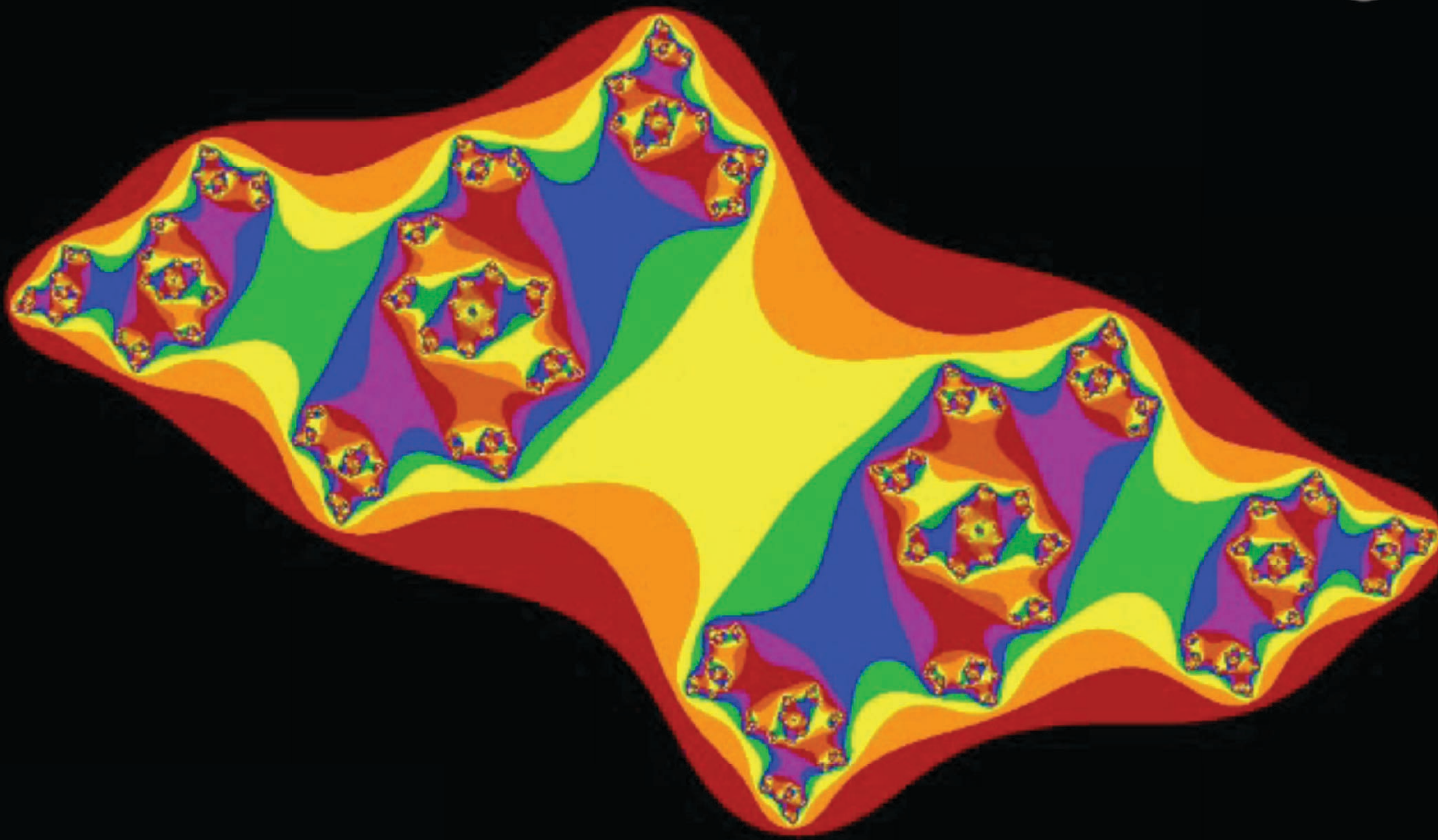
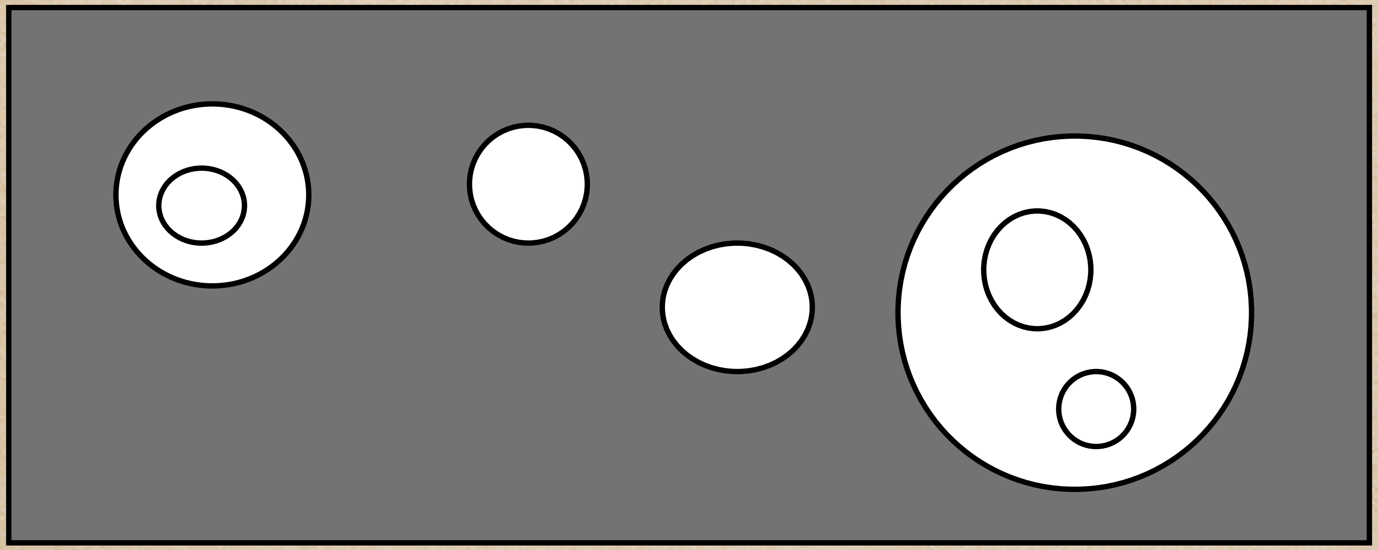
Termination

Well-Founded Orderings

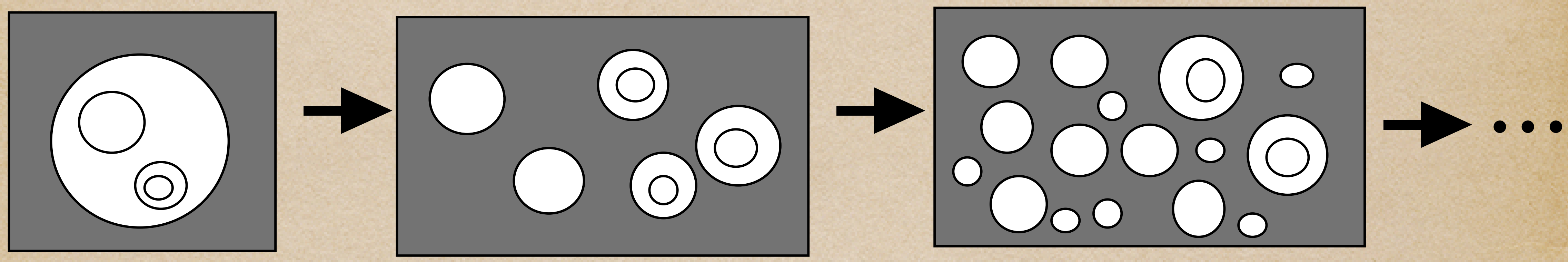
```
n := 0
while x > 0  do
    n := n + 1
    y := 0; while y^2 + 2y ≤ x  do y := y + 1
    if  x = y^2
    then x := y − 1
    else s := 0
        r := 0; while r^2 + 2r ≤ x − y^2  do r := r + 1
        while x > y^2 + r^2  do
            y := 0; while y^2 + 2y ≤ x  do y := y + 1
            s := s + (s + y^2 + y − x)^2
            x := x − y^2
            r := 0; while r^2 + 2r ≤ x − y^2  do r := r + 1
        for i := 1  to n  do x := r^2 + r − 1
        while s > 0  do
            r := 0; while r^2 + 2r ≤ s  do r := r + 1
            x := x + (x + r^2 + r − s)^2
            s := s − r^2
```

