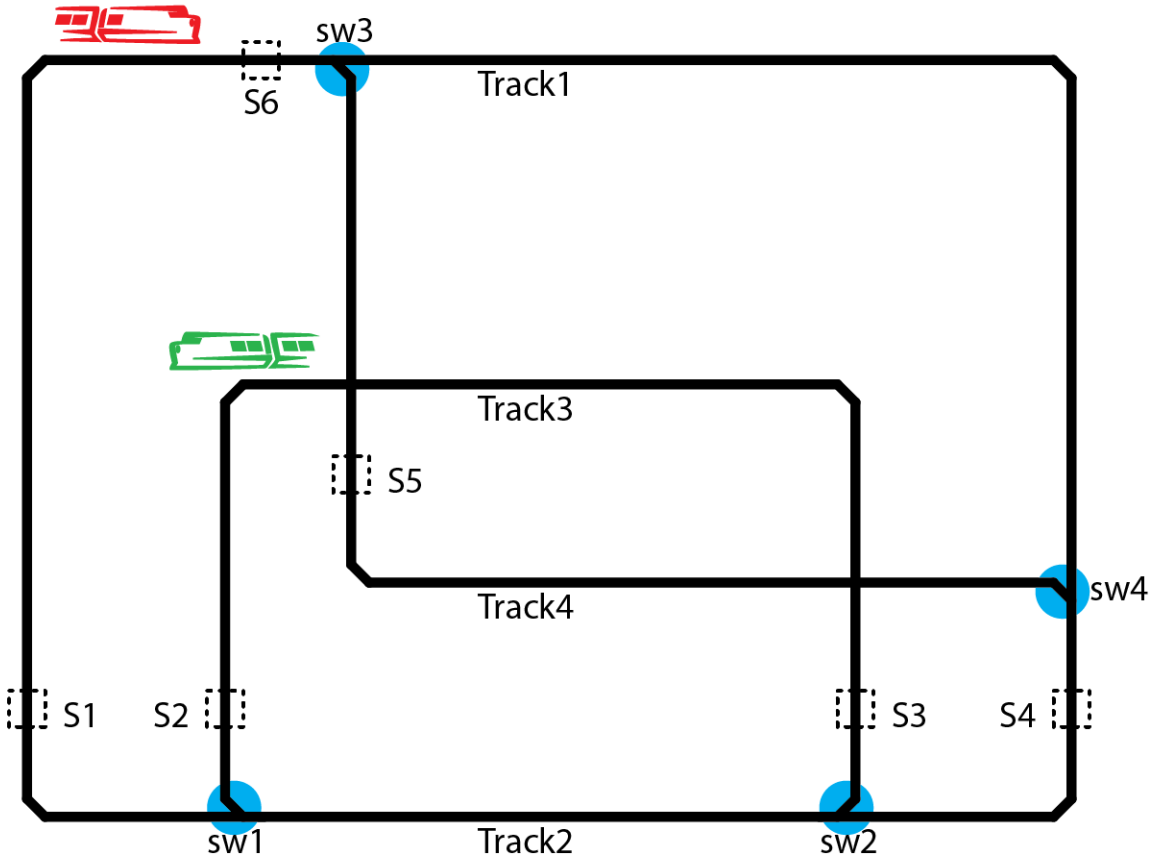
Spring 2015 Train Assignment

Digital Design Laboratory

Starting positions

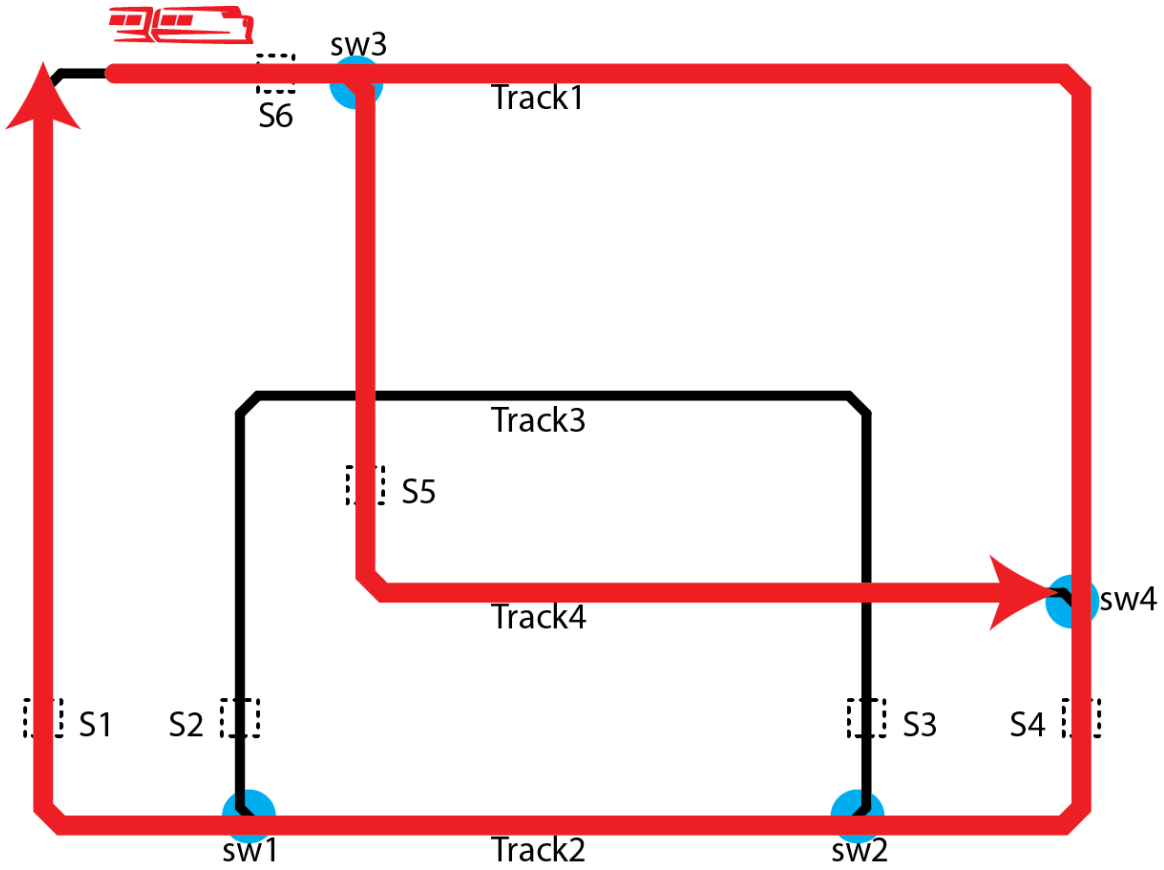


Both trains will start approximately where shown.

A (on Track 1) should always be moving in the “reverse” direction, as pointed.

