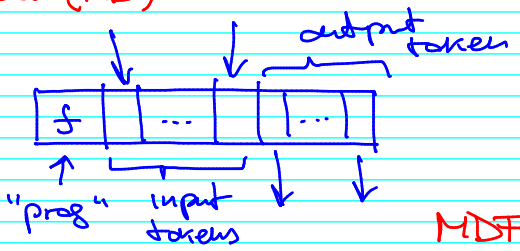
MACRO DATA FLOW (MDF)

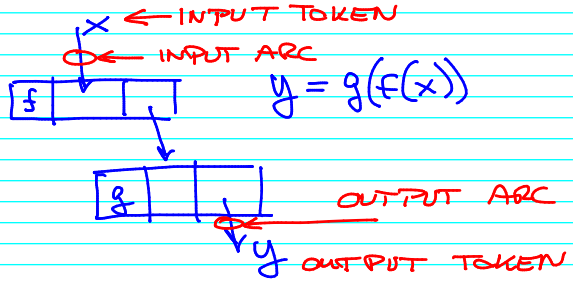
data deps

MDF instructions (MDF_i)

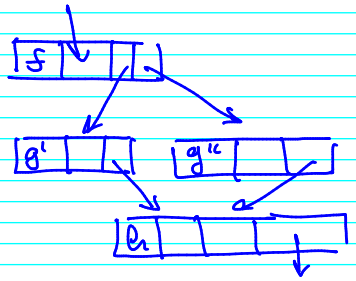


$MDF_g = (N, A)$

Es:



$y = g(f(x))$



FIREABLE INSTRUCTION:

MDF_i with all input tokens

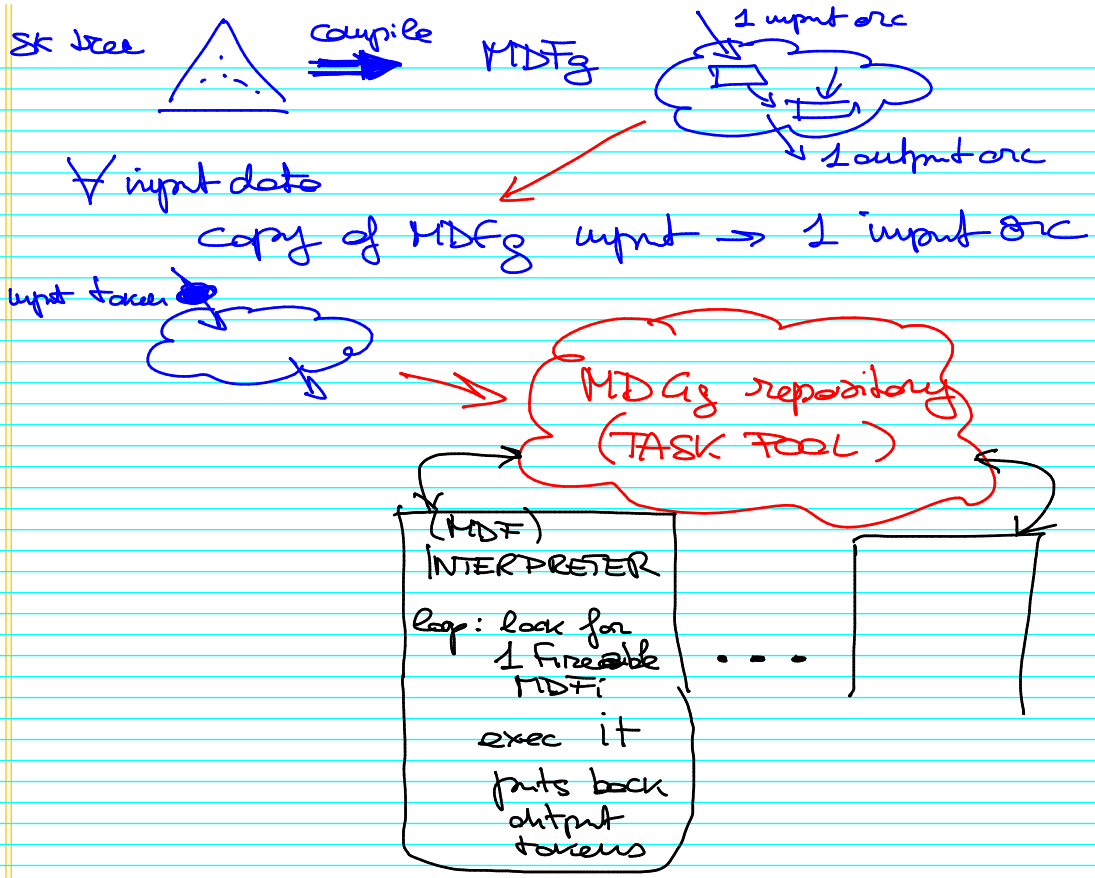
Requirement:

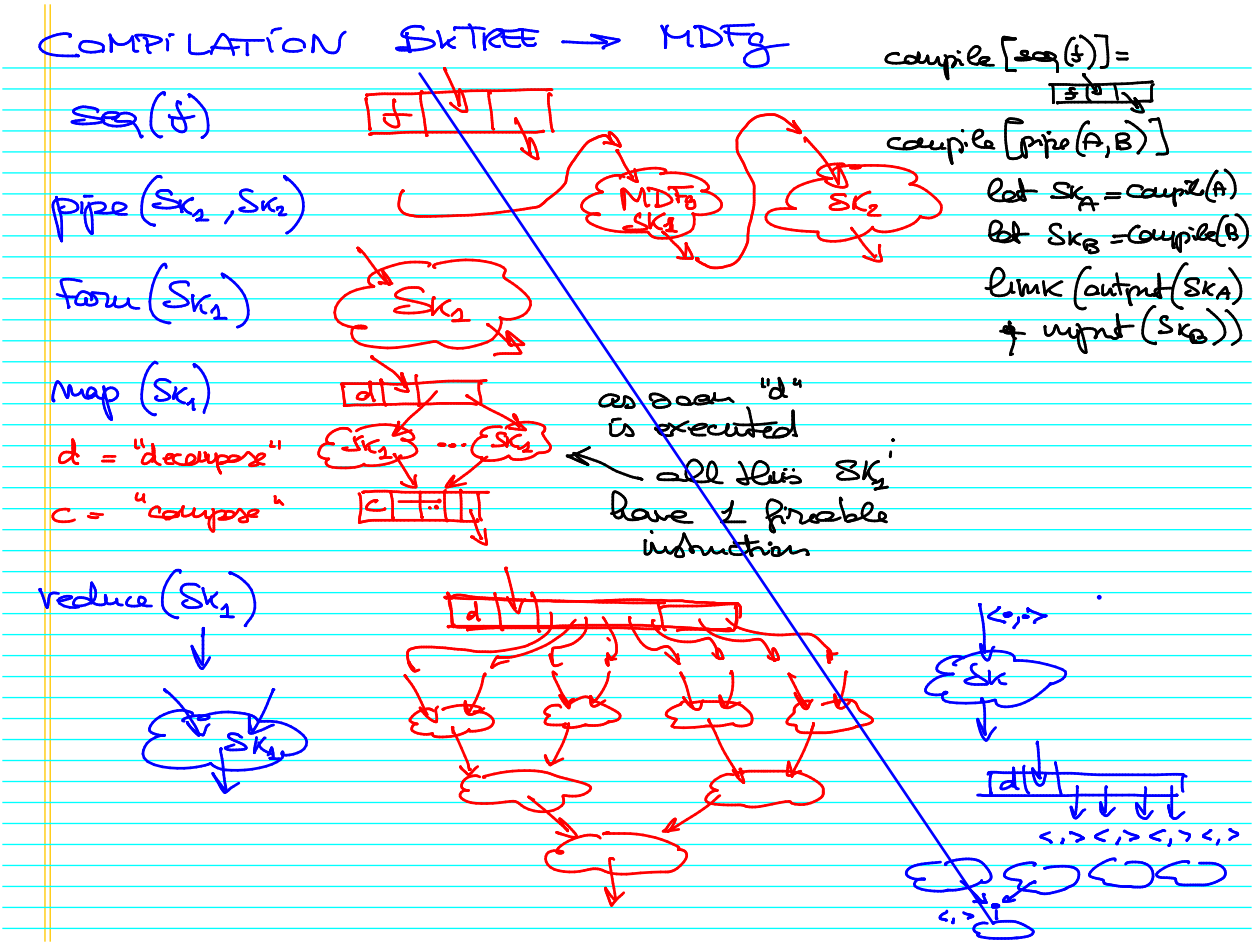
1 input arc

