High-Performance Embedded Systems-on-a-Chip

Sanjay Rajopadhye
Computer Science, Colorado State University

Lecture 8: Review & Exercises

Prove closure of SURE's under same CoB

Consider the following SRE:

$$\begin{aligned} X[z] &= \{z \in D_X\} : g(\ldots, X[f_{xx}(z)], Y[f_{xy}(z)]) \\ Y[z] &= \{z \in D_Y\} : h(\ldots, X[f_{yx}(z)], Y[f_{yy}(z)]) \end{aligned}$$

and a bijective affine function, $\mathcal{T} : z \mapsto Tz + t$ where T is an integer unimodular matrix, and t is a vector. Show that when all the dependence functions are uniform (i.e., the SRE is actually a SURE) applying the same transformation, $\mathcal{T}$, to both X and Y, the resulting system remains uniform. Determine the new dependence vectors. Can you think of a (slightly) more general transformation that retains this closure property?

Solution

Note: $\mathcal{T}^{-1}(z) = T^{-1}z - T^{-1}t$, since

Solution

Note: $\mathcal{T}^{-1}(z) = T^{-1}z - T^{-1}t$, since

$T^{-1}(Tz + t) - T^{-1}t = z$

Solution

Note: $\mathcal{T}^{-1}(z) = T^{-1}z - T^{-1}t$, since

$$T^{-1}(Tz + t) - T^{-1}t = z$$

Applying CoB by $\mathcal{T}$ first to X

$$\begin{aligned} X[z] &= \{z \in \mathcal{T}(D_X)\} : g(\ldots, X[\mathcal{T} \circ f_{xx} \circ \mathcal{T}^{-1}(z)], \\ &\qquad Y[f_{xy} \circ \mathcal{T}^{-1}(z)]) \\ Y[z] &= \{z \in D_Y\} : h(\ldots, X[\mathcal{T} \circ f_{yx}(z)], \\ &\qquad Y[f_{yy}(z)]) \end{aligned}$$

. . . and then a CoB by $\mathcal{T}$ to Y:

. . . and then a CoB by $\mathcal{T}$ to Y:

$$
\begin{aligned}
X[z] &= \{z \in \mathcal{T}(D_X)\} : g(\ldots, X[\mathcal{T} \circ f_{xx} \circ \mathcal{T}^{-1}(z)], \\
&\qquad Y[\mathcal{T} \circ f_{xy} \circ \mathcal{T}^{-1}(z)]) \\
Y[z] &= \{z \in \mathcal{T}(D_Y)\} : h(\ldots, X[\mathcal{T} \circ f_{yx} \circ \mathcal{T}^{-1}(z)], \\
&\qquad Y[\mathcal{T} \circ f_{yy} \circ \mathcal{T}^{-1}(z)])
\end{aligned}
$$

Each new dependence is of the form $\mathcal{T} \circ f \circ \mathcal{T}^{-1}(z)$, where f is uniform, i.e., $f(z) = z + \delta$

$$
\begin{aligned}
\mathcal{T} \circ f \circ \mathcal{T}^{-1}(z) &= Tf(T^{-1}z - T^{-1}t) + t \\
&= T(T^{-1}z - T^{-1}t + \delta) + t \\
&= (z - t + T\delta) + t \\
&= z + T\delta
\end{aligned}
$$

Bubble Sort

Starting SURE(subscripts on lhs for compaction)

$$
\begin{aligned}
y_i &= \begin{cases} i = 1 & : \; m[n, n-1] \\ i > 1 & : \; M[n, n-i+1] \end{cases} \\
m_{i,j} &= \begin{cases} i = 2; j = 1 & : \; \min(x_1, x_2) \\ 2 < i \leq n; j = 1 & : \; \min(M[i-1, j], x_i) \\ 3 \leq j+1 = i \leq n & : \; \min[m[i-1, j-1], m[i, j-1]) \\ j+2 \leq i \leq n; 1 < j & : \; \min(M[i-1, j], m[i, j-1]) \end{cases} \\
M_{i,j} &= \begin{cases} i = 2; j = 1 & : \; \max(x_1, x_2) \\ 2 < i \leq n; j = 1 & : \; \max(M[i-1, j], x_i) \\ 3 \leq j+1 = i \leq n & : \; \max[m[i-1, j-1], m[i, j-1] \\ j+2 \leq i \leq n; 1 < j & : \; \max(M[i-1, j], m[i, j-1]) \end{cases}
\end{aligned}
$$

Space-time Transformation

Schedule: $t(i, j) = i + j$ (for both variables)

Space-time Transformation

Schedule: $t(i, j) = i + j$ (for both variables)

Allocation: $a(i, j) = j$ (for both variabales).

Space-time Transformation

Schedule: $t(i, j) = i + j$ (for both variables)

Allocation: $a(i, j) = j$ (for both variabales).

(Same old) CoB transformation: $\mathcal{T} = (i, j \rightarrow i + j, j)$

Space-time Transformation

Schedule: $t(i,j) = i + j$ (for both variables)

Allocation: $a(i,j) = j$ (for both variabales).

(Same old) CoB transformation: $\mathcal{T} = (i,j \rightarrow i+j, j)$

$$\begin{pmatrix} t \\ p \end{pmatrix} = \mathcal{T}\begin{pmatrix} i \\ j \end{pmatrix} = \begin{pmatrix} i+j \\ j \end{pmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{pmatrix} i \\ j \end{pmatrix}$$

Transformed SURE

$$y_i = \begin{cases} i = 1 \ : \ m[2n-1, n-1] \\ i > 1 \ : \ M[2n-i+1, n-i+1] \end{cases}$$

$$m_{t,p} = \begin{cases} t = 3; p = 1 & : \ \min(x_1, x_2) \\ 3 < t \leq n+1; p = 1 & : \ \min(M[t-1, p], x_{t-1}) \\ 5 \leq t = 2p+1 \leq n+p & : \\ \qquad \min[m[t-2, p-1], m[t-1, p-1]) \\ 2p+2 \leq t \leq n+p; 1 < p & : \\ \qquad \min(M[t-1, p], m[t-1, p-1]) \end{cases}$$

$M_{t,p} =$ similar

I/O and Control

Processors involved in I/O operations:

I/O and Control

Processors involved in I/O operations: function of

- Alignment of the I/O variables
- Allocation function

I/O and Control

Processors involved in I/O operations: function of

- Alignment of the I/O variables
- Allocation function

Basic Rule: I/O must occur at the boundary processors