An algebra of scans

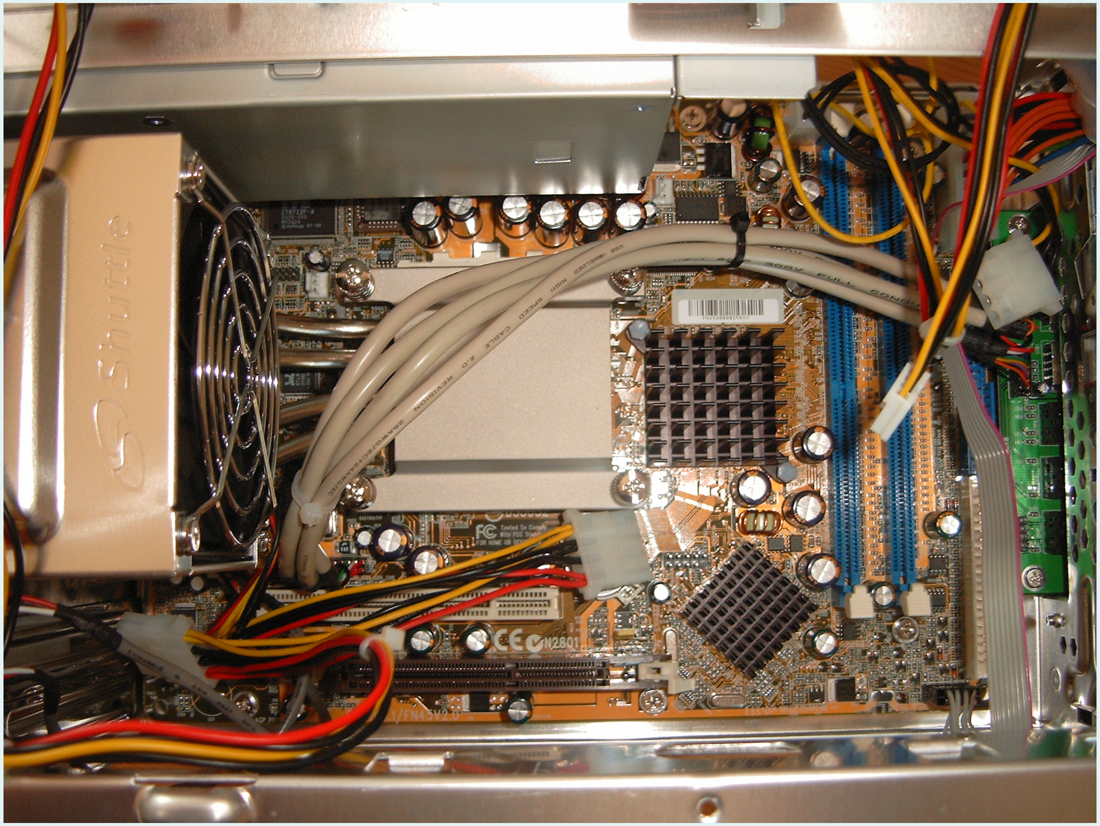
RALF HINZE

Institut für Informatik III, Universität Bonn
Römerstraße 164, 53117 Bonn, Germany
Email: `ralf@informatik.uni-bonn.de`
Homepage: `http://www.informatik.uni-bonn.de/~ralf`

