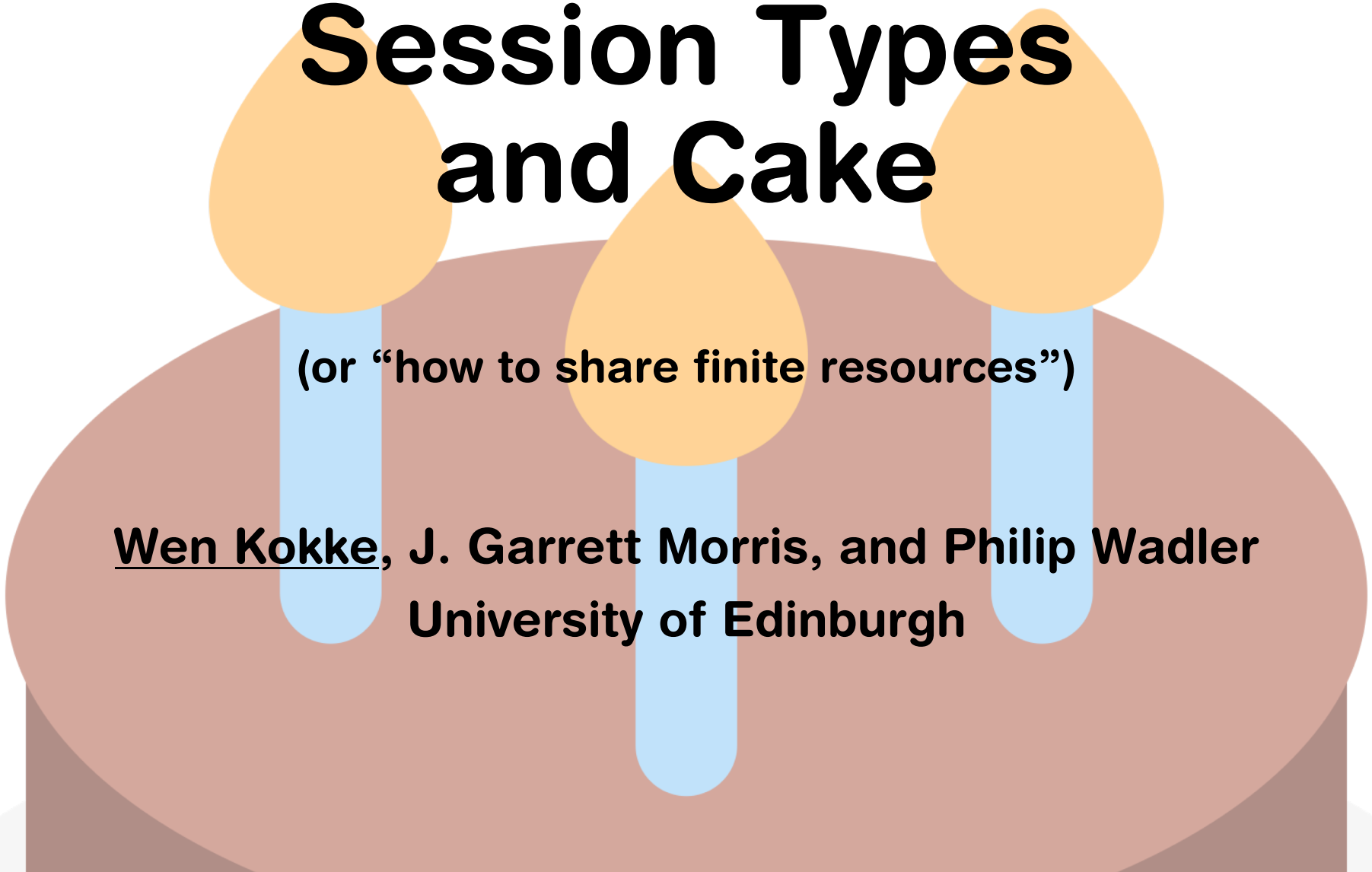
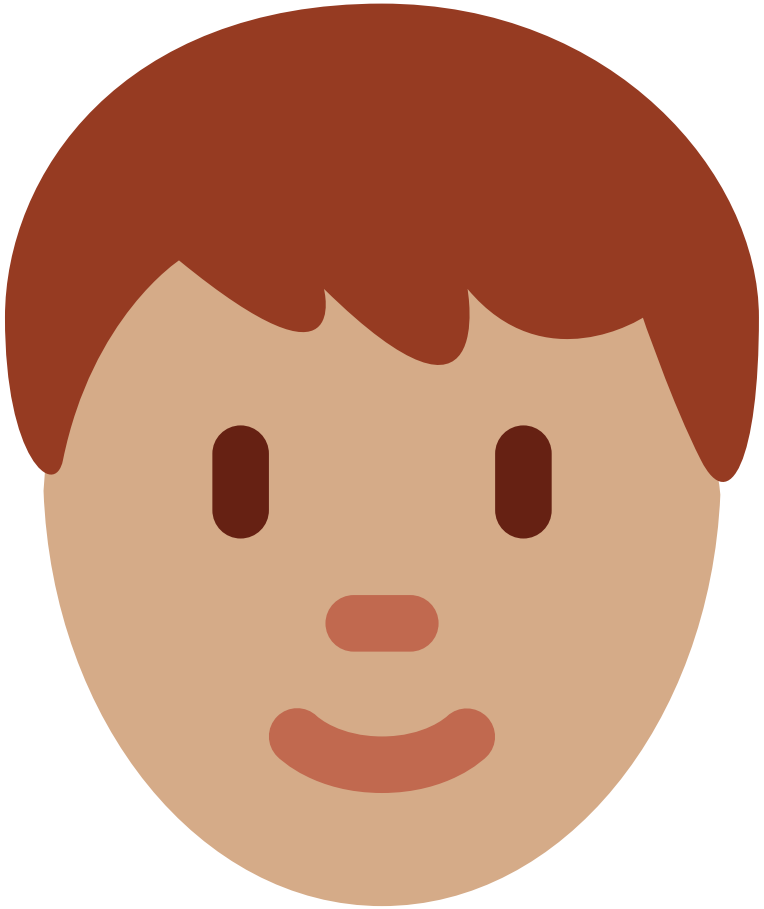
Session Types and Cake

