# Kleene Modules for Routing Procedures

(work in progress)

Peter Höfner

Amsterdam

September 13, 2012

# Routing Procedures

- routing procedures
  - (shortest) path algorithms
  - routing protocols
    - networks (e.g. Wireless Mesh Networks)
    - internet

- examples
  - Dijkstra's Shortest Path
  - Floyd/Warshall
  - Border Gateway Protocol (BGP)
  - Ad-hoc On-demand Distance Vector (AODV) protocol

# Formal Methods for Mesh Networks

- aim
  - model, analyse, verify and increase the performance of wireless mesh protocols
  - develop suitable formal methods techniques

- benefits
  - more reliable protocols
  - finding and fixing bugs
  - better performance
  - proving correctness
  - reduce “time-to-market”

# Why Formal Methods

# Why Formal Methods

# Why (Kleene) Algebra

- suitable Operators

$$
\begin{array}{ccl}
+ & \longleftrightarrow & \text{choice} \\
\cdot & \longleftrightarrow & \text{composition} \\
{}^{*} & \longleftrightarrow & \text{iteration} \\
\leq & \longleftrightarrow & \text{natural order} \\
 & & (x \leq y \Leftrightarrow x + y = y)
\end{array}
$$

- successful in the past
  - shortest path algorithm [MaxPlus,Carre,Backhouse...]
  - Border Gateway protocol [SobrinhoGriffin,Roughan]

# Routing Algebra – Elements, Operators

- routing table entries (example)

$$(\mathtt{nhip}\,,\,\mathtt{hops})$$

- special symbols: $(0, \_)$ , $(\infty, \_)$

- choice: $(5, A) + (2, B) = (2, B)$
- multiplication: $(5, A) \cdot (2, B) = (7, A)$
  - destination and source must coincide

- both structures form monoid
- composition distributes over addition

# Combining Algebras

- direct product (HSP-theorem)
- natural order should coincide with lexicographical order

# Combining Algebras

- direct product (HSP-theorem)
- natural order should coincide with lexicographical order
- Theorem:
  assume a cancellative semiring $S$ and a semiring $T$
  then $(M, \oplus, \odot, (0,0), (1,1))$ is a semiring if
  - $M =_{df} (S - \{0\} \times T) \cup \{(0,0)\}$
  - addition is defined as

$$(a,x) \oplus (b,y) =_{df} \begin{cases} (a, x+y) & \text{if } a = b \\ (a,x) & \text{if } b < a \\ (b,y) & \text{if } a < b \\ (a+b, 0) & \text{otherwise} \end{cases}$$

  - multiplication is defined point-wise

# Combing Algebras

- Kleene star

$$(a,x)^* \;=_{df}\; \begin{cases} (1,1) & \text{if } a<1 \\ (1,x^*) & \text{if } a=1 \\ (a^*,0) & \text{otherwise} \end{cases}$$

- the maximal test set in the  lexicographical model consists only of the units
- domain (image) and codomain (range) is trivial,
  and so are the modal operators

# Matrices

- routing procedures usually use matrices over algebras
- matrices over routing table entries

$$
\begin{array}{cc}
 & \begin{matrix} A & \quad B & \quad C & \quad D & \quad \cdots \end{matrix} \\
\begin{matrix} A \\ B \\ C \\ D \\ \vdots \end{matrix} &
\begin{pmatrix}
(0, \_) & (1, B) & (2, B) & (\infty, \_) & \\
(1, A) & (0, \_) & (1, C) & (\infty, \_) & \cdots \\
(\infty, \_) & (1, B) & (0, \_) & (\infty, \_) & \\
(\infty, \_) & (\infty, \_) & (\infty, \_) & (0, \_) & \\
 & \vdots & & & \ddots
\end{pmatrix}
\end{array}
$$