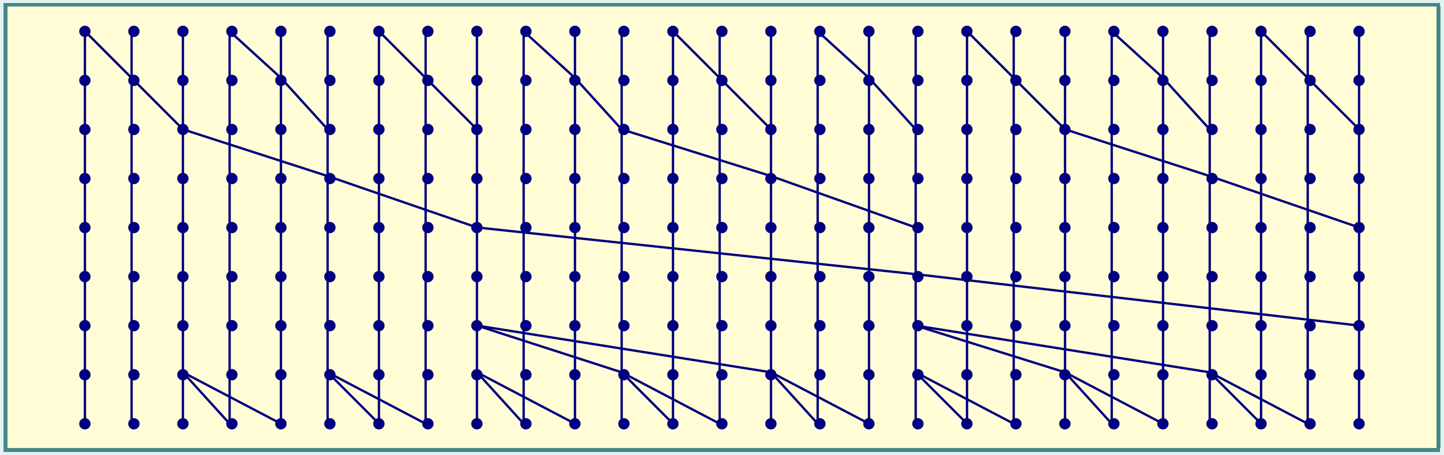
July, 2004

(Pick the slides at `.../~ralf/talks.html#T35`.)

. . . too concrete



. . . more abstract



Parallel prefix circuits or scans

A parallel prefix circuit or scan takes n inputs

$$x_1, x_2, \ldots, x_n$$

and produces the n outputs

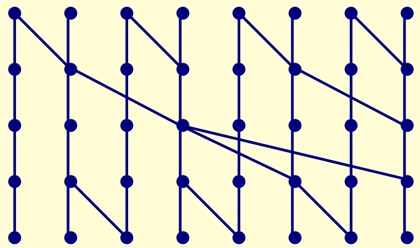
$$x_1, x_1 \circ x_2, \ldots, x_1 \circ x_2 \circ \cdots \circ x_n,$$

where '$\circ$' is an arbitrary associative binary operation.

☞ Range of applications: fast integer addition, parallel sorting, convex hull problems.

Scans as directed acyclic oriented graphs

A scan can be modelled as a directed acyclic oriented graph.



The edges are directed downwards; a node of in-degree two, an operation node, represents the 'sum' of its two inputs; a node of in-degree one and out-degree greater than one, a duplication node, distributes its input to its outputs.

☞ Measures: size, depth, fan-out (maximal out-degree of an operation node), height difference (length of the path from the first input to the last output).

Aim of the talk

☞ A description in form of a graph obscures the structure of a scan and is hard to manipulate.

Define and manipulate scans algebraically.

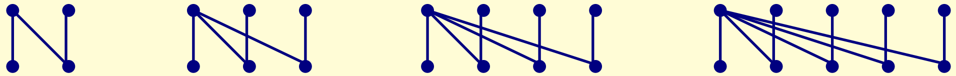
☞ Using only two basic building blocks (fan and id) and four combinators ($\times$, $;$, $\succ\!-$, $-\!\prec$) all standard designs can be described succinctly and rigorously.

Outline of the talk

Fans

LORD DARLINGTON. . . . *[Sees a fan lying on the table.] And what a wonderful fan! May I look at it?*
LADY WINDERMERE. *Do. Pretty, isn't it! It's got my name on it, and everything. I have only just seen it myself. It's my husband's birthday present to me. You know to-day is my birthday?*
— Oscar Wilde, *Lady Windermere's Fan*

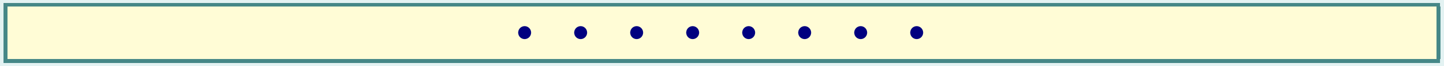
☞ A scan can be seen as a composition of fans, denoted fan_n.



A fan adds its first input—counting from left to right—to each of its remaining inputs. It consists of a duplication node and $n-1$ operation nodes.