B (on Track 3) should always be moving in the “forward” direction, as pointed.

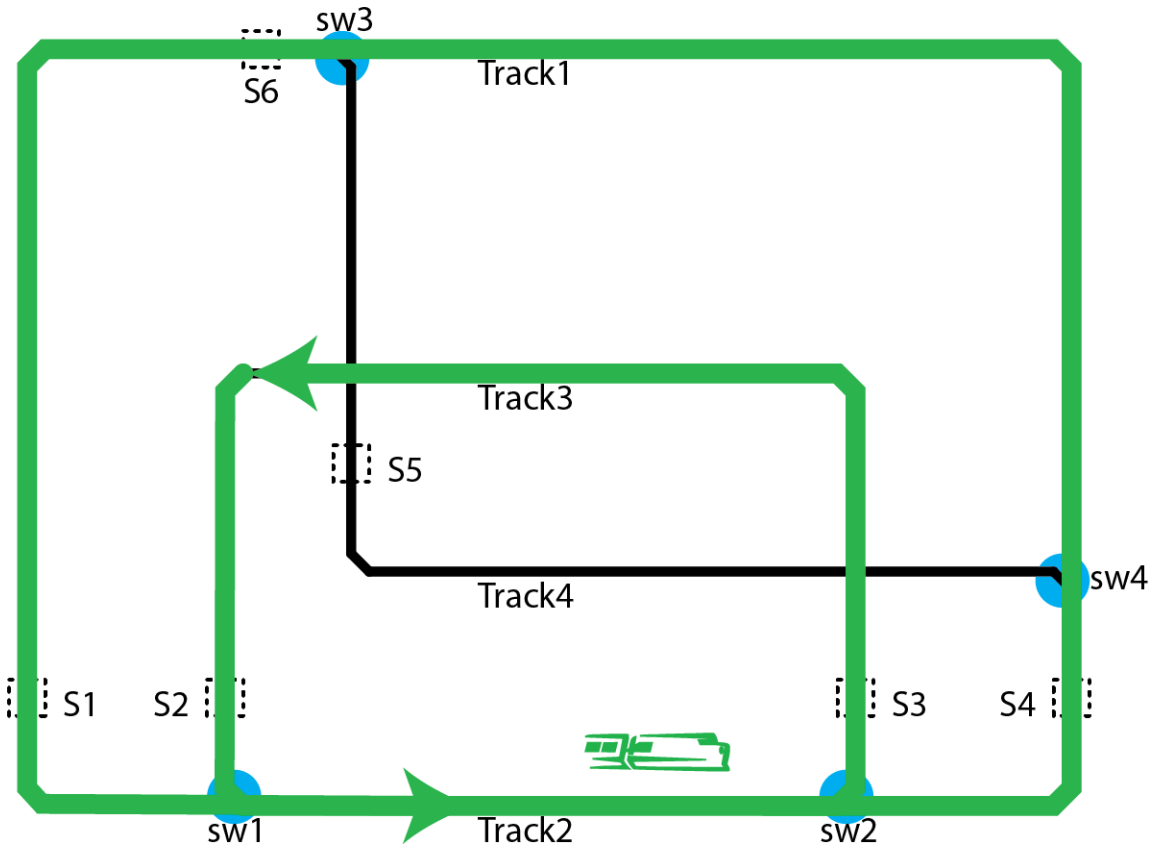
Path of Train A



Train A never leaves these sections of track, and it always goes in this direction.

It will stop under conditions to be described.

