The ℵ-Calculus

a declarative model of reversible programming

Hannah Earley

h@nnah.io • DAMTP, Cambridge • Vaire Computing

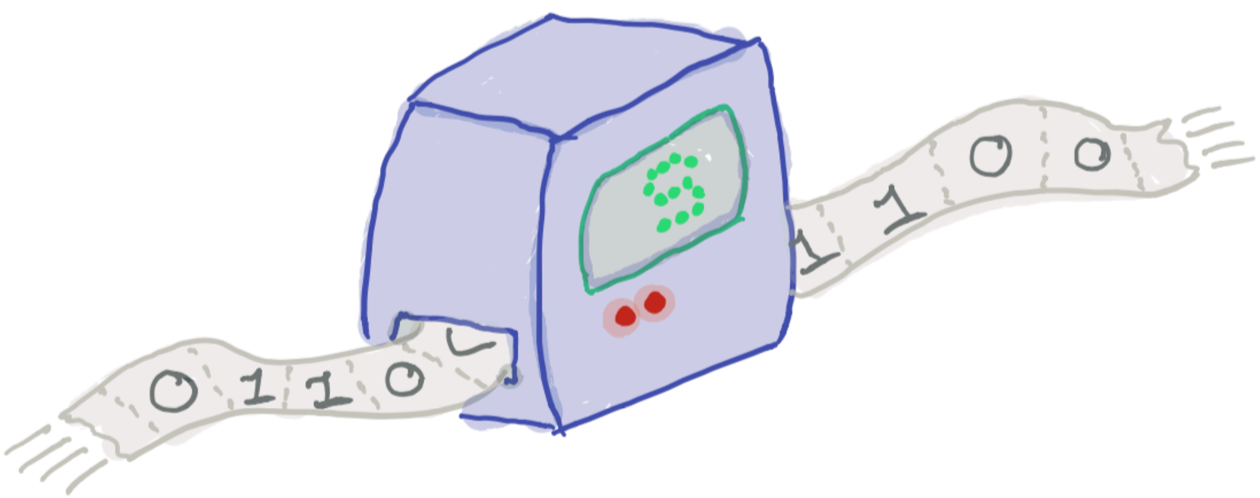
reversible programming

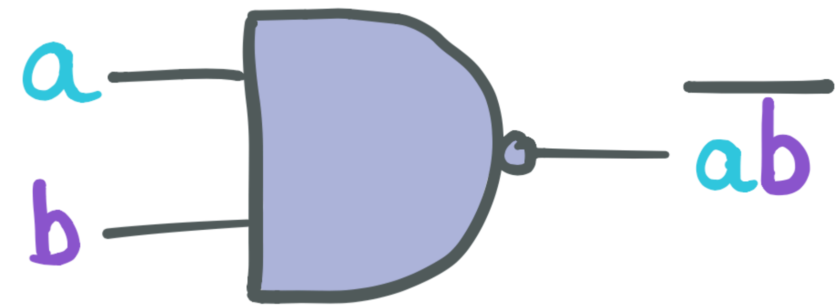
ℵ: motivation, semantics, & tutorial

ℵ: advanced features & properties

alethe + ℵ concurrency

Irreversible Computing





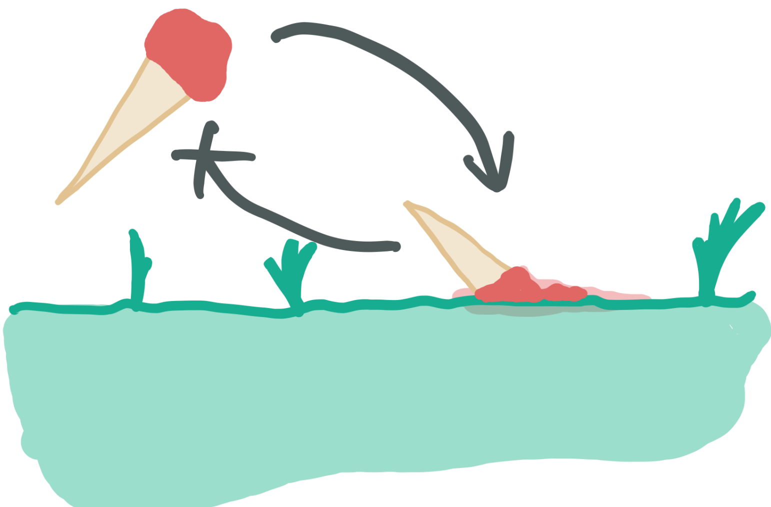
a	b	c
0	0	1
0	1	1
1	0	1
1	1	0

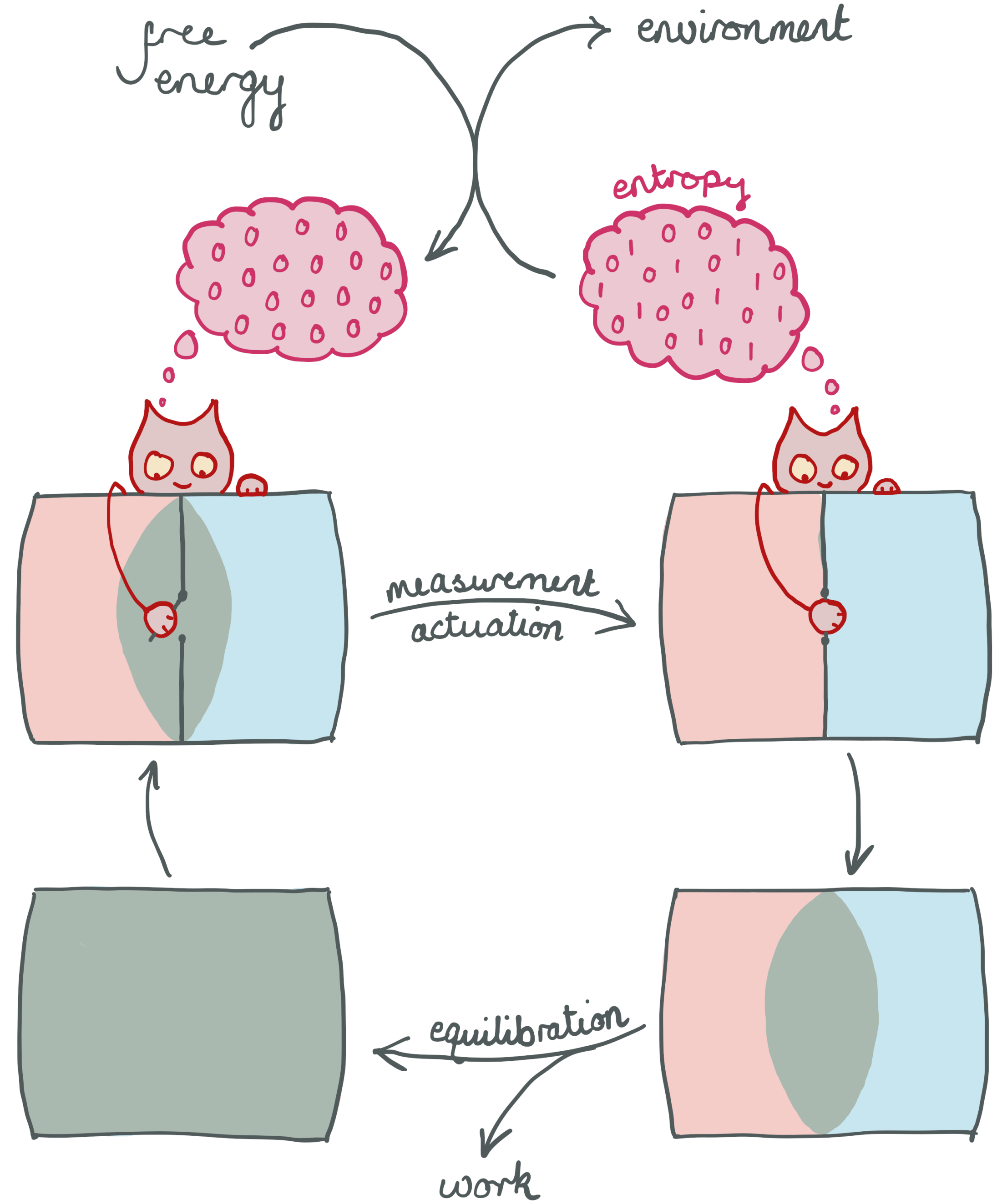
```
mov AL, 314
jne loop
```

```
sort :: Ord a ⇒ [a] → [a]
sort [] = []
sort (x:xs) =
  let le = sort [a | a ← xs, a ≤ x]
      gt = sort [a | a ← xs, a >  x]
  in  le ++ [x] ++ gt
```

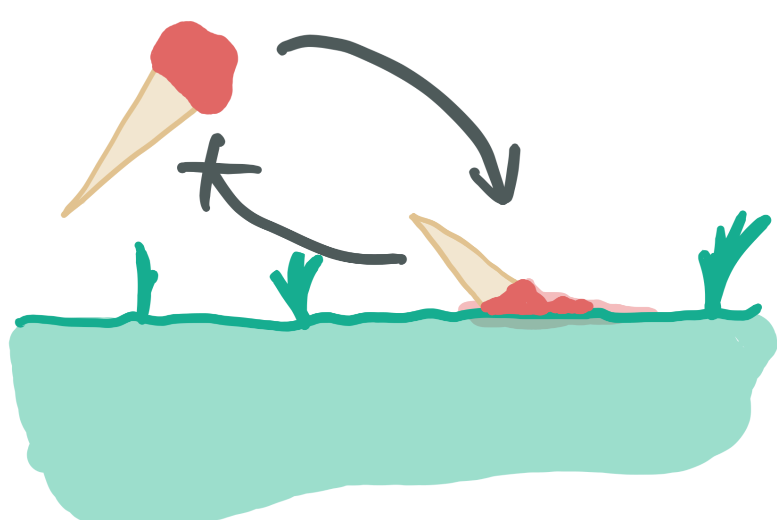
```
int i = 5;
while (i--) {
  printf("%d\n", i);
}
```

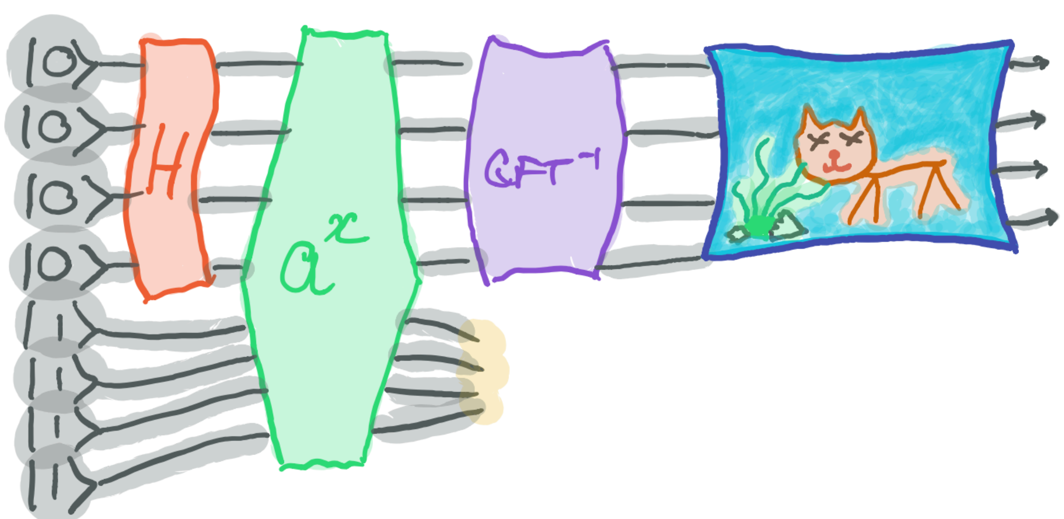
Invertibility?

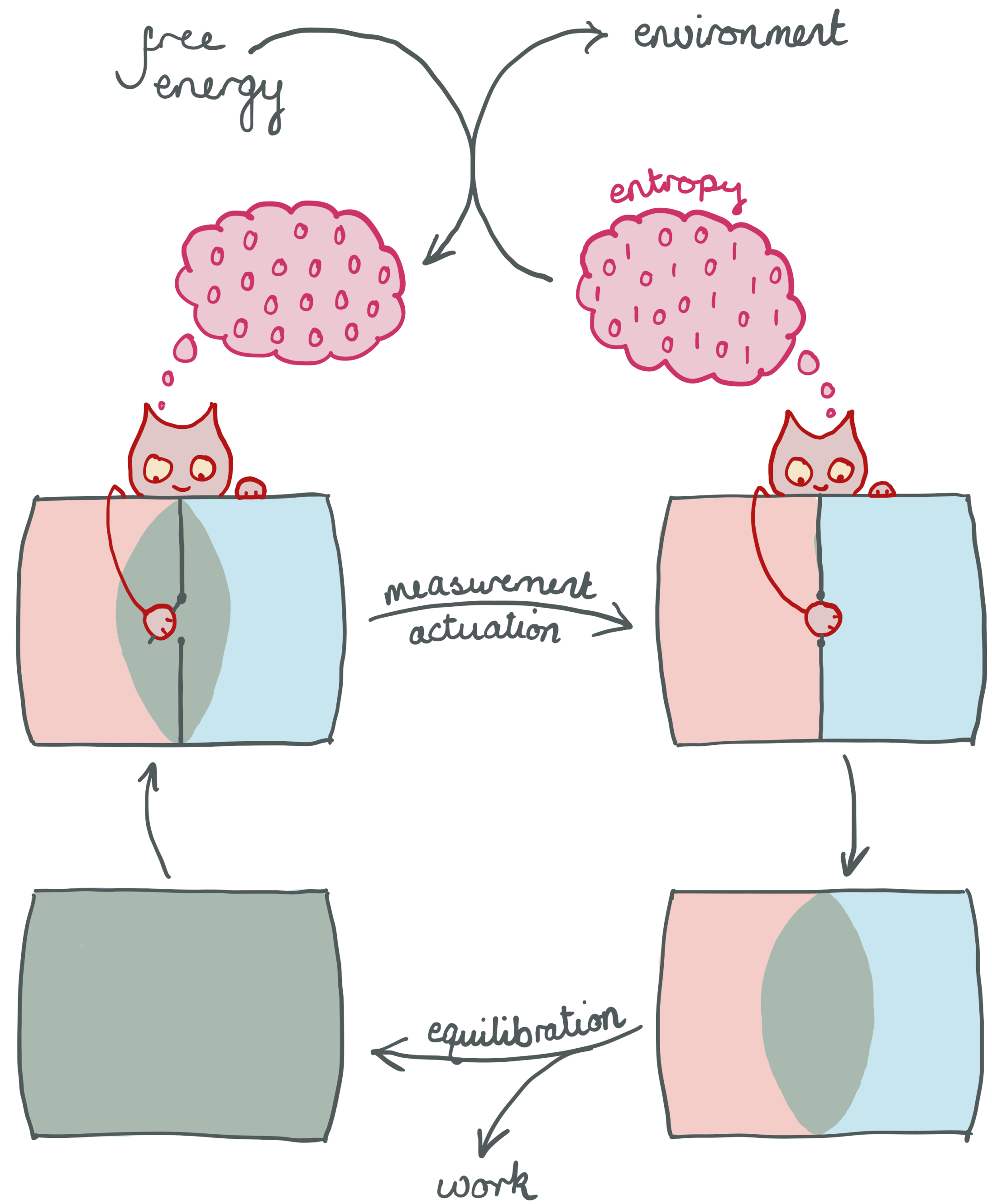




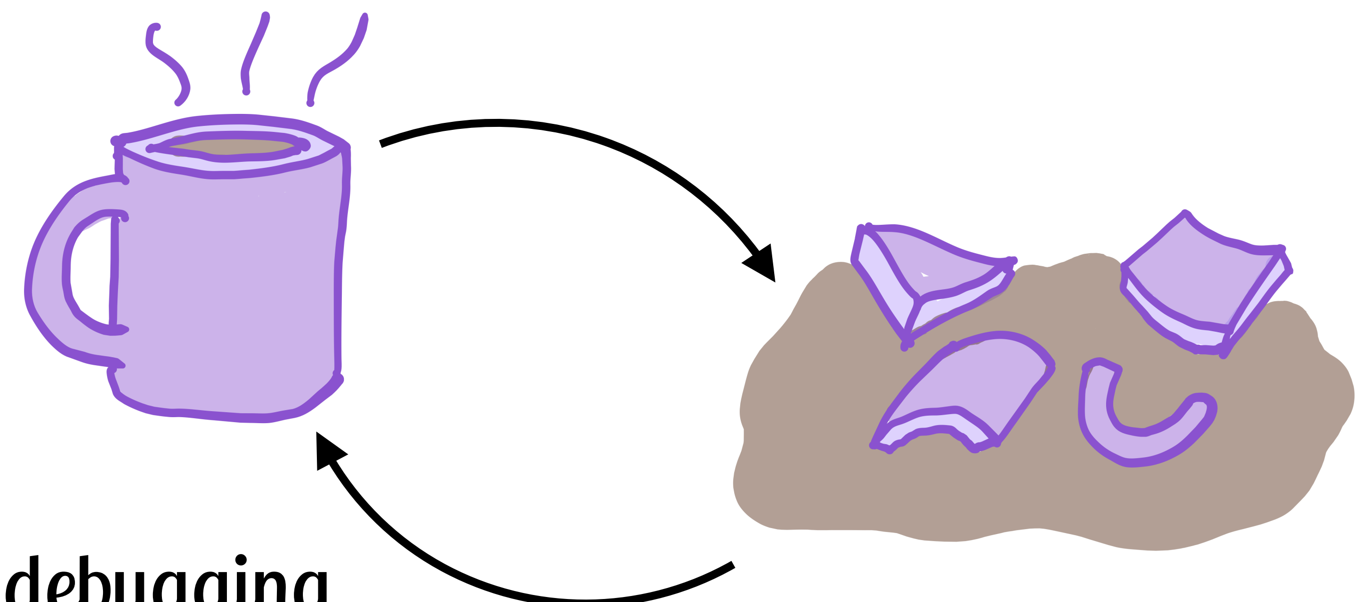
Invertibility?







Invertibility?



debugging

THE QUICK BROWN
FOX JUMPS OVER
THE LAZY DOG

dec ⟷ enc

QEB NRFZH YOLTK
CLU GRJMP LSBO
QEB IXWV ALD

```
{"fruit":[
  "apple",
  "banana",
  "cherry"]}
```

bidirectional transforms ⟷

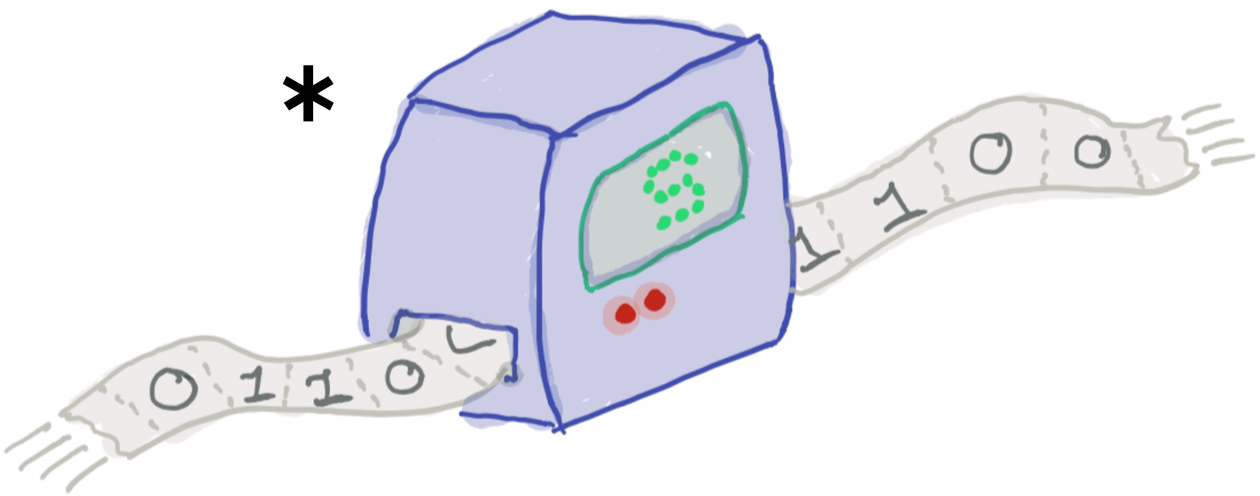
```
fruit:
- apple
- banana
- cherry
```

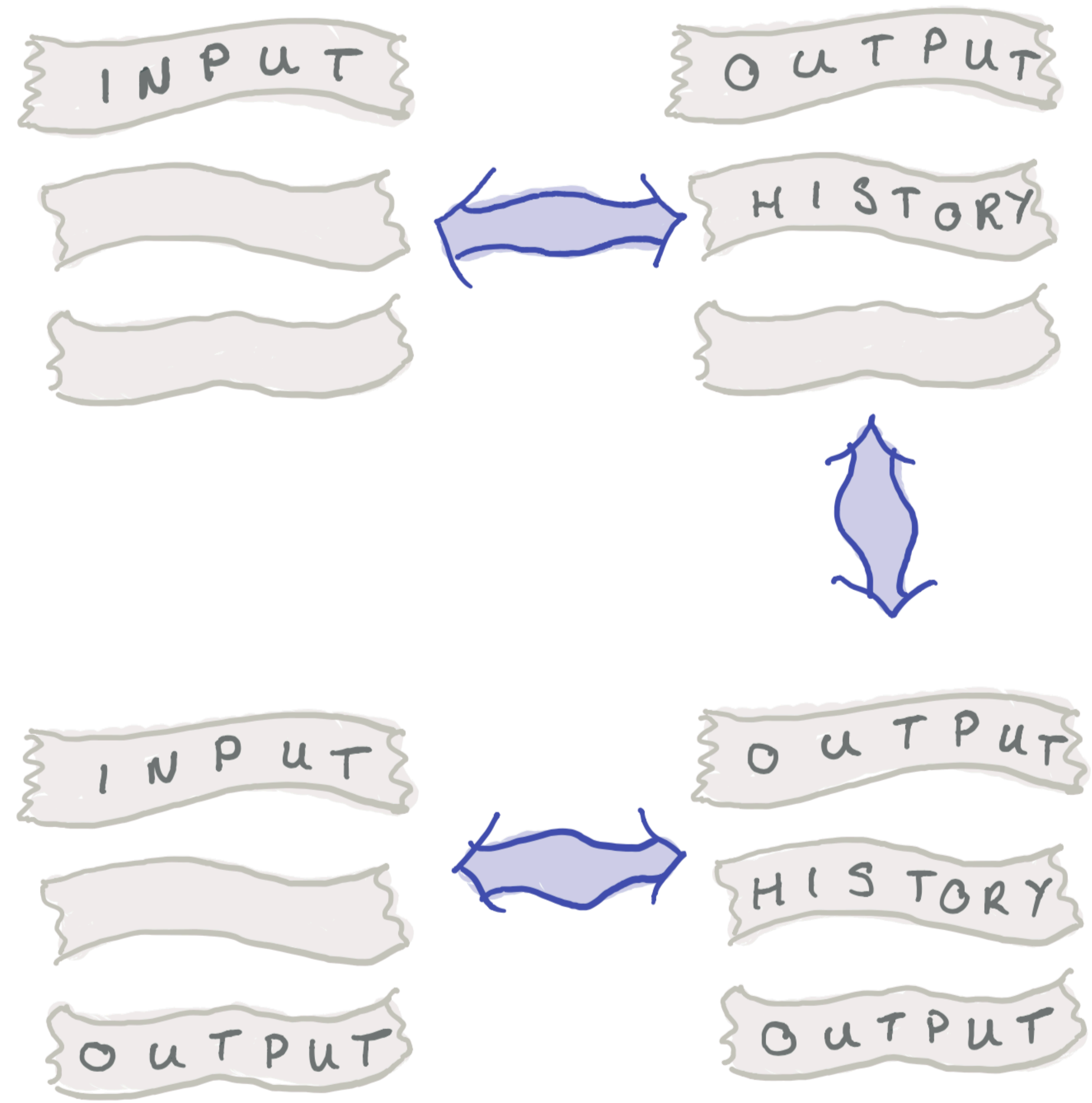
inverse programs

divmod ↕ muladd

Reversible Computing

*





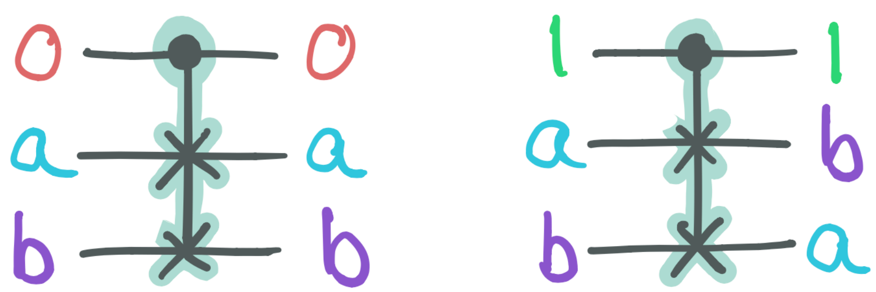
$$x \mapsto f(x)$$

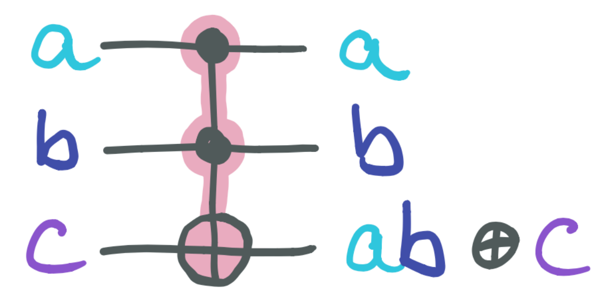
$$\Downarrow$$

$$x \leftrightarrow (x, f(x))$$

* but reversible!

