## A Naïve Oracle Based on Enumeration

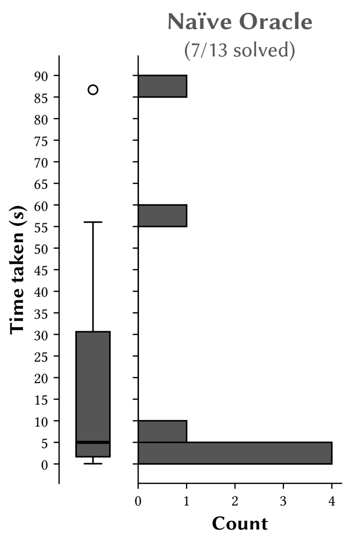


Fig. A1. **Completion rates for benchmarks with *finitely* many solutions (the Fin suite).**

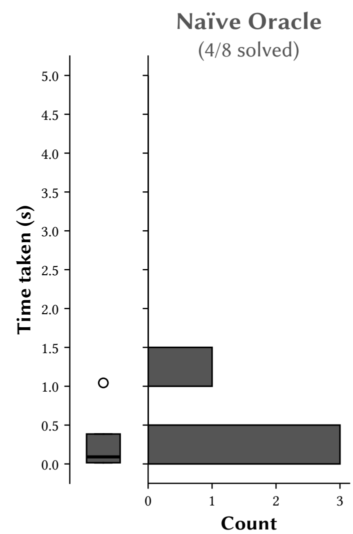


Fig. A2. **Completion rates for benchmarks with *infinitely* many solutions (the Inf suite).**

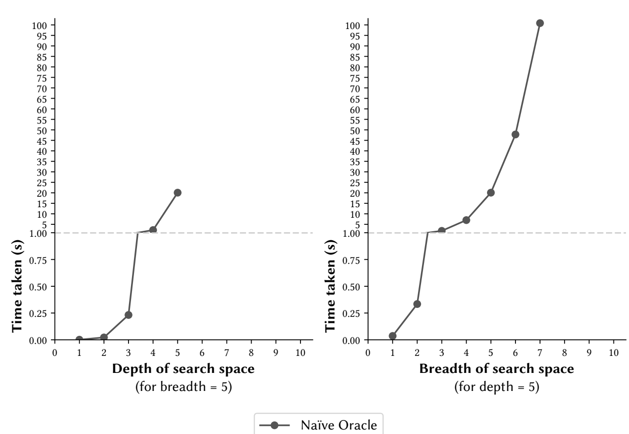


Fig. A3. **Time to solve the scalability benchmarks.**

Here we include basic performance results for a naïve oracle for Programming by Navigation to supplement the results from in Section 7. (This appendix is best read after Section 7.) We speculated in Section 2 that a fully-constructive oracle would do needless work, and here we show that that is indeed the case. We based our fully-constructive oracle on the Pruned Enumeration enumerator. As expected, the performance of this oracle is bad. Because it performs worse than both Pruned Enumeration and Honeybee, we do not explore it further. These results offer further evidence for the need for a truly inhabitation-based oracle, rather than a constructive one.

## B Proofs for Main Paper

Proof of Theorem 3.2. The rules for the top-down step relation are syntax-directed and deterministic (**Determinism**), so all that remains to show are the three properties for the $\prec$ relation. To do so, define $\phi(e)$ to be the number of function symbols in $e$. Note that if $e \prec e'$, then $\phi(e) < \phi(e')$.

(1) **No Loops**: $\prec$ is transitive by Step/Seq and irreflexive because if $e \prec e'$, then $\phi(e) < \phi(e')$, so $e \neq e'$.
(2) **Reachability** For all expressions $f(e_1, \ldots, e_N)$, we have $?_0 \xrightarrow{?_0 \rightsquigarrow f(e_1,\ldots,e_N)} f(e_1, \ldots, e_N)$. By assumption, holes are not valid expressions.
(3) **Finite Between**: If $e_0 \prec e_1 \prec \cdots$ is bounded by $e^*$, then $\phi(e_0) < \phi(e_1) < \cdots < \phi(e^*)$. By infinite descent, this chain must be finite. □

Proof of Theorem 3.7. If there are no valid expressions, then $C(e_{\text{start}}) = \varnothing$, and the only step set that covers $\varnothing$ is $\varnothing$. □