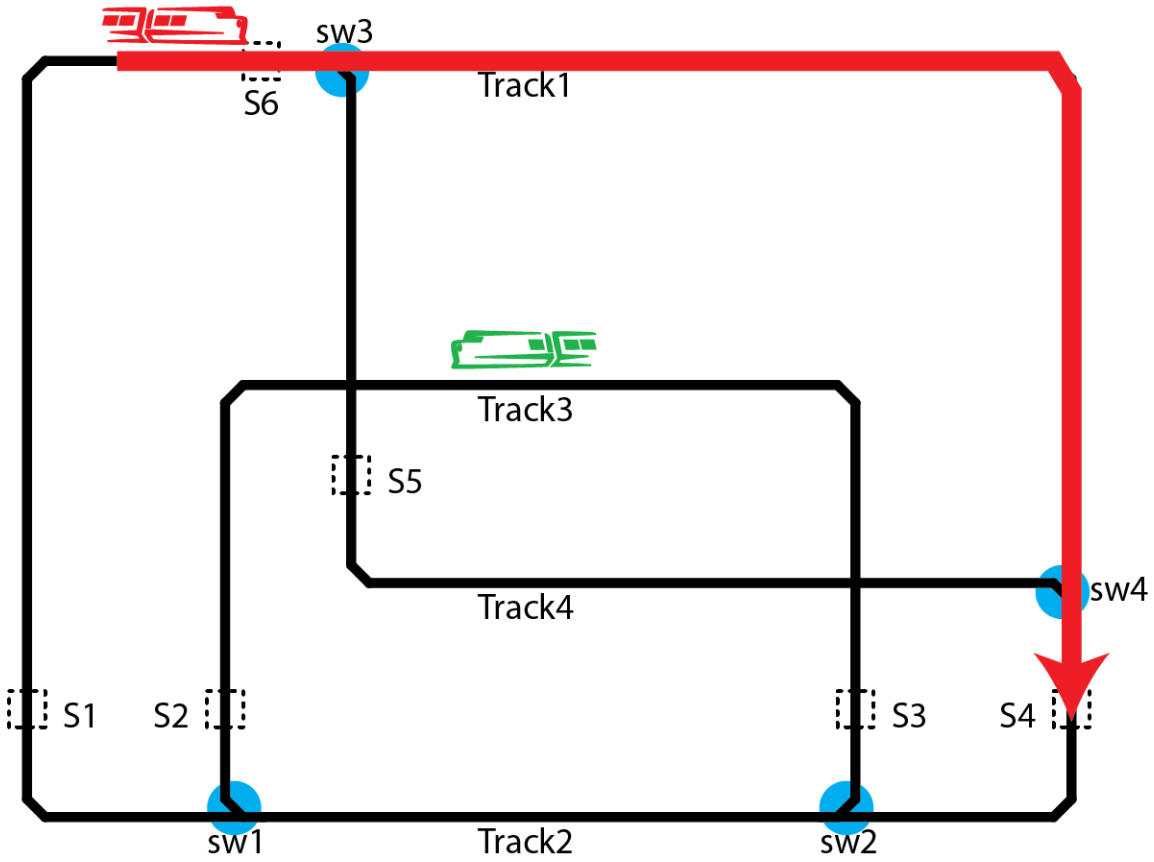
Path of Train B



Train B never leaves these sections of track, and it always goes in this direction.

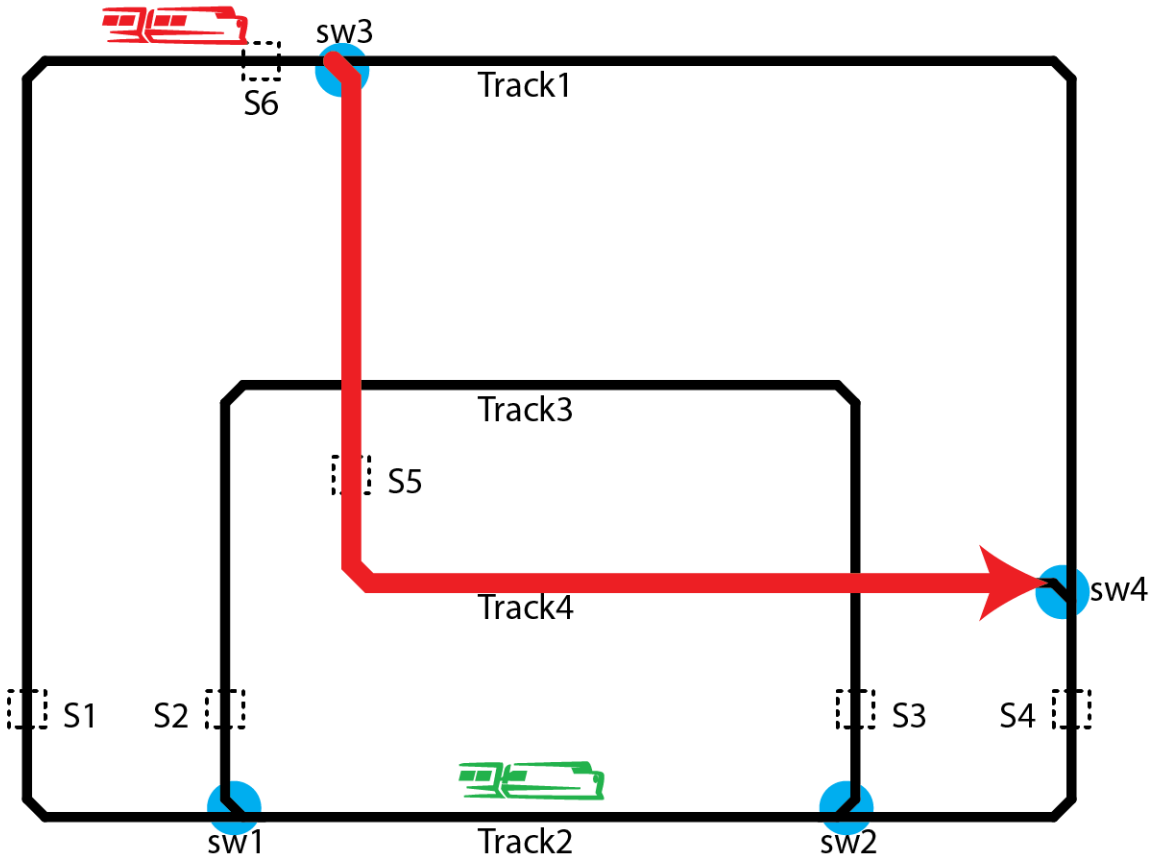
It will stop under conditions to be described.