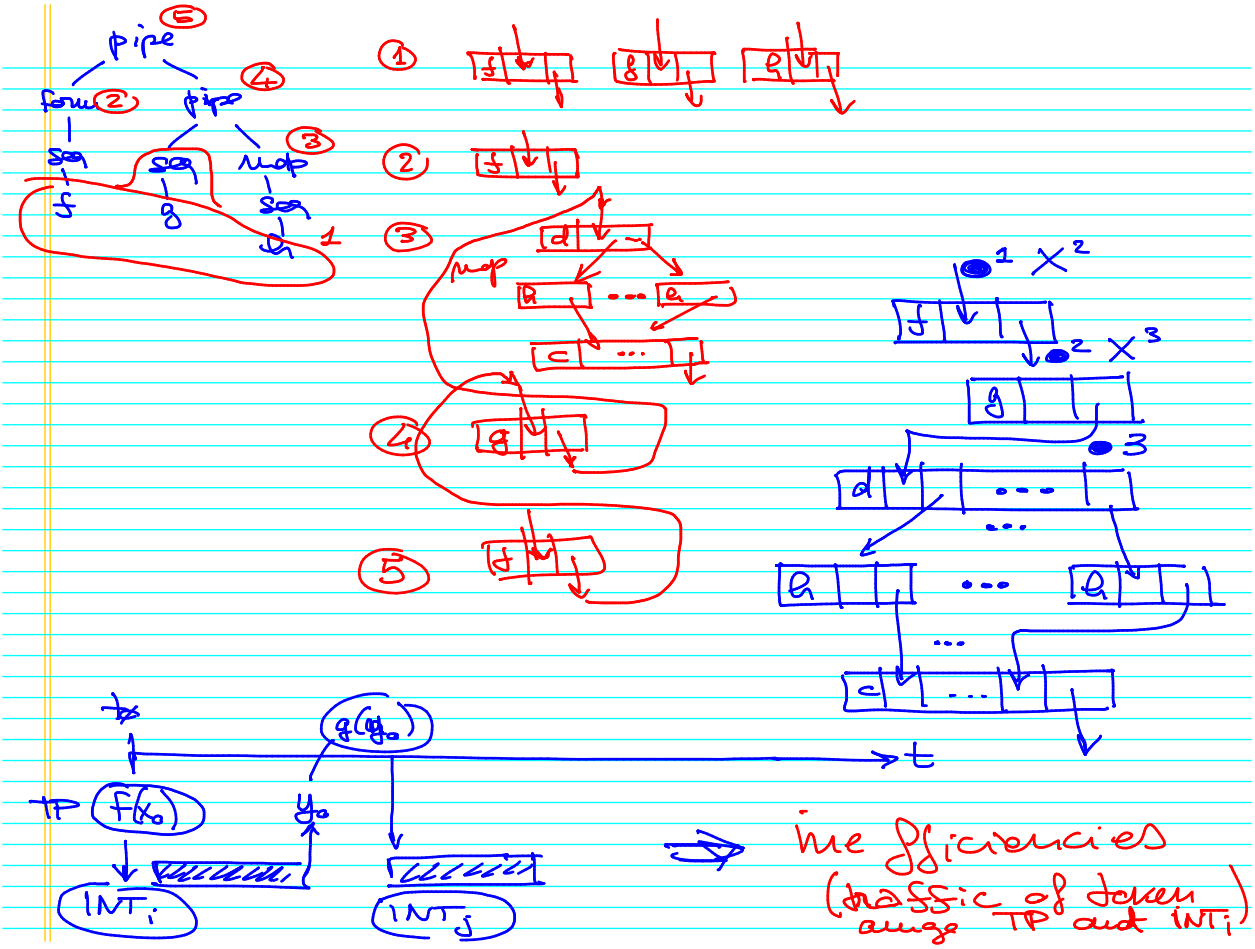
1 output arc