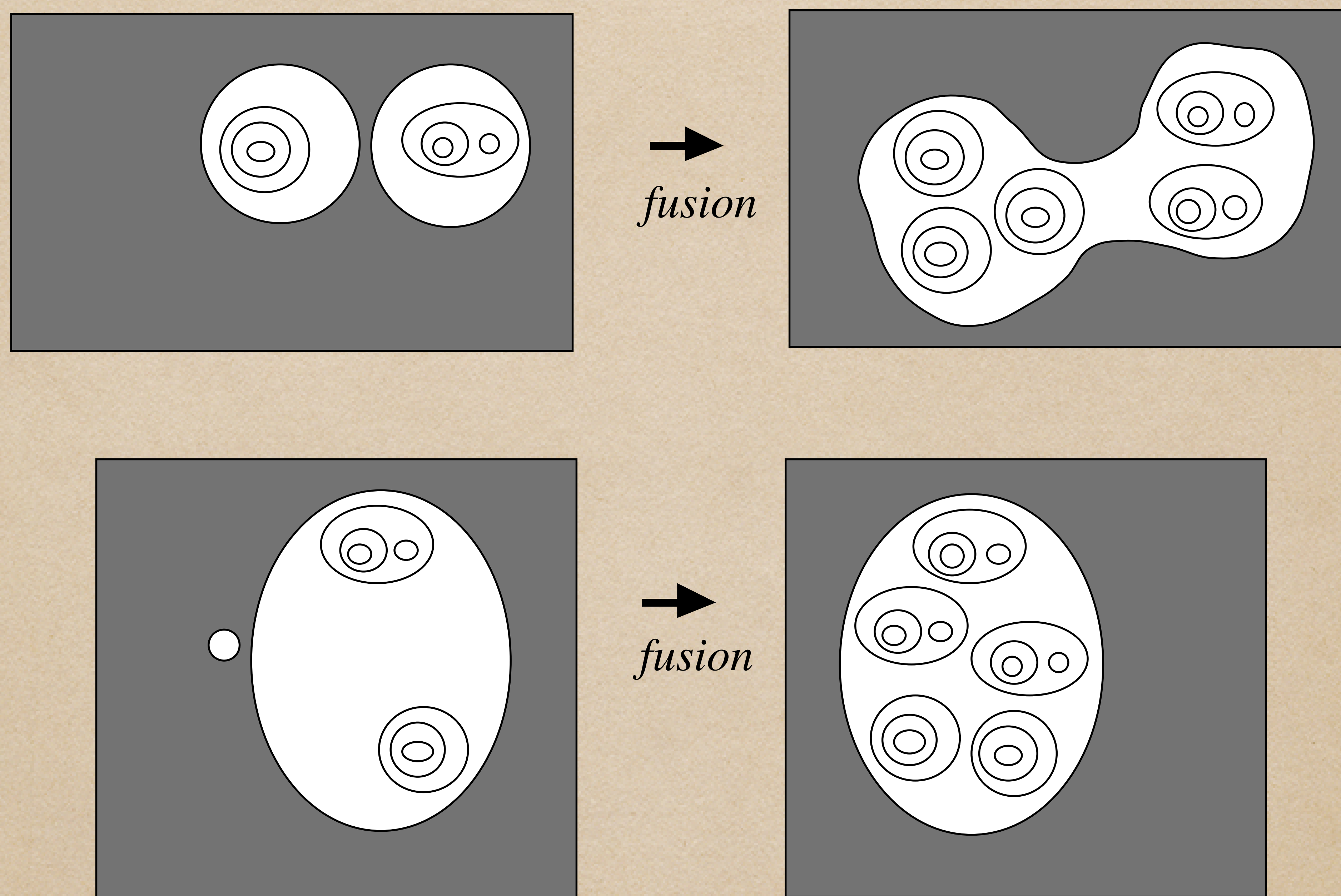


Amoebae



Fission




fusion
fusion

Colony Dies Out

- depth(o) = 0
- depth((a1 ... an)) = 1+max{depth{ai}}
- { (depth(a),|a|) : subcolony a }
- fission: depth decreases
- fusion: size decreases

Big Picture

- Programs are state-transition systems
- Choose a well-founded order on states
- Show that transitions are decreases

Real Picture

- Programs are state-transition systems
- Choose a function for “ranking” states
- Choose a well-founded order on ranks
- Show that transitions always decrease rank

Imaginary Picture

- Programs are state-transition systems
- Choose a function for “ranking” states
- Choose a well-founded order on ranks
- Show that transitions eventually decrease rank

Nested Loops

```
r := 1
u := 1
loop   v := u
       until r≥n
       s := 1
       loop   u := u+v
              s := s+1
              while s≤r
              repeat
       r := r+1
       repeat
```

$(n - r)\omega^2 + (r - s)\omega + k$

Per Iteration

```
r := 1
u := 1
loop    v := u
        until r≥n
        s := 1
        loop    u := u+v
                s := s+1
                while s≤r
                repeat
        r := r+1
        repeat
```

$\omega(n-r)+r+1-s$

Lexicographic

```
r := 1
u := 1
loop    v := u
        until r≥n
        s := 1
        loop  u := u+v
              s := s+1
              while s≤r
              repeat
        r := r+1
        repeat
```

(n-r,r+1-s)

Invariants

```
r := 1
u := 1
loop    v := u                          1≤r≤n
        until r≥n
        s := 1
        loop   u := u+v
               s := s+1         1≤s≤r+1
               while s≤r
               repeat
        r := r+1
        repeat
```

Well-Founded Orderings

- No infinite descending sequences

  $x1 > x2 > x3 > \ldots$

Well-Founded Induction

> is a wfo of X

$$\frac{\forall x \in X.\ [\forall y < x.\ P(y)] \Rightarrow P(x)}{\forall x \in X.\ P(x)}$$

Why?

David Gries

- Under the reasonable assumption that nondeterminism is bounded, the two methods are equivalent…. In this situation, we prefer using strong termination.

All-Purpose Ranks

$0 < 1 < 2 < \cdots$

$< \omega < \omega+1 < \omega+2 < \cdots$

$< \omega2 < \omega2+1 < \cdots < \omega3 < \cdots < \omega4 < \cdots$

$< \omega^2 < \omega^2+1 < \cdots < \omega^2+\omega < \omega^2+\omega+1 < \cdots$

$< \omega^3 < \omega^3+1 < \cdots < \omega^4 < \cdots < \omega^5 < \cdots$

$< \omega^\omega < \cdots < \omega^{\omega^\omega} < \cdots < \omega^{\omega^{\omega^\omega}} < \cdots$

Ordinals

$$0, 1, 2, \ldots,$$

$$\omega, \omega+1, \omega+2, \ldots,$$

$$2\omega, 2\omega+1, \ldots, 3\omega, \ldots,$$

$$\omega^2, \ldots, \omega^2+2\omega+3, \ldots, \omega^3, \ldots,$$

$$\omega^\omega, \ldots, \omega^{\omega^\omega}, \ldots,$$

$$\varepsilon_0, \varepsilon_0+1, \ldots, 2\varepsilon_0+\omega^\omega+2\omega+3, \ldots,$$

$$\varepsilon_1, \ldots, \varepsilon_{\varepsilon_0}, \ldots,$$

$$\ldots$$

THE GROWTH

$0, 1, 2, 3, \ldots, \omega, \omega+1, \omega+2, \omega+3, \ldots, \omega+\omega = \omega 2, \omega 2+1, \omega 2+2, \omega 2+3, \ldots, \omega 3, \omega 3+1, \omega 3+2, \omega 3+3,$

$\omega\omega = \omega^2, \omega^2+1, \omega^2+2, \ldots, \omega^2+\omega, \ldots, \omega^2+\omega 2, \ldots, \omega^2+\omega 3, \ldots, \omega^2 2, \ldots, \omega^2 3, \ldots, \omega^3, \ldots, \ldots,$

$\ldots, \omega^\omega \omega^2, \ldots, \omega^\omega \omega^3, \ldots, \omega^\omega \omega^\omega = \omega^{\omega 2}, \ldots, \omega^{\omega 3}, \ldots, \ldots, \omega^{\omega^2}, \ldots, \omega^{\omega^\omega}, \ldots, \omega^{\omega^{\omega^{\cdot^{\cdot^{\cdot}}}}} = \varepsilon_0, \varepsilon_0+1, \ldots,$

$\varepsilon_0+\omega 2, \ldots, \varepsilon_0+\omega^2, \ldots, \varepsilon_0+\omega^\omega, \ldots, \varepsilon_0+\varepsilon_0 = \varepsilon_0 2, \ldots, \varepsilon_0 3, \ldots, \varepsilon_0 \omega, \ldots, \varepsilon_0 \omega 2, \ldots, \varepsilon_0 \omega^2, \ldots,$

$\varepsilon_0 \varepsilon_0 = \varepsilon_0^2, \ldots, \varepsilon_0^3, \ldots, \varepsilon_0^\omega, \ldots, \varepsilon_0^{\omega^\omega}, \ldots, \varepsilon_0^{\varepsilon_0}, \ldots, \varepsilon_0^{\varepsilon_0^{\varepsilon_0^{\cdot^{\cdot^{\cdot}}}}} = \varepsilon_1, \ldots, \varepsilon_2, \ldots, \varepsilon_3, \ldots, \varepsilon_\omega$

$\varepsilon_{\omega^\omega}, \ldots, \varepsilon_{\varepsilon_0}, \ldots, \varepsilon_{\varepsilon_1}, \ldots, \varepsilon_{\varepsilon_\omega}, \ldots, \varepsilon_{\varepsilon_{\omega^\omega}}, \ldots, \varepsilon_{\varepsilon_{\omega^{\omega^\omega}}}, \ldots, \varepsilon_{\varepsilon_{\varepsilon_0}}, \ldots, \varepsilon_{\varepsilon_{\varepsilon_{\cdot_{\cdot_{\cdot}}}}} = \eta_0, \ldots, \eta_1, \ldots$

$\eta_{\varepsilon_0}, \ldots, \eta_{\varepsilon_\omega}, \ldots, \eta_{\varepsilon_{\varepsilon_0}}, \ldots, \eta_{\varepsilon_{\varepsilon_{\varepsilon_{\cdot_{\cdot_{\cdot}}}}}} = \eta_{\eta_0}, \ldots, \eta_{\eta_{\eta_{\cdot_{\cdot_{\cdot}}}}} = \zeta_0, \ldots, \ldots\ldots\ldots\ldots$

$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$

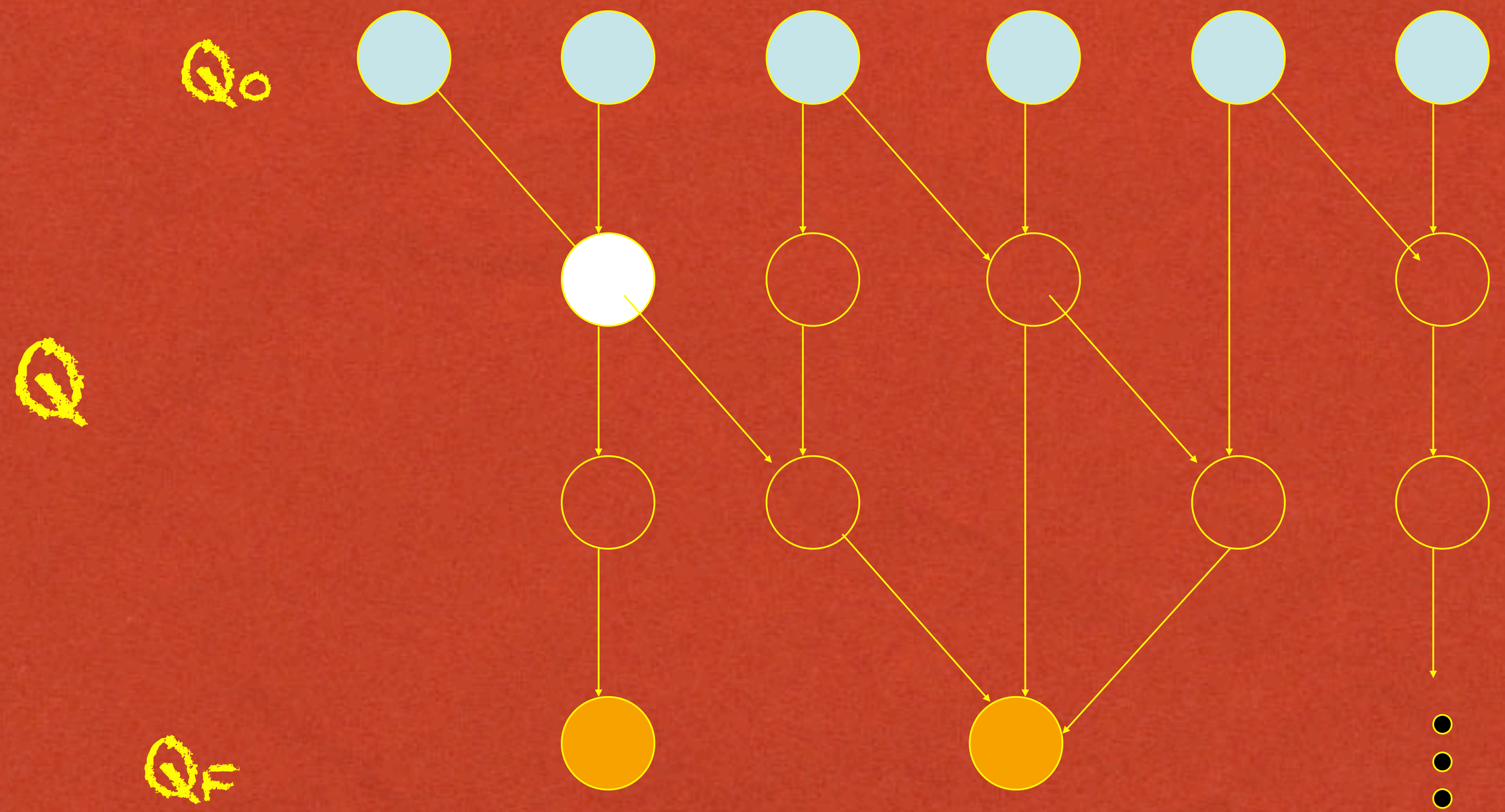
$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$

