Concurrency Theory
Sheet 5

Philipp Habelitz Patrick Hansert
Tobias Zimmermann

Coin game (How large can a finite Petri net get?)

We define a strategy for N buckets. Let $f(N)$ be the number of coins we get with this strategy.

We keep the coin in bucket 1 that is initially containted in it and apply our strategy to the remaining $N-1$ buckets. This results in the game state $(1, 0, \dots, f(N-1))$. Now we spend the coin in bucket 1 to apply action 2: $swap(1, 2, N)$, resulting in state $(0, f(N-1), 0, \dots, 0)$.

For this we define another strategy, which takes a game in state $(n, 0, \dots, 0)$ with m buckets. Let $g(m, n)$ be the number of coins resulting from this strategy. Therefore $f(n) = g(n-1, f(n-1))$. For this second strategy, in the first step our only option is to take one coin from the first bucket and put 2 in the next, resulting in state $(n-1, 2, 0, \dots, 0)$. Now we apply the strategy for $m-1$ buckets, which leads to state $(n-1, 0, \dots, g(m-1, 2))$ and use $swap(1, 2, m)$, which gives us state $(n-2, g(m-1, 2), 0, \dots)$. At this point we can apply the same strategy recursively n times, so $g(m, n) = g(m-1, g(m, n-1))$.

As a basis for the two recursions, we know that $f(1) = 1, f(2) = 3, f(3) = 7$, which can easily be shown by looking at all possible actions, $g(m, 1) = g(m-1, 2)$, because with one coin we cannot do the swap action for the next recursion step, so we only have two coins we get by using action 1, and $g(2, n) = 2n$, since the only option with two buckets is to use action 1.

Theorem 1. $g(m, n) \geq 2 \uparrow^{m-2} n$

Proof by induction.

- $m = 3$:

Proof by induction.
$n = 1$:
$g(3, 1) = g(2, 2) = 2 \cdot 2 = 4 \geq 2 \uparrow^1 2.$

$n \to n + 1$:
$g(3, n + 1) = g(2, g(3, n)) \overset{IH_2}{\geq} g(2, 2 \uparrow^1 n) = 2 \uparrow^1 (n + 1).$

- $m \to m + 1$:

  Proof by induction.
  $n = 1$:
  $g(m + 1, 1) = g(m, 2) \overset{IH_1}{\geq} 2 \uparrow^{m-2} 2 = 2 \uparrow^{m-1} 1.$

  $n \to n + 1$:
  $g(m + 1, n + 1) = g(m, g(m + 1, n)) \overset{IH_2}{\geq} g(m, 2 \uparrow^{m-1} n) \overset{IH_1}{\geq} 2 \uparrow^{m-2} (2 \uparrow^{m-1} n) = 2 \uparrow^{m-1} (n + 1)$

$\square$

Theorem 2.
For any sequence of actions that uses inc(1) while $(0, 2^{n-2}, 2^{n-3}, \ldots, 2^{n-n})S \geq 4$, *where* S *is the current state, there is another strategy that produces at least as many coins and does not use inc(1) in such a situation, for* $n \geq 3$ *buckets.*

Proof.
Suppose inc(1) is used in such a situation on state S. Choose $i > 1$ minimal with $S_i > 0$. Then we can replace inc(1) by the following sequence of operations:

- propagate all coins from bucket 1 to bucket n
- swap(1,2,n)
- leave $S_2 + 2$ coins in bucket 2 and propagate the rest to bucket i
- leave S_i coins in bucket i and propagate the rest to bucket n

We show that the state obtained by applying this sequence of operations covers this state obtained by applying inc(1). Therefore the remainder of the original sequence of operations remains valid.
Bucket 1 has 1 coin less after both sequences of operations, bucket 2 has 2 additional coins and therefore the same number of coins in both cases, bucket n has at least as many as in the other case and all other buckets have exactly the same number of coins.

- Case $i = n$:
  In this case we know that $S_n \geq 4$.
  Then we have $S_2' = S_n$, $S_n' = 2^{n-2}(S_n - 2) \geq 2(S_n - 2) \geq S_n$

- Case $i = n - 1$:
  After the first propagation $S_2' = S_i' = 0$, $S_n' = S_n + 2^{n-i}S_i$. After swapping and the second propagation, $S_2'' = 2$, $S_i'' = 2^{i-2}(S_n + 2^{n-i}S_i - 2) = 2^{n-3}(S_n + 2S_i - 2) = 2^{n-3}S_n + 2^{n-2}(S_i - 1)$.