Train A avoids Train B on Track3



At sw3 (passing S6), Train A takes Track1 if Train B is anywhere on Track3 (i.e., if B has passed S3, but not yet passed S2)

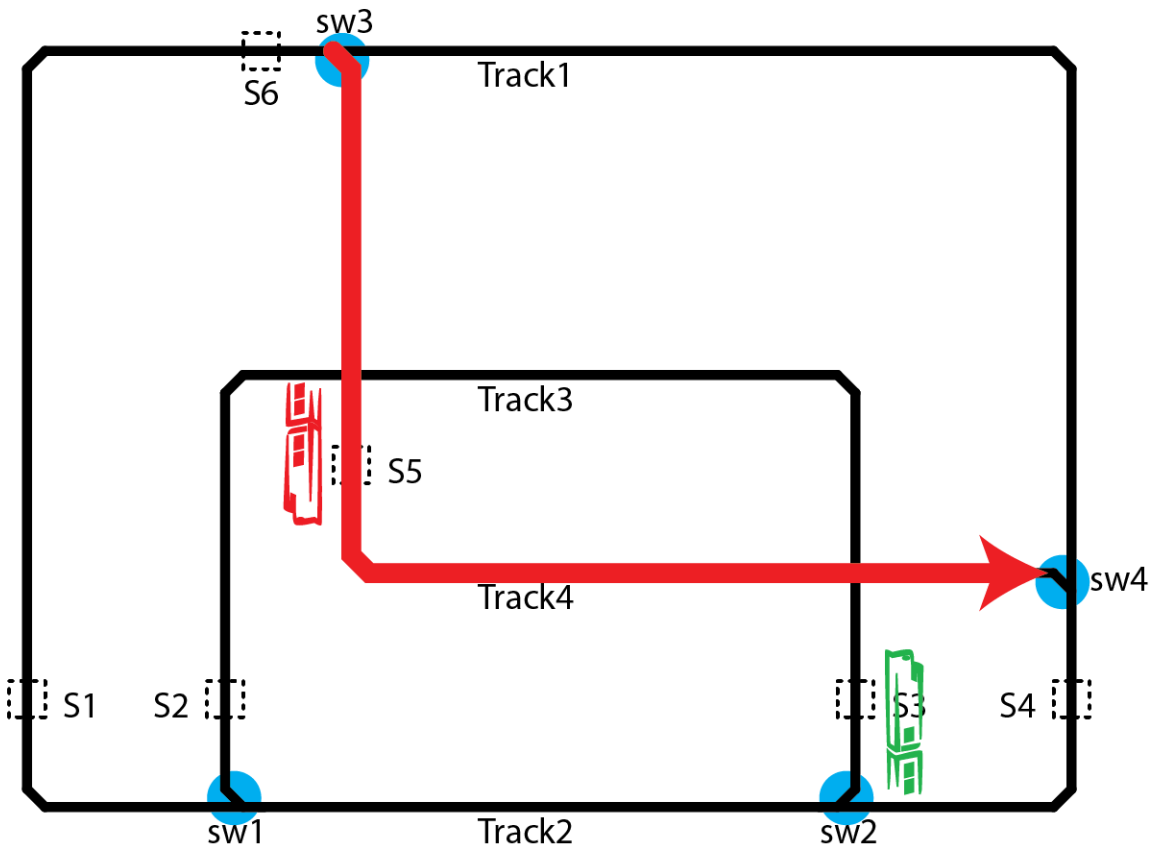
But Train A prefers Track4, if safe



If Train B is on Track2 when Train A passes S6, then Train A takes Track4

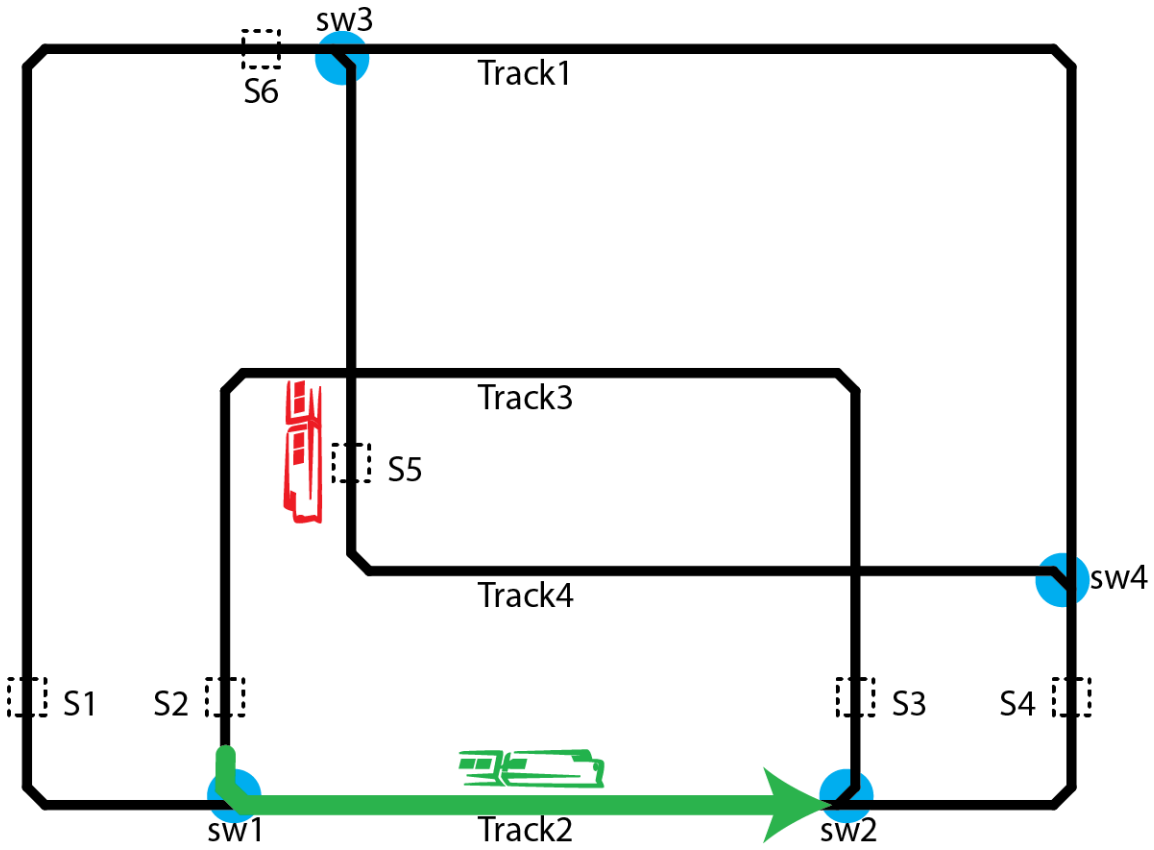
(i.e., if Train B has passed S1 or S2 to get on Track2, but has not yet passed S3 or S4 , then Train A takes Track4)

Train B cannot pass S3 when unsafe



If Train A is anywhere on Track4 (even stopped) when Train B reaches S3, Train B must stop. Only when Train A eventually clears Track 4 (passes S4) can Train B continue.