$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$

$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$

Transition System

Discrete Transition System



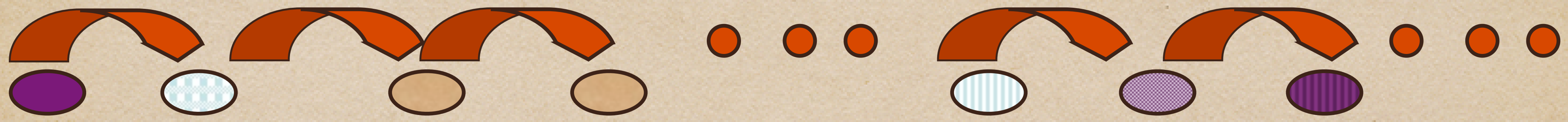
Well-Founded Method

- States Q
- Algorithm R ⊆ QxQ
- Well-founded order > on Q
- R ⊆ >

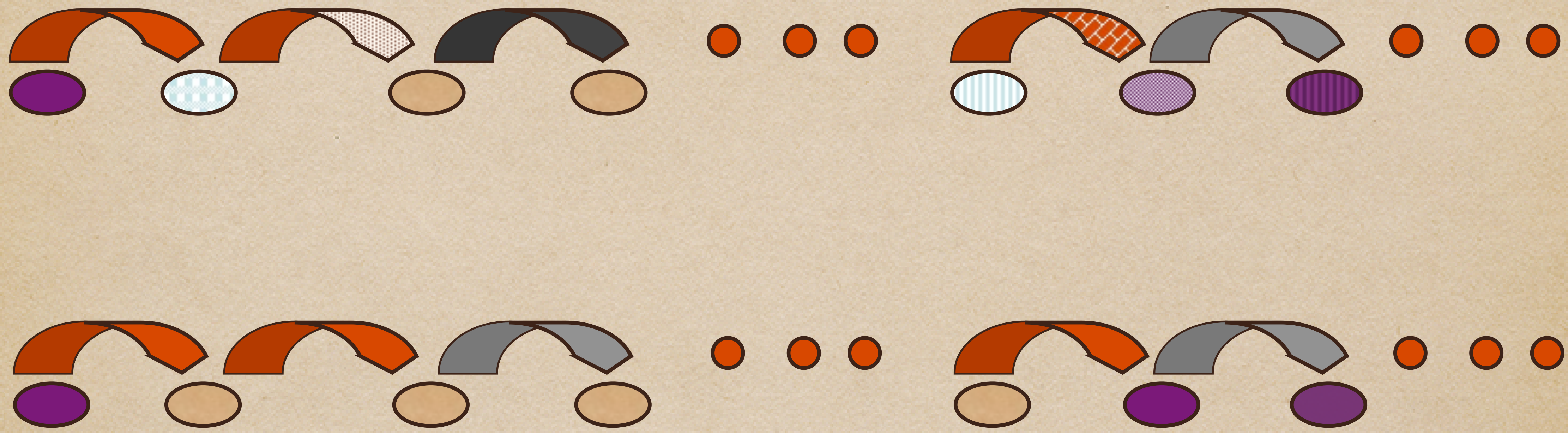
All-Purpose Ranking

- $r : Q \to \mathrm{Ord}$
- $r(x) = \sup \{ r(y)+1 : x \to y \}$

Computation



Abstraction



Frank Ramsey

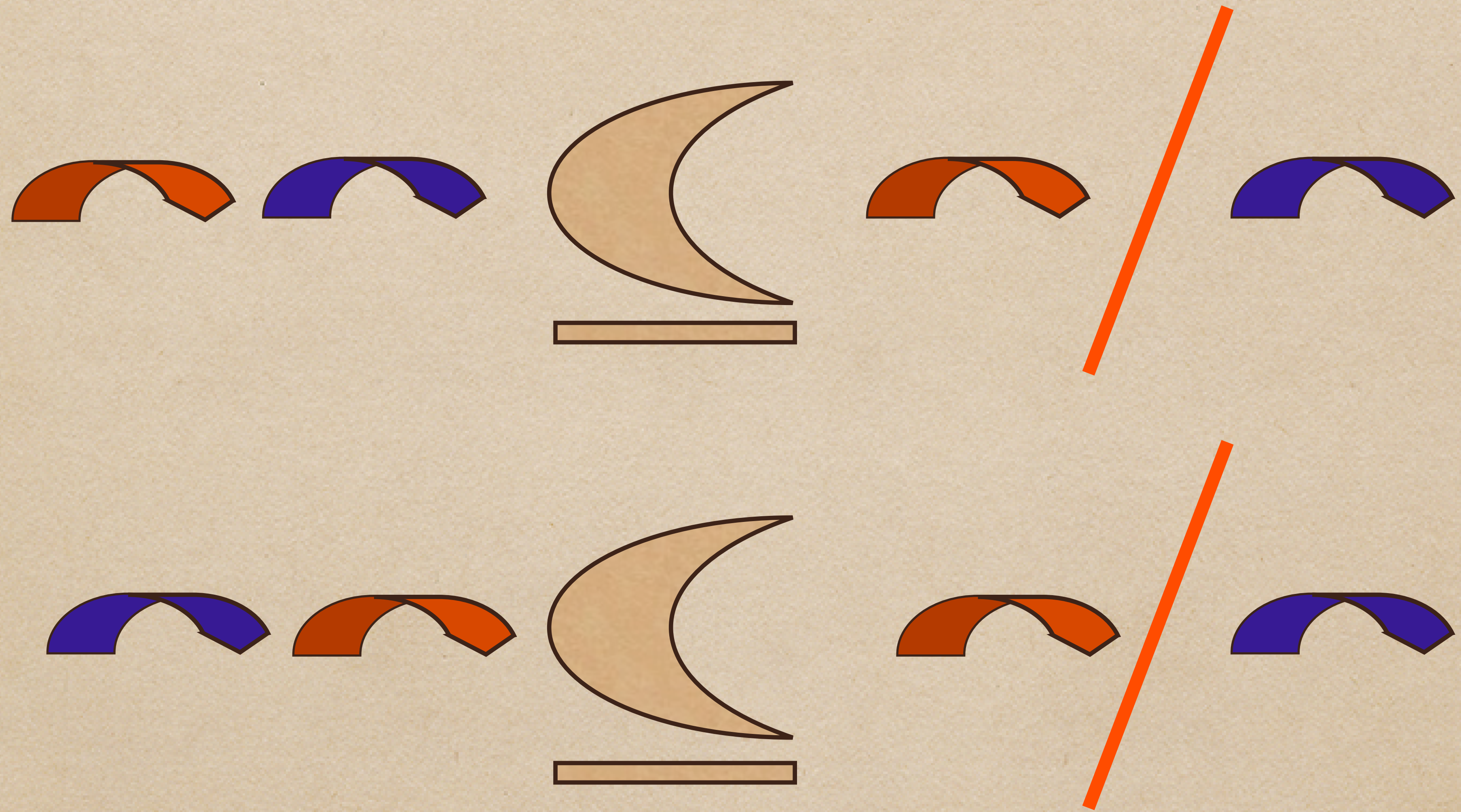
Ramsey's Theorem

Infinite complete graph

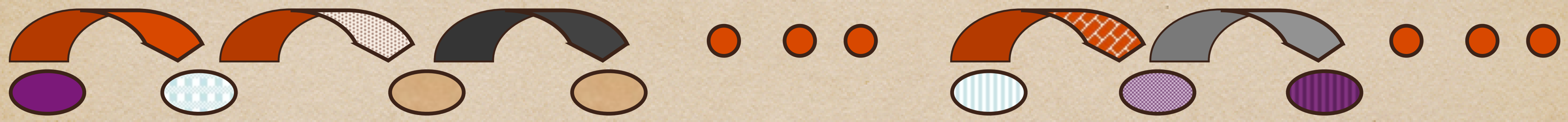