(or “how to share finite resources”)

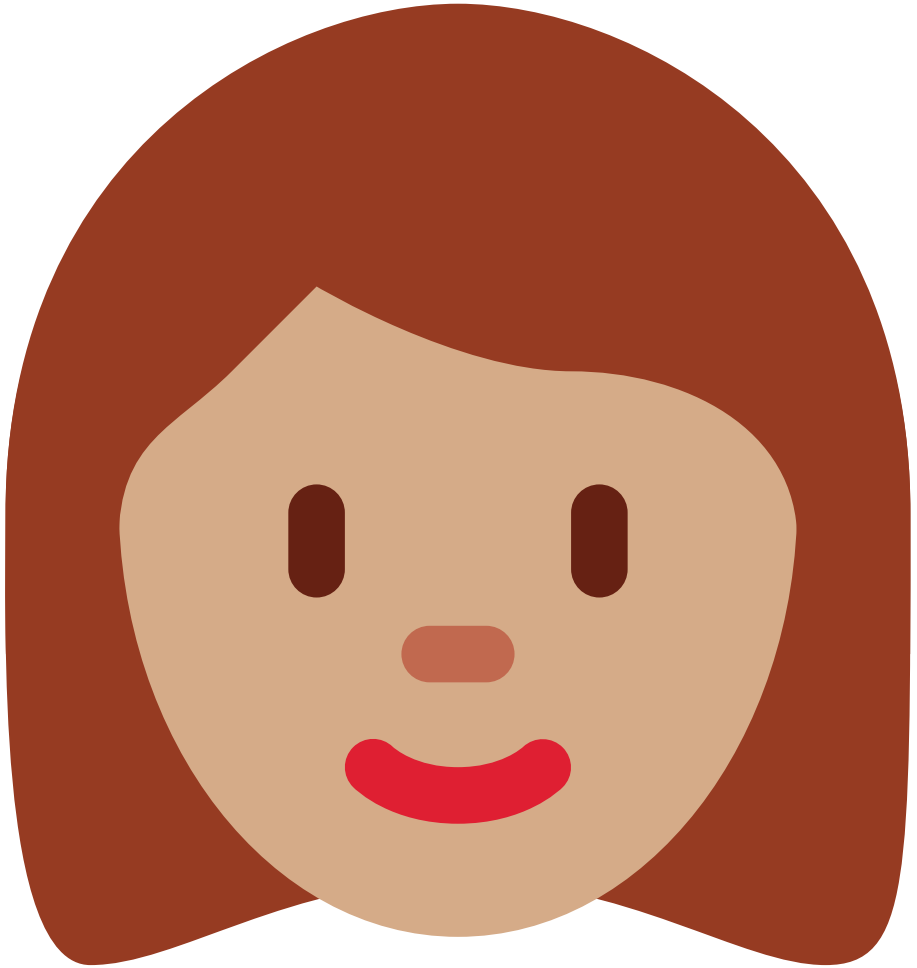
Wen Kokke, J. Garrett Morris, and Philip Wadler
University of Edinburgh



First, a story.