routing table of $A$

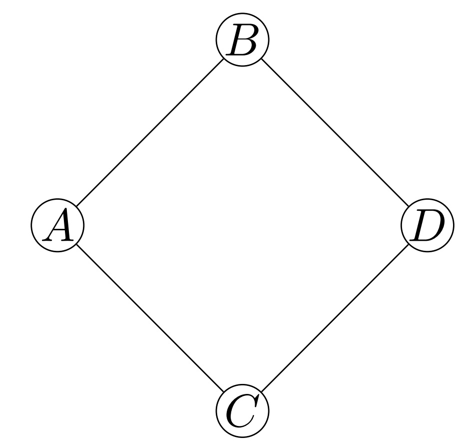
“routes” to $B$

- standard matrix operations

# Example



- a route request is broadcast




$$\underbrace{\begin{pmatrix} (0,\ \_) & (1,B) & (1,C) & (\infty,\_) \\ (1,A) & (0,\ \_) & (\infty,\_) & (D,1) \\ (1,A) & (\infty,\_) & (0,\ \_) & (1,D) \\ (\infty,\_) & (1,B) & (1,C) & (0,\ \_) \end{pmatrix}}_{\text{topology}} \bullet \underbrace{\begin{pmatrix} (0,\ \_) & (\infty,\_) & (\infty,\_) & (\infty,\_) \\ (\infty,\_) & (\infty,\_) & (\infty,\_) & (\infty,\_) \\ (\infty,\_) & (\infty,\_) & (\infty,\_) & (\infty,\_) \\ (\infty,\_) & (\infty,\_) & (\infty,\_) & (\infty,\_) \end{pmatrix}}_{\text{sender}} \bullet \underbrace{\begin{pmatrix} (0,\ \_) & (1,B) & (\infty,\_) & (\infty,\_) \\ (\mathbf{3},\mathbf{D}) & (0,\ \_) & (\infty,\_) & (\infty,\_) \\ (1,A) & (\infty,\_) & (0,\ \_) & (1,D) \\ (2,C) & (\infty,\_) & (1,C) & (0,\ \_) \end{pmatrix}}_{\text{routing table}}$$

$$= \underbrace{\begin{pmatrix} (0,\ \_) & (1,B) & (\infty,\_) & (\infty,\_) \\ (\mathbf{1},\mathbf{A}) & (0,\ \_) & (\infty,\_) & (\infty,\_) \\ (1,A) & (\infty,\_) & (0,\ \_) & (1,D) \\ (2,C) & (\infty,\_) & (1,C) & (0,\ \_) \end{pmatrix}}_{\text{updated routing table}}$$

# Example – Sending Messages

- sending messages

$$a + p \cdot b \cdot q \cdot (1 + c)$$

- by distributivity

$$a \;\; + \;\; p \cdot b \cdot q \;\; + \;\; p \cdot b \cdot q \cdot c$$

  snapshot, 1-hop connection learnt, content sent
- broadcast, unicast, groupcast are the same (modelled by different topologies)
- Kleene star models flooding the network (modal operators terminate flooding)

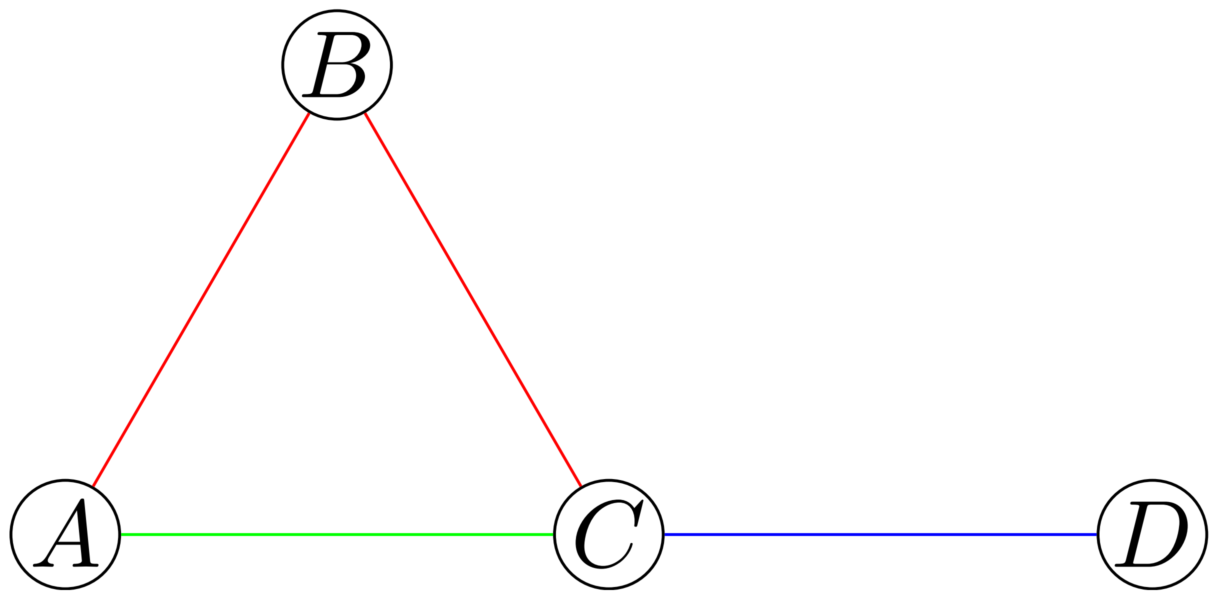
# Matrices

- matrices of semirings are semirings
- matrices of Kleene algebras are Kleene algebras
- ...

