

Test-based Solution Filtering for Program Synthesis

Ziteng (Zetten) Wang, University of California, San Diego

Advised by Nadia Polikarpova



	(2)	\xs -> (\$) (Last []) xs
	(3)	\xs -> last (zip [] xs)
"Bad"! -	(4)	<pre>\xs -> head (zip [] xs)</pre>
	(5)	<pre>\xs -> last (zip xs [])</pre>
	(6)	$xs \rightarrow$ (head (xs++xs), head xs)
		• • •
	(15)	<pre>\xs -> (head xs, last xs)</pre>

Invalid results never return meaningful outputs

Eliminate Invalid Programs NOTCRASH(f)

Eliminate Duplicates NOTDUPLICATE(f)

- Always crash e.g. last []
- Always diverge e.g. last (repeat x)
- Duplicate results are functionally equivalent to another result
 - E.g. head xs and head (xs ++ xs)

Most uninteresting results can be classified into these categories... We will filter them by testing!

last (zip [] xs)

head (zip [] xs)

Prop. 1 \forall args, f(args) either fails or diverges. Prop. 2 \forall *args*, *f*(*args*) returns.

Case (prop_1, prop_2) of success, _ -> reject _, success -> accept (total) _ -> accept (partial)

Prop. 1 $\forall f' \in \mathbb{S}, f \neq f' \text{ where } \mathbb{S} \text{ is the}$ set of all previous results

Case prop_1 of failure -> reject success -> accept

Challenges: Does it ever fail? 4.

- Infinite Data Structures: Evaluation
 - $x \rightarrow$ (repeat x, head [])
- Higher-order Functions: Verification and Input-generation
 - \f -> f (head [])



discarded

xs: [a] -> (a, a) Query	f: (a->b) -> g	: (a->c) -> x	: a -> (b, c) Q	f: (a->Either	b c) -> xs: [a] -> ([b], [c]) Quer
(,) (head xs) (head xs)	(,) (f x) (g x)			partitionEithers (map f xs)	
[-1, -2] -> (-1, -1) [] -> error	{>0} {>0} 3 -> (0, 0)		{>Right 0} [] -> ([], [0])		
(,) (last xs) (last xs)	discardod	<pre>(,) (f (fromJust Nothing)) (g x) (,) (g (head [])) (f x) (,) (g (last [])) (f x)</pre>		<pre>partitionEithers (repeat (f (head xs))</pre>	
[-1, -2] -> (-2, -2) [] -> error	urscurueu			{>Right 0} [] -> ([], [0, 0,])	
(,) (head xs) (last xs) [-1, -2] -> (-1, -2)				discarded	<pre>partitionEithers (repeat (f (last xs)) curry (last []) xs f</pre>
L] -> error					