TPPmark 2011

Uniform Candy Distribution

Yoshinao Isobe

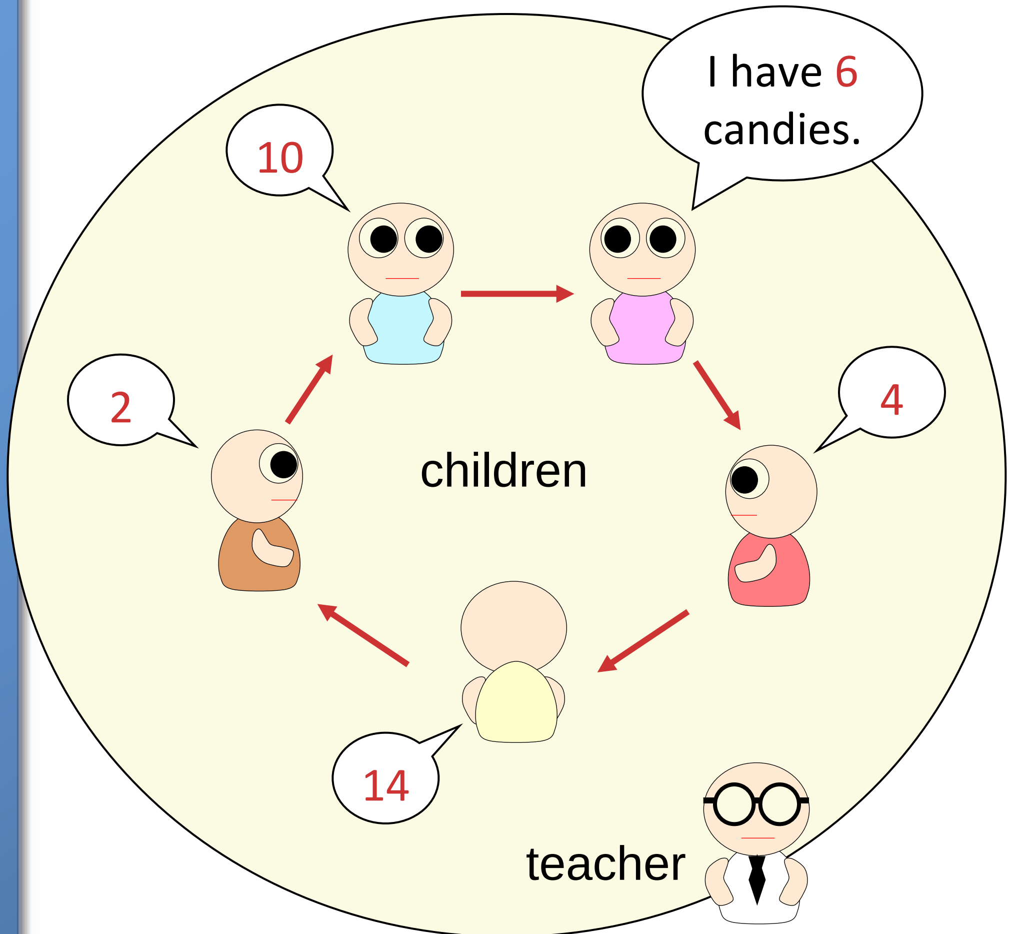
Information Technology Research Institute

AIST, Japan

TPP 2011 (18 November 2011)

TPPmark 2011

Uniform Candy Distribution puzzle: a classical example of self-organizing systems



(1) There are N children sitting in a circle.

(2) Each child has an even number of candies.

