

# SKELETON DESIGN

par. exploitation pattern  
parametric function ( $f$ )  
non function ( $m_w$ )

reusable / common  
programming abstraction

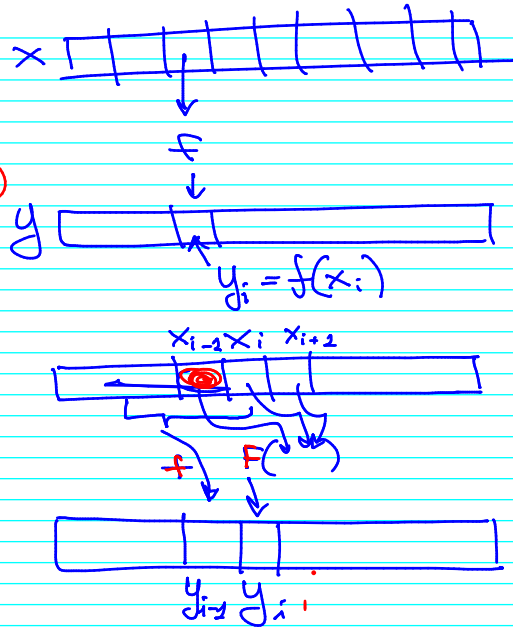
efficient & portable

## COLE'S MANIFESTO

BRINGING SKELETONS OUT OF  
THE CLOSET:

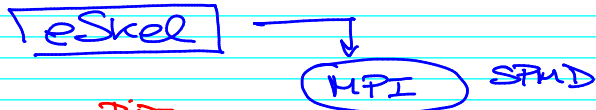
① PROPAGATE THE CONCEPT  
WITHA MINIMAL DISRUPTION

```
#pragma omp parallel for  
for (i=0; i<N; i++)  
{ ... body  $y[i] = f(x[i])$  }
```

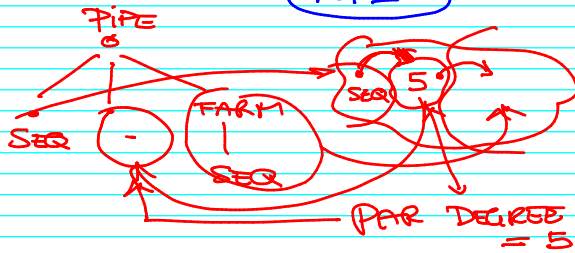


② INTEGRATE AD HOC PARALLELISM

"ESCAPE" SKELETON



④ STOP THE PAYBACK



③ ACCOMMODATE DIVERSITY

farm (f)      farms → workers  
farm (f, school)

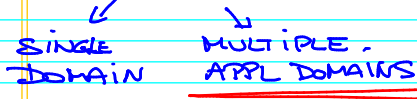
# LOOKING FOR NEW SKELETONS

## ANALYSIS

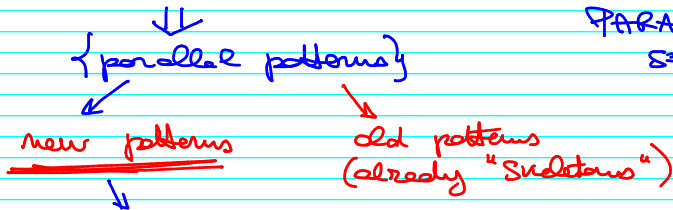
## SYNTHESIS

next slide

① Set of APPLS



② "DISTILL STEP"



③ Select candidates

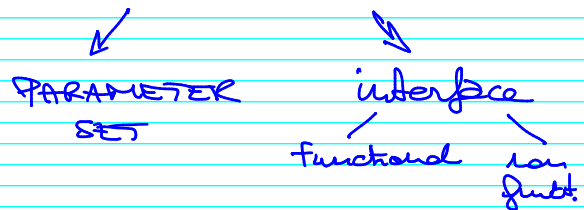
those with properties (see before)

④ Analysis

eliminate skeletons that are components of existing models

comp( $S_1, S_2$ ) do  $S_1$  then  
do  $S_2$

⑤ SKELETON REFINEMENT



# SYNTHESIS

START with APPL<sub>x</sub>

→ we decide  
a new pattern / srel  
is needed: SK<sub>x</sub>

① ANALYSIS (as before)

② Look for GENERALITY  
of SK<sub>x</sub>

↙  
it useful  
only in  
my application

↘  
or also  
in other  
cases

③ REFINEMENT  
STEP

## SKELETONS vs TEMPLATES

{applications, ... }

? look for templates  
rather than skeletons