Flat coalgebraic fixed point logics

Daniel Hausmann

FAU Erlangen-Nürnberg

November 20, 2012

Coalgebraic Modal Logic

- Generic semantic framework
- Coverage of e.g.:
  - **K**, **KD**, $\mathcal{HML}$
  - Graded modal logic: $\Diamond_n \phi$ – "ϕ holds in at least *n* successor states"
  - Probabilistic modal logic: $\Diamond_p \phi$ – "there is a successor where ϕ holds with probability *p*"
  - Coalition logic: $\Box_C \phi$ – "coalition *C* can enforce ϕ"
  - Conditional logics, *binary*: $\phi \Rightarrow \psi$ – "if ϕ then usually ψ"
  - Basic description logics (e.g. **K** + global assumptions)
  - Fixed point logics (e.g. $AF\phi$ – "ϕ holds eventually on all paths")
- Generic complexity bounds, algorithms

Coalgebraic Modal Logic, ctd.

Set-endofunctor T, T-coalgebra $\mathcal{C} = (X, \xi)$, structure $\xi : X \to TX$.

Syntax

Modal operators $\Lambda \ni \heartsuit$:

$$\mathcal{CML}(\Lambda) \ni A_1, \dots, A_n := \neg A_1 \mid A_1 \wedge A_2 \mid \heartsuit(A_1, \dots, A_n)$$

Semantics

Predicate Liftings $([\![\heartsuit]\!]_Y : (\mathcal{P}Y)^n \to \mathcal{P}TY)_{Y \in \mathbf{Set}}$.

Satisfaction of modal operators (for $[\![\phi]\!]_{\mathcal{C}} = \{c \in X \mid \mathcal{C}, c \models \phi\}$):

$$[\![\heartsuit(A_1, \dots, A_n)]\!]_{\mathcal{C}} = \xi^{-1} \circ [\![\heartsuit]\!]_X([\![A_1]\!]_{\mathcal{C}}, \dots, [\![A_n]\!]_{\mathcal{C}})$$

K: $T = \mathcal{P}$, $[\![\Box]\!]_X(Y) = \{A \in \mathcal{P}(X) \mid A \subseteq Y\}$,
$[\![\Diamond]\!]_X(Y) = \{A \in \mathcal{P}(X) \mid A \cap Y \neq \emptyset\}$.

CK: $T = \mathcal{CK}$, $\mathcal{CK}(X) = \{f : \mathcal{P}(X) \to \mathcal{P}(X)\}$,
$[\![\Rightarrow]\!]_X(Y, Z) = \{f \in \mathcal{CK}(X) \mid f(Y) \subseteq Z\}$.

Coalgebraic Modal Logic, ctd.

Set-endofunctor T, T-coalgebra $\mathcal{C} = (X, \xi)$, structure $\xi : X \to TX$.

Syntax

Modal operators $\Lambda \ni \heartsuit$:

$$\mathcal{CML}(\Lambda) \ni A_1, \ldots, A_n := \neg A_1 \mid A_1 \wedge A_2 \mid \heartsuit(A_1, \ldots, A_n)$$

Semantics

Predicate Liftings $([\![\heartsuit]\!]_Y : (\mathcal{P}Y)^n \to \mathcal{P}TY)_{Y \in \mathbf{Set}}$.

Satisfaction of modal operators (for $[\![\phi]\!]_\mathcal{C} = \{c \in X \mid \mathcal{C}, c \models \phi\}$):

$$[\![\heartsuit(A_1, \ldots, A_n)]\!]_\mathcal{C} = \xi^{-1} \circ [\![\heartsuit]\!]_X([\![A_1]\!]_\mathcal{C}, \ldots, [\![A_n]\!]_\mathcal{C})$$

K: $T = \mathcal{P}$, $[\![\Box]\!]_X(Y) = \{A \in \mathcal{P}(X) \mid A \subseteq Y\}$,
$[\![\Diamond]\!]_X(Y) = \{A \in \mathcal{P}(X) \mid A \cap Y \neq \emptyset\}$.

CK: $T = \mathcal{CK}$, $\mathcal{CK}(X) = \{f : \mathcal{P}(X) \to \mathcal{P}(X)\}$,
$[\![\Rightarrow]\!]_X(Y, Z) = \{f \in \mathcal{CK}(X) \mid f(Y) \subseteq Z\}$.

Coalgebraic Modal Logic, ctd.

Set-endofunctor T, T-coalgebra $\mathcal{C} = (X, \xi)$, structure $\xi : X \to TX$.

Syntax

Modal operators $\Lambda \ni \heartsuit$:

$$\mathcal{CML}(\Lambda) \ni A_1, \ldots, A_n := \neg A_1 \mid A_1 \wedge A_2 \mid \heartsuit(A_1, \ldots, A_n)$$

Semantics

Predicate Liftings $([\![\heartsuit]\!]_Y : (\mathcal{P}Y)^n \to \mathcal{P}TY)_{Y \in \mathbf{Set}}$.

Satisfaction of modal operators (for $[\![\phi]\!]_{\mathcal{C}} = \{c \in X \mid \mathcal{C}, c \vDash \phi\}$):

$$[\![\heartsuit(A_1, \ldots, A_n)]\!]_{\mathcal{C}} = \xi^{-1} \circ [\![\heartsuit]\!]_X([\![A_1]\!]_{\mathcal{C}}, \ldots, [\![A_n]\!]_{\mathcal{C}})$$

K: $T = \mathcal{P}$, $[\![\Box]\!]_X(Y) = \{A \in \mathcal{P}(X) \mid A \subseteq Y\}$,
$[\![\Diamond]\!]_X(Y) = \{A \in \mathcal{P}(X) \mid A \cap Y \neq \varnothing\}$.

CK: $T = \mathcal{CK}$, $\mathcal{CK}(X) = \{f : \mathcal{P}(X) \to \mathcal{P}(X)\}$,
$[\![\Rightarrow]\!]_X(Y, Z) = \{f \in \mathcal{CK}(X) \mid f(Y) \subseteq Z\}$.

Deciding satisfiability

- A formula ϕ is satisfiable if there is a coalgebra $\mathcal{C} = (X, \xi)$ containing a state $x \in X$ s.t. $x \in [[\phi]]_{\mathcal{C}}$.
  Fact: ϕ is satisfiable iff $\neg\phi$ is not valid. Focus here: satisfiability.
- For rank-1 modal logics, validity may be decided coalgebraically in PSPACE [Schröder, Pattinson, 2009]; sequent calculus.
- EXPTIME if global assumptions are allowed; global caching algorithm [Goré, Kupke, Pattinson, 2010].
- EXPTIME if (flat) fixed point operators are allowed [Cirstea, Kupke, Pattinson, 2009; Schröder, Venema, 2010]; here: focusing global caching algorithm.

Tableau rules

$$(\neg\wedge)\quad \frac{\Gamma, \neg(\neg A \wedge \neg B)}{\Gamma, A \qquad \Gamma, B}$$

$$(\neg\neg)\quad \frac{\Gamma, \neg\neg A}{\Gamma, A} \qquad (\wedge)\quad \frac{\Gamma, (A \wedge B)}{\Gamma, A, B}$$

Modal rule(s), e.g.:

$$(\neg\Box_{\mathbf{K}})\quad \frac{\Gamma, \bigwedge_{i=1}^{n} \Box A_i, \neg\Box B}{\bigwedge_{i=1}^{n} A_i, \neg B} \qquad (\Rightarrow_{\mathbf{CK}})\quad \frac{\Gamma, \bigwedge_{i=1}^{n} (A_i \Rightarrow B_i), \neg(A_0 \Rightarrow B_0)}{\bigwedge_{i=1}^{n} A_{i-1} \not\leftrightarrow A_i \qquad \bigwedge_{i=1}^{n} B_i, \neg B_0}$$

Tableau rules

$$(\neg\wedge) \quad \frac{\Gamma, \neg(\neg A \wedge \neg B)}{\Gamma, A \qquad \Gamma, B}$$

$$(\neg\neg) \quad \frac{\Gamma, \neg\neg A}{\Gamma, A} \qquad (\wedge) \quad \frac{\Gamma, (A \wedge B)}{\Gamma, A, B}$$

Modal rule(s), e.g.:

$$(\neg\Box_{\mathbf{K}}) \quad \frac{\Gamma, \bigwedge_{i=1}^{n} \Box A_i, \neg \Box B}{\bigwedge_{i=1}^{n} A_i, \neg B} \qquad (\Rightarrow_{\mathbf{CK}}) \quad \frac{\Gamma, \bigwedge_{i=1}^{n} (A_i \Rightarrow B_i), \neg(A_0 \Rightarrow B_0)}{\bigwedge_{i=1}^{n} A_{i-1} \not\leftrightarrow A_i \qquad \bigwedge_{i=1}^{n} B_i, \neg B_0}$$