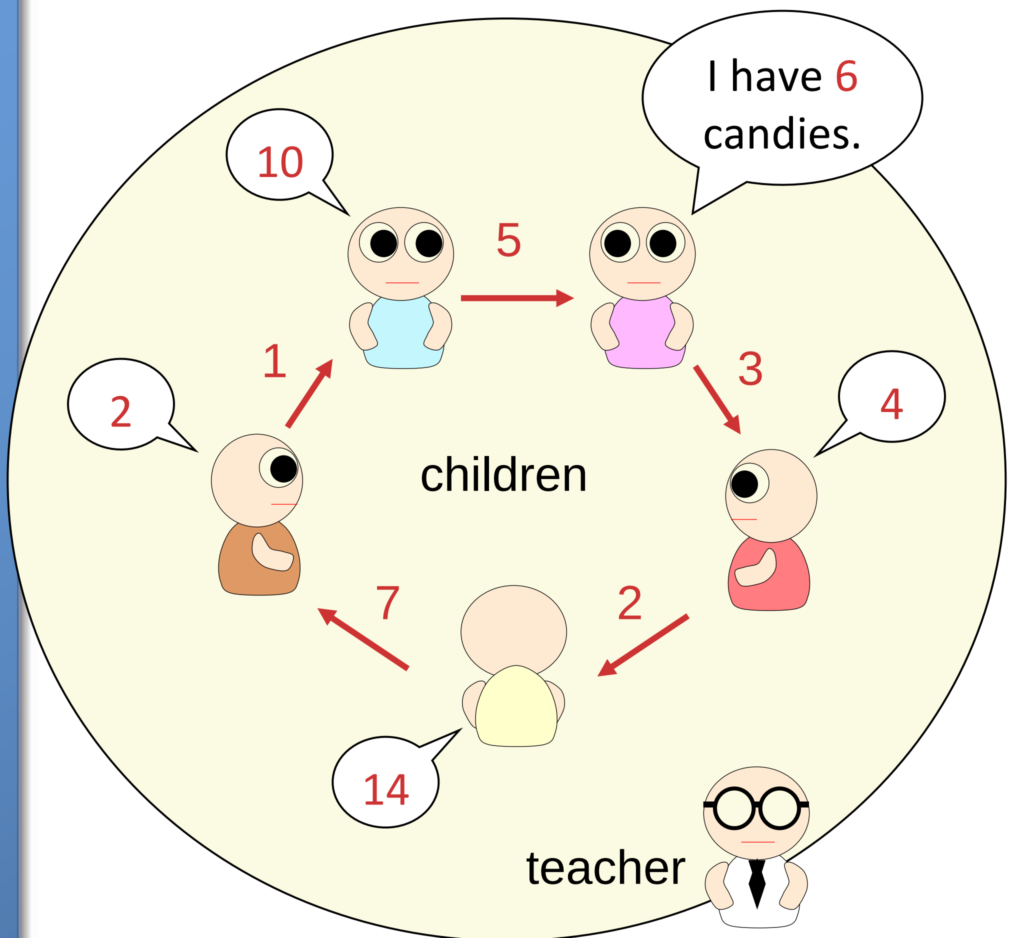
(3) Every child passes half of their candies to the child on their left.

(4) Any child who ends up with an odd number of candies is given another candy by the teacher.

(5) Repeat (3) and (4).

TPPmark 2011

Uniform Candy Distribution puzzle: a classical example of self-organizing systems



(1) There are N children sitting in a circle.

(2) Each child has an even number of candies.