Finitely colored edges

Monochrome infinite clique

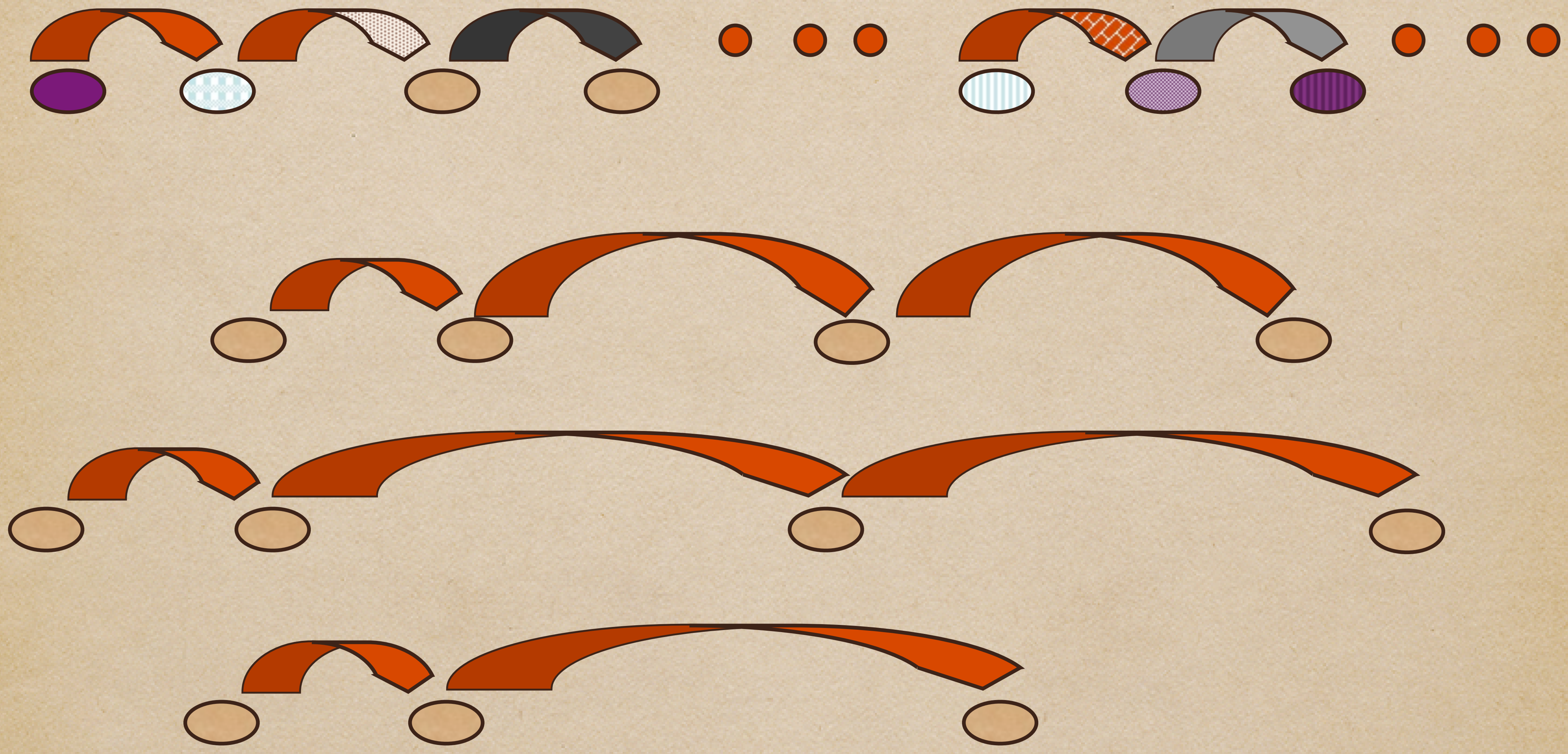
Closure



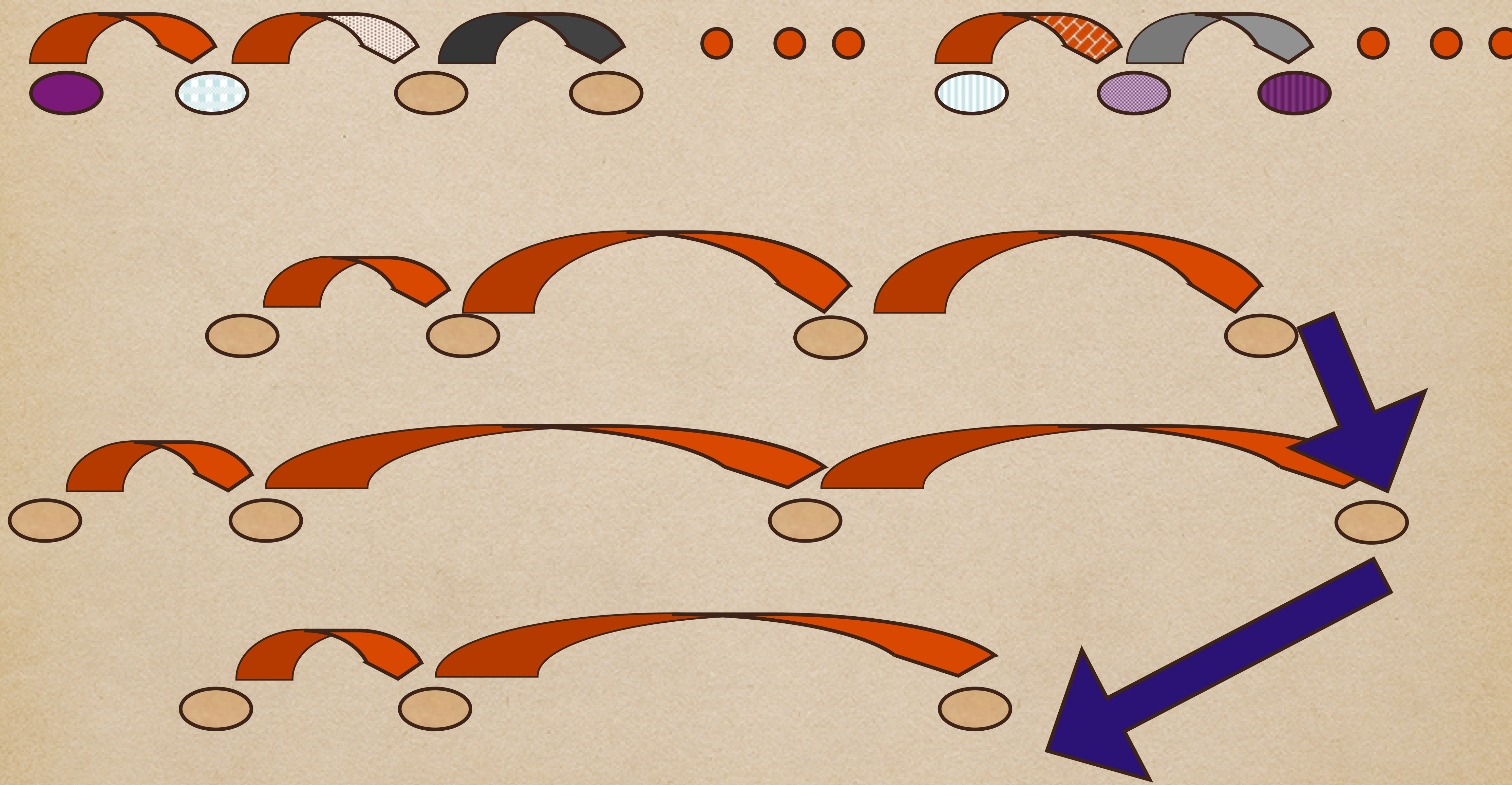
Proof



Proof



Proof



Disjunctive Orders

- States Q
- Algorithm $R \subseteq Q \times Q$
- Transitive closure R^+
- Well-founded orders $>$ and $\sqsupset$ on Q
- $R^+ \subseteq > \cup \sqsupset$

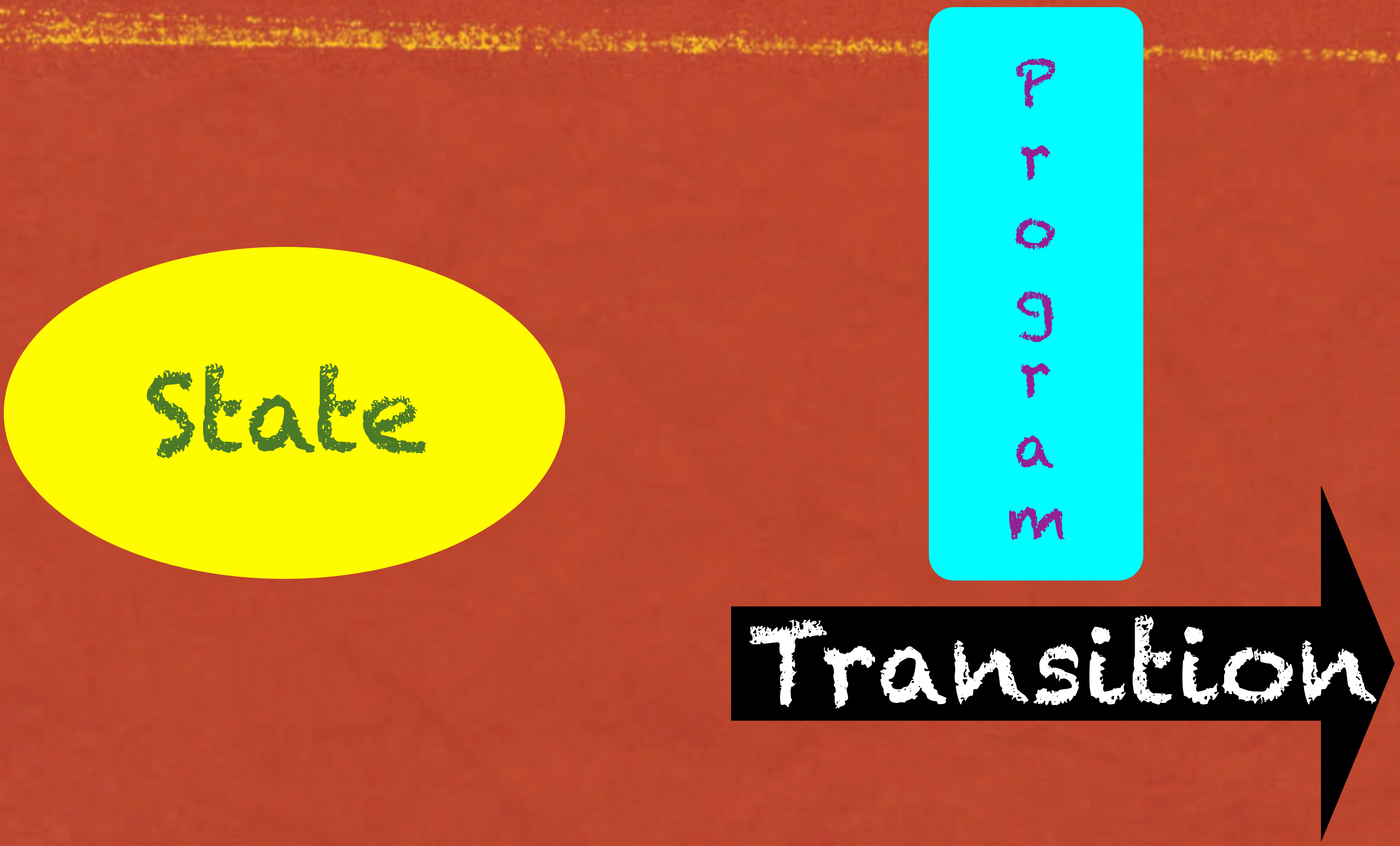
Ranking Method

- States Q
- Algorithm $R \subseteq Q \times Q$
- Well-founded order $\succ$ on W
- Ranking function $r : Q \rightarrow W$
- Define $X > Y$ if $r(X) \succ r(Y)$
- $R \subseteq >$