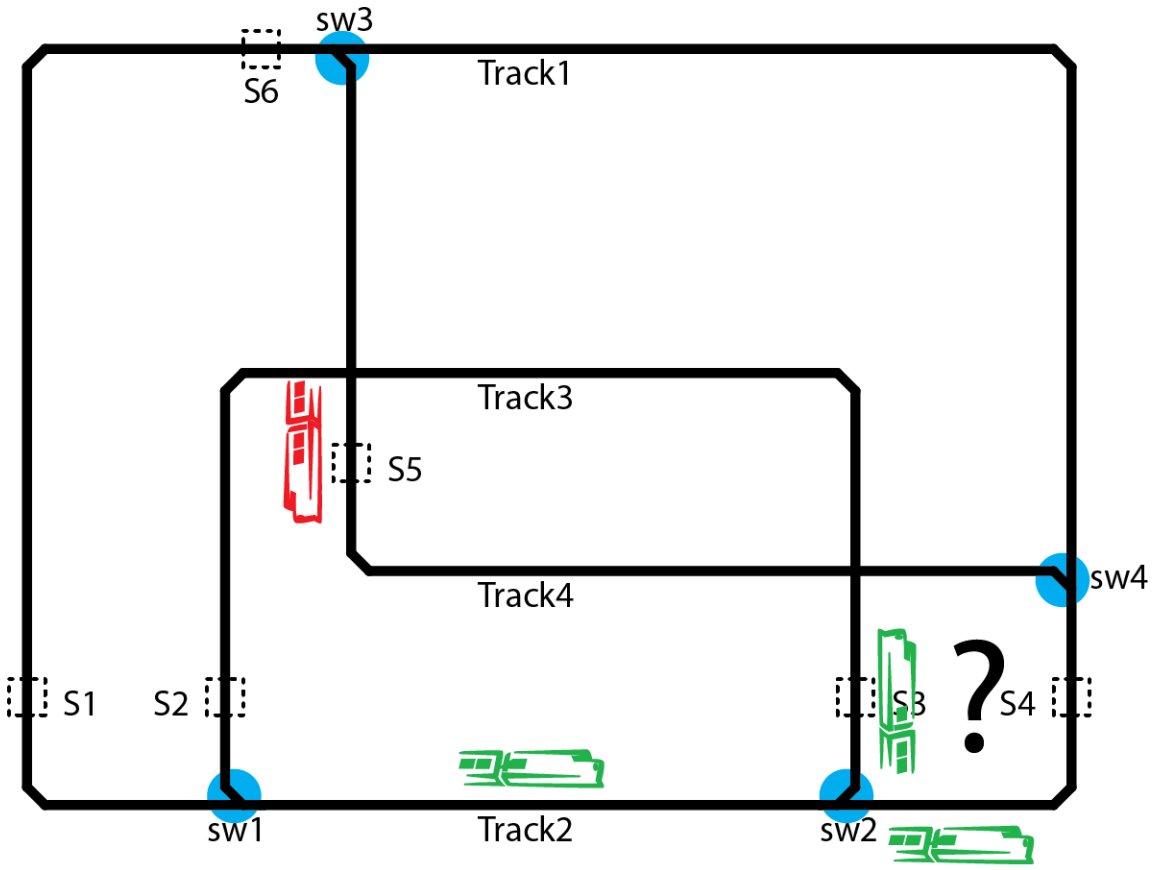
Train A always stops on Track4



When it reaches Sensor 5, Train A ALWAYS stops.

When this happens, Train B will be on Track 2, or it will possibly be stopped at S3 (see previous slide).

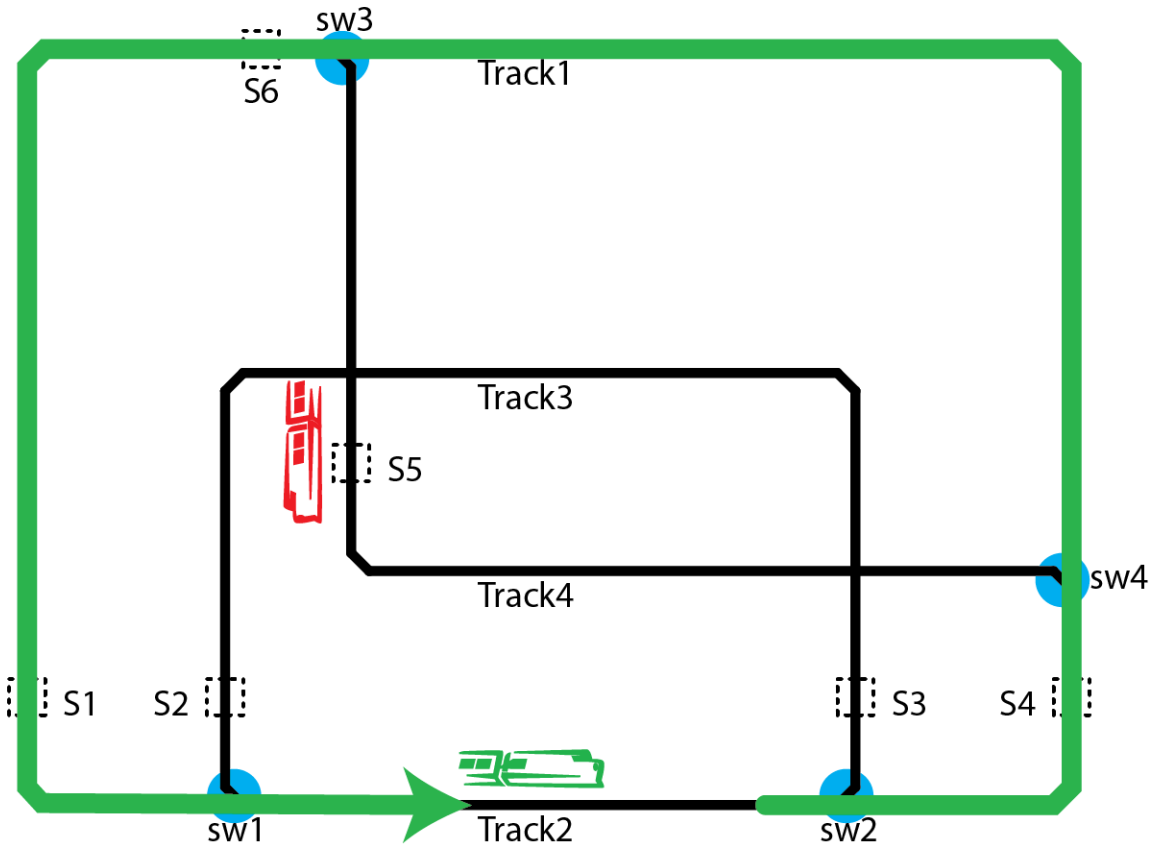
Train B behavior changes while A stopped



The instant Train A stops at S5, switch 2 allows Train B to go straight, leaving the Track2/Track3 inner loop.

