SmartCard protokoll formális verifikációja

Machine Checked Formal Proof of a Scheduling Protocol for Smartcard Personalization

Leonard Lensink, Sjaak Smetsers, and Marko van Eekelen

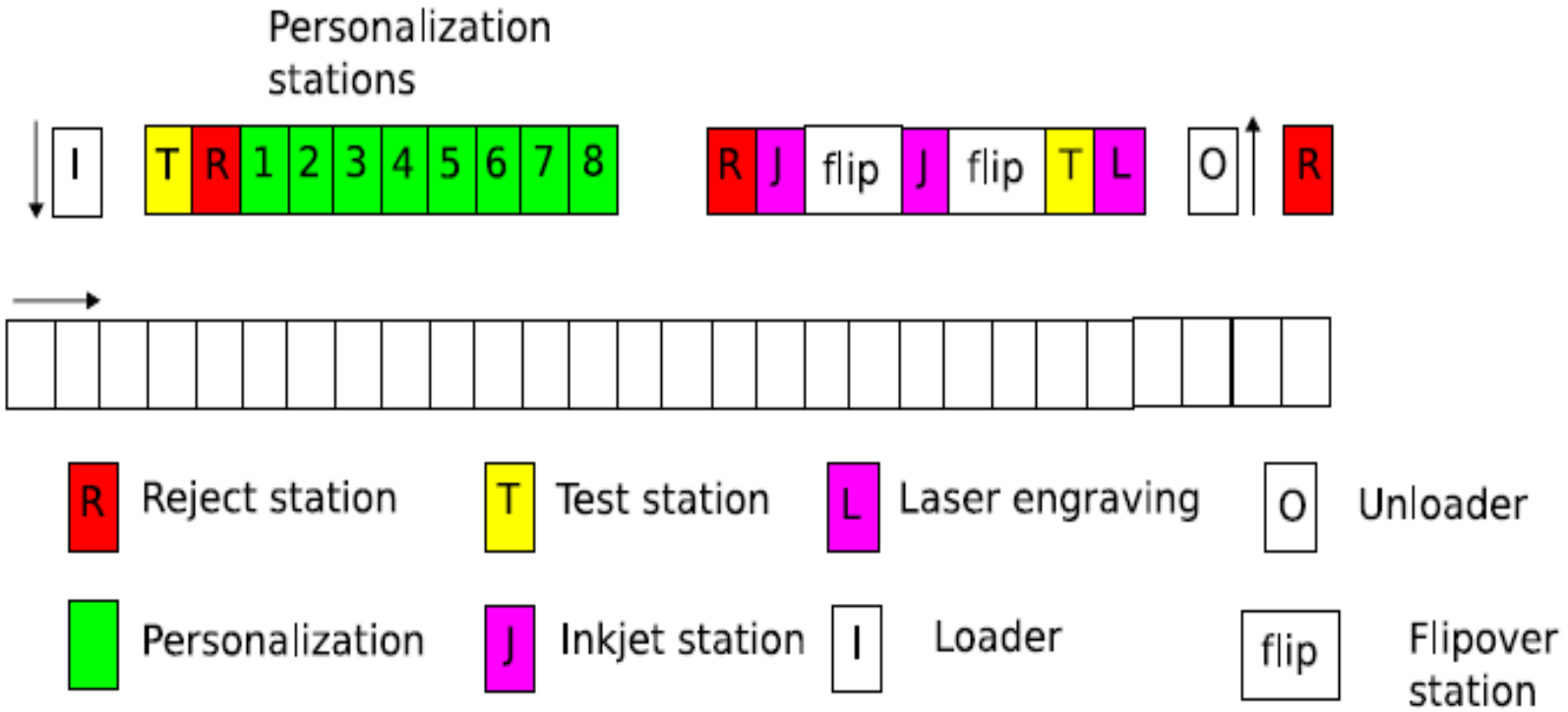
Cserpán Dorottya

Bevezetés, miért érdekes

- SmartCard protokol biztonságának és optimalitásának bizonyítása PVS segítségével
- PVS (Prototype Verification System) egy specifikációs és validációs rendszer
  - Magasabb rendű logikára épül