Reversible Computing





Fredkin / CSWAP

0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

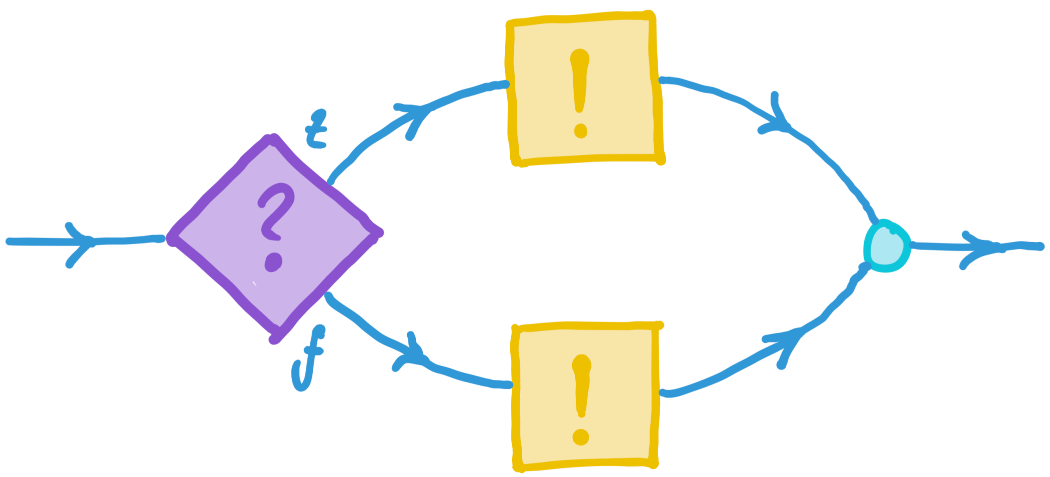
Toffoli / CCNOT

0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Reversible Computing

$n\ x[2]$

procedure *fib*
    if $n = 0$
    then $x_0 \mathrel{+}= 1$
        $x_1 \mathrel{+}= 1$
    else $n \mathrel{-}= 1$
        call *fib*
        $x_0 \mathrel{+}= x_1$
        $x_0 \Leftrightarrow x_1$
    fi $x_0 = x_1$



Reversible Computing

$n\ x[2]$

procedure *fib*

  if $n = 0$

  then $x_0 \mathrel{+}= 1$

    $x_1 \mathrel{+}= 1$

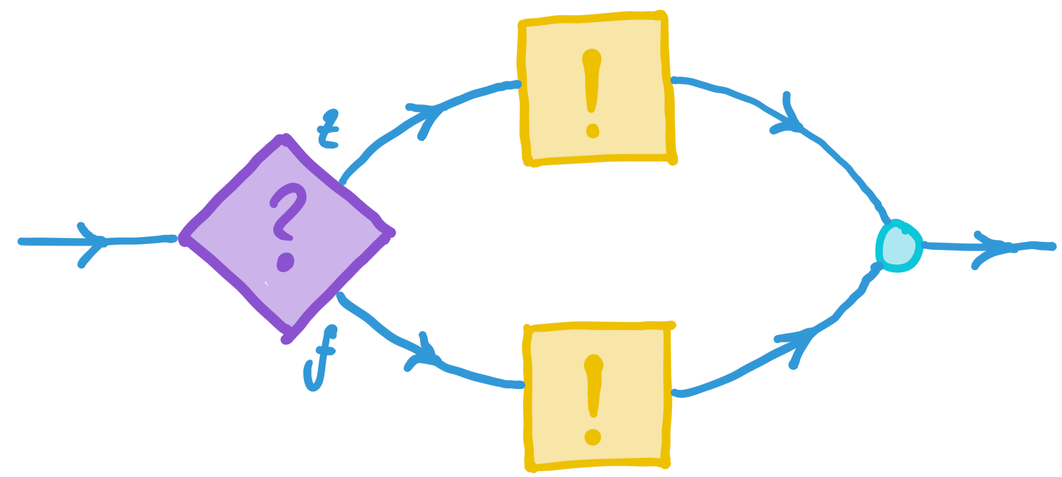
  else $n \mathrel{-}= 1$

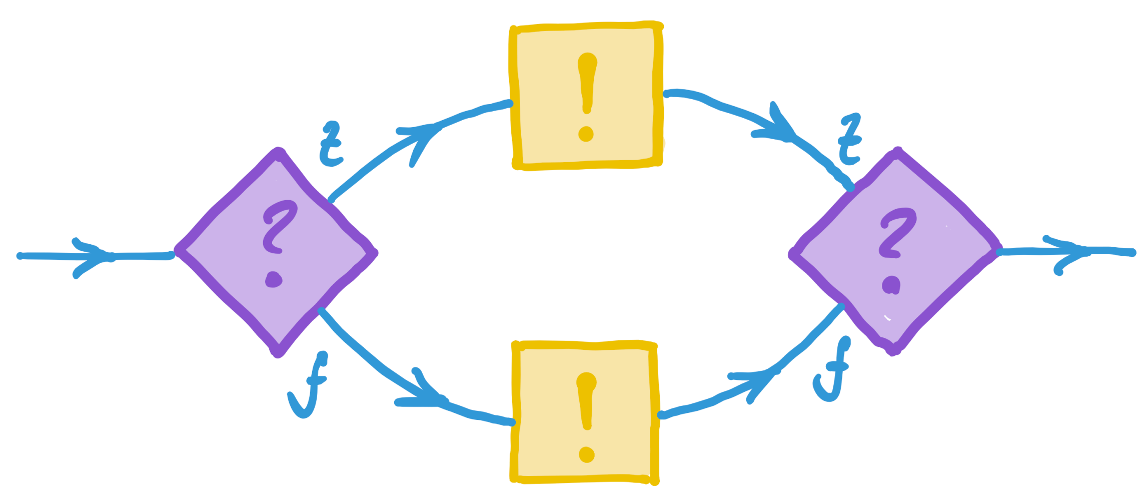
    call *fib*

    $x_0 \mathrel{+}= x_1$

    $x_0 \Leftrightarrow x_1$

  fi $x_0 = x_1$





Reversible Computing

$n\ m\ k$

procedure *fac*

  $m \mathrel{+}= 1$

  from $m = 1$

  loop $m \Leftrightarrow k$

    from $m = 0$

    loop $m \mathrel{+}= n$

      $k \mathrel{-}= 1$

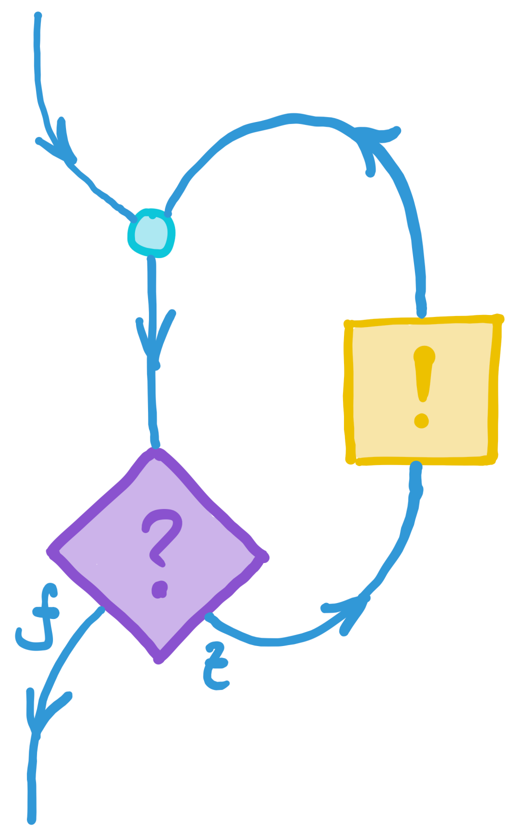
    until $k = 0$

    $n \mathrel{-}= 1$

  until $n = 1$

  $n \mathrel{-}= 1$

  $m \Leftrightarrow n$



Reversible Computing

$n\ m\ k$

procedure *fac*

$\quad m \mathrel{+}= 1$

$\quad$ **from** $m = 1$

$\quad$ **loop** $m \Leftrightarrow k$

$\quad\quad$ **from** $m = 0$

$\quad\quad$ **loop** $m \mathrel{+}= n$

$\quad\quad\quad k \mathrel{-}= 1$

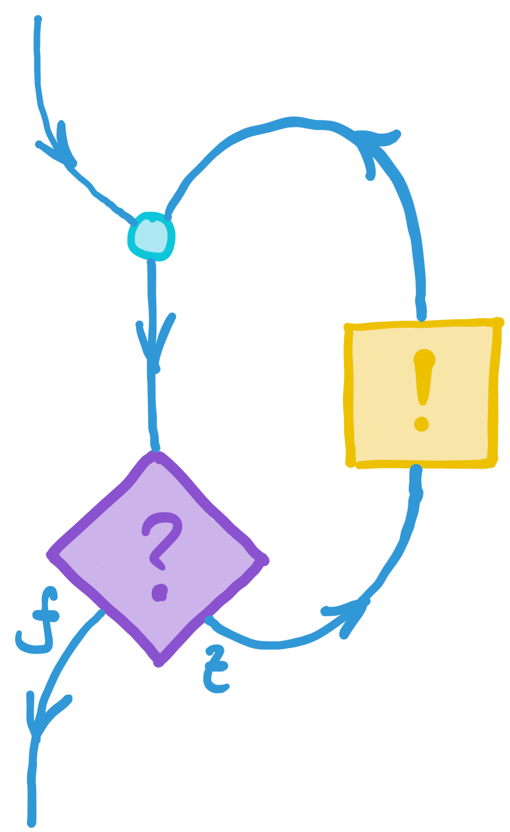
$\quad\quad$ **until** $k = 0$

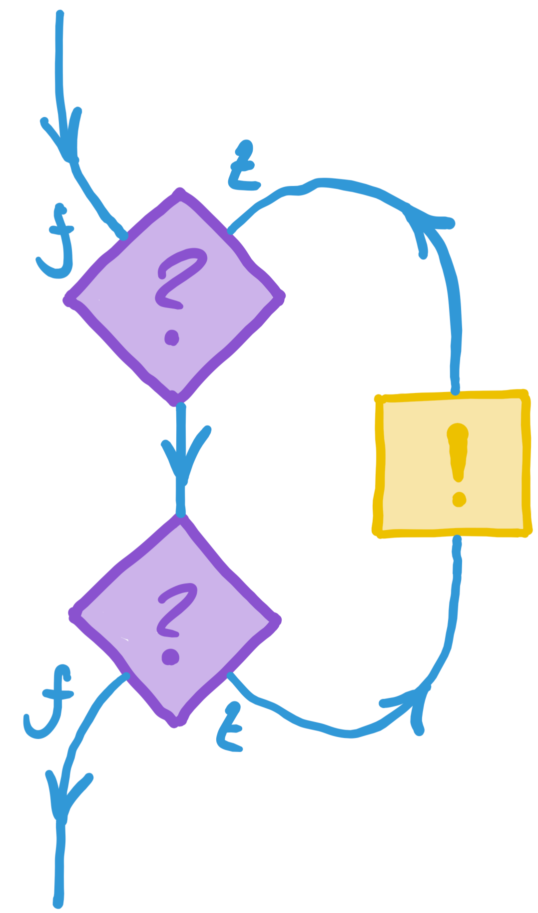
$\quad\quad n \mathrel{-}= 1$

$\quad$ **until** $n = 1$

$\quad n \mathrel{-}= 1$

$\quad m \Leftrightarrow n$





$$
\begin{aligned}
&\mathit{plus}\ \langle x, y\rangle \triangleq \textbf{case}\ y\ \textbf{of}\\
&\qquad Z \quad\ \ \to \lfloor\langle x\rangle\rfloor\\
&\qquad S(u) \to \textbf{let}\ \langle x', u'\rangle = \mathit{plus}\ \langle x, u\rangle\ \textbf{in}\ \langle x', S(u')\rangle\\
&\mathit{fib}\ n \triangleq \textbf{case}\ n\ \textbf{of}\\
&\qquad Z \quad\ \ \to \langle S(Z), S(Z)\rangle\\
&\qquad S(m) \to \textbf{let}\ \langle x, y\rangle = \mathit{fib}\ n\ \textbf{in}\\
&\qquad\qquad \textbf{let}\ z = \mathit{plus}\ \langle y, x\rangle\ \textbf{in}\ z
\end{aligned}
$$

RFUN

```
swap(a|b) {} (a|b)paws

add(a|b) {a[ add(a|b)bus z|b ]a} (a|b)bus

less-than(a|b|p) {a
    [b[ less-than(a|b|p)or-equal ]b] | [ p|p ]
|a} (a|b|p)or-equal

remquo(n|d|q) {
  less-than(d|n|p)or-equal
  p[ sub(n|d)dda remquo(n|d|q)ddalum ]q
} (n|d|q)ddalum
```

```
fact(n|d) {n[n[
    z|n z|d
    fact(n|d)orial'
    muladd(n|d|z)ouqmer
    swap(n|z)paws
    d|z n|z n|z
]n]n} (n|d)orial'

fact(n) {z|d
    fact(n|d)orial'
d|z} (n)orial
```

kayak

```
type a + b = InL a | InR b
type a × b = (a, b)
type ℕ = Z | S ℕ

trace :: f:(a + b ↔ a + c) → b ↔ c
  | -- definition omitted

addSub :: ℕ + ℕ ↔ ℕ + ℕ
  | InL (S n) ↔ InL n
  | InL Z     ↔ InR Z
  | InR n     ↔ InR (S n)

add₁ :: ℕ ↔ ℕ :: sub₁
  | n ↔ trace ~f:addSub n

add :: ℕ × ℕ ↔ ℕ × ℕ :: sub
  | a, b              ↔ iter $ a, Z, b
  | iter $ S a, a', b ↔ iter $ a, S a', add₁ b
  | iter $ Z, a, b    ↔ a, b
  where iter :: ℕ × ℕ × ℕ
```

```
cantorUnpair :: ℕ ↔ ℕ × ℕ :: cantorPair
  | n                   ↔ iter $ n, (Z, Z)
  | iter $ S n, (Z, b)   ↔ iter $ n, (S b, Z)
  | iter $ S n, (S a, b) ↔ iter $ n, (a, S b)
  | iter $ Z, (a, b)     ↔ (a, b)
  where iter :: ℕ × (ℕ × ℕ)

fib' :: (ℕ × ℕ) + ℕ ↔ (ℕ × ℕ) + (ℕ × ℕ)
  | InR n                  ↔ iter $ (n, (Z, Z))
  | iter $ (S n, (a, b))   ↔ iter' $ (n, add (b, a))
  | iter $ (Z, (a, b))     ↔ InR (add₁ a, add₁ b)
  | iter' $ (n, (a, b))    ↔ iter $ (n, (a, S b))
  | InL (n, (S a))         ↔ iter $ (n, (S a, Z))
  | InL (n, Z)             ↔ InL (cantorUnpair n)
  where iter :: ℕ × (ℕ × ℕ)
        iter' :: ℕ × (ℕ × ℕ)

fib :: ℕ ↔ ℕ × ℕ
  | n ↔ trace ~f:fib' n
```

theseus

$$
\begin{aligned}
&(\textbf{defsub}\ \mathit{pfunc}\ \ (\psi'\ \psi\ i\ \alpha\!\!\!/\ \varepsilon)\\
&\quad ((\psi'\ \_\ i) \mathrel{+}= ((\alpha\!\!\!/\ \_\ i) \times\!/\ (\psi\ \_\ i)))\\
&\quad ((\psi'\ \_\ i) \mathrel{-}= (\varepsilon \times\!/\ (\psi\ \_\ ((i+1) \wedge 127))))\\
&\quad ((\psi'\ \_\ i) \mathrel{-}= (\varepsilon \times\!/\ (\psi\ \_\ ((i-1) \wedge 127)))))\\
&(\textbf{defsub}\ \mathit{schstep}\ \ (\psi_R\ \psi_I\ \alpha\!\!\!/\ \varepsilon)\\
&\quad (\textbf{for}\ i = 0\ \textbf{to}\ 127\ (\textbf{call}\ \mathit{pfunc}\ \psi_R\ \psi_I\ i\ \alpha\!\!\!/\ \varepsilon))\\
&\quad (\textbf{for}\ i = 0\ \textbf{to}\ 127\ (\textbf{rcall}\ \mathit{pfunc}\ \psi_I\ \psi_R\ i\ \alpha\!\!\!/\ \varepsilon)))
\end{aligned}
$$

ЯR

reversible programming

א: motivation, semantics, & tutorial

א: advanced features & properties

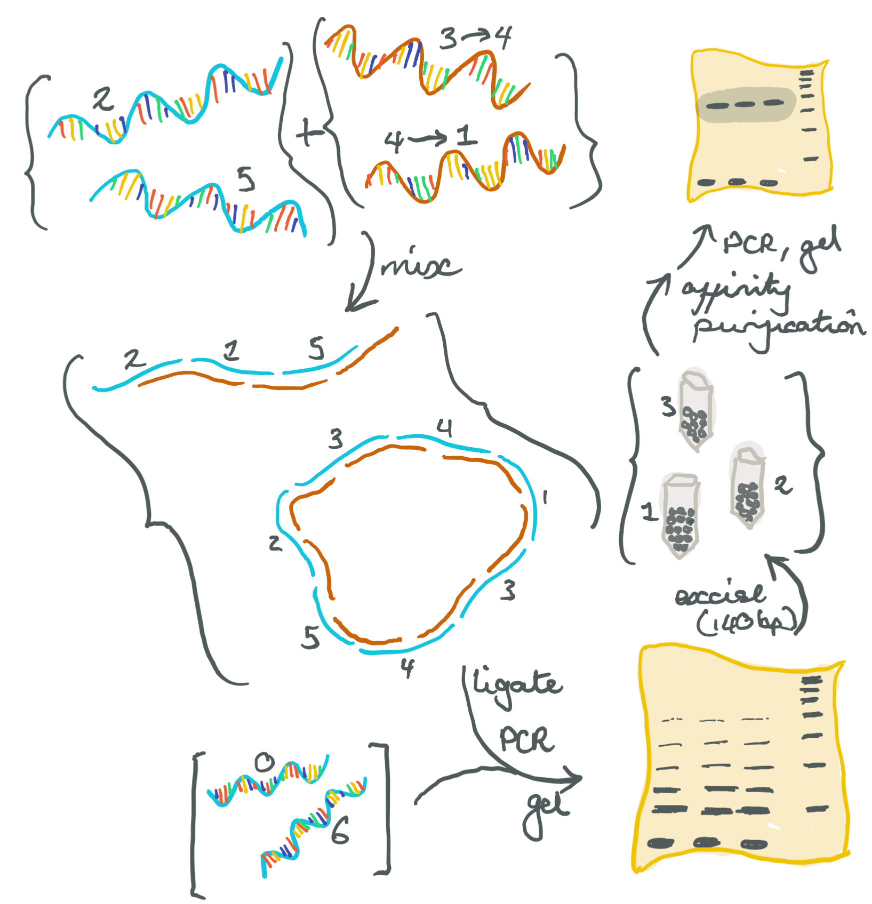
alethe + א concurrency

The ℵ-Calculus: Motivation

- λ-calculus inspiration
  - simple definition
  - reduction semantics
  - self-contained execution
- molecular programming

The ℵ-Calculus: Motivation

- λ-calculus inspiration
  - simple definition
  - reduction semantics
  - self-contained execution
- **molecular programming**



Attempt 1: The Σ-Calculus

$$\tau ::= \langle \pi\tau^* : \tau^*\pi \rangle \mid \text{VAR} \mid \text{REF} \mid \tau *$$

definition

$$\text{CONS} \equiv \langle \pi\{nc\}zf : c\{nf\}z\pi \rangle$$
$$\text{NIL} \equiv \langle \pi\{nc\}zf : n\{fc\}z\pi \rangle$$