Tableau rules

$$(\neg\wedge)\quad \frac{\Gamma, \neg(\neg A \wedge \neg B)}{\Gamma, A \qquad \Gamma, B}$$

$$(\neg\neg)\quad \frac{\Gamma, \neg\neg A}{\Gamma, A} \qquad (\wedge)\quad \frac{\Gamma, (A \wedge B)}{\Gamma, A, B}$$

Modal rule(s), e.g.:

$$(\neg\Box_{\mathbf{K}})\quad \frac{\Gamma, \bigwedge_{i=1}^{n} \Box A_i, \neg \Box B}{\bigwedge_{i=1}^{n} A_i, \neg B} \qquad (\Rightarrow_{\mathbf{CK}})\quad \frac{\Gamma, \bigwedge_{i=1}^{n} (A_i \Rightarrow B_i), \neg(A_0 \Rightarrow B_0)}{\bigwedge_{i=1}^{n} A_{i-1} \not\leftrightarrow A_i \qquad \bigwedge_{i=1}^{n} B_i, \neg B_0}$$

Tableau Algorithm

Tableau system

Apply all tableau rules, depth first search.

A set of formulae Γ is successful if for all ($\forall$) rule applications to Γ there is ($\exists$) a premise which is successful.

+ Generic and realizes PSPACE proof search.
– Not particularly open to optimisations, per se not capable of treating global assumptions. Thus:

Global Caching

Construct a two-kinded *proof graph* which employs tableau rules.

Propagate provability status through the graph.

Tableau Algorithm

Tableau system

Apply all tableau rules, depth first search.
A set of formulae Γ is successful if for all ($\forall$) rule applications to Γ there is ($\exists$) a premise which is successful.

+ Generic and realizes PSPACE proof search.
– Not particularly open to optimisations, per se not capable of treating global assumptions. Thus:

Global Caching

Construct a two-kinded *proof graph* which employs tableau rules.
Propagate provability status through the graph.

Tableau Algorithm

Tableau system

Apply all tableau rules, depth first search.
A set of formulae Γ is successful if for all ($\forall$) rule applications to Γ there is ($\exists$) a premise which is successful.

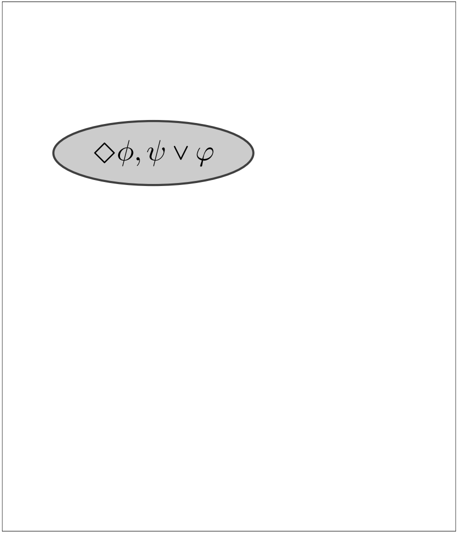
+ Generic and realizes PSPACE proof search.
– Not particularly open to optimisations, per se not capable of treating global assumptions. Thus:

Global Caching

Construct a two-kinded *proof graph* which employs tableau rules.
Propagate provability status through the graph.

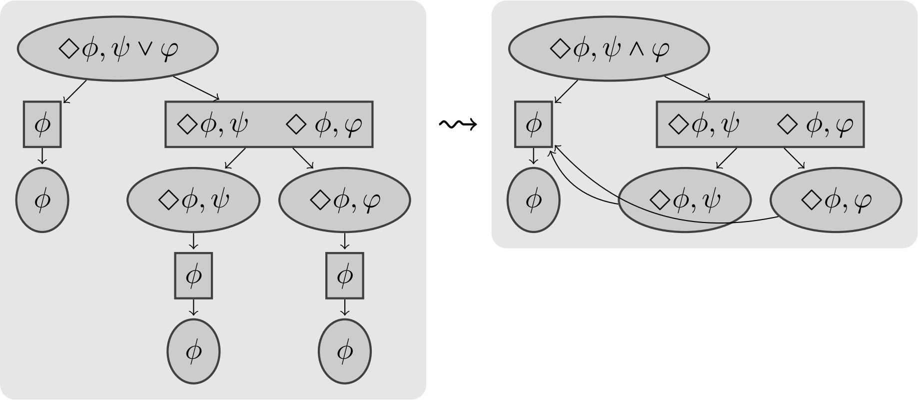
Expansion

E.g.:



Caching

E.g.:



$\Diamond\phi, \psi \vee \varphi$
ϕ
$\Diamond\phi, \psi \quad \Diamond\ \phi, \varphi$
ϕ
$\Diamond\phi, \psi$
$\Diamond\phi, \varphi$
ϕ
ϕ
ϕ
ϕ
$\rightsquigarrow$
$\Diamond\phi, \psi \wedge \varphi$
ϕ
$\Diamond\phi, \psi \quad \Diamond\ \phi, \varphi$
ϕ
$\Diamond\phi, \psi$
$\Diamond\phi, \varphi$

Propagation

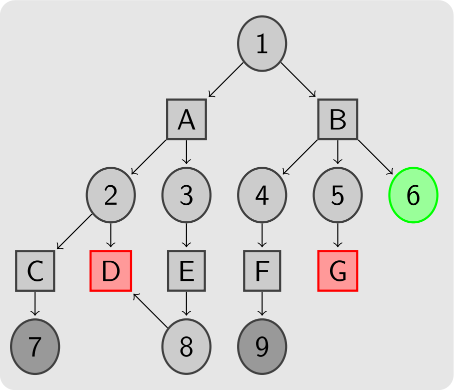
$$E' = E \cup \mu(R^{\forall\exists}{}_E), \quad R^{\forall\exists}{}_Y(X) = \{\Gamma \mid \forall\Sigma.\Gamma \to \Sigma.\exists\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$A' = A \cup \mu(R^{\exists\forall}{}_A), \quad R^{\exists\forall}{}_Y(X) = \{\Gamma \mid \exists\Sigma.\Gamma \to \Sigma.\forall\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$

E.g.:

Propagation

$$E' = E \cup \mu(R^{\forall\exists}{}_E), \quad R^{\forall\exists}{}_Y(X) = \{\Gamma \mid \forall\Sigma.\Gamma \to \Sigma.\exists\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$A' = A \cup \mu(R^{\exists\forall}{}_A), \quad R^{\exists\forall}{}_Y(X) = \{\Gamma \mid \exists\Sigma.\Gamma \to \Sigma.\forall\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$

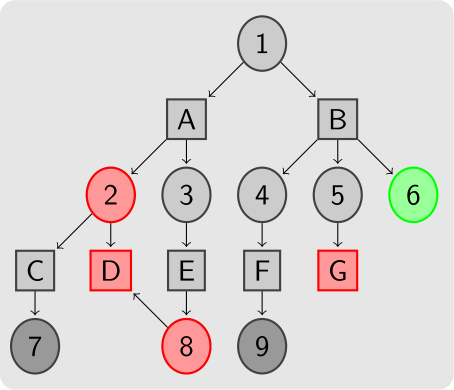
E.g.:



Propagation

$$E' = E \cup \mu(R^{\forall\exists}{}_E), \quad R^{\forall\exists}{}_Y(X) = \{\Gamma \mid \forall \Sigma.\Gamma \to \Sigma.\exists \Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$A' = A \cup \mu(R^{\exists\forall}{}_A), \quad R^{\exists\forall}{}_Y(X) = \{\Gamma \mid \exists \Sigma.\Gamma \to \Sigma.\forall \Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$

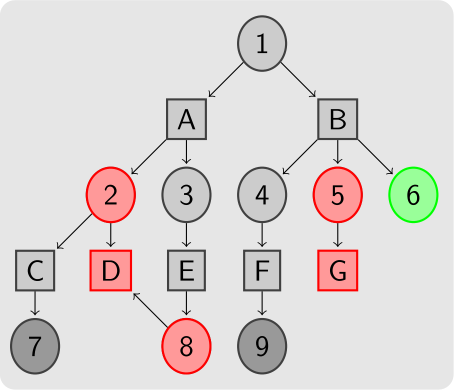
E.g.:



Propagation

$$E' = E \cup \mu(R^{\forall\exists}{}_E), \quad R^{\forall\exists}{}_Y(X) = \{\Gamma \mid \forall\Sigma.\Gamma \to \Sigma.\exists\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$A' = A \cup \mu(R^{\exists\forall}{}_A), \quad R^{\exists\forall}{}_Y(X) = \{\Gamma \mid \exists\Sigma.\Gamma \to \Sigma.\forall\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$

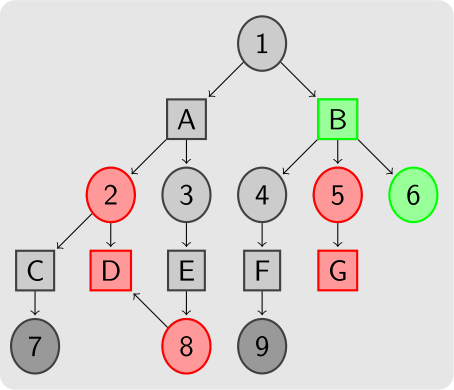
E.g.:



Propagation

$$E' = E \cup \mu(R^{\forall\exists}{}_E), \quad R^{\forall\exists}{}_Y(X) = \{\Gamma \mid \forall\Sigma.\Gamma \to \Sigma.\exists\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$A' = A \cup \mu(R^{\exists\forall}{}_A), \quad R^{\exists\forall}{}_Y(X) = \{\Gamma \mid \exists\Sigma.\Gamma \to \Sigma.\forall\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$

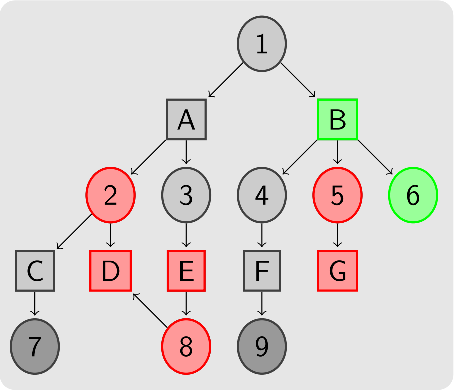
E.g.:



Propagation

$$E' = E \cup \mu(R^{\forall\exists}{}_E), \quad R^{\forall\exists}{}_Y(X) = \{\Gamma \mid \forall\Sigma.\Gamma \to \Sigma.\exists\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$A' = A \cup \mu(R^{\exists\forall}{}_A), \quad R^{\exists\forall}{}_Y(X) = \{\Gamma \mid \exists\Sigma.\Gamma \to \Sigma.\forall\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$

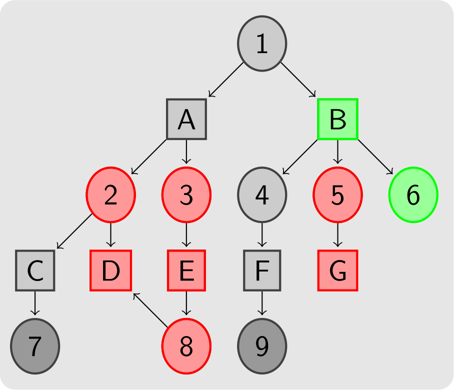
E.g.:



Propagation

$$E' = E \cup \mu(R^{\forall\exists}{}_E), \quad R^{\forall\exists}{}_Y(X) = \{\Gamma \mid \forall\Sigma.\Gamma \to \Sigma.\exists\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$A' = A \cup \mu(R^{\exists\forall}{}_A), \quad R^{\exists\forall}{}_Y(X) = \{\Gamma \mid \exists\Sigma.\Gamma \to \Sigma.\forall\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$

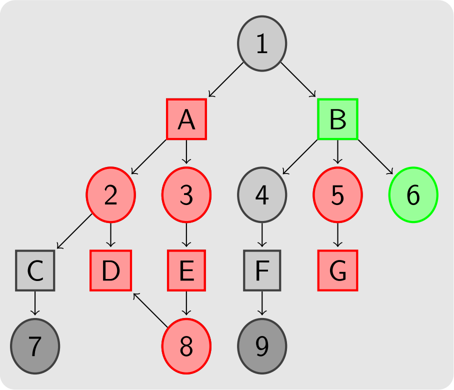
E.g.:



Propagation

$$E' = E \cup \mu(R^{\forall\exists}{}_E), \quad R^{\forall\exists}{}_Y(X) = \{\Gamma \mid \forall\Sigma.\Gamma \to \Sigma.\exists\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$A' = A \cup \mu(R^{\exists\forall}{}_A), \quad R^{\exists\forall}{}_Y(X) = \{\Gamma \mid \exists\Sigma.\Gamma \to \Sigma.\forall\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$

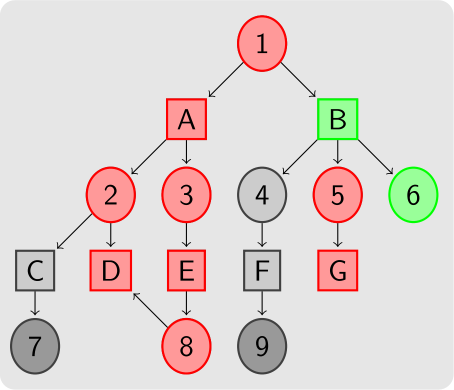
E.g.:



Propagation

$$E' = E \cup \mu(R^{\forall\exists}{}_E), \quad R^{\forall\exists}{}_Y(X) = \{\Gamma \mid \forall\Sigma.\Gamma \to \Sigma.\exists\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$A' = A \cup \mu(R^{\exists\forall}{}_A), \quad R^{\exists\forall}{}_Y(X) = \{\Gamma \mid \exists\Sigma.\Gamma \to \Sigma.\forall\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$

E.g.:



Global Assumptions

Extend *modal* rules by set of *global assumptions* $\Delta \subseteq \mathcal{CML}(\Lambda)$:

$$\boxed{\frac{\Gamma}{\Gamma_0, \dots, \Gamma_n}} \rightsquigarrow \boxed{\frac{\Gamma}{(\Delta \cup \Gamma_0), \dots, (\Delta \cup \Gamma_n)}}$$

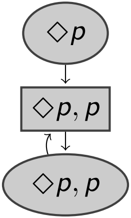
E.g. $\Delta = \{\Diamond p\}$:

Global Assumptions

Extend *modal* rules by set of *global assumptions* $\Delta \subseteq \mathcal{CML}(\Lambda)$:

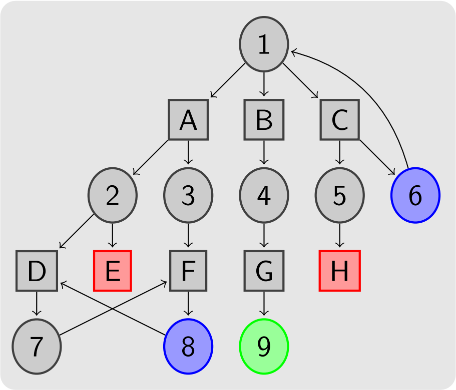
$$\boxed{\frac{\Gamma}{\Gamma_0, \ldots, \Gamma_n}} \rightsquigarrow \boxed{\frac{\Gamma}{(\Delta \cup \Gamma_0), \ldots, (\Delta \cup \Gamma_n)}}$$

E.g. $\Delta = \{\Diamond p\}$:



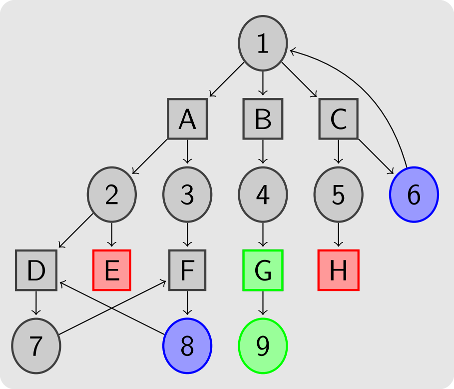
gfp-propagation

$$E' = E \cup \mu(R^{\forall\exists}{}_E), \quad R^{\forall\exists}{}_Y(X) = \{\Gamma \mid \forall\Sigma.\Gamma \to \Sigma.\exists\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$A' = A \cup \mu(R^{\exists\forall}{}_A), \quad R^{\exists\forall}{}_Y(X) = \{\Gamma \mid \exists\Sigma.\Gamma \to \Sigma.\forall\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$G' = G \cup \nu(R^{\forall\exists}{}_E)$$



gfp-propagation

$$E' = E \cup \mu(R^{\forall\exists}{}_E), \quad R^{\forall\exists}{}_Y(X) = \{\Gamma \mid \forall\Sigma.\Gamma \to \Sigma.\exists\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$A' = A \cup \mu(R^{\exists\forall}{}_A), \quad R^{\exists\forall}{}_Y(X) = \{\Gamma \mid \exists\Sigma.\Gamma \to \Sigma.\forall\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$G' = G \cup \nu(R^{\forall\exists}{}_E)$$