Személyreszabás folyamata

- Üres kártyák programozása
- Nyomtatás
- Tesztelés

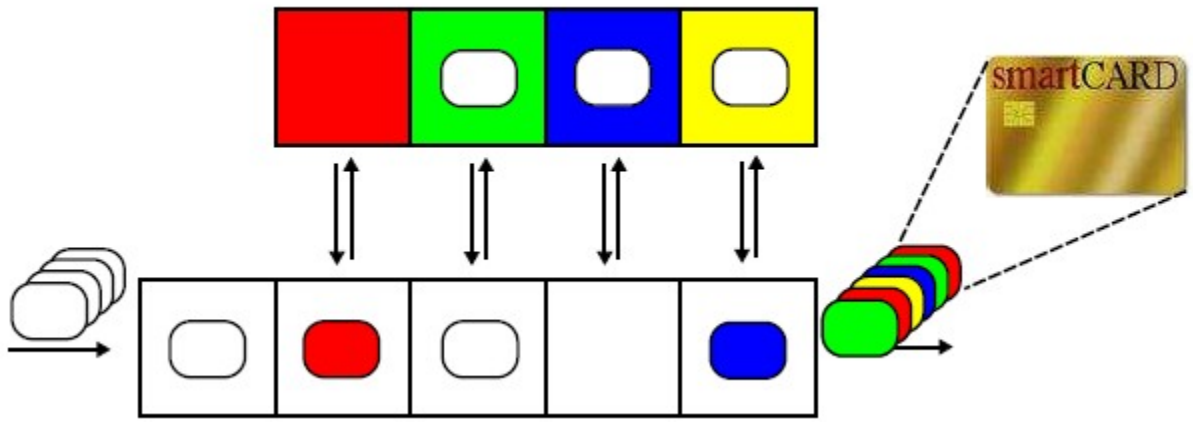


Elvárások

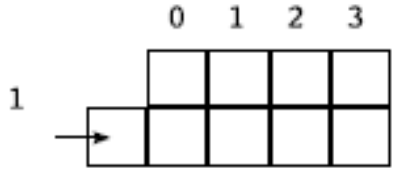
- A kártyák kimeneti sorrendje előre meghatározott
- Optimális működés
- Hibás kártyák eltávolításának lehetősége
- A rendszer konfigurálható és moduláris

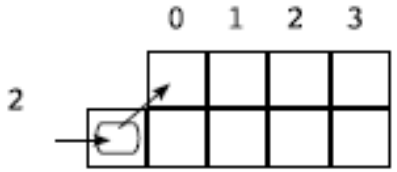
A személyreszabó rendszer PVS modellje

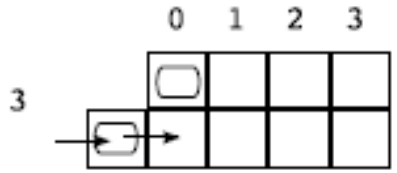
- M személyreszabó állomás
- Ezek képesek kártyákat felvenni és letenni
- A továbbító szalag szinkronizált

