Rewrite rules (skeletons & patterns)

Skel ::= Seq(...) | Pipeline(Skel, Skel) | Farm(Skel) | Comp(Skel, Skel)
| Map(Skel) | Reduce(Skel) | ...

pgm = Pipeline(Seq(f), Seq(g))

$x \to Seq(f) \to f(x)$

$g(f(x))$

Comp(Seq(f), Seq(g))

$g(f(x))$

Rewr. rule (1) $Pipeline(S_1, S_2) \equiv Comp(S_1, S_2)$

pipeline elimination → less parallelism, larger grain

pipeline introduction ← increase parallelism, smaller grain

Rewr. rule (2) $Farm(S_1) \equiv S_1$

farm elimination → less parallelism

increase parallelism ←

$p(f, g)$

$f: \alpha \to \beta \quad g: \beta \to \gamma$

$p(f,g): \alpha \to \gamma$

$\alpha \to \beta \quad \alpha \to \beta \quad \alpha \to \beta$

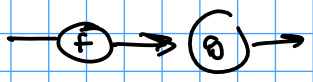
$p([f_1, f_2, f_3])$

$\alpha \to \alpha$

Rewr. rule (3) associativity rule

$Pipe(S_1, Pipe(S_2, S_3)) \equiv Pipe(Pipe(S_1, S_2), S_3)$

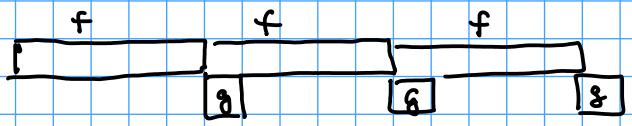
$Comp(S_1, Comp(S_2, S_3)) = Comp(Comp(S_1, S_2), S_3)$



PIPE (SEQ (f), SEQ(g))

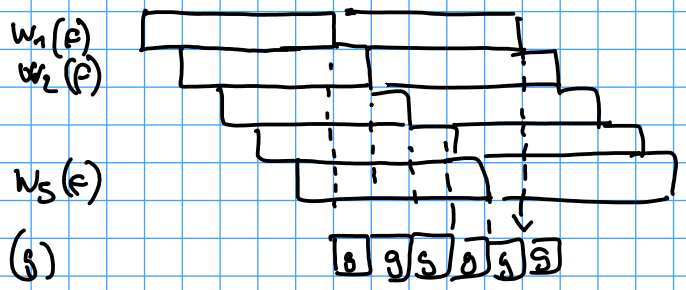
x_m ---- x_{n+2} x_n .... x_1

$t_f = 5t_g$



RR_2

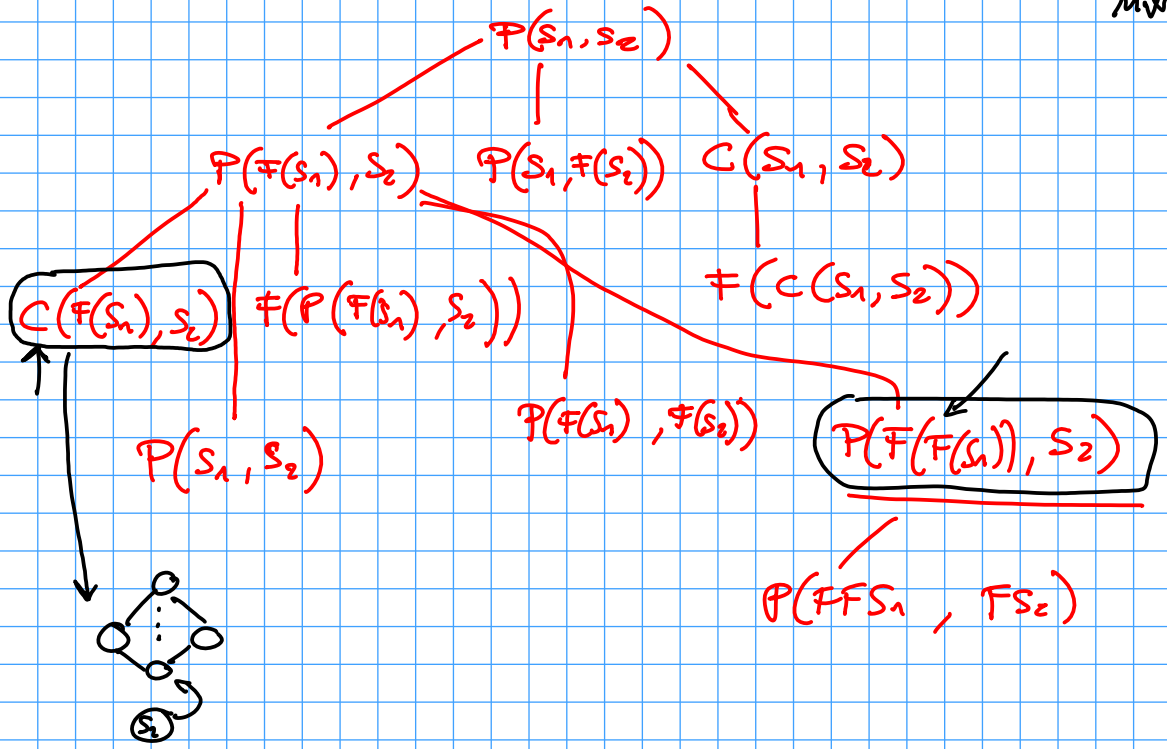
PIPE (FARM(SEQ(f)), SEQ(g))

