2

1. Reserve launcher
2. Reserve radar
3. Release launcher
4. Release radar

# *Deadlock*

## Process 1

1. Reserve radar
2. Reserve launcher
3. Release radar
4. Release launcher

## Process 2

1. Reserve launcher
2. Reserve radar
3. Release launcher
4. Release radar

# *Deadlock*

## Process 1

1. Reserve radar
2. Reserve launcher
3. Release radar
4. Release launcher

## Process 2

1. Reserve launcher
2. Reserve radar
3. Release launcher
4. Release radar

# *Deadlock*

## Process 1

1. Reserve radar
2. Reserve launcher
3. Release radar
4. Release launcher

## Process 2

1. Reserve launcher
2. Reserve radar - BLOCK
3. Release launcher
4. Release radar

# *Deadlock*

## Process 1

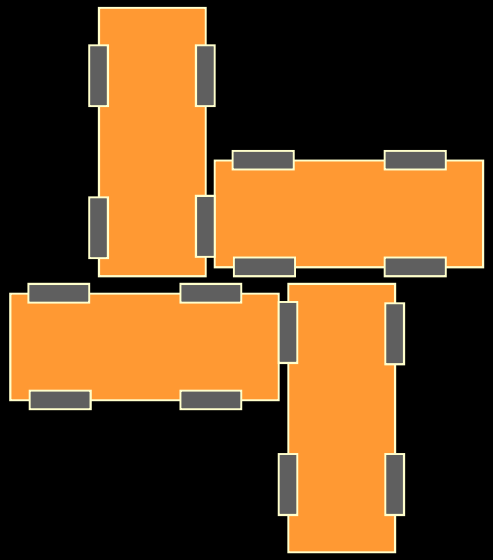
1. Reserve radar
2. Reserve launcher - BLOCK
3. Release radar
4. Release launcher