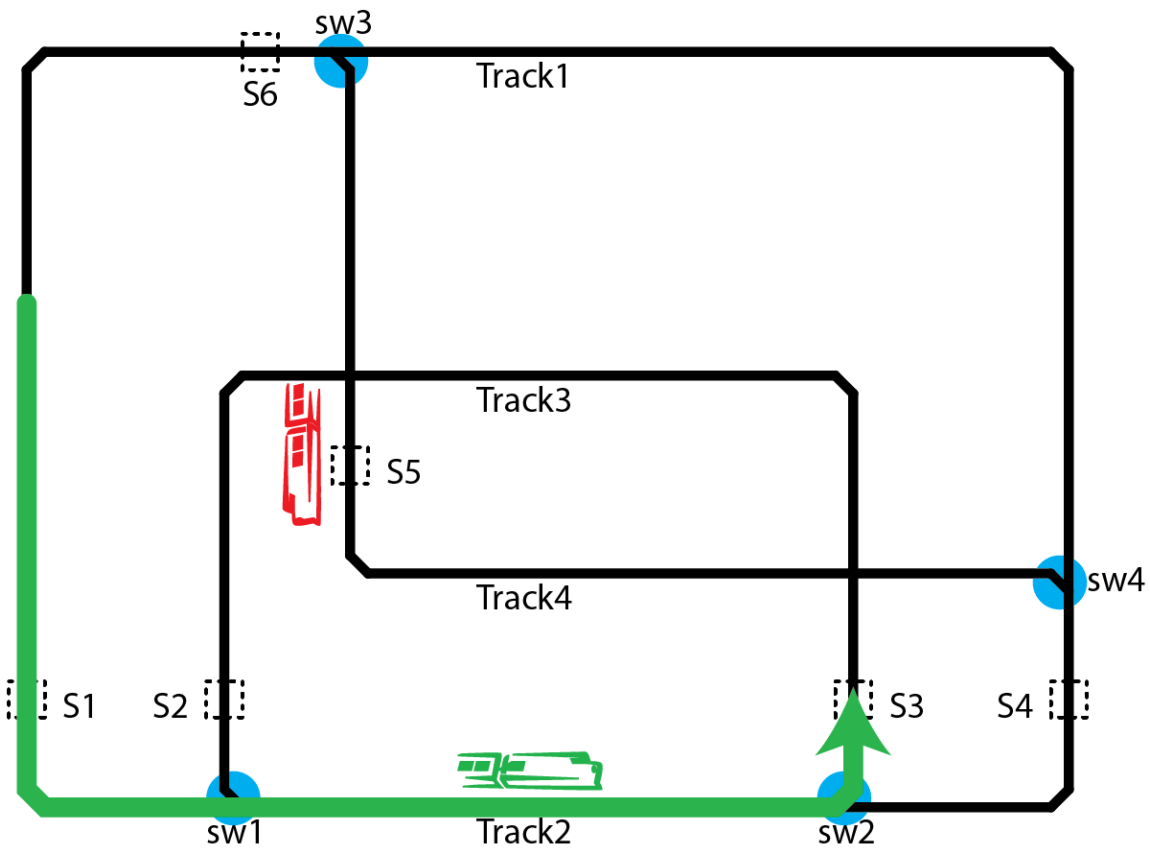
But Train B may have already passed switch 2 (either stopped at S3 or about to stop at S3).

Train B takes a spin on the outer loop



If Train B goes straight, it makes exactly one loop, returning to Track 2. Train A remains stopped.