Church encoding

$$\text{REV} \equiv \langle \pi l \top : \text{NIL}\{\lambda\nu\}\{\{\varepsilon\varepsilon\}l\}\pi \rangle$$
$$\lambda \equiv \langle \pi\{\text{REV}\,\nu\}\{\{r'r''\}\{\tilde{l}l'l''\}\}\tilde{r} : \tilde{l}\{\text{REV}'\,\nu\}\{\{\tilde{r}r'r''\}\{l'l''\}\}\pi \rangle$$
$$\nu \equiv \langle \pi\{\text{REV}'\,\lambda\}\{r\{l'l''\}\}\text{CONS} : \text{CONS}\{\text{REV}\,\lambda\}\{\{l'r\}l''\}\pi \rangle$$
$$\text{REV}' \equiv \langle \pi\{\lambda\nu\}\{r\{\varepsilon\varepsilon\}\}\text{NIL} : \bot r\pi \rangle$$

list reversal

Hannah Earley — 2017 — sigma Interpreter — https://github.com/hannah-earley/sigma-repl
— https://github.com/hannah-earley/sigma-examples

Attempt 2: The $\aleph$-Calculus

- declarative
- reversible TRS semantics, without history
- minimalistic definition

$$
\begin{array}{rl}
(\text{PATTERN TERM}) & \pi ::= \text{SYM} \mid \text{VAR} \mid (\,\pi^*\,) \\
(\text{RULE}) & \rho ::= \pi^* = \pi^* \\
(\text{DEFINITION}) & \delta ::= \rho : \rho^*_. \mid \mathbf{!}\ \pi^*;
\end{array}
$$

Addition

$$
\begin{aligned}
&\mathbf{!}\ +\ a\ b\ ();\quad \mathbf{!}\ ()\ c\ b\ +;\\
&\quad +\ a\ \mathrm{Z}\ () = ()\ a\ \mathrm{Z}\ +; && (\textsc{add–base})\\
&\quad +\ a\ (\mathrm{S}b)\ () = ()\ (\mathrm{S}c)\ (\mathrm{S}b)\ +: && (\textsc{add–step})\\
&\qquad\quad +\ a\ b\ () = ()\ c\ b\ +. && (\textsc{add–step–sub})
\end{aligned}
$$

$$
\begin{aligned}
(\textsc{pattern term}) \quad & \pi ::= \textsc{sym} \mid \textsc{var} \mid (\ \pi^*\ )\\
(\textsc{rule}) \quad & \rho ::= \pi^* = \pi^*\\
(\textsc{definition}) \quad & \delta ::= \rho : \rho.^* \mid \mathbf{!}\ \pi^*;
\end{aligned}
$$

Addition

$$\textbf{!}\ +\ a\ b\ ();\quad \textbf{!}\ ()\ c\ b\ +;$$

$$+\ a\ \mathrm{Z}\ () = ()\ a\ \mathrm{Z}\ +; \qquad (\textsc{add–base})$$

$$+\ a\ (\mathrm{S}b)\ () = ()\ (\mathrm{S}c)\ (\mathrm{S}b)\ +: \qquad (\textsc{add–step})$$

$$+\ a\ b\ () = ()\ c\ b\ +. \qquad (\textsc{add–step–sub})$$

$$\textbf{!}\quad +\ 3\ 2\ () \qquad (\textsc{add–step})$$

Addition

$$
\begin{aligned}
&\mathbf{!}\ + a\ b\ ();\quad \mathbf{!}\ ()\ c\ b\ +; && \\
&\quad + a\ \mathrm{Z}\ () = ()\ a\ \mathrm{Z}\ +; && (\textsc{add–base}) \\
&\quad + a\ (\mathrm{S}b)\ () = ()\ (\mathrm{S}c)\ (\mathrm{S}b)\ +: && (\textsc{add–step}) \\
&\qquad\quad + a\ b\ () = ()\ c\ b\ +. && (\textsc{add–step–sub})
\end{aligned}
$$

$$
\begin{aligned}
&\mathbf{!}\quad + 3\ 2\ () \leftrightsquigarrow \underline{\{a \mapsto 3, b \mapsto 1\}} && (\textsc{add–step}) \\
& && (\textsc{add–step–sub}) \\
&\quad \overline{+ 3\ 1\ ()} && (\textsc{add–step})
\end{aligned}
$$

Addition

$$\begin{aligned}
&\textbf{!}\ + a\ b\ ();\quad \textbf{!}\ ()\ c\ b\ +; \\
&\quad + a\ \mathrm{Z}\ () = ()\ a\ \mathrm{Z}\ +; && (\textsc{add–base}) \\
&\quad + a\ (\mathrm{S}b)\ () = ()\ (\mathrm{S}c)\ (\mathrm{S}b)\ +: && (\textsc{add–step}) \\
&\qquad\quad + a\ b\ () = ()\ c\ b\ +. && (\textsc{add–step–sub})
\end{aligned}$$

$$\begin{array}{lll}
\textbf{!}\quad + 3\ 2\ () \leftrightsquigarrow \underline{\{a \mapsto 3, b \mapsto 1\}} & & (\textsc{add–step}) \\
 & & (\textsc{add–step–sub}) \\
\quad \overline{+ 3\ 1\ ()} \leftrightsquigarrow \underline{\{a \mapsto 3, b \mapsto 0\}} & & (\textsc{add–step}) \\
 & & (\textsc{add–step–sub}) \\
\qquad\qquad \overline{+ 3\ \mathrm{Z}\ ()} & &
\end{array}$$

Addition

$$
\begin{aligned}
&\textbf{!}\ + \ a\ b\ ();\quad \textbf{!}\ ()\ c\ b\ +; \\
&\quad + \ a\ \mathrm{Z}\ () = ()\ a\ \mathrm{Z}\ +; && (\textsc{add–base}) \\
&\quad + \ a\ (\mathrm{S}b)\ () = ()\ (\mathrm{S}c)\ (\mathrm{S}b)\ +: && (\textsc{add–step}) \\
&\qquad\quad + \ a\ b\ () = ()\ c\ b\ +. && (\textsc{add–step–sub})
\end{aligned}
$$

$$
\begin{aligned}
&\textbf{!}\quad + \ 3\ 2\ () \leftrightsquigarrow \underline{\{a \mapsto 3, b \mapsto 1\}} && (\textsc{add–step}) \\
& && (\textsc{add–step–sub}) \\
&\quad \overline{+ \ 3\ 1\ ()} \leftrightsquigarrow \underline{\{a \mapsto 3, b \mapsto 0\}} && (\textsc{add–step}) \\
& && (\textsc{add–step–sub}) \\
&\qquad\quad \overline{+ \ 3\ \mathrm{Z}\ ()} \leftrightsquigarrow \{a \mapsto 3\} \leftrightsquigarrow ()\ 3\ \mathrm{Z}\ + && (\textsc{add–base})
\end{aligned}
$$

Addition

$$
\begin{array}{ll}
\textbf{!}\ +\ a\ b\ ();\quad \textbf{!}\ ()\ c\ b\ +; & \\
\quad +\ a\ \mathrm{Z}\ () = ()\ a\ \mathrm{Z}\ +; & (\textsc{add–base}) \\
\quad +\ a\ (\mathrm{S}b)\ () = ()\ (\mathrm{S}c)\ (\mathrm{S}b)\ +: & (\textsc{add–step}) \\
\qquad\quad +\ a\ b\ () = ()\ c\ b\ +. & (\textsc{add–step–sub})
\end{array}
$$

$$
\begin{array}{lllll}
\textbf{!} & +\ 3\ 2\ () \leftrightsquigarrow \underline{\{a \mapsto 3, b \mapsto 1\}} & & & (\textsc{add–step}) \\
 & & & & (\textsc{add–step–sub}) \\
 & \overline{+\ 3\ 1\ ()} \leftrightsquigarrow \underline{\{a \mapsto 3, b \mapsto 0\}} & & \underline{\{c \mapsto 3, b \mapsto 0\}} \leftrightsquigarrow ()\ 4\ 1\ + & (\textsc{add–step}) \\
 & & & & (\textsc{add–step–sub}) \\
 & \qquad \overline{+\ 3\ \mathrm{Z}\ ()} \leftrightsquigarrow \{a \mapsto 3\} \leftrightsquigarrow \overline{()\ 3\ \mathrm{Z}\ +} & & & (\textsc{add–base})
\end{array}
$$

Addition

$$\mathbf{!}\ + a\ b\ ();\quad \mathbf{!}\ ()\ c\ b\ +;$$

$$+ a\ \mathrm{Z}\ () = ()\ a\ \mathrm{Z}\ +; \qquad (\textsc{add–base})$$

$$+ a\ (\mathrm{S}b)\ () = ()\ (\mathrm{S}c)\ (\mathrm{S}b)\ +: \qquad (\textsc{add–step})$$

$$+ a\ b\ () = ()\ c\ b\ +. \qquad (\textsc{add–step–sub})$$

$$\mathbf{!}\quad + 3\ 2\ () \leftrightsquigarrow \underline{\{a \mapsto 3, b \mapsto 1\}} \qquad \underline{\{c \mapsto 4, b \mapsto 1\}} \leftrightsquigarrow ()\ 5\ 2\ + \quad \mathbf{!} \qquad (\textsc{add–step})$$

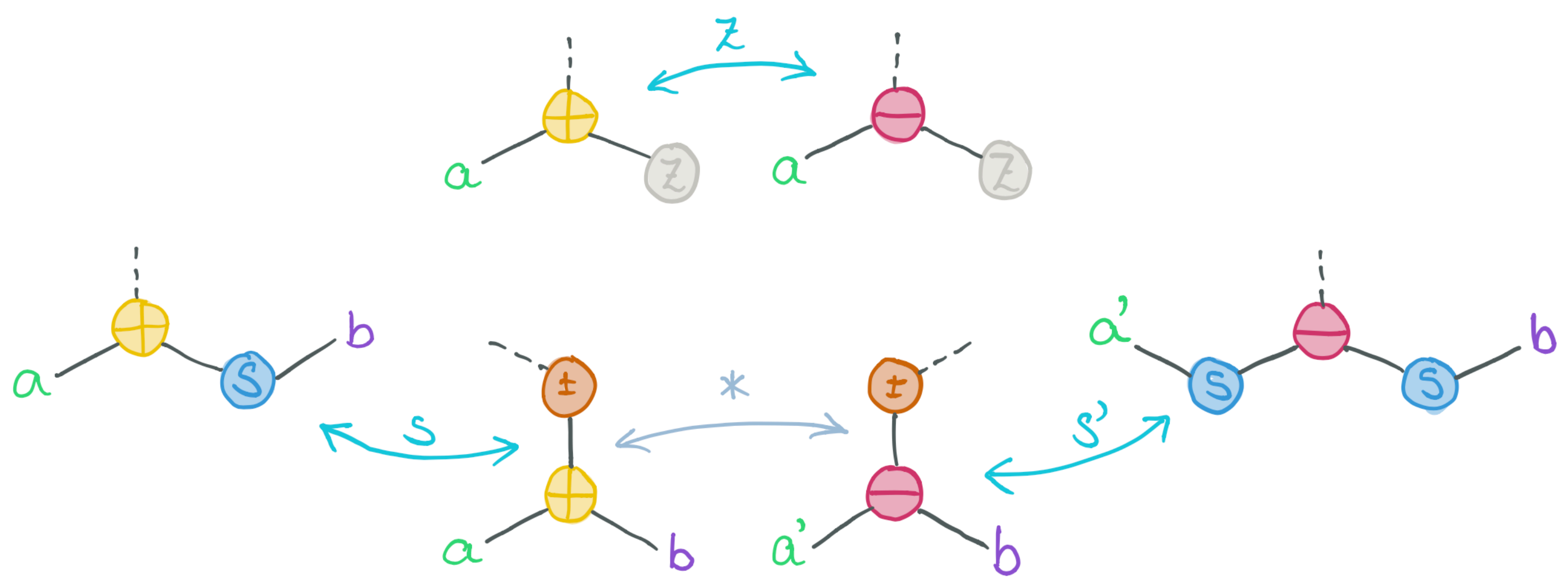
$$(\textsc{add–step–sub})$$

$$\overline{+ 3\ 1\ ()} \leftrightsquigarrow \underline{\{a \mapsto 3, b \mapsto 0\}} \qquad \underline{\{c \mapsto 3, b \mapsto 0\}} \leftrightsquigarrow \overline{()\ 4\ 1\ +} \qquad (\textsc{add–step})$$

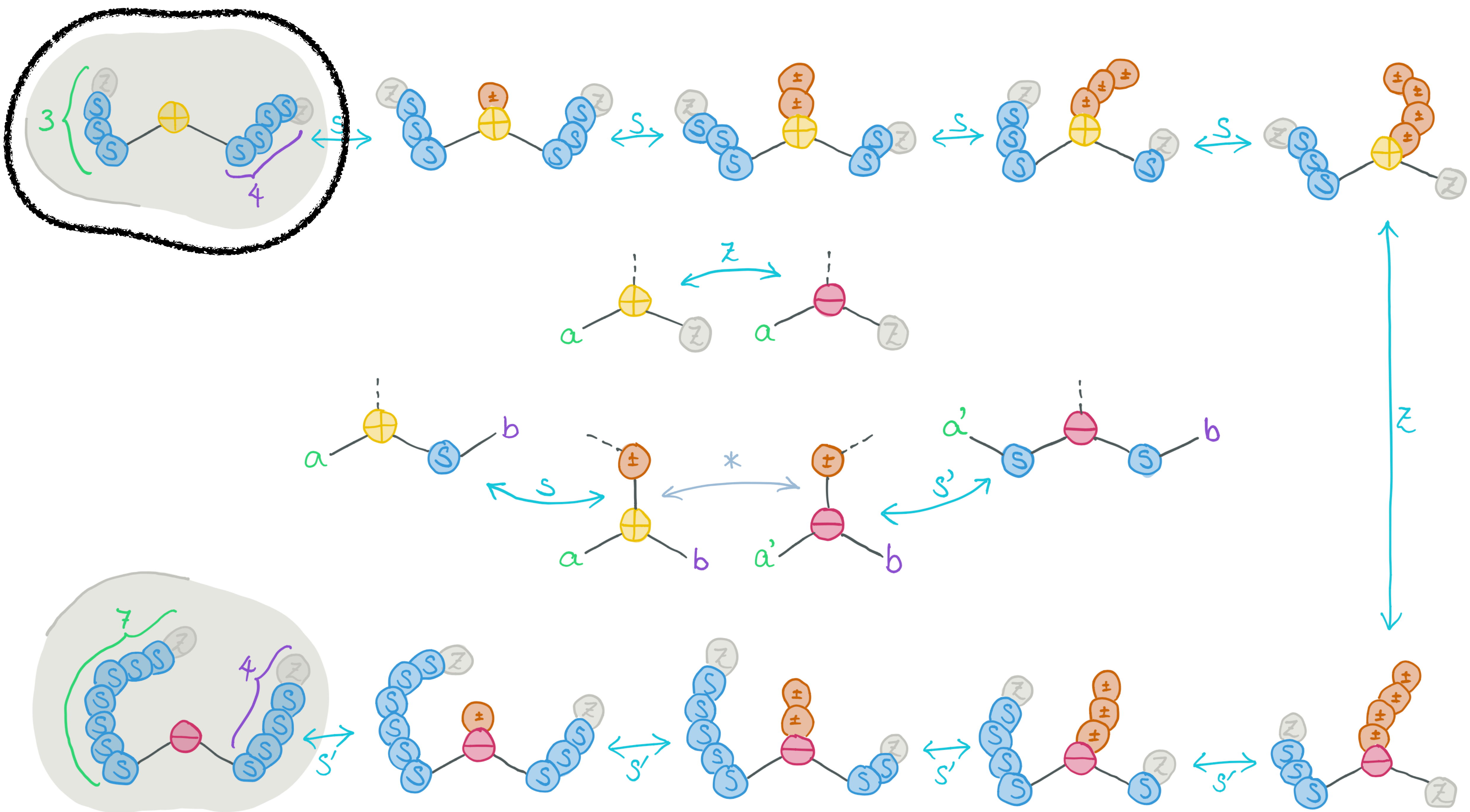
$$(\textsc{add–step–sub})$$

$$\overline{+ 3\ \mathrm{Z}\ ()} \leftrightsquigarrow \{a \mapsto 3\} \leftrightsquigarrow \overline{()\ 3\ \mathrm{Z}\ +} \qquad (\textsc{add–base})$$

