DAY 9

EFFECT SYSTEM (DETECTION OF SIDE EFFECTS)

$$effect : \text{KNormal.t} \rightarrow \text{bool}$$
$$effect(e) = \text{true/false}$$ effect free = false
$$effect(1) = \text{false}$$
$$effect(\text{println}(1)) = \text{true}$$
$$effect(\text{array.}(0) \text{ <- } 0) = \text{true} \qquad \text{\# Array assignment}$$

IMPLEMENTATION OF REDUNDANCY ELIMINATION

For `let`:

$$\varepsilon(\text{let } x = e_1 \text{ in } e_2) = \begin{cases} \varepsilon(e_2) & \neg effect(\varepsilon(e_1)) \text{ (e1 is effect free)} \; \sout{\wedge \neg effect(\varepsilon(e_2))} \text{ (dont need this condition)} \; \sout{\wedge} \\ & x \notin free\_variables(\varepsilon(e_2)) \\ \text{let } x = \varepsilon(e_1) \text{ in } \varepsilon(e_2) & \text{Otherwise} \end{cases}$$

```
| Let((x, t), e1, e2) ->
    let e1' = f e1 in
    let e2' = f e2 in
    if effect e1' || S.mem x (fv e2') then Let((x, t), e1', e2') else e2'
```

划红线的case等于下面这个case

$$\mathcal{E}(\texttt{let } x = e_1 \texttt{ in } e_2) \quad = \quad \mathcal{E}(e_2) \qquad effect(\mathcal{E}(e_1)) = false \text{ かつ } x \notin FV(\mathcal{E}(e_2)) \text{ の場合}$$

fv=free variable

【深入淺出教你寫編譯器（Compiler）】一、簡介
【深入淺出教你寫編譯器（Compiler）】二、掃描器（Scanner）－詞法分析（Lexical analysis）（上）
【深入淺出教你寫編譯器（Compiler）】二、掃描器（Scanner）－詞法分析（Lexical analysis）（下）
【深入淺出教你寫編譯器（Compiler）】三、語法分析器（Parser）－語法分析（Syntactic analysis）（上）
【深入淺出教你寫編譯器（Compiler）】三、語法分析器（Parser）－語法分析（Syntactic analysis）（中）
【深入淺出教你寫編譯器（Compiler）】三、語法分析器（Parser）－語法分析（Syntactic analysis）（下）

【深入淺出教你寫編譯器（Compiler）】四、語意分析（Semantic analysis）
【深入淺出教你寫編譯器（Compiler）】五、虛擬機（Virtual Machine）
【深入淺出教你寫編譯器（Compiler）】六、編譯器（Compiler）－生成代碼（Code Generation）（上）
【深入淺出教你寫編譯器（Compiler）】六、編譯器（Compiler）－生成代碼（Code Generation）（下）
【深入淺出教你寫編譯器（Compiler）】七、優化器（Optimizer）－還可以更好
https://github.com/python/cpython/blob/master/Tools/parser/unparse.py