Quantitative Coalgebras for Optimal Synthesis

Corina Cîrstea

University of Southampton

17 December 2018

SYCO-2 Workshop, Glasgow

Motivation

- need for quantitative methods for complex system analysis / design
- challenges:
  - system heterogeneity: multitude of quantitative concerns (probabilistic / resource-aware / non-deterministic behaviour)
  - devise generic, compositional techniques
  - systematic use of abstraction

Plan of Talk

1. Quantitative systems as coalgebras (joint with I. Hasuo, S. Shimizu)
    - behaviour as (quantitative) traces, extents
    - quantitative linear-time logics
    - verification and synthesis
2. Quantitative components as coalgebras
    - trace semantics for components
    - linear-time logics for component-based systems
    - verification and synthesis: from homogeneous to heterogeneous systems

Compositionality at different levels ...

Quantitative Systems as Coalgebras

Systems as Coalgebras

- F-coalgebra: $X \xrightarrow{\delta} FX$ $\qquad$ (F : Set $\rightarrow$ Set)
- provides powerful abstraction:
  - labelled transition systems: $X \xrightarrow{\delta} \mathcal{P}_\omega(A \times X)$

Systems as Coalgebras

- F-coalgebra: $X \xrightarrow{\delta} FX$ $\qquad$ (F : Set $\to$ Set)
- provides powerful abstraction:
  - Markov Chains : $X \xrightarrow{\delta} \mathcal{D}X$

Systems as Coalgebras

- F-coalgebra: $X \xrightarrow{\delta} FX$ $\qquad$ ($F : \mathsf{Set} \to \mathsf{Set}$)
- provides powerful abstraction:
  - probabilistic transition systems: $X \xrightarrow{\delta} \mathcal{D}(A \times X)$

Systems as Coalgebras

- F-coalgebra: $X \xrightarrow{\delta} FX$ ($F : \mathsf{Set} \to \mathsf{Set}$)
- provides powerful abstraction:
  - weighted transition systems: $X \xrightarrow{\delta} W^{A \times X}$

Systems as Coalgebras

- F-coalgebra: $X \xrightarrow{\delta} FX$ $\quad$ (F : Set $\to$ Set)
- provides powerful abstraction:
  - weighted transition systems: $X \xrightarrow{\delta} W^{A \times X}$
  - determ. automata: $X \xrightarrow{\delta} \{0,1\} \times X^A$

Systems as Coalgebras

- F-coalgebra: $X \xrightarrow{\delta} FX$ $\qquad$ (F : Set $\to$ Set)
- provides powerful abstraction:
  - weighted transition systems: $X \xrightarrow{\delta} W^{A\times X}$
  - nondet. automata: $X \xrightarrow{\delta} \{0,1\} \times \mathcal{P}(X)^A$

Systems as Coalgebras

- F-coalgebra: $X \xrightarrow{\delta} FX$ $\qquad$ (F : Set $\to$ Set)
- provides powerful abstraction:
  - weighted transition systems: $X \xrightarrow{\delta} W^{A \times X}$
  - probabilistic automata: $X \xrightarrow{\delta} \mathcal{P}(\mathcal{D}X)^A$

Systems as Coalgebras

- F-coalgebra: $X \xrightarrow{\delta} FX$ ($F : \mathsf{Set} \to \mathsf{Set}$)
- provides powerful abstraction:
  - weighted transition systems: $X \xrightarrow{\delta} W^{A \times X}$
  - probabilistic automata: $X \xrightarrow{\delta} \mathcal{P}(\mathcal{D}X)^A$
- observational indistinguishability as bisimilarity
  - instantiates to known equivalences

Systems as Coalgebras

- F-coalgebra: $X \xrightarrow{\delta} FX$ ($F : \mathsf{Set} \to \mathsf{Set}$)
- provides powerful abstraction:
  - weighted transition systems: $X \xrightarrow{\delta} W^{A \times X}$
  - probabilistic automata: $X \xrightarrow{\delta} \mathcal{P}(\mathcal{D}X)^A$
- observational indistinguishability as bisimilarity
  - instantiates to known equivalences
- abstract behaviours as states in final coalgebra
  - e.g. determ. automata: $\{0,1\}^{A^*}$, behaviour as accepted language

Systems as Coalgebras

- F-coalgebra: $X \xrightarrow{\delta} FX$ ($F : \mathsf{Set} \to \mathsf{Set}$)
- provides powerful abstraction:
  - weighted transition systems: $X \xrightarrow{\delta} W^{A \times X}$
  - probabilistic automata: $X \xrightarrow{\delta} \mathcal{P}(\mathcal{D}X)^A$
- observational indistinguishability as bisimilarity
  - instantiates to known equivalences
- abstract behaviours as states in final coalgebra
  - e.g. determ. automata: $\{0,1\}^{A^*}$, behaviour as accepted language
- compositionality (at the level of system types):
  - logics, their expressiveness, completeness of proof systems
  - notions of simulation
  - ...