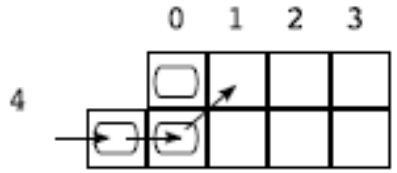


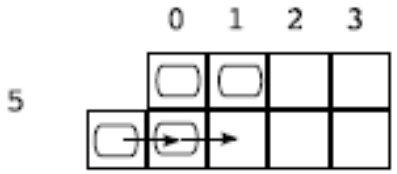
t


0 1 2 3
1

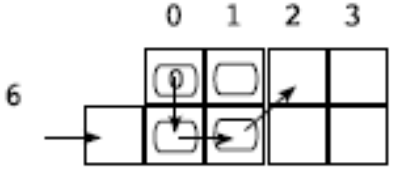

0 1 2 3
2

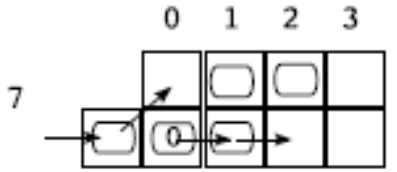

0 1 2 3
3

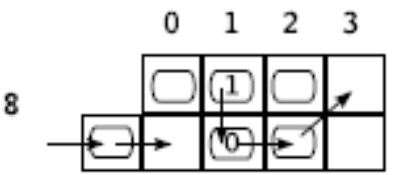

0 1 2 3
4

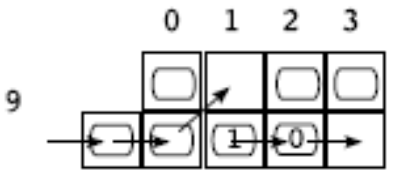

0 1 2 3
5

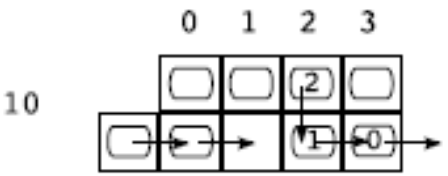
t


0 1 2 3
6
0

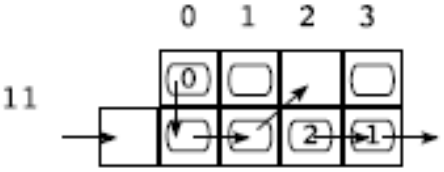

0 1 2 3
7
0

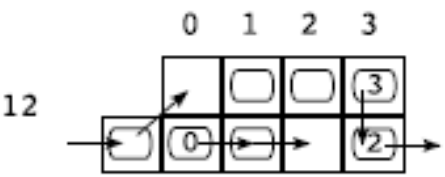

0 1 2 3
8
1
0

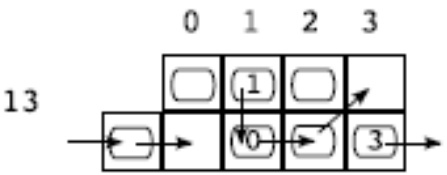

0 1 2 3
9
1
0

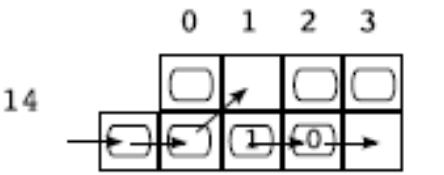

0 1 2 3
10
2
1
0

t


0 1 2 3
11
0
2
1


0 1 2 3
12
3
0
2


0 1 2 3
13
1
0
3


0 1 2 3
14
1
0

Cél

A PVS modell segítségével verifikálni , hogy az időzítési algoritmusra teljesülnek a következők:

- A személyreszabott kártyák az állomások sorrendjében hagyják el a gépet
- Optimális működés

A szalag

```
slot : DATATYPE

BEGIN

empty : empty?

new_card : new_card?

personalization (number : personalization_nr)
: personalization?

END slot
```

Az állomások

Időzítő (hol tart a folyamat?)

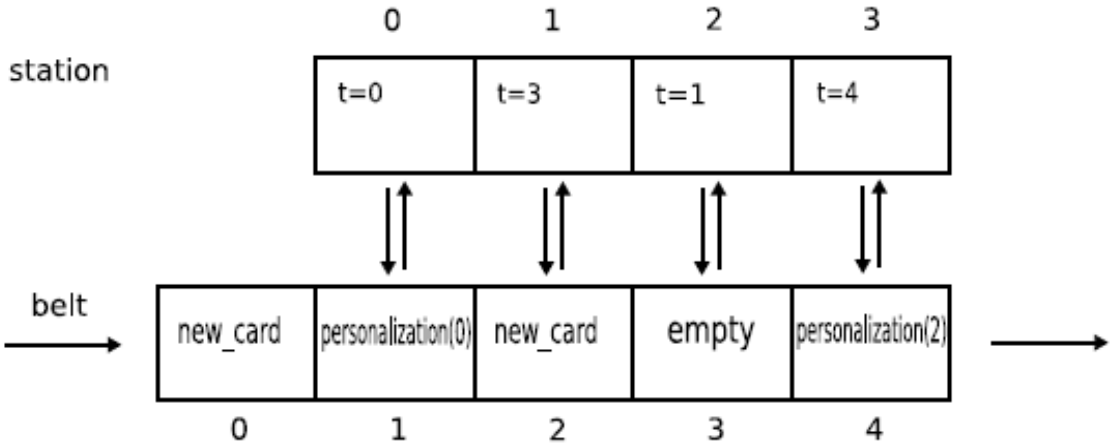
```
timer : TYPE = nat
```

állomások

```
stationposition : TYPE = below(M)
```

A gép

- M állomás , 1 + M lyuk
- Időzítő szinkronizál




```
machine_state : TYPE =
[# stations : [stationposition -> timer] ,
belt : [beltposition -> slot] ,
global_timer : global_timer
#]
```

- f_next függvény lépteti a rendszert a következő állapotba

```
f_next(ps:machine_state) : machine_state =
(# stations := f_operate_station(ps),
belt := f_operate_belt(ps),
global_timer := global_timer(ps)+1
#)
```

A gép működésének legalapvetőbb szituációi:

