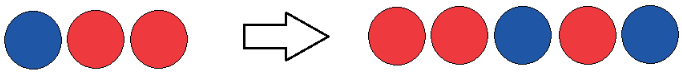
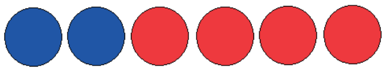
5. Sequenze di palline.

In questo paragrafo, e nei due successivi, vi proporrò alcuni giochetti “indecidibili” − in generale e per via algoritmica, s’intende: qui una soluzione c’è, potete trovarne una col ragionamento e poi confrontarla con quella che ho scritto nell’appendice del volume *on-line.*

Iniziamo con questo: supponete di avere una riserva illimitata di palline, sia rosse sia blu, e di metterne in fila un certo numero, a vostro piacere. Poi, per continuare, avete una sola regola, qui sotto illustrata: una sequenza di palline come quella a sinistra (se è presente nella fila) sarà sostituita con la sequenza a destra (che, notate, è più lunga); dovrete procedere finché sarà possibile fare qualche sostituzione.



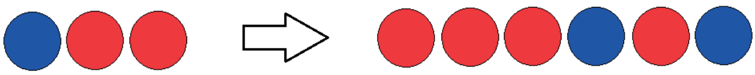
Se partite da questa fila di palline:



quale fila otterrete alla fine?

Esisterà qualche fila di palline partendo dalla quale il procedimento non termina?

Se ora modifichiamo la nostra regola nella seguente (notate la “piccola” differenza):



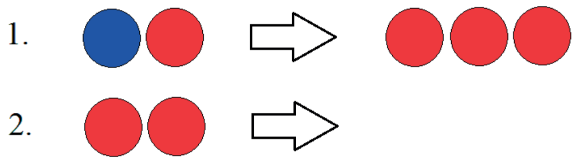
sapreste trovare una fila di palline − possibilmente di lunghezza minima − partendo dalla quale il procedimento non termina?

Accenniamo alle importanti nozioni di informatica nascoste sotto questo giochetto.

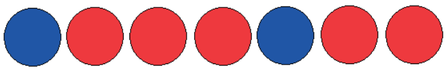
Un *sistema di riscrittura di stringhe* (storicamente *semi-sistema di Thue*: Axel Thue, inizi del Novecento) è costituito da un insieme finito di simboli e da un insieme finito di regole che descrivono tutte le ammissibili trasformazioni di *sequenze finite di simboli*, le stringhe per l’appunto. Per fare un esempio, più complesso ma di certo familiare a tutti, consideriamo le espressioni aritmetiche da “ridurre” a un singolo numero, proposte in tanti esercizi alle scuole medie. Ad ogni passo di riscrittura, una sottoespressione è sostituita con un’altra − di solito più breve! − ad essa equivalente, sicché il valore (ossia il significato) della intera espressione rimane lo stesso; e il processo continua, appunto, finché si può applicare qualche regola di riscrittura.

In generale, dato un sistema di riscrittura, bisogna chiedersi se tale processo termina sempre, da qualsiasi stringa si parta, e, in caso affermativo, se è anche *confluente*, vale a dire se, quando a un passo di riscrittura sono permesse più sostituzioni, tutte le scelte possibili conducono infine alla stessa stringa non più riscrivibile, la cosiddetta “forma normale” della stringa di partenza (che, in queste ipotesi, esiste ed è unica).

Facciamo ora un esempio con due regole: ad ogni passo, si applica o l'una o l'altra regola, finché è possibile applicare almeno una delle due. (Si noti che la seconda stabilisce che una sequenza di due palline rosse può essere eliminata.)



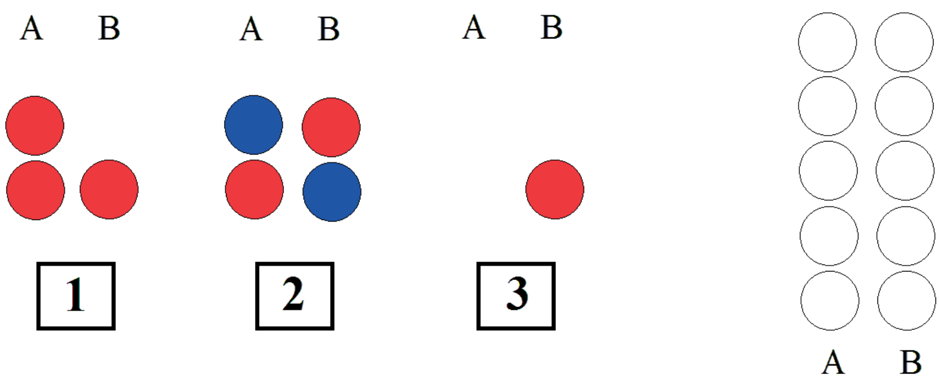
Anche qui chiediamoci: esiste qualche fila di palline partendo dalla quale il processo non termina? E inoltre: *quali* sequenze possono essere ottenute a partire da quella sotto disegnata? (Questo è un esempio di sistema di riscrittura *non confluente*!)