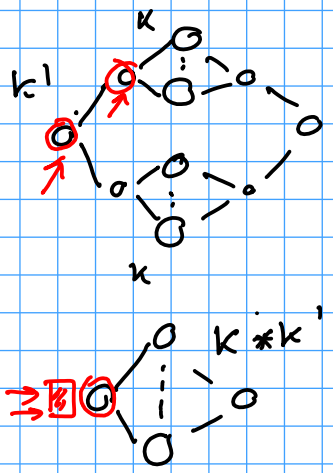


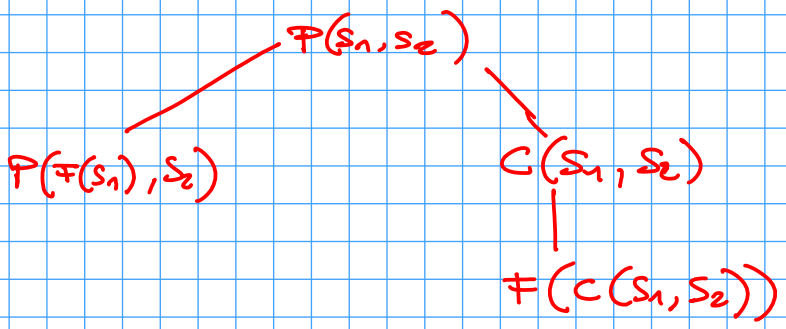
P(seq(f), seq(g)) → pipe elim → C(seq(f), seq(g)) → form into → F(C(seq(f), seq(g)))

$6t_g$

mvr = 6 → 1 task computed each t_g







Visitors

n of PE needed

Ts of the pattern exp

$nofPE(Seq(f)) = 1$

$nofPE(farm(S_1) \text{ with } nw \text{ workers}) = nofPE(S_1) \cdot nw + 2$

$nofPE(PIPE(S_1, S_2)) = nofPE(S_1) + nofPE(S_2)$

$nofPE(comp(S_1, S_2)) = max(nofPE(S_1), nofPE(S_2))$

need to be computed (e.g. by profiling)

