Parity Automata for Quantitative Linear Time Logics

Corina Cirstea, Shunsuke Shimizu, and Ichiro Hasuo

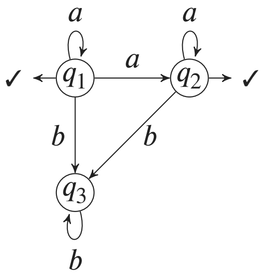
June 14, CALCO 2017

UNIVERSITY OF Southampton

NII 大学共同利用機関法人 情報・システム研究機構 国立情報学研究所 National Institute of Informatics

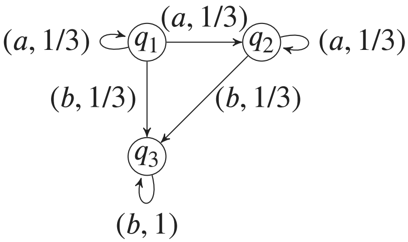
Introduction:
Capturing Linear Time Behaviour by Logic

Action + (Quantitative) Branching of Systems



Labeled Transition System

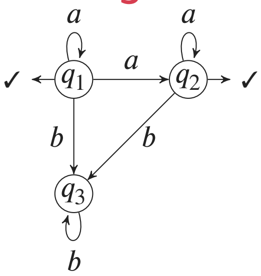
- Action:
  $X \to \{a, b\} \times X + \checkmark = FX$
- Branching:
  $FX \to \mathcal{P}FX$

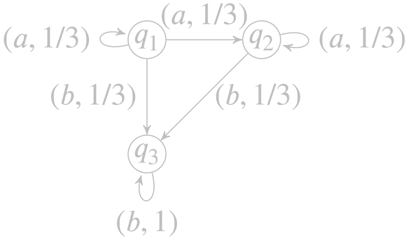


Markov Chain

- Action:
  $X \to \{a, b\} \times X = F'X$
- Branching:
  $F'X \to \mathcal{D}F'X$

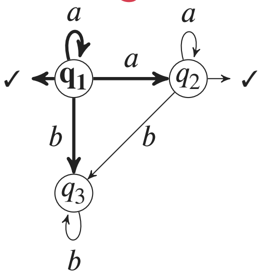
Branchings can be Flatten

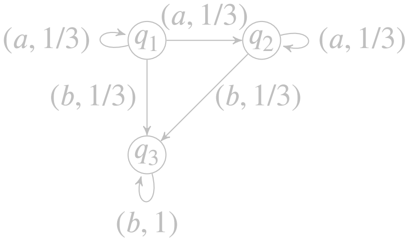




$q_1 \mapsto \{(a, q_1), (a, q_2), (b, q_3), \checkmark\}$

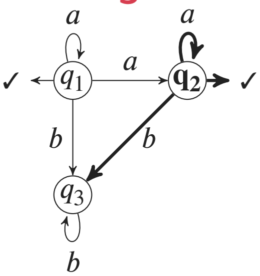
Branchings can be Flatten

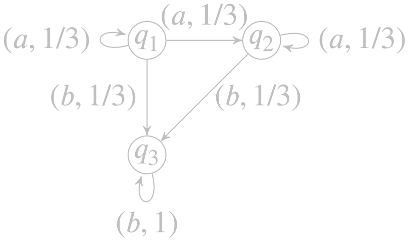




$$
\begin{aligned}
q_1 &\mapsto \{(\mathbf{a}, \mathbf{q_1}), (a, q_2), (b, q_3), \checkmark\} \\
&\mapsto \{(\mathbf{aa}, \mathbf{q_1}), (\mathbf{aa}, \mathbf{q_2}), (\mathbf{ab}, \mathbf{q_3}), \mathbf{a}\checkmark\} \cup \\
&\quad\ \{(aa, q_2), (ab, q_3), a\checkmark\} \cup \\
&\quad\ \{(bb, q_3)\} \cup \\
&\quad\ \{\checkmark\}
\end{aligned}
$$

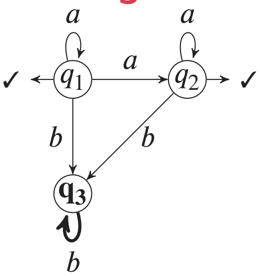
Branchings can be Flatten

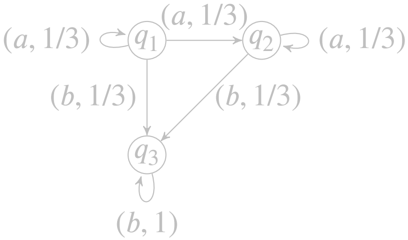




$$
\begin{aligned}
q_1 &\mapsto \{(a, q_1), (\mathbf{a}, \mathbf{q_2}), (b, q_3), \checkmark\} \\
&\mapsto \{(aa, q_1), (aa, q_2), (ab, q_3), a\checkmark\} \cup \\
&\quad\ \{(\mathbf{aa}, \mathbf{q_2}), (\mathbf{ab}, \mathbf{q_3}), \mathbf{a}\checkmark\} \cup \\
&\quad\ \{(bb, q_3)\} \cup \\
&\quad\ \{\checkmark\}
\end{aligned}
$$

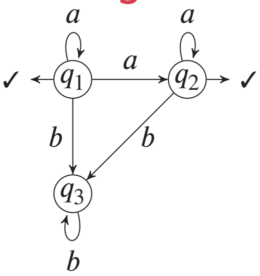
Branchings can be Flatten

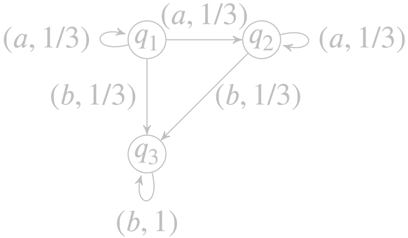




$$
\begin{aligned}
q_1 &\mapsto \{(a, q_1), (a, q_2), (\mathbf{b}, \mathbf{q_3}), ✓\} \\
&\mapsto \{(aa, q_1), (aa, q_2), (ab, q_3), a✓\} \cup \\
&\quad\ \{(aa, q_2), (ab, q_3), a✓\} \cup \\
&\quad\ \{(\mathbf{bb}, \mathbf{q_3})\} \cup \\
&\quad\ \{✓\}
\end{aligned}
$$

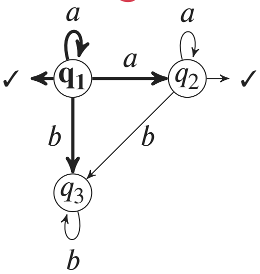
Branchings can be Flatten

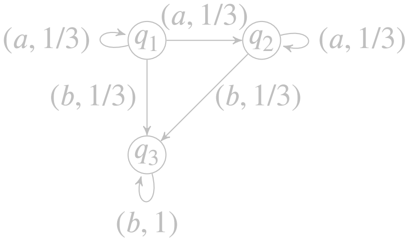




$$q_1 \mapsto \{(a, q_1), (aa, q_2), (ab, q_3), (bb, q_3), \checkmark, a\checkmark\}$$

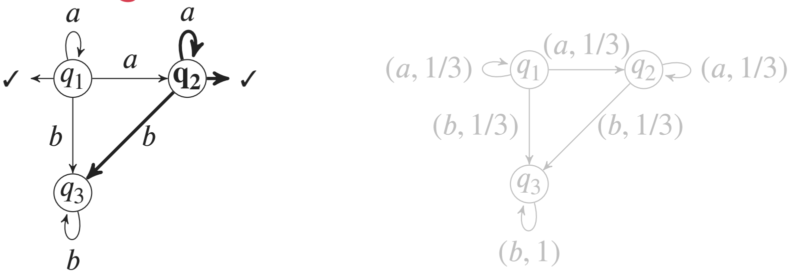
Branchings can be Flatten





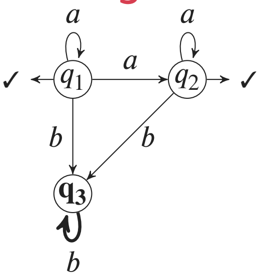
$$
\begin{aligned}
q_1 &\mapsto \{(\mathbf{a}, \mathbf{q_1}), (aa, q_2), (ab, q_3), (bb, q_3), \checkmark, a\checkmark\} \\
&\mapsto \{(\mathbf{aaa}, \mathbf{q_1}), (\mathbf{aaa}, \mathbf{q_2}), (\mathbf{aab}, \mathbf{q_3}), \mathbf{aa}\checkmark\} \cup \\
& \{(aaa, q_2), (aab, q_3), aa\checkmark\} \cup \\
& \{(abb, q_3), (bbb, q_3)\} \cup \\
& \{\checkmark, a\checkmark\}
\end{aligned}
$$

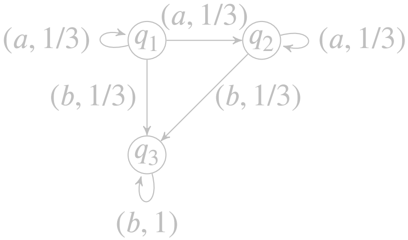
Branchings can be Flatten



$$
\begin{aligned}
q_1 &\mapsto \{(a, q_1), (\mathbf{aa}, \mathbf{q_2}), (ab, q_3), (bb, q_3), \checkmark, a\checkmark\} \\
&\mapsto \{(aaa, q_1), (aaa, q_2), (aab, q_3), aa\checkmark\} \cup \\
&\quad\ \{(\mathbf{aaa}, \mathbf{q_2}), (\mathbf{aab}, \mathbf{q_3}), \mathbf{aa}\checkmark\} \cup \\
&\quad\ \{(abb, q_3), (bbb, q_3)\} \cup \\
&\quad\ \{\checkmark, a\checkmark\}
\end{aligned}
$$