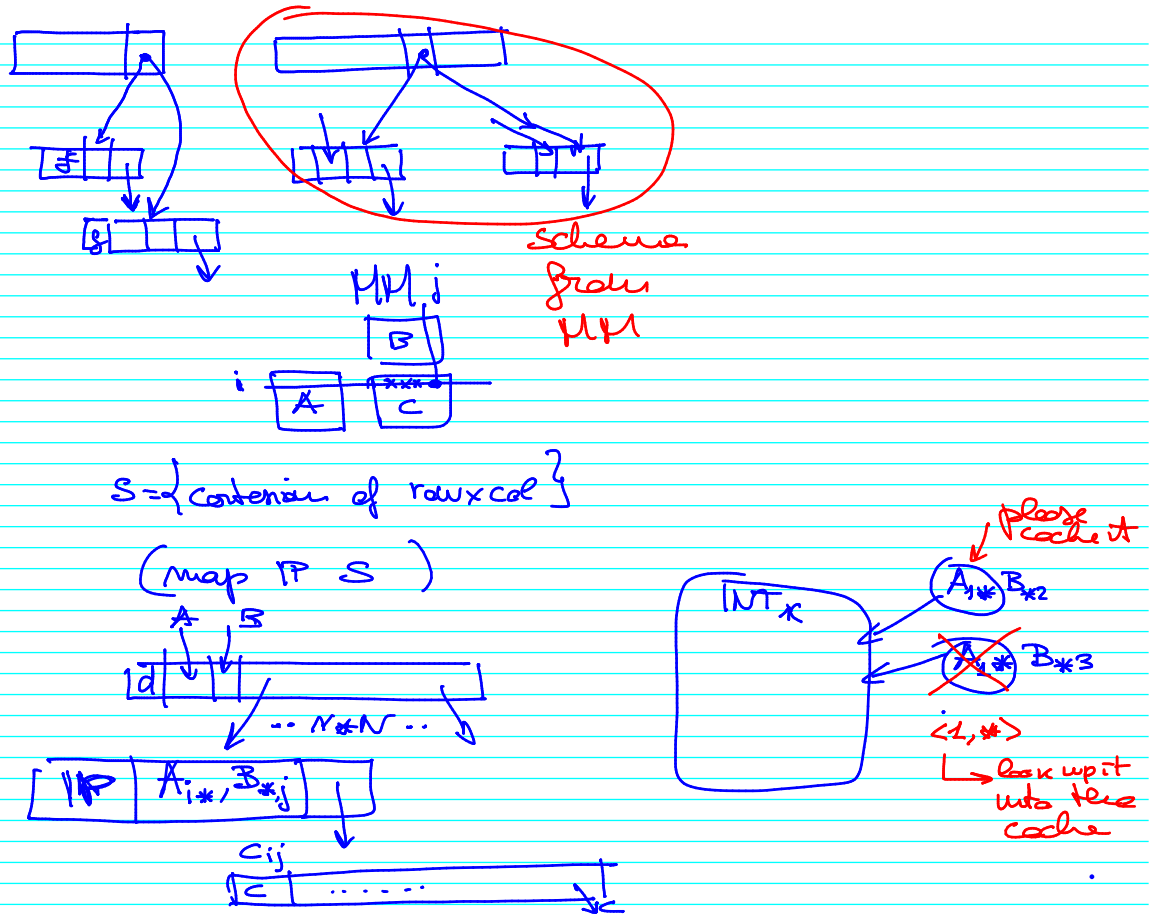
"MACRO"