- Üres állomás, új kártya elérhető az előző szalagpozíción
- Az állmás időzítője jelzi, hogy elkészült a személyreszabás és van egy üres hely a kártya számára szalagon
- Ha a fentiek közül egyik sem teljesül, a szalagot és az időzítőt eggyel léptejük

Ha a perszonalizáció ideje az állomások számánál:

- Nagyobb: a safety feltétel nem teljesül, üres kártyák is kikerülnek
- Kisebb: nincs elég üres hely a szalagon, M ütem múlva érkezik meg

Az időzítő

- M új kártyát tesz a szalagra, majd egy ütemnyi szünet után ismétlés

A modell validációja

- Hitelesen reprezentálja a renszer működését?
  - Vizuális szemléltetés
  - Különböző méretek

új kártya

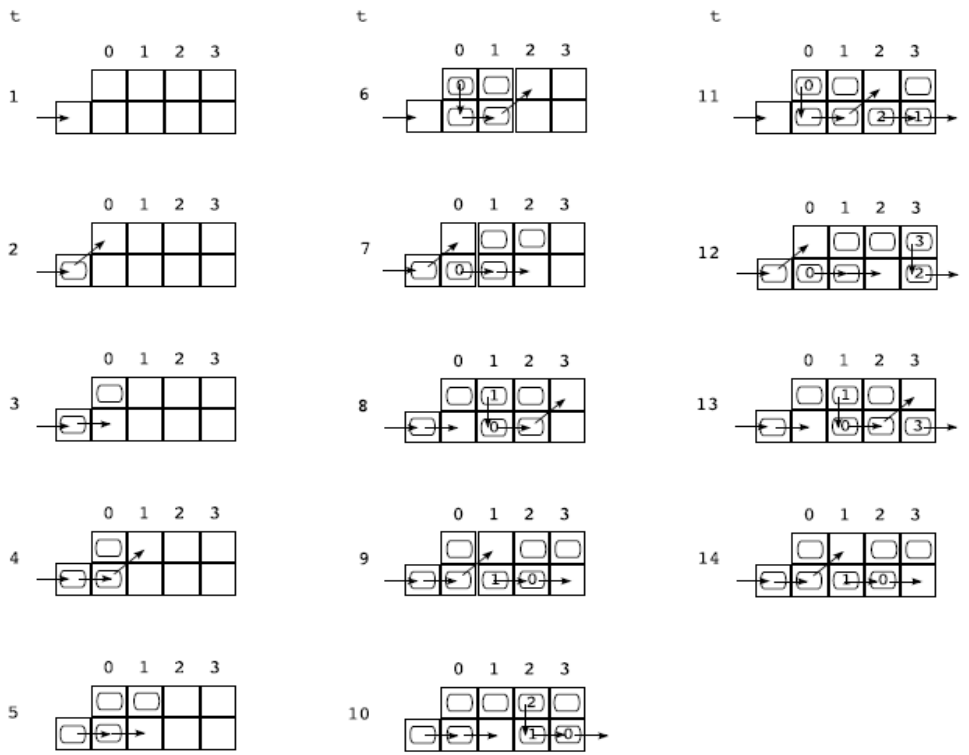
* perszonalizáló állomás

^ üres állomás

! elkészült perszonalizáció

n perszonalizált kártyák

```
<PVSio> simulation(14);
1    ^ ^ ^ ^
  __________
2    ^ ^ ^ ^
  _#________
3    * ^ ^ ^
  _#________
4    * ^ ^ ^
  _#_#______
5    * * ^ ^
  _#_#______
6    ! * ^ ^
  ___#_#____
7    ^ * * ^
  _#_0_#____
8    * ! * ^
  _#___0_#__
9    * ^ * *
  _#_#_1_0__
10   * * ! *
  _#_#___1_0
11   ! * ^ *
  ___#_#_2_1
12   ^ * * !
  _#_0_#___2
13   * ! * ^
  _#___0_#_3
14   * ^ * *
  _#_#_1_0__
```



A szimuláció az elvárt módon viselkedik!

```
<PVSio> simulation(14);
1     ^ ^ ^ ^
  ____________

2     ^ ^ ^ ^
  _#__________

3     * ^ ^ ^
  _#__________

4     * ^ ^ ^
  _#_#________

5     * * ^ ^
  _#_#________

6     ! * ^ ^
  ___#_#______

7     ^ * * ^
  _#_0_#______

8     * ! * ^
  _#___0_#____

9     * ^ * *
  _#_#_1_0____

10    * * ! *
  _#_#___1_0

11    ! * ^ *
  ___#_#_2_1

12    ^ * * !
  _#_0_#___2

13    * ! * ^
  _#___0_#_3

14    * ^ * *
  _#_#_1_0____
```