Quantitative Systems as Coalgebras

- partial commutative semiring for quantitities: $(S, +, 0, \bullet, 1)$
  - Boolean semiring: $(\{0,1\}, \vee, 0, \wedge, 1)$
  - Probab. semiring: $([0,1], +, 0, \times, 1)$
  - Tropical semiring: $(\mathbb{N}^\infty, \min, \infty, +, 0)$
- natural preorder $\sqsubseteq$ on S induced by $+$:
  - $\leq$ on $\{0,1\}$, $\quad \leq$ on $[0,1]$, $\quad \geq$ on $\mathbb{N}^\infty$
- (closed) system with quantitative branching: $X \xrightarrow{\delta} \mathsf{T}_S F X$
  - $\mathsf{T}_S X = \sum_{i \in \{1,2,\ldots,n\}} s_i \bullet x_i$ for weighted choices
  - F : Set $\rightarrow$ Set for "linear" behaviour

Systems with Branching and Actions

- actions with associated arities: $(\Lambda, \mathsf{ar} : \Lambda \to \mathbb{N})$

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

- e.g. finite/infinite words: $\{a \mapsto 1, b \mapsto 1, \checkmark \mapsto 0\}$

$$FX \ = \ X \ + \ X \ + \ 1 \simeq \{a, b\} \times X + 1$$

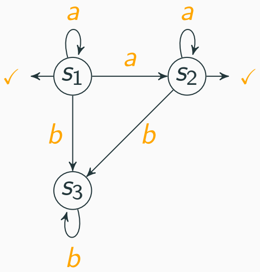
- e.g. finite/infinite labelled binary trees: $\{a \mapsto 2, b \mapsto 2, \checkmark \mapsto 0\}$

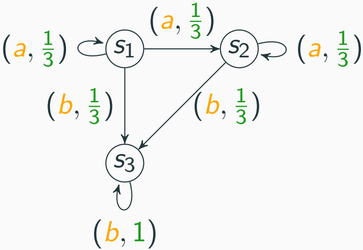
$$FX = X \times X \ + \ X \times X \ + \ 1 \simeq \{a, b\} \times X \times X + 1$$

- more complex behaviour: $\{a \mapsto 2, b \mapsto 1, \checkmark \mapsto 0\}$

$$FX = X \times X \ + \ X \ + \ 1 \simeq \{a\} \times X \times X + \{b\} \times X + 1$$

Example: Non-deterministic and Probabilistic Branching





LTSs with explicit termination

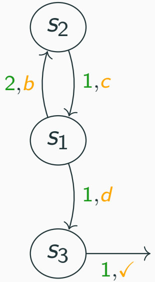
- Actions:
  $X \to \{a, b\} \times X + \{\checkmark\} = FX$
- Nondet. branching:
  $X \to \mathcal{P}FX$

Markov chains

- Actions:
  $X \to \{a, b\} \times X = F'X$
- Probab. branching:
  $X \to \mathcal{D}F'X$

Example: Weighted Branching

- weights for resource usage:



- minimise resource usage
- must also model resource gain ...

Goals: trace semantics, logics, verification, synthesis

- different types of branching, uniformly
- systems with several types of branching

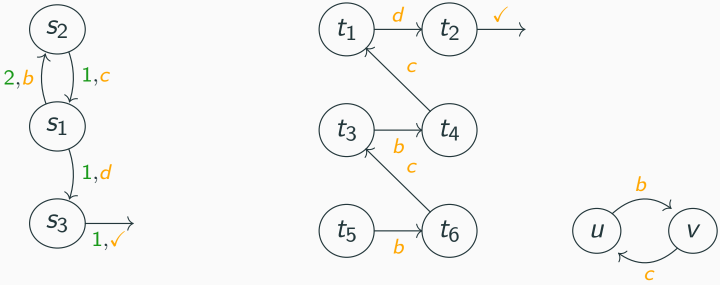
Maximal Trace Semantics for Branching Systems [C'17]

