Static Dataflow Graphs

Arvind
Computer Science & Artificial Intelligence Lab
Massachusetts Institute of Technology

Motivation: Dataflow Graphs

A common Base Language

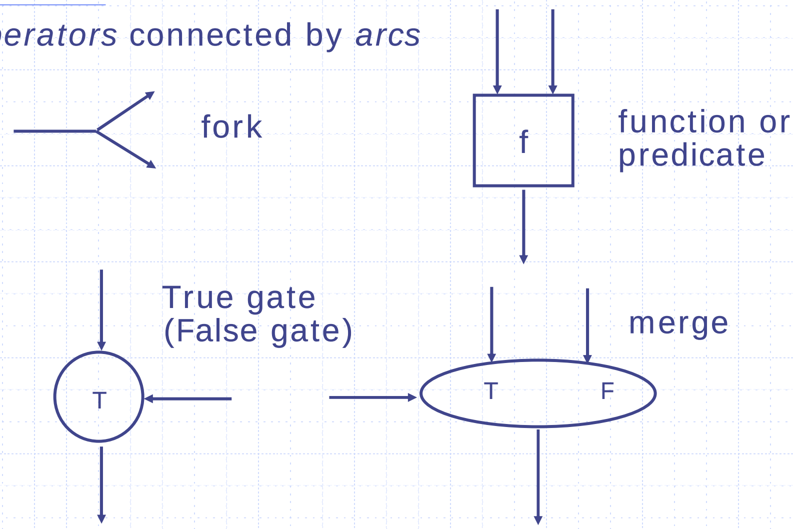
- to serve as target representation for high-level languages;

- to serve as machine language for a *highly parallel* machine.

Jack Dennis
Computation Structures Group, MIT
during 1967-75

Dennis' Program Graphs

Operators connected by *arcs*



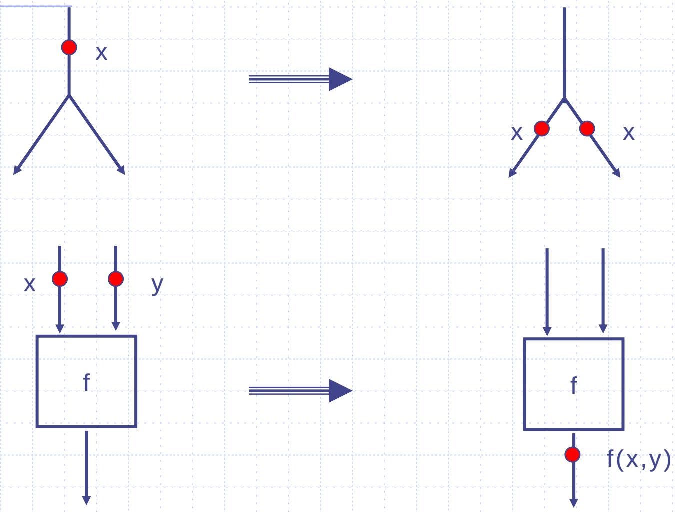
Dataflow

- Execution of an operation is *enabled* by *availability of the required operand* values. The completion of one operation makes the resulting values available to the elements of the program whose execution depends on them.