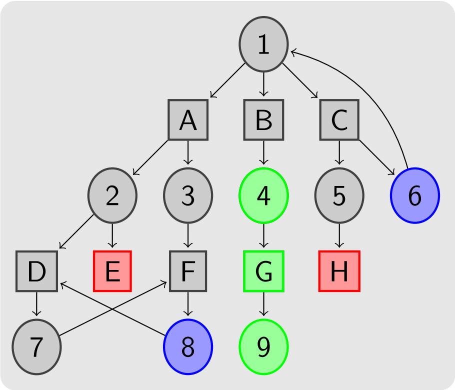


gfp-propagation

$$E' = E \cup \mu(R^{\forall\exists}{}_E), \quad R^{\forall\exists}{}_Y(X) = \{\Gamma \mid \forall\Sigma.\Gamma \to \Sigma.\exists\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$

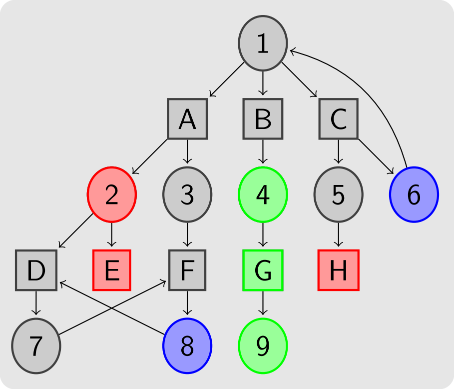
$$A' = A \cup \mu(R^{\exists\forall}{}_A), \quad R^{\exists\forall}{}_Y(X) = \{\Gamma \mid \exists\Sigma.\Gamma \to \Sigma.\forall\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$

$$G' = G \cup \nu(R^{\forall\exists}{}_E)$$



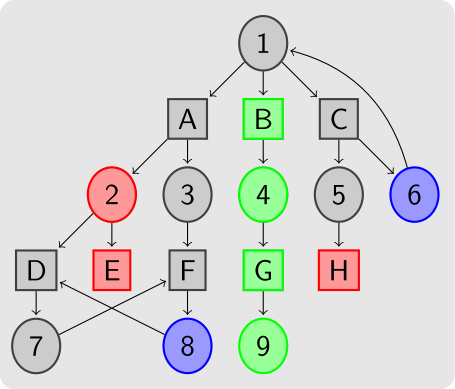
gfp-propagation

$$E' = E \cup \mu(R^{\forall\exists}{}_E), \quad R^{\forall\exists}{}_Y(X) = \{\Gamma \mid \forall\Sigma.\Gamma \to \Sigma.\exists\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$A' = A \cup \mu(R^{\exists\forall}{}_A), \quad R^{\exists\forall}{}_Y(X) = \{\Gamma \mid \exists\Sigma.\Gamma \to \Sigma.\forall\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$G' = G \cup \nu(R^{\forall\exists}{}_E)$$



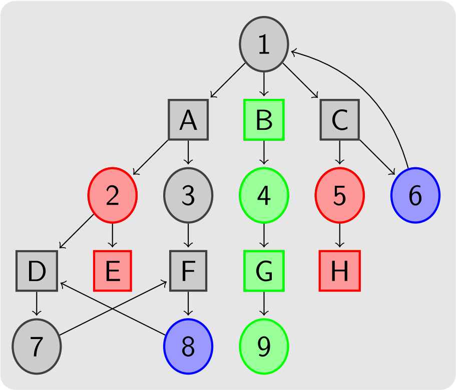
gfp-propagation

$$E' = E \cup \mu(R^{\forall\exists}{}_E), \quad R^{\forall\exists}{}_Y(X) = \{\Gamma \mid \forall \Sigma.\Gamma \to \Sigma.\exists \Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$A' = A \cup \mu(R^{\exists\forall}{}_A), \quad R^{\exists\forall}{}_Y(X) = \{\Gamma \mid \exists \Sigma.\Gamma \to \Sigma.\forall \Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$G' = G \cup \nu(R^{\forall\exists}{}_E)$$



gfp-propagation

$$E' = E \cup \mu(R^{\forall\exists}{}_E), \quad R^{\forall\exists}{}_Y(X) = \{\Gamma \mid \forall \Sigma.\Gamma \to \Sigma.\exists \Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$A' = A \cup \mu(R^{\exists\forall}{}_A), \quad R^{\exists\forall}{}_Y(X) = \{\Gamma \mid \exists \Sigma.\Gamma \to \Sigma.\forall \Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$G' = G \cup \nu(R^{\forall\exists}{}_E)$$

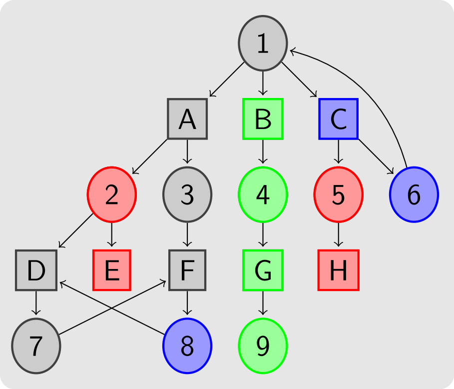


gfp-propagation

$$E' = E \cup \mu({R^{\forall\exists}}_E), \quad {R^{\forall\exists}}_Y(X) = \{\Gamma \mid \forall \Sigma.\Gamma \to \Sigma.\exists \Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$

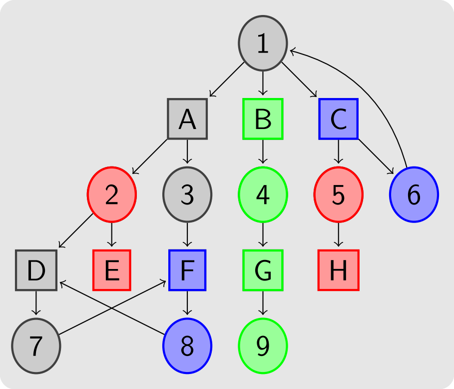
$$A' = A \cup \mu({R^{\exists\forall}}_A), \quad {R^{\exists\forall}}_Y(X) = \{\Gamma \mid \exists \Sigma.\Gamma \to \Sigma.\forall \Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$

$$G' = G \cup \nu({R^{\forall\exists}}_E)$$



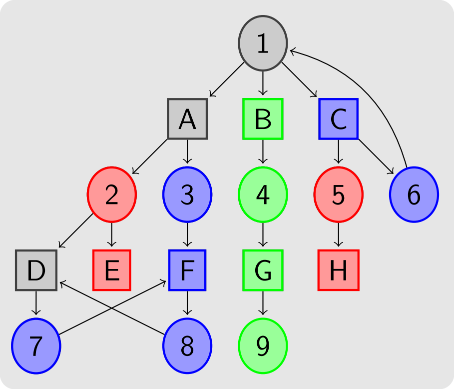
gfp-propagation

$$E' = E \cup \mu(R^{\forall\exists}{}_E), \quad R^{\forall\exists}{}_Y(X) = \{\Gamma \mid \forall \Sigma.\Gamma \to \Sigma.\exists \Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$A' = A \cup \mu(R^{\exists\forall}{}_A), \quad R^{\exists\forall}{}_Y(X) = \{\Gamma \mid \exists \Sigma.\Gamma \to \Sigma.\forall \Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$G' = G \cup \nu(R^{\forall\exists}{}_E)$$



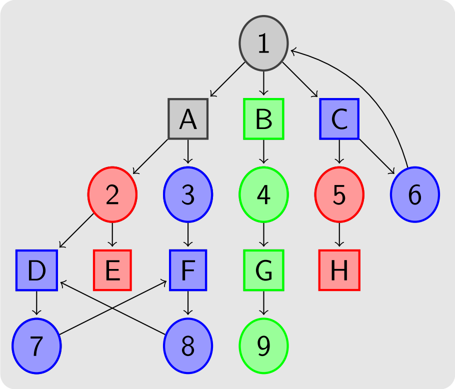
gfp-propagation

$$E' = E \cup \mu({R^{\forall\exists}}_E), \quad {R^{\forall\exists}}_Y(X) = \{\Gamma \mid \forall\Sigma.\Gamma \to \Sigma.\exists\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$A' = A \cup \mu({R^{\exists\forall}}_A), \quad {R^{\exists\forall}}_Y(X) = \{\Gamma \mid \exists\Sigma.\Gamma \to \Sigma.\forall\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$G' = G \cup \nu({R^{\forall\exists}}_E)$$



gfp-propagation

$$E' = E \cup \mu({R^{\forall\exists}}_E), \quad {R^{\forall\exists}}_Y(X) = \{\Gamma \mid \forall\Sigma.\Gamma \to \Sigma.\exists\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$A' = A \cup \mu({R^{\exists\forall}}_A), \quad {R^{\exists\forall}}_Y(X) = \{\Gamma \mid \exists\Sigma.\Gamma \to \Sigma.\forall\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$$
$$G' = G \cup \nu({R^{\forall\exists}}_E)$$