- $X \xrightarrow{\delta} \mathsf{T}_S F X$
- why maximal traces ?
- domain for maximal traces: final F-coalgebra $Z \xrightarrow{\zeta} FZ$
  - e.g. $Z = \{a, b\}^* \cup \{a, b\}^\omega$
- maximal trace semantics maps $(x \in X, t \in Z)$ to $s \in S$
  - obtained as greatest fixpoint of operator:

$$\begin{array}{ccccccc} X \times Z & & FX \times FZ & & \mathsf{T}_S F X \times F Z & & X \times Z \\ \downarrow & \xmapsto{\mathbf{Rel}_F} & \downarrow & \xmapsto{E_{\mathsf{T}_S}} & \downarrow & \xmapsto{(\delta \times \zeta)^*} & \downarrow \\ S & & S & & S & & S \end{array}$$

  - non-determ./probab. models: realisability/likelihood of each maximal trace
  - resource-aware models: minimal resources needed for each maximal trace

Example: Resource-Aware Models



. . .

(s_1, t_1)	(s_1, t_2)	(s_1, t_3)	(s_2, t_4)	(s_1, u)	(s_2, v)
0	0	0	0	0	0
1	∞	2	1	2	1
2		3	3	3	3
. . .					
2	∞	5	3	∞	∞

Modelling Offsetting

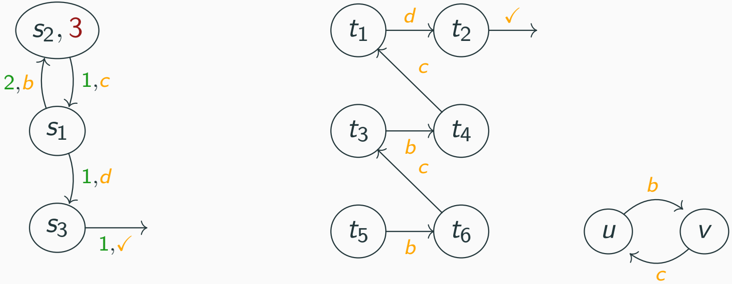
- move to coalgebras of type $S \times (\mathsf{T}_S \circ F)$
  - first component models offsetting
  - e.g. $S = (\mathbb{N}^\infty, \min, \infty, +, 0)$:
    - weights model resource usage
    - offsets model resource gains
- define $\oslash : S \times S \to S$ by

$$s \oslash t = \inf\{u \mid u \bullet t \sqsupseteq s\}\,.$$

  - e.g. $S = (\mathbb{N}^\infty, \min, \infty, +, 0)$:

$$n \ominus m = \begin{cases} \max(n - m, 0), & \text{if } m \neq \infty \text{ or } n \neq \infty, \\ \infty, & \text{otherwise.} \end{cases}$$

Example: Resource-aware Models with Offsetting



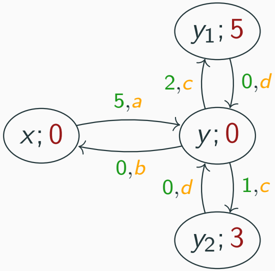
...

(s_1, t_1)	(s_1, t_2)	(s_1, t_3)	(s_2, t_4)	(s_1, u)	(s_2, v)
0	0	0	0	0	0
1	∞	2	0	2	0
2		2	0	2	0
...					
2	∞	2	0	2	0

Generalising Non-Emptiness: Extents

$$X \xrightarrow{\delta} S \times \mathsf{T}_S F X$$

- extent ext : $X \to S$
  - instantiates to existence/likelihood/minimal resources across *all* traces
  - defined as greatest fixpoint ...
- e.g. $S = (\mathbb{N}^\infty, \min, \infty, +, 0)$, $F = A \times \mathrm{Id}$:



$$\begin{bmatrix} e_x & =_\nu & e_y + 5 \\ e_y & =_\nu & \min(e_x, e_{y_1} + 2, e_{y_2} + 1) \\ e_{y_1} & =_\nu & e_y \ominus 5 \\ e_{y_2} & =_\nu & e_y \ominus 3 \end{bmatrix}$$

