Building Algebraic Structures with Combinators

Vilius Naudžiūnas Timothy G. Griffin

Vilius.Naudziunas@cl.cam.ac.uk timothy.griffin@cl.cam.ac.uk
Computer Laboratory
University of Cambridge, UK

Model-Based Systems Engineering Colloquium
ECE — University of Maryland
7 November, 2011

Semirings

A few examples

name	S	$\oplus$,	$\otimes$	$\overline{0}$	$\overline{1}$	possible routing use
sp	$\mathbb{N}^{\infty}$	min	$+$	∞	0	minimum-weight routing
bw	$\mathbb{N}^{\infty}$	max	min	0	∞	greatest-capacity routing
rel	$[0,\ 1]$	max	$\times$	0	1	most-reliable routing
use	$\{0,\ 1\}$	max	min	0	1	usable-path routing
	2^W	$\cup$	$\cap$	$\{\}$	W	shared link attributes?
	2^W	$\cap$	$\cup$	W	$\{\}$	shared path attributes?

Path problems focus on global optimality

$$\mathbf{A}^*(i,\ j) = \bigoplus_{p \in P(i,\ j)} w(p)$$

There are many **generic** algorithms ...

Encyclopaedic survey

Graphs, Dioids and Semirings: New Models and Algorithms, *M. Gondran and M. Minoux*, 2008.

Suppose you have a **library** of generic algorithms (some may be distributed algorithms for network routing).

The problem

How do we construct (complex) algebraic structures to use a selected generic algorithms?

Properties needed by some algorithms ...

description	P	**meaning**
Associativity	ass	$\forall x\ y\ z \in S, x \circ (y \circ z) = (x \circ y) \circ z$
Commutativity	com	$\forall x\ y \in S, x \circ y = y \circ x$
Idempotence	idm	$\forall x \in S, x \circ x = x$
Selectivity	sel	$\forall x\ y \in S, x \circ y \in \{x, y\}$
Identity	ide	$\exists i \in S, \forall x \in S, i \circ x = x = x \circ i$
Annihilator	ann	$\exists w \in S, \forall x \in S, w \circ x = w = x \circ w$
L Consistency	l.con	$\mathcal{W}(\text{ide}(S, \oplus)) = \mathcal{W}(\text{ann}(S, \otimes))$
R Consistency	r.con	$\mathcal{W}(\text{ide}(S, \otimes)) = \mathcal{W}(\text{ann}(S, \oplus))$
L absorbing	abs	$\forall x\ y \in S, x \oplus (y \otimes x) = x$
L strict absorbing	str	$\forall x\ y \in S, x \oplus (y \otimes x) = x \wedge x \neq y \otimes x$
L distributivity	l.d	$\forall x\ y\ z \in S, z \otimes (x \oplus y) = (z \otimes x) \oplus (z \otimes y)$
R distributivity	r.d	$\forall x\ y\ z \in S, (x \oplus y) \otimes z = (x \otimes z) \oplus (y \otimes z)$

$\mathcal{W}(\exists x \in S, P(x))$ represents an element $s \in S$ such that $P(s)$ holds.

Approach : a domain-specific language for algebraic structures

Starting with an initial set of properties $\mathcal{P}_0$ ...

Our (fuzzy wuzzy) goals

- Define a language $\mathcal{L}$,
- a well-formedness condition $\mathrm{WF}(E)$, for $E \in \mathcal{L}$,
- and a set of properties $\mathcal{P}$, with $\mathcal{P}_0 \subseteq \mathcal{P}$

so that properties are decidable for well-formed expressions:

$$\forall \mathrm{Q} \in P : \forall E \in \mathcal{L} : \mathrm{WF}(E) \implies (\mathrm{Q}([\![E]\!]) \vee \neg \mathrm{Q}([\![E]\!]))$$

The logic is constructive!

The challange: increase expressive power while preserving decidability ...

Combinators for binary operations ...

- $\circ \in S \times S \to S$
- $\mathrm{id}\ c \circ \in (S \uplus \{c\}) \times (S \uplus \{c\}) \to (S \uplus \{c\})$

where

$$S \uplus T = \{\mathrm{inl}(s) \mid s \in S\} \cup \{\mathrm{inr}(t) \mid t \in T\}$$

$$\begin{aligned} \mathrm{inr}(c) \bullet x &= x, \\ x \bullet \mathrm{inr}(c) &= x, \\ \mathrm{inl}(s_1) \bullet \mathrm{inl}(s_2) &= \mathrm{inl}(s_1 \circ s_2). \end{aligned}$$

where $\bullet = \mathrm{id}\ c \circ$

... in a similar way ...