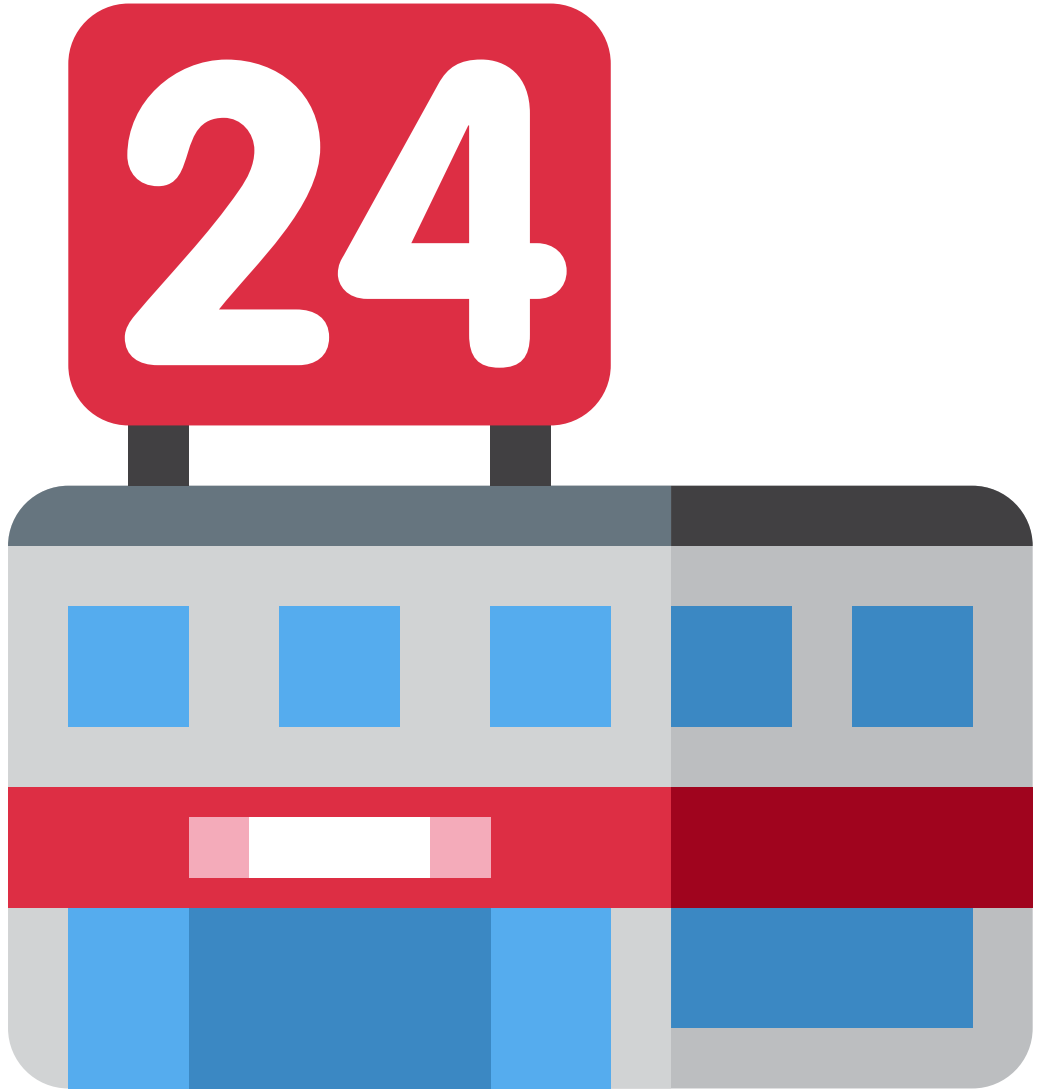
This is Ami.
They love cake.



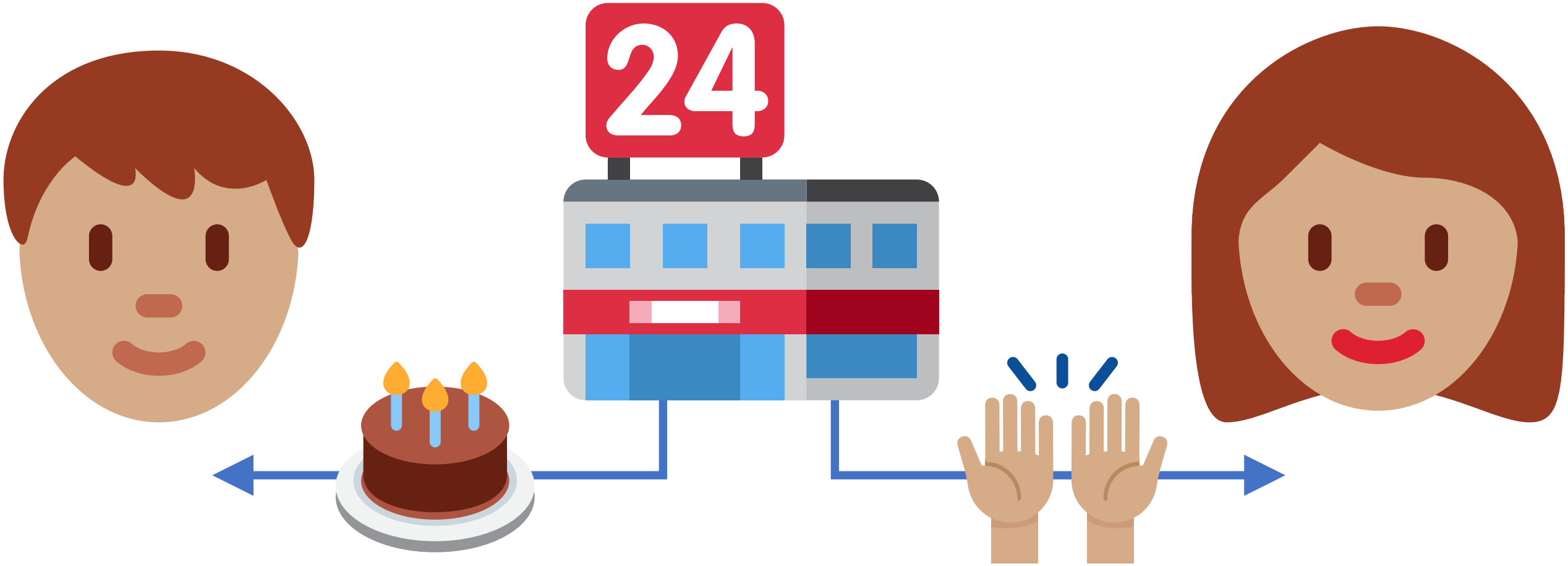
This is Boé.
She loves cake too.