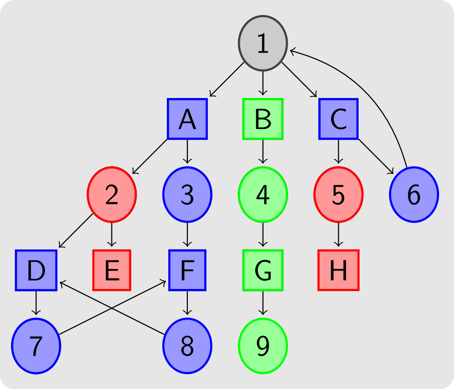


gfp-propagation

$E' = E \cup \mu(R^{\forall\exists}{}_E)$, $\quad R^{\forall\exists}{}_Y(X) = \{\Gamma \mid \forall\Sigma.\Gamma \to \Sigma.\exists\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$

$A' = A \cup \mu(R^{\exists\forall}{}_A)$, $\quad R^{\exists\forall}{}_Y(X) = \{\Gamma \mid \exists\Sigma.\Gamma \to \Sigma.\forall\Gamma' \in \Sigma.\Gamma' \in X \cup Y\}$

$G' = G \cup \nu(R^{\forall\exists}{}_E)$

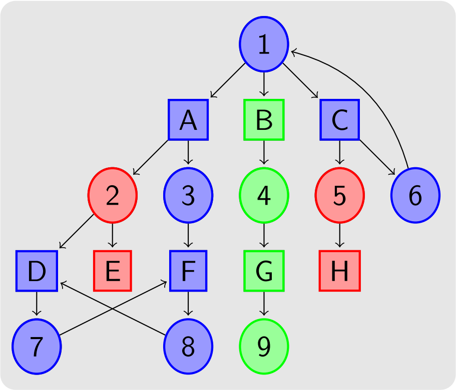


gfp-propagation

$$E' = E \cup \mu(R^{\forall\exists}{}_E), \quad R^{\forall\exists}{}_Y(X) = \{\Gamma \mid \forall \Sigma. \Gamma \to \Sigma. \exists \Gamma' \in \Sigma. \Gamma' \in X \cup Y\}$$

$$A' = A \cup \mu(R^{\exists\forall}{}_A), \quad R^{\exists\forall}{}_Y(X) = \{\Gamma \mid \exists \Sigma. \Gamma \to \Sigma. \forall \Gamma' \in \Sigma. \Gamma' \in X \cup Y\}$$

$$G' = G \cup \nu(R^{\forall\exists}{}_E)$$



Efficient gfp-propagation

Let C be a set of *circular nodes* and let $R_Y^{Q\overline{Q}} \in \{R_Y^{\forall\exists}, R_Y^{\exists\forall}\}$.

$$\Theta_C^0(R_Y) = \mu(R_Y)$$
$$\Theta_C^{n+1}(R_Y) = \mu(R_{Y\cup ccg(R_Y, C_{n+1}, \Theta_C^n(R_Y))})$$

where $ccg(R_Y, C_{n+1}, \Theta_C^n(R_Y))$ computes *closed circular graphs* over C_{n+1} and $\Theta_C^n(R_y)$.

Proposition

Let n be the number of strongly connected components in a proof graph. Then

$$\Theta_C^n(R_Y) = \nu(R_Y).$$

Efficient gfp-propagation

Let C be a set of *circular nodes* and let $R_Y^{Q\overline{Q}} \in \{R_Y^{\forall\exists}, R_Y^{\exists\forall}\}$.

$$\Theta_C^0(R_Y) = \mu(R_Y)$$
$$\Theta_C^{n+1}(R_Y) = \mu(R_{Y \cup ccg(R_Y, C_{n+1}, \Theta_C^n(R_Y))})$$

where $ccg(R_Y, C_{n+1}, \Theta_C^n(R_Y))$ computes *closed circular graphs* over C_{n+1} and $\Theta_C^n(R_y)$.

Proposition

Let n be the number of strongly connected components in a proof graph. Then

$$\Theta_C^n(R_Y) = \nu(R_Y).$$

Flat fixed point logics

- Depth-1 fixed point schemes
  $\mathcal{FPS} \ni \gamma_1, \gamma_2 ::= v \mid x \mid \gamma_1 \wedge \gamma_2 \mid \gamma_1 \vee \gamma_2 \mid \heartsuit\gamma_1.$
- Add flat fixed point operators $\nu\gamma\psi$, $\mu\gamma\psi$ with $[[\nu\gamma\psi]]_{\mathcal{C}} = \nu[[\gamma_\psi]]$ where $[[\gamma_\psi]](Y) = [[\ \gamma[x/\psi, v/\phi]\ ]]_{\mathcal{C}}$ with $[[\phi]]_{\mathcal{C}} = Y$.

Examples

- PLTL: $\mathcal{I}(X) = X$, $\mathcal{I}$-coalgebras are ω-sequences.
  Predicate lifting $[[\Box]]_C(X) = X$, $\psi_1 U \psi_2 \hat{=} \mu(x_2 \vee (x_1 \wedge \Box v))(\psi_1, \psi_2)$.
- CTL: For base logic **K**, $A\psi_1 U \psi_2 \hat{=} \mu(x_2 \vee (x_1 \wedge \Box v)(\psi_1, \psi_2))$.
- ATL: $\mathcal{G}(X) = \{(S_1, \ldots, S_n, f) \mid \emptyset \neq S_i \in \mathbf{Set}, f : \prod_{i \in N} S_i \to X\}$.
  $\mathcal{G}$-coalgebras are *game frames*. Predicate lifting
  $$[[\mathrm{D}]]_C(X) = \{(S_1, \ldots, S_n, f) \in \mathcal{G}(C) \mid \exists s_D \in S_D. \forall s_{\overline{D}} \in S_{\overline{D}}.\ f(s_D, s_{\overline{D}}) \in X\}$$
  Define $\langle\langle D \rangle\rangle \psi_1 \mathcal{U} \psi_2 \hat{=} \mu(x_2 \vee (x_1 \wedge [D]v)(\psi_1, \psi_2)$.

Flat fixed point logics

- Depth-1 fixed point schemes
  $\mathcal{FPS} \ni \gamma_1, \gamma_2 ::= v \mid x \mid \gamma_1 \wedge \gamma_2 \mid \gamma_1 \vee \gamma_2 \mid \heartsuit\gamma_1.$
- Add flat fixed point operators $\nu\gamma\psi$, $\mu\gamma\psi$ with $[[\nu\gamma\psi]]_{\mathcal{C}} = \nu[[\gamma_\psi]]$ where $[[\gamma_\psi]](Y) = [[\ \gamma[x/\psi, v/\phi]\ ]]_{\mathcal{C}}$ with $[[\phi]]_{\mathcal{C}} = Y$.

Examples

- PLTL: $\mathcal{I}(X) = X$, $\mathcal{I}$-coalgebras are ω-sequences.
  Predicate lifting $[[\Box]]_C(X) = X$, $\psi_1 U \psi_2 \mathrel{\hat{=}} \mu(x_2 \vee (x_1 \wedge \Box v))(\psi_1, \psi_2)$.
- CTL: For base logic **K**, $A\psi_1 U \psi_2 \mathrel{\hat{=}} \mu(x_2 \vee (x_1 \wedge \Box v)(\psi_1, \psi_2))$.
- ATL: $\mathcal{G}(X) = \{(S_1, \dots, S_n, f) \mid \emptyset \neq S_i \in \mathbf{Set}, f : \prod_{i \in N} S_i \to X\}$.
  $\mathcal{G}$-coalgebras are *game frames*. Predicate lifting
  $$[[[D]]]_C(X) = \{(S_1, \dots, S_n, f) \in \mathcal{G}(C) \mid \exists s_D \in S_D. \forall s_{\overline{D}} \in S_{\overline{D}}. f(s_D, s_{\overline{D}}) \in X\}$$
  Define $\langle\langle D \rangle\rangle \psi_1 \mathcal{U} \psi_2 \mathrel{\hat{=}} \mu(x_2 \vee (x_1 \wedge [D]v)(\psi_1, \psi_2)$.