- $\circ \in S \times S \to S$
- $\text{ann } c \circ \in (S \uplus \{c\}) \times (S \uplus \{c\}) \to (S \uplus \{c\})$

$$\begin{aligned} \text{inr}(c) \star x &= \text{inr}(c), \\ x \star \text{inr}(c) &= \text{inr}(c), \\ \text{inl}(s_1) \star \text{inl}(s_2) &= \text{inl}(s_1 \circ s_2). \end{aligned}$$

where $\star = \text{ann } c \circ$.

Direct product

- $\circ \in S \times S \to S$
- $\diamond \in T \times T \to T$
- $\circ \times \diamond \in (S \times T) \times (S \times T) \to (S \times T)$

$$(s_1, t_1) \bullet (s_2, t_2) = (s_1 \circ s_2, t_1 \diamond t_2).$$

where $\bullet = \circ \times \diamond$.

lexicographic product

- $\circ \in S \times S \to S$
- $\diamond \in T \times T \to T$
- $\circ \vec{\times} \diamond \in (S \times T) \times (S \times T) \to (S \times T)$

$$(s_1, t_1) \bullet (s_2, t_2) = \begin{cases} (s_1, t_1 \diamond t_2), & \text{if } s_1 = s_2 \\ (s_1, t_1), & \text{if } s_1 = (s_1 \circ s_2) \neq s_2 \\ (s_2, t_2), & \text{if } s_1 \neq (s_1 \circ s_2) = s_2 \\ (s_1 \circ s_2, 1_\diamond), & \text{if } s_1 \neq (s_1 \circ s_2) \neq s_2 \end{cases}$$

where $\bullet = \circ \vec{\times} \diamond$ and $1_\diamond \in T$ denotes an identity for T, if it exists,

Let's start with a small language fragement ...

$$
\begin{array}{rcl}
E & ::= & \texttt{bNatMinPlus} \\
 & | & \texttt{bNatMaxMin} \\
 & | & \texttt{bAddOne}\ c\ E \\
 & | & \texttt{bAddZero}\ c\ E \\
 & | & \texttt{bLex}\ E\ E \\
 & | & \texttt{bSelLex}\ E\ E
\end{array}
$$

where c represents constants supplied by the user.

untyped semantics

$$[\![E]\!] = (S,\ \oplus,\ \otimes),$$

$[\![E]\!] = (S,\ \oplus,\ \otimes)$

$$
\begin{aligned}
[\![\texttt{bNatMinPlus}]\!] &= (\mathbb{N},\ \min,\ +) \\
[\![\texttt{bNatMaxMin}]\!] &= (\mathbb{N},\ \max,\ \min) \\
[\![\texttt{bAddOne}\ c\ E]\!] &= (S \uplus \{c\},\ \mathrm{ann}\ c \oplus_S,\ \mathrm{id}\ c \otimes_S) \\
&\quad \text{where } [\![E]\!] = (S,\ \oplus_S,\ \otimes_S) \\
[\![\texttt{bAddZero}\ c\ E]\!] &= (S \uplus \{c\},\ \mathrm{id}\ c \oplus_S,\ \mathrm{ann}\ c \otimes_S) \\
&\quad \text{where } [\![E]\!] = (S,\ \oplus_S,\ \otimes_S) \\
[\![\texttt{bLex}\ E\ E']\!] &= (S \times T,\ \oplus_S \vec{\times} \oplus_T,\ \otimes_S \times \otimes_T) \\
&\quad \text{where } [\![E]\!] = (S,\ \oplus_S,\ \otimes_S) \\
&\quad\quad \text{and } [\![E']\!] = (T,\ \oplus_T,\ \otimes_T) \\
[\![\texttt{bSelLex}\ E\ E']\!] &= [\![\texttt{bLex}\ E\ E']\!]
\end{aligned}
$$

Typed Semantics

Either

$$[\![E]\!] = \mathrm{ERROR}$$

or

$$[\![E]\!] = ((S,\ \oplus,\ \otimes),\ \vec{\rho},\ \vec{\pi})$$

$\vec{\rho}$ proofs of required properties

$\vec{\pi}$ proofs or refutations of optional properties

Where to draw the line is a *design decision*!

For bisemigroups we only require $\oplus$ and $\otimes$ to be associative.

When does $\text{L.D}(S \vec{\times} T)$ hold?