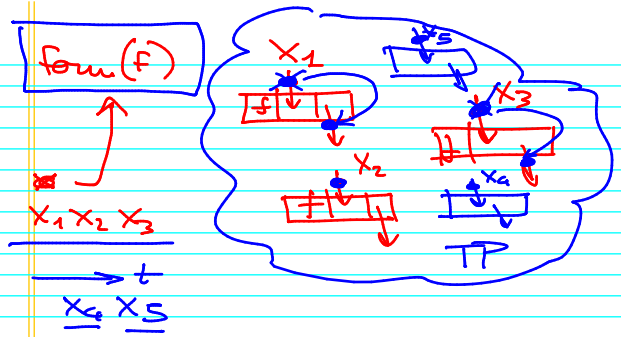
↳ f, g, h may be big "codes"

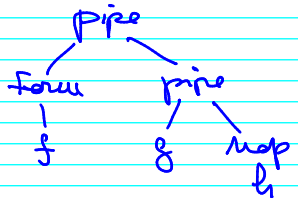

SK tree
compile
MDFg
1 input arc
1 output arc
∀ input data
copy of MDFg input → 1 input arc
input tokens
MDGg repository
(TASK POOL)
(MDF)
INTERPRETER
loop: look for
1 Fireable
MDFi
exec it
puts back
output
tokens


COMPILATION SKTREE → MDFg
seq(f)
pipe(SK1, SK2)
farm(SK1)
map(SK1)
d = "decompose"
c = "compose"
reduce(SK1)
compile[seq(f)] =
compile[pipe(A,B)]
let SKA = compile(A)
let SKB = compile(B)
link(output(SKA) → input(SKB))
MDFg SK1
SK2
SK1
d
SK1 ... SK1
c
as soon "d" is executed
all these SK1 have 1 fireable instruction
SK1
d
SK
<,>
<,> <,> <,> <,>
<,>


pipe
farm
pipe
seq
f
seq
g
map
seq
h
1
2
3
4
5
map
1
2
3
t
TP
$f(x_0)$
$g(y_0)$
y_0
INT_i
INT_j
inefficiencies
(traffic of tokens
TP and INT_i)


Scheme from MM
MM.j
G
i
A
C
S = { content of row x col }
(map fP S)
A
B
Id
.. N*N ..
fP
A_i*, B_*j
C_ij
C
INT x
A_1*
B_*2
A_1*
B_*3
<1,*>
look up it into the cache
Place cache it

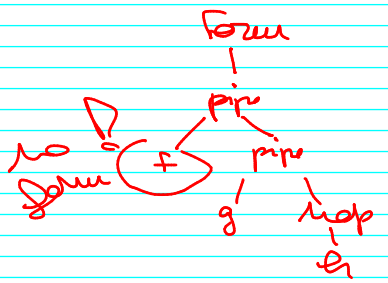


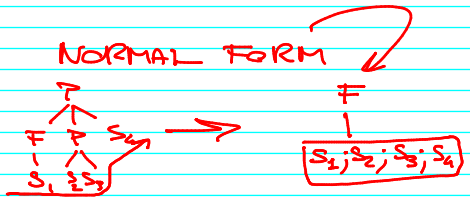


pipe f g = ...

farm f = ...

Streamer f s =





Performance Models

PRIMITIVE MEASURES

LATENCY — f : code

L_f = 10 secs

WALL CLOCK TIME

SERVICE TIME (wall clock time)

time between delivery of two results (only for stream parallel computation)

T_S

#tasks = m

COMPLETION TIME (m) = ... (par. degree)

m tasks L_{task}

BANDWIDTH = $= 1/T_S(m)$

1 processing element $\Rightarrow m = 1$?

$T_C(1) = m * L_{task}$

ideally $T_C(m) = \frac{T_C(1)}{m}$

DERIVED MEASURES

$$\text{Speedup}(n) = \frac{T_{seq}}{T(n)}$$

T_{seq} = the best seq time

$$\text{Scalability}(n) = \frac{T(1)}{T(n)}$$