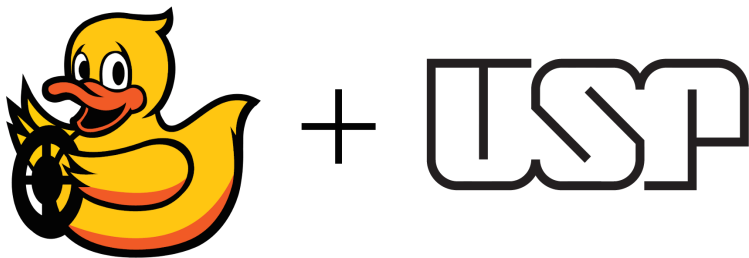


# Obedecendo leis de trânsito com mapa topológico



MAC0318 - Introdução à Programação de Robôs Móveis

Instituto de Matemática e Estatística (IME)  
Universidade de São Paulo (USP)

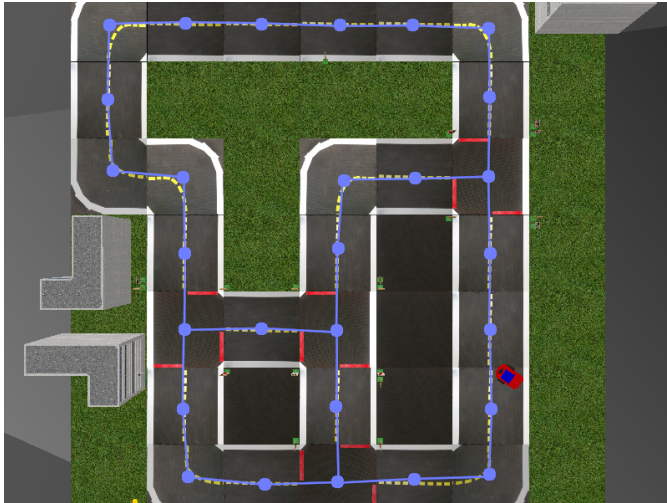
# Representação do mapa por grafo

Grafo  $G = (V, E)$ , onde:

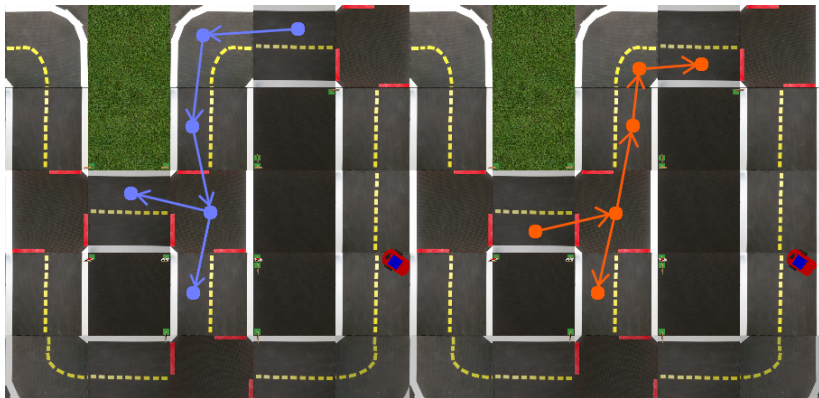
- ▶ Cada “pedaço” (*tile*) de rua é um nó  $v \in V$ ;
- ▶ Dois nós  $u, v \in V$  estão conectados por uma aresta se é possível ir de  $u$  até  $v$  sem sair da rua.

$G$  é uma representação ingênua. Como melhorar  $G$  para respeitar as leis de trânsito?

# Representação ingênua



# Representação por único nó em intersecção





# No código

```
class TopoGraph:
    def __init__(self, r):
        self._L = {}
        self._r = r

    def nodes(self):
        return list(self._L.keys())

    def edge(self, p, q):
        if self.invalid_tile(p, q):
            return False
        return q in self._L[p]

    def add_node(self, p):
        if p not in self._L:
            self._L[p] = {}

    def add_edge(self, p, q):
        self._L[p][q] = True
        self._L[q][p] = True

    def add_dir_edge(self, p, q):
        self._L[p][q] = True
```

```
    def bfs(self, p, q):
        p = self.closest_node(p)
        q = self.closest_node(q)
        Q = [p]
        V = {}
        Pa = {}
        while len(Q) != 0:
            n = Q.pop(0)
            if n == q:
                P = [q]
                while P[-1] != p:
                    P.append(Pa[P[-1]])
                for i, u in enumerate(P):
                    P[i] = u
                return P
            for c in self._L[n]:
                if c not in V:
                    V[c] = True
                    Pa[c] = n
                    Q.append(c)
        return None
```

# Tarefa

Dado o grafo não direcionado da representação “ingênua”, construa um grafo direcionado tal que o agente obedeça as seguintes regras de trânsito:

- ▶ Ande sempre na faixa da direita;
- ▶ Retorno não é permitido em retas ou curvas;
- ▶ Sempre dirija “para frente”.

Arquivo esqueleto: `assignments/topomap.py`