For every combinator C and every property P

find $\text{wf}_{P,C}$ and $\beta_{P,C}$ such that

$$\text{wf}_{P,C}(\vec{a}) \Rightarrow (P(C(\vec{a})) \Leftrightarrow \beta_{P,C}(\vec{a}))$$

... which is then turned into two "bottom-up rules" ...

$$\begin{aligned} \text{wf}_{P,C}(\vec{a}) \wedge \beta_{P,C}(\vec{a}) &\Rightarrow P(C(\vec{a})) \\ \text{wf}_{P,C}(\vec{a}) \wedge \neg\beta_{P,C}(\vec{a}) &\Rightarrow \neg P(C(\vec{a})), \end{aligned}$$

When does $\mathrm{L.D}(S \vec{\times} T)$ hold?

... and finally, for each ¬P

introduce not.P that exposes the constructive content of $\neg F$.

description	P	**meaning**
¬Associativity	not.ass	$\exists x\ y\ z \in S, x \circ (y \circ z) \neq (x \circ y) \circ z$
¬Commutativity	not.com	$\exists x\ y \in S, x \circ y \neq y \circ x$
⋮	⋮	⋮
¬L distributivity	not.l.d	$\exists x\ y\ z \in S, z \otimes (x \oplus y) \neq (z \otimes x) \oplus (z \otimes y)$
⋮	⋮	⋮

When does $\text{L.D}(S \vec{\times} T)$ hold?

$$\text{wf}_{\text{l.dist},\vec{\times}} = \text{COM}(S, \oplus_S) \wedge \text{IDM}(S, \oplus_S) \wedge \text{IDE}(T, \oplus_T)$$

This is needed to guarantee associativity

When does $\text{L.D}(S \vec{\times} T)$ hold?

$$(\text{COM}(S, \oplus_S) \wedge \text{IDM}(S, \oplus_S) \wedge \text{IDE}(T, \oplus_T)) \Rightarrow$$

$$\begin{aligned} \text{L.D}(S \vec{\times} T) \iff \text{L.D}(S) \wedge \text{L.D}(T) \quad & \wedge \quad (\text{L.SS}(S) \vee \text{L.K}(T_\otimes)) \\ & \wedge \quad (\text{L.EC}(S) \vee \text{L.SMILE}(T)) \\ & \wedge \quad (\text{L.C}(S_\otimes) \vee \text{L.CON}(T)) \end{aligned}$$

This forces us to add these to $\mathcal{P}$

Property	Definition
L.C	$\forall xyz \in S, z \otimes y = z \otimes y \implies x = y$
L.EC	$\forall xyz \in S, z \otimes y = z \otimes y \implies (x \leq y) \vee (y \leq x)$
L.SS	$\forall xyz \in S, x < y \iff z \otimes x < z \otimes y$
L.K	$\forall xyz \in T, z \otimes x = z \otimes y$
L.SMILE	$\forall xyz \in T, (z \otimes x) \oplus (z \otimes y) = z \otimes 0_T$

This reflects design choices! Note that neither S nor T can be interesting semirings!

