"Nested parallelism"

$\langle (a := f() \;||\; b := g());\ r := a+b, \{\} \rangle$
$\mapsto^* \langle (skip \;||\; skip);\ r := a+b, \{a=12, b=42\} \rangle$
$\mapsto^2 \langle r := a+b, \{a=12, b=42\} \rangle$
$\mapsto \langle skip, \{a=12, b=42, r=54\} \rangle$

A little more realistic

$s ::= \ldots \mid s \| s \mid wait(a)$
$a \in Threads$

$$\frac{a \text{ fresh}}{\langle s_1 \| s_2, \sigma \rangle \mapsto \langle s_1; wait(a), \sigma[a \mapsto s_1] \rangle}$$

(a fresh: Not used before)

$$\frac{\sigma(a) = skip}{\langle wait(a), \sigma \rangle \mapsto \langle skip, \sigma \rangle}$$

$$\frac{\langle s_t, \sigma \rangle \mapsto \langle s_t', \sigma' \rangle}{\langle s, \sigma[a \mapsto s_t] \rangle \mapsto \langle s, \sigma[a \mapsto s_t'] \rangle}$$

Why is the main thread special?
Instead of $\langle s, \sigma \rangle$, start with $\sigma[main \mapsto s]$

$$\frac{\langle s_a, \sigma \rangle \mapsto \langle s_a', \sigma' \rangle}{\sigma[a \mapsto s_a] \mapsto \sigma'[a \mapsto s_a']}$$

"Actual" parallelism:

$$\frac{\forall i \in [1,n].\ \langle s_i, \sigma \rangle \mapsto \langle s_i', \sigma_i' \rangle \quad 1 \le P \quad \sigma := merge(\sigma \ldots \sigma_n)}{\sigma[a_1 \mapsto s_1] \cdots [a_n \mapsto s_n] \mapsto \sigma'[a_1 \mapsto s_1'] \cdots [a_n \mapsto s_n']}$$

(P: # of processors; merge: resolve data races)

Not limited to nested parallelism!

$e ::= \ldots \mid a$

$s ::= \ldots \mid x := \text{spawn } s \mid \text{wait } x$

$$\frac{a \notin \text{esh}}{\langle x := \text{spawn } s, \sigma \rangle \mapsto \langle x := a, \sigma[a \mapsto s] \rangle}$$

$$\frac{\sigma(a) = a \quad \sigma(a) = \text{skip}}{\langle \text{wait } x, \sigma \rangle \mapsto \langle \text{skip}, \sigma \rangle}$$

```
while 1
 do
   add- := listen l);
   _ := spawn (handle conn add-)
 od
```

Note: If $\sigma(\sigma(x)) \neq \text{skip}$, then wait x is stuck. - Progress must change

If $\vdash \sigma$ and $\Gamma \vdash s$ ok then $\langle s, \sigma \rangle$ final or $\langle s, \sigma \rangle \mapsto \langle s', \sigma' \rangle$
or $s = \text{wait } x$ and $\sigma(\sigma(x)) \neq \text{skip}$

So does that mean $\sigma(\sigma(x))$ can step? Yes, as we've set things up. But be careful!

```
x := spawn (wait y)      Deadlock!
y := spawn (wait x)
```

Futures / Promises

$s ::= \ldots \mid x := \text{future } e \;\; e$

$\langle x := \text{future } e, \sigma \rangle \mapsto \langle \text{skip}, \sigma[a \mapsto x := e][x \mapsto \_] \rangle$

(a: placeholder)

$s \triangleq x := \text{future}(f())$ - spawns a new thread
$y := \text{future}(g())$ - spawns a new thread
$z := x + y$ - waits until x and y are filled in
- same problem w/ progress

$\langle s, \{\} \rangle \mapsto^* \langle z := x+y, \{a \mapsto x := f(), b \mapsto y := g(), x \mapsto -, y \mapsto - \rangle$
$\mapsto^* \langle z := x+y, \{a \mapsto x := 8, b \mapsto y := 5, x \mapsto -, y \mapsto - \rangle$
$\mapsto^* \langle z := x+y, \{a \mapsto \text{skip}, b \mapsto \text{skip}, x \mapsto 8, y \mapsto 5 \rangle$
$\mapsto^* \langle \text{skip}, \{a \mapsto \text{skip}, b \mapsto \text{skip}, x \mapsto 8, y \mapsto 5, z \mapsto 13 \rangle$

a, b computed in parallel $\Rightarrow$ Parallelism
a & b computed when trying to read $\Rightarrow$ Laziness!