How to decide circular subgraphs?

$$\dfrac{EpU(q \wedge \overline{q})}{(\vee)\,\dfrac{(q \wedge \overline{q}) \vee (p \wedge \Diamond EpU(q \wedge \overline{q}))}{\dfrac{(q \wedge \overline{q})}{\dfrac{q, \overline{q}}{\text{↯}}} \quad \dfrac{(p \wedge \Diamond EpU(q \wedge \overline{q}))}{(\Diamond)\,\dfrac{p, \Diamond EpU(q \wedge \overline{q})}{\Gamma := EpU(q \wedge \overline{q})}}}}$$

$$\dfrac{AF\phi, \Diamond EGAF\phi}{(\vee)\,\dfrac{\phi \vee \Box AF\phi, \Diamond EGAF\phi}{(\Diamond)\,\dfrac{\phi, \Diamond EGAF\phi}{\dfrac{EGAF\phi}{\Gamma_1 := AF\phi, \Diamond EGAF\phi}} \quad (\Diamond)\,\dfrac{\Box AF\phi, \Diamond EGAF\phi}{\dfrac{AF\phi, EGAF\phi}{\Gamma_2 := AF\phi, \Diamond EGAF\phi}}}}$$

Treating eventualities

Solution: Focusing [Brünnler, Lange, 2008].

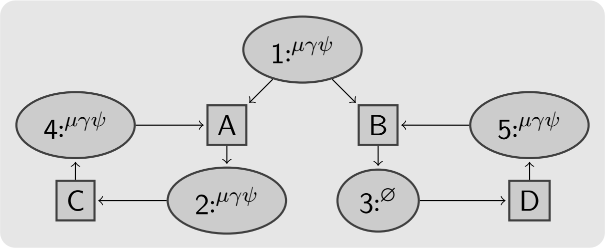
Global caching algorithm over focused pre-states $\Gamma :^{\mu\gamma\psi}$.
Focus-respecting unfolding of lfps via technical construction Σ_γ^ψ, e.g.
$\Sigma_{x_2 \vee (x_1 \wedge \Box v)}^{ApUq} = \{\{p, q\} :^{\varnothing}, \{q, \Box ApUq\} :^{ApUq}\}$

$$(\nu\text{UNF}) \quad \frac{\Gamma, \nu\gamma\psi :^{foc}}{\Gamma, \gamma[x/\psi, v/\nu\gamma\psi] :^{foc}} \qquad (\text{FOC}) \quad \frac{\Gamma, \mu\gamma\psi :^{\varnothing}}{\Gamma, \mu\gamma\psi :^{\mu\gamma\psi}}$$

$$(\mu\text{UNF}) \quad \frac{\Gamma, \mu\gamma\psi :^{foc}}{\Gamma, \gamma[x/\psi, v/\mu\gamma\psi] :^{foc}} \; \varnothing \neq foc \neq \mu\gamma\psi \qquad (\mu\text{UNF}^f) \quad \frac{\Gamma, \mu\gamma\psi :^{\mu\gamma\psi}}{\Gamma, \Sigma_\gamma^{\mu\gamma\psi}}$$

Focused propagation

$$E' = E \cup \Theta^n_{C_\varnothing}(R_E^{\forall\exists}),\quad \text{unfocused circular nodes } C_\varnothing$$
$$A' = A \cup \Theta^n_{C_\alpha}(R_A^{\exists\forall}),\quad \text{focused circular nodes } C_\alpha$$

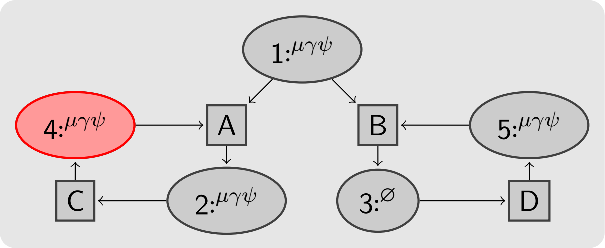


Proposition

The focusing global caching algorithm correctly decides the validity problem of depth-1 flat coalgebraic fixed point logics.

Focused propagation

$$E' = E \cup \Theta^n_{C_\varnothing}(R_E^{\forall\exists}),\quad \text{unfocused circular nodes } C_\varnothing$$
$$A' = A \cup \Theta^n_{C_\alpha}(R_A^{\exists\forall}),\quad \text{focused circular nodes } C_\alpha$$

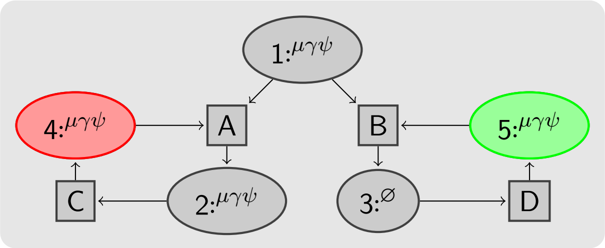


Proposition

The focusing global caching algorithm correctly decides the validity problem of depth-1 flat coalgebraic fixed point logics.

Focused propagation

$$E' = E \cup \Theta^{n}_{C_{\varnothing}}(R^{\forall\exists}_{E}), \quad \text{unfocused circular nodes } C_{\varnothing}$$
$$A' = A \cup \Theta^{n}_{C_{\alpha}}(R^{\exists\forall}_{A}), \quad \text{focused circular nodes } C_{\alpha}$$



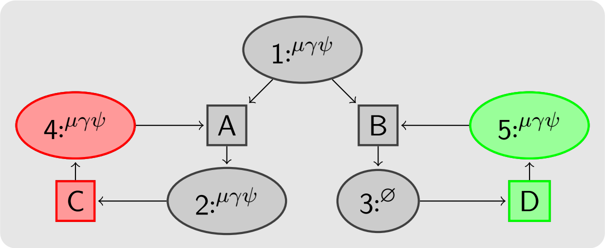
Proposition

The focusing global caching algorithm correctly decides the validity problem of depth-1 flat coalgebraic fixed point logics.

Focused propagation

$E' = E \cup \Theta^n_{C_\varnothing}(R_E^{\forall\exists})$, unfocused circular nodes $C_\varnothing$

$A' = A \cup \Theta^n_{C_\alpha}(R_A^{\exists\forall})$, focused circular nodes C_α

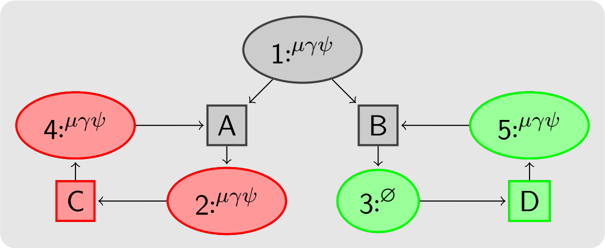


Proposition

The focusing global caching algorithm correctly decides the validity problem of depth-1 flat coalgebraic fixed point logics.

Focused propagation

$$E' = E \cup \Theta^n_{C_\varnothing}(R_E^{\forall\exists}), \quad \text{unfocused circular nodes } C_\varnothing$$
$$A' = A \cup \Theta^n_{C_\alpha}(R_A^{\exists\forall}), \quad \text{focused circular nodes } C_\alpha$$

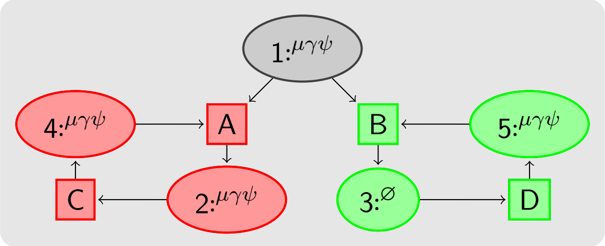


Proposition

The focusing global caching algorithm correctly decides the validity problem of depth-1 flat coalgebraic fixed point logics.

Focused propagation

$E' = E \cup \Theta^n_{C_\varnothing}(R^{\forall\exists}_E)$, unfocused circular nodes $C_\varnothing$
$A' = A \cup \Theta^n_{C_\alpha}(R^{\exists\forall}_A)$, focused circular nodes C_α

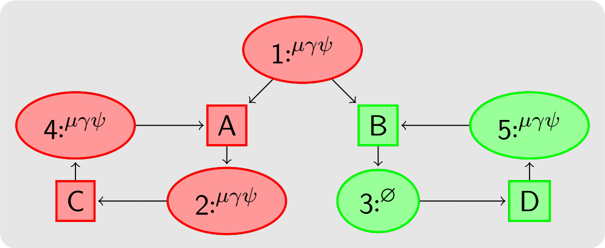


Proposition

The focusing global caching algorithm correctly decides the validity problem of depth-1 flat coalgebraic fixed point logics.