- both distributivity laws are essential!
- otherwise one looses associativity

# Lost Distributivity

- routing protocols use several components (a.o. timers)
- adding time stamps




$$
\begin{aligned}
r \cdot b &= (5, 2, B) \cdot (10, 1, D) = (\max(5, 10), 2 + 1, B \cdot D) = (10, 3, B) \\
g \cdot b &= (3, 1, C) \cdot (10, 1, D) = (\max(3, 10), 1 + 1, C \cdot D) = (10, 2, C)
\end{aligned}
$$

$$r \cdot b + g \cdot b \neq (r + g) \cdot b$$

# Kleene Modules

- Idea: use modules
  - Kleene algebra for the topologies
  - monoids for routing tables

# Kleene Modules

- Definition [Leiß06]:
  $(K, M, :)$ is a (left) semiring module if
  - $K$ is a Kleene algebra
  - $M$ is idempotent commutative monoid
  - $: K \times M \to M$ scalar product (Peirce product)

$$1 : a = a$$
$$t : 0 = 0$$
$$0 : a = a$$
$$(t + t') : a = (t : a) + (t' : a)$$
$$t : (a + a') = (t : a) + (t : a')$$
$$(t \cdot t') : a = t : (t' : a)$$

$$(t : a + b \leq b \Rightarrow t^* : a \leq b)$$

# Examples

- Kleene Algebras with partial multiplication
- reducts of Boolean modules (e.g. Peirce)
  - algebra of relations (using inverse image)
- context-free languages, linear languages
- head languages and head grammars (Pollard)
- routing procedures
  - Kleene algebra: matrices of pairs
  - monoid: routing tables with time stamps

$$(2, B) : (10, 3, A) \;=\; (10, 2+3, B \cdot A) \;=\; (10, 5, B)$$

(plus additional cases for 0)

# Tests and Points

- tests and points are used to restrict matrices (by use of multiplication)
- no multiplication on monoid (needed to select particular information)

- it is possible in the model if
  - if converse is available

$$\mathtt{sel}(i,j,a) \;=_{df}\; (j : (i : a)^{\smile})^{\smile}$$

  - selection over atoms (finitely generated)

$$\mathtt{diag}(a) \;=_{df}\; \sum_{i \in atoms} (i : (i : a)^{\smile})^{\smile}$$

# Ad Hoc On-Demand Distance Vector Protocol

- AODV control messages
  - route request (RREQ)
  - route reply (RREP)
  - route error message (RERR)

- Main Mechanism
  - if route is needed
    BROADCAST RREQ
  - if node has information about a destination
    UNICAST RREP
  - if unicast fails or link break is detected
    SEND RERR

# Example – Sending Messages

- Sending messages

$$
\begin{aligned}
& a + (i \cdot t \cdot j) : (\texttt{diag}(a) + c) \\
= \; & a + (i \cdot t \cdot j) : \texttt{diag}(a) + (i \cdot t \cdot j) : c
\end{aligned}
$$

- Broadcasting messages through entire network

$$a + (t' \cdot |t'^{*}\rangle j) : \texttt{diag}(a) + t'^{*} : \texttt{sel(i,j,a)}$$

where $t' \;=_{df}\; t \cdot |a^{\uparrow}]\neg i$

# On-Going Work Open Questions

- inverse of scalar product $^{\uparrow} : M \to K$

$$0^{\uparrow} = 0$$

$$(t : a)^{\uparrow} = t \cdot (a^{\uparrow})$$

$$(a + b)^{\uparrow} \neq a^{\uparrow} + b^{\uparrow}$$

- next hop operator

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & \dots \end{array} \\ \begin{pmatrix} (0, \_) & (1, B) & (2, B) & (\infty, \_) & \\ (1, A) & (0, \_) & (1, C) & (\infty, \_) & \dots \\ (\infty, \_) & (1, B) & (0, \_) & (\infty, \_) & \\ (\infty, \_) & (\infty, \_) & (\infty, \_) & (0, \_) & \\ & \vdots & & & \ddots \end{pmatrix} \end{array}$$

# Miscellaneous / Future Work

- implementation of the model to play with
- theorems at algebraic level proven automatically (Prover9/Sledgehammer of Isabelle/HOL)

- can everything be lifted to the algebraic level?
- important properties loop freedom, route correctness
- probably domain-theoretic (model) knowledge needed
- use Isabelle/HOL to switch between model and algebra

From **imagination** to **impact**

NICTA

From **imagination** to **impact**

# RFC 3561

- Request for Comments (de facto standard)

```
sequence number field is set to false.  The route is only updated if
the new sequence number is either

(i)       higher than the destination sequence number in the route
          table, or

(ii)      the sequence numbers are equal, but the hop count (of the
          new information) plus one, is smaller than the existing hop
          count in the routing table, or

(iii)     the sequence number is unknown.
```