$ts(Seq(f)) = t_f$

$ts(pipe(S_1, S_2)) = max\{ts(S_1), ts(S_2)\}$

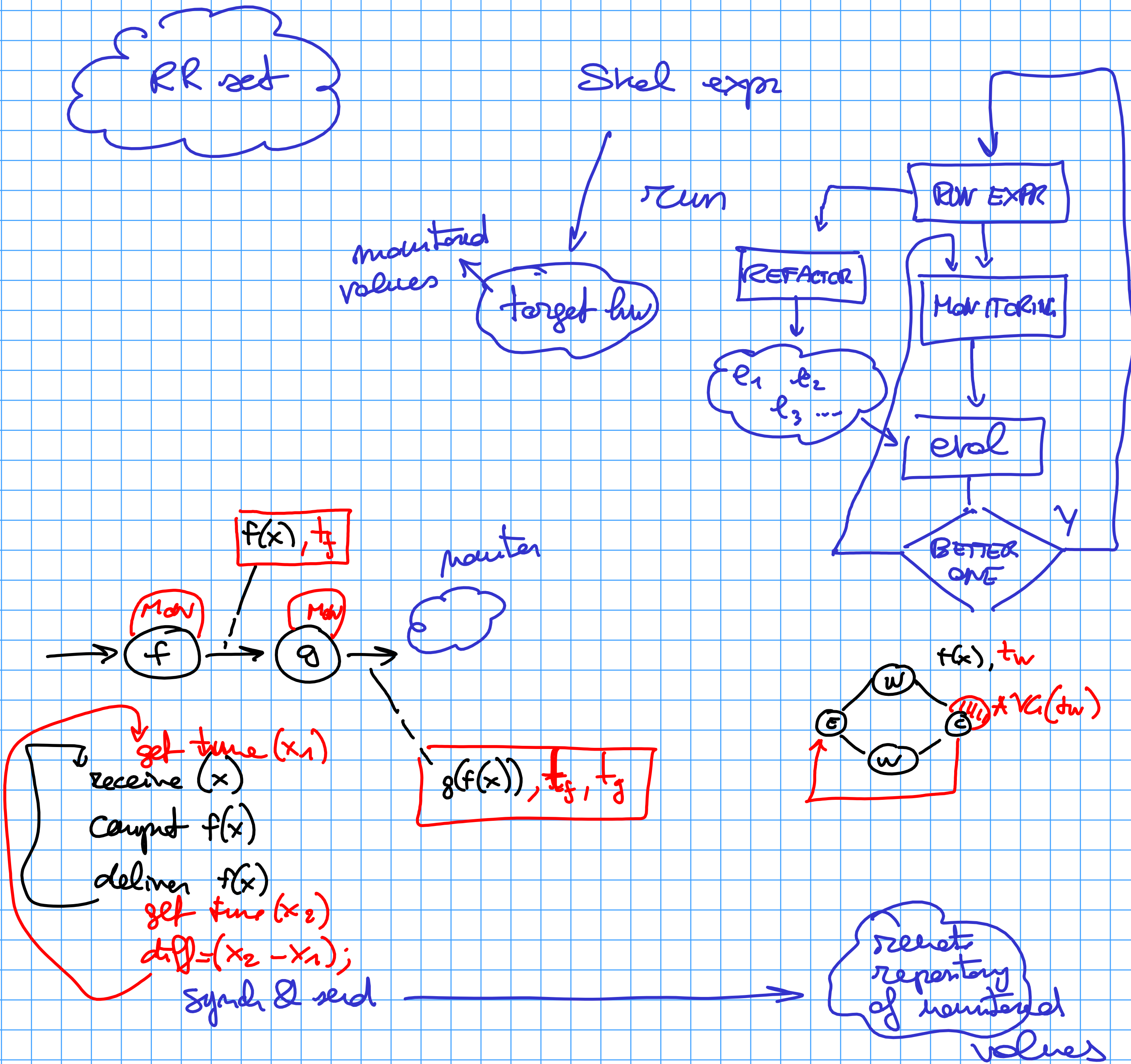
$ts(farm(S_1) \text{ with } nw \text{ workers}) = ts(S_1) / nw$

$ts(comp(S_1, S_2)) = ts(S_1) + ts(S_2)$

Skel ::= Seq(f) | ... | Skel with annotation list

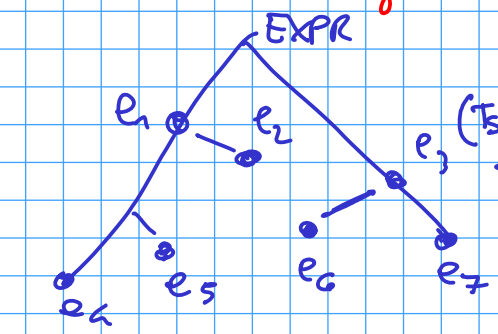
farm(S_1) with [nw(10)]

seq(f) with [ts(10s)]


RR set
Skel expr
run
monitored values
target hw
RW EXPR
REFACTOR
MONITORING
e1 e2 e3 ...
eval
BETTER ONE
Y
f(x), tf
MON
MON
f
g
monitor
get time (x1)
receive (x)
compute f(x)
deliver f(x)
get time (x2)
diff=(x2-x1);
synch & send
g(f(x)), tf, tg
f(x), tw
W
E
C
W
remote repository of monitored values

path expression EXPR

Rew Rules → REFACTORER → generate a population of equivalent preferred expressions



VISITORS ($T_5 = \cdots$, initFE = ...)