Focused propagation

$$E' = E \cup \Theta^n_{C_\varnothing}(R_E^{\forall\exists}), \quad \text{unfocused circular nodes } C_\varnothing$$
$$A' = A \cup \Theta^n_{C_\alpha}(R_A^{\exists\forall}), \quad \text{focused circular nodes } C_\alpha$$

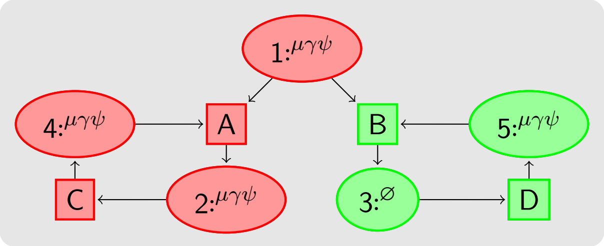


Proposition

The focusing global caching algorithm correctly decides the validity problem of depth-1 flat coalgebraic fixed point logics.

Focused propagation

$$E' = E \cup \Theta^n_{C_\varnothing}(R_E^{\forall\exists}), \quad \text{unfocused circular nodes } C_\varnothing$$
$$A' = A \cup \Theta^n_{C_\alpha}(R_A^{\exists\forall}), \quad \text{focused circular nodes } C_\alpha$$



Proposition

The focusing global caching algorithm correctly decides the validity problem of depth-1 flat coalgebraic fixed point logics.

Model construction

Given $\mathcal{C} = (X, \xi)$, a pair $(x, \xi(x))$ is *coherent* if

$$\heartsuit\phi \in x \text{ iff } \xi(x) \in [\![\heartsuit]\!]_X [\![\phi]\!]_{\mathcal{C}}.$$

Existence Lemma

Coherent coalgebra $\mathcal{C} = (X, \xi)$ exists over *pre-tableaux* (which may be extracted from the proof graph constructed by the algorithm).

– Coherence does not ensure satisfaction of eventualities $\mu\gamma\psi$.

Truth Lemma

If the algorithm is successful, $\forall \mu\gamma\psi. [\![\widehat{\mu\gamma\psi}]\!]_{\mathcal{C}} \subseteq \mu[\![\widehat{\gamma_\psi}]\!]_{\mathcal{C}}$.

– This yields a model, the crucial part for correctness, but it is a big model.

Small model lemma

Fisher-Ladner closure $\mathcal{FL}(\phi)$ is the least set contaning ϕ and closed under negation, subformulae and unfolding of fixed points.

Set of all *maximal consistent sets* $MCS(S)$, $x \in MCS(S)$ if x is consistent and $S \ni \psi \notin x$ implies that $x \cup \{\psi\}$ is inconsistent.

Fact

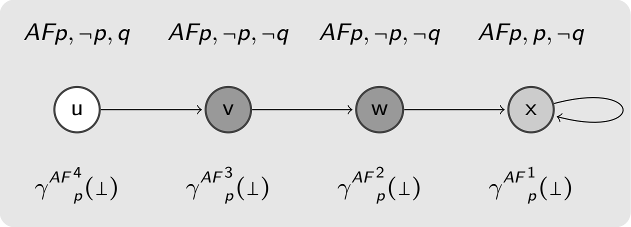
$|MCS(\mathcal{FL}(\phi))| \leq 2^{|\phi|_m}$.

Annotate the model $\mathcal{C} = (X, \xi)$ from Existence Lemma with unfoldings s.t.

$$\gamma_\psi^n(\bot) \in x \text{ iff } \mathcal{C}, x \vDash \gamma_\psi^n(\bot).$$

The reduction $r : Y \to MCS(\mathcal{FL}(\phi))$ is defined as $r(y) = y \cap \mathcal{FL}(\phi)$ (Tr follows).

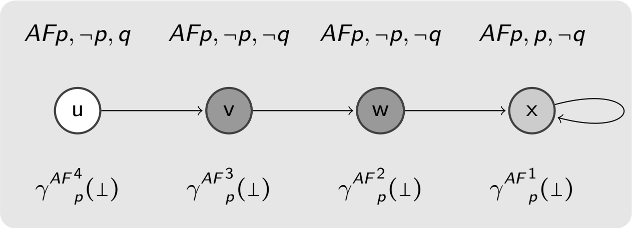
Interior states



State x interior if there is $\mu\gamma\psi \in x$ and $m > 0$ s.t. $\mathcal{C}, x \models \overline{\gamma}^m_\psi(\wedge r(x))$.

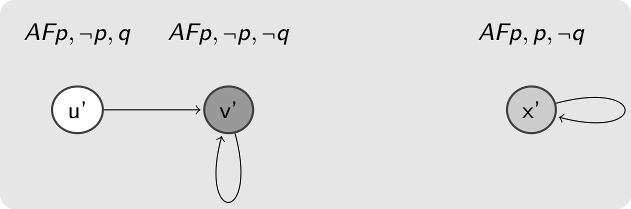
E.g. $\overline{\gamma^{AF}}^1_p(\wedge(v)) = \neg p \wedge \Diamond(\wedge r(v))$, $r(v) = r(w) = AFp, \neg p, \neg q$

Interior states

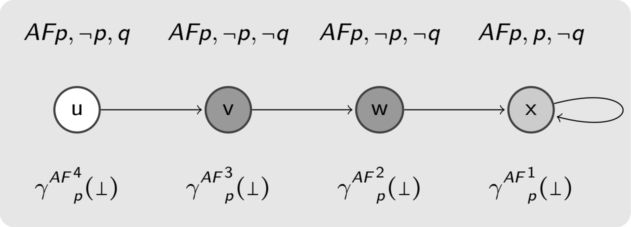


State x interior if there is $\mu\gamma\psi \in x$ and $m > 0$ s.t. $\mathcal{C}, x \models \overline{\gamma}^{m}_{\overline{\psi}}(\wedge r(x))$.

E.g. $\overline{\gamma^{AF}}^{1}_{\overline{p}}(\wedge(v)) = \neg p \wedge \Diamond(\wedge r(v))$, $r(v) = r(w) = AFp, \neg p, \neg q$

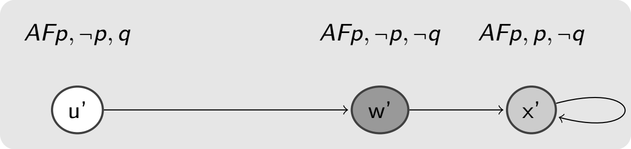


Interior states, ctd.

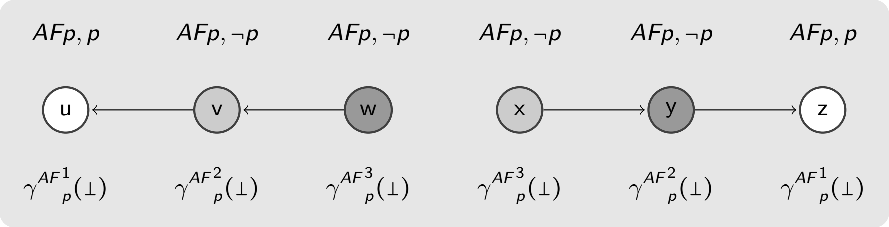


State x interior if there is $\mu\gamma\psi \in x$ and $m > 0$ s.t. $\mathcal{C}, x \models \overline{\gamma}^{m}_{\psi}(\wedge r(x))$.

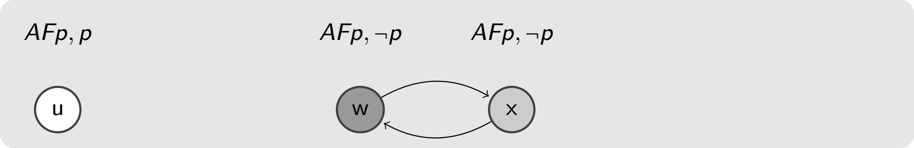
E.g. $\overline{\gamma^{AF}}^{1}_{\overline{p}}(\wedge(v)) = \neg p \wedge \Diamond(\wedge r(v))$, $r(v) = r(w) = AFp, \neg p, \neg q$



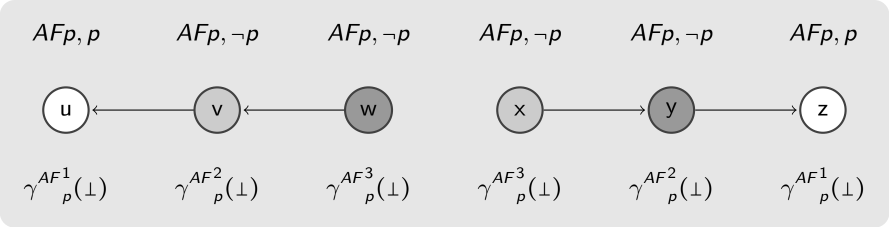
Minimal states



State x is minimal if there is no state y with $r(y) = r(x)$ s.t. y has a lesser *eventual* degree than x.