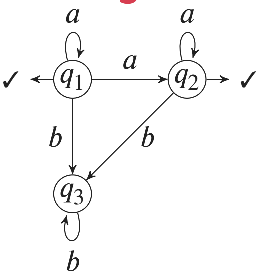
Branchings can be Flatten

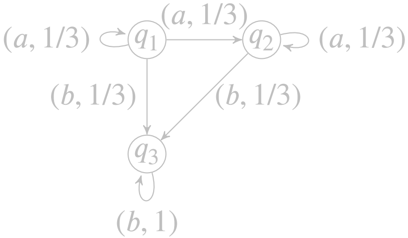




$$
\begin{aligned}
q_1 &\mapsto \{(a, q_1), (aa, q_2), (\mathbf{ab}, \mathbf{q_3}), (\mathbf{bb}, \mathbf{q_3}), \checkmark, a\checkmark\} \\
&\mapsto \{(aaa, q_1), (aaa, q_2), (aab, q_3), aa\checkmark\} \cup \\
&\quad\ \{(aaa, q_2), (aab, q_3), aa\checkmark\} \cup \\
&\quad\ \{(\mathbf{abb}, \mathbf{q_3}), (\mathbf{bbb}, \mathbf{q_3})\} \cup \\
&\quad\ \{\checkmark, a\checkmark\}
\end{aligned}
$$

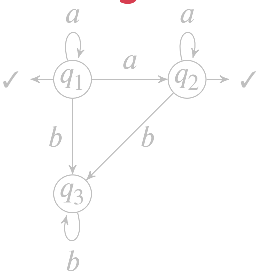
Branchings can be Flatten

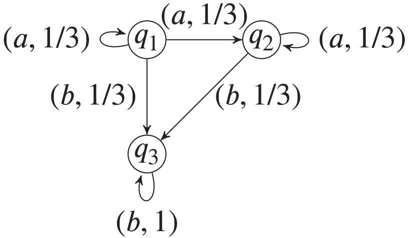




$$q_1 \mapsto \{(aaa, q_1), (aaa, q_2), (aab, q_3), (abb, q_3), (bbb, q_3), \checkmark, a\checkmark, aa\checkmark\}$$

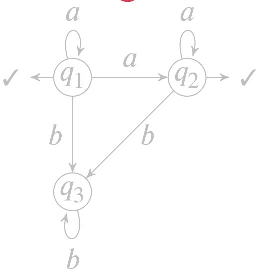
Branchings can be Flatten

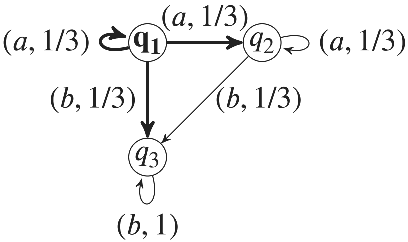




$$q_1 \mapsto \frac{1}{3}(a, q_1) + \frac{1}{3}(a, q_2) + \frac{1}{3}(b, q_3)$$

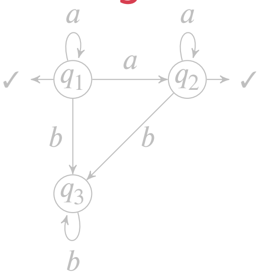
Branchings can be Flatten

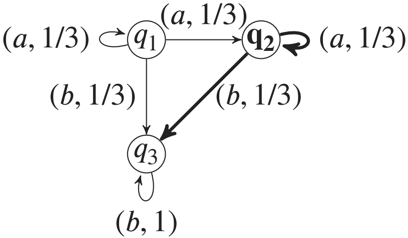




$$
\begin{aligned}
q_1 &\mapsto \mathbf{\frac{1}{3}}(\mathbf{a}, \mathbf{q_1}) + \frac{1}{3}(a, q_2) + \frac{1}{3}(b, q_3) \\
&\mapsto \mathbf{\frac{1}{9}}(\mathbf{aa}, \mathbf{q_1}) + \mathbf{\frac{1}{9}}(\mathbf{aa}, \mathbf{q_2}) + \mathbf{\frac{1}{9}}(\mathbf{ab}, \mathbf{q_3}) + \\
&\quad \frac{1}{9}(aa, q_2) + \frac{1}{9}(ab, q_3) + \\
&\quad \frac{1}{3}(bb, q_3)
\end{aligned}
$$

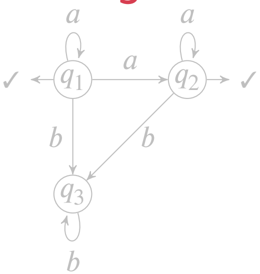
Branchings can be Flatten

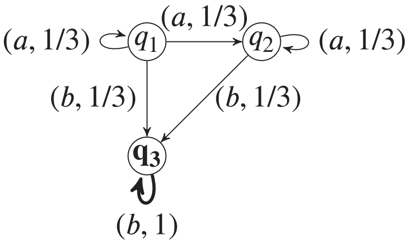




$$
\begin{aligned}
q_1 &\mapsto \frac{1}{3}(a, q_1) + \mathbf{\frac{1}{3}(a, q_2)} + \frac{1}{3}(b, q_3) \\
&\mapsto \frac{1}{9}(aa, q_1) + \frac{1}{9}(aa, q_2) + \frac{1}{9}(ab, q_3) + \\
&\quad \mathbf{\frac{1}{9}(aa, q_2)} + \mathbf{\frac{1}{9}(ab, q_3)} + \\
&\quad \frac{1}{3}(bb, q_3)
\end{aligned}
$$

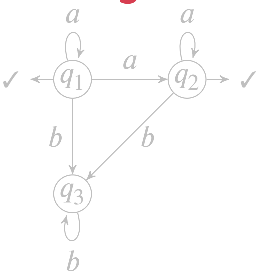
Branchings can be Flatten

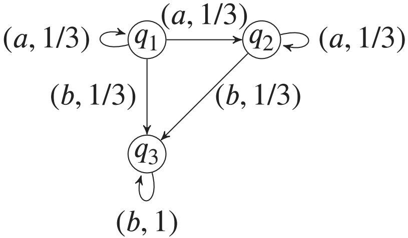




$$\begin{aligned} q_1 &\mapsto \frac{1}{3}(a, q_1) + \frac{1}{3}(a, q_2) + \boldsymbol{\frac{1}{3}(\mathbf{b}, \mathbf{q_3})} \\ &\mapsto \frac{1}{9}(aa, q_1) + \frac{1}{9}(aa, q_2) + \frac{1}{9}(ab, q_3) + \\ &\quad \frac{1}{9}(aa, q_2) + \frac{1}{9}(ab, q_3) + \\ &\quad \boldsymbol{\frac{1}{3}(\mathbf{bb}, \mathbf{q_3})} \end{aligned}$$

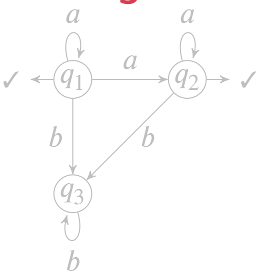
Branchings can be Flatten

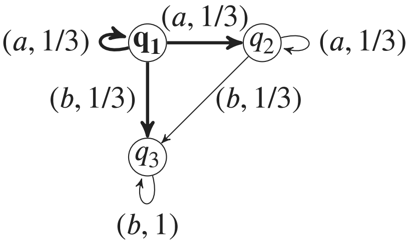




$$q_1 \mapsto \frac{1}{9}(aa, q_1) + \frac{2}{9}(aa, q_2) + \frac{2}{9}(ab, q_3) + \frac{1}{3}(bb, q_3)$$

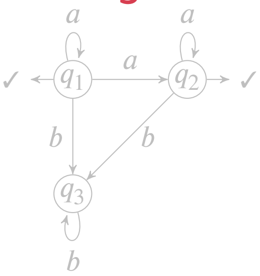
Branchings can be Flatten

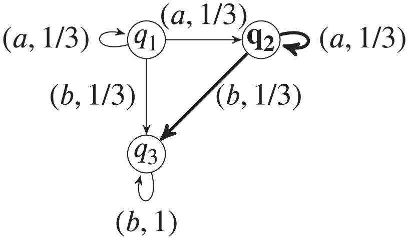




$$
\begin{aligned}
q_1 &\mapsto \mathbf{\frac{1}{9}}(\mathbf{aa}, \mathbf{q_1}) + \frac{2}{9}(aa, q_2) + \frac{2}{9}(ab, q_3) + \frac{1}{3}(bb, q_3) \\
&\mapsto \mathbf{\frac{1}{27}}(\mathbf{aaa}, \mathbf{q_1}) + \mathbf{\frac{1}{27}}(\mathbf{aaa}, \mathbf{q_2}) + \mathbf{\frac{1}{27}}(\mathbf{aab}, \mathbf{q_3}) + \\
&\quad \frac{2}{27}(aaa, q_2) + \frac{2}{27}(aab, q_3) + \\
&\quad \frac{2}{9}(abb, q_3) + \frac{1}{3}(bbb, q_3)
\end{aligned}
$$

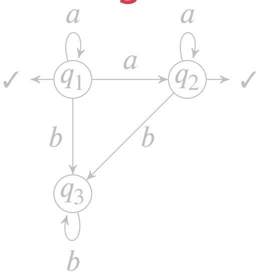
Branchings can be Flatten

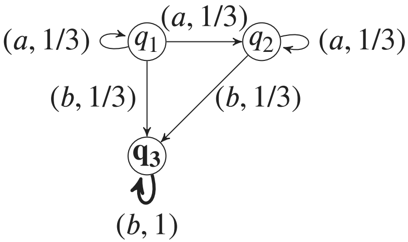




$$
\begin{aligned}
q_1 &\mapsto \frac{1}{9}(aa, q_1) + \boldsymbol{\frac{2}{9}(\mathbf{aa}, \mathbf{q_2})} + \frac{2}{9}(ab, q_3) + \frac{1}{3}(bb, q_3) \\
&\mapsto \frac{1}{27}(aaa, q_1) + \frac{1}{27}(aaa, q_2) + \frac{1}{27}(aab, q_3) + \\
&\quad \boldsymbol{\frac{2}{27}(\mathbf{aaa}, \mathbf{q_2})} + \boldsymbol{\frac{2}{27}(\mathbf{aab}, \mathbf{q_3})} + \\
&\quad \frac{2}{9}(abb, q_3) + \frac{1}{3}(bbb, q_3)
\end{aligned}
$$