	e_x	e_y	e_{y_1}	e_{y_2}
ext	6	1	0	0

Dealing with More Complex Structure, Compositionally

- $X \xrightarrow{\delta} F_1 \, \mathsf{T}_S \, F_2 \, \mathsf{T}_S \ldots \mathsf{T}_S \, F_n \, X$ or combinations using $+/\times$!
  - e.g. $X \xrightarrow{\delta} A \times \mathsf{T}_S(X \times X) + B \times \mathsf{T}_S(1 + X)$
- final $F_1 \circ \ldots \circ F_n$-coalgebra (Z, ζ) gives linear behaviours
- trace semantics as g.f.p. of operator on S-valued relations:

$$\mathbf{Rel}_{F_1} \, ; \, E_{\mathsf{T}_S} \, ; \, \mathbf{Rel}_{F_2} \, ; \, E_{\mathsf{T}_S} \, ; \, \ldots E_{\mathsf{T}_S} \, ; \, \mathbf{Rel}_{F_n}$$

- generalises to coalgebras with offsetting:

$$X \xrightarrow{\delta} S \times \ldots$$

Fixpoint Logics for Quantit. Systems, Compositionally [C'14]

$X \xrightarrow{\delta} F_1\, \mathsf{T}_S\, F_2\, \mathsf{T}_S \ldots \mathsf{T}_S\, F_n\, X$ or combinations using $+/\times$!

- system structure drives associated *multi-sorted* S-valued logic
  - $\top$ interpreted as extent !
  - modal operators induced by linear type $F_1 \circ F_2 \circ \ldots \circ F_n$
  - fixpoint operators, interpreted over $(S, \sqsubseteq)$
- e.g. $X \xrightarrow{\delta} G\mathsf{T}_S F X$
  - modal operators induced by $G, F \implies$ modal formulas $[\lambda][\lambda']\varphi$
  - semantics of formulas induced by quantitative predicate liftings:

$$\begin{array}{c} X \\ \downarrow \\ S \end{array} \overset{[\![\lambda]\!]}{\longmapsto} \begin{array}{c} FX \\ \downarrow \\ S \end{array} \overset{\text{ext}}{\longmapsto} \begin{array}{c} \mathsf{T}_S F X \\ \downarrow \\ S \end{array} \overset{[\![\lambda']\!]}{\longmapsto} \begin{array}{c} G\mathsf{T}_S F X \\ \downarrow \\ S \end{array} \overset{\delta^*}{\longmapsto} \begin{array}{c} X \\ \downarrow \\ S \end{array}$$

- generalises to coalgebras with offsetting ...

Note: step-wise semantics for the logics !

Fixpoint Logics for Quantitative Systems: Example (more later!)

- $X \xrightarrow{\delta} S \times \mathsf{T}_S(\{c, d\} \times X)$
- modalities derived directly from F:
  - binary modality $(c, \_) \sqcup (d, \_)$ makes up for absence of $\wedge/\vee$
  - e.g. eventually c: $\mu x.((c, \top) \sqcup (d, x))$
  - e.g. infinitely often c: $\nu x.\mu y.((c, x) \sqcup (d, y))$
- e.g. $S = (\mathbb{N}^\infty, \min, \infty, +, 0)$:
  - measures minimal resources required for linear property

Quantitative, Automata-Based Verification [CSH'17]

Given:

- system: pointed $S \times \mathsf{T}_S F$-coalgebra $\mathcal{S}$
- property: φ

compute $[\![\varphi]\!]_{\mathcal{S}}$!

We need:

1. notion of parity automaton $\mathcal{A}$
   $=$ "disjunctive" F-coalgebra (A, α) $+$ parity map $\Omega : A \to \mathbb{N}$
2. translation from formula φ to parity automaton $\mathcal{A}_\varphi$
3. product automaton $\mathcal{S} \otimes \mathcal{A}$
4. extent of quantitative parity automaton

such that

$$\text{extent}(\mathcal{S} \otimes \mathcal{A}_\varphi) = [\![\varphi]\!]_{\mathcal{S}}$$

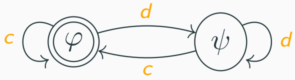
2. Translation from Formulas to Automata: Example

$FX = \{c, d\} \times X$

$\varphi = \nu x.\mu y.((c, x) \sqcup (d, y))$

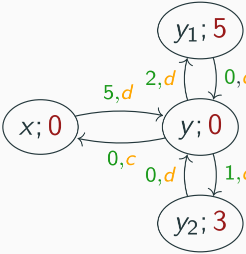
$\psi = \mu y.((c, \varphi) \sqcup (d, y))$