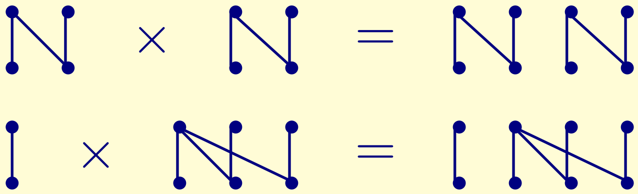
Identity circuits

The identity circuit of width n is denoted id_n.



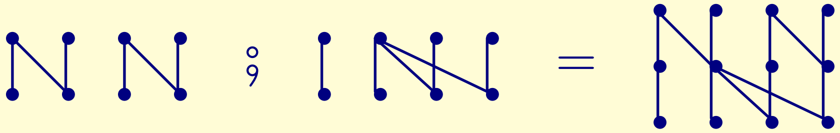
Parallel or horizontal composition

Placing two circuits side by side is called parallel or horizontal composition, denoted ‘$\times$’.



Serial or vertical composition

Placing two circuits on top of each other is called serial or vertical composition, denoted '$\fatsemi$'.



☞ We require that the two circuits have the same width.

Laws: composition

The combinators have to satisfy a number of laws: '$⨾$' is associative with id_n as its neutral element; '$\times$' is associative with id_0 as its neutral element; '$\times$' preserves identity and vertical composition ($|f|$ denotes the width of f).

$$\begin{aligned} id_{|f|} ⨾ f &= f \\ f ⨾ id_{|f|} &= f \\ f ⨾ (g ⨾ h) &= (f ⨾ g) ⨾ h \end{aligned} \qquad \begin{aligned} id_0 \times f &= f \\ f \times id_0 &= f \\ f \times (g \times h) &= (f \times g) \times h \\ id_m \times id_n &= id_{m+n} \\ (f \times g) ⨾ (f' \times g') &= (f ⨾ f') \times (g ⨾ g') \end{aligned}$$

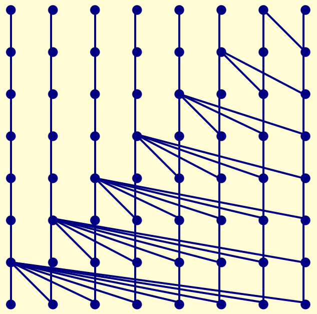
☞ These laws are purely structural: they do not depend on the associativity of the underlying binary operation '$\circ$' (simply because they do not involve fans).

Specification

We specify scans as follows (scans as repeated folds):

$$\begin{array}{lcl} scan_0 & = & id_0 \\ scan_{n+1} & = & succ\ scan_n \\ succ\ f & = & id_1 \times f \fatsemi fan_{|f|+1} \end{array}$$

Here is a scan of width 8.



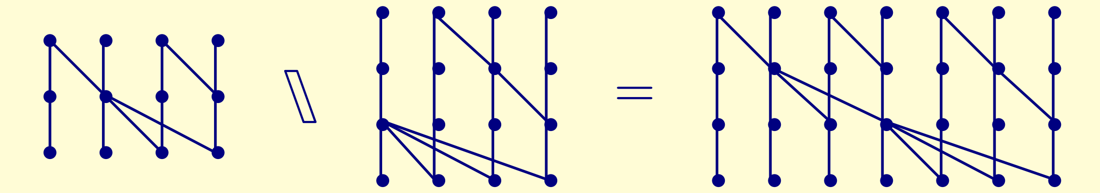
Outline of the talk

Serial or vertical composition of scans

Using the basic building blocks we can define derived combinators, for instance, the serial or vertical composition of scans.

$$\boxed{f \mathbin{\backslash\!\!\backslash} g \;=\; f \times id_{|g|-1} \mathbin{⨾} id_{|f|-1} \times g}$$

The last output of the first circuit is fed into the first input of the second circuit.