Branchings can be Flatten

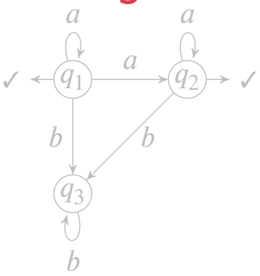


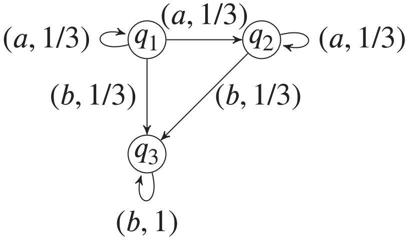


$$q_1 \mapsto \frac{1}{9}(aa, q_1) + \frac{2}{9}(aa, q_2) + \mathbf{\frac{2}{9}(ab, q_3)} + \mathbf{\frac{1}{3}(bb, q_3)}$$

$$\mapsto \frac{1}{27}(aaa, q_1) + \frac{1}{27}(aaa, q_2) + \frac{1}{27}(aab, q_3) + \frac{2}{27}(aaa, q_2) + \frac{2}{27}(aab, q_3) + \mathbf{\frac{2}{9}(abb, q_3)} + \mathbf{\frac{1}{3}(bbb, q_3)}$$

Branchings can be Flatten





$$q_1 \mapsto \frac{1}{27}(aaa, q_1) + \frac{1}{9}(aaa, q_2) + \frac{1}{9}(aab, q_3) + \frac{2}{9}(abb, q_3) + \frac{1}{3}(bbb, q_3)$$

Representing Branching by Semirings

A **partial commutative semiring**: $(S, +, 0, \bullet, 1)$, "+" can be partial.

- Boolean semiring: $(\{0, 1\}, \vee, 0, \wedge, 1)$ nondet. branching
- Prob. partial semiring: $([0, 1], +, 0, \times, 1)$ prob. branching
- Tropical semiring: $(\mathbb{N}^{\infty}, \min, \infty, +, 0)$ weighted branching

Representing Actions by Ranked Alphabets

A **ranked alphabet** consists of:

- an alphabet Λ,
- arities $\mathrm{ar} : \Lambda \to \mathbb{N}$.

It defines the **polynomial functor** F (which is the "action") by:

$$FX = \coprod_{\lambda \in \Lambda} X^{\mathrm{ar}(\lambda)}$$

Examples

- determ. LTS: $\{a \mapsto 1, b \mapsto 1\}$
- determ. LTS with explicit termination: $\{a \mapsto 1, b \mapsto 1, \checkmark \mapsto 0\}$

Systems are Coalgebras

A partial commutative semiring S yields a **commutative monad** T_S:

- $\mathsf{T}_S X = \{p\colon X \to S \mid$ p has finite supp., $\sum_{x \in \mathrm{supp}(p)} p(x)$ is defined$\}$
- unit: $\eta_X(x)(x') = \begin{cases} 1 & \text{if } x = x' \\ 0 & \text{otherwise} \end{cases}$
- multiplication: $\mu_X(\Psi)(x') = \sum_{p \in \mathrm{supp}(\Psi)} \Psi(p) \bullet p(x')$
- double strength: $\mathrm{dst}_{X,Y}(p,q)(x,y) = p(x) \bullet q(y)$

A branching system $\mathcal{M}$ is a $\mathsf{T}_S F$-coalgebra $(M,\ \delta\colon\ M \to \mathsf{T}_S F M)$.

Quantitative Linear Time Logic [Cîrstea '14,'15]

- The (coalgebraic) logic to capture "linear time" behaviour.
- Equipped with $\sum$, where the semiring S is from model's branching, instead of propositional operators.

$$\varphi ::= x \mid [\lambda](\varphi_1, \ldots, \varphi_{\mathrm{ar}(\lambda)}) \mid \sum_{i \in \{1,2,\ldots,n\}} s_i \bullet \varphi_i \mid \mu x.\, \varphi \mid \nu x.\, \varphi$$

Example:

- For nondet. branching, the $\sum$ is disjunctions $\bigvee$.
- For prob. branching, the $\sum$ is sub-convex combinations.
- In general, the $\sum$ is weighted sum.

Its **quantitative** semantics, relying on predicate liftings, is mostly standard.

Quantitative Linear Time Logic (Cont.)

- The (coalgebraic) logic to capture "linear time" behaviour.
- Equipped with $\sum$, where the semiring S is from model's branching, instead of propositional operators.

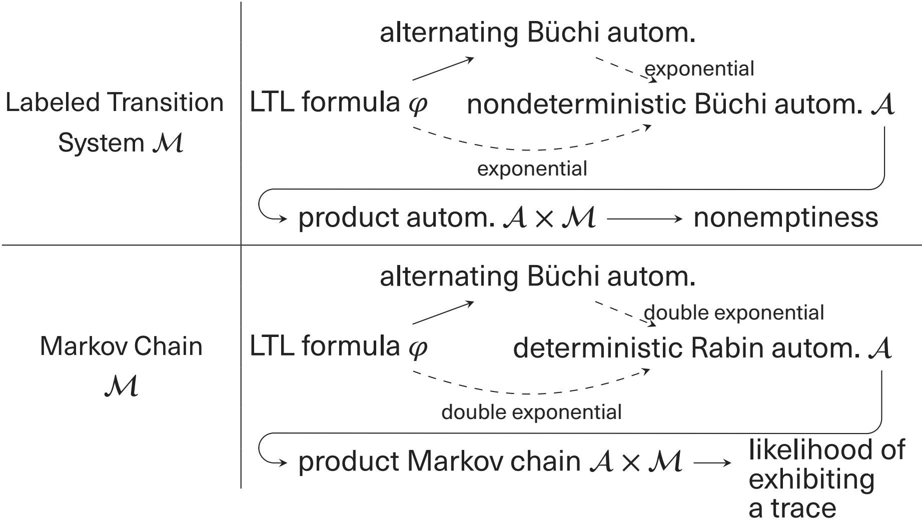
$$\varphi ::= x \mid [\lambda](\varphi_1, \ldots, \varphi_{\mathrm{ar}(\lambda)}) \mid \sum_{i \in \{1,2,\ldots,n\}} s_i \bullet \varphi_i \mid \mu x. \varphi \mid \nu x. \varphi$$

- monad multiplication used to accumulate values from different branches.

Step-based and **path-based** semantics coincide with this logic.

- **Step-based semantics**: $\mathrm{T}_{\mathcal{S}} \circ F\text{-CoAlg} \xrightarrow{[\![\varphi]\!]} S^M$.
  Standard in coalgebraic logic. Directly verified.
- **Path-based semantics**: $\mathrm{T}_{\mathcal{S}} \circ F\text{-CoAlg} \to \mathrm{T}_{\mathcal{S}} Z \xrightarrow{\mathrm{T}_{\mathcal{S}}[\![\varphi]\!]} \mathrm{T}_{\mathcal{S}} S^M \xrightarrow{\mu} S^M$.
  Seen in (existential) Linear Temporal Logic (LTL).
  Indirectly verified.

LTL Verification Techniques



Our Work:
Connection of These Logics and Automata

What We Want?

- notion of quantitative automaton (coalgebra + acceptance condition),
- translation from a formula φ to an automaton $\mathcal{A}_\varphi$,
- product $\otimes$ between formula automaton $\mathcal{A}_\varphi$ and model coalgebra $\mathcal{M}$,
- notion of **extent** $\mathrm{ext}(\mathcal{A})$ for an automaton $\mathcal{A}$,

such that for any given φ and $\mathcal{M}$

$$[\![\varphi]\!]_{\mathcal{M}} = \mathrm{ext}(\mathcal{A}_\varphi \otimes \mathcal{M})\,.$$

Piece 1: Product Coalgebras

Given two $\mathsf{T}_{\mathcal{S}} \circ F$-coalgebras $\mathcal{M} = (M, \gamma)$ and $\mathcal{N} = (N, \delta)$, the **product** $\mathsf{T}_{\mathcal{S}} \circ F$-coalgebra $\mathcal{M} \times \mathcal{N} = (M \times N, \gamma \otimes \delta)$ is defined by:

$$\gamma \otimes \delta = C \times D \xrightarrow{\gamma \times \delta} \mathsf{T}_{\mathcal{S}} FC \times \mathsf{T}_{\mathcal{S}} FD \xrightarrow{\mathrm{dst}_{FC,FD}} \mathsf{T}_{\mathcal{S}}(FC \times FD) \xrightarrow{\langle F\pi_1, F\pi_2 \rangle^*} \mathsf{T}_{\mathcal{S}} F(C \times D),$$

where $\langle F\pi_1, F\pi_2 \rangle^*$ denotes precomposition with $\langle F\pi_1, F\pi_2 \rangle$

$$F(C \times D) \xrightarrow{\langle F\pi_1, F\pi_2 \rangle} FC \times FD \to S.$$

Piece 2: Extents

Given a $\mathsf{T}_S \circ F$-coalgebra $\mathcal{M} = (M, \gamma)$,
the **(ν-extent / μ-extent)** is an S-valued predicate over M that is the (greatest / least) fixed point of the below operator.