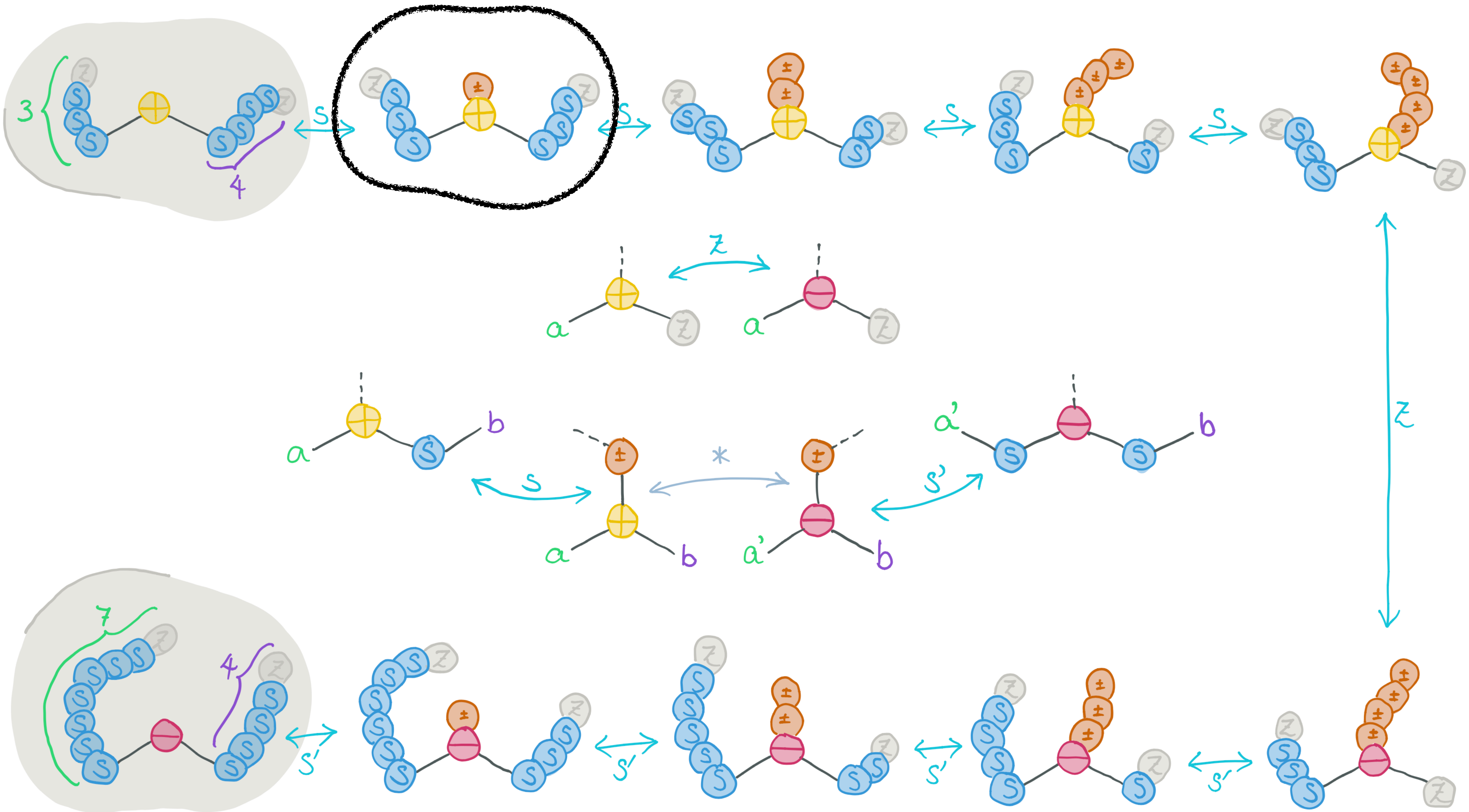
Addition



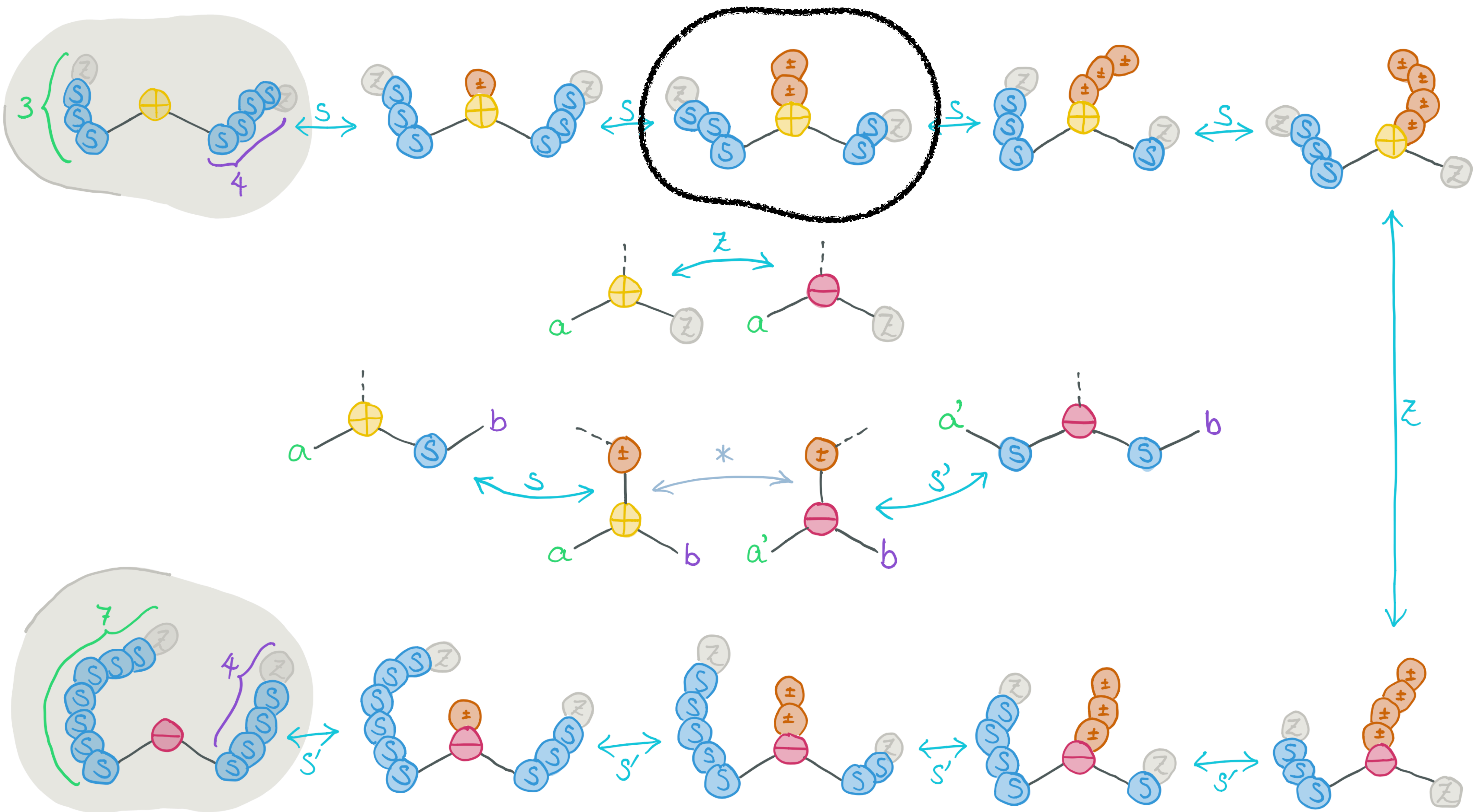
Addition



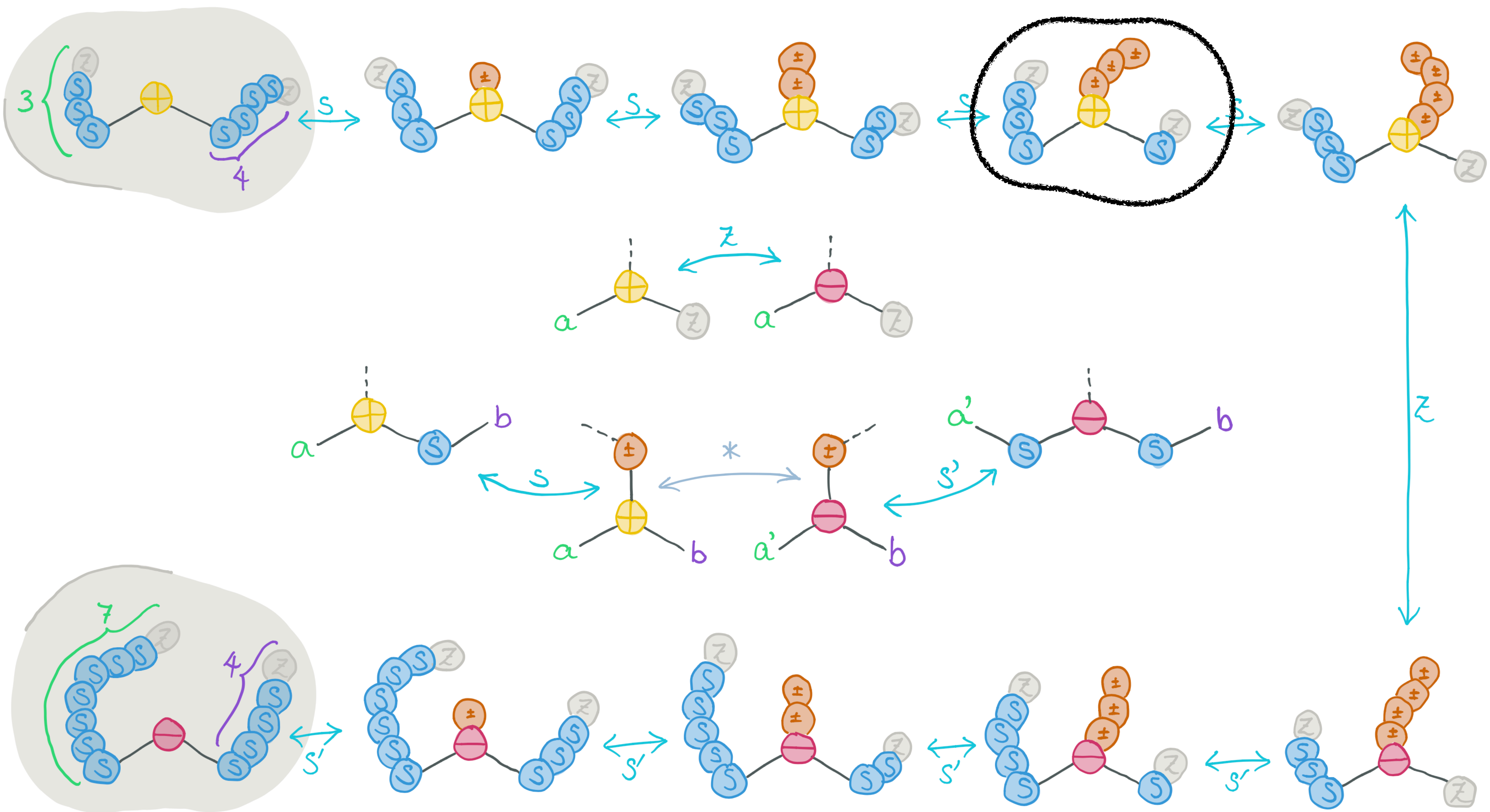
Addition



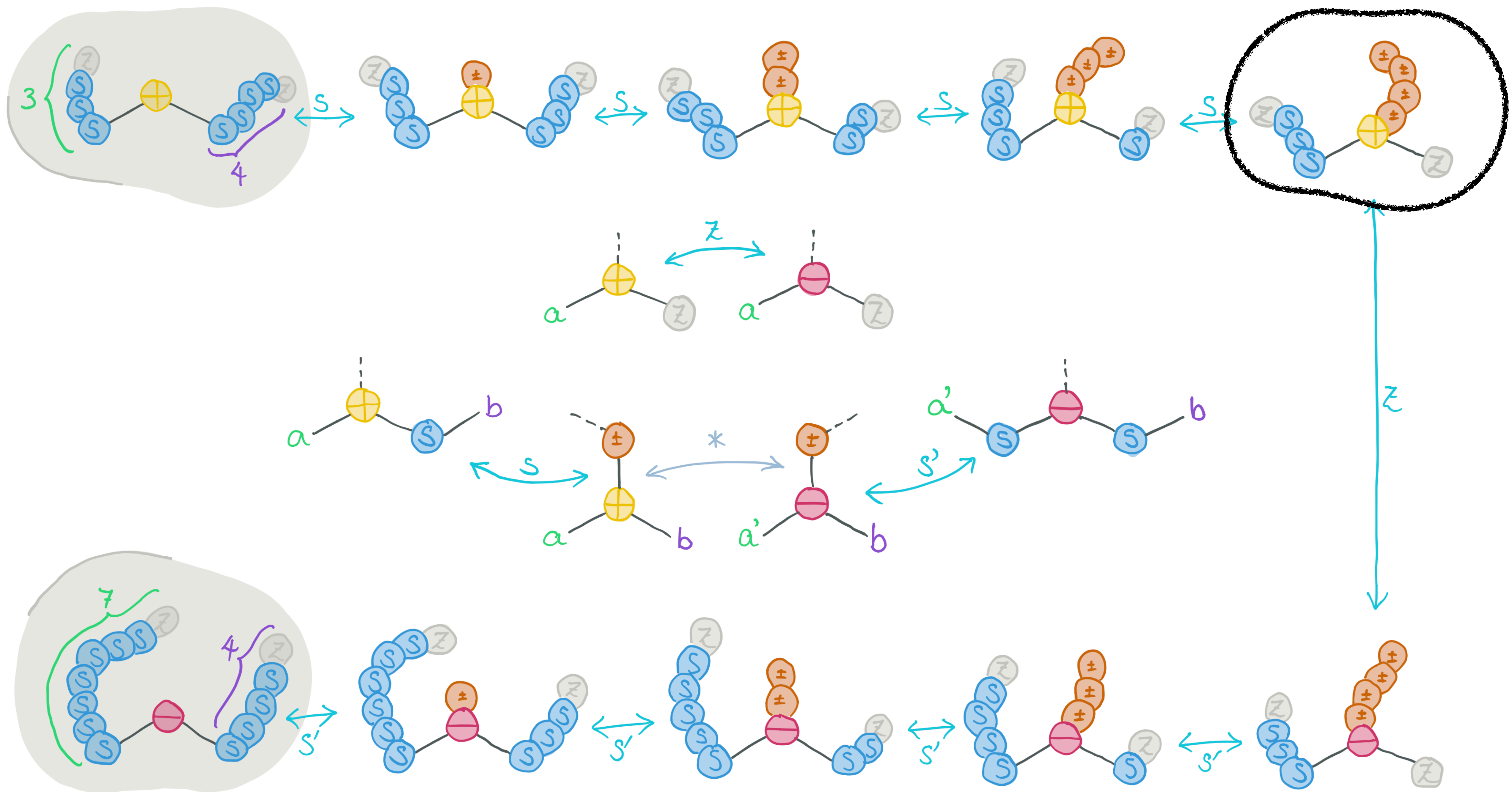
Addition



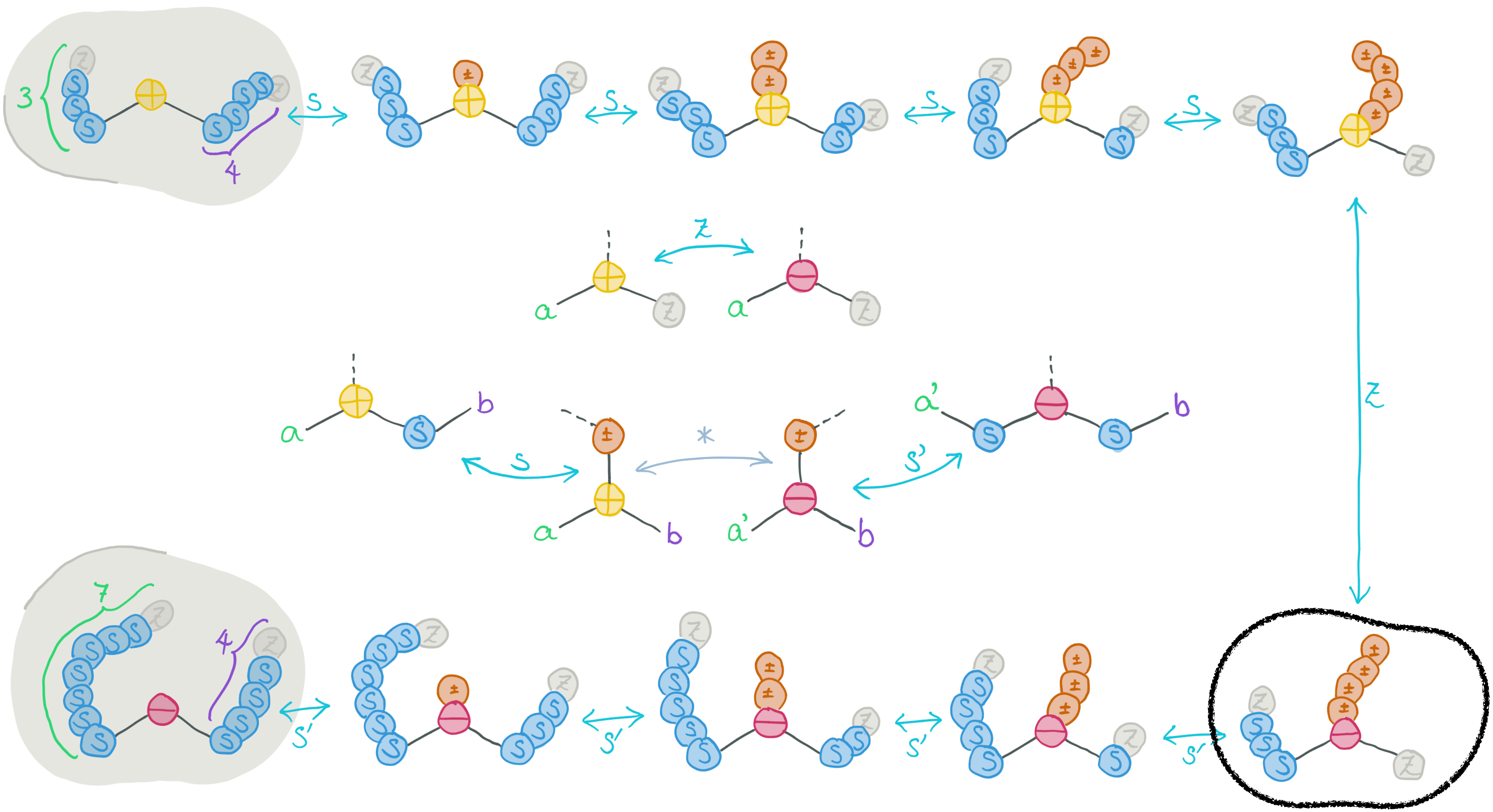
Addition



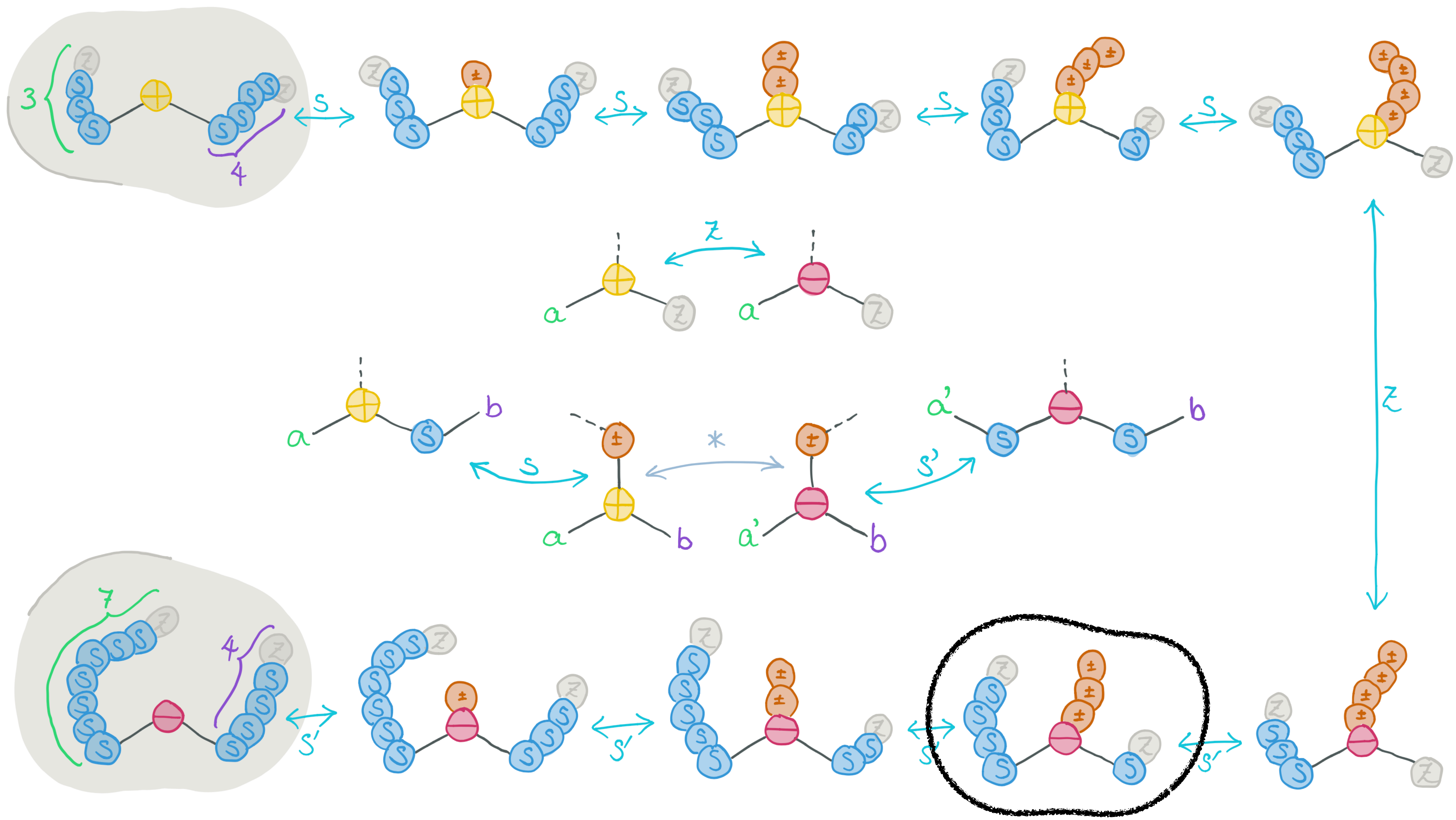
Addition



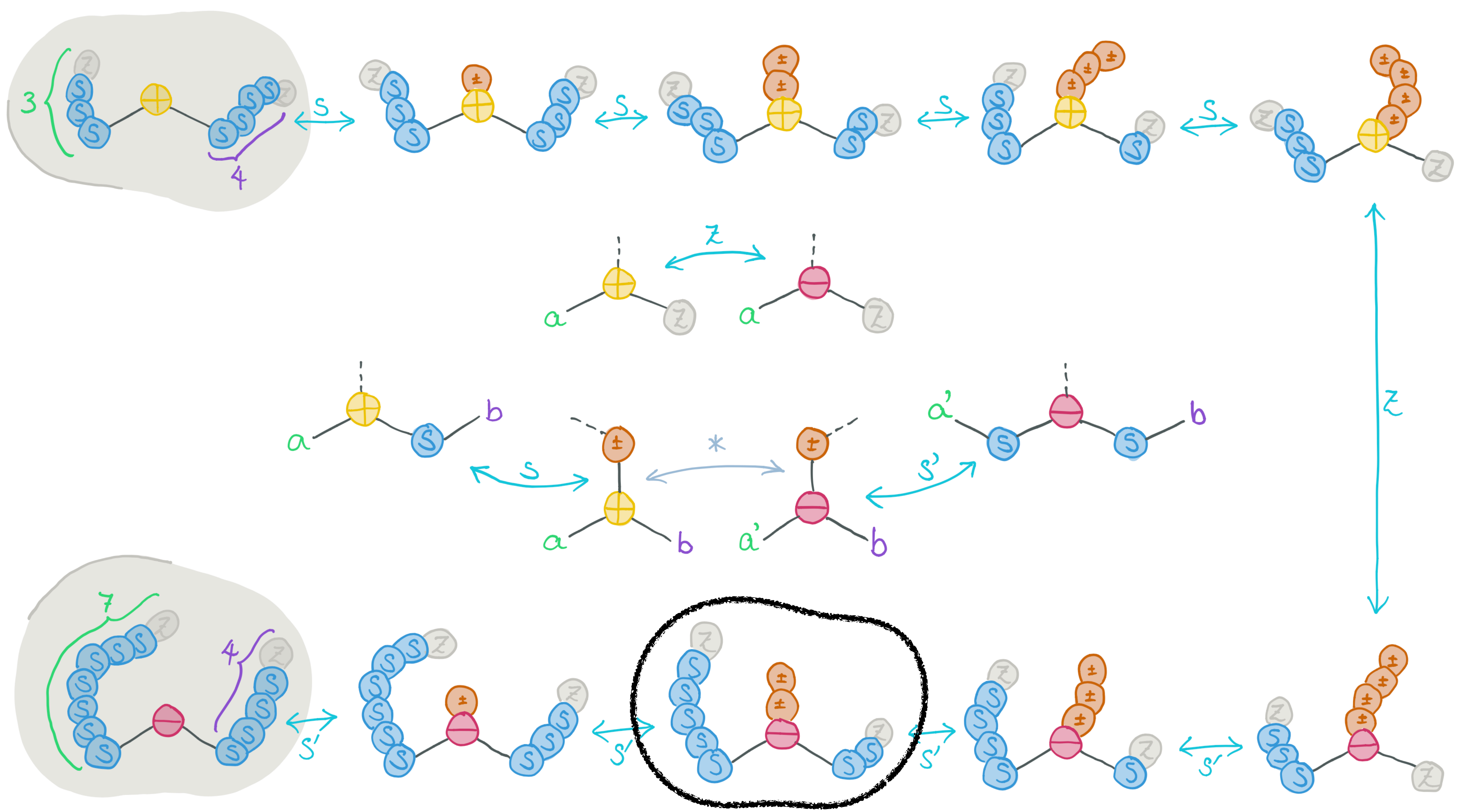
Addition



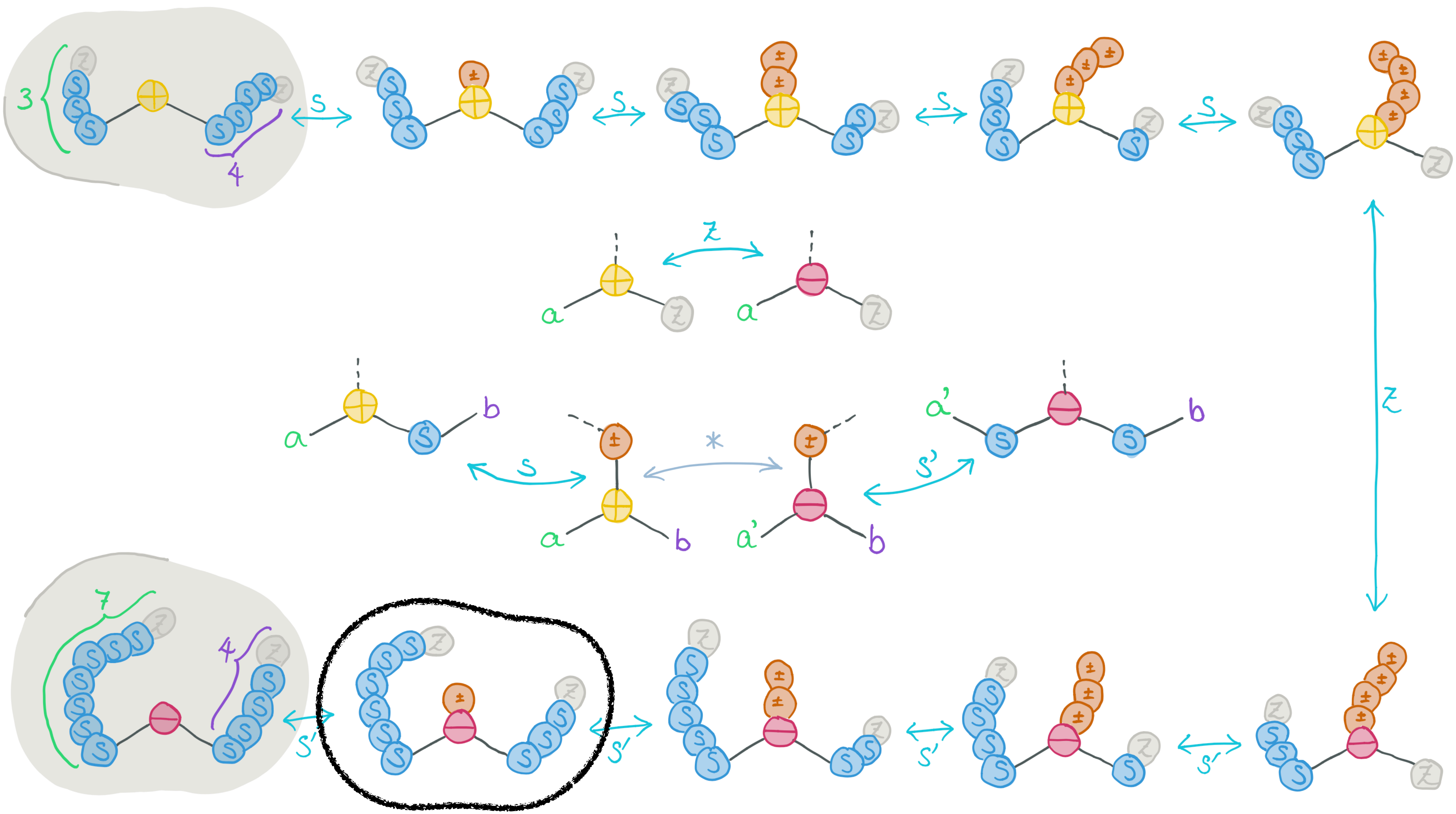
Addition



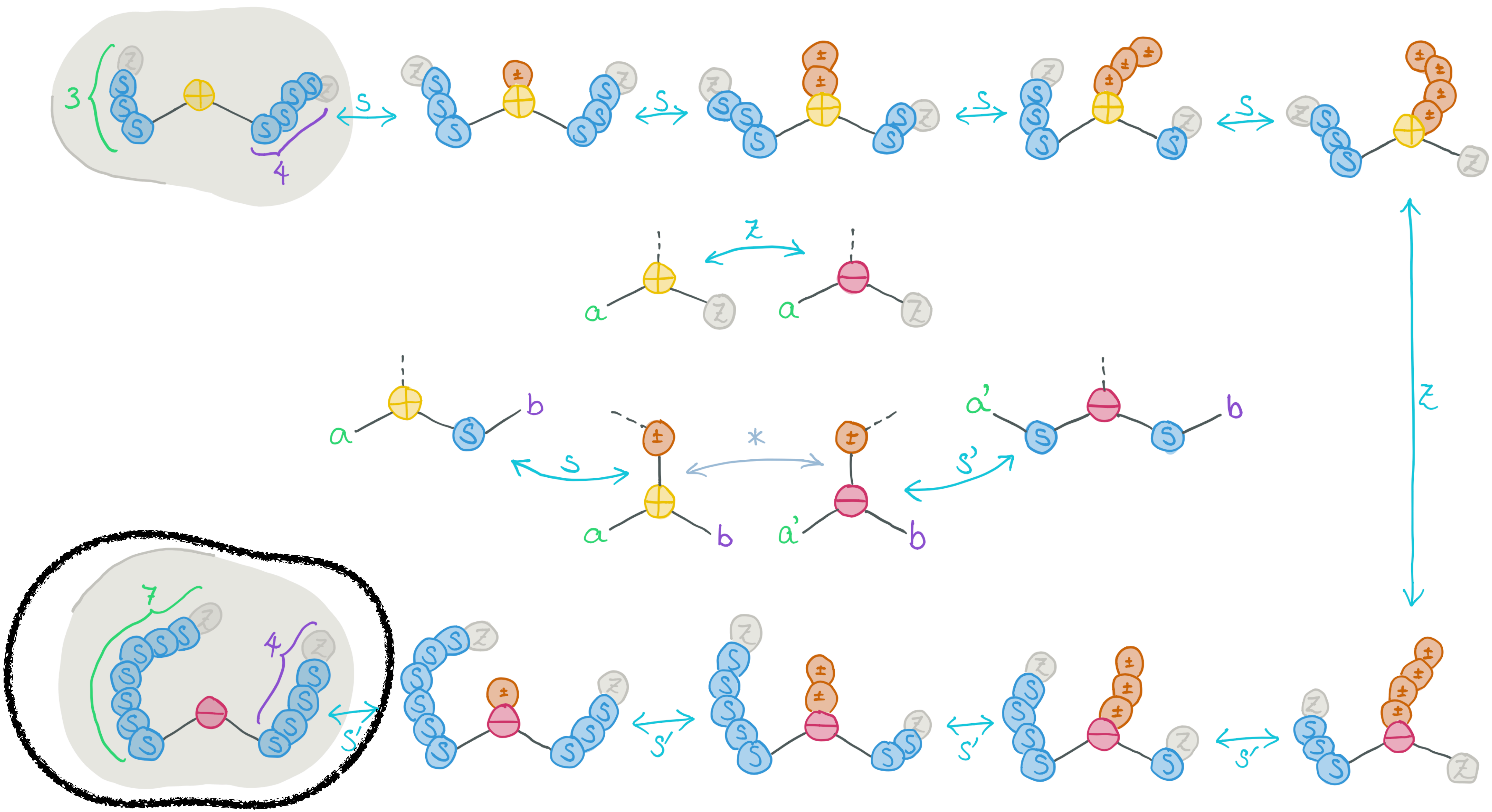
Addition



Addition



Addition



Squaring

$m^2 = \sum_{k=0}^{m-1}(k + k + 1)$

Squaring

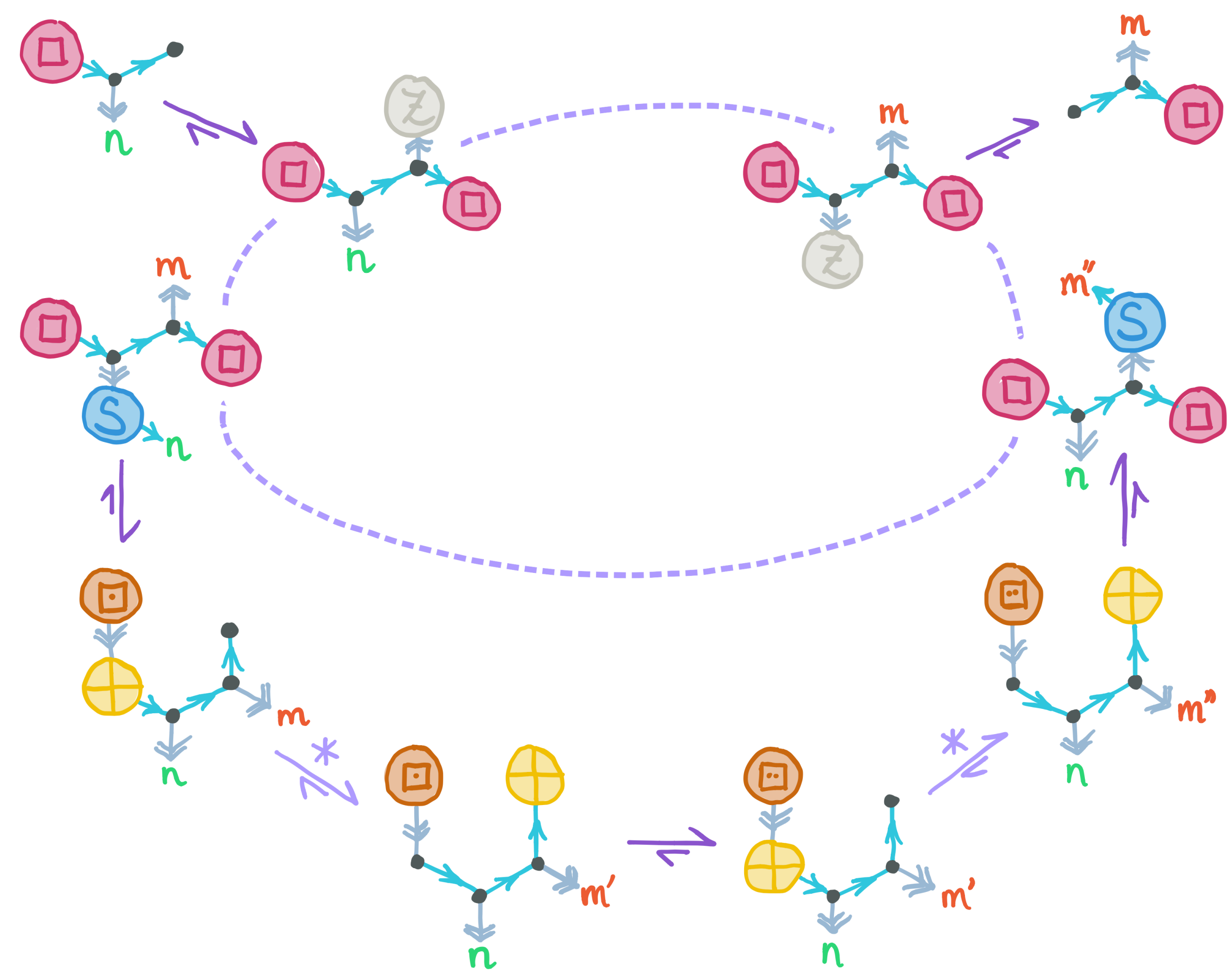
$$m^2 = \sum_{k=0}^{m-1}(k + k + 1)$$

$$
\begin{array}{ll}
\textbf{!}\ \mathrm{S{\small Q}}\ m\ ();\quad \textbf{!}\ ()\ n\ \mathrm{S{\small Q}}; & \\
\quad \mathrm{S{\small Q}}\ m\ () = \mathrm{S{\small Q}}\ \mathrm{Z}\ m\ \mathrm{S{\small Q}}; & (\textsc{sq–begin}) \\
\quad \mathrm{S{\small Q}}\ s\ (\mathrm{S}k)\ \mathrm{S{\small Q}} = \mathrm{S{\small Q}}\ (\mathrm{S}s'')\ k\ \mathrm{S{\small Q}}: & (\textsc{sq–step}) \\
\qquad\quad +\ s\ k\ () = ()\ s'\ k\ +. & \text{-- } s' \leftarrow s + k \\
\qquad\quad +\ s'\ k\ () = ()\ s''\ k\ +. & \text{-- } s'' \leftarrow s' + k \\
\quad \mathrm{S{\small Q}}\ n\ \mathrm{Z}\ \mathrm{S{\small Q}} = ()\ n\ \mathrm{S{\small Q}}; & (\textsc{sq–end})
\end{array}
$$

Squaring

$$m^2 = \sum_{k=0}^{m-1}(k+k+1)$$

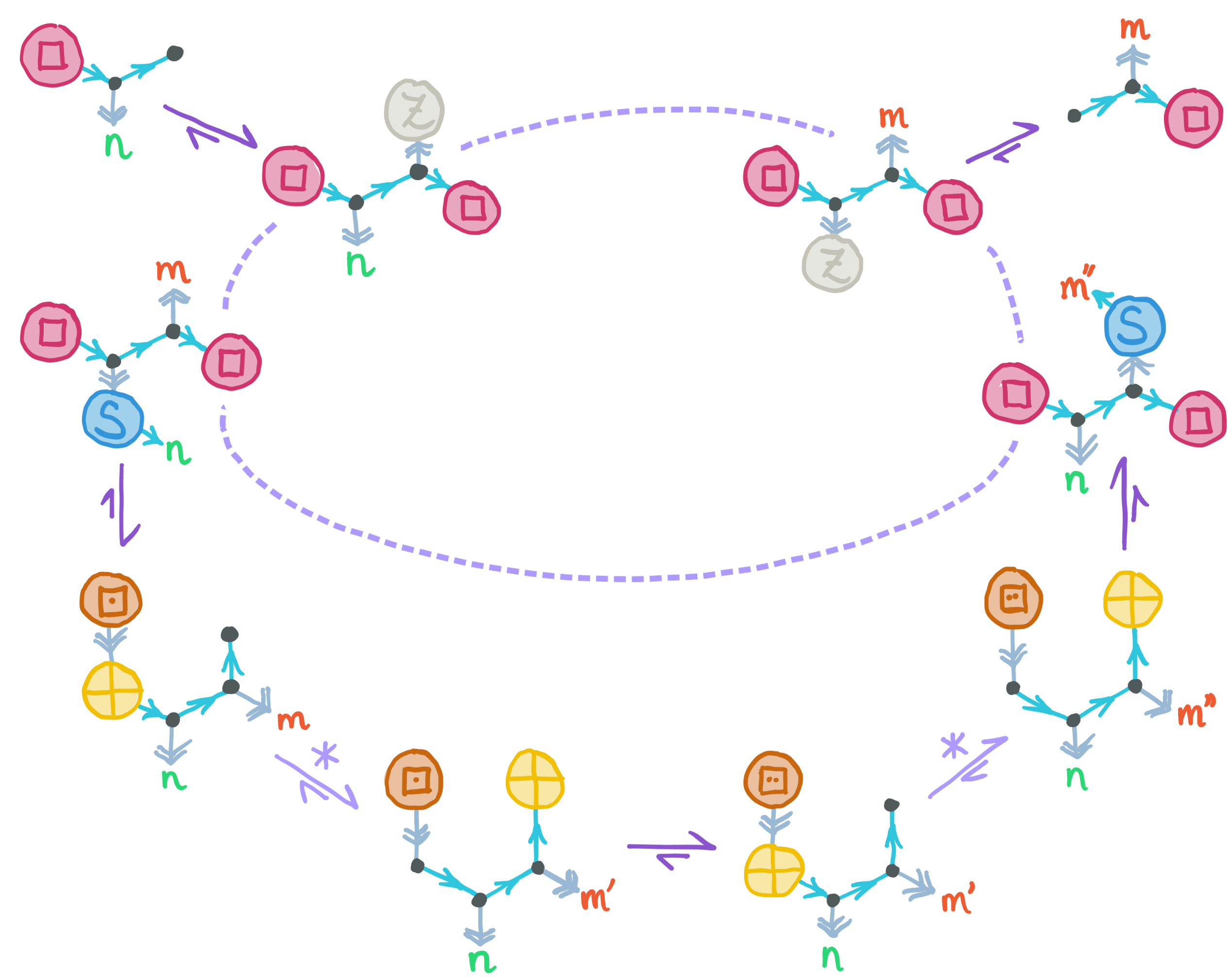
$$\begin{array}{ll}
\mathbf{!}\ \mathrm{S}\textsc{q}\ m\ ();\quad \mathbf{!}\ ()\ n\ \mathrm{S}\textsc{q}; & \\
\quad \mathrm{S}\textsc{q}\ m\ () = \mathrm{S}\textsc{q}\ \mathrm{Z}\ m\ \mathrm{S}\textsc{q}; & (\textsc{sq–begin}) \\
\quad \mathrm{S}\textsc{q}\ s\ (\mathrm{S}k)\ \mathrm{S}\textsc{q} = \mathrm{S}\textsc{q}\ (\mathrm{S}s'')\ k\ \mathrm{S}\textsc{q}: & (\textsc{sq–step}) \\
\qquad\quad + \ s\ k\ () = ()\ s'\ k\ +. & \text{-- } s' \leftarrow s + k \\
\qquad\quad + \ s'\ k\ () = ()\ s''\ k\ +. & \text{-- } s'' \leftarrow s' + k \\
\quad \mathrm{S}\textsc{q}\ n\ \mathrm{Z}\ \mathrm{S}\textsc{q} = ()\ n\ \mathrm{S}\textsc{q}; & (\textsc{sq–end})