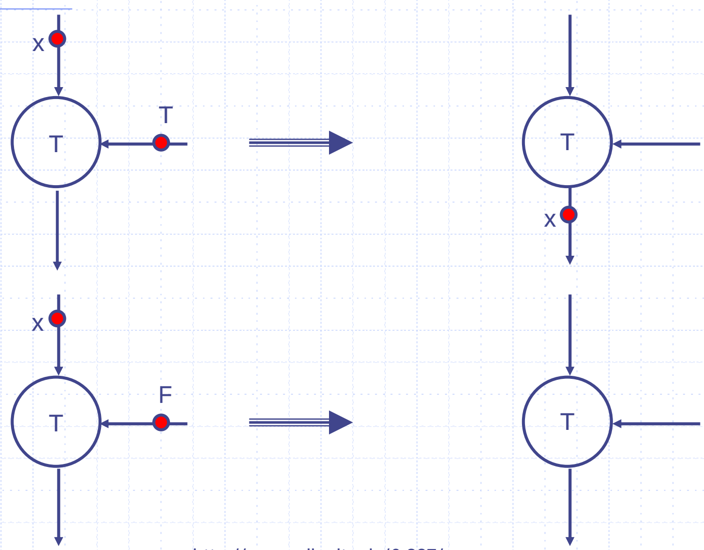
  Dennis

- Execution of an operation must not cause *side-effect* to preserve *determinacy*. The effect of an operation must be local.