This is a store.
It sells cake.

There is only
one cake left.

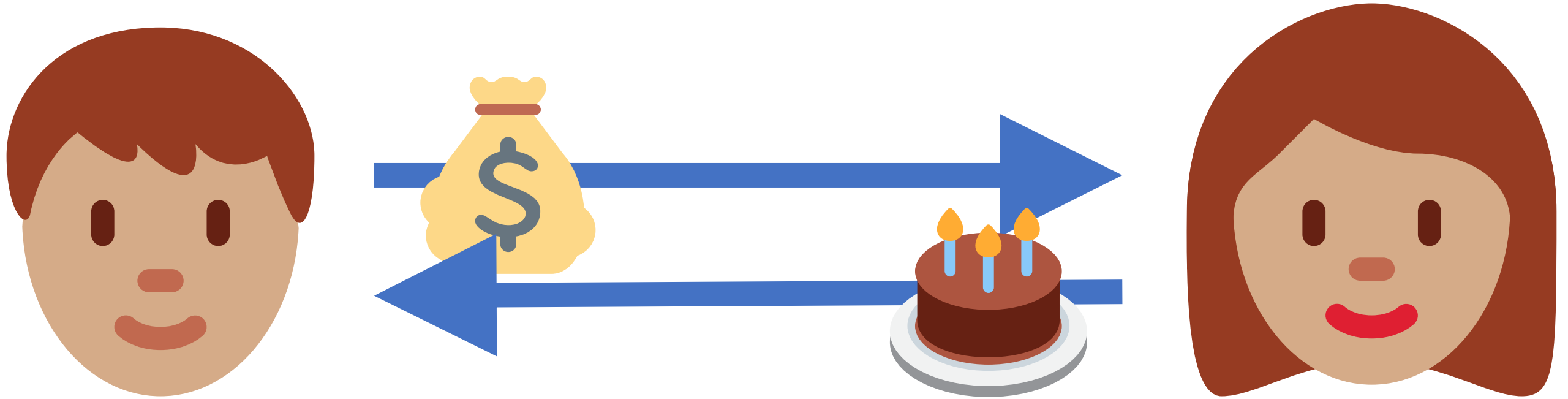


Ami and Boé have to *race*.
That's ok. The store doesn't mind.



So…
Races are good sometimes!

And deadlocks are bad.
I’m sure we all know.



This is Classical Processes.

$$\frac{}{x{\leftrightarrow}y \vdash x : A, y : A^{\perp}}\ \textsc{Ax} \qquad \frac{P \vdash \Gamma, x : A \qquad Q \vdash \Delta, y : A^{\perp}}{(\nu xy)(P \mid Q) \vdash \Gamma, \Delta}\ \textsc{Cut}$$

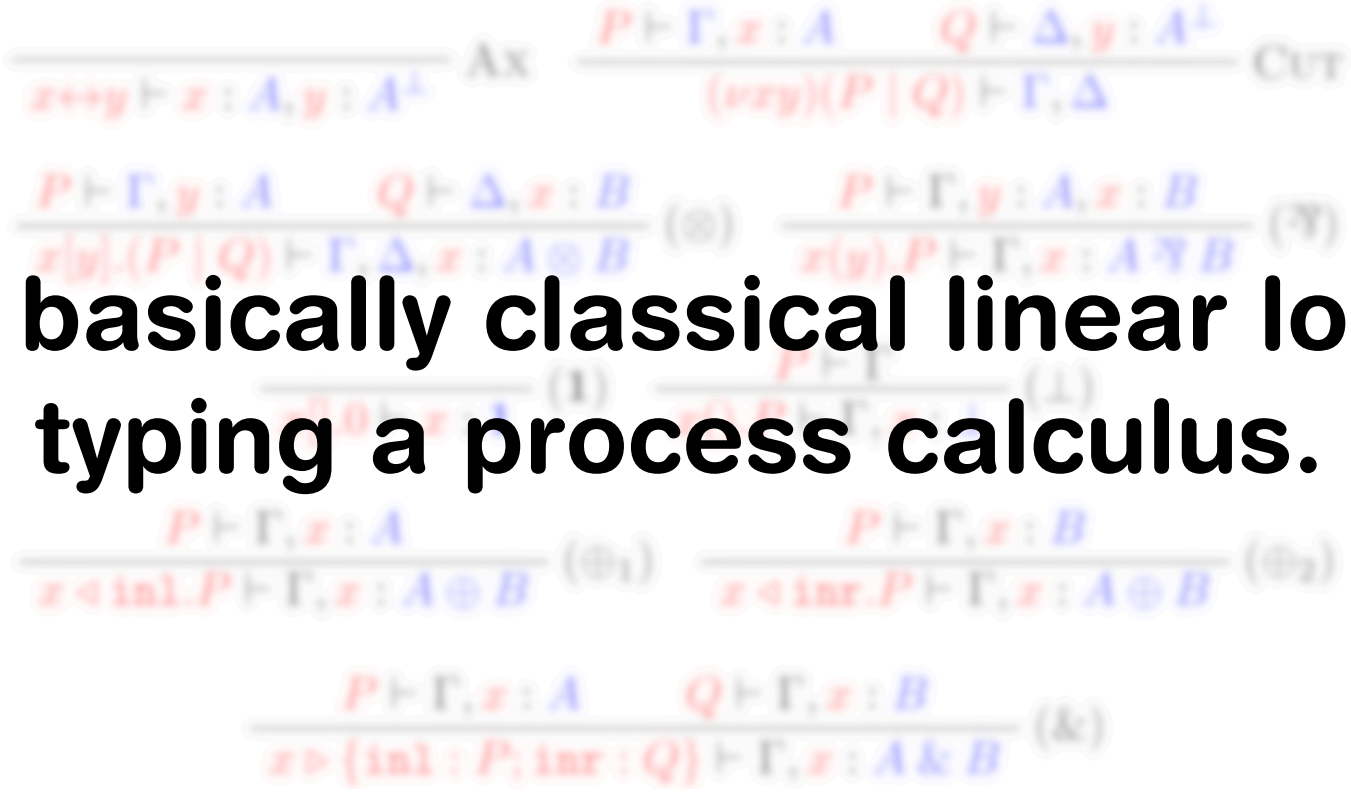
$$\frac{P \vdash \Gamma, y : A \qquad Q \vdash \Delta, x : B}{x[y].(P \mid Q) \vdash \Gamma, \Delta, x : A \otimes B}\ (\otimes) \qquad \frac{P \vdash \Gamma, y : A, x : B}{x(y).P \vdash \Gamma, x : A \parr B}\ (\parr)$$