Proof of Theorem 3.8. By **Reachability**, $C(e_{\text{start}}) = \{e \mid e \text{ valid}\}$, which is nonempty by assumption. Therefore, if it is not the case that $e_{\text{start}}$ valid, then $\mathbb{S}(e_{\text{start}})$ must be nonempty by the definition of covering, which handles the case in which $N = 0$. If $N > 0$, suppose $\mathbb{S}(e_N) = \varnothing$. Then $C(e_N) \setminus \{e_N\} = \varnothing$ by **Strong Completeness**. However, $C(e_N) \neq \varnothing$ by **Strong Soundness** from the previous round of interaction. Therefore, we must have $e_N \in C(e_N)$, so $e_N$ valid. □

Proof of Theorem 3.9. Suppose $e_0 \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_k} e_k$ is an $\mathbb{S}$-interaction ($k \geq 0$) with $e_k \prec e$. Let $\Sigma = \mathbb{S}(e_k)$. Then $e \in C(e_k) \setminus \{e_k\} \subseteq \bigcup_{\sigma \in \Sigma} C(\sigma e_k)$ by **Strong Completeness**, so there exists $\sigma_{k+1} \in \Sigma$ such that $e \in C(\sigma_{k+1} e_k)$. Thus, letting $e_{k+1} = \sigma_{k+1} e_k$, we have that $e_0 \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_{k+1}} e_{k+1}$ is an $\mathbb{S}$-interaction with $e_{k+1} \preceq e$. If equality is achieved, we are done. Otherwise, we can repeat the process and extend the interaction further. However, **Finite Between** implies there must exist some point at which equality is achieved. □

Proof of Theorem 4.2. **Validity** and **Strong Soundness** follow directly from the definition of an inhabitation oracle. For **Strong Completeness**, suppose $e' \in C(e) \setminus \{e\}$. Then there exists $\sigma$ such that $e \xrightarrow{\sigma} e'$. Let $?_h \rightsquigarrow f(e_1, \ldots, e_N)$ be the left-most step of $\sigma$. As $e'$ valid, each $e_i$ must be a function application by our requirement that valid expressions do not contain holes. Further, we must have $?_h \triangleleft e$ and $(h, f) \in O(e)$, so $\sigma \in \mathbb{S}(e)$ where $\sigma = ?_h \rightsquigarrow f(?_{h_1}, \ldots, ?_{h_{\text{arity}(f)}})$. Then $\sigma e \xrightarrow{?_h \rightsquigarrow e_1 \,;\, \cdots \,;\, ?_h \rightsquigarrow e_{\text{arity}(f)}} e'$, so $e' \in C(\sigma e)$, and thus **Strong Completeness** holds. □

Proof of Lemma 5.4.

- *(1) $\Longrightarrow$ (2).* Proceed by induction on the derivation of $\text{vals}(\Gamma) \cup \text{vals}(\Delta), \Delta, \mathcal{H}[\![\Gamma]\!] \vdash_{\text{DL}} \tau(\overline{v})$. By assumption, this derivation ends in Rule$_f$ with premises $\tau_1(\overline{v_1}), \ldots, \tau_N(\overline{v_N})$ and $\varphi[\overline{v_1}, \ldots, \overline{v_N}; \overline{v}]$. The inductive hypothesis implies there exist $e_1, \ldots, e_N$ with $\Gamma, \Delta \vdash e_i : \tau(\overline{v_i})$ for each $i$, so all preconditions of $\Gamma, \Delta \vdash f^{\overline{v}}(e_1, \ldots, e_N) : \tau(\overline{v})$ are met.
- *(2) $\Longrightarrow$ (1).* As proving "$(\exists x.\, P(x)) \implies Q$" is equivalent to proving "$\forall x (P(x) \implies Q)$", it suffices to prove the equivalent proposition that if $\Gamma, \Delta \vdash f^{\overline{v}}(e_1, \ldots, e_N) : \tau(\overline{v})$, then $\text{vals}(\Gamma) \cup \text{vals}(\Delta), \Delta, \mathcal{H}[\![\Gamma]\!] \vdash_{\text{DL}} \tau(\overline{v})$ with a derivation tree ending in Rule$_f$. To do so, proceed by induction on the derivation of $\Gamma, \Delta \vdash f^{\overline{v}}(e_1, \ldots, e_N) : \tau(\overline{v})$. There is only one case, Well-Typed/Fun. By the inductive hypothesis, $\text{vals}(\Gamma) \cup \text{vals}(\Delta), \Delta, \mathcal{H}[\![\Gamma]\!] \vdash_{\text{DL}} \tau_i(\overline{v_i})$ for each $i$. Therefore, with the remaining premises, we can construct a derivation tree for $\text{vals}(\Gamma) \cup \text{vals}(\Delta), \Delta, \mathcal{H}[\![\Gamma]\!] \vdash_{\text{DL}} \tau(\overline{v})$ ending in Rule$_f$. □

Proof of Lemma 5.6. Let $(?_{h_\alpha})_{\alpha=1}^{M}$ be the holes of $e$, each of type $\tau_\alpha$.