$$M \xrightarrow{p} S \quad \overset{\mathbf{opr}}{\longmapsto} \quad \begin{array}{ccc} \mathsf{T}_S F M & \xrightarrow{\mathsf{T}_S F p} & \mathsf{T}_S F S \\ \gamma \uparrow & & \downarrow \mathsf{T}_S(\bullet_F) \\ & & \mathsf{T}_S S = \mathsf{T}_S^2 1 \\ & & \downarrow \mu_1 \\ M & & \mathsf{T}_S 1 = S \end{array}$$

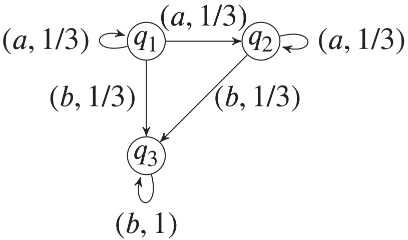
where $\bullet_F \colon FS \to S$ is given by

$$\bullet_F(\lambda(s_1, \dots, s_{\mathrm{ar}(\lambda)})) = s_1 \bullet \cdots \bullet s_{\mathrm{ar}(\lambda)} \quad \text{for } \lambda \in \Lambda$$

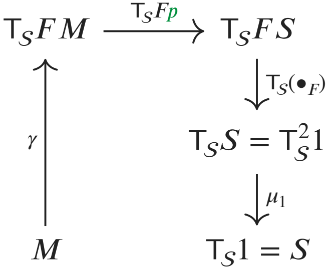
Intuitively ν-extent is:

- exixtence of a possibly infinite trace,
- likelihood of exhibitting a possibly infinite trace.

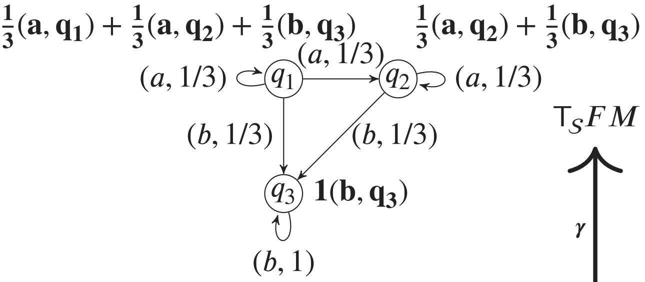
Calculating ν-Extent

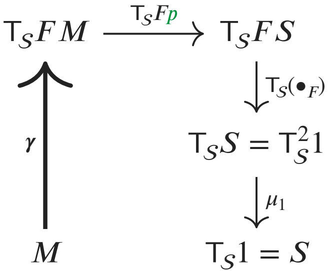


$$p = \begin{bmatrix} q_1 \mapsto 1 \\ q_2 \mapsto 1 \\ q_3 \mapsto 1 \end{bmatrix}$$



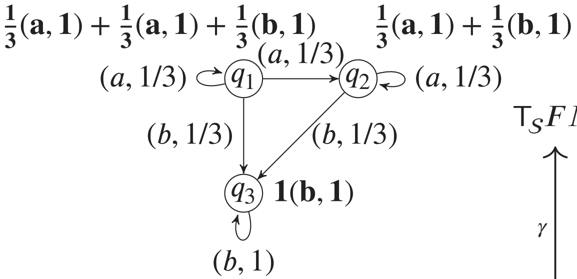
Calculating ν-Extent



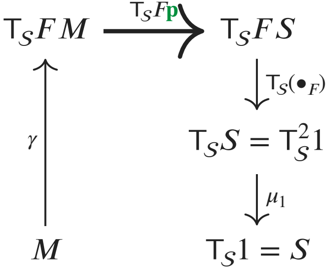


$$p = \begin{bmatrix} q_1 \mapsto 1 \\ q_2 \mapsto 1 \\ q_3 \mapsto 1 \end{bmatrix}$$

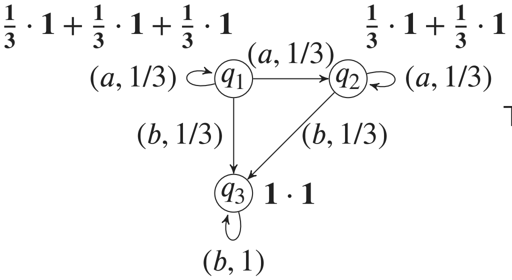
Calculating ν-Extent



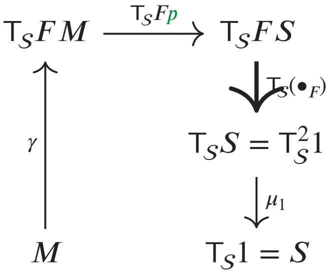
$$p = \begin{bmatrix} q_1 \mapsto 1 \\ q_2 \mapsto 1 \\ q_3 \mapsto 1 \end{bmatrix}$$



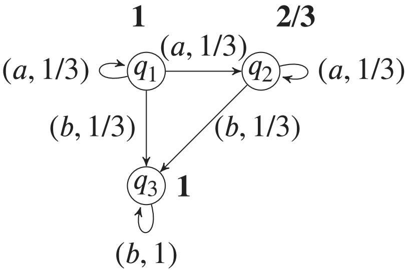
Calculating ν-Extent



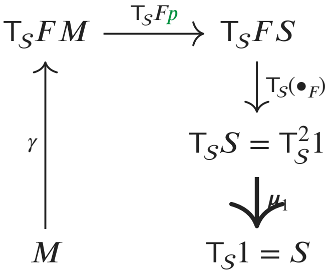
$$p = \begin{bmatrix} q_1 \mapsto 1 \\ q_2 \mapsto 1 \\ q_3 \mapsto 1 \end{bmatrix}$$



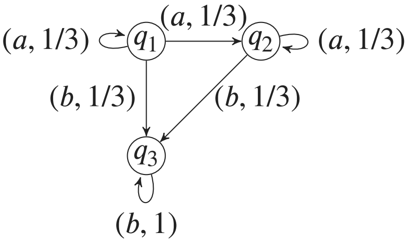
Calculating ν-Extent



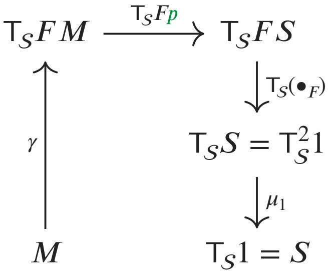
$$p = \begin{bmatrix} q_1 \mapsto 1 \\ q_2 \mapsto 1 \\ q_3 \mapsto 1 \end{bmatrix}$$



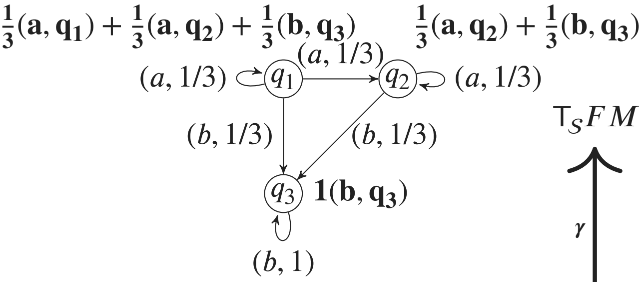
Calculating ν-Extent

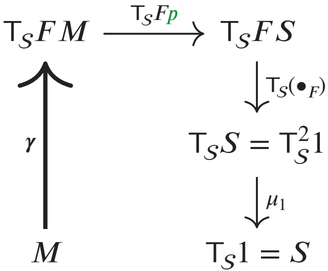


$$p = \begin{bmatrix} q_1 \mapsto 1 \\ q_2 \mapsto 2/3 \\ q_3 \mapsto 1 \end{bmatrix}$$



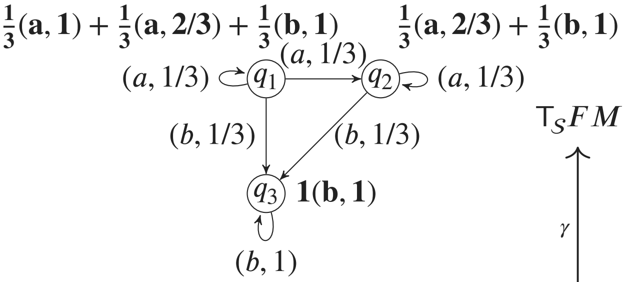
Calculating ν-Extent



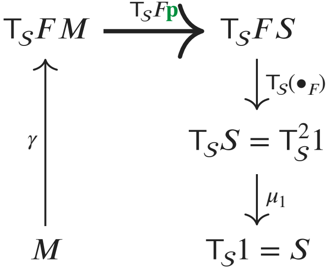


$$p = \begin{bmatrix} q_1 \mapsto 1 \\ q_2 \mapsto 2/3 \\ q_3 \mapsto 1 \end{bmatrix}$$

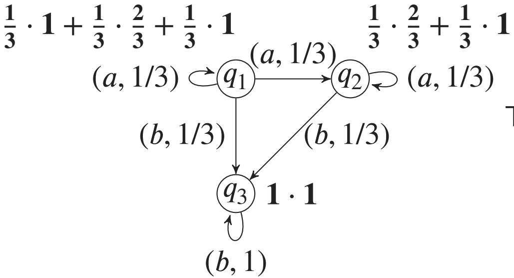
Calculating ν-Extent



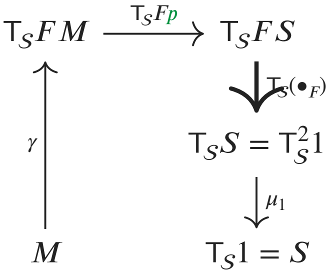
$$p = \begin{bmatrix} q_1 \mapsto 1 \\ q_2 \mapsto 2/3 \\ q_3 \mapsto 1 \end{bmatrix}$$