Invariants

- States Q
- Algorithm $R \subseteq Q \times Q$
- Well-founded order $\succ$ on W
- Ranking function $r : Q \rightarrow W$
- Define $X > Y$ if $r(X) \succ r(Y)$
- $R \subseteq >$

Algorithmic System



Classical Algorithms

- Every algorithm can be expressed precisely as a set of conditional assignments, executed in parallel repeatedly.

if c then f(s1,...,sn) := t

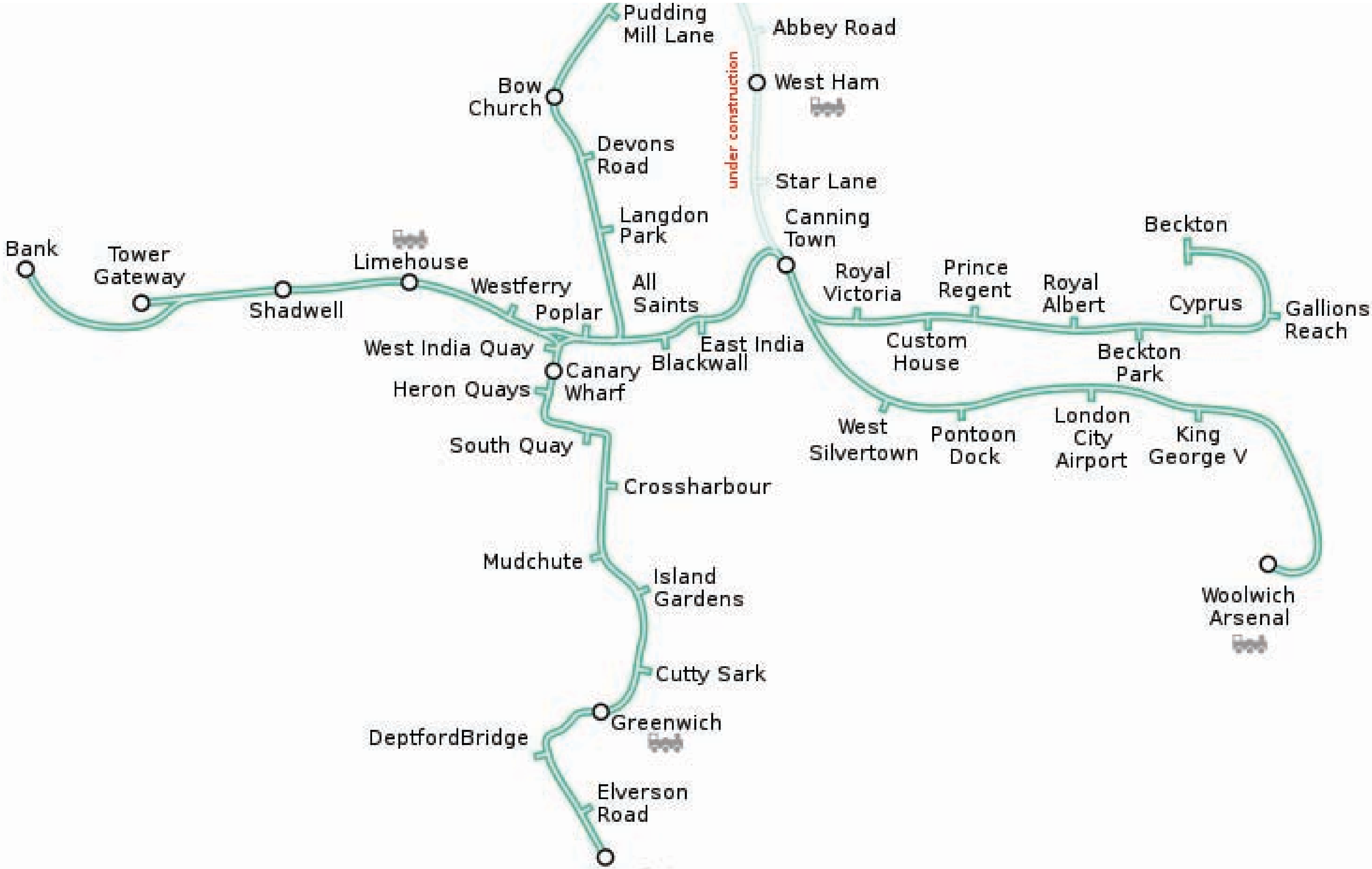
if c then f(s1,...,sn) := t

if c then f(s1,...,sn) := t

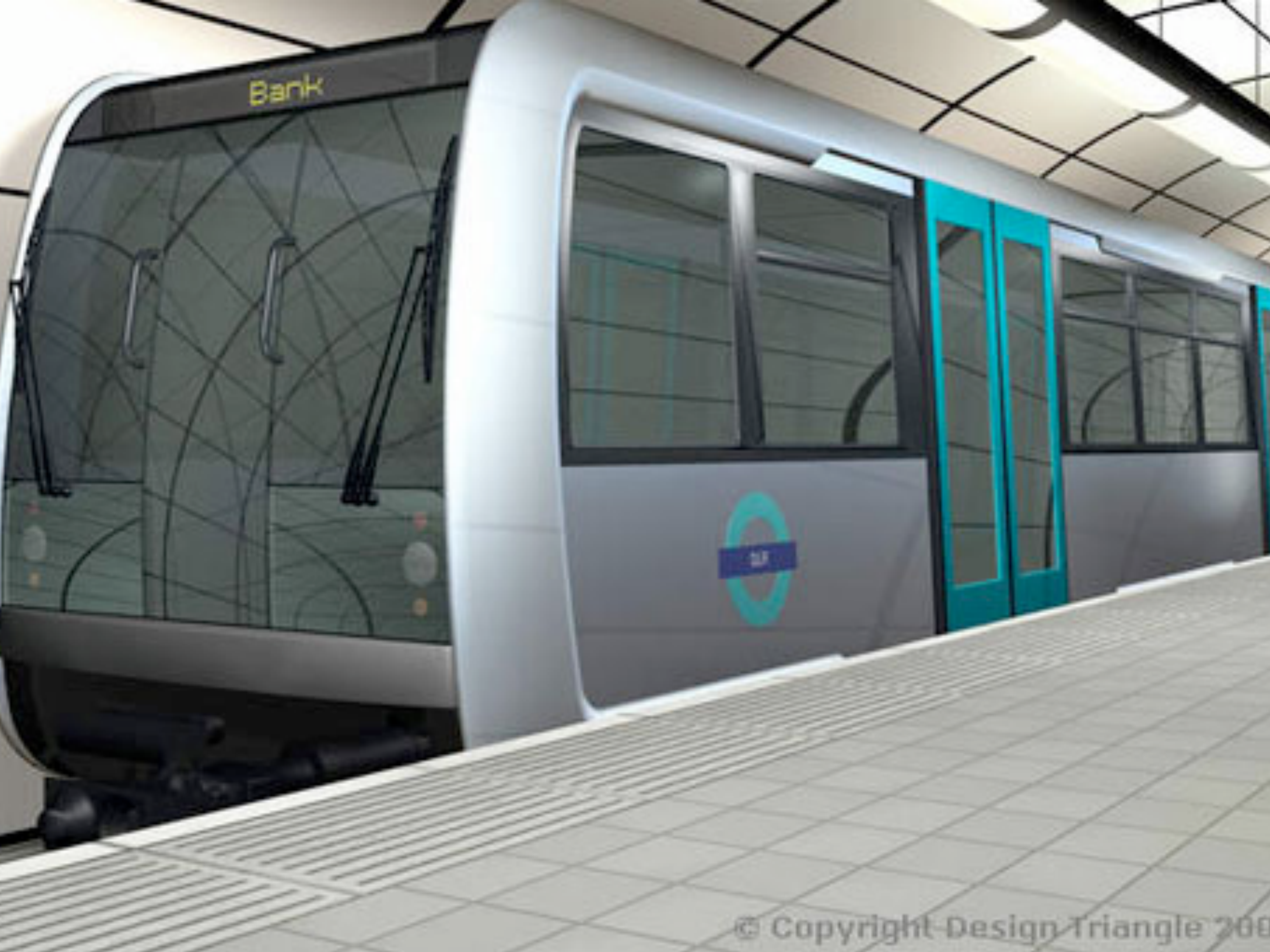
Practical Method

- States Q
- Algorithm $R \subseteq Q \times Q$
- Well-founded order $\succ$ on W
- Ranking function $r : Q \to W$
- Define $X > Y$ if $r(X) \succ r(Y)$
- $R \subseteq >$