$$\frac{}{x[].0 \vdash x : \mathbf{1}}\ (\mathbf{1}) \qquad \frac{P \vdash \Gamma}{x().P \vdash \Gamma, x : \bot}\ (\bot)$$

$$\frac{P \vdash \Gamma, x : A}{x \triangleleft \mathtt{inl}.P \vdash \Gamma, x : A \oplus B}\ (\oplus_1) \qquad \frac{P \vdash \Gamma, x : B}{x \triangleleft \mathtt{inr}.P \vdash \Gamma, x : A \oplus B}\ (\oplus_2)$$

$$\frac{P \vdash \Gamma, x : A \qquad Q \vdash \Gamma, x : B}{x \triangleright \{\mathtt{inl} : P; \mathtt{inr} : Q\} \vdash \Gamma, x : A \,\&\, B}\ (\&)$$

This is Classical Processes.



It’s basically classical linear logic, typing a process calculus.

This is CP's syntax.

$$
\begin{array}{rcl}
P, Q & ::= & (\nu xy)(P \mid Q) \\
& \mid & x[y].(P \mid Q) \\
& \mid & x(y).P \\
& \mid & \ldots
\end{array}
$$

We can make new channels *and split*.

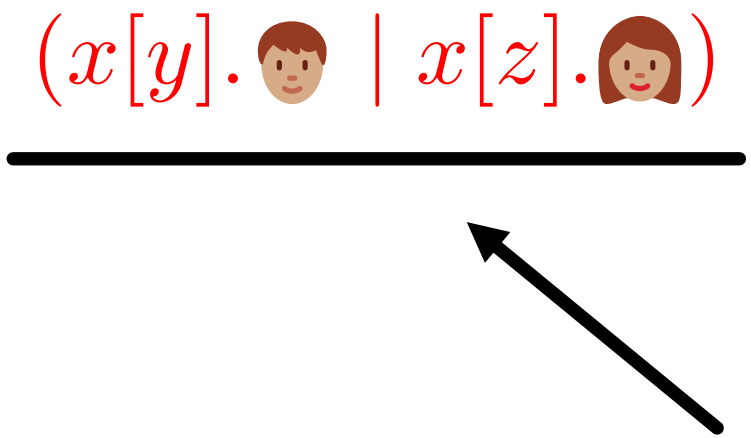
We can send a message *and split*.

We can receive a message.

Oh no! Something's wrong.

$(\nu xy)($ $(x[y].👦 \mid x[z].👩)$ $\mid$ $y(🎂).y(🙌).🏨$ $)$

Oh no! Something's wrong.



pool of clients

No parallel composition?!

$$\frac{P \vdash \Gamma, x : A \qquad Q \vdash \Delta, y : A^{\perp}}{(\nu xy)(P \mid Q) \vdash \Gamma, \Delta}\ \textsc{Cut}$$

$$\frac{P \vdash \Gamma, y : A \qquad Q \vdash \Delta, x : B}{x[y].(P \mid Q) \vdash \Gamma, \Delta, x : A \otimes B}\ (\otimes)$$

Just doing it is dangerous!

$$\frac{P \vdash \Gamma \qquad Q \vdash \Delta}{P \mid Q \vdash \Gamma, \Delta}\ \textsc{Mix}$$

$$\frac{P \vdash \Gamma, x : A, y : A^{\perp}}{(\nu xy)P \vdash \Gamma}\ \textsc{Cut} \qquad \frac{P \vdash \Gamma, y : A, x : B}{x[y].P \vdash \Gamma, x : A \otimes B}\ (\otimes)$$

Remember what's in parallel!

