Theorem (Immerman-Szelepcsényi): For reasonable $s(n) \geq \log n$,

$$NSPACE(s(n))=co\text{-}NSPACE(s(n)).$$

Let M be a nondeterministic machine using s(n) space. We will create a nondeterministic machine N such that for all inputs x, N(x) accepts if and only if M(x) rejects.

Fix an input x and let $s=s(|x|)$. The total number of configurations of M(x) can be at most c^s for some constant c. Let $t=c^s$. We can also bound the running time of M(x) by t because any computation path of length more than t must repeat a configuration and thus could be shortened.

Let I be the initial configuration of M(x). Let m be the number of possible configurations reachable from I on some nondeterministic path. Suppose we knew the value of m. We now show how N(x) can correctly determine that M(x) does not accept.

```
Let r=0
For all nonaccepting configurations C of M(x)
  Try to guess a computation path from I to C
  If found let r=r+1
If r=m then accept o.w. reject
```

If M(x) accepts then there is some accepting configuration reachable from I so there must be less than m non-accepting configurations reachable from I so N(x) cannot accept. If M(x) rejects then there is no accepting configurations reachable from I so N(x) on some nondeterministic path will find all m nonaccepting paths and accept. The total space is at most O(s) since we are looking only at one configuration at a time.

Of course we cannot assume that we know m. To get m we use an idea called *inductive counting*. Let m_i be the number of configurations reachable from I in at most i steps. We have $m_0=1$ and $m_t=m$. We show how to compute m_{i+1} from m_i. Then starting at m_0 we compute m_1 then m_2 all the way up to $m_t=m$ and then run the algorithm above.

Here is the algorithm to nondeterministically compute m_{i+1} from m_i.

```
Let mi+1=0
For all configurations C
  Let b=0, r=0
  For all configurations D
    Guess a path from I to D in at most i steps
    If found
      Let r=r+1
      If D=C or D goes to C in 1 step
        Let b=1
  If r<mi halt and reject
  Let mi+1=mi+1+b
```

The test that $r<m_i$ guarantees that we have looked at all of the configurations D reachable from I in i steps. If we pass the test each time then we will have correctly computed b to be equal to 1 if C is reachable from I in at most i+1 steps and b equals 0 otherwise.

We are only remembering a constant number of configurations and variables so again the space is bounded by O(s). Since we only need to remember m_i to get m_{i+1} we can run the whole algorithm in space O(s).