## Process 2

1. Reserve launcher
2. Reserve radar - BLOCK
3. Release launcher
4. Release radar

# Italian Traffic Deadlock

# Italian Traffic Deadlock

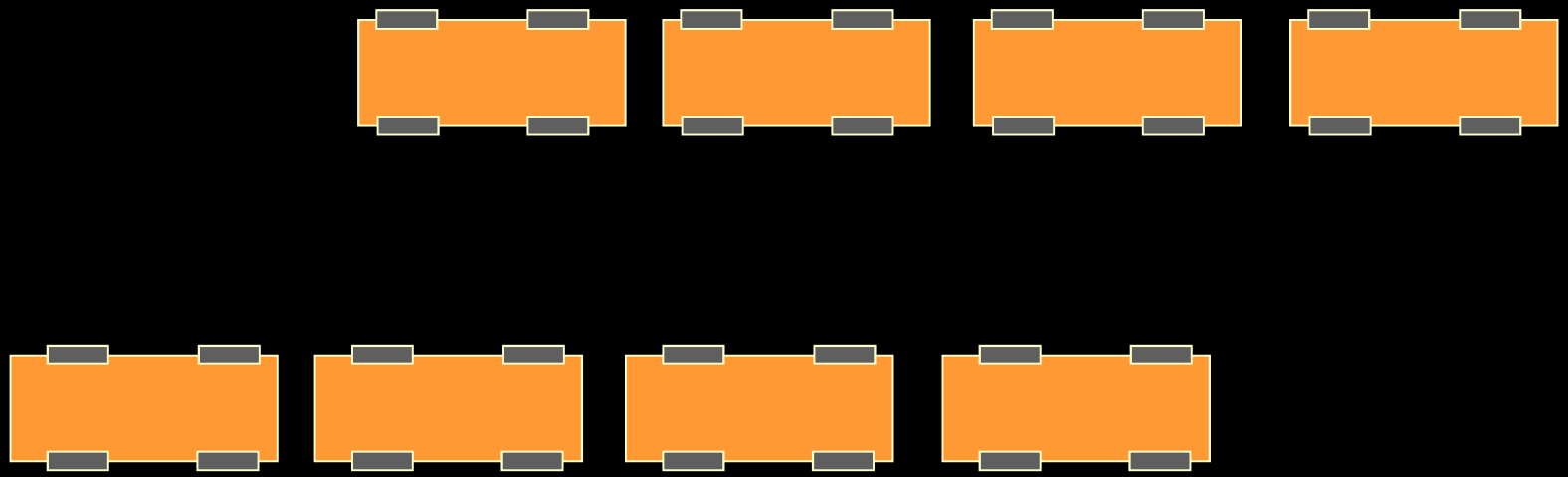


# New York Traffic Deadlock

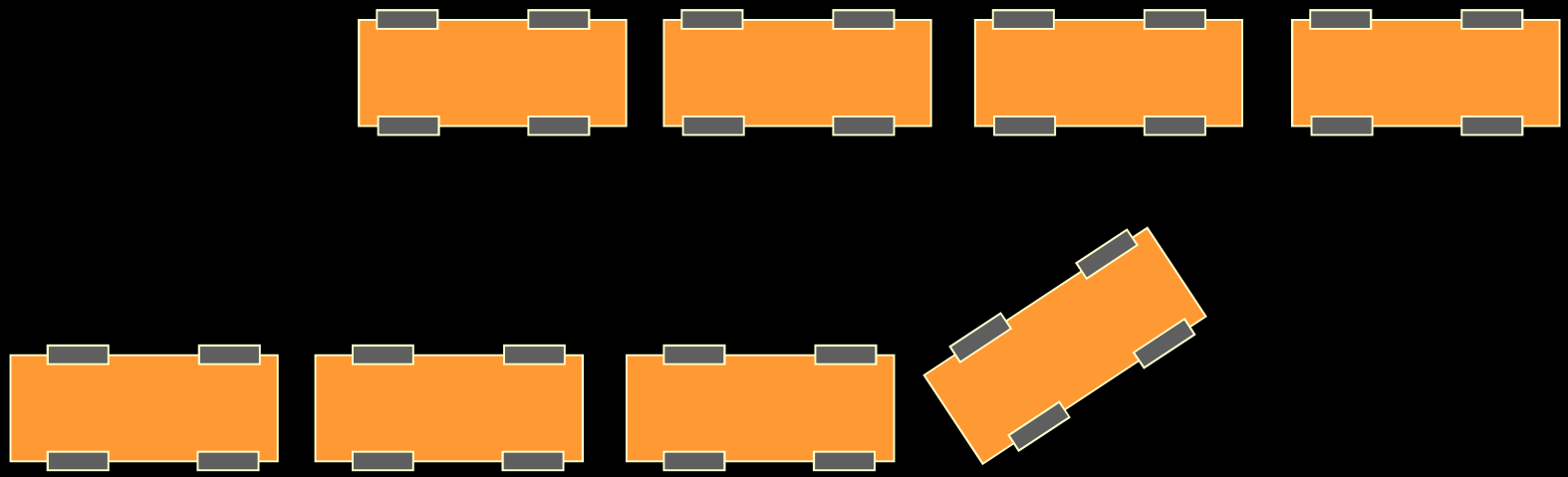
http://www.youtube.com/watch?v=vde81SqzmEo

# Route 109 Deadlock

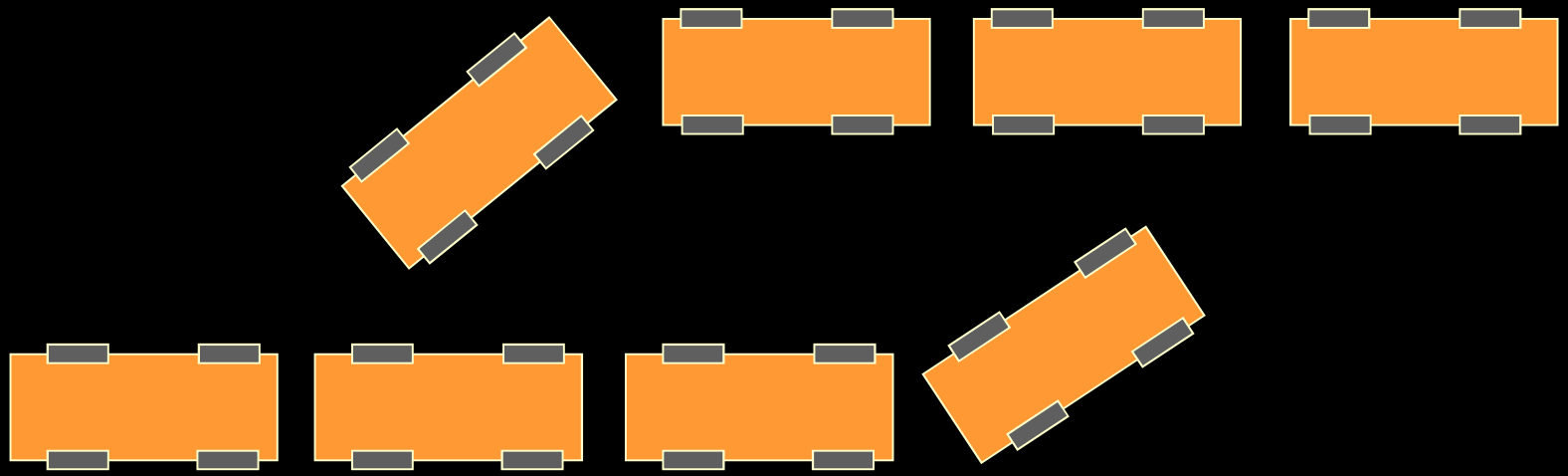
# Route 109 Deadlock



# Route 109 Deadlock



# Route 109 Deadlock



# *Dealing With Deadlock*

- Back someone up

# *Dealing With Deadlock*

- Back someone up
- Blow someone up


# *Dealing With Deadlock*

- Back someone up
- Blow someone up
- Reboot


# *Dealing With Deadlock*

- Back someone up
- Blow someone up


- Reboot
- Prevent it
  - Everyone requests resources in the same order

# *Dealing With Deadlock*

- Back someone up
- Blow someone up


- Reboot
- Prevent it
  - Everyone requests resources in the same order
  - Don't grant a resource request if it could lead to deadlock
    - How do you know?

# The Generalized Deadlock Resolution Problem

In this paper we initiate the study of the AND-OR directed feedback vertex set problem from the viewpoint of approximation algorithms. This AND-OR feedback vertex set problem is motivated by a practical deadlock resolution problem that appears in the development of distributed database systems. This problem also turns out be a natural generalization of the directed feedback vertex set problem. Awerbuch and Micali gave a polynomial time algorithm to find a minimal solution for this problem. Unfortunately, a minimal solution can be arbitrarily more expensive than the minimum cost solution. We show that finding the minimum cost solution is as hard as the directed Steiner tree problem (and thus $O(log_2 n)$ hard to approximate). On the positive side, we give algorithms which work well when the number of writers (AND nodes) or the number of readers (OR nodes) are small.

# No, YOU let go...