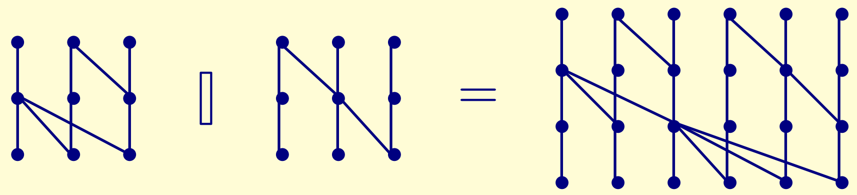


Parallel or horizontal composition of scans

The second scan combinator is the parallel or horizontal composition of scans:

$$f \;\square\; g \;=\; f \times g \fatsemi id_{|f|-1} \times fan_{|g|+1}$$

Both circuits are placed side by side, an additional fan adds the last output of the left circuit to each output of the right circuit.



Properties

$$
\begin{array}{rclcrcl}
id_1 \mathbin{\backslash\!\backslash} f & = & f & \qquad & f \mathbin{\Box} id_0 & = & f \\
f \mathbin{\backslash\!\backslash} id_1 & = & f & & f \mathbin{\Box} (g \mathbin{\Box} h) & = & (f \mathbin{\Box} g) \mathbin{\Box} h \\
f \mathbin{\backslash\!\backslash} (g \mathbin{\backslash\!\backslash} h) & = & (f \mathbin{\backslash\!\backslash} g) \mathbin{\backslash\!\backslash} h & & f \mathbin{\Box} g & = & f \mathbin{\backslash\!\backslash} succ\ g
\end{array}
$$

☞ The last law on the right shows that the parallel composition of scans is a serial composition in disguise ($f \mathbin{\Box} g$ and $f \mathbin{\backslash\!\backslash} succ\ g$ are even structurally equal).

Furthermore, $succ$, '$\backslash\!\backslash$' and '$\Box$' are scan combinators.

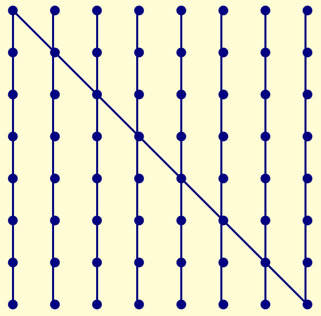
$$
\begin{array}{rcl}
succ\ scan_n & = & scan_{n+1} \\
scan_{m+1} \mathbin{\backslash\!\backslash} scan_n & = & scan_{m+n} \\
scan_m \mathbin{\Box} scan_n & = & scan_{m+n}
\end{array}
$$

Serial scans

If the parallel composition is bracketed to the left, we obtain the serial scan.

$$\begin{array}{lcl} ser_0 & = & id_0 \\ ser_1 & = & id_1 \\ ser_{n+1} & = & ser_n \,[\!]\, id_1 \end{array}$$

The graphical representation illustrates why ser_n is called serial scan.



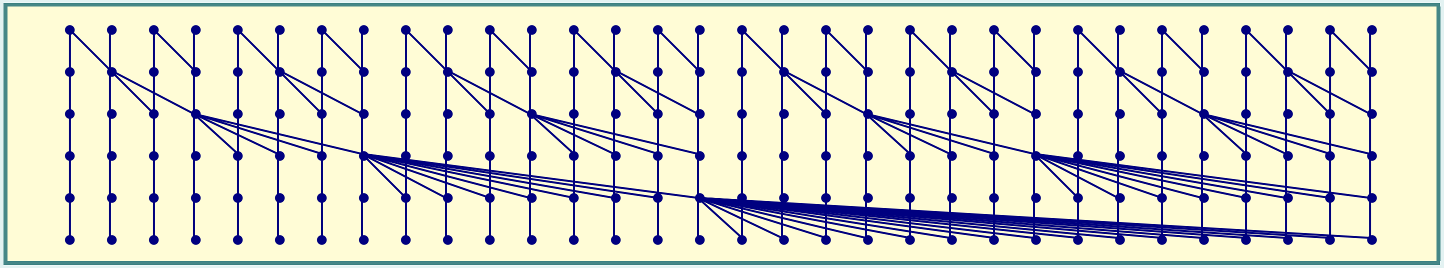
☞ The serial scan has maximum depth, but the least number of operation nodes, namely, $n-1$ among all scans of the same width.

