Day 3

1. Evaluation: Relations on Terms

The `eval` function represents our first attempt to formally give meaning to terms of our language. Let's try to pluck it out of the context of a Haskell program.

$$\frac{}{z \Downarrow z} \qquad \frac{t_1 \Downarrow z_1 \quad t_2 \Downarrow z_2}{t_1 + t_2 \Downarrow z_1 + z_2} \qquad \frac{t_1 \Downarrow z_1 \quad t_2 \Downarrow z_2}{t_1 \times t_2 \Downarrow z_1 \times z_2}$$

Evaluation ($\Downarrow$) is a *relation* between *terms* and *integers*... mathematically $\Downarrow \subseteq \mathcal{T} \times \mathbb{Z}$. The rules give a schematic view of that relation. We can look at the expected contents of the relation:

? ? ? term 2 => 2
syntax ==> semantic
下划线代表language

$(2, 2) \in \Downarrow$

$(2, 3) \notin \Downarrow$

$((2 + 2) \times 3, 12) \in \Downarrow$

$((2 + 2) \times 3, 8) \notin \Downarrow$

How would we go about demonstrating some of these? Recall inference trees:

$$\dfrac{\dfrac{\dfrac{}{2 \Downarrow 2} \quad \dfrac{}{2 \Downarrow 2}}{2 + 2 \Downarrow 4} \quad \dfrac{}{3 \Downarrow 3}}{(2 + 2) \times 3 \Downarrow 12}$$

It's more interesting to wonder how we would demonstrate that things *aren't* in $\Downarrow$.

We can also talk about *properties* of $\Downarrow$. (Recall notions of "well-defined" from first day's discussion.)

- Evaluation is *total*: for every $t \in \mathcal{T}$, there is some $z \in \mathbb{Z}$ s.t. $t \Downarrow z$.
- Evaluation is *deterministic*: if $t \Downarrow z_1$ and $t \Downarrow z_2$, then $z_1 = z_2$. (I.e., evaluation is a *function*.)

确定性

How do we prove these things? By *induction* on the assumptions. Key idea: structure of proof parallels structure of data.

> Q: Why bother doing proofs about programming languages? They are almost always boring if the definitions are right.
> A: The definitions are almost always wrong.

We could also talk about the relationship of our pen-and-paper notion of evaluation to our Haskell model. Is it the case that if $t \Downarrow z$, then in Haskell `eval t` will reduce to z?

How do these properties relate to properties of "real" languages? Would we expect the evaluation relation for C or Haskell (assuming we know what such a thing would look like) to be total or deterministic? Why or why not?

2. Multiple Notions of Evaluation

The evaluation relation in the previous section seems to capture our intuitive understanding of arithmetic expressions. What other interpretations are possible?

One possibility—arithmetic modulo b (where b is probably something like 2^{32} or 2^{64}). So $\Downarrow_b \subset \mathcal{T} \times \mathbb{Z}_b$ (or, if you prefer number theory notation, $\Downarrow_b \subset \mathcal{T} \times (\mathbb{Z}/b\mathbb{Z})$). 包含多种可能性

$$\frac{}{n \Downarrow_b z}\ (n = z \bmod b) \qquad \frac{t_1 \Downarrow_b z_1 \quad t_2 \Downarrow_b z_2}{t_1 + t_2 \Downarrow_b z_3}\ (z_1 + z_2 = z_3 \bmod b)$$

$$\frac{t_1 \Downarrow_b z_1 \quad t_2 \Downarrow_b z_2}{t_1 \times t_2 \Downarrow_b z_3}\ (z_1 \times z_2 = z_3 \bmod b)$$

Does this evaluation relation have the same properties as "normal" evaluation?

Let's define one more notion of evaluation. Suppose we're only concerned about whether the result of evaluating something is positive, zero, or negative. (Why might we be concerned about such things?) We could define something like the following. Let $\mathcal{S} = \{-, 0, +\}$, and let S range over *subsets of* $\{-, 0, +\}$ (i.e., $S \in \mathcal{P}(\mathcal{S})$. First, let's have a few lookup tables:

$\hat{+}$	$-$	0	$+$
$-$	$\{-\}$	$\{-\}$	$\{-, 0, +\}$
0	$\{-\}$	$\{0\}$	$\{+\}$
$+$	$\{-, 0, +\}$	$\{+\}$	$\{+\}$

$\hat{\times}$	$-$	0	$+$
$-$	$\{+\}$	$\{0\}$	$\{-\}$
0	$\{0\}$	$\{0\}$	$\{0\}$
$+$	$\{-\}$	$\{0\}$	$\{+\}$

So we have that $\hat{+}, \hat{\times} \in \mathcal{S} \times \mathcal{S} \to \mathcal{P}(\mathcal{S})$. Now we can define a new evaluation relation ($\Downarrow_\pm \subseteq \mathcal{T} \times \mathcal{P}(\mathcal{S})$):

$$\frac{}{n \Downarrow_\pm \{-\}}\ (n < 0) \qquad \frac{}{0 \Downarrow_\pm \{0\}} \qquad \frac{}{n \Downarrow_\pm \{+\}}\ (n > 0)$$

$$\frac{t_1 \Downarrow_\pm S_1 \quad t_2 \Downarrow_\pm S_2}{t_1 + t_2 \Downarrow_\pm \{s_3 \mid s_1 \in S_1, s_2 \in S_3, s_3 \in s_1 \hat{+} s_2\}}$$ wrong for s2,s2 belong to S2

$$\frac{t_1 \Downarrow_\pm S_1 \quad t_2 \Downarrow_\pm S_2}{t_1 \times t_2 \Downarrow_\pm \{s_3 \mid s_1 \in S_1, s_2 \in S_3, s_3 \in s_1 \hat{\times} s_2\}}$$

Alternative presentation of latter two rules:

$$\frac{t_1 \Downarrow_\pm S_1 \quad t_2 \Downarrow_\pm S_2}{t_1 + t_2 \Downarrow_\pm \bigcup\{s_1 \hat{+} s_2 \mid s_1 \in S_1, s_2 \in S_2\}} \qquad \frac{t_1 \Downarrow_\pm S_1 \quad t_2 \Downarrow_\pm S_2}{t_1 \times t_2 \Downarrow_\pm \bigcup\{s_1 \hat{\times} s_2 \mid s_1 \in S_1, s_2 \in S_2\}}$$