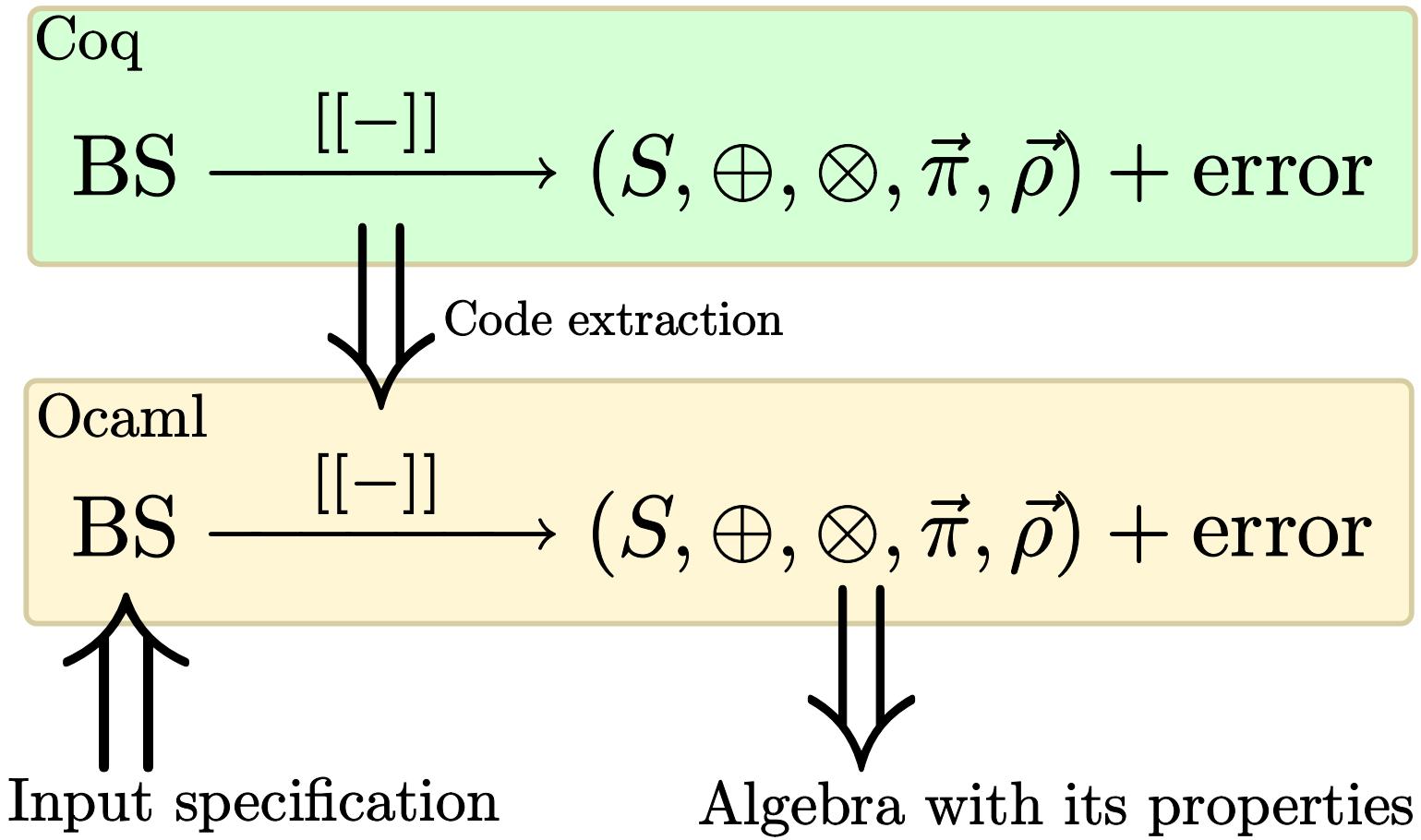
Current snapshot

name	signature	prefix	(positive) properties	constructors
Sets	(S)	d	3	9
Semigroups	$(S, \oplus)$	s	14	17
Preorders	$(S, \leq)$	p	4	5
Bisemigroups	$(S, \oplus, \otimes)$	b	22	20
Order semigroups	$(S, \leq, \oplus)$	o	17	6
Transforms	$(S, L, \triangleright)$	t	2	8
Order transforms	$(S, L, \leq, \triangleright)$	ot	3	2
Semigroup transforms	$(S, L, \oplus, \triangleright)$	st	4	10

where $\triangleright \in L \to S \to S$.

This represents over 1700 bottom-up rules ...