Minimum depth scans

If we balance the parallel composition evenly, we obtain scans of minimum depth.

$$
\begin{array}{lll}
\multicolumn{3}{l}{rec_n} \\
\quad \mid n \leqslant 1 & = & id_n \\
\quad \mid otherwise & = & rec_{\lceil n/2 \rceil} \,[\!]\, rec_{\lfloor n/2 \rfloor}
\end{array}
$$

Here is a minimum-depth circuit of width 32.

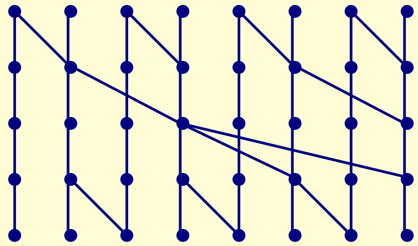


☞ The tree of operation nodes that computes the last output is fully balanced, which explains why the depth is minimal.

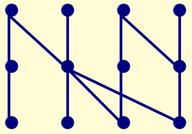
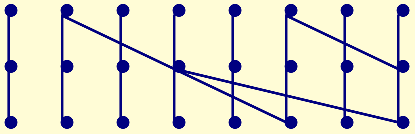
Outline of the talk

Stretch combinators

Horizontal and vertical composition, however, are not sufficient.



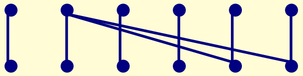
The middle part is stretched by a factor of two:

$[2, 2, 2, 2] \succ$  $=$ 

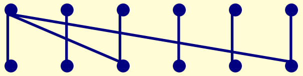
Stretch combinators—continued

Another stretch combinator is ‘$\prec$’ which is similar to ‘$\succ$’ except that it connects the first input of each group to its argument circuit.

$$[2,3,1] \succ fan_3 =$$



$$fan_3 \prec [2,3,1] =$$



The inputs of the resulting circuit are grouped according to the given positive widths. The last respectively first input of each group is connected to the argument circuit; the other inputs are wired through.

☞ ‘$\succ$’ is useful for combining scans, while ‘$\prec$’ is a natural choice for combining fans.

Derived combinators

More derived combinators:

$$par \;=\; foldr\ (\times)\ id_0$$

The combinator par generalizes '$\times$' and places a list of circuits side by side.

$$\begin{aligned} fs \succ f &= par\ fs \mathbin{;} |fs| \rightarrowtail f \\ f \prec fs &= f \leftarrowtail |fs| \mathbin{;} par\ fs \end{aligned}$$

The combinators '$\succ$' and '$\prec$' are convenient variants of '$\rightarrowtail$' and '$\leftarrowtail$': the expression $f \prec [f_1, \ldots, f_n]$ connects the i-th output of f to the first input of f_i while $[f_1, \ldots, f_n] \succ f$ connects the last output of f_i to the i-th input of f.

Laws: stretching

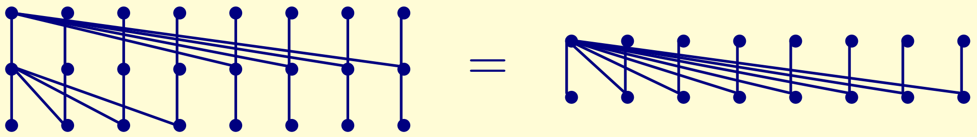
The combinators ‘$-\!\!\!\prec$’ and ‘$\succ\!\!\!-$’ have to satisfy a number of laws:

$$
\begin{array}{rcl}
id_{\#x} -\!\!\!\prec x & = & id_{\Sigma x} \\
f -\!\!\!\prec \mathit{replicate}\ |f|\ 1 & = & f \\
(f \fatsemi g) -\!\!\!\prec x & = & (f -\!\!\!\prec x) \fatsemi (g -\!\!\!\prec x) \\
(f \times g) -\!\!\!\prec (x +\!\!\!+ y) & = & (f -\!\!\!\prec x) \times (g -\!\!\!\prec y) \\
(f -\!\!\!\prec x) -\!\!\!\prec y & = & f -\!\!\!\prec [\Sigma z \mid z \leftarrow \mathit{group}\ x\ y] \\
id_{i-1} \times (f -\!\!\!\prec y +\!\!\!+ [k]) & = & ([i] +\!\!\!+ y \succ\!\!\!- f) \times id_{k-1}
\end{array}
$$

☞ ‘$-\!\!\!\prec$’ preserves identity and composition ($\mathit{replicate}\ n\ a$ constructs a list containing exactly n copies of a). The second but last law shows that nested occurrences of stretch combinators can be flattened. The last equation, termed flip law, shows that ‘$-\!\!\!\prec$’ can be defined in terms of ‘$\succ\!\!\!-$’ and vice versa.