\end{array}$$



Squaring

$$m^2 = \sum_{k=0}^{m-1}(k+k+1)$$

$$\begin{array}{ll}
\textbf{!}\ \textsc{Sq}\ m\ ();\quad \textbf{!}\ ()\ n\ \textsc{Sq}; & \\
\quad \textsc{Sq}\ m\ () = \textsc{Sq}\ \mathrm{Z}\ m\ \textsc{Sq}; & (\textsc{sq–begin}) \\
\quad \textsc{Sq}\ s\ (\mathrm{S}k)\ \textsc{Sq} = \textsc{Sq}\ (\mathrm{S}s'')\ k\ \textsc{Sq}\!: & (\textsc{sq–step}) \\
\qquad\quad + \ s\ k\ () = ()\ s'\ k\ +. & \text{-- } s' \leftarrow s\ + k \\
\qquad\quad + \ s'\ k\ () = ()\ s''\ k\ +. & \text{-- } s'' \leftarrow s' + k \\
\quad \textsc{Sq}\ n\ \mathrm{Z}\ \textsc{Sq} = ()\ n\ \textsc{Sq}; & (\textsc{sq–end})
\end{array}$$



$$\textbf{!}\quad \textsc{Sq}\ 3\ ()$$

Squaring

$$m^2 = \sum_{k=0}^{m-1}(k+k+1)$$

$\mathbf{!}\ \textsc{Sq}\ m\ ();\quad \mathbf{!}\ ()\ n\ \textsc{Sq};$

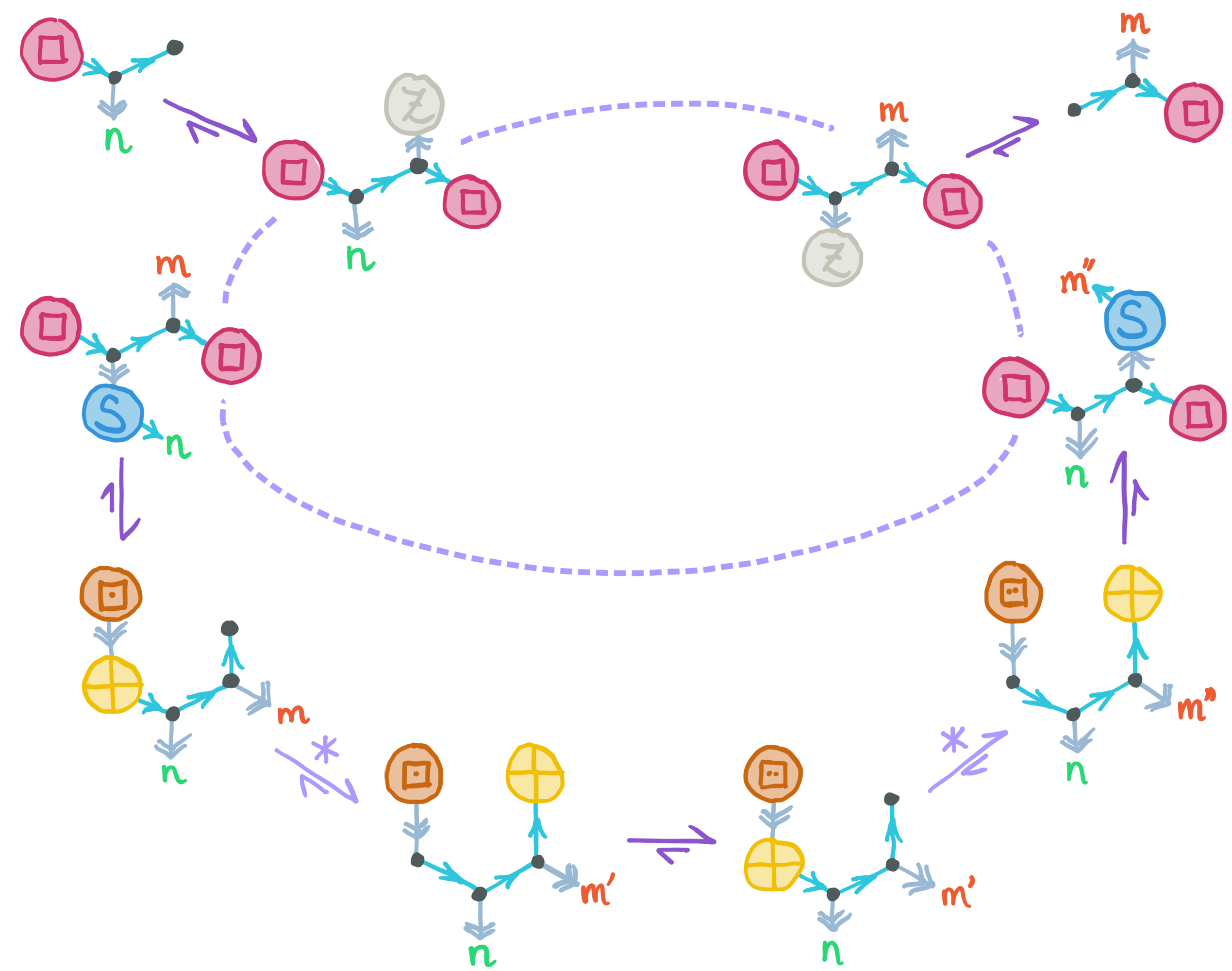
$\textsc{Sq}\ m\ () = \textsc{Sq}\ \mathrm{Z}\ m\ \textsc{Sq};$ (SQ–BEGIN)

$\textsc{Sq}\ s\ (\mathrm{S}k)\ \textsc{Sq} = \textsc{Sq}\ (\mathrm{S}s'')\ k\ \textsc{Sq}:$ (SQ–STEP)

$\quad + \ s\ k\ () = ()\ s'\ k\ +.$ -- $s' \leftarrow s + k$

$\quad + \ s'\ k\ () = ()\ s''\ k\ +.$ -- $s'' \leftarrow s' + k$

$\textsc{Sq}\ n\ \mathrm{Z}\ \textsc{Sq} = ()\ n\ \textsc{Sq};$ (SQ–END)

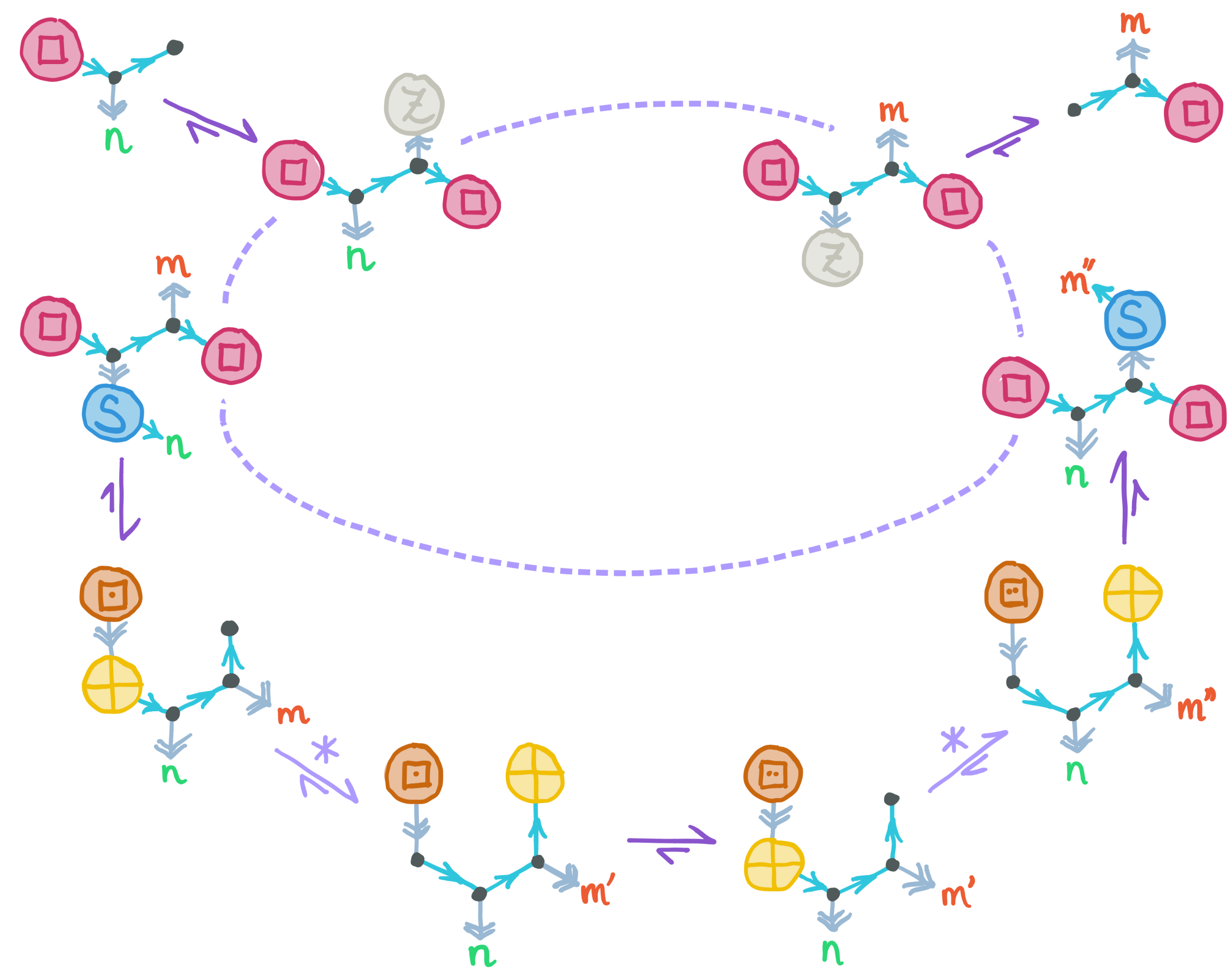


$\mathbf{!}\quad \textsc{Sq}\ 3\ () = \textsc{Sq}\ \mathrm{Z}\ 3\ \textsc{Sq}$

Squaring

$$m^2 = \sum_{k=0}^{m-1} (k + k + 1)$$

$$\begin{array}{ll}
\textbf{!}\ \textsc{Sq}\ m\ ();\quad \textbf{!}\ ()\ n\ \textsc{Sq}; & \\
\quad \textsc{Sq}\ m\ () = \textsc{Sq}\ \mathrm{Z}\ m\ \textsc{Sq}; & (\textsc{sq–begin}) \\
\quad \textsc{Sq}\ s\ (\mathrm{S}k)\ \textsc{Sq} = \textsc{Sq}\ (\mathrm{S}s'')\ k\ \textsc{Sq}: & (\textsc{sq–step}) \\
\qquad\quad + \ s\ k\ () = ()\ s'\ k\ +. & \text{--}\ s' \leftarrow s\ + k \\
\qquad\quad + \ s'\ k\ () = ()\ s''\ k\ +. & \text{--}\ s'' \leftarrow s' + k \\
\quad \textsc{Sq}\ n\ \mathrm{Z}\ \textsc{Sq} = ()\ n\ \textsc{Sq}; & (\textsc{sq–end})
\end{array}$$