MrE Implementation using the Coq theorem prover



```
MrE> max_min <- bNatMaxMin
------- additive properties -------------
Commutativity       TRUE  : -
Idempotence         TRUE  : -
Selectivity         TRUE  : -
Identity            TRUE  : 0
Annihilator         FALSE : -
------- multiplicative properties --------
Commutativity       TRUE  : -
Idempotence         TRUE  : -
Selectivity         TRUE  : -
Identity            FALSE : -
Annihilator         TRUE  : 0
------- bisemigroup properties -----------
Consistency(+,*)    TRUE  : -
Consistency(*,+)    IRRELEVANT
L absorbing         TRUE  : -
L strict absorbing FALSE : 0, 0
L distributivity    TRUE  : -
```

```
MrE> min_plus <- bNatMinPlus
------- additive properties -------------
Commutativity       TRUE  : -
Idempotence         TRUE  : -
Selectivity         TRUE  : -
Identity            FALSE : -
Annihilator         TRUE  : 0
------- multiplicative properties --------
Commutativity       TRUE  : -
Idempotence         FALSE : 1
Selectivity         FALSE : 1, 1
Identity            TRUE  : 0
Annihilator         FALSE : -
------- bisemigroup properties -----------
Consistency(+,*)    IRRELEVANT
Consistency(*,+)    TRUE  : -
L absorbing         TRUE  : -
L strict absorbing FALSE : 0, 0
L distributivity    TRUE  : -
```

```
MrE> bw <- bAddOne INF max_min
------- additive properties -------------
Commutativity       TRUE  : -
Idempotence         TRUE  : -
Selectivity         TRUE  : -
Identity            TRUE  : inl 0
Annihilator         TRUE  : inr INF
------- multiplicative properties --------
Commutativity       TRUE  : -
Idempotence         TRUE  : -
Selectivity         TRUE  : -
Identity            TRUE  : inr INF
Annihilator         TRUE  : inl 0
------- bisemigroup properties -----------
Consistency(+,*)    TRUE  : -
Consistency(*,+)    TRUE  : -
L absorbing         TRUE  : -
L strict absorbing FALSE : inr INF, inr INF
L distributivity    TRUE  : -
```

```
MrE> sp <- bAddZero INF min_plus
------- additive properties -------------
Commutativity       TRUE  : -
Idempotence         TRUE  : -
Selectivity         TRUE  : -
Identity            TRUE  : inr INF
Annihilator         TRUE  : inl 0
------- multiplicative properties --------
Commutativity       TRUE  : -
Idempotence         FALSE : inl 1
Selectivity         FALSE : inl 1, inl 1
Identity            TRUE  : inl 0
Annihilator         TRUE  : inr INF
------- bisemigroup properties -----------
Consistency(+,*)    TRUE  : -
Consistency(*,+)    TRUE  : -
L absorbing         TRUE  : -
L strict absorbing FALSE : inr INF, inr INF
L distributivity    TRUE  : -
```

```
MrE> lex_sp_bw <- bLex sp bw

------- bisemigroup properties -----------
Consistency(+,*)    TRUE  : -
Consistency(*,+)    TRUE  : -
L absorbing         TRUE  : -
L strict absorbing FALSE :
                           (inr INF, inr INF),
                           (inr INF, inr INF)
L distributivity    FALSE :
                           (inl 0, inl 0),
                           (inr INF, inr INF),
                           (inr INF, inr INF)
R distributivity    FALSE :
                           (inl 0, inl 0),
                           (inr INF, inr INF),
                           (inr INF, inr INF)
```

```
MrE> lex_min_plus_max_min
                  <- bLex min_plus max_min

------- bisemigroup properties -----------
Consistency(+,*)    IRRELEVANT
Consistency(*,+)    IRRELEVANT
L absorbing         TRUE  : -
L strict absorbing FALSE : (0, 0) , (0, 0)
L distributivity    TRUE  : -
R distributivity    TRUE  : -
```

```
MrE> lex_sp_bw_v2 <- bAddOne NIL
       (bAddZero INF lex_min_plus_max_min)


------- bisemigroup properties -----------
Consistency(+,*)   TRUE  : -
Consistency(*,+)   TRUE  : -
L absorbing        TRUE  : -
L strict absorbing FALSE : inr NIL, inr NIL
L distributivity   TRUE  : -
R distributivity   TRUE  : -
```

Now let's try swtiching order ...

```
MrE> lex_max_min_min_plus <-
                    bLex max_min min_plus

  Error : min_plus does not have a
          multiplicative identity.
```

```
MrE> slex_max_min_min_plus <-
                  bSelLex max_min min_plus


------- bisemigroup properties -----------
Consistency(+,*)    IRRELEVANT
Consistency(*,+)    IRRELEVANT
L absorbing         TRUE  : -
L strict absorbing FALSE : (0, 0), (0, 0)
L distributivity   FALSE : (1, 1),
                           (0, 0),
                           (0, 1)
R distributivity   FALSE : (1, 1),
                           (0, 0),
                           (0, 1)
```

```
MrE> slex_bw_sp <- bAddOne NIL
          (bAddZero INF slex_max_min_min_plus)


------- bisemigroup properties -----------
Consistency(+,*)    TRUE  : -
Consistency(*,+)    TRUE  : -
L absorbing         TRUE  : -
L strict absorbing FALSE : inr NIL,
                           inr NIL
L distributivity    FALSE : inl inl (1, 1),
                           inl inl (0, 0),
                           inl inl (0, 1)
R distributivity    FALSE : inl inl (1, 1),
                           inl inl (0, 0),
                           inl inl (0, 1)
```

Minimal Sets

min-sets, or finite anti-chains (almost)