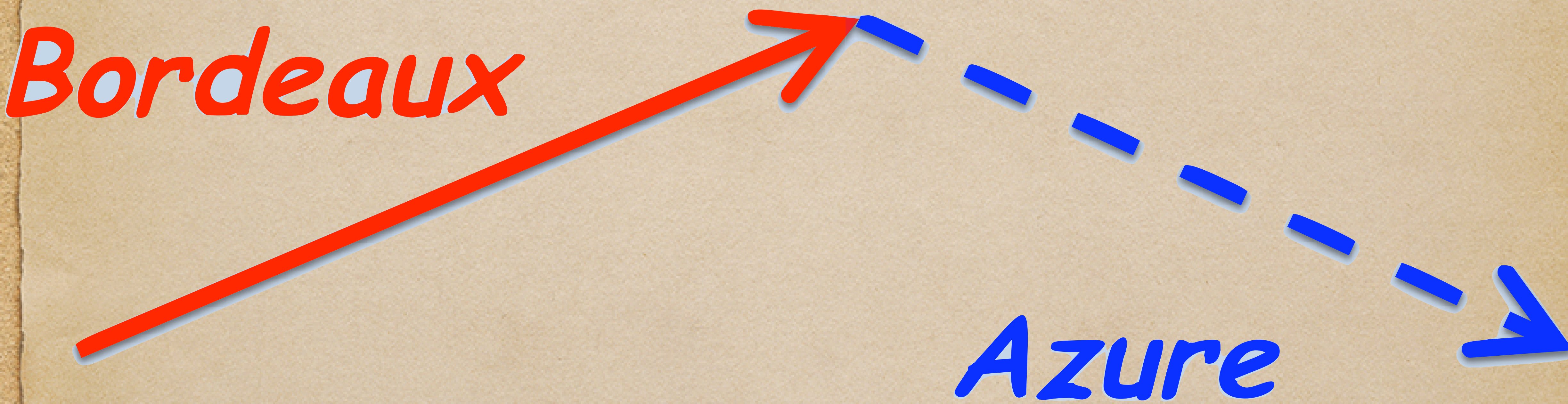
DLR

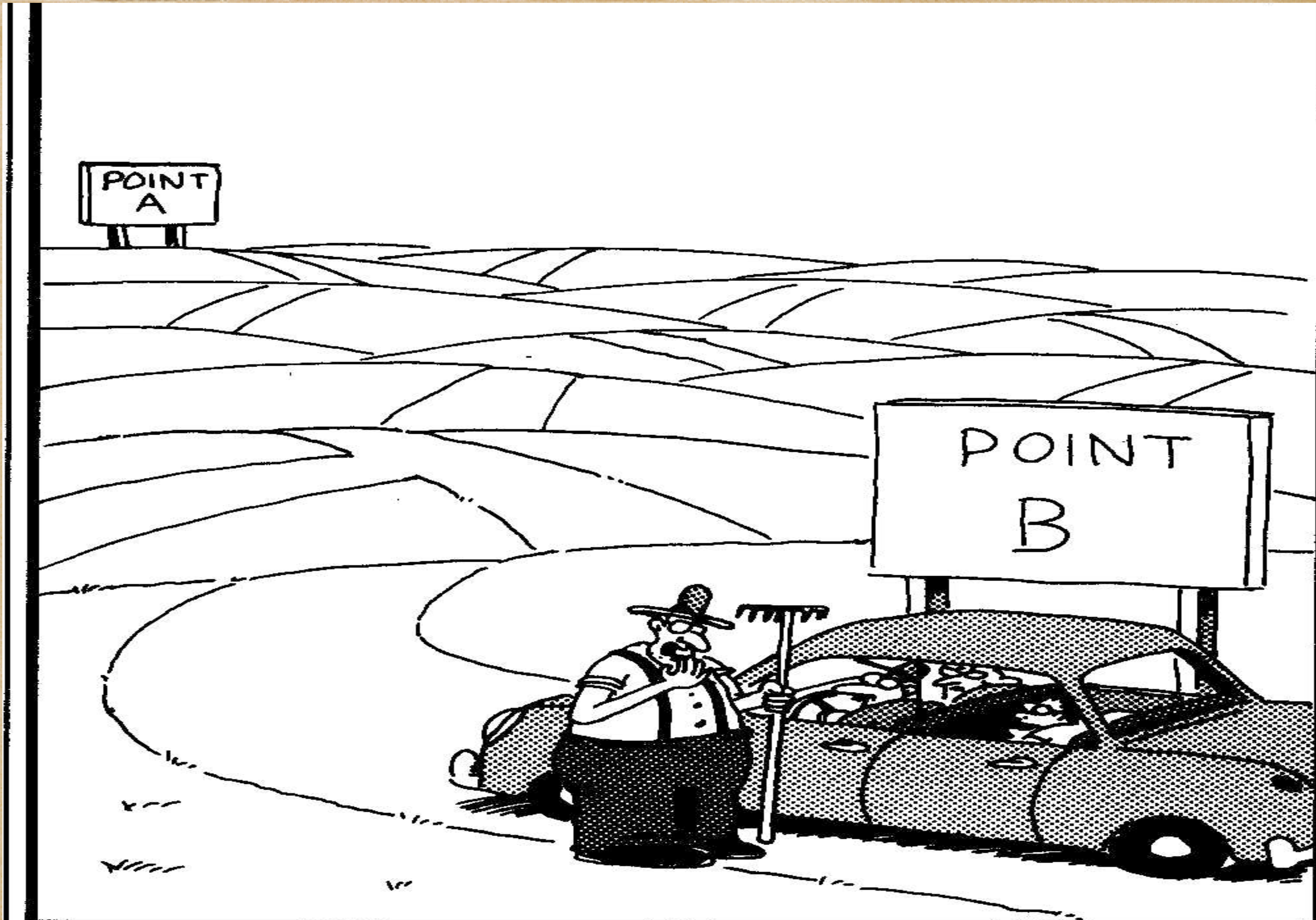

Pudding Mill Lane
Abbey Road
Bow Church
West Ham
under construction
Devons Road
Star Lane
Langdon Park
Canning Town
Beckton
Bank
Tower Gateway
Limehouse
All Saints
Royal Victoria
Prince Regent
Royal Albert
Cyprus
Gallions Reach
Westferry
Shadwell
Poplar
East India
Custom House
Beckton Park
West India Quay
Blackwall
Canary Wharf
Heron Quays
West Silvertown
Pontoon Dock
London City Airport
King George V
South Quay
Crossharbour
Mudchute
Island Gardens
Woolwich Arsenal
Cutty Sark
Greenwich
DeptfordBridge
Elverson Road


Town centre
Woolwich New Rd
National Rail
Bank
www.dlr.co.uk
DLR
95


Bank
© Copyright Design Triangle 20

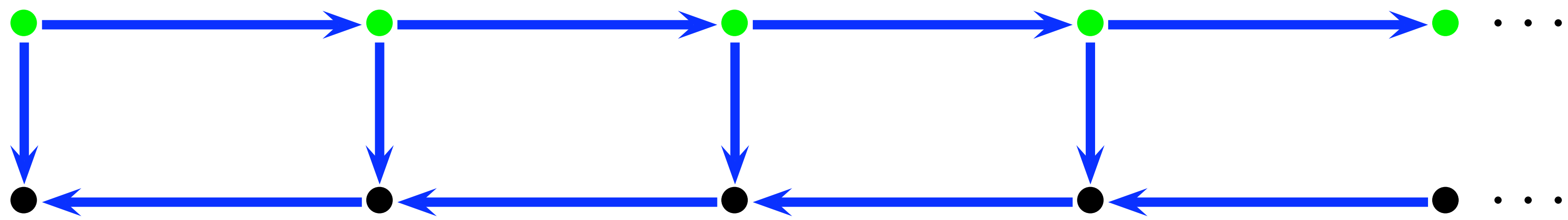
Color Code



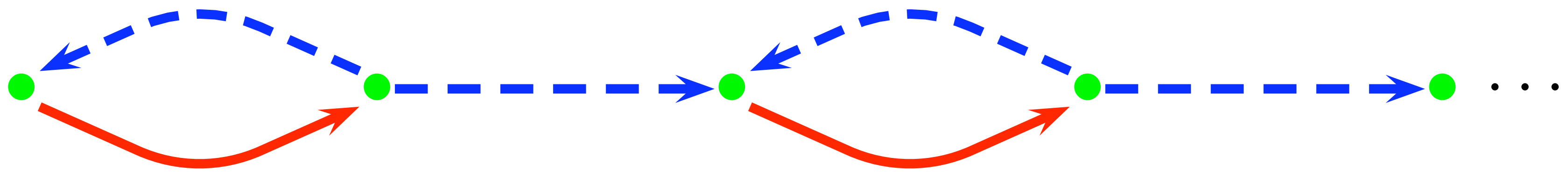


"Well, lemme think. . . . You've stumped me, son. Most folks only wanna know how to go the other way."

Mortal (black) nodes on bottom and immortal (green) nodes on top

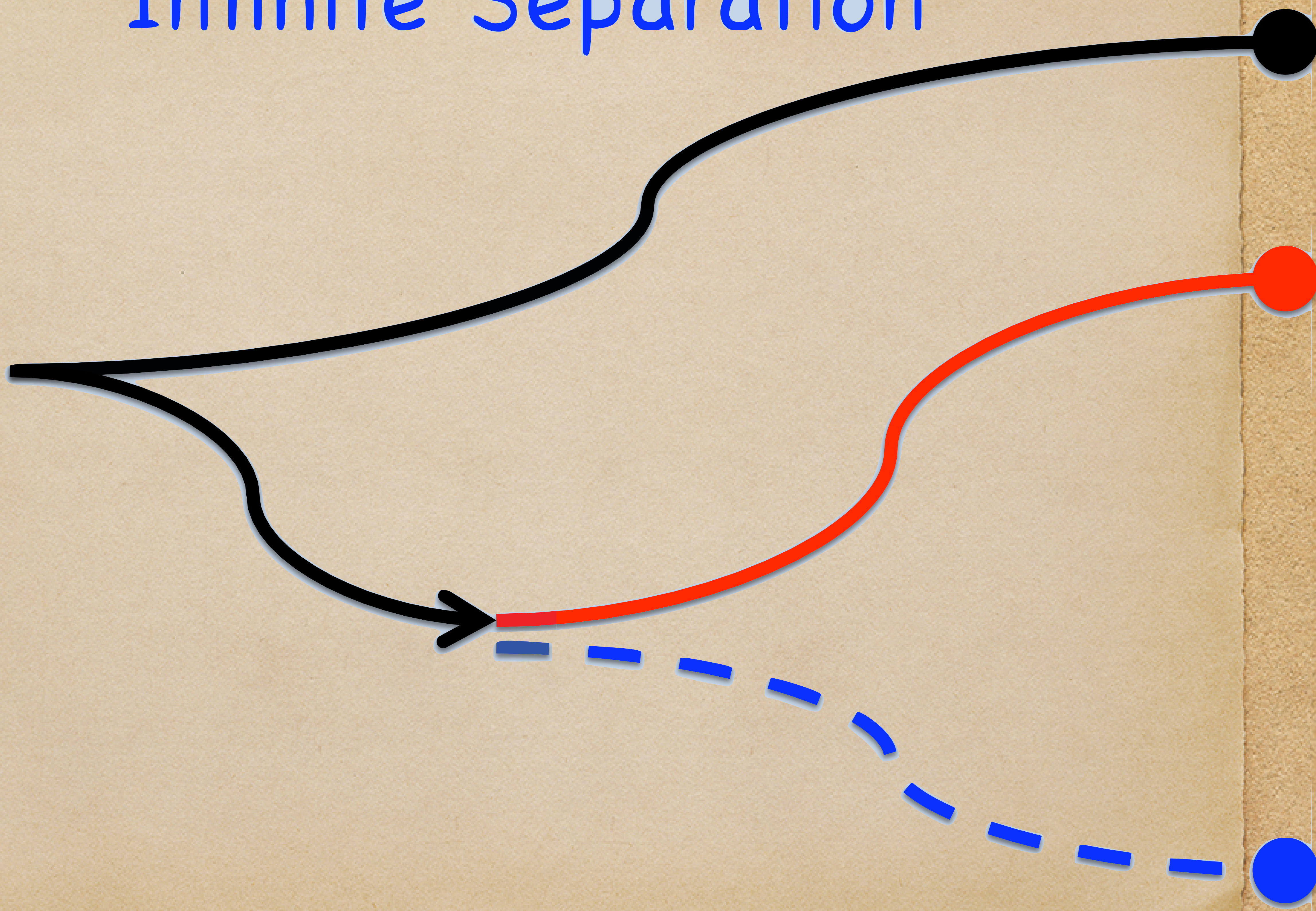


Mortal in each alone (dashed **Azure** or solid **Bordeaux**), but immortal in their union

