A specifikáció tétele szerint elég belátni hogy Q => lf(S,R)

!! lf == R ahol behelyettesítjük S dolgait

**Szekvencia** tehát elég ehelyett belátni 2 másik állítást:

!! usually Q, R, Q’ meg a szekvencia sorai is meg vannak adva so pretty easy

1. Q => lf(szekvencia első sora, Q’)
2. Q’ => lf(szekvencia második része, R)

**Ciklus** tehát elég ehelyett belátni 5 másik állítást:

!! usually Q, P, R és t meg van adva és PI = a ciklusfeltétel

1. Q => P
2. P ^ ¬PI => R
3. P => PI v ¬PI ((usually nem is kell kiírni P-t és elég könnyű belátni hogy PI v ¬PI))
4. P ^ PI => t > 0
5. P ^ PI ^ t = t0 => lf(S0, P ^ t < t0)

**Elágazás** tehát elég ehelyett belátni 3 másik állítást:

!! PI1..PIn a feltételek balról jobbra

1. Q => összeéselve az összes feltételre (PI v ¬PI)
2. Q => összevagyolva az összes feltétel
3. sorra vesszük a feltételeket és hozzá éseljük őket a Qhoz => lf(Si, R)

**Várakoztató** tehát elég ehelyett belátni 2 másik állítást:

!! β az await feltétele

1. Q => β v ¬β ((usually nem is kell kiírni Q-t és elég könnyű belátni hogy β v ¬β))
2. Q ^ β => lf(awaiten belüli rész, R)

**Párhuzamos blokk** tehát elég ehelyett belátni 5 másik állítást:

1. Q => Q1 ^ Q2 ((a blokkok első sorai))
2. R1 ^ R2 ((a blokkok utolsó sorai)) => R
3. mindkét komponensre egy Qi => lf(Si, Ri) ((Si minden ami középen van))
4. keresünk kritikus utasításokat ((értékadást)) Si-ben ez lesz az Ui

HA van akkor mindegyikre végre kell hajtani 2 további állítást pl U1 akkor:

* 1. Q1 ^ R2 => lf(U1, R2)
  2. Q1 ^ Q2 => lf(U1, Q2)

1. D(T) = van-e külső await VAGY D1(T)

!! ha nincs await az egyik blokkban annak a D(Si) mindig hamis

ahol D1(T) = (D(S1) ^ post(S2)) v (post(S1)^D(S2)) v (D(S1) ^ D(S2))

ahol D(Si) = pre(await) ^ ¬β

ahol post(Si) = a blokk utolsó sora