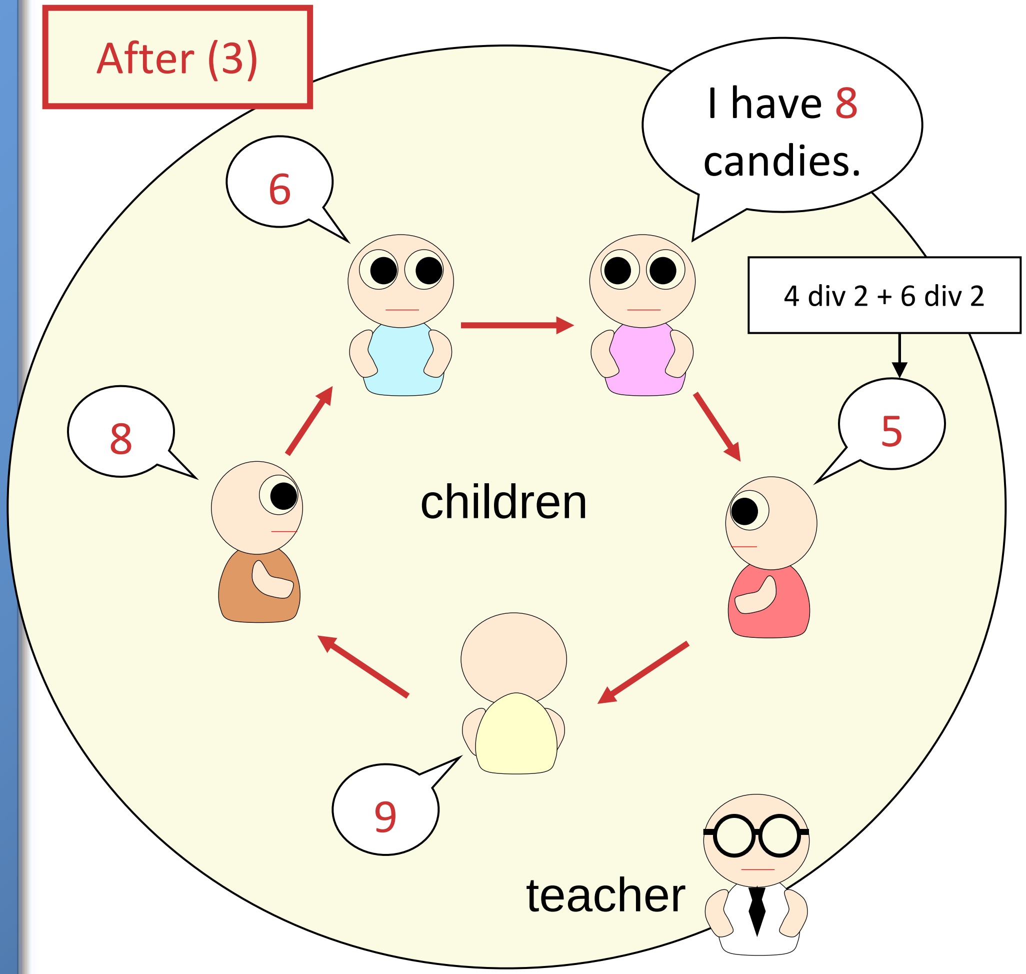
(3) Every child passes half of their candies to the child on their left.

(4) Any child who ends up with an odd number of candies is given another candy by the teacher.

(5) Repeat (3) and (4).

TPPmark 2011

Uniform Candy Distribution puzzle: a classical example of self-organizing systems



(1) There are N children sitting in a circle.

(2) Each child has an even number of candies.