újkártya

* perszonalizáló állomás

^ üres állomás

! elkészült perszonalizáció

n perszonalizált kártyák

Elvárások

- Biztonság
  - Csak perszonalizált kártyák vagy üres helyek lehetnek a szalag utolsó helyén
  - Az n-edik elkészült kártya a végleges heyére került a szalagon a következő kártyának az n+1-es kártyának kell lennie vagy üres helynek
  - Tudjuk, hogy csak egy üres hely van, ezért a kártyák kijöveteli sorrendje és a global_timer között lineáris kapcsolat van (c)

$$\begin{aligned} &\texttt{empty?(belt(ps)(M))} \\ \vee\ &(\texttt{personalization?(belt(ps)(M))} \wedge \\ &\quad \exists\ \texttt{c: mod(global\_timer(ps)+c,1+M)} = \texttt{number(belt(ps)(M))}) \end{aligned}$$

- Optimalitás
  - Egy üres állomás képes egy kártyát felvenni
  - Ha elkészült a perszonalizáció, van üres hely a szalagon a kártya számára

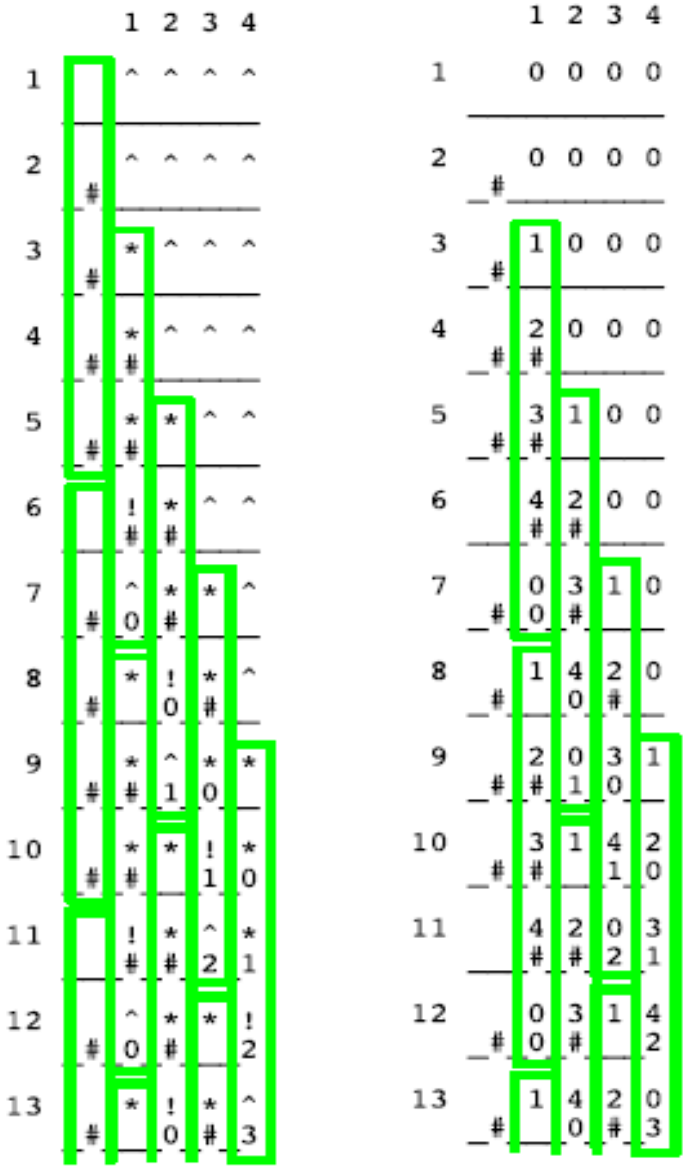
$$\forall\ \texttt{pos}: \exists\ \texttt{ps'}: \texttt{global\_timer}(\texttt{ps'}) = \texttt{global\_timer}(\texttt{ps})+1 \wedge$$
$$(\texttt{empty?}(\texttt{station}(\texttt{ps})(\texttt{pos})) \Rightarrow \texttt{start?}(\texttt{station}(\texttt{ps'})(\texttt{pos}))) \wedge$$
$$(\texttt{done?}(\texttt{station}(\texttt{ps})(\texttt{pos})) \Rightarrow \texttt{empty?}(\texttt{station}(\texttt{ps'})(\texttt{pos})))$$

