

Motivation

What is Dataflow

- Dataflow is a variant of Kahn Process Networks where a process is computed as a sequence of atomic firings, which are finite computations enabled by firing rules.
- Each firing rule must satisfy certain technical conditions to avoid nondeterminism.
- In a firing, an actor consumes a finite number of input tokens and produces a finite number of output tokens. A possibly infinite sequence of firings is called a dataflow process.

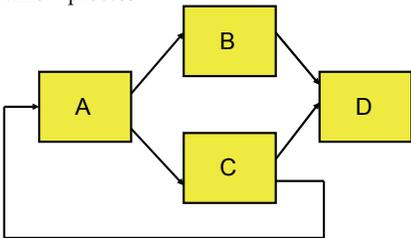


Figure 1: A dataflow graph with four actors.

Why Dataflow Programming

In a dataflow graph, each function node executes *concurrently* conceptually with the only constraint imposed by data availability. Therefore it greatly facilitates efficient use of concurrent resources in the implementation phase.

Where Dynamic Dataflow Sits

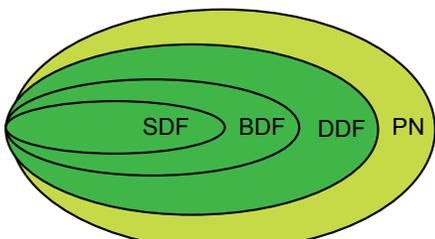


Figure 2: Dynamic dataflow (DDF) is a subset of process networks (PN) and a superset of synchronous dataflow (SDF) and Boolean Dataflow (BDF).

DDF Scheduling

- In SDF, each actor consumes and produces a fixed number of tokens, yielding compile-time scheduling.
- In BDF, certain dynamic actors such as Select and Switch are allowed, sometimes yielding compile-time scheduling.
- In DDF, all dataflow (dynamic or static) actors are allowed, which requires run-time scheduling. However, it avoids the complexities of context switching overhead of process suspension and resumption incurred in most implementations of PN by scheduling the actor firings, each of which is a finite quantum of computation.

DDF scheduling Criteria

- Correctness:** After any finite time every signal is a prefix of the LUB signal given by the denotational semantics.
- Liveness:** The scheduler should be able to execute a graph forever if it is possible to execute a graph forever. In particular, it should not stop prematurely if there are enabled actors.
- Boundedness:** The scheduler should be able to execute a graph forever in bounded memory if it is possible to execute the graph forever in bounded memory.
- Determinacy:** The scheduler should execute the graph in a sequence of well-defined and determinate iterations so that the user can control the length of an execution by specifying the number of iterations to execute.

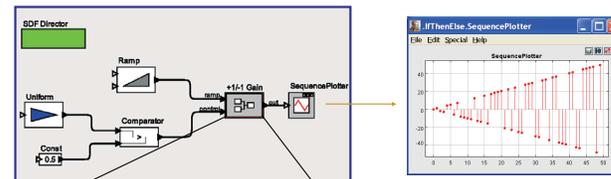
DDF scheduling algorithm

At the start of each basic iteration compute {
 E = set of enabled actors
 D = set of deferrable and enabled actors
 $\text{minimax}(D)$ = subset of D with the smallest maximum number of tokens on their output channels which satisfy the demands of destination actors.
 }

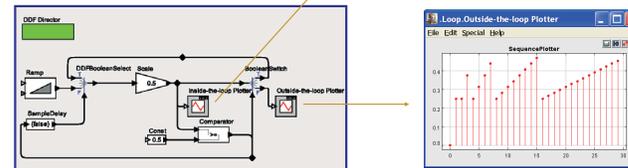
One basic (default) iteration consists of {
 If $(E \setminus D \neq \emptyset)$ fire $(E \setminus D)$
 else if $(D \neq \emptyset)$ fire $\text{minimax}(D)$
 else declare deadlock }

Examples in DDF Domain

Conditionals with If-Else Structure



Data-Dependent Iterations



Recursion: Sieve of Eratosthenes