Un'ultima nota: il classico *problema della parola* (date le regole, la stringa iniziale e quella finale, stabilire se quest'ultima può essere ottenuta a partire dalla stringa iniziale) è in generale indecidibile (è soltanto *semidecidibile*) per via algoritmica.
Inoltre, è stato provato che, quando vi è una sola regola, la confluenza è decidibile, ma sulla terminazione nulla si sa; in generale, entrambe le questioni sono indecidibili.

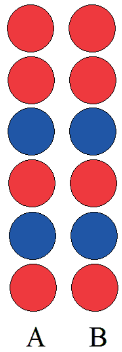
6. Cascate di palline.

Leone sta provando un nuovo *puzzle*, che ha appena caricato sul proprio computer.
Alcune coppie di colonne, formate da palline di vari colori, compaiono sul display, a sinistra; ora sono uscite tre coppie, con palline rosse e blu, e la colonna A della terza coppia è vuota (il bottone 3 lascerà cadere una sola pallina rossa nel cilindro B):

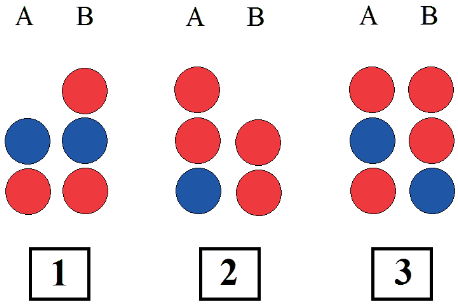


Ogni volta che Leone clicca su uno dei bottoni numerati, la corrispondente coppia di colonne cade, dall'alto, nei due cilindri A e B (all'inizio vuoti) a destra sul display.
Il *puzzle* riesce quando nei due cilindri si ottengono due colonne *identiche*.

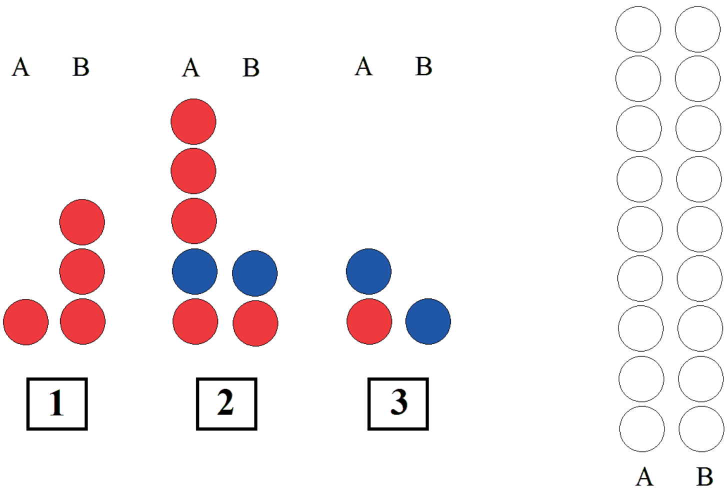
Non c'è limite al numero di *click* sui bottoni: assumiamo pure che le coppie di palline di stesso colore, affiancate in fondo ai cilindri, scompaiano verso il basso, e quelle sopra scendano. Pigiando la sequenza [3, 2, 2, 1] si formerebbe la cascata qui a destra, e il *puzzle* sarebbe risolto. Notiamo che, in questo caso, ogni sequenza della forma [3, n volte 2, 1], con $n \geq 0$, è una soluzione, e così tutte quelle costituite dalle sue ripetizioni; sicché la soluzione più breve è [3, 1], che evita persino di pigiare il bottone 2!

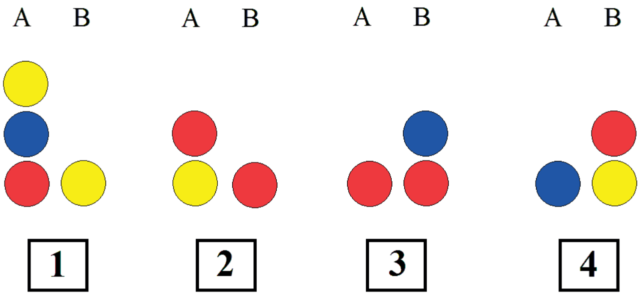


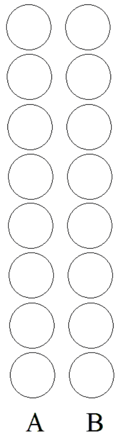
In generale, da ciascuna soluzione, se ne può costruire un'infinità di altre, ripetendo tale soluzione un numero arbitrario di volte. Certi casi, come quello ora illustrato, ammettono infinite soluzioni *non* ottenute per ripetizione. E, naturalmente, vi sono (infiniti) casi che non ammettono alcuna soluzione; eccone un esempio:

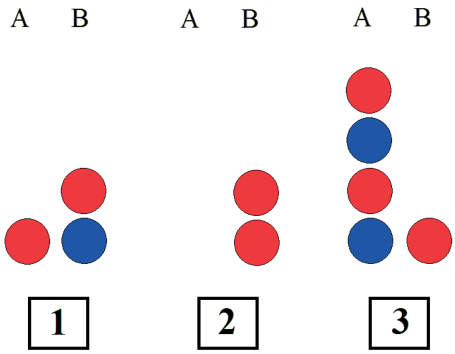


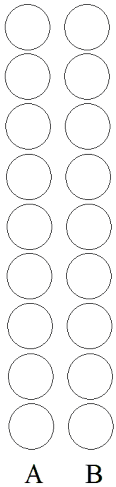
Vi propongo qui di seguito dei *puzzle* da risolvere, in ordine più o meno crescente di difficoltà: alcuni hanno quattro bottoni, il secondo anche palline gialle, ma sono tutti risolubili. Buon divertimento… ma non sperate di progettare un programma che possa preparare e poi risolvere *puzzle* casuali, pur limitati a pochi bottoni, poiché si tratta di istanze del *problema della corrispondenza* di Post, anch'esso in generale soltanto semidecidibile (Emil Leon Post, 1946), di certo se i bottoni sono almeno 5.

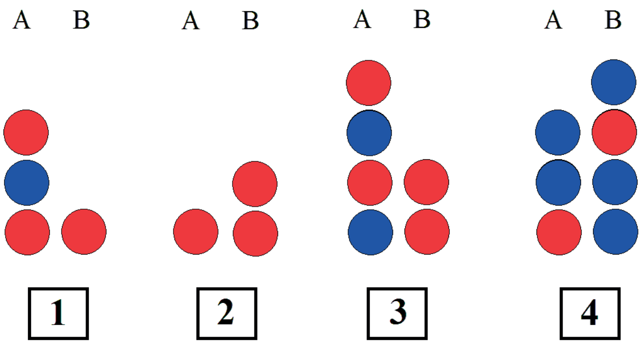


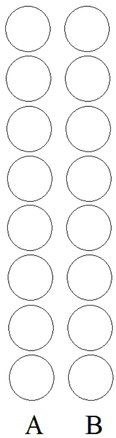




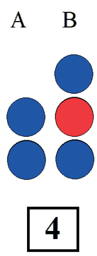








Se, in quest'ultimo caso, il bottone 4 si presentasse come nella figura qui a destra (notate che sono state tolte soltanto le due palline alla base della precedente configurazione), allora il *puzzle* non sarebbe più risolvibile…



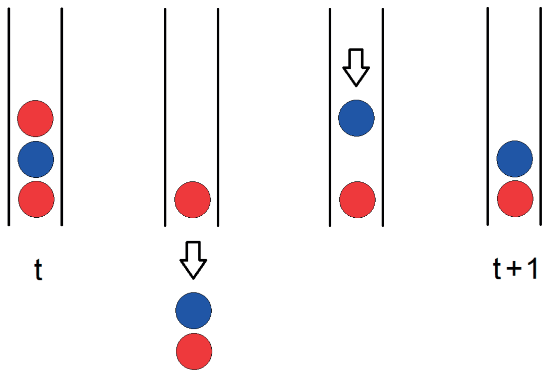
Il matematico e logico statunitense, di origine polacca, Emil Leon Post (1897-1954) diede notevoli contributi all'informatica teorica. Già negli anni '20 del Novecento, egli ideò dei particolari sistemi di riscrittura, equivalenti alle *macchine di Turing*, ma pubblicò i risultati ottenuti soltanto nel 1943. Proprio questi *sistemi di Post* hanno ispirato il prossimo quesito, da me proposto per le gare Bebras nel 2016.