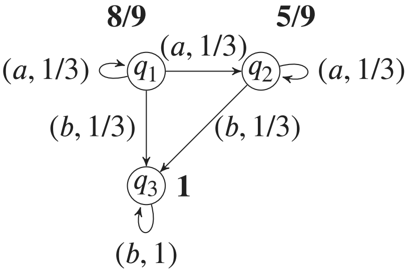
Calculating ν-Extent



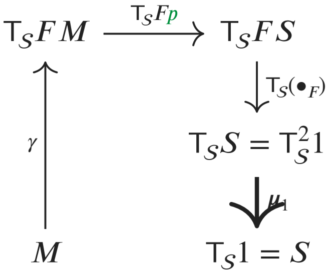
$$p = \begin{bmatrix} q_1 \mapsto 1 \\ q_2 \mapsto 2/3 \\ q_3 \mapsto 1 \end{bmatrix}$$



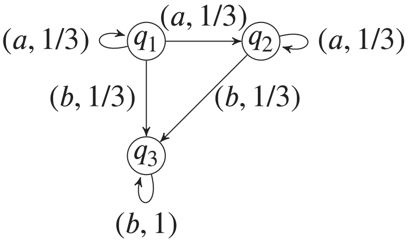
Calculating ν-Extent



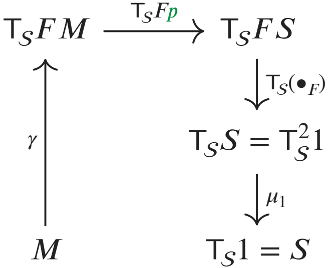
$$p = \begin{bmatrix} q_1 \mapsto 1 \\ q_2 \mapsto 2/3 \\ q_3 \mapsto 1 \end{bmatrix}$$



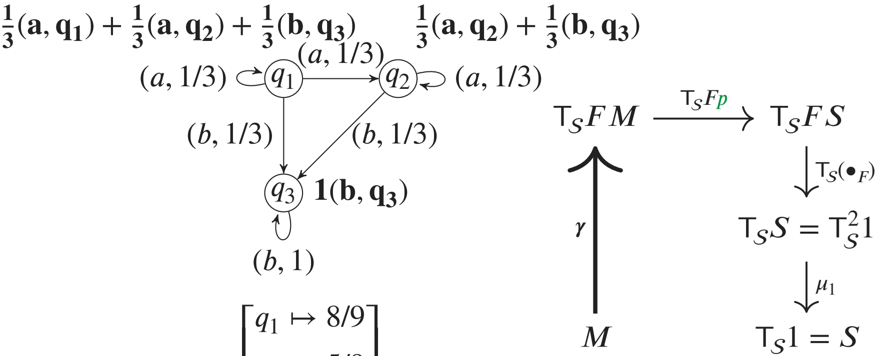
Calculating ν-Extent



$$p = \begin{bmatrix} q_1 \mapsto 8/9 \\ q_2 \mapsto 5/9 \\ q_3 \mapsto 1 \end{bmatrix}$$

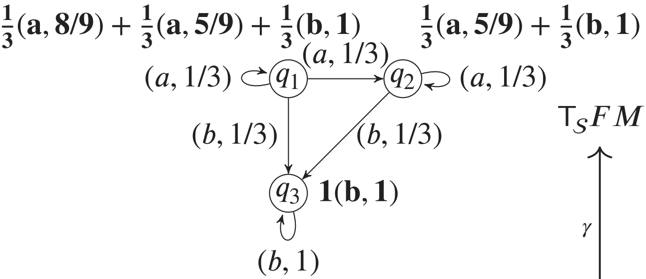


Calculating ν-Extent

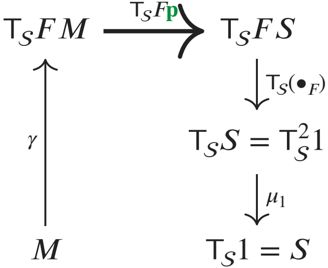


$$p = \begin{bmatrix} q_1 \mapsto 8/9 \\ q_2 \mapsto 5/9 \\ q_3 \mapsto 1 \end{bmatrix}$$

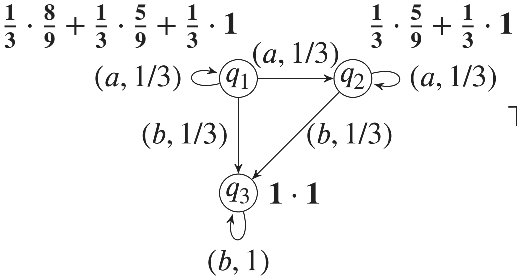
Calculating ν-Extent



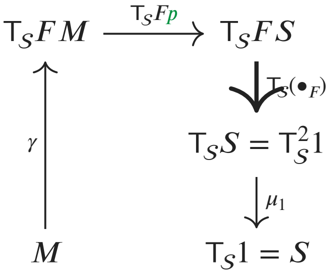
$$p = \begin{bmatrix} q_1 \mapsto 8/9 \\ q_2 \mapsto 5/9 \\ q_3 \mapsto 1 \end{bmatrix}$$



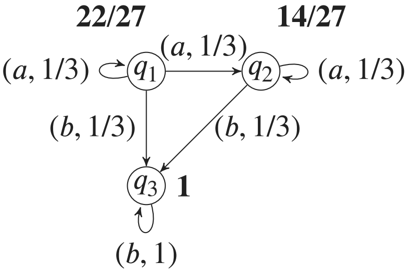
Calculating ν-Extent



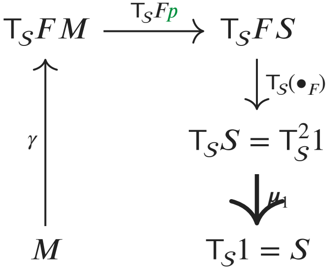
$$\textcolor{green}{p} = \begin{bmatrix} q_1 \mapsto 8/9 \\ q_2 \mapsto 5/9 \\ q_3 \mapsto 1 \end{bmatrix}$$



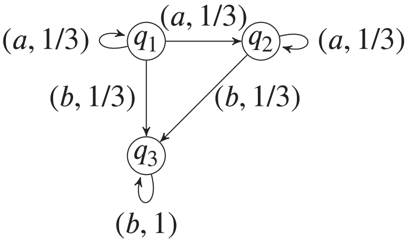
Calculating ν-Extent



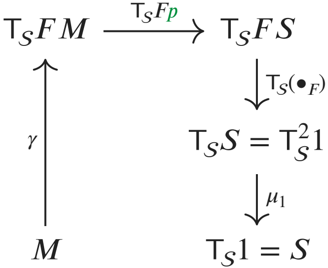
$$p = \begin{bmatrix} q_1 \mapsto 8/9 \\ q_2 \mapsto 5/9 \\ q_3 \mapsto 1 \end{bmatrix}$$



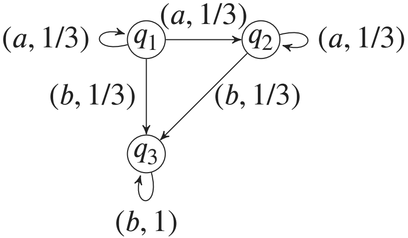
Calculating ν-Extent

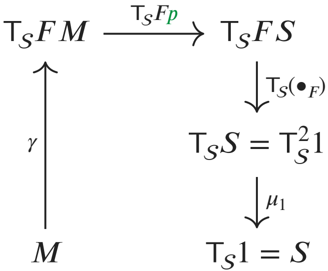


$$p = \begin{bmatrix} q_1 \mapsto 22/27 \\ q_2 \mapsto 14/27 \\ q_3 \mapsto 1 \end{bmatrix}$$



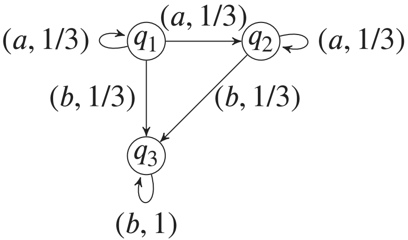
Calculating ν-Extent

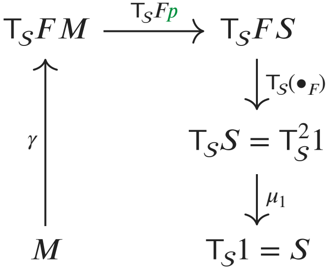




$$p = \begin{bmatrix} q_1 \mapsto \cdots \\ q_2 \mapsto \cdots \\ q_3 \mapsto \cdots \end{bmatrix}$$

Calculating ν-Extent





$$p = \begin{bmatrix} q_1 \mapsto 3/4 \\ q_2 \mapsto 1/2 \\ q_3 \mapsto 1 \end{bmatrix}$$

In this case, ν-extent is the liveness probability.

Trace Similarity

Given two $\mathsf{T}_{\mathcal{S}} \circ F$-coalgebras $\mathcal{M} = (M, \gamma)$ and $\mathcal{N} = (N, \delta)$, the (**maximal** / **finite**) **trace similarity relation** is an $\mathcal{S}$-valued relation between M and N that is the (greatest / least) fixed point of the operator below.

$$
\left(M \times N \xrightarrow{r} S\right) \;\overset{\mathbf{opr}}{\longmapsto}\;
\begin{array}{ccc}
\mathsf{T}_{\mathcal{S}}(FM \times FN) & \xrightarrow{\mathsf{T}_{\mathcal{S}}\mathrm{Rel}(F)(r)} & \mathsf{T}_{\mathcal{S}}S = \mathsf{T}_{\mathcal{S}}\mathsf{T}_{\mathcal{S}}1 \\
\uparrow{\scriptstyle \mathrm{dst}_{FC,FD}} & & \\
\mathsf{T}_{\mathcal{S}}FM \times \mathsf{T}_{\mathcal{S}}FN & & \downarrow{\scriptstyle \mu_1} \\
\uparrow{\scriptstyle \gamma \times \delta} & & \\
M \times N & & S
\end{array}
$$

Theorem. (Maximal / finite) trace similarity between $\mathcal{M}$ and $\mathcal{N}$
(which is an $\mathcal{S}$-valued relation between M and N)
= (ν-extent / μ-extent) of the product $\mathcal{M} \times \mathcal{N}$
(which is an $\mathcal{S}$-valued predicate over $M \times N$)

Parity (S, F)-Automata

A **parity** (S, F)**-automaton** $\mathcal{A} = (A, \alpha, \Omega)$ consists of:

- a $\mathsf{T}_{\mathcal{S}} \circ F$-coalgebra $(A,\ \alpha \colon\ A \to \mathsf{T}_{\mathcal{S}} F)$, and
- a **parity map** $\Omega \colon\ A \to \{1, 2, \dots\}$, such that $\mathrm{ran}(\Omega)$ is finite.