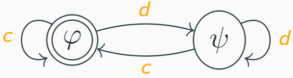
- automaton states given by $\mathsf{Cl}(\varphi)$:



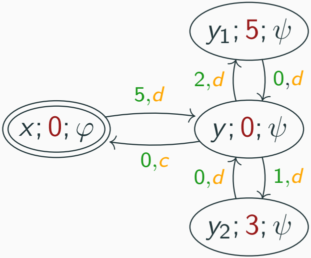
- "disjunctive" branching in $\mathcal{A}_\varphi$
- parity assignment:
  - outer fixpoints have larger priorities; odd for μ, even for ν

3. Product of System and Parity Automaton [CSH'17, C'19]



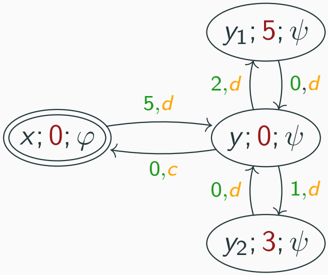


$\mathcal{S} \otimes \mathcal{A}$ inherits weights/offsetting from $\mathcal{S}$ and parities from $\mathcal{A}$:



4. Extent of Parity Automaton [CSH'17], Strategies [C'19]

- extent only measures traces which conform to the parity condition:



$$\begin{bmatrix} e_x & =_\nu & e_y + 5 \\ e_y & =_\mu & \min(e_x, e_{y_1} + 2, e_{y_2} + 1) \\ e_{y_1} & =_\mu & e_y \ominus 5 \\ e_{y_2} & =_\mu & e_y \ominus 3 \end{bmatrix}$$

	e_x	e_y	e_{y_1}	e_{y_2}
ext	6	1	0	0

- view product as one-player game: objective is to minimise resources
- solution of equational system can be used to synthesise memory-full strategy for satisfying φ with minimal cost !
  - "good for energy" strategy improves resources: $(y, M < 6) \mapsto y_1$
  - "attractor" strategy satisfies parity condition: $(y, M \geq 6) \mapsto x$

Computing Extents [CSH'17]

Assume: strictly increasing/decreasing chains in S are finite

(e.g. in bounded version of tropical semiring)

- $O(m \times n^{|\mathrm{ran}(\Omega)|})$ complexity for basic algorithm
  - large hidden constant !
- improved complexity $O(m \times n^2)$ when $FX = \Sigma \times X + \Delta$
  - translation from parity to Büchi automaton which preserves quantitative language !

Quantitative Components as Coalgebras

Components as Coalgebras [Barbosa, Hasuo&Jacobs, ...]

For T commutative monad:

- coalgebraic component:

  $$X \times A \xrightarrow{\gamma} T(X \times B) \quad \in \; \mathbf{Comp}(T, A, B)$$

- sequential composition (uses Kleisli composition):

  $$\ggg \, : \mathbf{Comp}(T, A, B) \times \mathbf{Comp}(T, B, C) \to \mathbf{Comp}(T, A, C)$$

- multiplicative parallel composition (uses monad commutativity):

  $$\| \, : \mathbf{Comp}(T, A, B) \times \mathbf{Comp}(T, C, D) \to \mathbf{Comp}(T, A \times C, B \times D)$$

Take $T := T_S$ above. Some questions:

1. Trace semantics for components ? Compositionality w.r.t. $\ggg$ and $\|$?
2. Combine heterogeneous components ?
3. Verification of component-based systems ?

1. Trace Semantics for Components

- viewing $X \times A \to \mathsf{T}(X \times B)$ as coalgebra $X \xrightarrow{\delta} \mathsf{T}(X \times B)^A$ yields wrong notion of trace semantics ...
- e.g. $S = (\mathbb{N}^\infty, \min, \infty, +, 0)$:
  - final $(\mathsf{Id} \times B)^A$-coalgebra Z: causal stream functions $f : A^\omega \to B^\omega$
  - trace semantics gives minimal resources needed to exhibit $f : A^\omega \to B^\omega$ from $x \in X$:

$$X \times (B^\omega)^{A^\omega} \to S$$

  - must capture minimal resources for exhibiting $b \in B^\omega$ from $x \in X$ on input $a \in A^\omega$:

$$X \times A^\omega \times B^\omega \to S$$

  - can not get this by changing the relation liftings for $(\mathsf{Id} \times B)^A$!