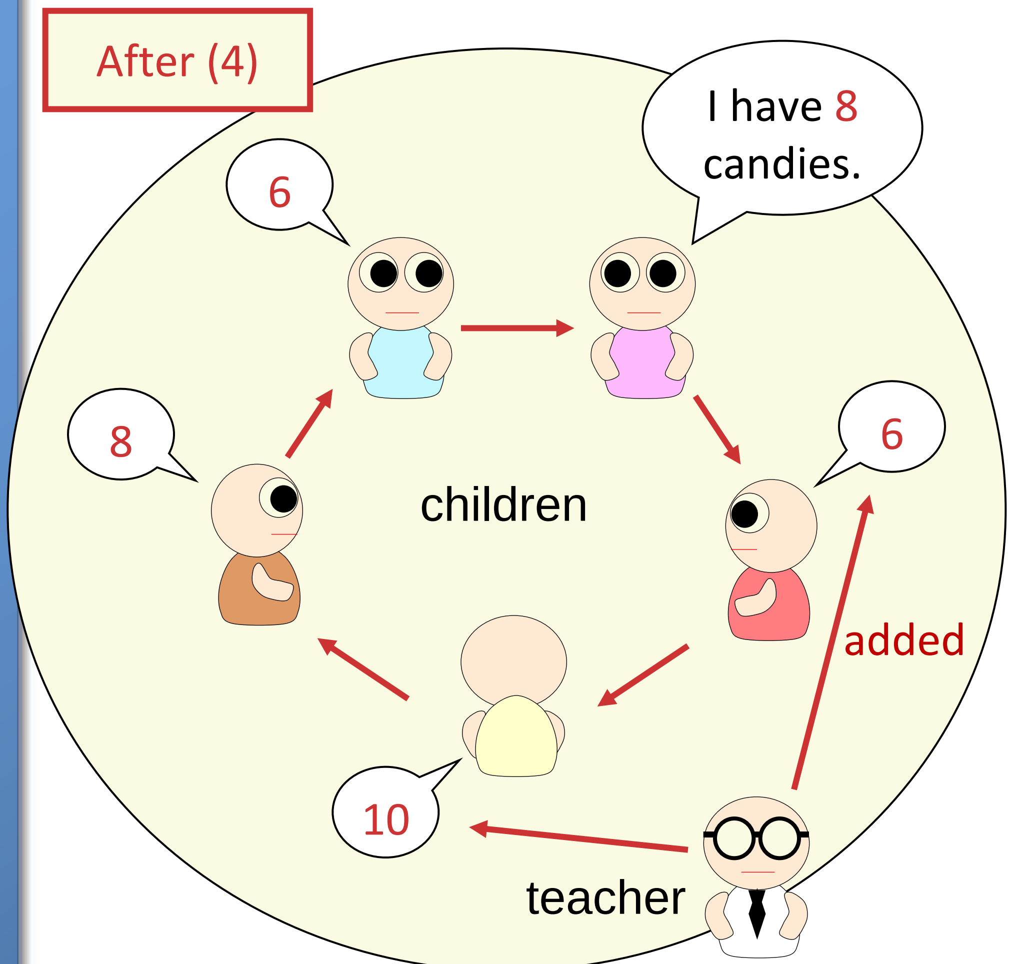
(3) Every child passes half of their candies to the child on their left.

(4) Any child who ends up with an odd number of candies is given another candy by the teacher.

(5) Repeat (3) and (4).

TPPmark 2011

Uniform Candy Distribution puzzle: a classical example of self-organizing systems



(1) There are N children sitting in a circle.

(2) Each child has an even number of candies.

(3) Every child passes half of their candies to the child on their left.

(4) Any child who ends up with an odd number of candies is given another candy by the teacher.