The notion of acceptance is not explicitly specified in this work.

Translation from formulas to automata

Syntax: $\varphi ::= x \mid [\lambda](\varphi_1, \dots, \varphi_{\mathrm{ar}(\lambda)}) \mid \sum_{i \in \{1,2,\dots,n\}} s_i \bullet \varphi_i \mid \mu x.\, \varphi \mid \nu x.\, \varphi$

- Assumption: φ is clean, strictly guarded.
- The set of states is: $\mathsf{Cl}(\varphi)$, the **closure** of φ, under taking subformulas and expanding fixed point binders.
- The transition $\beta\colon \mathsf{Cl}(\varphi) \to \mathsf{T}_{\mathcal{S}} F \mathsf{Cl}(\varphi)$ is given by:

$$\begin{aligned} \beta(\,[\lambda](\varphi_1, \dots, \varphi_{\mathrm{ar}(\lambda)})\,) &= \eta_{F\mathsf{Cl}(\varphi)}(\,\lambda(\varphi_1, \dots, \varphi_{\mathrm{ar}(\lambda)})\,), \\ \beta\Big(\sum_{i \in \{1,2,\dots,n\}} s_i \bullet \varphi_i\Big) &= \mu_{F\mathsf{Cl}(\varphi)}\Big(\sum_{i \in \{1,2,\dots,n\}} s_i \bullet \beta(\varphi_i)\Big), \\ \beta(\eta x.\, \varphi') &= \beta(\varphi'[\eta x.\, \varphi'/x]) \quad \text{for } \eta \in \{\mu, \nu\}. \end{aligned}$$

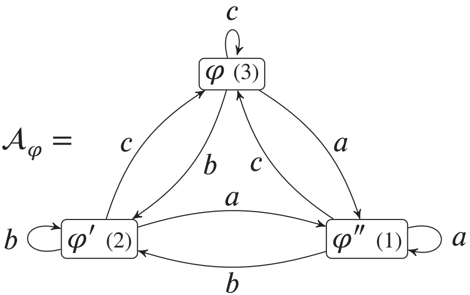
Parity assignment $\Omega\colon \mathsf{Cl}(\varphi) \to \mathbb{N}^+$

- Binders: •outer binders have larger parities.
  •odd number assigned for μ, even number for ν,
- Others: parities inherited “from above”.

Example of Translation

$$\varphi = \mu z.\, \nu y.\, \mu x.\, ([a]x + [b]y + [c]z)$$

$\mathcal{A}_\varphi =$

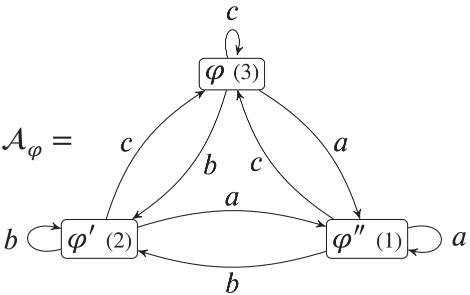


Example of Translation

$$\begin{aligned}\varphi &= \mu z.\, \nu y.\, \mu x.\, ([a]x + [b]y + [c]z)\\ \varphi' &= \nu y.\, \mu z.\, ([a]x + [b]y + [c]\varphi)\\ \varphi'' &= \mu x.\, ([a]x + [b]\varphi' + [c]\varphi)\end{aligned}$$

1. $\mathrm{Cl}(\varphi) = \{\varphi, \varphi', \varphi''\}$

$\mathcal{A}_\varphi =$

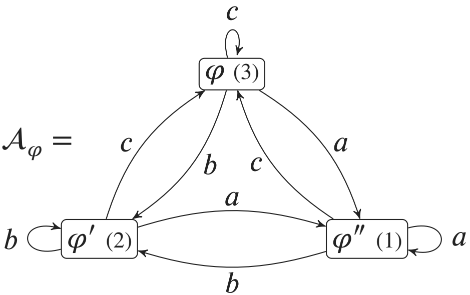


Example of Translation

$$\varphi \;= \mu z.\, \nu y.\, \mu x.\, ([a]x + [b]y + [c]z)$$
$$\varphi' \;= \nu y.\, \mu z.\, ([a]x + [b]y + [c]\varphi)$$
$$\varphi'' = \mu x.\, ([a]x + [b]\varphi' + [c]\varphi)$$

1. $\mathrm{Cl}(\varphi) = \{\varphi, \varphi', \varphi''\}$
2. parities: $[z \mapsto 3, y \mapsto 2, x \mapsto 1]$

$\mathcal{A}_\varphi =$

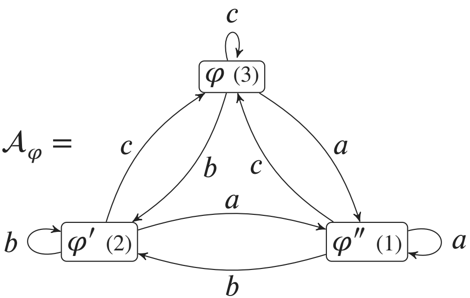


Example of Translation

$$\varphi = \mu z.\, \nu y.\, \mu x.\, ([a]x + [b]y + [c]z)$$
$$\varphi' = \nu y.\, \mu z.\, ([a]x + [b]y + [c]\varphi)$$
$$\varphi'' = \mu x.\, ([a]x + [b]\varphi' + [c]\varphi)$$

1. $\mathrm{Cl}(\varphi) = \{\varphi, \varphi', \varphi''\}$
2. parities: $[z \mapsto 3, y \mapsto 2, x \mapsto 1]$
3. $[\lambda] \to$ a label, "+"$\to$branching.

$\mathcal{A}_\varphi =$



Extents Generalized

The **extent** of a parity (S, F)-automaton is defined using an equational system [Hasuo, S and Cistea '16].

Given a parity (S, F)-automaton $\mathcal{A} = (A, \alpha, \Omega)$,
the extent of $\mathcal{A}$ is an S-valued predicate over A that is the solution of the equational system below.

$$\begin{bmatrix} u_1 & =_\mu & \mu_1 \circ \mathsf{T}_S(\bullet_F) \circ \mathsf{T}_S F[u_1, \ldots, u_n] \\ u_2 & =_\nu & \mu_1 \circ \mathsf{T}_S(\bullet_F) \circ \mathsf{T}_S F[u_1, \ldots, u_n] \\ & \vdots & \\ u_n & =_\eta & \mu_1 \circ \mathsf{T}_S(\bullet_F) \circ \mathsf{T}_S F[u_1, \ldots, u_n] \end{bmatrix} \Big\downarrow \begin{matrix} \text{inner formula} \\ \\ \text{outer formula} \end{matrix}$$

(c.f. $\left[\, u \;=_\nu\; \mu_1 \circ \mathsf{T}_S(\bullet_F) \circ \mathsf{T}_S F u \,\right]$ ν-extent
$\left[\, u \;=_\mu\; \mu_1 \circ \mathsf{T}_S(\bullet_F) \circ \mathsf{T}_S F u \,\right]$ μ-extent)

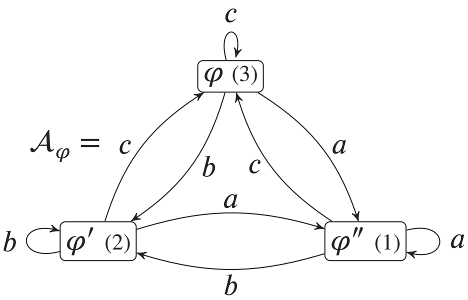
Product Coalgebras Generalized

The **product** automaton between

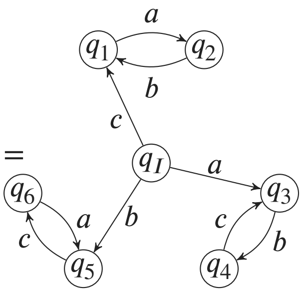
- $\mathcal{A} = (A, \alpha, \Omega)$ and
- $\mathcal{M} = (M, m)$

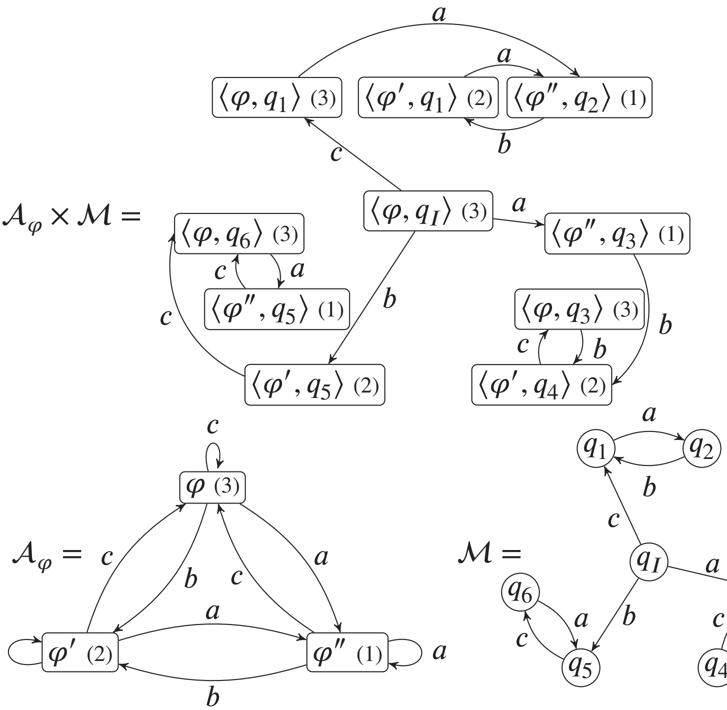
is: $\mathcal{A} \times \mathcal{M} = (A \times M,\ \alpha \otimes m,\ \Omega \circ \pi_1)$.