Suppose that $(S, \lesssim)$ is a pre-ordered set. Let $A \subseteq S$ be finite. Define

$$\begin{aligned} \min_{\lesssim}(A) &\equiv \{a \in A \mid \forall b \in A : \neg(b < a)\} \\ \mathcal{P}(S, \lesssim) &\equiv \{A \subseteq S \mid A \text{ is finite and } \min_{\lesssim}(A) = A\} \end{aligned}$$

min-set semigroup

$$\mathcal{P}^{\cup}_{\min}(S, \lesssim) = (\mathcal{P}(S, \lesssim), \cup^{\lesssim})$$

is the semigroup where

$$A \cup^{\lesssim} B = \min_{\lesssim}(A \cup B).$$

A few constructions ...

$$
\begin{aligned}
[\![\texttt{pRightNaturalOrder}\ E]\!] &= (S,\ \lesssim) \\
&\quad \text{where } [\![E]\!] = (S,\ \oplus) \\
&\quad \text{and } a \lesssim b \iff a \oplus b = b \\
[\![\texttt{sFMinSetsUnion}\ E]\!] &= (\mathcal{P}(S,\ \lesssim),\ \cup^{\lesssim}) \\
&\quad \text{where } [\![E]\!] = (S,\ \lesssim) \\
[\![\texttt{bFMinSets}\ E]\!] &= (\mathcal{P}(S,\ \lesssim),\ \cup^{\lesssim},\ \otimes^{\lesssim}_{\text{min}}) \\
&\quad \text{where } [\![E]\!] = (S,\ L,\ \otimes,\ \otimes) \\
\\
\text{define } A\ \otimes^{\lesssim}_{\text{min}}\ B &= \min_{\lesssim}(\{a \otimes b \mid a \in A,\ b \in B\})
\end{aligned}
$$

A bottleneck semiring[1]

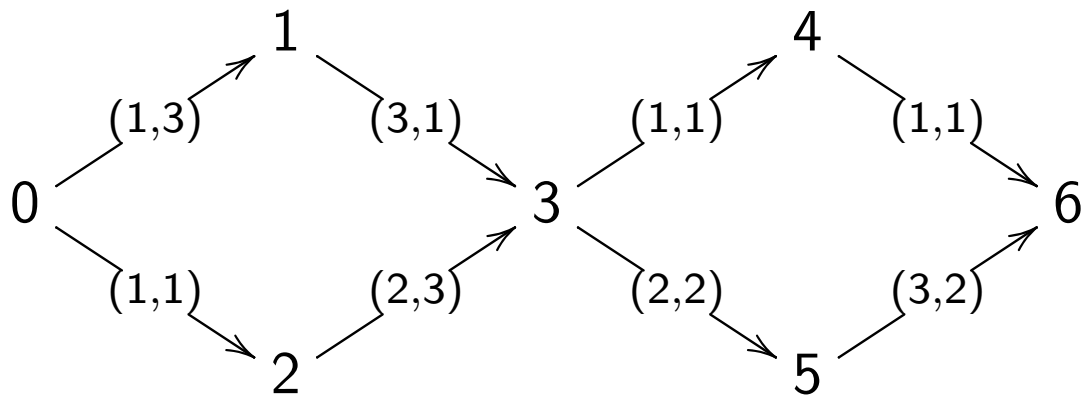
The idea ...

- arc weights from a partial order $\leq$
- "s is better than s'" means $s \leq s'$
- path weight $w(p)$ = set of worst edge weights in p.
- $w(p) \leq w(q) \iff \forall s \in w(p), \exists s' \in w(q), s \leq s'$

... in MrE

```
MrE> s1 <- sProduct sNatMin sNatMin
MrE> s2 <- sFMinSetsUnion (pRightNaturalOrder s1)
MrE> bottleneck <- bFMinSets (oRightNaturalOrder s2)
```

[1] Originally defined in **Bottleneck shortest paths on a partially ordered scale.**, Monnot, J. and Spanjaard, O., 4OR: A Quarterly Journal of Operations Research, 2003