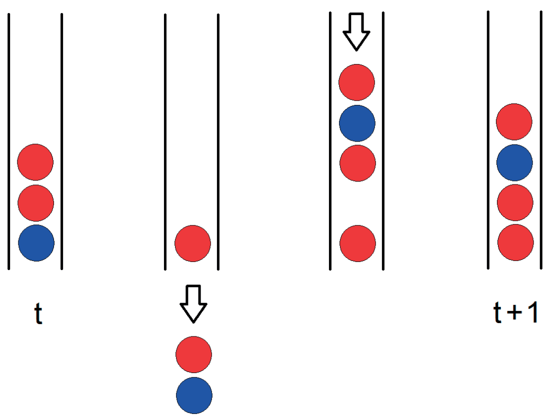
7. Palline in un cilindro.

Leone ha ideato un altro gioco con palline di due colori, e lo ha realizzato mediante un programma al computer che, attualmente, permette all'utente di predisporre, in un cilindro riprodotto sullo schermo, una colonna di almeno tre palline, ciascuna delle quali può essere o rossa o blu. Dopodiché, ad ogni unità di tempo prefissata, le due palline più in basso escono dal fondo del cilindro e, immediatamente, ne cade qualcun'altra dall'alto, secondo queste regole:

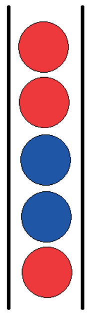
- se la pallina che esce per prima dal fondo è rossa, allora cade una nuova pallina blu in cima al cilindro;
- se invece la pallina che esce per prima dal fondo è blu, allora cadono tre nuove palline in cima al cilindro: una rossa, una blu e una rossa, in quest'ordine.

Il processo termina soltanto se e quando rimangono meno di tre palline nel cilindro.
Le seguenti figure esemplificano le due regole; nel primo caso, il processo termina al tempo t +1, poiché restano due palline.





E veniamo a un paio di domande interessanti.
Se il programma è fatto partire con la colonna raffigurata qui a destra, dopo quante unità di tempo si arresterà?
Sapreste trovare una colonna iniziale di sole tre palline, partendo dalla quale il programma non si fermerà mai più?



Come abbiamo anticipato, questo gioco non è altro che un esempio di *sistema di produzioni di Post*, un modello di calcolo fondato ancora sulla riscrittura di stringhe, intese come sequenze finite di simboli.

In effetti, un sistema di Post, esattamente come una *macchina di Turing*, costituisce una possibile definizione di un ben preciso *algoritmo*, inteso come procedimento di calcolo per ottenere un "risultato" a partire da un "dato di input" rappresentato dalla stringa iniziale, ammesso che a tale ingresso corrisponda un'uscita.
Volendo essere più precisi, il sistema sul quale si basa il nostro gioco è un 2-*tag system*, poiché due simboli (palline) sono cancellati in fondo alla stringa (escono dal cilindro) ad ogni passo. L'insieme dei sistemi 2-tag è stato provato *Turing-completo*: ciò significa che esso ha la stessa capacità di calcolo di una *macchina di Turing universale*; da quando è stata ideata, nel 1936, si ritiene che quest'ultima − di cui i nostri computer costituiscono una realizzazione, seppur con memoria limitata − sia in grado di calcolare tutto quanto è, in linea di principio, calcolabile.
Più in generale, si può affermare che, per ogni $m > 1$, l'insieme dei sistemi m-tag è Turing-completo: per ogni data macchina di Turing T, esiste un sistema m-tag che emula T. In realtà, a dire il vero, si possono costruire diversi sistemi 2-tag (ma parecchio più complessi di quello qui presentato, che ha due sole regole!) ciascuno dei quali è in grado di emulare una macchina di Turing universale (H. Wang, 1963; J. Cocke e M. Minsky, 1964).
A chi desideri approfondire l'argomento, segnalo la tesi di Turlough Neary: *Small universal Turing machines* (`www.ini.uzh.ch/~tneary/tneary_Thesis.pdf`).

8. Tchouka e Tchoukaillon.

Nel decimo capitolo del libro, abbiamo parlato di una delle centinaia di varianti di un'antichissima famiglia di giochi d'origine africana, detta degli Awélé o Mancala: sono giochi "di semina", poiché i due contendenti depositano semi (o, in mancanza, sassolini) nelle buche che formano il tavoliere, poi prelevano tutti i semi da una buca e li ridistribuiscono uno dopo l'altro nelle buche successive, rispettando determinate regole. Ne esistono anche tante versioni da giocare "in solitario", diffuse soprattutto in India, Malesia e Indonesia.
Il primo a riferire di *Tchouka* fu il matematico e storico francese, intendente militare Henri Auguste Delannoy, in una memoria pubblicata nel 1895, ove affermava che tale gioco gli era stato descritto in una lettera dal matematico Édouard Lucas, più volte citato nel mio libro, e non soltanto per le sue più note "ricreazioni", quali la torre di Hanoi e il congiungimento di punti (*pipopipette*, o *Dots-and-Boxes*, o *punti e linee*).