$\mathcal{A}_\varphi =$

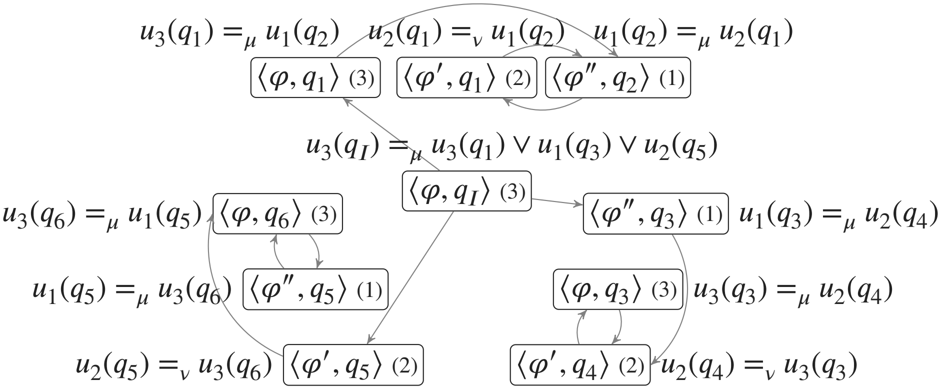


$\mathcal{M} =$

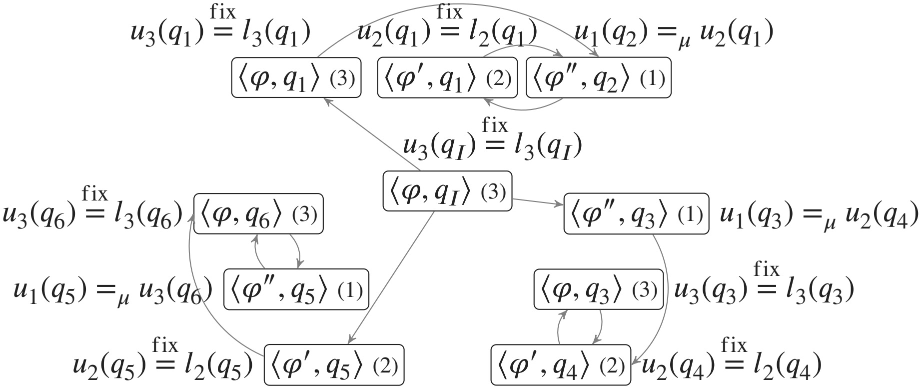



$\mathcal{A}_\varphi \times \mathcal{M} =$
$\langle \varphi, q_1 \rangle$ (3)
$\langle \varphi', q_1 \rangle$ (2)
$\langle \varphi'', q_2 \rangle$ (1)
$\langle \varphi, q_I \rangle$ (3)
$\langle \varphi'', q_3 \rangle$ (1)
$\langle \varphi, q_6 \rangle$ (3)
$\langle \varphi'', q_5 \rangle$ (1)
$\langle \varphi', q_5 \rangle$ (2)
$\langle \varphi, q_3 \rangle$ (3)
$\langle \varphi', q_4 \rangle$ (2)
$\mathcal{A}_\varphi =$
φ (3)
φ' (2)
φ'' (1)
$\mathcal{M} =$
q_1
q_2
q_I
q_3
q_4
q_5
q_6
a
b
c

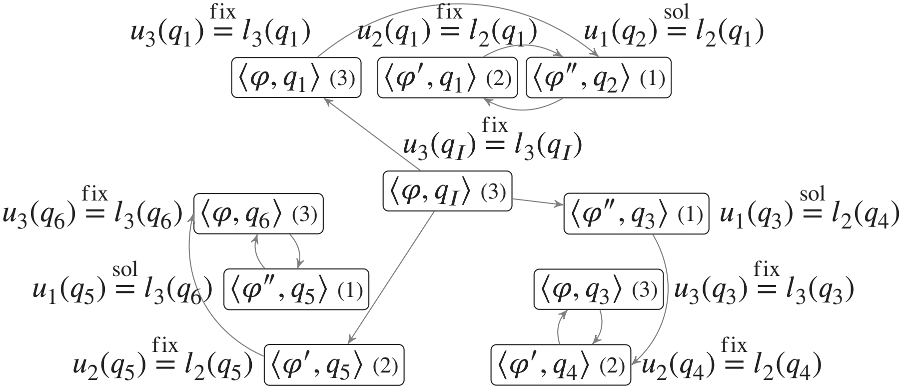
Extent-Based Semantics



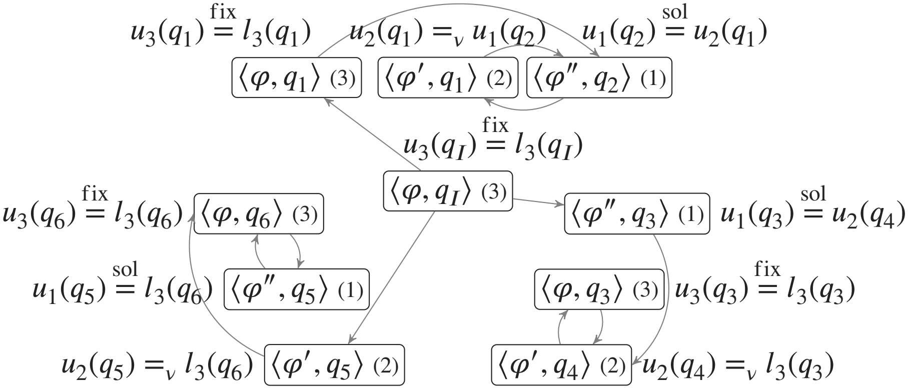
Extent-Based Semantics



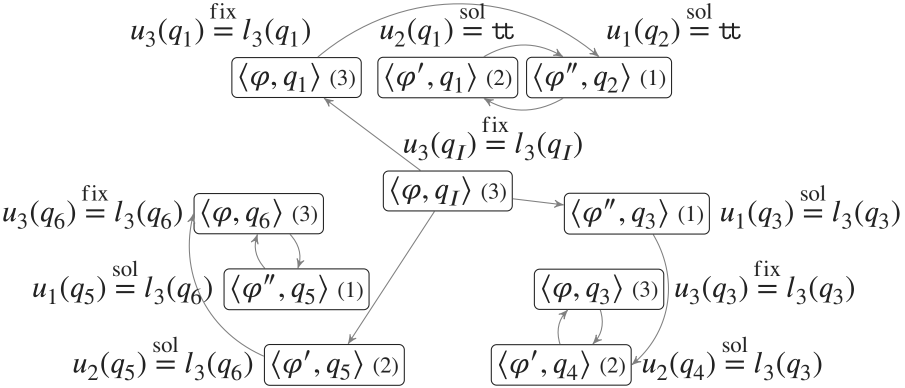
Extent-Based Semantics



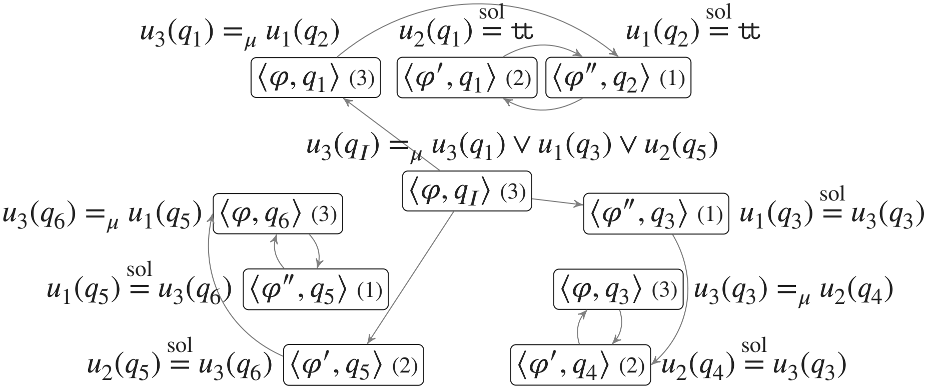
Extent-Based Semantics



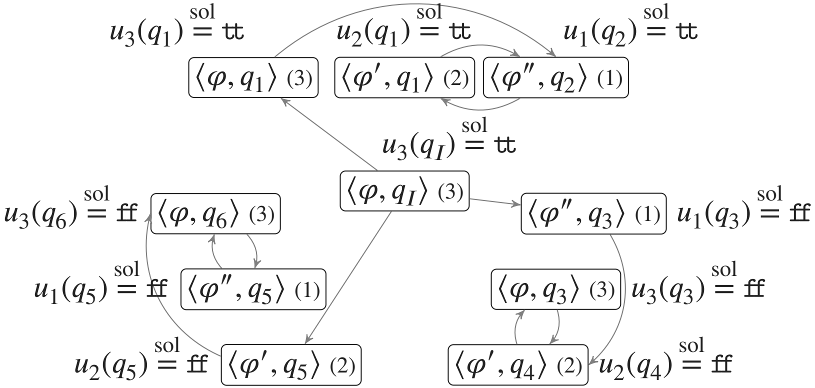
Extent-Based Semantics



Extent-Based Semantics



Extent-Based Semantics



Correctness of Extent-Based Semantics

Theorem. For

- a formula φ with automaton $\mathcal{A}_\varphi = (\mathsf{Cl}(\varphi), \beta, \Omega)$, $\mathrm{ran}(\Omega) \subseteq \{1, \ldots, n\}$,
- a coalgebra $\mathcal{M} = (M,\ \gamma\colon\ M \to \mathsf{T}_S F M)$,
- $e\colon\ A \to S$ the extent of the product parity automaton $\mathcal{A} \times \mathcal{M}$,

$$[\![\phi]\!]_\gamma(c)\ =\ e(c, \phi)\,.$$

Proof idea. Let $\varphi_x \coloneqq \nu x.\,(\varphi_y + [b]x)$ with $\varphi_y \coloneqq \mu y.\,(* + [a]y)$.