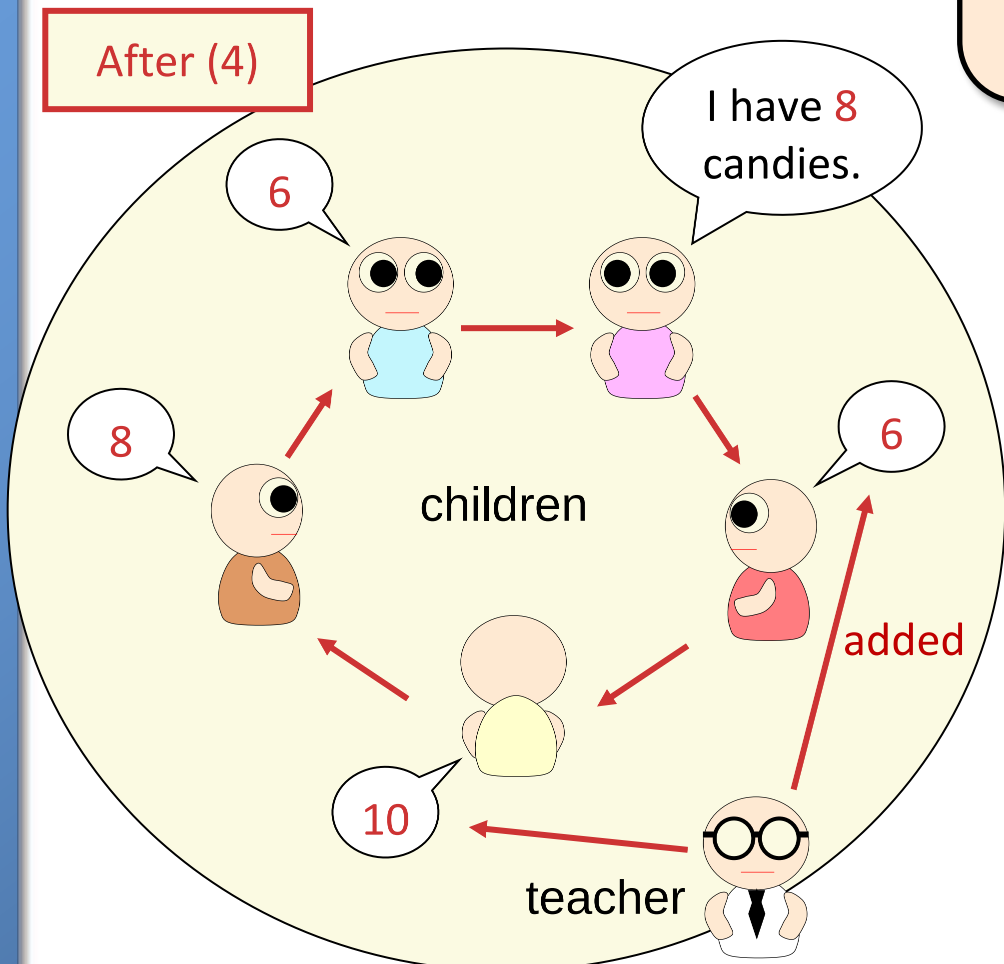
(5) Repeat (3) and (4).

TPPmark 2011

Uniform Candy Distribution puzzle: a classical

Question:

Will all the children eventually have the same number of candies?



(1) There are N children sitting in a circle.

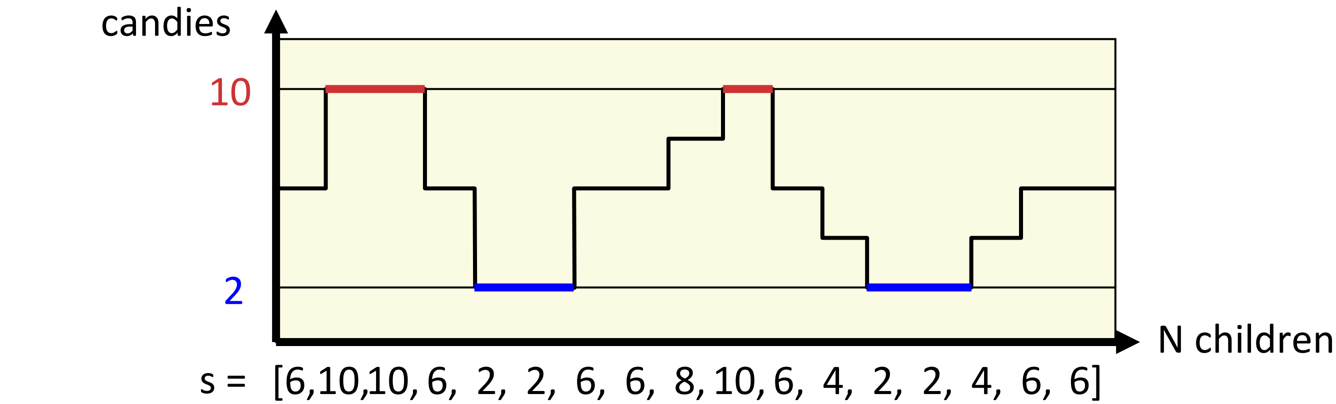
(2) Each child has an even number of candies.