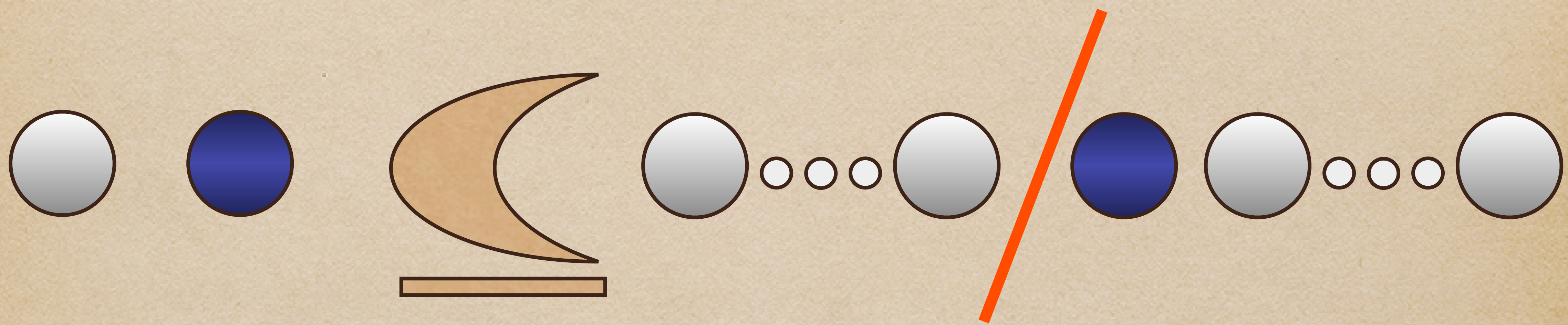


Infinite Separation

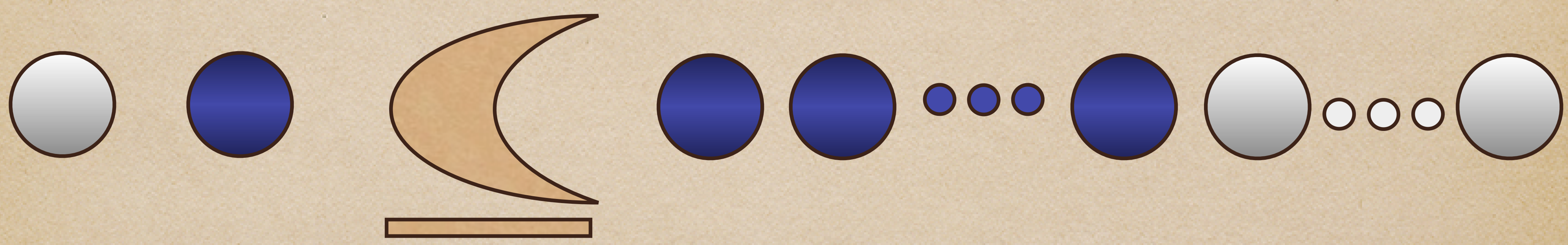
Infinite Separation



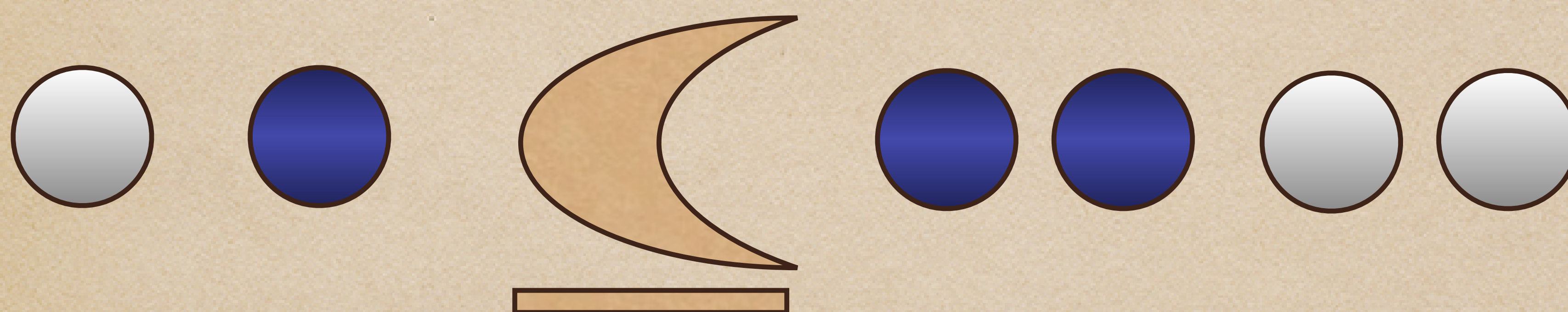
Enough?



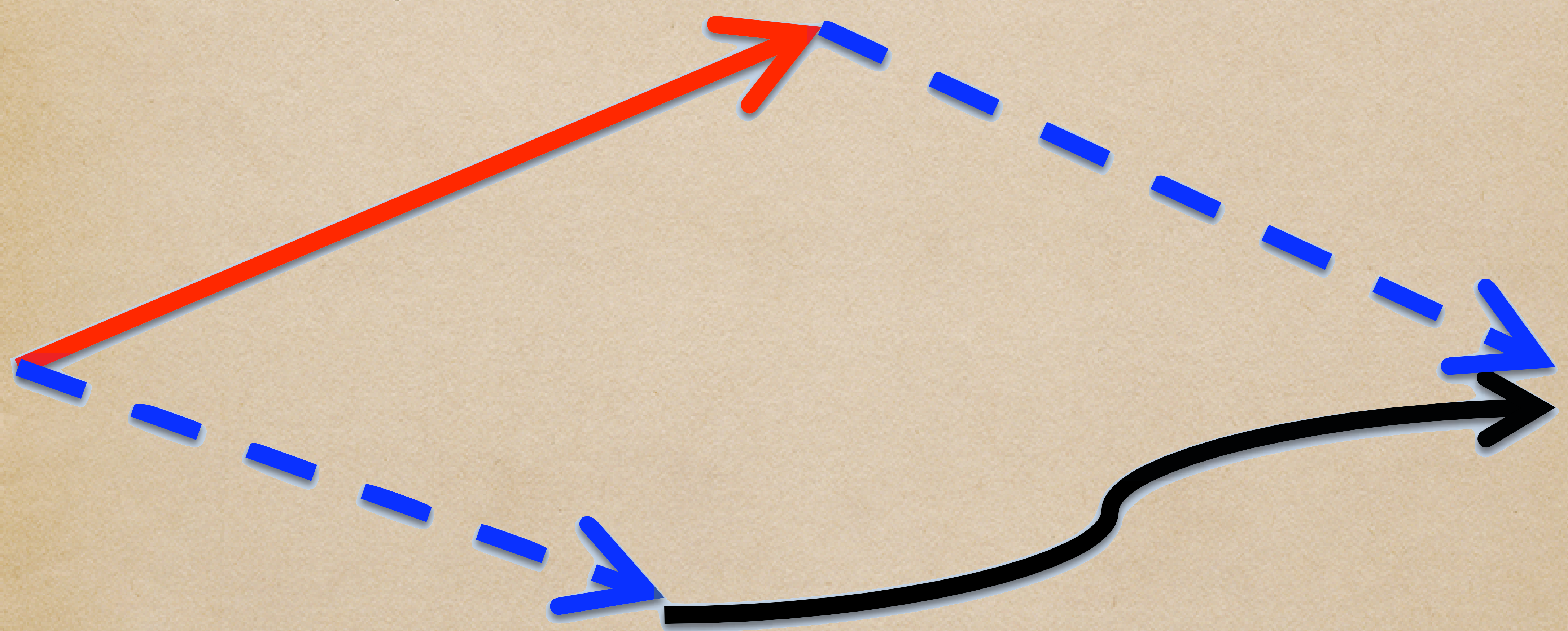
Enough?



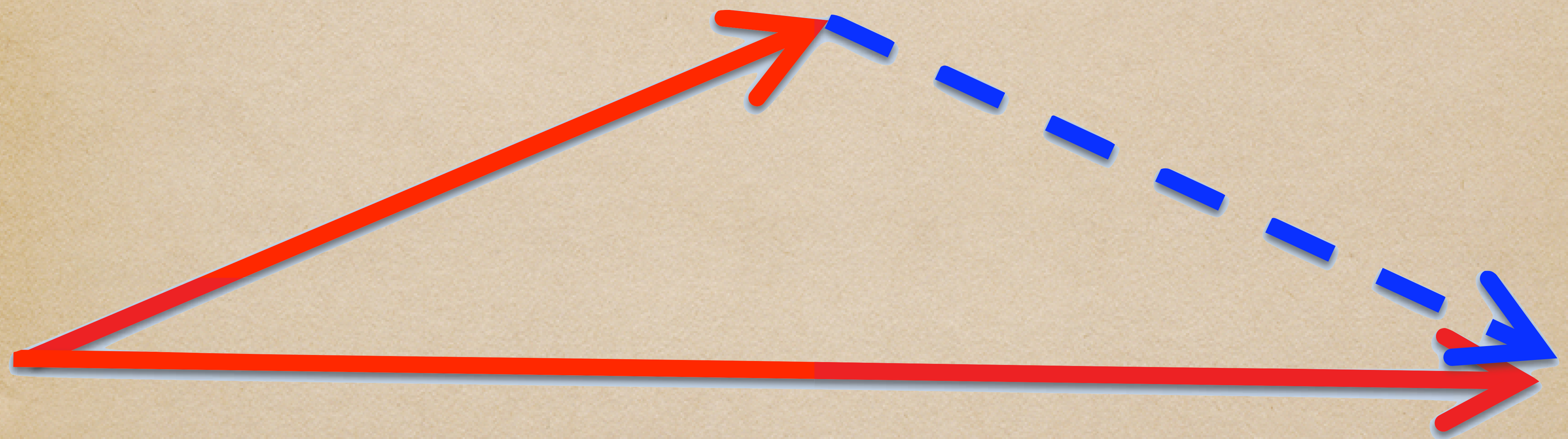
Enough?



