NP-hardness reductions from 3SAT / 1-in-3SAT / NAE 3SAT:

**Nintendo games:**
- Super Mario Bros. & World
  - glitches
- Legend of Zelda
  - push-once blocks
  - hookshot (A Link to the Past)
- Metroid
- Donkey Kong Country
- Pokémon
  - weak trainer: player always wins
  - strong trainer: player always loses
Phutball (Philosopher's Football) [Conway]
- one white stone (ball), many black stones (men)
- move = place new black stone
  or "kick the ball" by jumping horizontally
  or vertically over black stones, immediately removing men, & repeat
- goal: reach opponent's side with ball

- PSPACE-hard [Dereniowski 2009]
- OPEN: EXPTIME-complete?
- NP-complete to decide mate-in-1:
  can you win in one move? (kick)
- reduction from 3SAT [Demaine, Demaine, Eppstein 2008]

Checkers:
- EXPTIME-complete [Fraenkel, Garey, Johnson, Schaefer, Yesha 1978]
- mate-in-1 is polynomial
  - jumps preserve $x$ & $y$ parity
  $\Rightarrow$ Euler path problem
Cryptarithms/alphametics [Madachi 1979]
- given formula $x+y=z$ with each number written in base $b$ & encoded with “letters” by unknown bijection between $\{0,1,\ldots,b-1\}$ & letters
- goal: feasible? / recover bijection
- strongly NP-complete [Eppstein 1987]

Reduction from 3SAT:
- variable gadget:
  - $b_i = 2a_i$
  - $v_i = 2b_i + C = 4a_i + C \equiv C \pmod{4}$
  - $d_i = 2c_i + C$
  - $e_i = d_i + 1 + C = 2c_i + 1 + 2C$
  - $\overline{v_i} = d_i + e_i = 4c_i + 1 + 3C \equiv 3C + 1 \equiv 1 - C \pmod{4}$

- clause gadget:
  - $g_i = 2f_i$
  - $h_i = 2g_i + \{0,1\} = 4f_i + \{0,1\}$
  - $t_i = h_i + 1 + \{0,1,2\} = 4f_i + 1 + \{0,1,2\}$
  - $v_a + v_b + v_c = \overline{t_i} \equiv \{1,2,3\} \pmod{4}$
Simplified reduction from 1-in-3 SAT:
- variable gadget: just \( v_i \), no \( \overline{v}_i \) (monotone)
- clause gadget:
  - \( g_i = 2f_i \)
  - \( h_i = 2g_i = 4f_i \)
  - \( t_i = h_i + 1 = 4f_i + 1 \)
  - \( v_a + v_b + v_c = t_i = 4f_i + 1 \equiv 1 \pmod{4} \)

3SAT solvable \( \Rightarrow \) cryptarithmetic solvable:
- distinguish \( a_i, b_i, c_i, \ldots \) by value \( \pmod{128} \)
- e.g. \( v_i = 8 \pmod{128} \) if true
  \( = 9 \pmod{128} \) if false
  \( a_i = \{ 2, 34, 66, 98 \} \pmod{128} \)
  
- set \( [v_i / 128] \) & \( [\overline{v}_i / 128] \in [0, (2n)^3] \) such that distinct sums of triples
  [Bose & Chowla 1959]
- easy proof of polynomial range: \( \text{(based on fusion trees)} \)
  - if \( i < j \) set by induction, \( v_i \) must avoid
    \( v_j + v_k + v_x - v_m - v_p \sim (2n)^5 \) choices
  \( \Rightarrow (2n)^5 \) suffices
  \( \Rightarrow \) strongly NP-hard
Origami: flat-foldable crease patterns

- each crease
- folded $\pm 180^\circ$
- graph drawn in square with straight edges
- stacking order of overlapping faces

- strongly NP-hard [Bern & Hayes 1996]

Reduction from NAE 3SAT:

- wire = pleat
- NAE clause $\approx$ triangular twist
- splitter/negation (don’t need negation)
- crossover
**Vertex-disjoint paths:**

- in a graph [Lynch 1975]
- in a planar graph [Lynch 1975]
- in a rectangle with all spots filled [Adcock, Demaine, Demaine, O’Brien, Reidl, Sánchez Villaamil, Sullivan 2014]

- use terminals as obstacles
- neighboring terminal pairs can just connect or fill some uncovered space
- issue 1: must have even-parity fill regions
- issue 2: clause path may be absent!
  - more parity trouble → add row
- issue 3: gadget width is odd parity
  - won’t connect to split → add column
- issue 4: crossing true or false line gadgets

= Zigzag Numberlink [Loyd 1897; Nikoli]
- classic Numberlink also NP-complete [Kotsuma & Takenaga 2010]