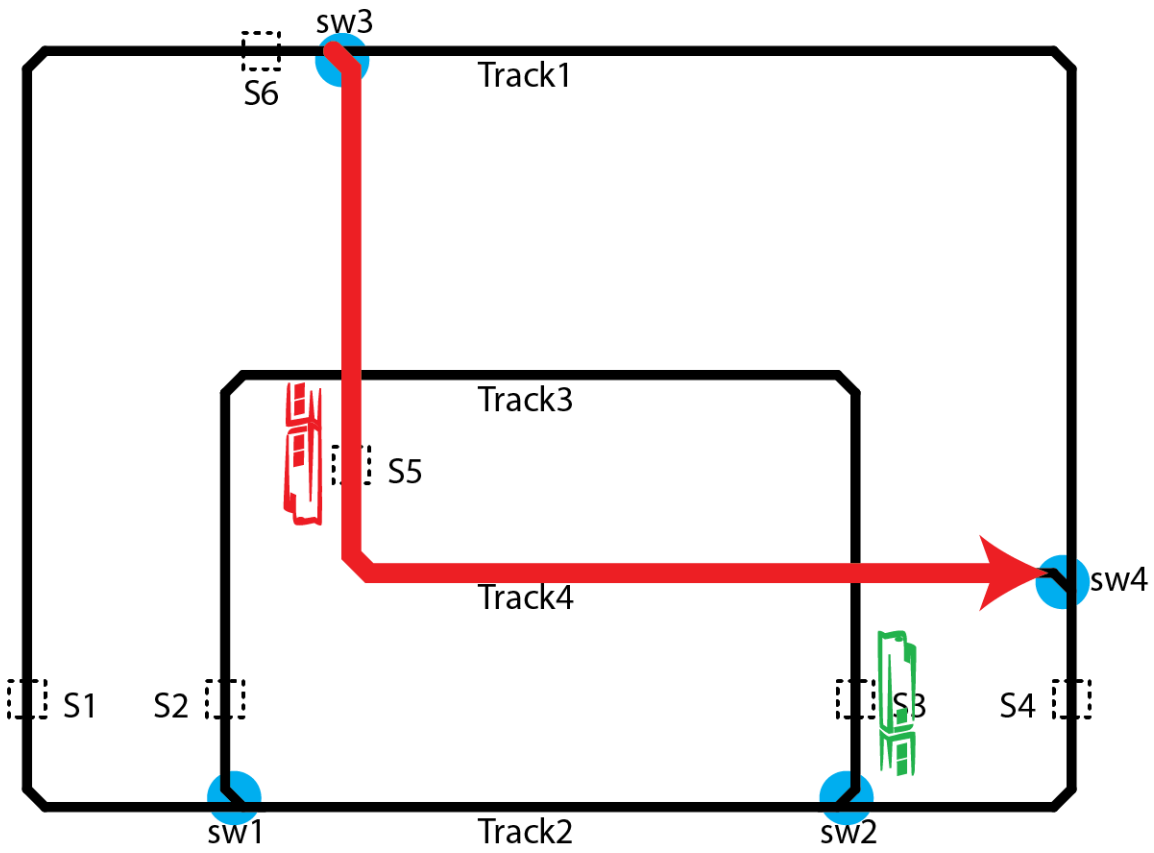
Train B returns to inner loop & stops



By the time Train B passes sensor 1 (or sooner), switch 2 switches back to the inner loop, and Train B stops at sensor 3.

This is the same stopping condition mentioned earlier (see slide titled “Train B cannot pass S3 when unsafe”), except that it occurs after the “extra” loop around Track1.

Train A continues



Once B is stopped, A continues. This can occur 3 ways:

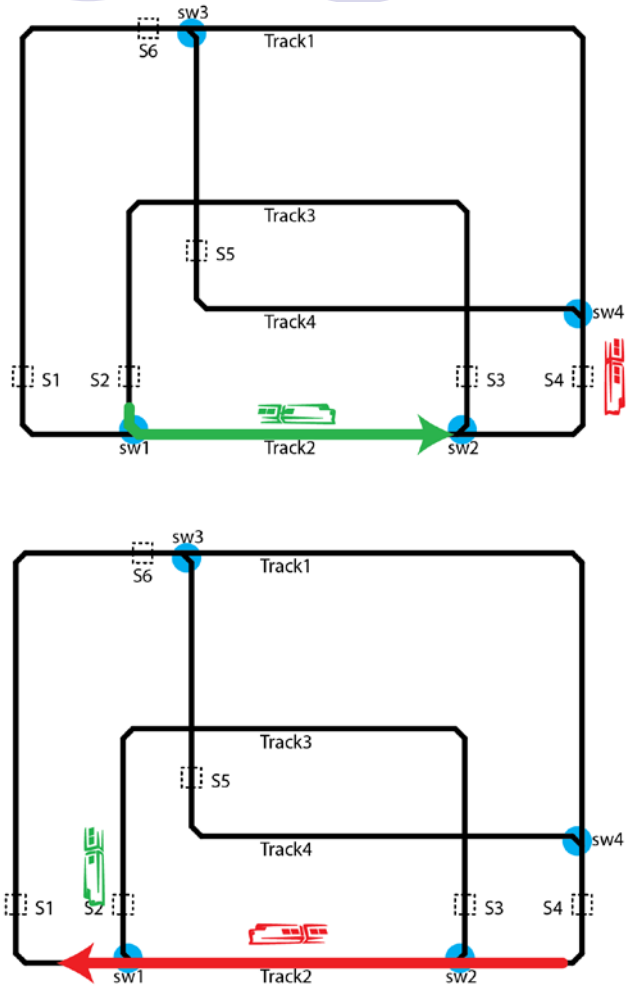
1) B has just taken the Track1 loop and stopped (previous slide).

2) B missed the chance to take the Track 1 loop because it was already past switch 2, so it had to stop at S3.

3) B was already stopped and waiting for Track 4 to clear when A stopped at S5.

In all cases, A continues first. Once A clears Sensor 4, B continues.

Track 2 is arbitrated as in example



As in our example problem, Track 2 is part of the “normal” path for both trains.

Train A may have to stop at S4 if Train B is in Track 2.

And Train B may have to stop at S2 if Train A is in Track 2.

Clarifications

- What if Train A shows up at sensor 6 just as Train B shows up at sensor 3?
  - ALWAYS let trains keep moving, if possible. In this case, that means A takes Track 1, while B takes Track 3 as usual.
- What if Train A shows up at sensor 5 just as Train B shows up at sensor 3?
  - Train A must always stop at sensor 5. Train B must always stop when Train A is on Track 4. So both trains stop for one state (not long enough for you to notice). Then the other rules explain what happens (Train A always continues after Train B is stopped at sensor 3.)
- Make similar considerations for other “ties” that may arise
- What happens if switch 2 changes as Train B is going over it?
  - As far as we know, nothing bad. But if it causes a crash, it’s OK. (It’s not a failure of your controller.)
- There is no “target” for number of states
  - Describe unique, necessary states on your worksheets, and you should generate an efficient solution