Turn Lanes

Introduction

Often enough at junctions a road will split off into different lanes containing different permissions for going through or turning into a different lane. Whenever they are clearly marked we need to model them.

Turn lanes should **not** be confused with situations where the road instead of being divided into different lanes, branches off completely into a separate way leading to another road. This is known as a link and is modeled in a completely different manner.

Turn:Lanes Key

The main values you will be using for the **turn:lanes** key are **left**, **right**, **through**, or **none**. Regardless of how many types of turn lanes may exist on a road segment, they are all recorded on the same **turn:lanes** key, though the values for each individual lane are separated by a vertical bar

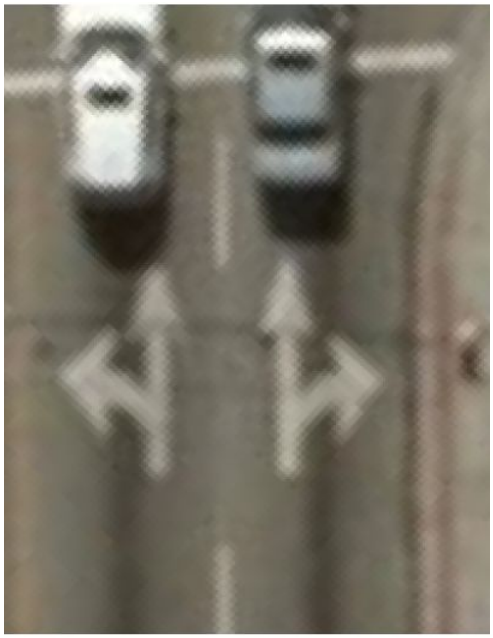
For Example:

turn:lanes= left|through|right

They are recorded in the order that they actually appear on the road going from left to right (left and right to the way direction)

Keep in mind that some lanes may allow for multiple values. For example, you could be on a lane where the arrow explicitly tells you that you can go through or turn left. In this case such values *within a lane* are divided with a semicolon rather than with a vertical bar, though the latter still divides the separate lanes.

For Example:



turn:lanes=left;through|through;right

Directions

One must also remember that many roads go in both directions and a way at a junction will only have turn:lanes going in one direction. In that case you must add either the **:forward** or **:backward** subkey to the **turn:lanes** key.

For Example:

Assume the direction of the road is going up

lanes=2
lanes:forward=2
lanes:backward=1
turn:lanes:forward=left|through;right

Alternatively if the direction of the way was going down then the tags would be:

lanes=2
lanes:forward=1
lanes:backward=2
turn:lanes:backward=left|through;right

Notice that the lanes are given in the same order as they were given when the directions were on the forward lane. It's counting from the center of the road, and the center of the road is always to your left regardless of the direction you are headed.

Let's try some more examples with this:

Assuming the way directionality is up:
lanes=4
lanes:forward=2
lanes:backward=2
turn:lanes:forward=left;through|right

Assuming the way directionality is down:
lanes=4
lanes:forward=2
lanes:backward=2
turn:lanes:backward=left;through|right

Assuming in both case that we're only concerned with the road above the red line:

Assuming the way directionality is up:
lanes=5
lanes:forward=3
lanes:backward=2
turn:lanes:forward=left|through|right

Assuming that the way directionality is down:
lanes=5
lanes:forward=2
lanes:backward=3
turn:lanes:backward=left|through|right

The 'none' value

All of the above examples have involved a combination of the **left**, **through**, and **right** values. The none value is still important however, and does come up often.

Whenever you use the **turn:lanes:** key, you have to account for **all** of the lanes going in the relevant direction. While the key is used whenever there is a turn lane, not all of the lanes on the segment may have any sort of turn markings. If the lane has no road markings indicating turn direction use the **none** value, even if you already know there are implied turn restrictions such as a middle lane, despite having no arrows, only being able to go through.

For example:

Assuming the direction of the way would be going up:
lanes=5
lanes:forward=3
lanes:backward=2
turn:lanes:forward=left|none|none

Alternatively instead of writing none, you can also leave the spaces between the vertical bars blank

ie the turn tag above would've been written as :
turn:lanes:forward=left||