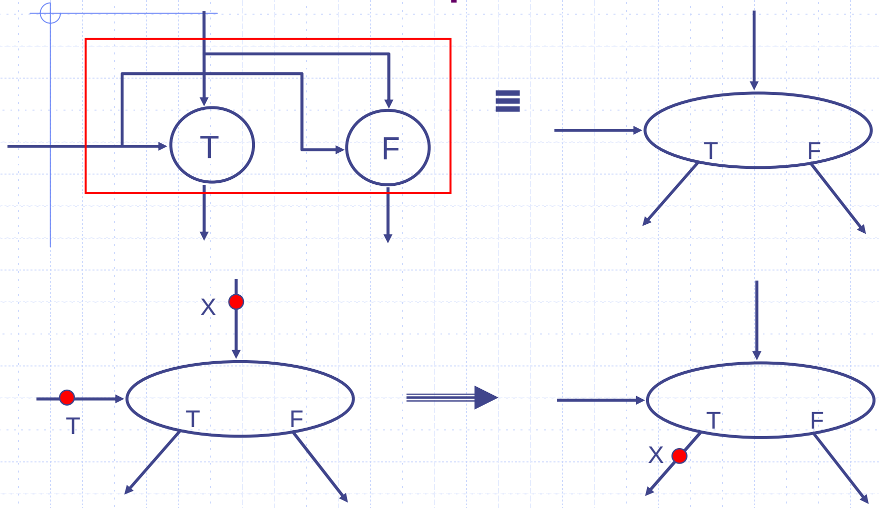
Firing Rules: Functional Operators



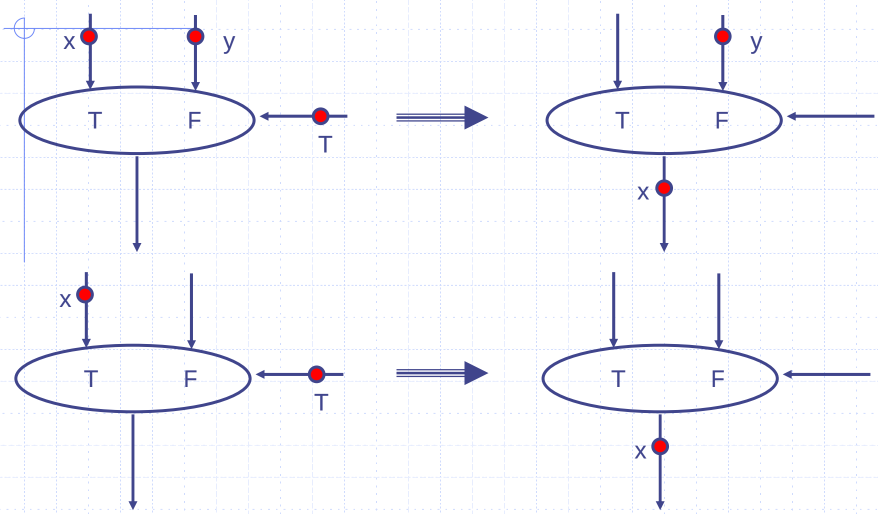
Firing Rules: T-Gate



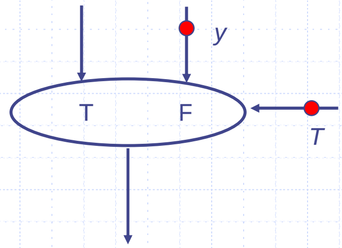
The Switch Operator



Firing Rules: Merge

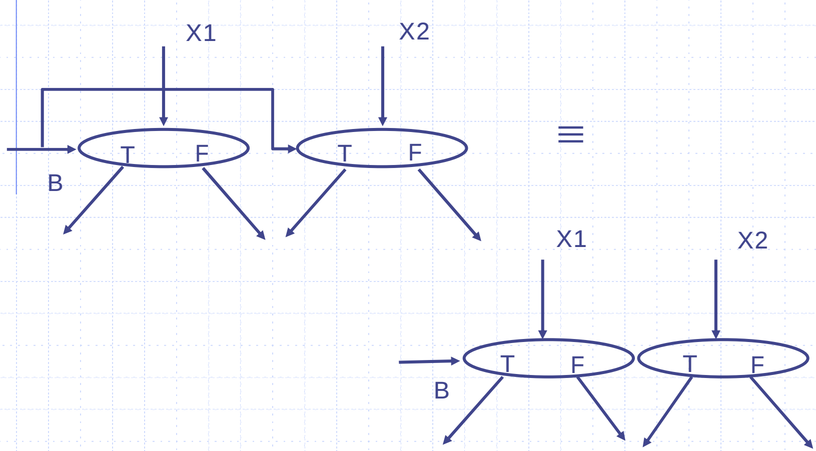


Firing Rules: Merge *cont*

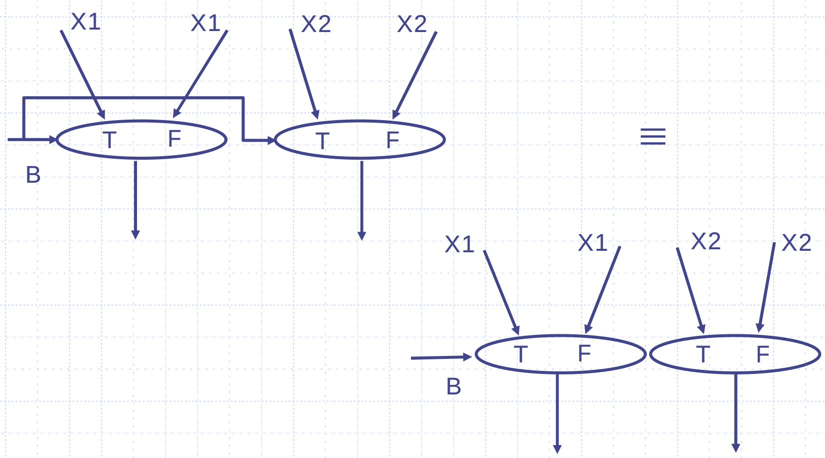


not ready
to fire

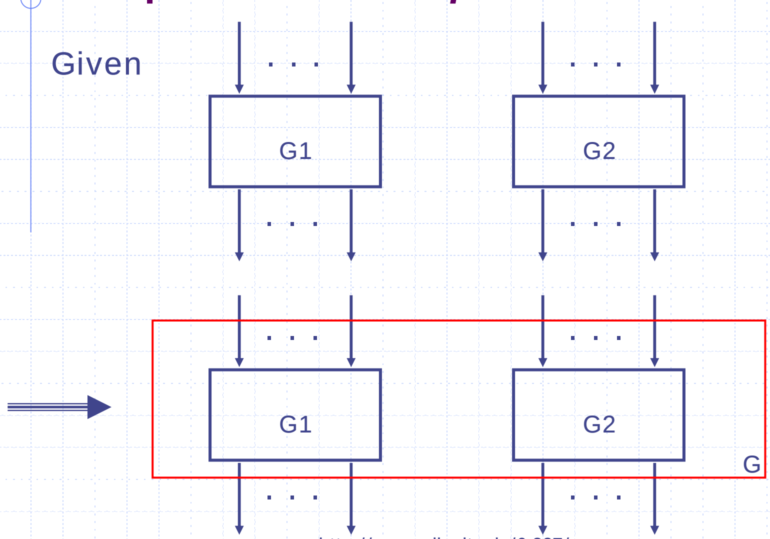
Some Conventions



Some Conventions *Cont.*

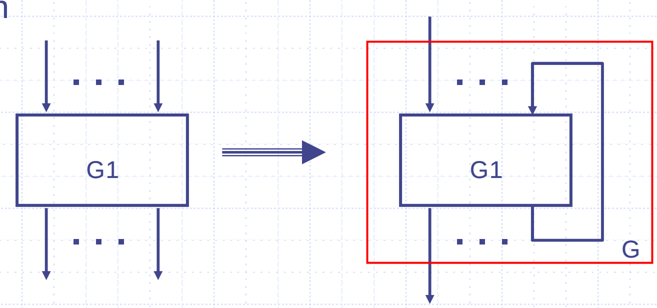


Rules To Form Dataflow Graphs: *Juxtaposition*



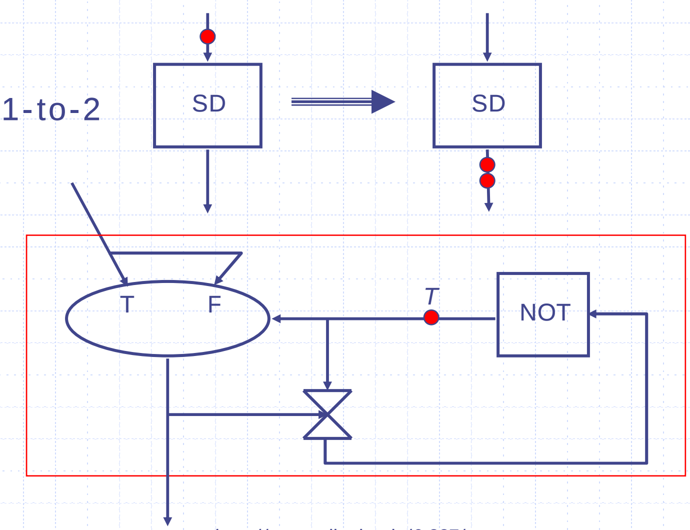
Rules To Form Dataflow Graphs: *Iteration*

