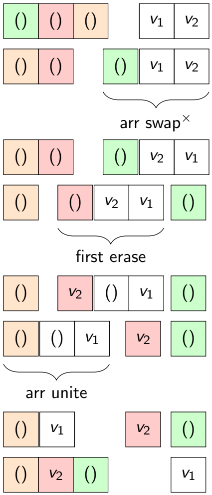
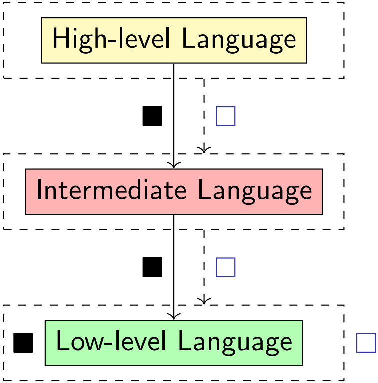
Reversibly erasing the first of a pair



Implementation Overview



- – Completed Implementation
- - - TODO: Implementation
- ■ Completed Proof
- □ TODO: Proof

Language definitions

Evaluation in high-level language LET:

$$\mathsf{eval_e} : \forall\{n : \mathbb{N}\}\{\Gamma : \mathsf{Vec}\ \mathbb{b}\ n\}\{b : \mathbb{b}\}$$
$$\to \Gamma\ \mathsf{env} \to \Gamma \vdash\!\mathsf{exp}\!:\ b \to \mathsf{val}\ b$$

Evaluation in low-level language Π:

$$\_[\_]^{\mathsf{f}} : \forall\ \{b\ b'\} \to \mathsf{comb_0}\ (b \leftrightarrow b') \to \mathsf{val}\ b \to \mathsf{val}\ b'$$
$$\_[\_]^{\mathsf{b}} : \forall\ \{b\ b'\} \to \mathsf{comb_0}\ (b \leftrightarrow b') \to \mathsf{val}\ b' \to \mathsf{val}\ b$$

Evaluation in intermediate language ML_Π:

$$\_[\_]^{\mathsf{a}} : \forall\ \{b\ b'\} \to \mathsf{comb}\ (b \rightsquigarrow b') \to \mathsf{val}\ b \to \mathsf{val}\ b'$$

Translations and proofs

Proof of Π's reversibility:

$$\text{lemma-3} : \forall \{b\ b'\}(c : \text{comb}_0\ (b \leftrightarrow b'))(v : \text{val}\ b) \rightarrow (c\ \dagger)\ [\ c\ [\ v\ ]^{f}\ ]^{f} \equiv v$$

T_1 that translates programs from LET to ML_Π is correct:

$$T_1\text{-proof} : \forall\{n : \mathbb{N}\}\{\Gamma : \text{Vec}\ \mathbb{b}\ n\}\{b : \mathbb{b}\} \rightarrow (\rho : \Gamma\ \text{env}) \rightarrow (e : \Gamma \vdash \text{exp}:\ b) \rightarrow (\text{eval}_e\ \rho\ e) \equiv ((T_1\ e)\ [\ ((\rho)_e{}^{x})\ ]^{a})$$

T_2 that translates programs from ML_Π to Π is correct:

$$T_2\text{-proof} : \forall\{b_1\ b_2\}(c : \text{comb}\ (b_1 \rightsquigarrow b_2))(v : \text{val}\ b_1) \rightarrow \Sigma\ (\text{val}\ (\text{garbage}(c)))\ (\lambda\ g' \rightarrow ((T_2\ c)\ [\ ([\ \varphi(\text{heap}(c))\ ,\ v\ ])\ ]^{f}) \equiv ([\ g'\ ,\ (c\ [\ v\ ]^{a})\ ]))$$