$$[\![\varphi_x]\!]_\gamma \equiv \begin{bmatrix} u_{\varphi_x} & =_2 & u_{\varphi_y} + u_{[b]\varphi_x} \\ u_{[b]\varphi_x} & =_2 & [\![b]\!] u_{\varphi_x} \\ u_{\varphi_y} & =_1 & u_* + u_{[a]\varphi_y} \\ u_* & =_1 & [\![*]\!] \\ u_{[a]\varphi_y} & =_1 & [\![a]\!] u_{\varphi_y} \end{bmatrix} \equiv \begin{bmatrix} u_{\varphi_x} & =_2 & [\![*]\!] + [\![a]\!] u_{\varphi_y} + [\![b]\!] u_{\varphi_x} \\ u_{\varphi_y} & =_1 & [\![*]\!] + [\![a]\!] u_{\varphi_y} \end{bmatrix} \equiv e(\_, \varphi_x)$$

Alternation Degree and Expressiveness of the Logic

Assume the arity of each letter is at most 1.

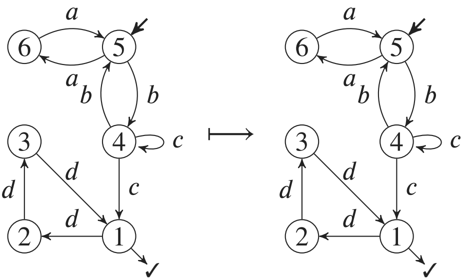
Theorem. For any parity (S, F)-automaton $\mathcal{A}$, there exists a parity (S, F)-automaton $\mathcal{A}'$ with the largest parity at most 2 such that

$$\mathrm{ext}(\mathcal{A} \times \mathcal{M}) \;=\; \mathrm{ext}(\mathcal{A}' \times \mathcal{M})\,.$$

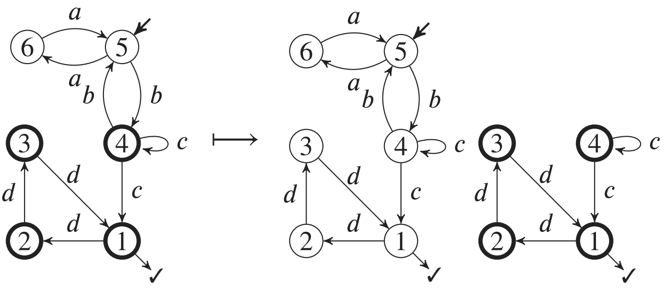
($\mathcal{A}'$ is a Büchi (S, F)-automaton.)

Corollary. Any linear time quantitative logic formula φ has an equivalent formula $\nu y_1. \dots \nu y_j. \mu x_1. \dots \mu x_i. \varphi'$.

Parity Word Autom.→Büchi Word Autom.

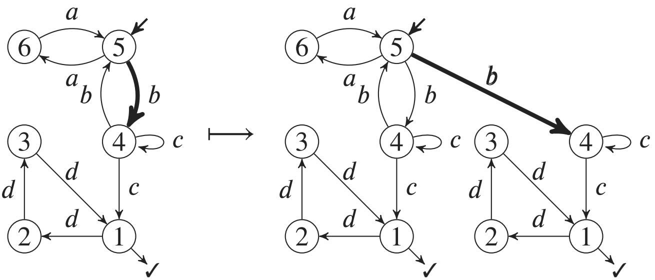


Parity Word Autom.→Büchi Word Autom.



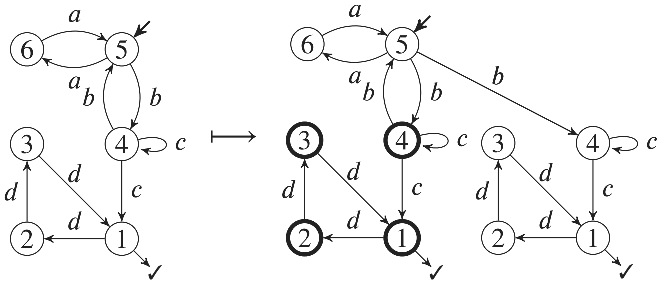
1. Create a copy of states with their parities lower than 5.

Parity Word Autom.→Büchi Word Autom.



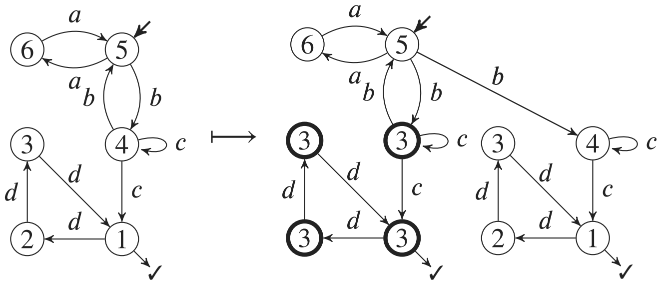
1. Create a copy of states with their parities lower than 5. Incoming edges to those states are also copied. (This is only possible if the semiring is **total**.)

Parity Word Autom.→Büchi Word Autom.



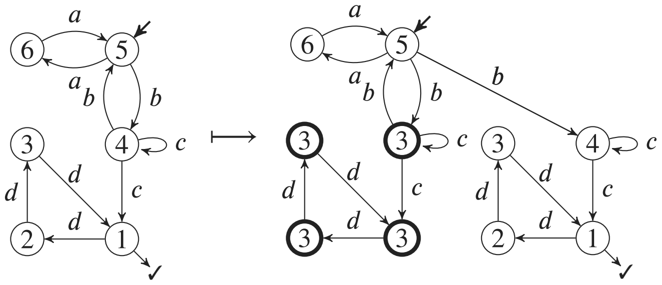
1. Create a copy of states with their parities lower than 5. Incoming edges to those states are also copied. (This is only possible if the semiring is **total**.)
2. For the old copy of states with priority lower than 4,

Parity Word Autom.→Büchi Word Autom.



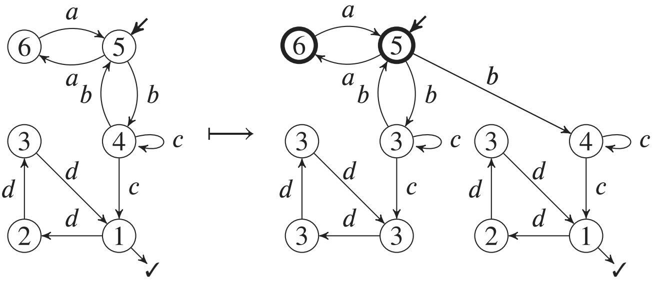
1. Create a copy of states with their parities lower than 5. Incoming edges to those states are also copied. (This is only possible if the semiring is **total**.)
2. For the old copy of states with priority lower than 4, let the new priority be 3

Parity Word Autom.→Büchi Word Autom.



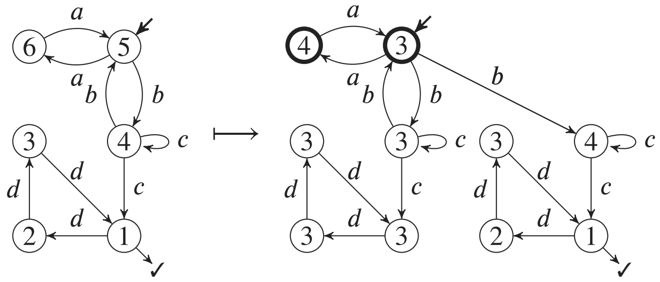
1. Create a copy of states with their parities lower than 5. Incoming edges to those states are also copied. (This is only possible if the semiring is **total**.)
2. For the old copy of states with priority lower than 4, let the new priority be 3 and drop explicit terminations (i.e. transitions with a nullary letter).

Parity Word Autom.→Büchi Word Autom.



1. Create a copy of states with their parities lower than 5. Incoming edges to those states are also copied. (This is only possible if the semiring is **total**.)
2. For the old copy of states with priority lower than 4, let the new priority be 3 and drop explicit terminations (i.e. transitions with a nullary letter).
3. Decrement the priorities of the other states by 2.

Parity Word Autom.→Büchi Word Autom.



1. Create a copy of states with their parities lower than 5. Incoming edges to those states are also copied. (This is only possible if the semiring is **total**.)
2. For the old copy of states with priority lower than 4, let the new priority be 3 and drop explicit terminations (i.e. transitions with a nullary letter).

Conclusion

Our contributions:

- We introduced extent-based semantics of quantitative linear time logic, and proved its correctness. This alternative semantics resembles to the workflow of LTL model checking.
- We showed quadratic reduction from quantitative parity word automata to quantitative Büchi word automata.

Future Work

- expressiveness result for the logics
- improved algorithm for computing extents