$0-1$	$\{\{(1,3)\}\}$	$1-4$	$\{\{(3,1)\}\}$	$3-4$	$\{\{(1,1)\}\}$
$0-2$	$\{\{(1,1)\}\}$	$1-5$	$\{\{(3,1);(2,2)\}\}$	$3-5$	$\{\{(2,2)\}\}$
$0-3$	$\{\{(1,3),(3,1)\},\{(2,3)\}\}$	$1-6$	$\{\{(3,1)\}\}$	$3-6$	$\{\{(1,1)\}\}$
$0-4$	$\{\{(1,3),(3,1)\},\{(2,3)\}\}$	$2-3$	$\{\{(2,3)\}\}$	$4-6$	$\{\{(1,1)\}\}$
$0-5$	$\{\{(1,3),(3,1),(2,2)\},\{2,3\}\}$	$2-4$	$\{\{(2,3)\}\}$	$5-6$	$\{\{(3,2)\}\}$
$0-6$	$\{\{(1,3),(3,1)\},\{(2,3)\}\}$	$2-5$	$\{\{(2,3)\}\}$		
$1-3$	$\{\{(3,1)\}\}$	$2-6$	$\{\{(2,3)\}\}$		

All minimal cutsets semiring[2]

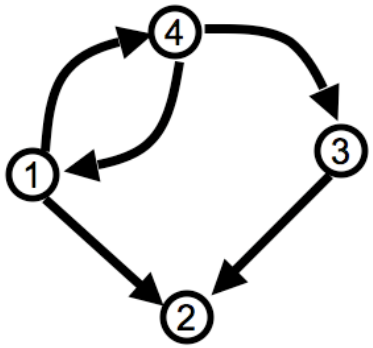
- A cut set $C \subseteq E$ for nodes i and j is a set of edges such there is no path from i to j in the graph $(V,\ E - C)$.
- C is minimal if no proper subset of C is a cut set.
- Martelli's semiring is such that $\mathbf{A}^{(*)}(i,\ j)$ is the set of all minimal cut sets for i and j.
- The arc $(i,\ j)$ is has weight $w(i,\ j) = \{\{(i,\ j)\}\}$.

```
MrE> s <- sFSetUnion (dProduct dNat dNat)
MrE> martelli <- bSwap (bFMinSets (oRightNaturalOrder s)
```

[2] Originally defined in **An application of regular algebra to the enumeration of cut sets in a graph**, Martelli, 1974

Martelli

$$(i,j) \in E \rightarrow w(i,j) = \{\{(i,j)\}\}$$

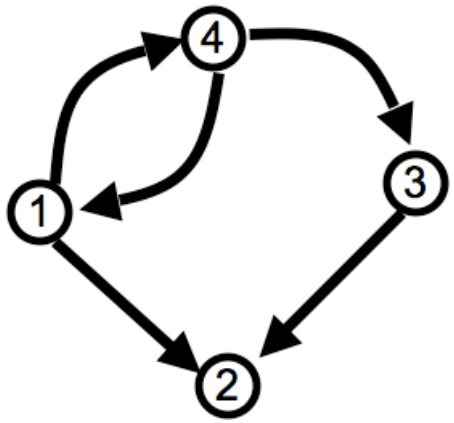


$$A = \begin{bmatrix} \{\phi\} & \{\{(1,2)\}\} & \{\phi\} & \{\{(1,4)\}\} \\ \{\phi\} & \{\phi\} & \{\phi\} & \{\phi\} \\ \{\phi\} & \{\{(3,2)\}\} & \{\phi\} & \{\phi\} \\ \{\{(4,1)\}\} & \{\phi\} & \{\{(4,3)\}\} & \{\phi\} \end{bmatrix}$$

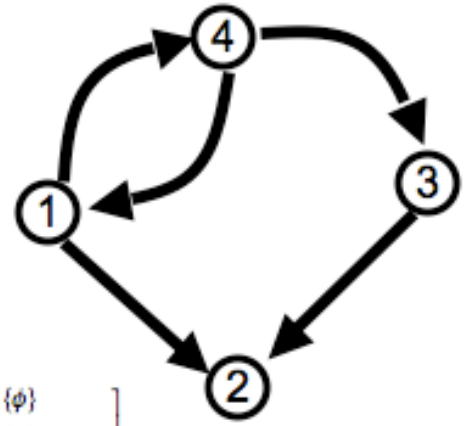
Martelli

$$A^2 = A \otimes A = \begin{bmatrix} \{\phi\} & \{\{(1,2)\}\} & \{\phi\} & \{\{(1,4)\}\} \\ \{\phi\} & \{\phi\} & \{\phi\} & \{\phi\} \\ \{\phi\} & \{\{(3,2)\}\} & \{\phi\} & \{\phi\} \\ \{\{(4,1)\}\} & \{\phi\} & \{\{(4,3)\}\} & \{\phi\} \end{bmatrix} \otimes \begin{bmatrix} \{\phi\} & \{\{(1,2)\}\} & \{\phi\} & \{\{(1,4)\}\} \\ \{\phi\} & \{\phi\} & \{\phi\} & \{\phi\} \\ \{\phi\} & \{\{(3,2)\}\} & \{\phi\} & \{\phi\} \\ \{\{(4,1)\}\} & \{\phi\} & \{\{(4,3)\}\} & \{\phi\} \end{bmatrix}$$