(3) Every child passes half of their candies to the child on their left.

(4) Any child who ends up with an odd number of candies is given another candy by the teacher.

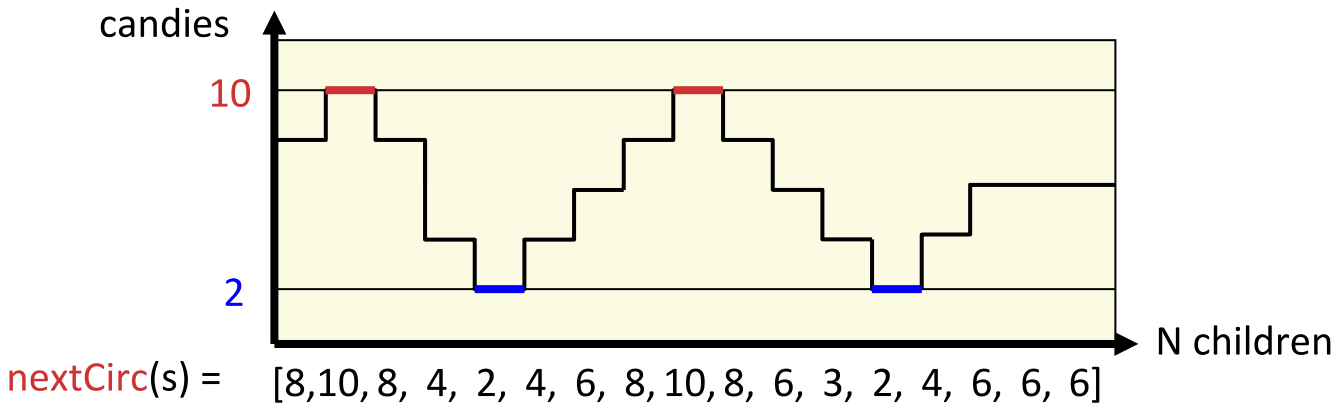
(5) Repeat (3) and (4).

Lemmas



max(k)=10
min(k)=2
num(2,k)=4

⇓ after one cycle (pass and addition)