Trace Semantics for Components (Cont'd)

$X \times A \xrightarrow{\delta} \mathsf{T}(X \times B)$

- final $\mathsf{Id} \times A$-coalgebra (A^ω, ζ)
- final $\mathsf{Id} \times B$-coalgebra (B^ω, ζ')
- trace semantics

$$\mathsf{tr} : X \times A^\omega \times B^\omega \to S$$

as greatest fixpoint:

$$X \times A^\omega \times B^\omega \xrightarrow{\mathsf{id} \times \zeta \times \mathsf{id}} (X \times A) \times A^\omega \times B^\omega \xrightarrow{\delta \times \mathsf{id} \times \zeta'} \mathsf{T}_S(X \times B) \times A^\omega \times (B \times B^\omega)$$

$$\begin{array}{ccccc} X \times A^\omega \times B^\omega & & (X \times B) \times A^\omega \times (B \times B^\omega) & & \mathsf{T}_S(X \times B) \times A^\omega \times (B \times B^\omega) \\ \downarrow & \overset{\mathbf{Rel}_{\mathsf{Id} \times B}}{\longmapsto} & \downarrow & \overset{E_{\mathsf{T}_S}}{\longmapsto} & \downarrow \\ S & & S & & S \end{array}$$

Note: generalises to components with offsetting !

Trace Semantics for Components is Compositional w.r.t. $\|$

$$X \times A \xrightarrow{c} \mathsf{T}(X \times B) \qquad Y \times C \xrightarrow{d} \mathsf{T}(Y \times D)$$

$$\Downarrow$$

$$X \times Y \times A \times C \xrightarrow{c\|d} \mathsf{T}(X \times Y \times B \times D)$$

Thm. For $x \in X$ and $y \in Y$:

$$\mathsf{tr}_{c\|d}(x, y, (as, cs), (bs, ds)) = \mathsf{tr}_c(x, as, bs) \bullet \mathsf{tr}_d(y, cs, ds)$$

Trace Semantics for Components is Compositional w.r.t. $\|$

$$X \times A \xrightarrow{c} \mathsf{T}(X \times B) \qquad\qquad Y \times C \xrightarrow{d} \mathsf{T}(Y \times D)$$

$$\Downarrow$$

$$X \times Y \times A \times C \xrightarrow{c\|d} \mathsf{T}(X \times Y \times B \times D)$$

Thm. For $x \in X$ and $y \in Y$:

$$\mathsf{tr}_{c\|d}(x, y, (as, cs), (bs, ds)) = \mathsf{tr}_c(x, as, bs) \bullet \mathsf{tr}_d(y, cs, ds)$$

To do: generalise to components with offsetting

$$\mathsf{tr}_{c\|d}(x, y, (as, cs), (bs, ds)) \sqsupseteq \mathsf{tr}_c(x, as, bs) \bullet \mathsf{tr}_d(y, cs, ds) \ ?$$

Trace Semantics for Components is Compositional w.r.t. $\ggg$

$$X \times A \xrightarrow{c} \mathsf{T}(X \times B) \qquad Y \times B \xrightarrow{d} \mathsf{T}(Y \times C)$$

$$\Downarrow$$

$$X \times Y \times A \xrightarrow{c \ggg d} \mathsf{T}(X \times Y \times B)$$

We would like:

$$\mathsf{tr}_{c \ggg d}(x, y, as, cs) = \sum_{bs \in B^\omega} \mathsf{tr}_c(x, as, bs) \bullet \mathsf{tr}_d(y, bs, cs)$$

Lemma

$$\mathsf{tr}_c(x, as, bs) = \prod_{n \in \omega} \mathsf{tr}_{c,n}(x, \pi_n(as), \pi_n(bs))$$

with $\mathsf{tr}_{c,n} : X \times A^n \times B^n \to S$ defined inductively ...