Laws: fan—trading depth for fan-out

The first fan law allows the designer of scans to trade depth for fan-out.



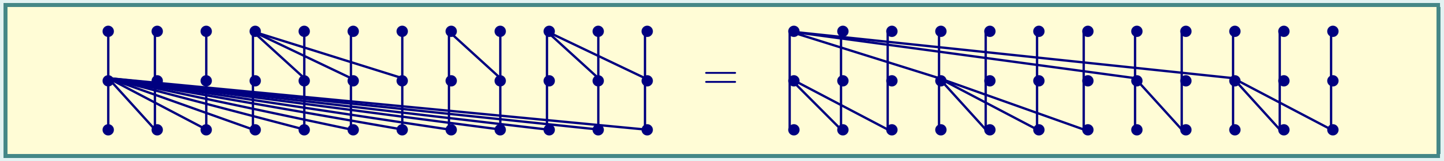
The circuit on the left has a depth of 2 and a fan-out of 5 while the circuit on the right has depth 1 and fan-out 8.

$$fan_{1+n} \prec [fan_m \prec fs] +\!\!+\, gs \;=\; fan_{m+n} \prec fs +\!\!+\, gs$$

☞ This rule is also structural as it does not rely on the associativity of the underlying operator.

Laws: fan—optimizing scans

The second fan law finally employs the associativity of '$\circ$'.



The left circuit consists of a big fan below a layer of smaller fans. The big fan adds its first input to each of the intermediate values; the same effect is achieved on the right by broadcasting the first input to each of the smaller fans.

$$\begin{aligned} & id_{1+\#x} \prec [\,id_i\,] \mathbin{+\!\!+} [\,fan_j \mid j \leftarrow x\,] \mathbin{;} fan_{i+\Sigma x} \\ = \; & fan_{1+\#x} \prec [\,fan_i\,] \mathbin{+\!\!+} [\,fan_j \mid j \leftarrow x\,] \end{aligned}$$

The size of the right circuit is at most the size of the left circuit while the depth of both circuits is the same.

☞ The second fan law allows us to optimize scans.

Laws: fan–summary

Summary:

$$\begin{aligned}
fan_0 &= id_0 \\
fan_1 &= id_1 \\
fan_{1+n} \prec [fan_m \prec fs] \mathbin{+\!\!\!+} gs &= fan_{m+n} \prec fs \mathbin{+\!\!\!+} gs \\
id_{1+\#x} \prec [id_i] \mathbin{+\!\!\!+} [fan_j \mid j \leftarrow x] \mathbin{⨾} fan_{i+\Sigma x} & \\
&= fan_{1+\#x} \prec [fan_i] \mathbin{+\!\!\!+} [fan_j \mid j \leftarrow x]
\end{aligned}$$

Derived law: a binary version of the second fan law.

$$id_m \times fan_{n+1} \mathbin{⨾} fan_{m+n+1} \;=\; fan_{1+m} \times id_n \mathbin{⨾} id_m \times fan_{n+1}$$

Outline of the talk

Associativity of parallel scan composition

☞ Recall: The parallel composition of scans is a serial composition in disguise: $f \mathbin{\Box} g = f \mathbin{\backslash\!\backslash} succ\ g$.

$$
\begin{array}{cl}
 & f \mathbin{\Box} (g \mathbin{\Box} h) \\
= & \quad \{ \text{ characterization of `}\Box\text{' } \} \\
 & f \mathbin{\backslash\!\backslash} succ\ (g \mathbin{\backslash\!\backslash} succ\ h) \\
= & \quad \{ \text{ proof obligation } \} \\
 & f \mathbin{\backslash\!\backslash} (succ\ g \mathbin{\backslash\!\backslash} succ\ h) \\
= & \quad \{ \text{ `}\backslash\!\backslash\text{' is associative } \} \\
 & (f \mathbin{\backslash\!\backslash} succ\ g) \mathbin{\backslash\!\backslash} succ\ h \\
= & \quad \{ \text{ characterization of `}\Box\text{' } \} \\
 & (f \mathbin{\Box} g) \mathbin{\Box} h
\end{array}
$$

☞ $(f \mathbin{\Box} g) \mathbin{\Box} h$ is better than $f \mathbin{\Box} (g \mathbin{\Box} h)$.