- *(1)* $\implies$ *(2).* Each premise $\tau_\alpha(\overline{v_\alpha})$ must be derivable with all validity conditions in $e$ met and with a derivation tree ending in, say, $\textsc{Rule}_{g_\alpha}$. By Lemma 5.4, there exist expressions $\overline{e_\alpha}$ such that $\Gamma, \Delta \vdash {g_\alpha}^{\overline{v_\alpha}}(\overline{e_\alpha}) : \tau_\alpha(\overline{v_\alpha})$. Let $\sigma_\alpha = ?_{h_\alpha} \rightsquigarrow {g_\alpha}^{\overline{v_\alpha}}(\overline{e_\alpha})$. Then $\Gamma, \Delta \vdash (\sigma_1 ; \cdots ; \sigma_M)e : \tau(\overline{v})$, so $C([?_h \mapsto {g_h}^{\overline{v_h}}(\overline{e_h})]e) \neq \varnothing$.
- *(2)* $\implies$ *(1).* As $C([?_h \mapsto g(e_1, \ldots, e_N)]e) \neq \varnothing$, each function application in $e$ must have the correct simple type, so $Q_{\Gamma,\tau}[\![e]\!]$ is defined. Additionally, there exists $k$ such that $\textsc{Query}_{h,k} \in Q_{\Gamma,\tau}[\![e]\!]$ because $?_h \triangleleft e$. Moreover, there must exist $\overline{v_\alpha}$ for each $\alpha$ such that $\tau_\alpha(\overline{v_\alpha})$ is inhabited and all validity conditions in $e$ are met. By Lemma 5.4, each $\tau_\alpha(\overline{v_\alpha})$ must be derivable, and, in particular, $\tau_h(\overline{v_h})$ must be derivable with a proof tree ending in $\textsc{Rule}_g$. Thus, the premises of $\textsc{Query}_{h,k}$ can be met with the $k$th subtree ending in $\textsc{Rule}_g$. □

Proof of Lemma 5.8. Suppose $\overline{v}$ satisfies $R_2$ with a derivation tree whose $k$th subtree ends in $R_1$ on value $\overline{v'}$. Then all premises of $R_1$ and $R_2$ hold with $\overline{y_k} = \overline{x} = \overline{v'}$. These are the premises needed for $R_1/k/R_2$, so $\overline{v}$ satisfies $R_1/k/R_2$.

Conversely, suppose $\overline{v}$ satisfies $R_1/k/R_2$ with $\overline{y_k} = \overline{x} = \overline{v'}$. Then all premises of $R_1$ hold with $\overline{x} = \overline{v'}$, and thus $P(\overline{v'})$ holds. Therefore, all premises of $R_2$ hold with $\overline{y} = \overline{v}$ and $\overline{y_k} = \overline{v'}$, where the $k$th premise can be satisfied by the derivation of $P(\overline{v'})$. □

Proof of Theorem 5.9. Lines 3 and 4 loop over each possible $h$ and $f$ for $e$, and Line 5 filters out any $f$ whose return type does not match the required type. By Lemma 5.8, the query to the rule $\textsc{Rule}_f/k/\textsc{Query}_{h,k}$ will return all solutions to $\textsc{Query}_{h,k}$ where there exists a derivation such that the $k$th subtree ends in $\textsc{Rule}_f$. By Lemma 5.6, this will result in the values $\overline{v}$ such that there exist $e_1, \ldots, e_N$ with $C([?_h \mapsto f^{\overline{v}}(e_1, \ldots, e_N)]e) \neq \varnothing$, which is true (for the top-down rules) if and only if $C([?_h \mapsto f(?_{h_1}, \ldots, ?_{h_{\text{arity}(f)}})]e) \neq \varnothing$ with $?_{h_1}, \ldots, ?_{h_{\text{arity}(f)}}$ fresh in $e$, as desired. □