Minimal states, ctd.



Final links

$\mathcal{C} = (X, \xi)$ is apt (w.r.t. $\mathcal{D} = (Y, \zeta)$) if $r(y) = x$ and $Tr(\zeta(y)) = \xi(x)$ imply that y is minimal (and interior).

Lemma

Let ϕ be satisfiable. Then there is an apt coalgebra $\mathcal{C} = (X, \xi)$ over $X = MCS(\mathcal{FL}(\phi))$.

Lemma

The reduction $r : Y \to X$ preserves coherence. If $\mathcal{C} = (X, \xi)$ is apt w.r.t. $\mathcal{D} = (Y, \zeta)$, r also preserves satisfaction of $\gamma_\psi^n(\bot)$.

Together with Lindenbaum: for each $x \in X$, $\mu\gamma\psi \in x$ there is an n s.t. $\mathcal{C}, x \models \gamma_\psi^n(\bot)$, i.e. each apt $\mathcal{C} = (X, \xi)$ is a model.

Small Model Lemma

Let ϕ be a *satisfiable* depth-1 fixed point formula. Then there is a model $\mathcal{C} = (X, \xi)$ over $X = MCS(\mathcal{FL}(\phi))$ (i.e. of size at most $2^{|\phi|_m}$).

Final links

$\mathcal{C} = (X, \xi)$ is apt (w.r.t. $\mathcal{D} = (Y, \zeta)$) if $r(y) = x$ and $Tr(\zeta(y)) = \xi(x)$ imply that y is minimal (and interior).

Lemma

Let ϕ be satisfiable. Then there is an apt coalgebra $\mathcal{C} = (X, \xi)$ over $X = MCS(\mathcal{FL}(\phi))$.

Lemma

The reduction $r : Y \to X$ preserves coherence. If $\mathcal{C} = (X, \xi)$ is apt w.r.t. $\mathcal{D} = (Y, \zeta)$, r also preserves satisfaction of $\gamma_{\psi}^{n}(\bot)$.

Together with Lindenbaum: for each $x \in X$, $\mu\gamma\psi \in x$ there is an n s.t. $\mathcal{C}, x \models \gamma_{\psi}^{n}(\bot)$, i.e. each apt $\mathcal{C} = (X, \xi)$ is a model.

Small Model Lemma

Let ϕ be a *satisfiable* depth-1 fixed point formula. Then there is a model $\mathcal{C} = (X, \xi)$ over $X = MCS(\mathcal{FL}(\phi))$ (i.e. of size at most $2^{|\phi|_m}$).

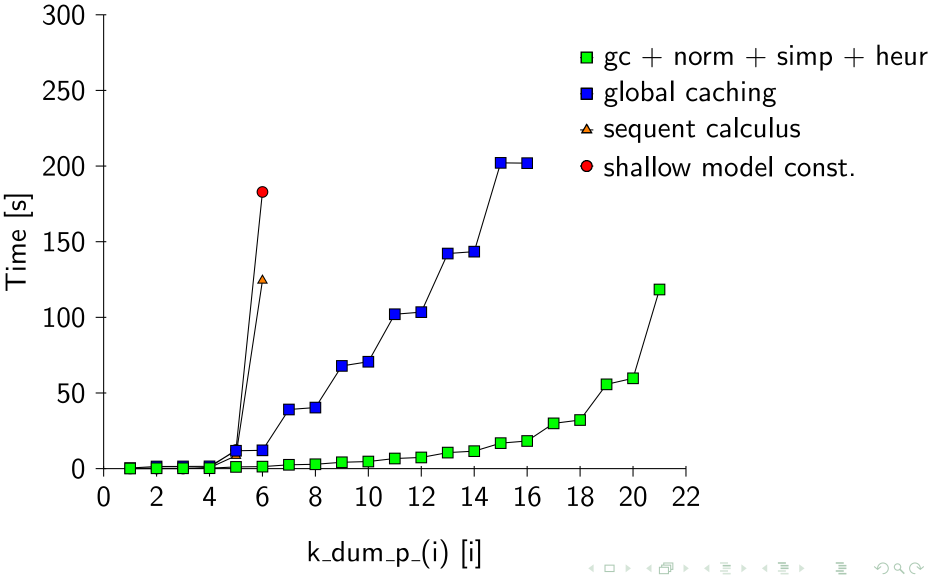
CoLoSS

*Co*algebraic *Lo*gic *S*atisfiability *S*olver (CoLoSS); haskell implementation of generic sequent algorithm, (basic) generic global caching algorithm.

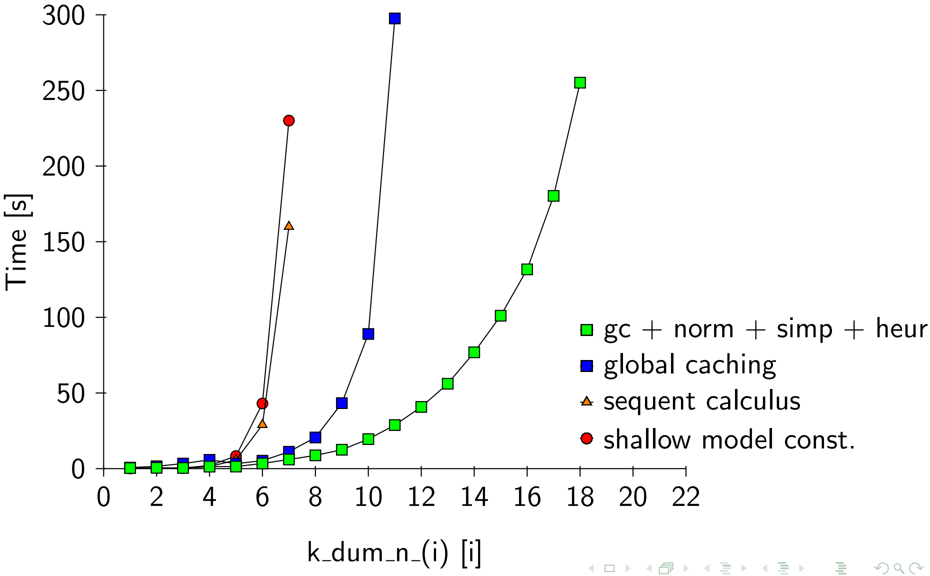
- Basic optimisations: Normalisation, simplification
- No global caching: Generalized semantic branching and generalized dependency directed backtracking are admissible
- Heuristics (*Most-often first*,*oldest first*)

`http://www.informatik.uni-bremen.de/cofi/CoLoSS/`

Benchmarking, k_dum_p, Tableaux 98



Benchmarking, k_dum_n, Tableaux 98



Conclusion / Future Work

- Consistency of coalgebraic modal logics by tableau algorithm.
- Treatment of global assumptions and fixed point logics by global caching algorithms.
- Efficient propagation by iterative approximation of gfps.
- Small ($2^{|\phi|_m}$) Model Lemma for depth-1 flat fixed point logics [Emerson, Halpern, 1985: $|\phi|_m(8^{|\phi|_m})$; Cirstea, Kupke, Pattinson, 2009: $2^{p|\phi|_m}$].

- Remove (technical?) restriction to depth-1 fixed point logics.
- Possibly extend technique to alternation-free fragment of μ-calculus.

Thank you for your Attention!

Conclusion / Future Work

- Consistency of coalgebraic modal logics by tableau algorithm.
- Treatment of global assumptions and fixed point logics by global caching algorithms.
- Efficient propagation by iterative approximation of gfps.
- Small ($2^{|\phi|_m}$) Model Lemma for depth-1 flat fixed point logics [Emerson, Halpern, 1985: $|\phi|_m(8^{|\phi|_m})$; Cirstea, Kupke, Pattinson, 2009: $2^{p|\phi|_m}$].

- Remove (technical?) restriction to depth-1 fixed point logics.
- Possibly extend technique to alternation-free fragment of μ-calculus.

Thank you for your Attention!