Given

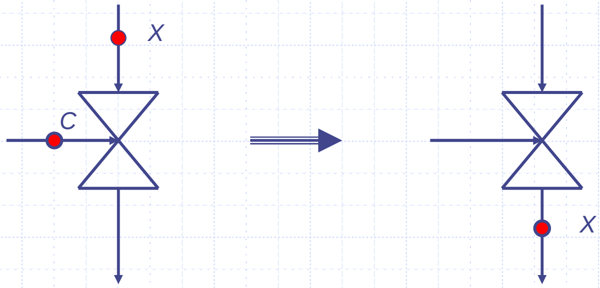


Example: The Stream Duplicator

1-to-2



The Gate Operator



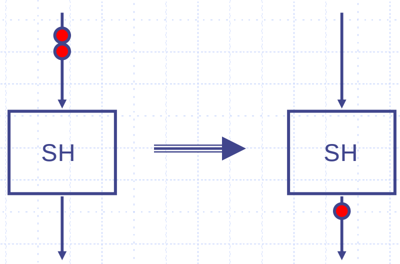
Lets X pass through only after C arrives.

What happens if we don't use the gate in the Stream Duplicator?

The Stream Halver

Throws away every other token.

2-to-1



Determinate Graphs

Graphs whose *behavior is time independent*, i.e., the values of output tokens are uniquely determined by the values of input tokens.

A dataflow graph formed by repeated *juxtaposition and iteration of deterministic dataflow operators* results in a deterministic graph.

Proof?