Proof obligation

It remains to show the proof obligation:

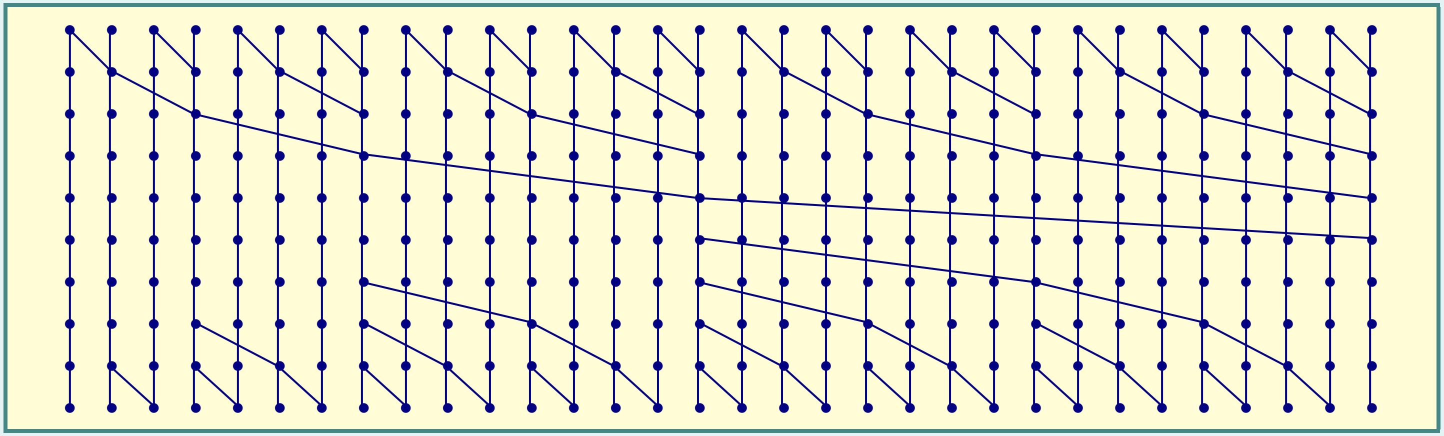
$$
\begin{array}{ll}
 & succ\ (f \mathbin{\backslash\!\backslash} succ\ g) \\
= & \quad \{\ \text{definition of } succ \text{ and } `\mathbin{\backslash\!\backslash}'\ \} \\
 & id_1 \times f \times id_{|g|} \mathbin{;} id_{|f|+1} \times g \mathbin{;} id_{|f|} \times fan_{|g|+1} \mathbin{;} fan_{|f|+|g|+1} \\
= & \quad \{\ \text{derived fan law}\ \} \\
 & id_1 \times f \times id_{|g|} \mathbin{;} id_{|f|+1} \times g \mathbin{;} fan_{|f|+1} \times id_{|g|} \mathbin{;} id_{|f|} \times fan_{|g|+1} \\
= & \quad \{\ \text{composition}\ \} \\
 & id_1 \times f \times id_{|g|} \mathbin{;} fan_{|f|+1} \times id_{|g|} \mathbin{;} id_{|f|+1} \times g \mathbin{;} id_{|f|} \times fan_{|g|+1} \\
= & \quad \{\ \text{definition of } succ \text{ and } `\mathbin{\backslash\!\backslash}'\ \} \\
 & succ\ f \mathbin{\backslash\!\backslash} succ\ g
\end{array}
$$

Since the proof relies on the second fan law, $succ\ f \mathbin{\backslash\!\backslash} succ\ g$ has fewer nodes than $succ\ (f \mathbin{\backslash\!\backslash} succ\ g)$.