$$\mathcal{G}, \mathcal{H} \quad ::= \quad \varnothing \quad \mid \quad \Gamma \mid \mathcal{G}$$

$$\frac{P \vdash \mathcal{G} \qquad Q \vdash \mathcal{H}}{P \mid Q \vdash \mathcal{G} \mid \mathcal{H}}\ \text{H-Mix}$$

$$\frac{P \vdash \mathcal{G} \mid \Gamma, x : A \mid \Delta, y : A^{\perp}}{(\nu xy)P \vdash \mathcal{G} \mid \Gamma, \Delta}\ \textsc{Cut} \qquad \frac{P \vdash \Gamma, y : A \mid \Delta, x : B}{x[y].P \vdash \Gamma, \Delta, x : A \otimes B}\ (\otimes)$$

This is Hypersequent CP.

$$\frac{}{x{\leftrightarrow}y \vdash x : A, y : A^{\perp}}\ \textsc{Ax} \qquad \frac{P \vdash \mathcal{G} \mid \Gamma, x : A \mid \Delta, y : A^{\perp}}{(\nu xy)P \vdash \mathcal{G} \mid \Gamma, \Delta}\ \textsc{Cut} \qquad \frac{P \vdash \mathcal{G} \qquad Q \vdash \mathcal{H}}{P \mid Q \vdash \mathcal{G} \mid \mathcal{H}}\ \textsc{H-Mix}$$

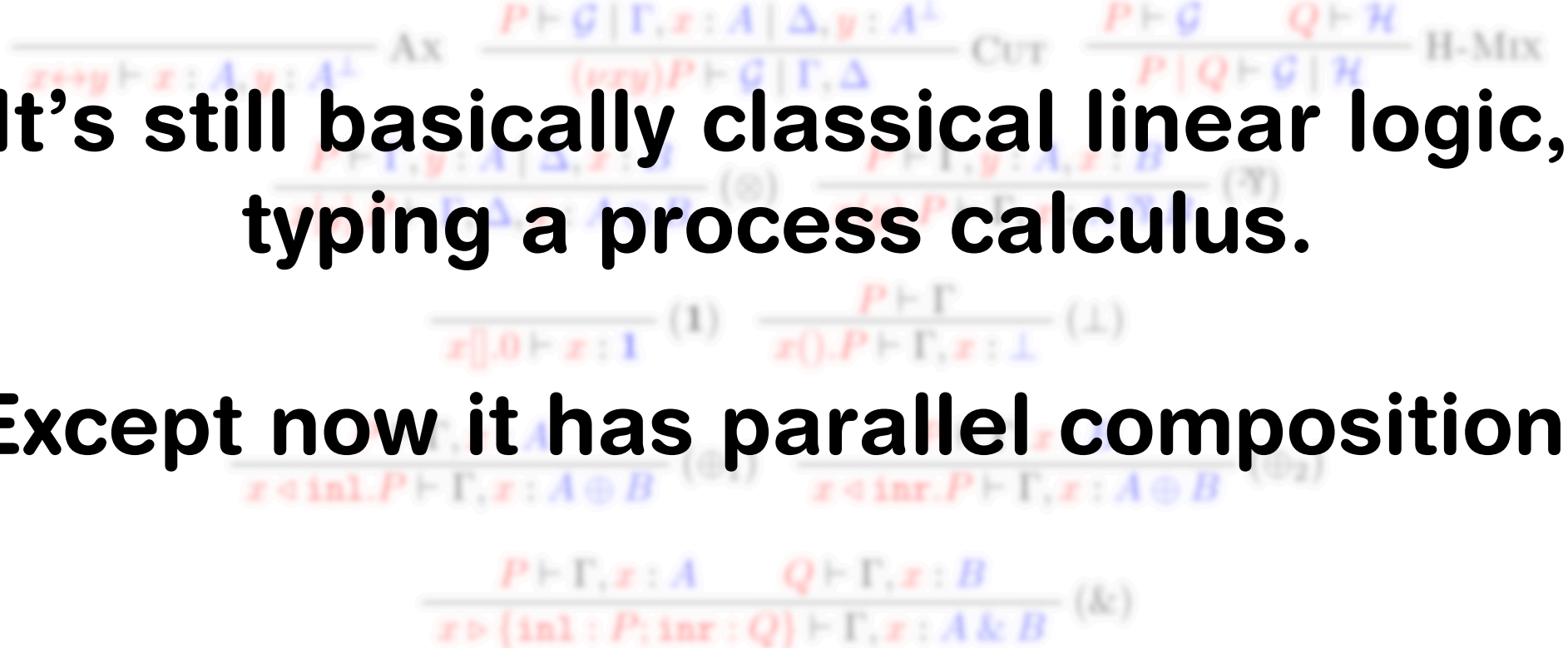
$$\frac{P \vdash \Gamma, y : A \mid \Delta, x : B}{x[y].P \vdash \Gamma, \Delta, x : A \otimes B}\ (\otimes) \qquad \frac{P \vdash \Gamma, y : A, x : B}{x(y).P \vdash \Gamma, x : A \parr B}\ (\parr)$$

$$\frac{}{x[].0 \vdash x : \mathbf{1}}\ (\mathbf{1}) \qquad \frac{P \vdash \Gamma}{x().P \vdash \Gamma, x : \bot}\ (\bot)$$

$$\frac{P \vdash \Gamma, x : A}{x \triangleleft \mathtt{inl}.P \vdash \Gamma, x : A \oplus B}\ (\oplus_1) \qquad \frac{P \vdash \Gamma, x : B}{x \triangleleft \mathtt{inr}.P \vdash \Gamma, x : A \oplus B}\ (\oplus_2)$$

$$\frac{P \vdash \Gamma, x : A \qquad Q \vdash \Gamma, x : B}{x \triangleright \{\mathtt{inl} : P; \mathtt{inr} : Q\} \vdash \Gamma, x : A \,\&\, B}\ (\&)$$

This is Hypersequent CP.