Jumping

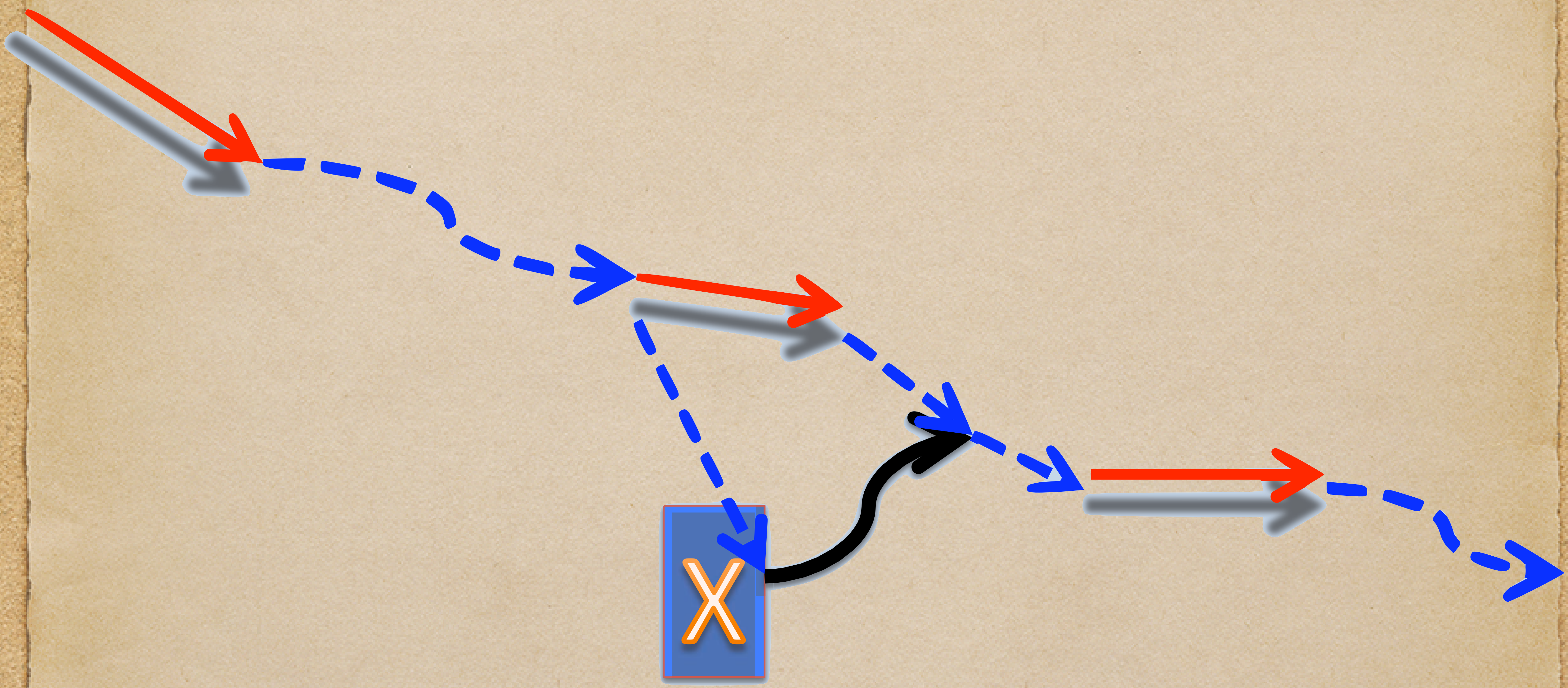


Jumping

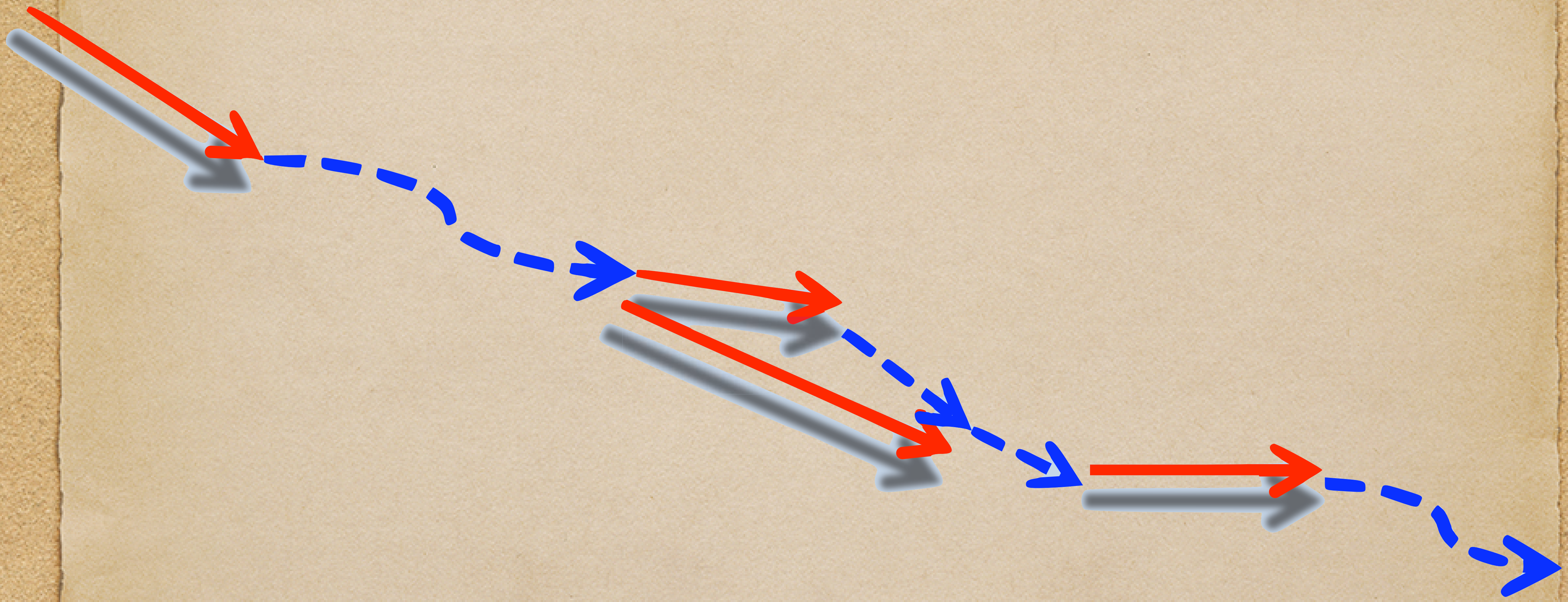



ONLY

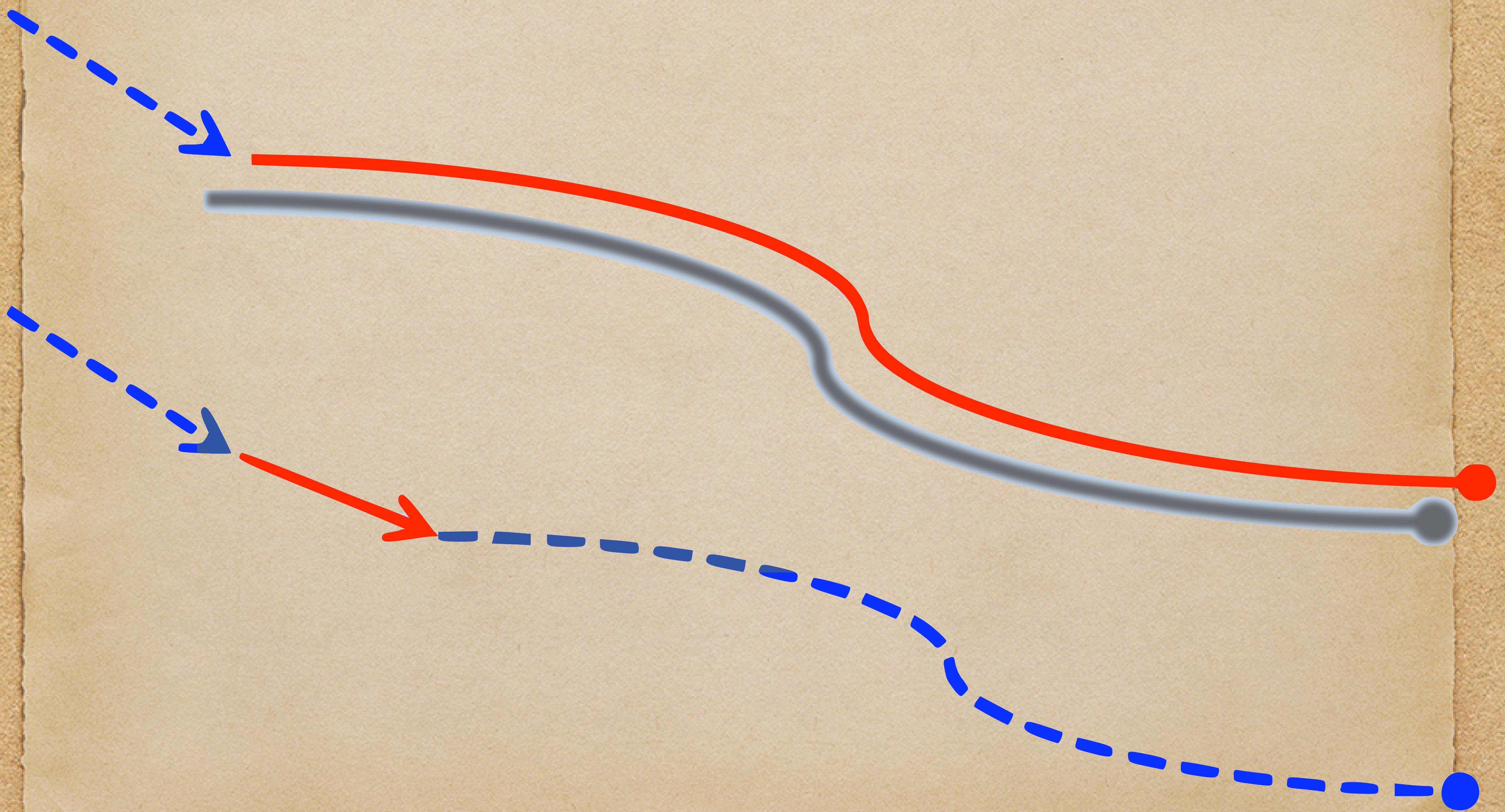
Constriction + Jumping



Constriction + Jumping



Constriction + Jumping



Gremlins

Gremlins