Outline of the talk

Brent-Kung scans

The rec_n family of circuits implements a simple divide-and-conquer scheme. A different recursive decomposition was devised by Brent and Kung.



The inputs are 'paired' using a layer of 2-fans. Every second output is then fed into a Brent-Kung circuit of half the width; the other inputs are wired through. A final layer of 2-fans, shifted by one position, distributes the results of the nested Brent-Kung circuit to the wired-through signals.

Brent-Kung scans—continued

The first layer of 2-fans can be generalized to a layer of scans of arbitrary, not necessarily equal widths. So, here is yet another scan combinator:

$$\begin{array}{lcl} [\,] \rhd g & = & g \\ (f : fs) \rhd g & = & (f : fs) \succ g \,⨾\, id_{|f|-1} \times par\ gs \\ \quad \textbf{where}\ gs & = & [fan_{|f|} \mid f \leftarrow fs] +\!\!+ [id_1] \end{array}$$

The Brent-Kung circuit is defined

$$\begin{array}{lcl} bk_n & & \\ \quad \mid n \leqslant 1 & = & id_n \\ \quad \mid otherwise & = & (replicate\ \lfloor n/2 \rfloor\ fan_2 +\!\!+ [id_1 \mid odd\ n]) \rhd bk_{\lceil n/2 \rceil} \end{array}$$

Brent-Kung circuits have logarithmic (not minimum) depth, but they use fewer operation nodes than the rec_n circuits and they have only a fan-out of 2!

Ladner-Fischer scans

☞ Observation: the left part of the rec_n circuit does not use the bottom level.

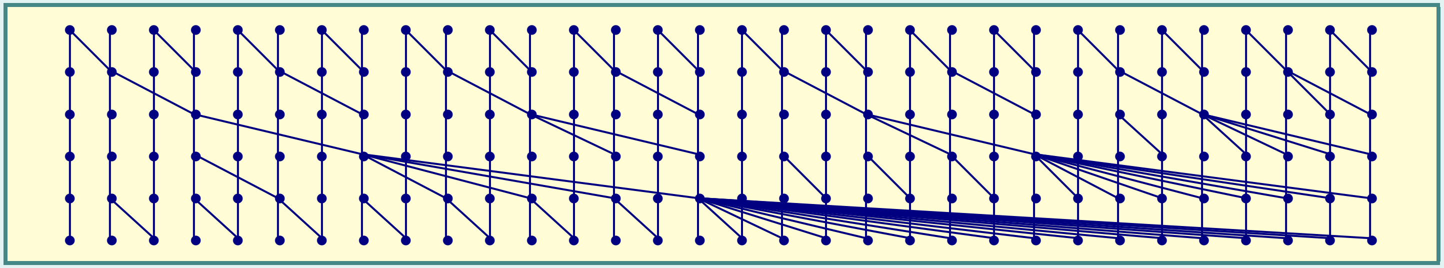
This motivates the following depth-optimal scan that has the minimal number of operation nodes among all minimum-depth circuits:

$$\begin{array}{lcl}
opt\ n & & \\
\quad \mid n \leqslant 1 & = & id_n \\
\quad \mid otherwise & = & stretch\ opt\ \lceil n/2 \rceil \mathbin{[]} opt\ \lfloor n/2 \rfloor \\
stretch\ s\ n & = & (replicate\ \lfloor n/2 \rfloor\ fan_2 \mathbin{+\!\!+} [id_1 \mid odd\ n]) \rhd s\ \lceil n/2 \rceil
\end{array}$$

☞ $stretch$ captures the recursive step of Brent-Kung.

Ladner-Fischer scans–continued

Here is a Ladner-Fischer scan of width 32, which illustrates that all layers are nicely utilized.



Outline of the talk

Conclusion

- Scans enjoy a surprisingly rich algebra.
- The algebraic approach has several benefits: it allows us to specify scans in a readable and concise way, to prove them correct, and to derive new designs.
- Almost all the laws are structural; only the second fan law relies on the associativity of the underlying operator.

Related work:

- Scans in parallel programming: correspond to clocked circuits while we study purely combinatorial ones.