路口方案

目录

- 输出字段
  - 2.输出示例

参考 路口虚拟link&虚拟lane 中方案2的 link 和lane的需求or实例

评审：baidu-adu-lab-andes-1190 [general][perception] add junction & virtual topo generator

1. 输出字段

文件页：baidu/asd/anp-common *proto_rename

```
message TopoMap {
  optional apollo.common.Header header = 1;
  required UltraPose pose = 2;

  map<int32, LaneLine> id2lane_line_map = 3;
  map<int32, RoadBoundary> id2road_boundary_map = 4;
  map<int32, StopLine> id2stop_line_map = 5;
  map<int32, PedCrossing> id2ped_crossing_map = 6;
  map<int32, GroundArrow> id2ground_arrow_map = 7;

  map<int32, LaneCenterLine> id2lane_center_line_map = 8;
  map<int32, LinkCenterLine> id2link_center_line_map = 9;

  // topology
  repeated Link links = 10;
  repeated Junction junctions = 11;
}
```

```
message Junction {
  // 路口的几何属性
  required int32 junction_id = 1 [ default = -1 ];

  // 路口的其他属性
  repeated int32 link_ids = 2;
  repeated int32 ped_crossing_ids = 3;
  repeated int32 stop_line_ids = 4;
  repeated int32 curb_ids = 5;
}
```

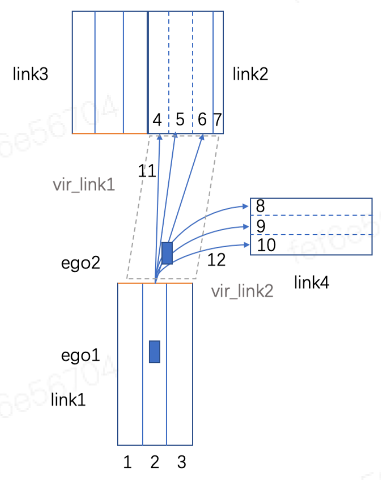
```
// LINK的其他属性
required int32 sd_link_id = 17 [ default = -1 ];
repeated int32 junction_id = 18;
repeated LinkDirection link_direction = 19;
required bool main_car_is_in_link = 20 [ default = false ];
repeated Segment segments = 21;

// LINK间的拓扑关系
repeated int32 precursor_link_ids = 22;
repeated int32 successor_link_ids = 23;
repeated LinkTurnDirection successor_link_turn_direction = 24;
repeated int32 left_link_ids = 25;
repeated int32 right_link_ids = 26;
```

```
// Lane间的拓扑关系
repeated int32 precursor_lane_ids = 22;
repeated int32 successor_lane_ids = 23;
repeated LaneTopo successor_lane_topo = 24; // 后继的车道拓扑类型
repeated int32 left_lane_ids = 25;
repeated int32 right_lane_ids = 26;
```

```
enum LaneTopo {
  LTP_UNKNOWN = 0;
  LTP_LANE_CONTINUITY = 1;
  LTP_LANE_BREAK = 2;
  LTP_LANE_CONTINUITY_ONLY_BY_LEFT = 3;
  LTP_LANE_CONTINUITY_ONLY_BY_RIGHT = 4;
  LTP_LANE_MERGE = 5;
  LTP_LANE_SPLIT = 6;
  LTP_LANE_CONTINUITY_BY_VIRTUAL = 7;
}
```

车道后继类型： 车道后继类型示例

2.输出示例



输出

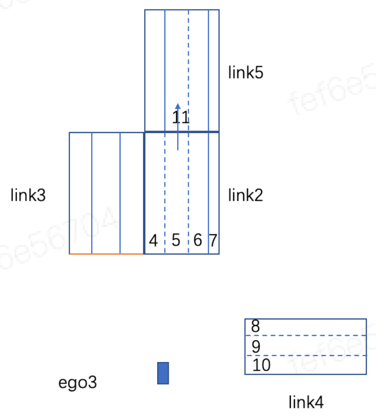
ego1/ego2-可以获取历史link

以主车道/历史主车道所在link为起点，构建拓扑

- Junction : link1, link2, link3, link4；vir_link1, vir_link2
  - link1: lane1, lane2, lane3
    - type: straight
    - successors: vir_link1, vir_link2
    - lane2
      - successors: vir_lane11, virtual_topo
        vir_lane12, virtual_topo
        ...
  - link4: lane1, lane2, lane3
    - type: turn right
      ...

    ...

注：拓扑关系建立在符合通行规则的基础上，依赖道路元素等信息，逐渐完善中



Ego3-路口中初始化，
没有历史link信息，也看不到之前的车道线:

以可通行link为起点link, 分别构建拓扑

- Junction : link2, link3, link4
  - link2: lane4, lane5, lane6
    - type: straight
    - successors: link5
    - lane5
      - successors: lane11，continue
      ...
  - link4: lane8, lane9, lane10
    ...