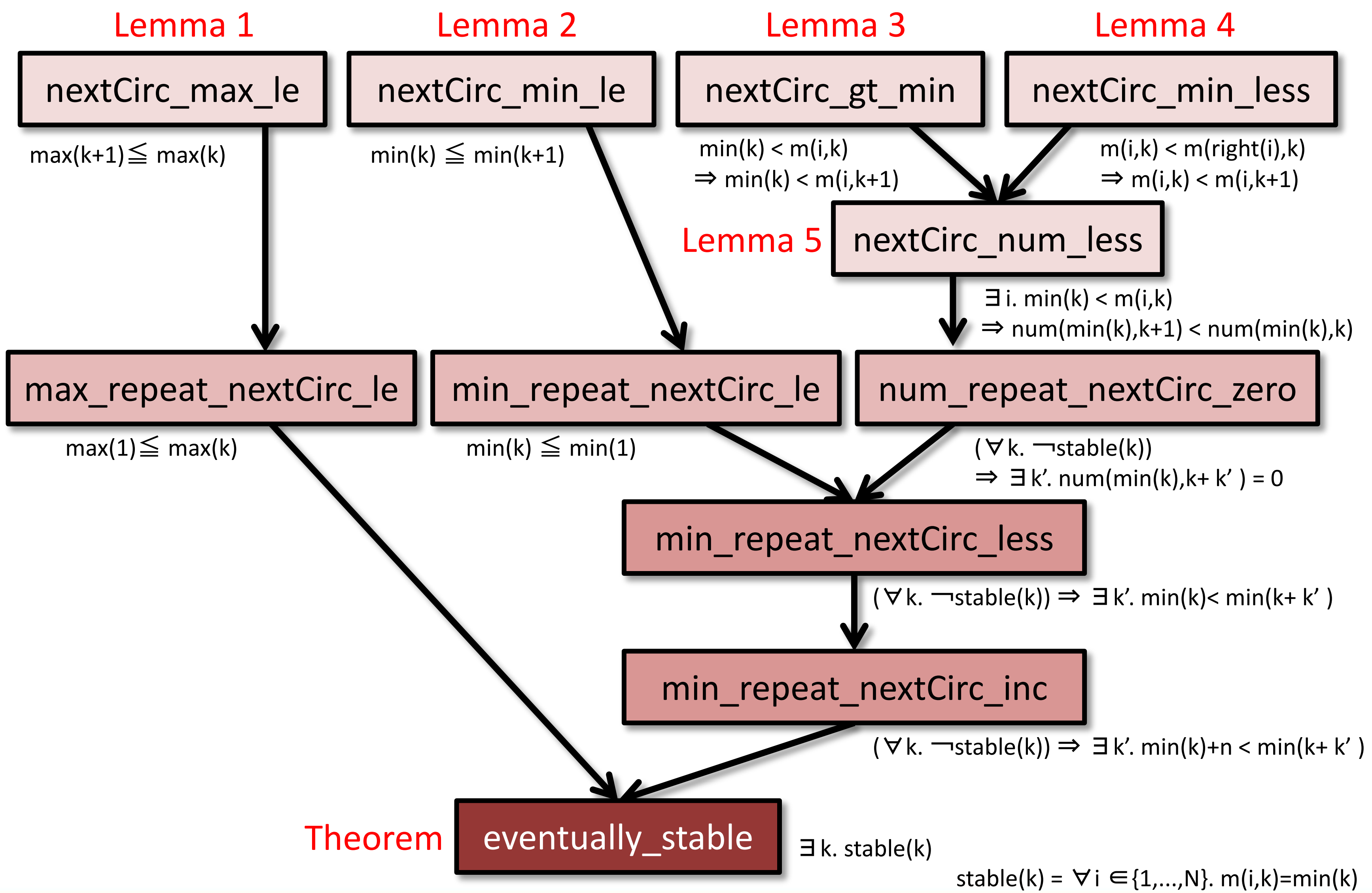


max(k+1)=10
min(k+1)=2
num(2,k+1)=2

(1) The max never increases and the min never decreases.

(2) The number of children who has the min number of candies strictly decreases.

Lemmas



Lemma 1
nextCirc_max_le
max(k+1)≦ max(k)
Lemma 2
nextCirc_min_le
min(k) ≦ min(k+1)
Lemma 3
nextCirc_gt_min
min(k) < m(i,k)
⇒ min(k) < m(i,k+1)
Lemma 4
nextCirc_min_less
m(i,k) < m(right(i),k)
⇒ m(i,k) < m(i,k+1)
Lemma 5
nextCirc_num_less
∃i. min(k) < m(i,k)
⇒ num(min(k),k+1) < num(min(k),k)
max_repeat_nextCirc_le
max(1)≦ max(k)
min_repeat_nextCirc_le
min(k) ≦ min(1)
num_repeat_nextCirc_zero
(∀k. ¬stable(k))
⇒ ∃k'. num(min(k),k+ k') = 0
min_repeat_nextCirc_less
(∀k. ¬stable(k)) ⇒ ∃k'. min(k)< min(k+ k')
min_repeat_nextCirc_inc
(∀k. ¬stable(k)) ⇒ ∃k'. min(k)+n < min(k+ k')
Theorem
eventually_stable
∃k. stable(k)
stable(k) = ∀i ∈{1,…,N}. m(i,k)=min(k)