$$= \begin{bmatrix} \{\{(1,4)\},\{(4,1)\}\} & \{\phi\} & \{\{(1,4)\},\{(4,3)\}\} & \{\phi\} \\ \{\phi\} & \{\phi\} & \{\phi\} & \{\phi\} \\ \{\phi\} & \{\phi\} & \{\phi\} & \{\phi\} \\ \{\phi\} & \{\{(1,2),(3,2)\},\{(1,2),(4,3)\},\{(4,1),(3,2)\},\{(4,1),(4,3)\}\} & \{\phi\} & \{\{(1,4)\},\{(4,1)\}\} \end{bmatrix}$$



Martelli



$$A = \begin{bmatrix} \{\phi\} & \{\{(1,2)\}\} & \{\phi\} & \{\{(1,4)\}\} \\ \{\phi\} & \{\phi\} & \{\phi\} & \{\phi\} \\ \{\phi\} & \{\{(3,2)\}\} & \{\phi\} & \{\phi\} \\ \{\{(4,1)\}\} & \{\phi\} & \{\{(4,3)\}\} & \{\phi\} \end{bmatrix}$$

$$A^2 = \begin{bmatrix} \{\{(1,4)\},\{(4,1)\}\} & \{\phi\} & \{\{(1,4)\},\{(4,3)\}\} & \{\phi\} \\ \{\phi\} & \{\phi\} & \{\phi\} & \{\phi\} \\ \{\phi\} & \{\phi\} & \{\phi\} & \{\phi\} \\ \{\phi\} & \{\{(1,2),(3,2)\},\{(1,2),(4,3)\},\{(4,1),(3,2)\},\{(4,1),(4,3)\}\} & \{\phi\} & \{\{(1,4)\},\{(4,1)\}\} \end{bmatrix}$$

$$A^3 = A^5 = \begin{bmatrix} \{\phi\} & \{\{(1,4)\},\{(1,2),(3,2)\},\{(1,2),(4,3)\},\{(4,1),(3,2)\},\{(4,1),(4,3)\}\} & \{\phi\} & \{\{(1,4)\},\{(4,1)\}\} \\ \{\phi\} & \{\phi\} & \{\phi\} & \{\phi\} \\ \{\phi\} & \{\phi\} & \{\phi\} & \{\phi\} \\ \{\{(1,4)\},\{(4,1)\}\} & \{\phi\} & \{\{(4,1)\},\{(1,4)\},\{(4,3)\}\} & \{\phi\} \end{bmatrix}$$

$$A^4 = \begin{bmatrix} \{\{(1,4)\},\{(4,1)\}\} & \{\phi\} & \{\{(1,4)\},\{(1,4),(4,3)\}\} & \{\phi\} \\ \{\phi\} & \{\phi\} & \{\phi\} & \{\phi\} \\ \{\phi\} & \{\phi\} & \{\phi\} & \{\phi\} \\ \{\phi\} & \{\{(4,1)\},\{(1,4)\},\{(1,2),(3,2)\},\{(1,2),(4,3)\}\} & \{\phi\} & \{\{(1,4)\},\{(1,4)\}\} \end{bmatrix}$$

$$A^{(4)} = \begin{bmatrix} \phi & \{\{(1,2),(1,4)\},\{(1,2),(3,2)\},\{(1,2),(4,3)\}\} & \{\{(1,4)\},\{(4,3)\}\} & \{\{(1,4)\}\} \\ \{\phi\} & \phi & \{\phi\} & \{\phi\} \\ \{\phi\} & \{(3,2)\} & \phi & \{\phi\} \\ \{\{(4,1)\}\} & \{\{(1,2),(3,2)\},\{(1,2),(4,3)\},\{(4,1),(3,2)\},\{(4,1),(4,3)\}\} & \{\{(4,3)\}\} & \phi \end{bmatrix}$$

To-do list ...

- Add more constuctions (semi-direct products, ...)
- Add abstaction to the language
- Hook up to C compiler
- Explore Operations Research applications
- Move algorithmic proofs into Coq
- Use Coq's type classes (somehow)
- Attract users ....