Thm.

$$\mathsf{tr}_{c \ggg d, n}(x, y, as, cs) = \sum_{bs \in B^n} \mathsf{tr}_{c,n}(x, as, bs) \bullet \mathsf{tr}_{d,n}(y, bs, cs)$$

2. Combining Heterogeneous Components ?

$$X \times A \xrightarrow{\gamma} \mathsf{T}(X \times B) \qquad Y \times B \xrightarrow{\delta} \mathsf{T}'(Y \times C)$$

- sequential composition $\gamma \ggg \delta$:

$$X \times Y \times A \xrightarrow{(\gamma \times \mathsf{id}_Y);\mathsf{st}_\mathsf{T}} \mathsf{T}(X \times Y \times B) \xrightarrow{\mathsf{T}(\mathsf{id}_X \times \delta);\mathsf{st}_{\mathsf{T}'}} \mathsf{T}\mathsf{T}'(X \times Y \times C)$$

- multiplicative parallel composition $\gamma \parallel \delta$:

$$X \times Y \times A \times C \xrightarrow{\gamma \times \delta} \mathsf{T}(X \times B) \times \mathsf{T}'(Y \times D) \xrightarrow{\mathsf{st}_\mathsf{T};\mathsf{st}_{\mathsf{T}'}} \mathsf{T}\mathsf{T}'(X \times Y \times B \times D)$$

Relevance ??

Trace Semantics for Heterogeneous Systems of Components ??

$$X \times Y \times A \xrightarrow{\gamma \ggg \delta} \mathsf{T}_S \mathsf{T}_{S'}(X \times Y \times C)$$

- unclear how to define trace sematics ...
- ... but can focus on a single type of quantity, e.g. S':
  - consider abstraction: $X \times Y \times A \xrightarrow{\gamma \ggg \delta} \mathcal{P}\mathsf{T}_{S'}(X \times Y \times C)$
  - T_S-component is cooperative: use $\max_S$ in def. of trace semantics
    $\implies$ $\mathrm{tr}(x, y, as, cs)$ captures best case
  - T_S-component is adversarial: use $\min_S$ in def. of trace semantics
    $\implies$ $\mathrm{tr}(x, y, as, cs)$ captures worst case

$X \xrightarrow{\delta} (\mathsf{T}_S(\{c,d\} \times X))^{\{a,b\}}$

- two-sorted logic, nested modalities !
  - $FX = \{c,d\} \times X \implies$ binary modality $(c, \_) \sqcup (d, \_)$
  - $GX = X^{\{a,b\}} \implies$ binary modality $(a, \_) \sqcap (b, \_)$

  $\sqcap$ interpreted as min in $(S, \sqsubseteq)$

- every a eventually followed by c:

$$\varphi \;:=\; \nu x.(\, [a](\mu y.(c,x) \sqcup (d, [a]y \sqcap [b]y)) \\ \sqcap\; [b]((c,x) \sqcup (d,x))\,)$$

- e.g. $S = (\mathbb{N}^\infty, \min, \infty, +, 0) \implies$ minimal resources needed for φ in the worst case (worst choice of input stream)

Fixpoint Logics for Heterogeneous Systems, Compositionally

$$X \xrightarrow{\delta} (\mathcal{P}\,\mathsf{T}_S(\{c,d\} \times X))^{\{a,b\}}$$

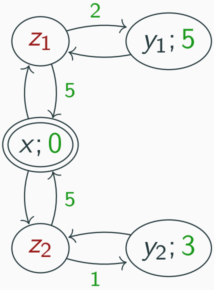
- same logic as before ...
- use $\min_S$ / $\max_S$ for adversarial / cooperative component

Quantitative Parity Games, Extents, Strategies [C'19]

- quantitative parity game:

$$X \xrightarrow{\delta} S \times \mathcal{P}\mathrm{T}_S X \quad + \quad \Omega : X \to \mathbb{N}$$

- derived from Adversary model ⋙ Component space
- extent:



$$\begin{bmatrix} e_x & =_\nu & \max(\min(e_x + 5, e_{y_1} + 2), \\ & & \min(e_x + 5, e_{y_2} + 1)) \\ e_{y_1} & =_\mu & \min(e_x + 5, e_{y_1} + 2) \ominus 5 \\ e_{y_2} & =_\mu & \min(e_x + 5, e_{y_2} + 1) \ominus 3 \end{bmatrix}$$

	e_x	e_{y_1}	e_{y_2}
ext	2	0	0

- synthesise memory-full strategy which minimises resources
  - "good for energy" strategy: $(x, z_1, M < 7) \mapsto y_1$
  - "attractor" strategy: $(x, z_1, M \geq 7) \mapsto x$

Conclusions

Further Challenges

- handle complexity (e.g. through abstraction)
- more general (e.g. dynamic) components?
- other quantitative monads?
- ...

Ultimate goal: algorithms and tools for quantitative verification and synthesis