Garbage Collection

```
decl n=10;      σ_1 = []
decl x=0;
decl r=0;
while (x<n){    σ_2 = [n↦10, x↦0, r↦0]
  x=x+1;
  r=r+1
};

decl y=0;       σ_3 = [n↦10, x↦10, r↦10, y↦0]
while (y<n){
  y=y+1;
  r=r+2.
}
```

x never used again
can get rid of it + free up memory

$s ::= \cdots \mid \text{decl } x = e$

$$\frac{x \notin \Gamma \quad \Gamma \vdash e : \tau}{\Gamma \vdash \text{decl } x = e \text{ ok}}$$

$[], (\text{decl } n=10; n=n+1) \mapsto^2 [n \mapsto 10]; n=n+1$

$\emptyset \vdash []$ $\qquad \emptyset \vdash [n \mapsto 10]$ $\qquad \emptyset \vdash n=n+1 \text{ ok?}$

No!

Preservation: If $\Gamma \vdash s$ ok and $\Gamma \vdash \sigma$ and $\sigma; s \mapsto \sigma'; s'$ then there exists Γ' s.t. $\Gamma' \vdash \sigma'$ and $\Gamma' \vdash s'$ ok.

$$\frac{\sigma; s \mapsto \sigma'; s'}{\sigma; s \Rightarrow \sigma'; s'}\ (\text{s-step}) \qquad \frac{\sigma' = [x \mapsto v \mid \sigma(x) = v \text{ and } x \in FV(s)]}{\sigma; s \Rightarrow \sigma'; s}\ (\text{s-GC})$$

$[n \mapsto 10; x \mapsto 10, r \mapsto 10]; \text{decl } y=0; \ldots \Rightarrow [n \mapsto 10, r \mapsto 10]; \text{decl } y=0; \ldots \quad (x \notin FV(s))$

$e ::= \dots \mid \lambda x. e \mid e\ e$

decl x = 10;
decl f = $\lambda y.\ x + y$;
decl r = f 5

$$
\begin{aligned}
[\,]; s &\Rightarrow^* [x \mapsto 10;\ f \mapsto \lambda y. x+y];\ \text{decl } r = f\ 5 \\
&\Rightarrow [f \mapsto \lambda y. x+y];\ \text{decl } r = f\ 5 \\
&\Rightarrow \text{''} \quad ;\ \text{decl } r = (\lambda y. x+y)\ 5 \\
&\Rightarrow \text{''} \quad ;\ \text{decl } r = x + 5 \\
&\not\Rightarrow
\end{aligned}
$$

$$\text{Reachable}(V, \sigma) = V \cup \bigcup \{\text{Reachable}(FV(\sigma(x)), \sigma) \mid x \in V\}$$

$$
\begin{aligned}
\text{Reachable}(FV(s), \sigma) &= \text{Reachable}(\{f\}, \sigma) \\
&= \{f\} \cup \text{Reachable}(FV(\lambda y\ x+y), \sigma) \\
&= \{f\} \cup \text{Reachable}(\{x\}, \sigma) \\
&= \{f, x\} \cup \text{Reachable}(FV(10), \sigma) \\
&= \{f, x\}
\end{aligned}
$$

$$\frac{\sigma' = [x \mapsto \sigma(x) \mid x \in \text{Reachable}(FV(s), \sigma)]}{\sigma; s \Rightarrow \sigma'; s} \ (\text{S-GC})$$

Memory Safety - never remove a var we'll need.
Type safety $\Rightarrow$ Memory Safety

Progress ✓

Note: If $x \in \text{Reachable}(V, \sigma)$ then $FV(\sigma(x)) \subset \text{Reachable}(V, \sigma)$

Preservation: 1. If $\Gamma \vdash s$ ok and $\Gamma \vdash \sigma$ and $\sigma; s \mapsto \sigma'; s'$ then $\exists\, \Gamma'$ s.t. $\Gamma' \vdash s'$ ok and $\Gamma' \vdash \sigma'$.

2. If $\Gamma \vdash s$ ok and $\Gamma \vdash \sigma$ and $\sigma; s \Rightarrow \sigma'; s'$ then $\exists\, \Gamma'$ s.t. $\Gamma' \vdash s'$ ok and $\Gamma' \vdash \sigma'$.

1. $$\frac{e \text{ val}}{\sigma; \text{decl } x = e \mapsto \sigma[x \mapsto e]; \text{skip}}$$

Then $e = \text{decl } x = e_0$ and e_0 val. By inversion, $\Gamma \vdash e : \tau$ and $x \notin \Gamma$. Let $\Gamma' = x:\tau$. $\Gamma' \vdash \text{skip}$ ok and $\Gamma' \vdash \sigma[x \mapsto e]$

2. (S-Step) - Easy by (1)

(S-GC)

Lemma 1. If $\Gamma \vdash s$ ok $(\Gamma \vdash e:\tau)$ and $\Gamma'(x) = \Gamma(x)\ \forall x \in FV(s)$ (e) then $\Gamma' \vdash s$ ok $(\Gamma' \vdash e:\tau)$

Pf. By induction on $\Gamma \vdash s$ ok $(\Gamma \vdash e:\tau)$

Lemma 2. If $\Gamma \vdash \sigma$ and $\Gamma'(x) = \Gamma(x)\ \forall x \in Reachable(V, \sigma)$ then $\Gamma' \vdash [x \mapsto \sigma(x) \mid x \in Reachable(V, \sigma)]$

Pf. WTS. for all $x:\tau \in \Gamma'$, $\Gamma' \vdash \sigma'(x):\tau$.

Let $x:\tau \in \Gamma'$. Then $\tau = \Gamma(x)$ and $x \in Reachable(V, \sigma)$.

$\sigma'(x) = \sigma(x)$

Let $y \in FV(\sigma'(x))$. $y \in Reachable(V, \sigma)$, so $\Gamma'(y) = \Gamma(y)$.

By Lemma 1, $\Gamma' \vdash \sigma'(x):\tau$. $\square$

Let $\Gamma' = \{x:\Gamma(x) \mid x \in Reachable(FV(s), \sigma)\}$

$\Gamma'(x) = \Gamma(x)\ \forall x \in FV(s)$, so $\Gamma' \vdash s$ ok by Lemma 1

$\Gamma'(x) = \Gamma(x)\ \forall x \in Reachable(FV(s), \sigma)$, so $\Gamma' \vdash \sigma'$ by Lemma 2. $\square$