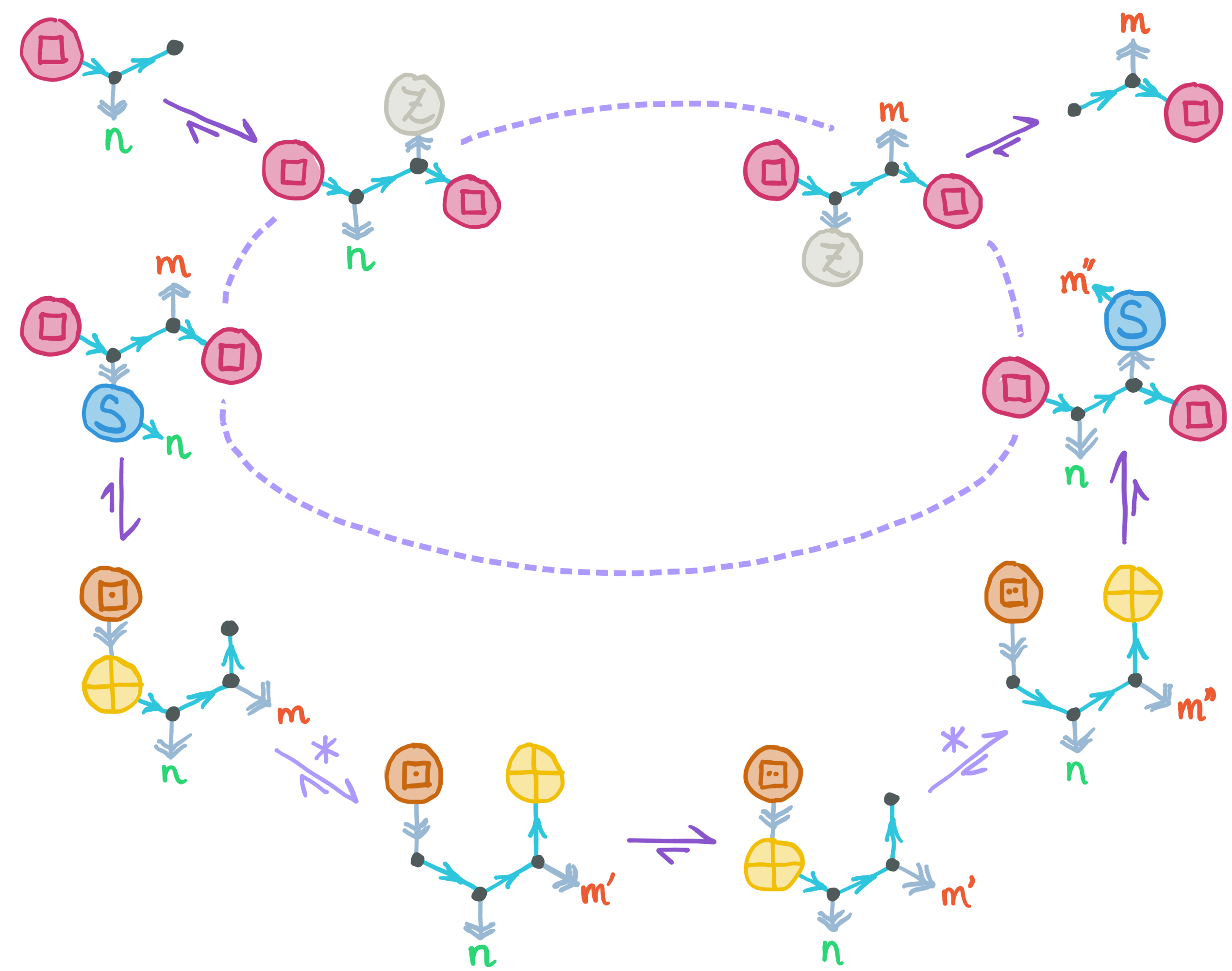


$$\textbf{!}\quad \textsc{Sq}\ 3\ () = \textsc{Sq}\ \mathrm{Z}\ 3\ \textsc{Sq} = \textsc{Sq}\ 5\ 2\ \textsc{Sq}$$

Squaring

$$m^2 = \sum_{k=0}^{m-1}(k+k+1)$$

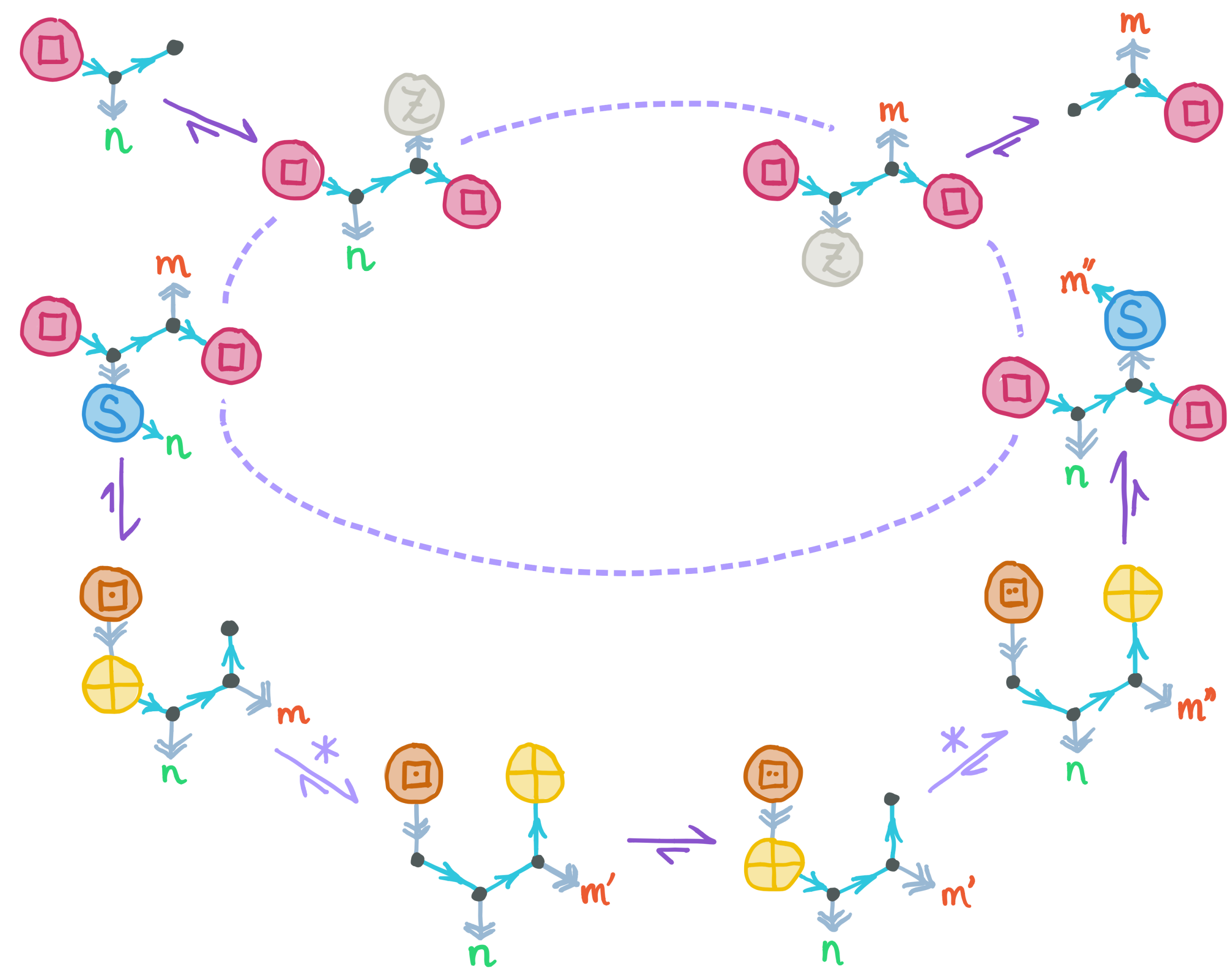
$$\begin{array}{ll}
\textbf{!}\ \textsc{Sq}\ m\ ();\quad \textbf{!}\ ()\ n\ \textsc{Sq}; & \\
\textsc{Sq}\ m\ () = \textsc{Sq}\ \mathrm{Z}\ m\ \textsc{Sq}; & (\textsc{sq–begin}) \\
\textsc{Sq}\ s\ (\mathrm{S}k)\ \textsc{Sq} = \textsc{Sq}\ (\mathrm{S}s'')\ k\ \textsc{Sq}: & (\textsc{sq–step}) \\
\qquad + \ s\ k\ () = ()\ s'\ k\ +. & \text{-- } s' \leftarrow s + k \\
\qquad + \ s'\ k\ () = ()\ s''\ k\ +. & \text{-- } s'' \leftarrow s' + k \\
\textsc{Sq}\ n\ \mathrm{Z}\ \textsc{Sq} = ()\ n\ \textsc{Sq}; & (\textsc{sq–end})
\end{array}$$



$$\textbf{!}\quad \textsc{Sq}\ 3\ () = \textsc{Sq}\ \mathrm{Z}\ 3\ \textsc{Sq} = \textsc{Sq}\ 5\ 2\ \textsc{Sq} = \textsc{Sq}\ 8\ 1\ \textsc{Sq}$$

Squaring

$$m^2 = \sum_{k=0}^{m-1} (k + k + 1)$$

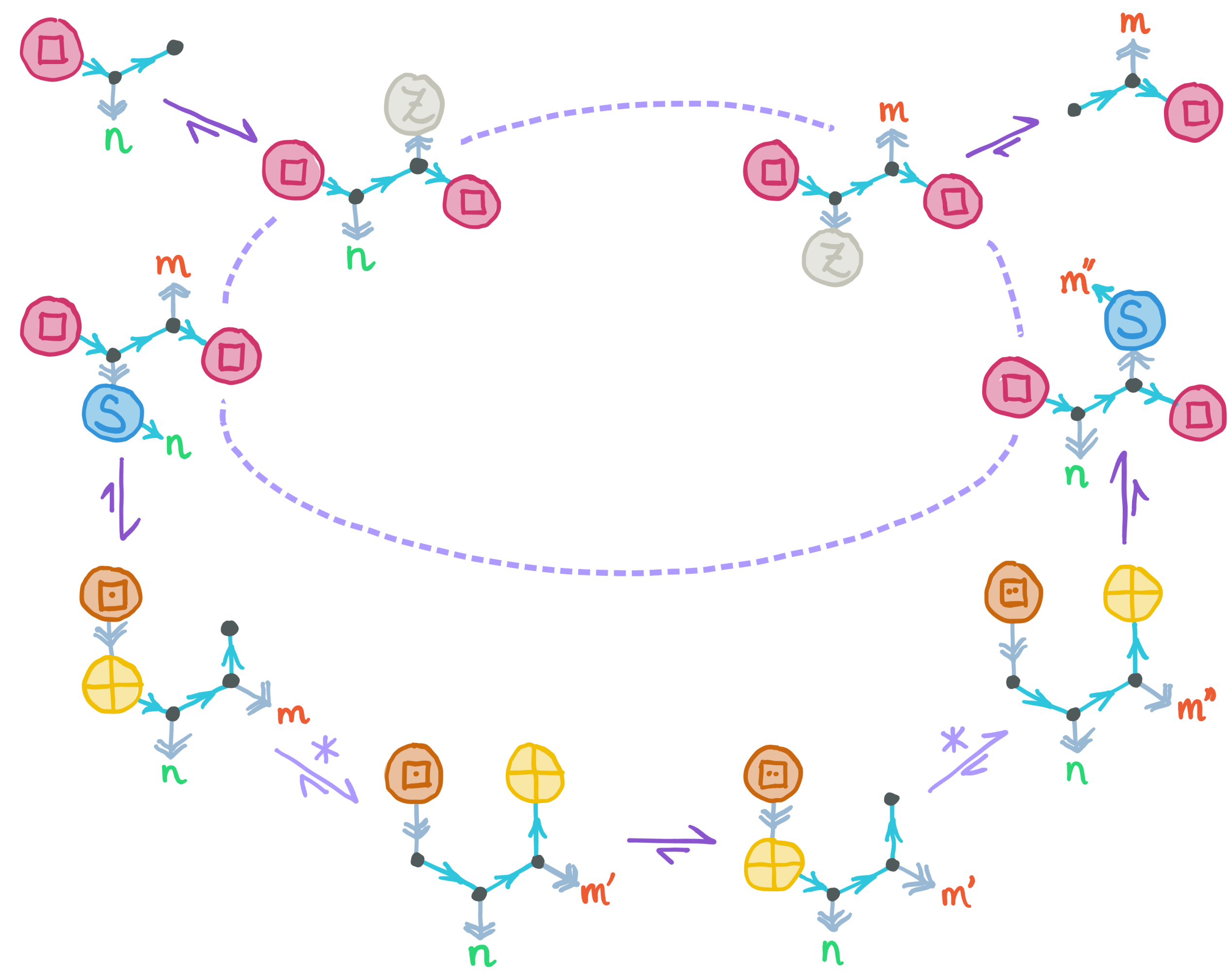


$$
\begin{array}{ll}
\textbf{!}\ \textsc{Sq}\ m\ ();\quad \textbf{!}\ ()\ n\ \textsc{Sq}; & \\
\quad \textsc{Sq}\ m\ () = \textsc{Sq}\ \mathrm{Z}\ m\ \textsc{Sq}; & (\textsc{sq–begin}) \\
\quad \textsc{Sq}\ s\ (\mathrm{S}k)\ \textsc{Sq} = \textsc{Sq}\ (\mathrm{S}s'')\ k\ \textsc{Sq}: & (\textsc{sq–step}) \\
\qquad\quad +\ s\ k\ () = ()\ s'\ k\ +. & \text{-- } s' \leftarrow s\ + k \\
\qquad\quad +\ s'\ k\ () = ()\ s''\ k\ +. & \text{-- } s'' \leftarrow s' + k \\
\quad \textsc{Sq}\ n\ \mathrm{Z}\ \textsc{Sq} = ()\ n\ \textsc{Sq}; & (\textsc{sq–end})
\end{array}
$$

$$\textbf{!}\quad \textsc{Sq}\ 3\ () = \textsc{Sq}\ \mathrm{Z}\ 3\ \textsc{Sq} = \textsc{Sq}\ 5\ 2\ \textsc{Sq} = \textsc{Sq}\ 8\ 1\ \textsc{Sq} = \textsc{Sq}\ 9\ \mathrm{Z}\ \textsc{Sq}$$

Squaring

$$m^2 = \sum_{k=0}^{m-1}(k+k+1)$$



$$
\begin{array}{lll}
\textbf{!}\ \textsc{Sq}\ m\ ();\quad \textbf{!}\ ()\ n\ \textsc{Sq}; & & \\
\textsc{Sq}\ m\ () = \textsc{Sq}\ \mathrm{Z}\ m\ \textsc{Sq}; & & (\textsc{sq–begin}) \\
\textsc{Sq}\ s\ (\mathrm{S}k)\ \textsc{Sq} = \textsc{Sq}\ (\mathrm{S}s'')\ k\ \textsc{Sq}: & & (\textsc{sq–step}) \\
\quad + \ s\ k\ () = ()\ s'\ k\ +. & & \text{-- } s' \leftarrow s + k \\
\quad + \ s'\ k\ () = ()\ s''\ k\ +. & & \text{-- } s'' \leftarrow s' + k \\
\textsc{Sq}\ n\ \mathrm{Z}\ \textsc{Sq} = ()\ n\ \textsc{Sq}; & & (\textsc{sq–end})
\end{array}
$$

$$\textbf{!}\quad \textsc{Sq}\ 3\ () = \textsc{Sq}\ \mathrm{Z}\ 3\ \textsc{Sq} = \textsc{Sq}\ 5\ 2\ \textsc{Sq} = \textsc{Sq}\ 8\ 1\ \textsc{Sq} = \textsc{Sq}\ 9\ \mathrm{Z}\ \textsc{Sq} = ()\ 9\ \textsc{Sq}\quad \textbf{!}$$

Squaring

$$m^2 = \sum_{k=0}^{m-1}(k + k + 1)$$

$!\ \mathrm{S}\textsc{q}\ m\ ();\quad !\ ()\ n\ \mathrm{S}\textsc{q};$

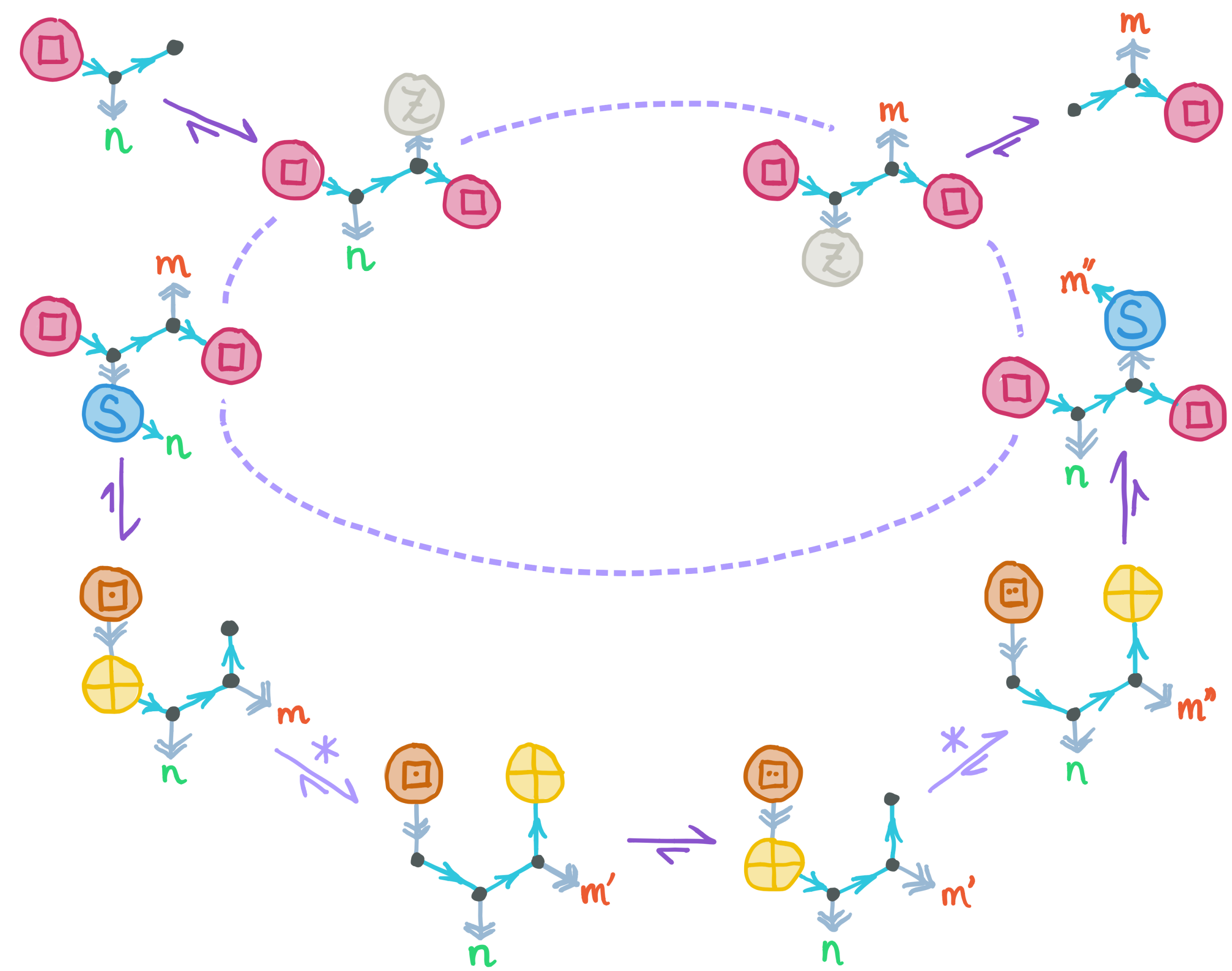
$\mathrm{S}\textsc{q}\ m\ () = \mathrm{S}\textsc{q}\ \mathrm{Z}\ m\ \mathrm{S}\textsc{q};$ (SQ–BEGIN)

$\mathrm{S}\textsc{q}\ s\ (\mathrm{S}k)\ \mathrm{S}\textsc{q} = \mathrm{S}\textsc{q}\ (\mathrm{S}s'')\ k\ \mathrm{S}\textsc{q}:$ (SQ–STEP)

$+\ s\ k\ () = ()\ s'\ k\ +.$ -- $s' \leftarrow s + k$

$+\ s'\ k\ () = ()\ s''\ k\ +.$ -- $s'' \leftarrow s' + k$

$\mathrm{S}\textsc{q}\ n\ \mathrm{Z}\ \mathrm{S}\textsc{q} = ()\ n\ \mathrm{S}\textsc{q};$ (SQ–END)