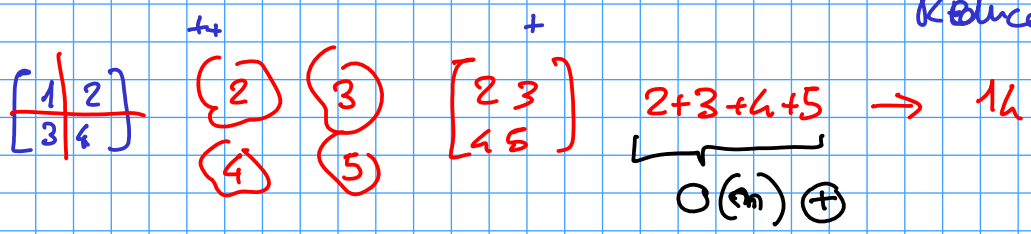
to label the expressions with the ran functional feature "values"

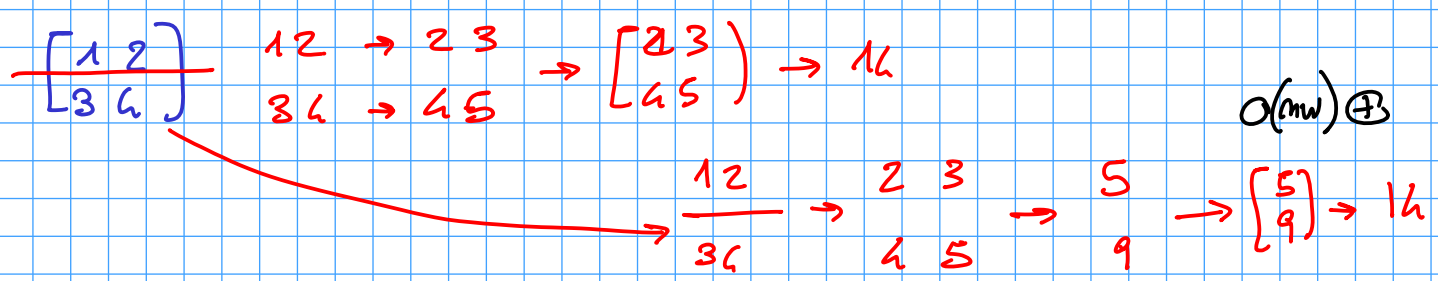
decision process to "pick up" the best solution

Pipe (Map (f), Map (g)) ≡ Map (Pipe (f, g))

Map fusion rule

Pipe (Map (f) , Reduce (⊕)) ≡ Pipe(Comp(Map(f), Reduce(⊕)), Reduce(⊕));





WC map reduce (google)

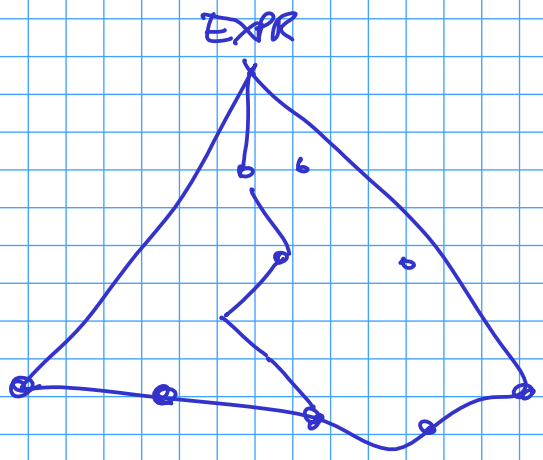
SkeTo

C++

```
array  x, y
   :
```

```
x1 = map (f, x);
y1 = map (g, x1);
```

```
x1 = map (f, x);
y1 = map (g, y);
z = mapzip (h, x1, y1);
```



Mandane vs. the target function

may lead to local minima

3t 6t

$p(S_1, S_2)$ $T_S = 6$

$p(S_1, F(S))$ $T_S = 3$

$F(p(S_1, F(S)))$ $T_S = te$

$c(S_1, S_2)$ $T_S = 9$

$F(c(S_1, S_2))$ $T_S = te$