- Eredménytelen, ennél erősebb invariáns bevezetése szükséges

- A ciklikus fázis két ütemenként halad előre egy pozíciót

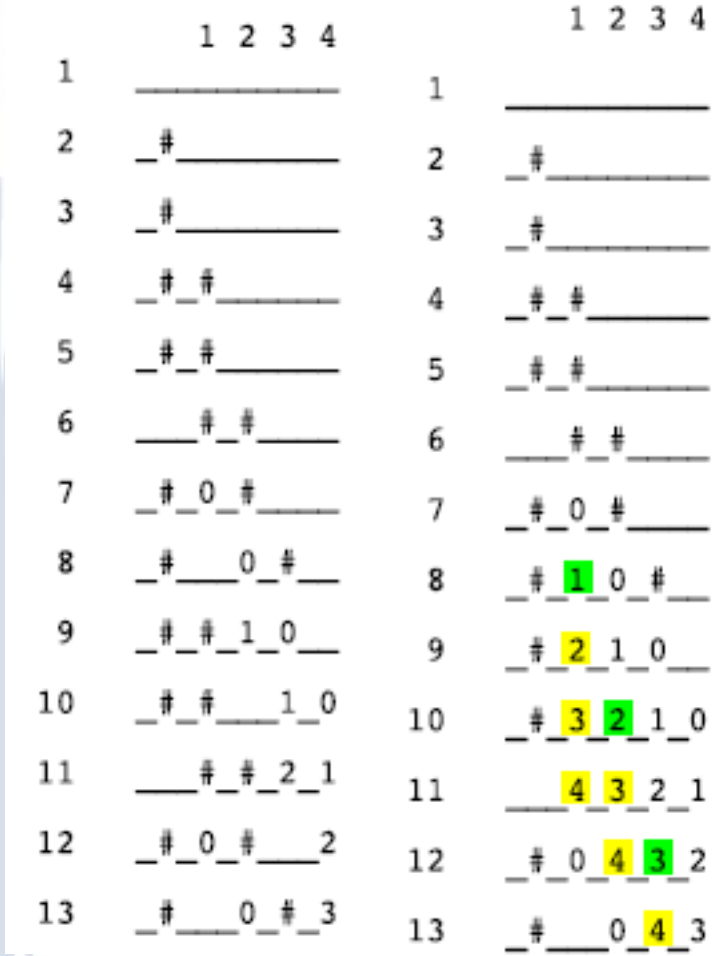
```
p_invariant(ps:machine_state) : bool =
 ∀ bpos : IF 2*bpos+1 ≥ global_timer(ps)
          THEN p_init(ps)(bpos)
          ELSE p_stable(ps)(bpos)
          ENDIF
```

- Kezdeti invariáns: az állomás és a hozzá tartozó hely is üres a szalagon
- Az állomások időzítője arányosan változik a globális idővel



A ciklikus invariáns terjedése és az állomások állapota

A global_timer és a szalag tartalma közti kapcsolat




```
belt = mod(global_timer(ps)-bpos-1,1+M) ∧
IF belt = bpos THEN empty
ELSIF belt > bpos THEN new_card
ELSE personalization(number(belt))
ENDIF
```

- A korábbi eredmények felhasználásával a rendszer invariánsa:

```
p_stable(ps:machine_state)(pos:beltposition) : bool =
(pos ≤ M-1 ⇒ mod(global_timer(ps)-2*(pos+1),1+M) = station(ps)(pos))
 ∧
LET timer = mod(global_timer(ps)-2*pos-1,1+M), belt = belt(ps)(pos) IN
IF timer = 0
THEN  empty?(belt)
ELSIF timer < 1+M-pos
THEN new_card?(belt)
ELSE personalization?(belt) ∧ number(belt) = timer-1-M+pos
ENDIF
```

Biztonság és optimalitás

- Biztonság

$$\begin{aligned} &\mathtt{empty?(belt(ps)(M))} \\ \vee\ &(\mathtt{personalization?(belt(ps)(M))} \wedge \\ &\quad \mathtt{mod(global\_timer(ps),1+M)} = \mathtt{number(belt(ps)(M))}) \end{aligned}$$

- Optimalitás: 1+ M egymást követő kártyát nem tud személyreszabni a rendszer, de M db-ot igen!

Köszönöm a figyelmet!