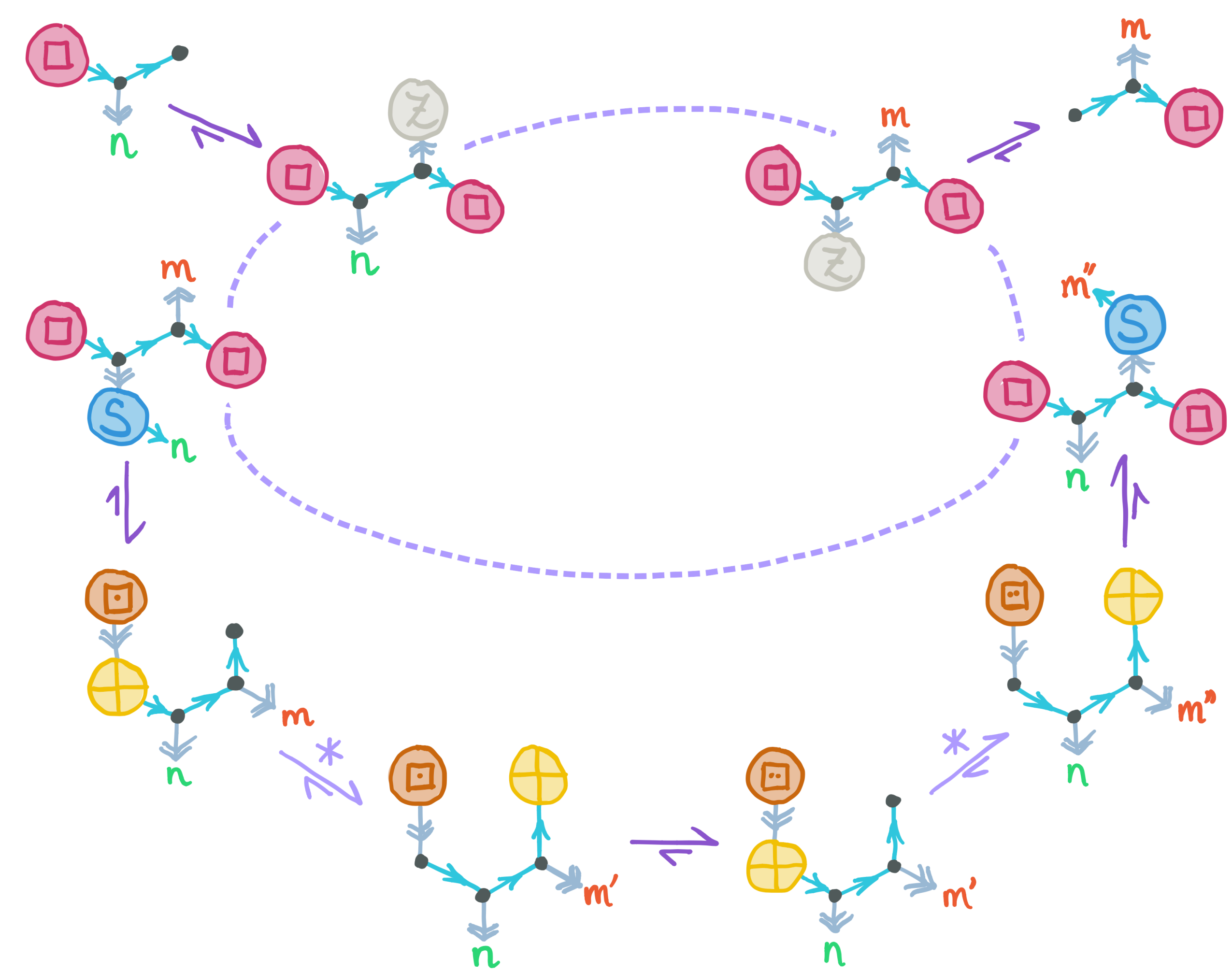
$!\quad ()\ 10\ \mathrm{S}\textsc{q}$

Squaring

$$m^2 = \sum_{k=0}^{m-1} (k + k + 1)$$

$! \ \textsc{Sq}\ m\ ();\quad !\ ()\ n\ \textsc{Sq};$

$\textsc{Sq}\ m\ () = \textsc{Sq}\ \mathrm{Z}\ m\ \textsc{Sq};$	(SQ–BEGIN)
$\textsc{Sq}\ s\ (\mathrm{S}k)\ \textsc{Sq} = \textsc{Sq}\ (\mathrm{S}s'')\ k\ \textsc{Sq}:$	(SQ–STEP)
$\quad + \ s\ k\ () = ()\ s'\ k\ +.$	-- $s' \leftarrow s\ + k$
$\quad + \ s'\ k\ () = ()\ s''\ k\ +.$	-- $s'' \leftarrow s' + k$
$\textsc{Sq}\ n\ \mathrm{Z}\ \textsc{Sq} = ()\ n\ \textsc{Sq};$	(SQ–END)

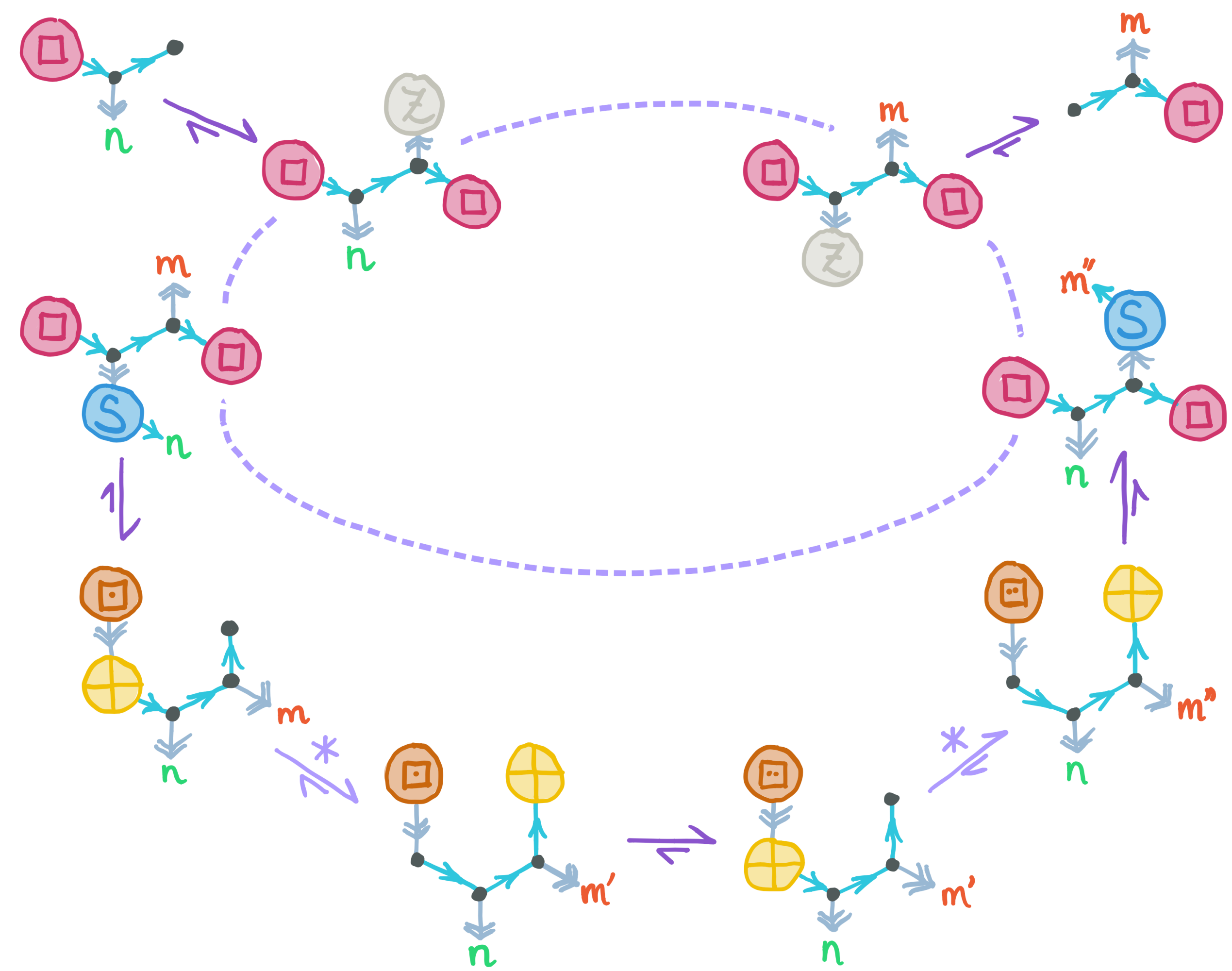


$$!\quad ()\ 10\ \textsc{Sq} = \textsc{Sq}\ 10\ \mathrm{Z}\ \textsc{Sq}$$

Squaring

$$m^2 = \sum_{k=0}^{m-1}(k + k + 1)$$

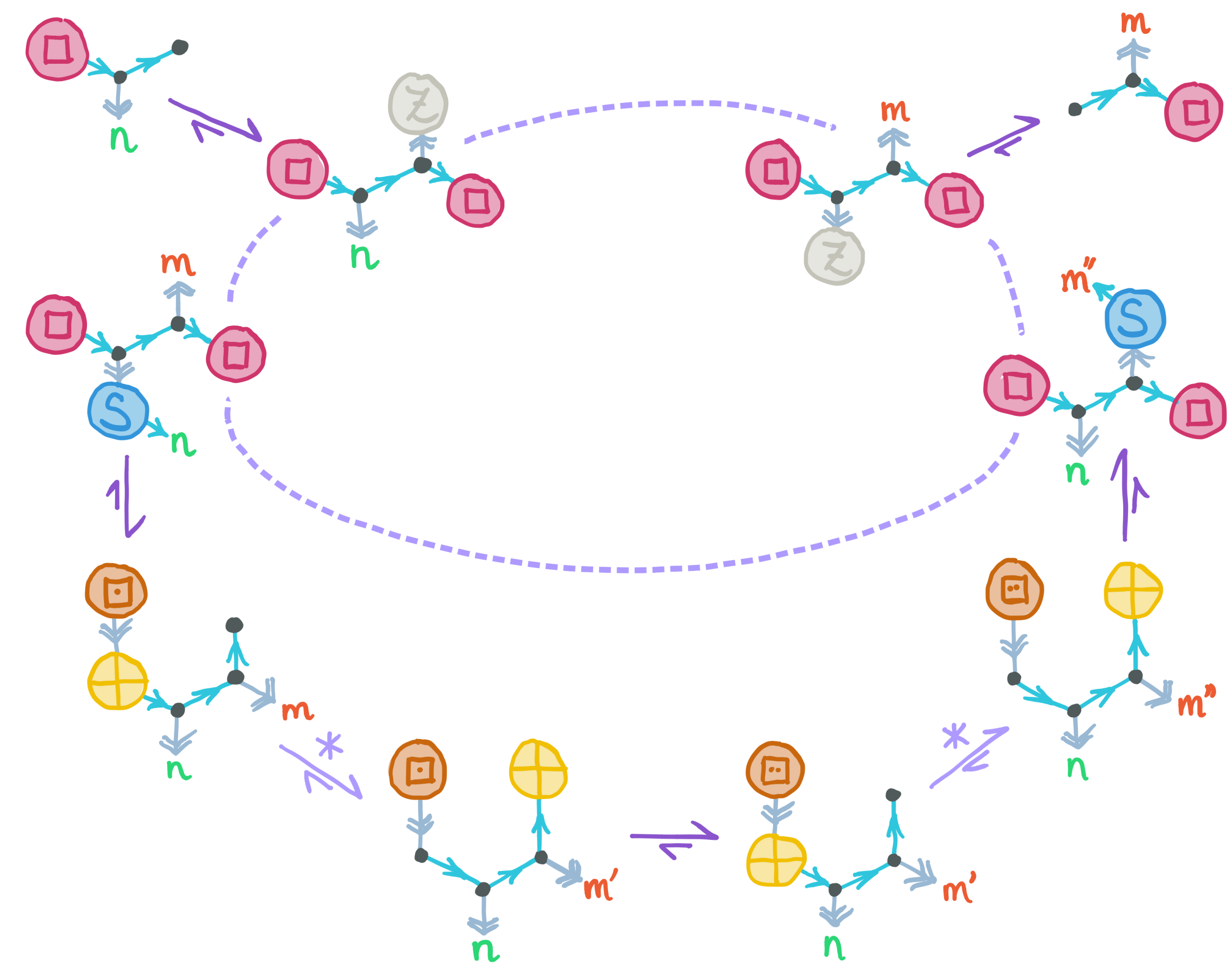
$$
\begin{array}{ll}
\textbf{!}\ \textsc{Sq}\ m\ ();\quad \textbf{!}\ ()\ n\ \textsc{Sq}; & \\
\quad \textsc{Sq}\ m\ () = \textsc{Sq}\ \mathrm{Z}\ m\ \textsc{Sq}; & (\textsc{sq–begin}) \\
\quad \textsc{Sq}\ s\ (\mathrm{S}k)\ \textsc{Sq} = \textsc{Sq}\ (\mathrm{S}s'')\ k\ \textsc{Sq}: & (\textsc{sq–step}) \\
\qquad\quad +\ s\ k\ () = ()\ s'\ k\ +. & \text{-- } s' \leftarrow s\ + k \\
\qquad\quad +\ s'\ k\ () = ()\ s''\ k\ +. & \text{-- } s'' \leftarrow s' + k \\
\quad \textsc{Sq}\ n\ \mathrm{Z}\ \textsc{Sq} = ()\ n\ \textsc{Sq}; & (\textsc{sq–end})
\end{array}
$$



$$\textbf{!}\quad ()\ 10\ \textsc{Sq} = \textsc{Sq}\ 10\ \mathrm{Z}\ \textsc{Sq} = \textsc{Sq}\ 9\ 1\ \textsc{Sq}$$

Squaring

$$m^2 = \sum_{k=0}^{m-1}(k+k+1)$$



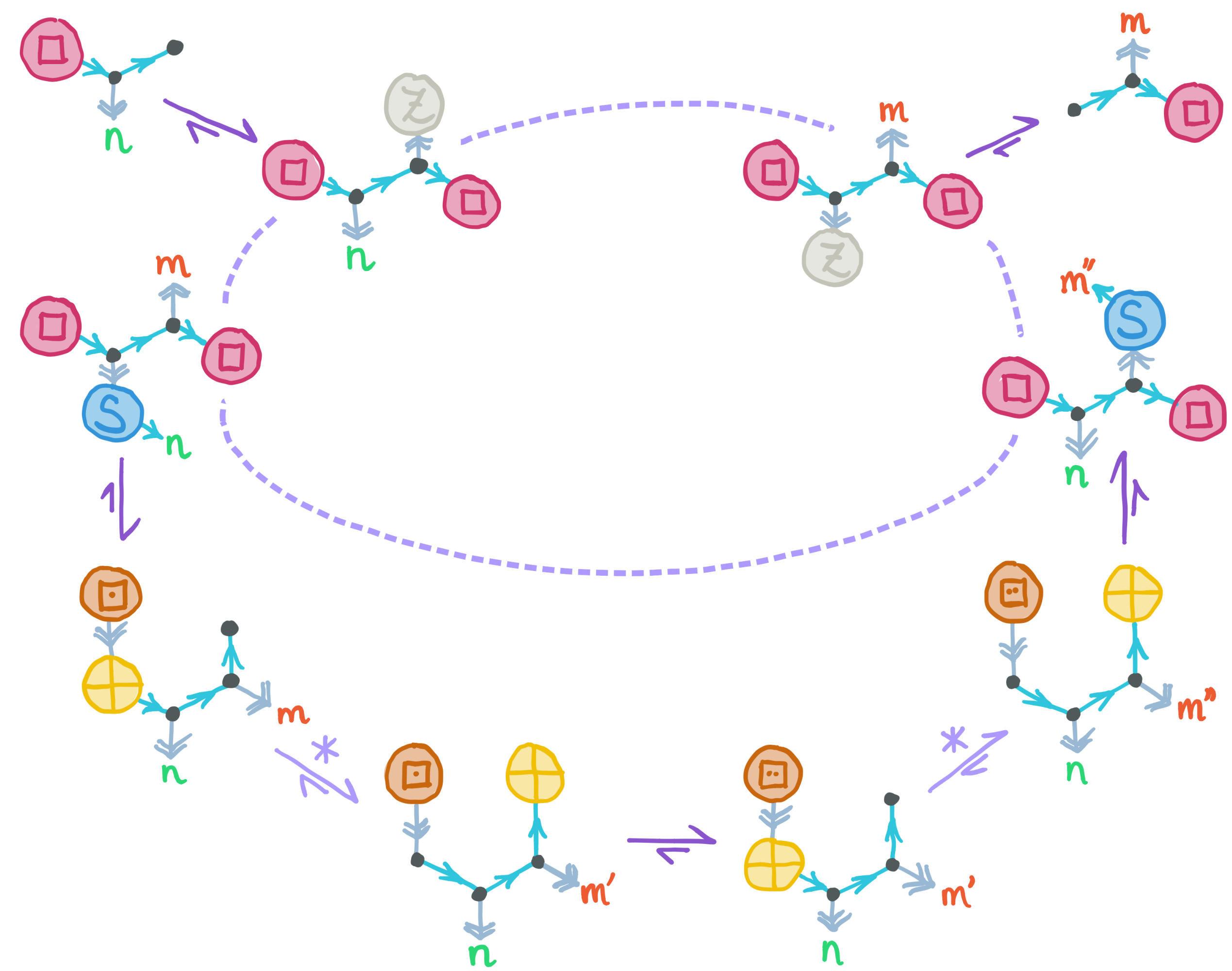
$$
\begin{array}{lll}
\textbf{!}\ \textsc{Sq}\ m\ ();\quad \textbf{!}\ ()\ n\ \textsc{Sq}; & & \\
\quad \textsc{Sq}\ m\ () = \textsc{Sq}\ \mathrm{Z}\ m\ \textsc{Sq}; & & (\textsc{sq–begin}) \\
\quad \textsc{Sq}\ s\ (\mathrm{S}k)\ \textsc{Sq} = \textsc{Sq}\ (\mathrm{S}s'')\ k\ \textsc{Sq}: & & (\textsc{sq–step}) \\
\qquad\quad +\ s\ k\ () = ()\ s'\ k\ +. & & \text{-- } s' \leftarrow s + k \\
\qquad\quad +\ s'\ k\ () = ()\ s''\ k\ +. & & \text{-- } s'' \leftarrow s' + k \\
\quad \textsc{Sq}\ n\ \mathrm{Z}\ \textsc{Sq} = ()\ n\ \textsc{Sq}; & & (\textsc{sq–end})
\end{array}
$$

$$\textbf{!}\quad ()\ 10\ \textsc{Sq} = \textsc{Sq}\ 10\ \mathrm{Z}\ \textsc{Sq} = \textsc{Sq}\ 9\ 1\ \textsc{Sq} = \textsc{Sq}\ 6\ 2\ \textsc{Sq}$$

Squaring

$$m^2 = \sum_{k=0}^{m-1}(k + k + 1)$$

$$\begin{array}{ll}
\textbf{!}\ \textsc{Sq}\ m\ ();\quad \textbf{!}\ ()\ n\ \textsc{Sq}; & \\
\quad \textsc{Sq}\ m\ () = \textsc{Sq}\ \mathrm{Z}\ m\ \textsc{Sq}; & (\textsc{sq–begin}) \\
\quad \textsc{Sq}\ s\ (\mathrm{S}k)\ \textsc{Sq} = \textsc{Sq}\ (\mathrm{S}s'')\ k\ \textsc{Sq}: & (\textsc{sq–step}) \\
\qquad\quad + \ s\ k\ () = ()\ s'\ k\ +. & \text{-- } s' \leftarrow s\ + k \\
\qquad\quad + \ s'\ k\ () = ()\ s''\ k\ +. & \text{-- } s'' \leftarrow s' + k \\
\quad \textsc{Sq}\ n\ \mathrm{Z}\ \textsc{Sq} = ()\ n\ \textsc{Sq}; & (\textsc{sq–end})
\end{array}$$

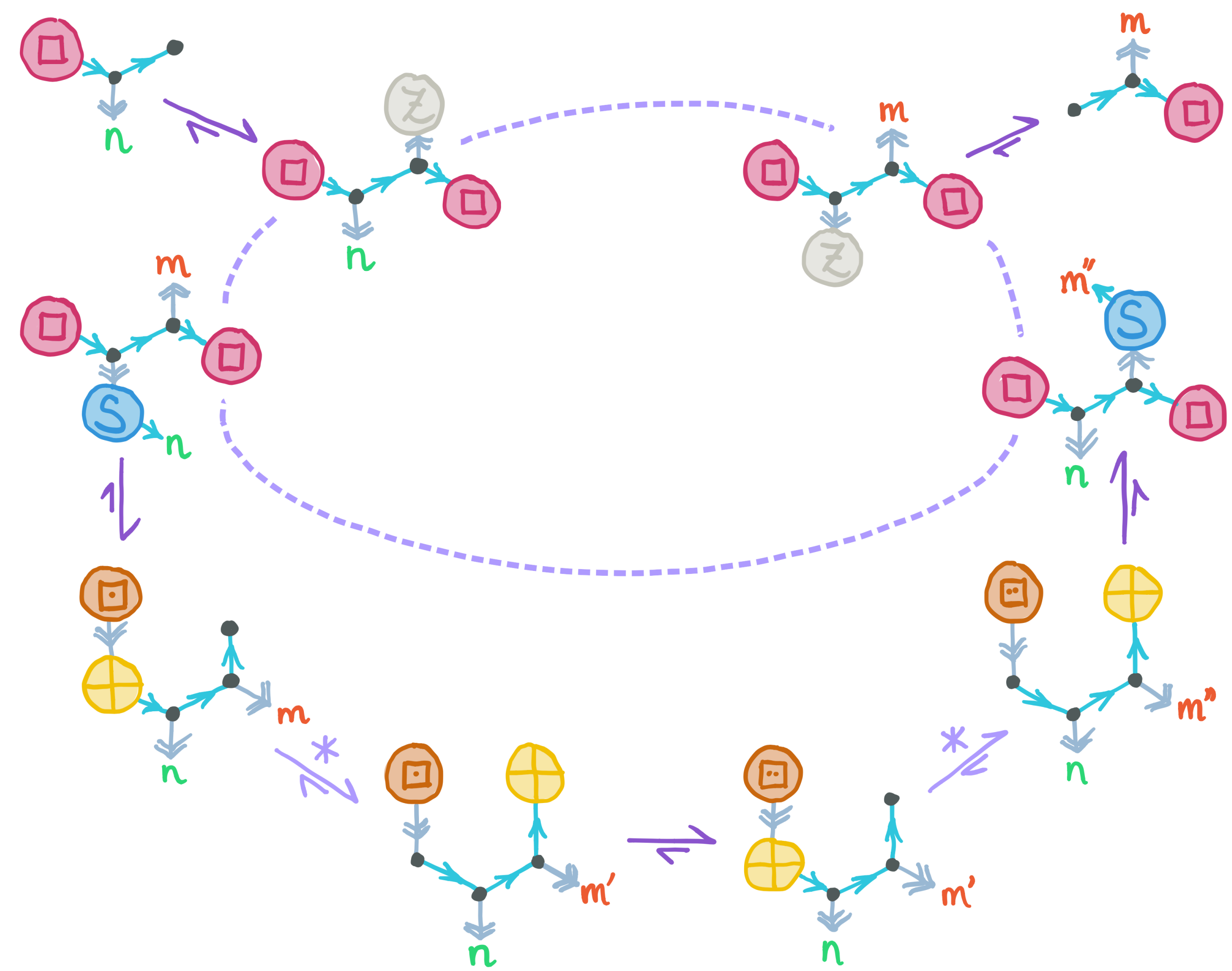


$$\textbf{!}\quad ()\ 10\ \textsc{Sq} = \textsc{Sq}\ 10\ \mathrm{Z}\ \textsc{Sq} = \textsc{Sq}\ 9\ 1\ \textsc{Sq} = \textsc{Sq}\ 6\ 2\ \textsc{Sq} = \textsc{Sq}\ 1\ 3\ \textsc{Sq}$$

