Alternative Syntaxes for Ptolemy Models

Chris Shaver
University of California, Berkeley

Block Combinator Syntaxes
- Ptolemy models can be represented in a combinator syntax rather than a point-to-point syntax with named ports.
- Models can be composed, edited, or reasoned about in such a syntax.

Blocks

$\begin{array}{c}
\text{Block Combinator Syntaxes} \\
\text{• Ptolemy models can be represented in a} \\
\text{combinator syntax rather than a point-to-point} \\
\text{syntax with named ports.} \\
\text{• Models can be composed, edited, or reasoned} \\
\text{about in such a syntax.} \\
\text{Abstract Syntax} \\
\end{array}$

$B^N_M$ signifies the type of a block in the combinator language with $N$ inputs and $M$ outputs.

Operators

Parallel Composition: $A \otimes B$

$\otimes : B^N_M \times B^K_P \rightarrow B^{N+K}_{M+P}$

Initiator/Terminator: $\text{init, term}$

Start with a Ptolemy model, the point-to-point syntax can be converted into a combinatorial visual syntax.

1. Split/Merge primitives replace multiply connected relations.
2. Initiator/Terminator primitives are added to unconnected ports.
3. Feedback edges are cut and drawn out to input/output pairs.
4. Organize blocks into columns dependent on predecessor columns.
5. Insert identity operators to make columns opaque.
6. Order identities to the bottom of columns.
7. Insert permutation primitives between columns.

Typical Ptolemy Model

$\begin{array}{c}
\text{Typical Ptolemy Model} \\
Expr_1 = \langle Expr_2 \rangle \\
Expr_2 = \text{init } \otimes \text{init} \\
\gg (1 3 4 5 2) \gg (2 4 3 1) \gg E \gg E \gg E \\
\gg (3 4 1 2) \gg (4 5 2 3 1 6) \gg E \gg E \\
\gg (4 1 6 2 5 7 8 3) \gg E \gg E \\
\end{array}$

Visual Combinator Form

Conversion from Ptolemy to Combinator Form

1. Split/Merge primitives replace multiply connected relations.
2. Initiator/Terminator primitives are added to unconnected ports.
3. Feedback edges are cut and drawn out to input/output pairs.
4. Organize blocks into columns dependent on predecessor columns.
5. Insert identity operators to make columns opaque.
6. Order identities to the bottom of columns.
7. Insert permutation primitives between columns.

Influences

• The Faust signal processing language.
• Diagrammatic Linear Algebra.
• Milner's Calculus of Communicating Systems.

Typical Ptolemy Model

$\begin{array}{c}
\text{Typical Ptolemy Model} \\
Expr_1 = \langle Expr_2 \rangle \\
Expr_2 = \text{init } \otimes \text{init} \\
\gg (1 3 4 5 2) \gg E \gg E \\
\gg (2 4 3 1) \gg E \gg E \gg <_3 \\
\gg (3 4 1 2) \gg E \gg E \gg <_4 \\
\gg (4 5 2 3 1 6) \gg E \gg E \gg <_2 \otimes <_2 \\
\gg (4 1 6 2 5 7 8 3) \gg E \gg E \\
\end{array}$

Influences

• The Faust signal processing language.
• Diagrammatic Linear Algebra.
• Milner's Calculus of Communicating Systems.