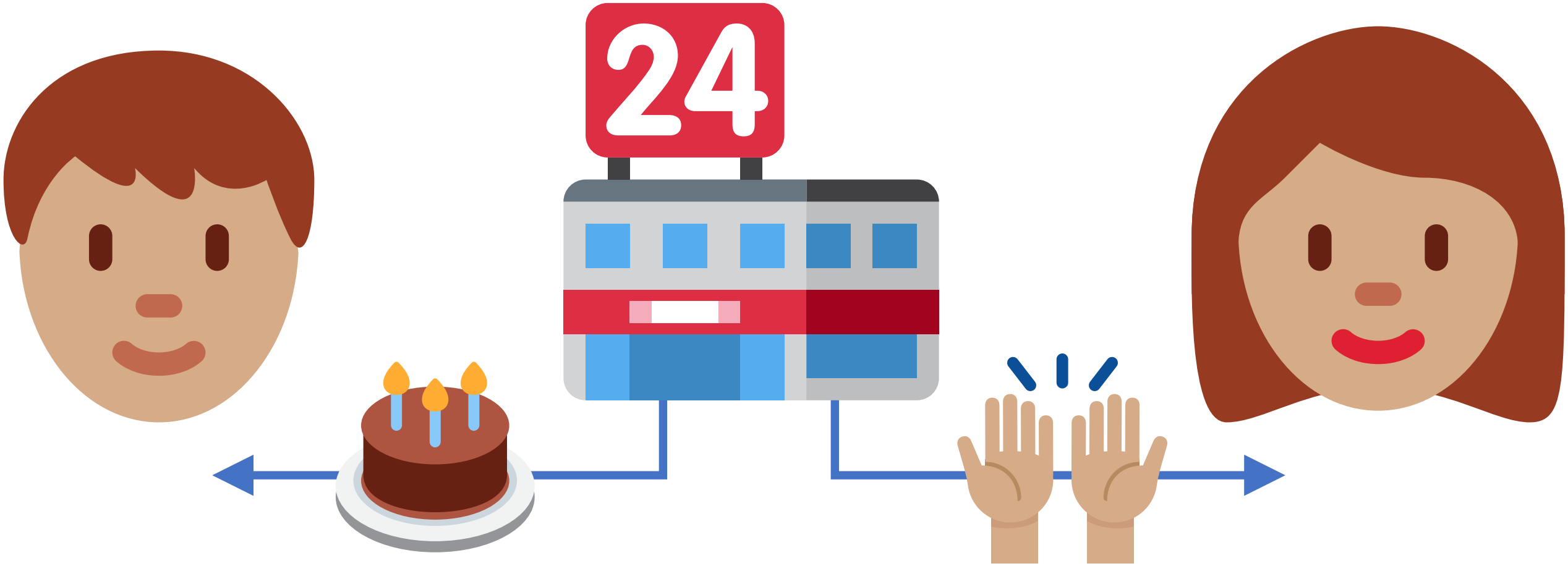
It's still basically classical linear logic, typing a process calculus.

Except now it has parallel composition.

This is HCP's syntax.

$$
\begin{array}{lcll}
P, Q & ::= & (\nu xy)P & \text{We can make new channels.} \\
 & | & P \mid Q & \text{We can split.} \\
 & | & x[y].P & \text{We can send message y over channel x.} \\
 & | & x(y).P & \text{We can receive message y over channel x.} \\
 & | & \ldots &
\end{array}
$$

Ami and Boé have to *race*.
That's ok. The store doesn't mind.

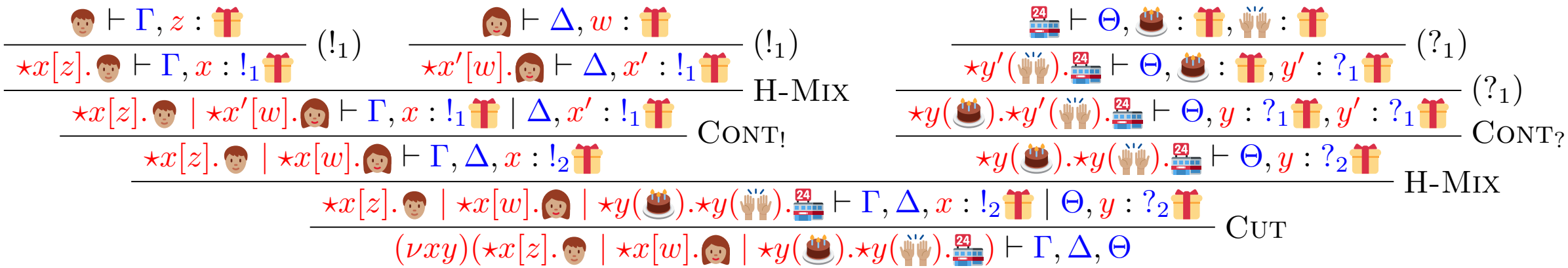


This is what we've added.