Squaring

$$m^2 = \sum_{k=0}^{m-1} (k + k + 1)$$

$$
\begin{array}{lll}
\mathbf{!}\ \textsc{Sq}\ m\ ();\quad \mathbf{!}\ ()\ n\ \textsc{Sq}; & & \\
\quad \textsc{Sq}\ m\ () = \textsc{Sq}\ \mathrm{Z}\ m\ \textsc{Sq}; & & (\textsc{sq–begin}) \\
\quad \textsc{Sq}\ s\ (\mathrm{S}k)\ \textsc{Sq} = \textsc{Sq}\ (\mathrm{S}s'')\ k\ \textsc{Sq}: & & (\textsc{sq–step}) \\
\qquad\quad +\ s\ k\ () = ()\ s'\ k\ +. & & \text{--}\ s' \leftarrow s\ + k \\
\qquad\quad +\ s'\ k\ () = ()\ s''\ k\ +. & & \text{--}\ s'' \leftarrow s' + k \\
\quad \textsc{Sq}\ n\ \mathrm{Z}\ \textsc{Sq} = ()\ n\ \textsc{Sq}; & & (\textsc{sq–end})
\end{array}
$$



$$\mathbf{!}\quad ()\ 10\ \textsc{Sq} = \textsc{Sq}\ 10\ \mathrm{Z}\ \textsc{Sq} = \textsc{Sq}\ 9\ 1\ \textsc{Sq} = \textsc{Sq}\ 6\ 2\ \textsc{Sq} = \textsc{Sq}\ 1\ 3\ \textsc{Sq}\quad \perp$$

reversible programming

א: motivation, semantics, & tutorial

א: advanced features & properties

alethe + א concurrency

Higher Order

$$\mathbf{!}\ \textsc{Map}\ \_;\quad \mathbf{!}\ (\textsc{Map}\ \_)\ \_\ ();\quad \mathbf{!}\ ()\ \_\ (\textsc{Map}\ \_);$$

$$(\textsc{Map}\ f)\ \textsc{Nil}\ () = ()\ \textsc{Nil}\ (\textsc{Map}\ f);$$
$$(\textsc{Map}\ f)\ (\textsc{Cons}\ x\ x\cancel{s})\ () = ()\ (\textsc{Cons}\ y\ y\cancel{s})\ (\textsc{Map}\ f):$$
$$\quad f\ x\ () = ()\ y\ f.$$
$$\quad (\textsc{Map}\ f)\ x\cancel{s}\ () = ()\ y\cancel{s}\ (\textsc{Map}\ f).$$

$$[]\ `\textsc{Map}\ f`\ [];$$
$$[x \cdot x\cancel{s}]\ `\textsc{Map}\ f`\ [y \cdot y\cancel{s}]:$$
$$\quad x\ `f`\ y.$$
$$\quad x\cancel{s}\ `\textsc{Map}\ f`\ y\cancel{s}.$$

$$\mathbf{!}\quad (\textsc{Map}\ \square)\ [3\ 5\ 8]\ () \leftrightsquigarrow \{\underline{f:\square, x:3}, \underline{f':\square, x\cancel{s}:[5\ 8]}\}$$

$$\overline{\square\ 3\ ()} = \underline{()\ 9\ \square} \qquad \overline{(\textsc{Map}\ \square)\ [5\ 8]\ ()} = \underline{()\ [25\ 64]\ (\textsc{Map}\ \square)}$$

$$\{\overline{f:\square, y:9}, \overline{f':\square, y\cancel{s}:[25\ 64]}\} \leftrightsquigarrow ()\ [9\ 25\ 64]\ (\textsc{Map}\ \square)\quad \mathbf{!}$$

r-Turing Completeness

$[\textsc{Blank}\ x \cdot xs]$ \`$\textsc{Pop}$\` $\textsc{Blank}\ [x \cdot xs]$;

$[(\textsc{Sym}\ x) \cdot xs]$ \`$\textsc{Pop}$\` $(\textsc{Sym}\ x)\ xs$;

$[]$ \`$\textsc{Pop}$\` $\textsc{Blank}\ []$;

$!\ \textsc{Tape}\ \ell\ x\ r$;

$!\ \textsc{Sym}\ x$;

$!\ \textsc{Blank}$;

$(\textsc{Tape}\ \ell\ x\ r)$ \`$\textsc{Left}$\` $(\textsc{Tape}\ \ell'\ x'\ r')$:

  ℓ \`$\textsc{Pop}$\` $x'\ \ell'$.

  r' \`$\textsc{Pop}$\` $x\ r$.

t \`$\textsc{Right}$\` t':

  t' \`$\textsc{Left}$\` t.

r-Turing Completeness

$$\textbf{!}\ \textsc{Start}\ t_1\ t_2\ t_3\ t_4\ t_5\ t_6;$$

$$\textbf{!}\ \textsc{Stop}\ t_1\ t_2\ t_3\ t_4\ t_5\ t_6;$$

$$\begin{aligned}
&S_1\ (\textsc{Tape}\ \ell_1\ (\textsc{Sym}\ C)\ r_1)\ (\textsc{Tape}\ \ell_2\ \textsc{Blank}\ r_2)\ t_3\ (\textsc{Tape}\ \ell_4\ (\textsc{Sym}\ D)\ r_4)\ t_5\ t_6\\
=\ &S_2\ (\textsc{Tape}\ \ell_1\ (\textsc{Sym}\ B)\ r_1)\ (\textsc{Tape}\ \ell_2\ (\textsc{Sym}\ A)\ r_2)\ t_3'\ (\textsc{Tape}\ \ell_4\ (\textsc{Sym}\ F)\ r_4)\ t_5\ t_6':\\
&\qquad t_3\ `\textsc{Left}`\ t_3'.\\
&\qquad t_6\ `\textsc{Right}`\ t_6'.
\end{aligned}$$

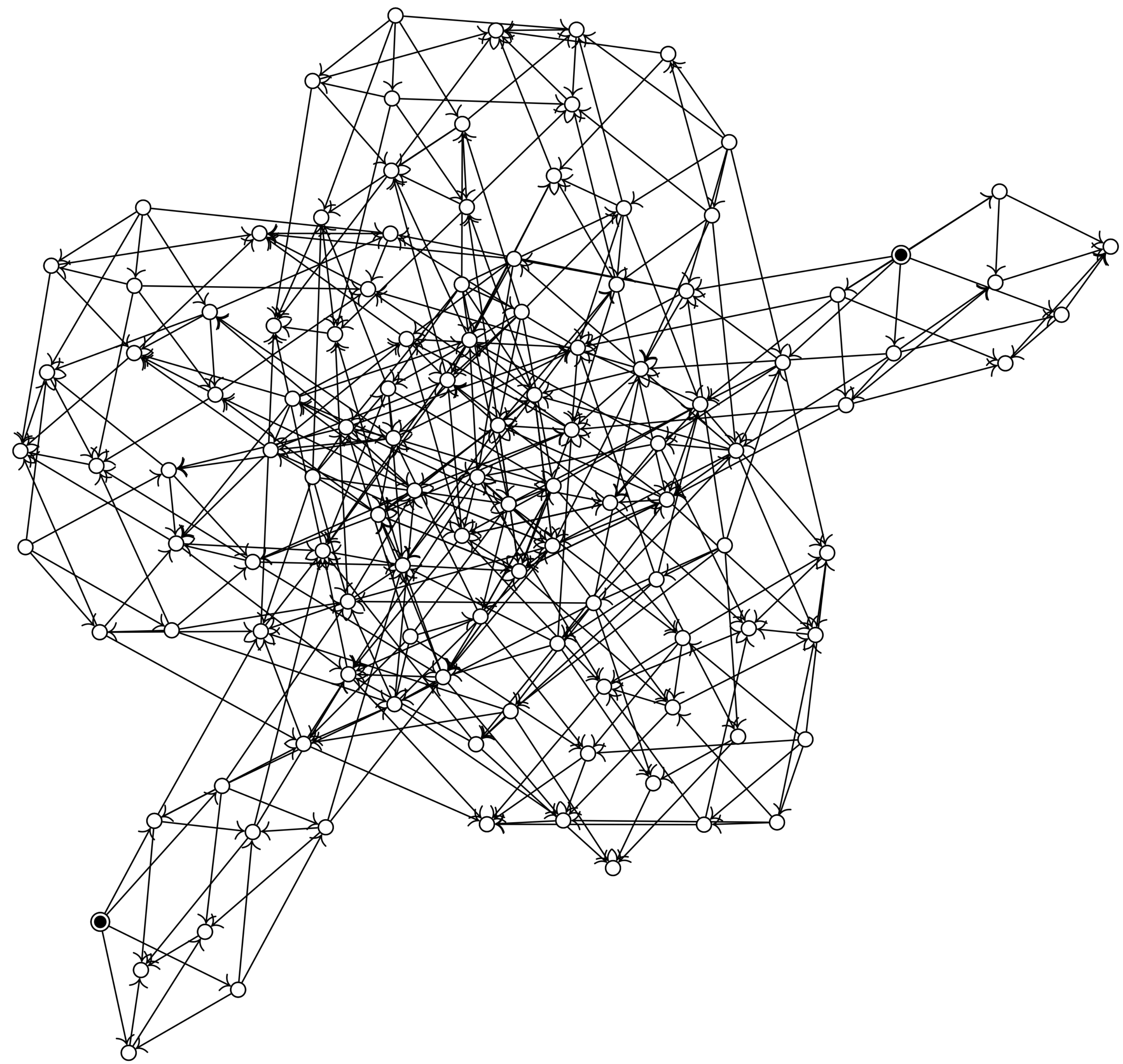
Ambiguity

- rTM: disjoint domains and codomains
- א: symmetric definitions
  - more subtle — no term can match >2 (comp) patterns
  - edge case — ≤1 comp pattern and any halt patterns
- simple graph-based algorithm
- relaxation $\Rightarrow$ non-deterministic א

Execution Planning

$a/b \text{`}{+}\ (\textsc{Frac}\ p\ q)\text{`}\ c/d$:

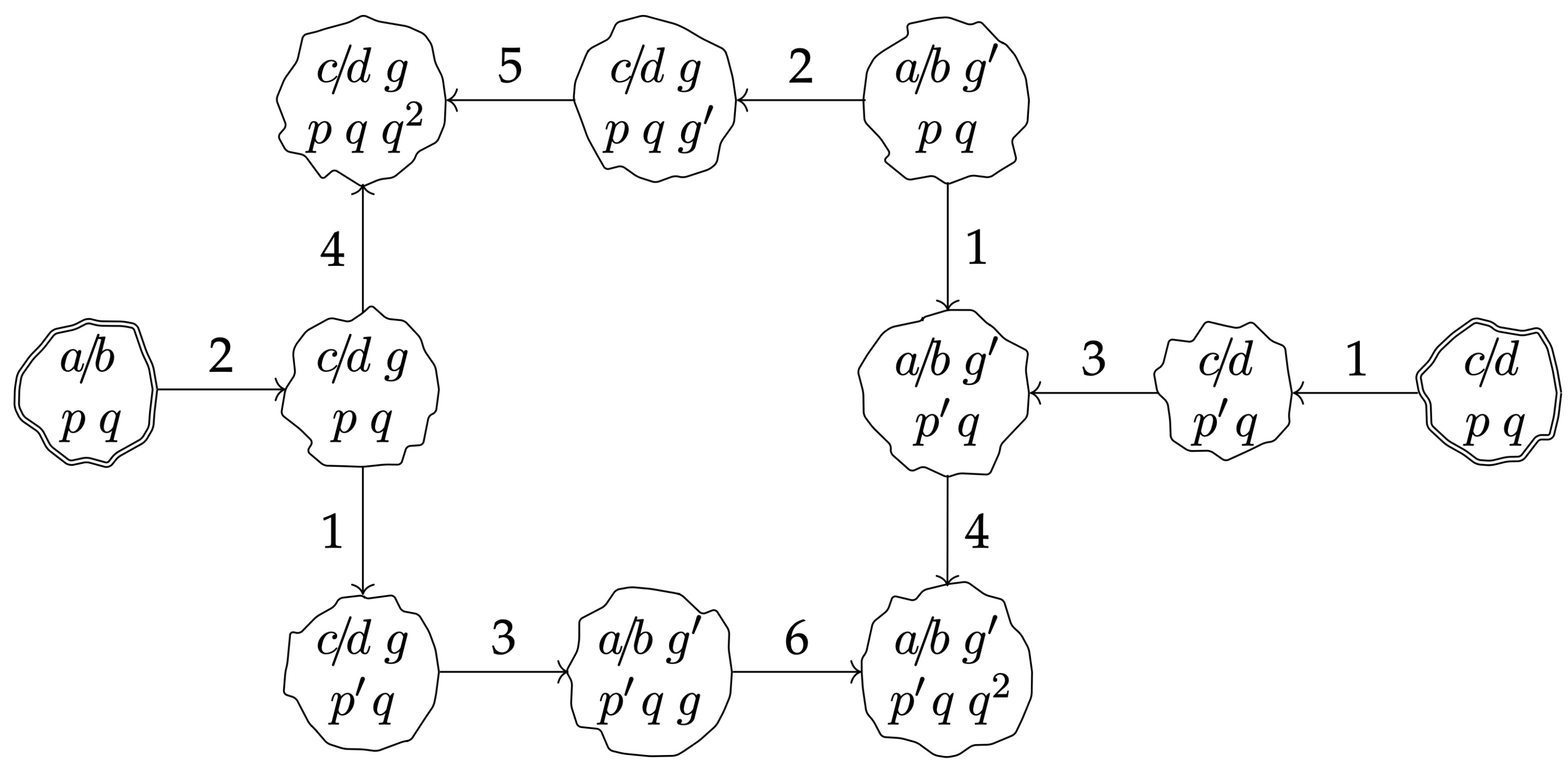
1. $p - p'$.
2. $a/b \text{`}{\sim}\ (\textsc{Frac}\ p\ q)\text{`}\ c/d\ g$.
3. $c/d \text{`}{\sim}\ (\textsc{Frac}\ p'\ q)\text{`}\ a/b\ g'$.
4. $(\mathrm{S}q) \,\square\, q^2$.
5. $g' \text{`}{\times}\ g\text{`}\ q^2$.
6. $g \text{`}{\times}\ g'\text{`}\ q^2$.



Execution Planning

$a/b\ \text{`}{+}\ (\textsc{Frac}\ p\ q)\text{`}\ c/d$:

1. $p - p'$.
2. $a/b\ \text{`}{\sim}\ (\textsc{Frac}\ p\ q)\text{`}\ c/d\ g$.
3. $c/d\ \text{`}{\sim}\ (\textsc{Frac}\ p'\ q)\text{`}\ a/b\ g'$.
4. $(\mathrm{S}q)\ \square\ q^2$.
5. $g'\ \text{`}{\times}\ g\text{`}\ q^2$.
6. $g\ \text{`}{\times}\ g'\text{`}\ q^2$.

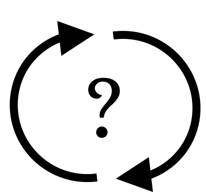


Directional Evaluation

$$
\begin{array}{ll}
\textbf{!}\ \mathrm{S}\textsc{q}\ m\ ();\quad \textbf{!}\ ()\ n\ \mathrm{S}\textsc{q}; & \\
\quad \mathrm{S}\textsc{q}\ m\ () = \mathrm{S}\textsc{q}\ \mathrm{Z}\ m\ \mathrm{S}\textsc{q}; & (\textsc{sq–begin}) \\
\quad \mathrm{S}\textsc{q}\ s\ (\mathrm{S}k)\ \mathrm{S}\textsc{q} = \mathrm{S}\textsc{q}\ (\mathrm{S}s'')\ k\ \mathrm{S}\textsc{q}: & (\textsc{sq–step}) \\
\qquad\quad + \ s\ k\ () = ()\ s'\ k\ +. & \text{--}\ s' \leftarrow s\ + k \\
\qquad\quad + \ s'\ k\ () = ()\ s''\ k\ +. & \text{--}\ s'' \leftarrow s' + k \\
\quad \mathrm{S}\textsc{q}\ n\ \mathrm{Z}\ \mathrm{S}\textsc{q} = ()\ n\ \mathrm{S}\textsc{q}; & (\textsc{sq–end})
\end{array}
$$

$$
\textbf{!}\quad \mathrm{S}\textsc{q}\ 3\ () = \mathrm{S}\textsc{q}\ \mathrm{Z}\ 3\ \mathrm{S}\textsc{q} \underset{*}{=} \mathrm{S}\textsc{q}\ 5\ 2\ \mathrm{S}\textsc{q} \underset{*}{=} \mathrm{S}\textsc{q}\ 8\ 1\ \mathrm{S}\textsc{q} \underset{*}{=} \mathrm{S}\textsc{q}\ 9\ \mathrm{Z}\ \mathrm{S}\textsc{q} = ()\ 9\ \mathrm{S}\textsc{q} \quad \textbf{!}
$$

$*$ $(\textsc{sq–step})$

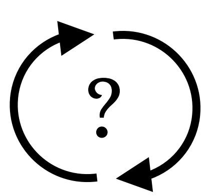


Directional Evaluation

$$
\begin{array}{ll}
\textbf{!}\ \mathrm{S}\textsc{q}\ m\ ();\quad \textbf{!}\ ()\ n\ \mathrm{S}\textsc{q}; & \\
\quad \mathrm{S}\textsc{q}\ m\ () = \mathrm{S}\textsc{q}\ \mathrm{Z}\ m\ \mathrm{S}\textsc{q}; & (\textsc{sq–begin}) \\
\quad \mathrm{S}\textsc{q}\ s\ (\mathrm{S}k)\ \mathrm{S}\textsc{q} = \mathrm{S}\textsc{q}\ (\mathrm{S}s'')\ k\ \mathrm{S}\textsc{q}: & (\textsc{sq–step}) \\
\qquad\quad + \ s\ k\ () = ()\ s'\ k\ +. & \text{-- } s' \leftarrow s\ + k \\
\qquad\quad + \ s'\ k\ () = ()\ s''\ k\ +. & \text{-- } s'' \leftarrow s' + k \\
\quad \mathrm{S}\textsc{q}\ n\ \mathrm{Z}\ \mathrm{S}\textsc{q} = ()\ n\ \mathrm{S}\textsc{q}; & (\textsc{sq–end})
\end{array}
$$

$$
\textbf{!}\quad \mathrm{S}\textsc{q}\ 3\ () = \mathrm{S}\textsc{q}\ \mathrm{Z}\ 3\ \mathrm{S}\textsc{q} \underset{*}{=} \mathrm{S}\textsc{q}\ 5\ 2\ \mathrm{S}\textsc{q} \underset{*}{=} \mathrm{S}\textsc{q}\ 8\ 1\ \mathrm{S}\textsc{q} \underset{*}{=} \mathrm{S}\textsc{q}\ 9\ \mathrm{Z}\ \mathrm{S}\textsc{q} = ()\ 9\ \mathrm{S}\textsc{q}\quad \textbf{!}
$$

$*$ $(\textsc{sq–step})$



computational inertia

$$t_1 \overset{r}{=} t_2 \overset{s}{=} t_3$$

$$s \neq r^{-1}$$

reversible programming

א: motivation, semantics, & tutorial

א: advanced features & properties

alethe + א concurrency

Alethe

```
% > 3 4 + x y
() 7 4 +
  x → 7
  y → 4
% < a b + 7 4
+ 3 4 ()
  a → 3
  b → 4
%
```

```
a   Z   +   a     Z;
a (S b) + (S c) (S b):
    a b + c b.
```

```
% > 7 ^2 z
() 49 ^2
  z → 49
% < c ^2 49
^2 7 ()
  c → 7
%
```

```
n ^2 n2:
  ! Go n     Z = Go Z n2.
    Go (S n) m = Go n (S k):
        m n + l n.
        l n + k n.
```

```
% > 2 9 `Pair` r
() 75 Pair
  r → 75
% < p q `Pair` 75
Pair 2 9 ()
  p → 2
  q → 9
%
```

```
a b `Pair` n:
  ! Go   n     Z   Z = Go Z   a    b.
    Go (S n)   Z   b = Go n (S b)  Z;
    Go (S n) (S a) b = Go n   a  (S b);
```

Hannah Earley — 2020 — Alethe Interpreter — https://github.com/hannah-earley/alethe-repl

Concurrency

$$
\begin{aligned}
&(\textsc{pattern term}) && \pi ::= \textsc{sym} \mid \textsc{var} \mid (\,\pi^*\,) \\
&(\textsc{rule}) && \rho ::= \pi^* = \pi^* \\
&(\textsc{definition}) && \delta ::= \rho : \rho^{*}_{.} \mid \mathbf{!}\ \pi^*;
\end{aligned}
$$

$$
\begin{aligned}
&(\textsc{pattern term}) && \pi ::= \textsc{sym} \mid \textsc{var} \mid (\,\pi^*\,) \\
&(\textsc{rule}) && \Pi ::= \pi : \pi^* \mid \textsc{var}' : \pi^* \\
&(\textsc{definition}) && \delta ::= \{\Pi^*\} = \{\Pi^*\} : \Pi^{*}_{.} \mid \mathbf{!}\ \pi^*;
\end{aligned}
$$

$$
\textsc{Alice}\ [x \cdot x\cancel{s}] = \begin{Bmatrix} \textsc{Alice}\ x\cancel{s} \\ \textsc{Courier}\ x \end{Bmatrix}; \qquad \begin{Bmatrix} \textsc{Bob}\ y\cancel{s} \\ \textsc{Courier}\ y \end{Bmatrix} = \textsc{Bob}\ [y \cdot y\cancel{s}];
$$

Properties + Future Work

- r-Turing Complete
- Confluent Semantics
- Concurrent variant
- Interpreter written

- Implement & study concurrent variant
- Type system
- Apply to molecular programming

Thank you!

UNIVERSITY OF CAMBRIDGE

Vaire

MICKLEM LAB

UKRI

Engineering and Physical Sciences Research Council

Department of Applied Mathematics and Theoretical Physics (DAMTP)

Hannah Earley — 2016–2021 — 'Modelling approaches to molecular computation' — EPSRC Project Reference 1781682
Hannah Earley — 2020/2022 — 'The ℵ-Calculus' — arXiv/Proceedings of RC22