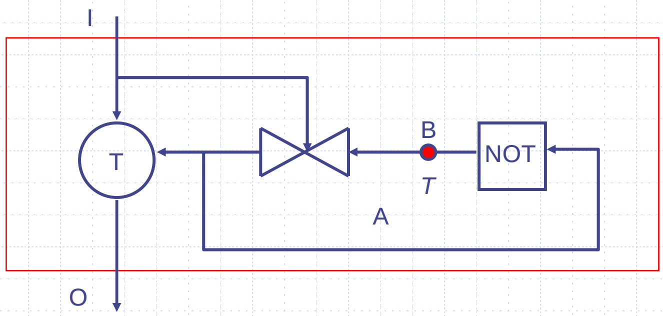
Dataflow Operators: Streams Functions

```
add(x:xs,y:ys)          =   +(x,y) : add(xs,ys)

T-gate (T:bs,x:xs)      =   x: T-gate(bs,xs)
T-gate (F:bs,x:xs)      =   T-gate(bs,xs)

merge(T:bs,x:xs,ys)     =   x: merge(bs,xs,ys)
merge(F:bs,xs,y:ys)     =   y: merge(bs,xs,ys)
```

Dataflow Graphs: A Set of Recursive Equations



O = T-gate (A,I) ;
A = gate (I,B) ;
B = T : Not (A) ;

G. Kahn

Domain of Sequences

Sequence: [x_1, … , x_n]

The least element: [ ] (aka $\perp$)

The partial order ($\leq$): prefix order on sequences

$$[\ ] \leq [x_1] \leq [x_1,x_2] \leq \ldots \leq [x_1, x_2, x_3 \ldots x_n]$$

[x_1, x_2, x_3] may be *approximated* by [ ] or [x_1] or[x_1,x_2]. However, [x_1,x_2] is a better approximation than [x_1] or [ ] for [x_1, x_2, x_3].

Kleene's Iterative Solution

I	[i1, i2, i3]		[i1, i2, i3]		[i1, i2, i3]	
A	[ ]		[T]		[T]	
B	[T]	$\Rightarrow$	[T]	$\Rightarrow$	[T, F]	$\Rightarrow$
O	[ ]		[ ]		[i1]	

O = T-gate (A,I) ;
A = gate (I,B) ;
B = T : Not (A) ;

Is the answer unique?

Yes, if all operators are *monotonic* and *continuous*!

Monotonicity

$\boxed{x \leq y \quad \Rightarrow \quad f(x) \leq f(y)}$

- a monotonic operator on sequences can only produce more output when given more input, i.e., it can never retract a value that has been produced.
- Is T-gate(B,X) monotonic ?

  $B \leq B' \quad \Rightarrow \quad \text{T-gate}(B,X) \leq \text{T-gate}(B',X)$

- The proof is straightforward by the induction on the length of the sequences

Continuity

$$\boxed{f\,(\textstyle\bigcup_i X_i) \;=\; \bigcup_i f\,(X_i)}$$

- A continuous operator on sequences does not suddenly produce an output after consuming an infinite amount of input.
- Is T-gate(B,X) continuous?

$$B_0 \le B_1 \le \ldots \le B_n \le \ldots \quad \Rightarrow$$
$$\text{T-gate}((\textstyle\bigcup_i B_i),X) = \bigcup_i \text{T-gate}(B_i,X)$$

- The proof is straightforward by the induction on the length of the sequences

Kleene's Fixed Point Theorem

If D is *partially ordered* with *one least element* ($\perp$) and is *ω-complete*, and

f : D --> D is a *monotonic* and *continuous* function then

$\bigcup_i f^i(\perp)$ is the least fixed point solution of f.

- *ω-complete* means that every chain ($\bigcup_i X_i$) in the domain has a least upper bound

$$\perp \le f(\perp) \le f(f(\perp)) \le \ldots$$

The limit of this chain is denoted by $\bigcup_i f_i(\perp)$