  - $S_i > 1$:
    Then $2^{n-2}(S_i - 1) \geq S_i$ and we propagate at least $2^{n-3}S_n$ coins to bucket n, which will contain at least $4S_n \geq S_n$ coins.
  - $S_i = 1$:
    Because this state was generated by a sequence of actions starting from $(1, \ldots, 1)$, we have $S_n \geq 2$. We then propagate all but one token from S_i to S_n, where we will have $2(2^{n-3}S_n - 1) = 2^{n-2}S_n - 2 \geq S_n$ coins.

- Case $i < n - 1$:
  After the first propagation $S_2' = S_i' = 0$, $S_n' = S_n + 2^{n-i}S_i$. After swapping and the second propagation, $S_2'' = 2$, $S_i'' = 2^{i-2}(S_n + 2^{n-i}S_i - 2) \geq 2^{i-2}S_n + 2^{i-2}\underbrace{(4S_i - 2)}_{\geq S_i}$. Therefore we propagate at least $2^{i-1}S_n$ coins to bucket n and get at least $2^{n-2}S_n \geq S_n$ coins.

$\square$

Theorem 3. *If swap(i,j,k) is used in an optimal sequence of operations on $n \geq 3$ buckets then*

(1) $\forall j \leq x < k : S_x = 0$

(2) $j = i + 1$

(3) $k = n$

Proof.

(1) If $\exists j \leq x < k : S_x > 0$ we can propagate the S_x coins to bucket n, so $S_n' = S_n + 2^{n-x}S_x$, then do the swap and propagate the $2^{n-x}S_x$ additional coins to bucket x, so $S_x' = S_x + 2^{n-x+x-j}S_x \geq S_x$. Therefore an optimal sequence cannot have such an S_x.

(2) If $j > i+1$ we can instead do $swap(i, i+1, k)$ and propagate the coins S_k in bucket i to bucket j, so $S'_j = 2^{j-i-1} S_k \geq S_k$. In case $S_{i+1} > 0$ we can apply the same strategy as in the previous case. This means $j = i + 1$ for all swap operations in an optimal sequence.

(3) If $k < n$ we can instead do propagate the coins in bucket k to bucket n, so $S'_n = S_n + 2^{n-k} S_k$ and do $swap(i, j, n)$ and then propagate the additional $S_n + (2^{n-k} - 1)S_k$ coins to bucket n, which yields $S''_n = 2^{n-j}(S_n + (2^{n-k} - 1)S_k) \geq S_n$. Therefore $k = n$ for all swap operations in optimal sequences.

□

Theorem 4. *$g(m, n)$ is optimal.*

Proof.

- $m = 2$: The only action that can be applied is increment, therefore the statement trivially holds.
- $m > 2$: This result follows immediately from Theorem 2 and Theorem 3.

□

Theorem 5. *$f(n)$ is optimal.*

Proof. The base cases $f(1), f(2), f(3)$ can be verified by brute force.

With Theorem 2 and Theorem 3 we know that as soon as we take the coin from bucket 1 in an optimal sequence of operations, we are in a state that has the form $(1, 0, \ldots, 0, X)$. Clearly $X = f(n-1)$. With Theorem 2 we know that the next action will be $swap(1, 2, n)$. Since, by Thorem 4 g is optimal, $g(n-1, f(n-1))$ is the optimal value for $f(n)$. □

Theorem 6. *f and g are not primitive recursive.*

Proof.
By Theorem 4 g has a lower bound given by $g(m, n) \geq 2 \uparrow^{m-2} n$, which we know not to be primitive recursive. Clearly, f cannot be primitive recursive, either. □

Theorem 7. *All of humanity does not have enough SMU to play the coin game optimally with 5 buckets.*

Proof.
Suppose there are 10^9 humans. We know that the richest human has $< 2^{40}$ SMU. Therefore all of humanity has $< 10^9 \cdot 2^{40}$ SMU.
Clearly $f(5) \geq 2 \uparrow^2 256 \gg 10^9 2^{40}$. $\square$

By Theorem 7 and the fact that $f(4) = 256 < 2^{40}$, 5 is the minimal N with which we can become the richest humans by playing the coin game. By Theorem 5, $f(N)$ is the maximum number of coins that can be gained by playing the coin game with N buckets.