$$\frac{P \vdash \Gamma, y : A}{\star x[y].P \vdash \Gamma, x : !_1 A}\ (!_1) \qquad \frac{P \vdash \Gamma, y : A}{\star x(y).P \vdash \Gamma, x : ?_1 A}\ (?_1)$$

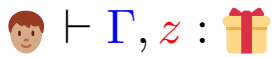
$$\frac{P \vdash \mathcal{G} \mid \Gamma, x : !_n A \mid \Delta, x' : !_m A}{P \vdash \mathcal{G} \mid \Gamma, \Delta, x : !_{n+m} A}\ \textsc{Cont}_! \qquad \frac{P \vdash \mathcal{G} \mid \Gamma, x : !_n A, x' : !_m A}{P \vdash \mathcal{G} \mid \Gamma, x : !_{n+m} A}\ \textsc{Cont}_?$$

This is our example race.



(🎁: Could be cake, could be disappointing.)

This is our example race.



(🎁: Could be cake, could be disappointing.)

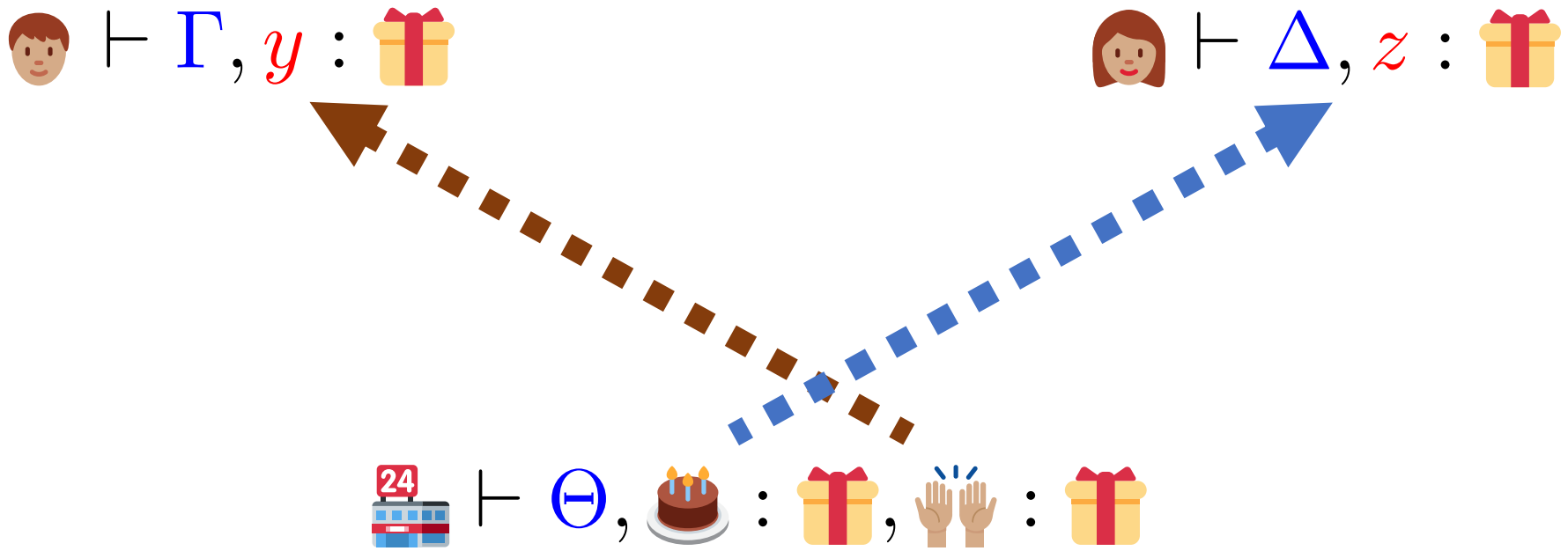
Ami gets the cake.

(🎁: Could be cake, could be disappointing.)

Ami gets the cake.

(🎁: Could be cake, could be disappointing.)

Boé gets the cake.



(🎁: Could be cake, could be disappointing.)

And so we had races, but no deadlocks.

What's not yet right?

- **No recursion** 😔
- **No infinite interactions** 😩

But:

Conflation confers concurrency.

What is ‘conflation’?

- Makes two duals isomorphic
- Conflation of !/? confers:
  - No lock freedom 😩
  - No termination 😔
  - Concurrent shared state 🙂
  - Recursion 😄

Other mechanisms?

- **Non-deterministic local choice**

$$P + Q \longrightarrow P$$
$$P + Q \longrightarrow Q$$

- **Equally expressive, not equal**
- **Translate from O(1), to O(n!)**

Other mechanisms?

Manifest Sharing:

- **Recursion** 🙂
- **No deadlock freedom** 😔
- **Cannot interleave requests** 😩

Non-deterministic HCP:

- Simple extension of HCP
- Finite non-determinism
- Deadlock free

Thanks!