Will Deflation Lead to Depletion?

On Non-Monotone Fixed Point Inductions

Erich Grädel and Stephan Kreutzer

Will Deflation Lead to Depletion?

On Non-Monotone Fixed Point Inductions

Erich Grädel and Stephan Kreutzer

Deflation: *reduction in size, importance, or effectiveness;*
contraction of economic activity resulting in a decline of prices;
the erosion of soil by the wind.

Depletion: *exhaustion of a principal substance, especially a natural resource;*
a reduction in quantity so as to endanger the ability to function.

The lazy engineer

Story: The lazy engineer (LE) has built a system $\mathfrak{A}$ (a relational structure), to satisfy a specification $\varphi(x)$ that should hold at all elements $a \in \mathfrak{A}$.

The lazy engineer

Story: The lazy engineer (LE) has built a system $\mathfrak{A}$ (a relational structure), to satisfy a specification $\varphi(x)$ that should hold at all elements $a \in \mathfrak{A}$.

Unfortunately, it does not: $\mathfrak{A} \not\models \forall x\, \varphi(x)$

The lazy engineer

Story: The lazy engineer (LE) has built a system $\mathfrak{A}$ (a relational structure), to satisfy a specification $\varphi(x)$ that should hold at all elements $a \in \mathfrak{A}$.

Unfortunately, it does not: $\mathfrak{A} \not\models \forall x\, \varphi(x)$

LE: "Well, I just throw away the bad elements ! "

$\mathfrak{A} \mapsto \mathfrak{A}|_\varphi$ (substructure induced by $\{a : \mathfrak{A} \models \varphi(a)\}$)

The lazy engineer

Story: The lazy engineer (LE) has built a system $\mathfrak{A}$ (a relational structure), to satisfy a specification $\varphi(x)$ that should hold at all elements $a \in \mathfrak{A}$.

Unfortunately, it does not: $\mathfrak{A} \not\models \forall x\, \varphi(x)$

LE: "Well, I just throw away the bad elements ! "

$$\mathfrak{A} \;\mapsto\; \mathfrak{A}|_\varphi \qquad \text{(substructure induced by } \{a : \mathfrak{A} \models \varphi(a)\}\text{)}$$

LE: "With a little luck, $\mathfrak{A}|_\varphi \models \forall x \varphi(x)$.
And if not, I just relativise again ! And again! And again ....! "

The lazy engineer

Story: The lazy engineer (LE) has built a system $\mathfrak{A}$ (a relational structure), to satisfy a specification $\varphi(x)$ that should hold at all elements $a \in \mathfrak{A}$.

Unfortunately, it does not: $\mathfrak{A} \not\models \forall x\, \varphi(x)$

LE: "Well, I just throw away the bad elements ! "

$$\mathfrak{A} \mapsto \mathfrak{A}|_{\varphi} \quad \text{(substructure induced by } \{a : \mathfrak{A} \models \varphi(a)\}\text{)}$$

LE: "With a little luck, $\mathfrak{A}|_{\varphi} \models \forall x \varphi(x)$.
And if not, I just relativise again ! And again! And again ....! "

$$\mathfrak{A} = \mathfrak{A}^0 \supseteq \mathfrak{A}^1 \supseteq \mathfrak{A}^2 \supseteq \cdots \quad \text{converges to fixed point } \mathfrak{A}^{\infty}$$

The lazy engineer

Story: The lazy engineer (LE) has built a system $\mathfrak{A}$ (a relational structure), to satisfy a specification $\varphi(x)$ that should hold at all elements $a \in \mathfrak{A}$.

Unfortunately, it does not: $\mathfrak{A} \not\models \forall x\, \varphi(x)$

LE: "Well, I just throw away the bad elements ! "

$$\mathfrak{A} \;\mapsto\; \mathfrak{A}|_\varphi \qquad \text{(substructure induced by } \{a : \mathfrak{A} \models \varphi(a)\}\text{)}$$

LE: "With a little luck, $\mathfrak{A}|_\varphi \models \forall x\varphi(x)$.
And if not, I just relativise again ! And again! And again ....! "

$$\mathfrak{A} = \mathfrak{A}^0 \supseteq \mathfrak{A}^1 \supseteq \mathfrak{A}^2 \supseteq \cdots \qquad \text{converges to fixed point } \mathfrak{A}^\infty$$

LE: "Obviously, $\mathfrak{A}^\infty \models \forall x\varphi(x)$! "

The lazy engineer

Story: The lazy engineer (LE) has built a system $\mathfrak{A}$ (a relational structure), to satisfy a specification $\varphi(x)$ that should hold at all elements $a \in \mathfrak{A}$.

Unfortunately, it does not: $\mathfrak{A} \not\models \forall x\, \varphi(x)$

LE: "Well, I just throw away the bad elements ! "

$$\mathfrak{A} \;\mapsto\; \mathfrak{A}|_\varphi \qquad \text{(substructure induced by } \{a : \mathfrak{A} \models \varphi(a)\}\text{)}$$

LE: "With a little luck, $\mathfrak{A}|_\varphi \models \forall x \varphi(x)$.
And if not, I just relativise again ! And again! And again ....! "

$$\mathfrak{A} = \mathfrak{A}^0 \supseteq \mathfrak{A}^1 \supseteq \mathfrak{A}^2 \supseteq \cdots \qquad \text{converges to fixed point } \mathfrak{A}^\infty$$

LE: "Obviously, $\mathfrak{A}^\infty \models \forall x \varphi(x)$! "

Problem: $\mathfrak{A}^\infty$ may be empty. "Does deflation lead to depletion?"

The lazy engineer

Story: The lazy engineer (LE) has built a system $\mathfrak{A}$ (a relational structure), to satisfy a specification $\varphi(x)$ that should hold at all elements $a \in \mathfrak{A}$.

Unfortunately, it does not: $\mathfrak{A} \not\models \forall x\, \varphi(x)$

LE: "Well, I just throw away the bad elements ! "

$$\mathfrak{A} \mapsto \mathfrak{A}|_\varphi \quad \text{(substructure induced by } \{a : \mathfrak{A} \models \varphi(a)\} \text{)}$$

LE: "With a little luck, $\mathfrak{A}|_\varphi \models \forall x \varphi(x)$.
And if not, I just relativise again ! And again! And again ....! "

$$\mathfrak{A} = \mathfrak{A}^0 \supseteq \mathfrak{A}^1 \supseteq \mathfrak{A}^2 \supseteq \cdots \quad \text{converges to fixed point } \mathfrak{A}^\infty$$

LE: "Obviously, $\mathfrak{A}^\infty \models \forall x \varphi(x)$! "

Problem: $\mathfrak{A}^\infty$ may be empty. "Does deflation lead to depletion?"

In what kind of (fixed point) logic can we express this question?

Fixed point logics

Extend some basic logical formalism by fixed points of relational operators

Fixed point logics

Extend some basic logical formalism by fixed points of relational operators

Most popular: fixed point logics based on least and greatest fixed points:

ML	(modal logic)	$\longrightarrow$	L_μ	(modal μ-calculus)
FO	(first-order logic)	$\longrightarrow$	LFP	(least fixed point logic)
conjunctive queries		$\longrightarrow$	Datalog / Stratified Datalog	

Fixed point logics

Extend some basic logical formalism by fixed points of relational operators

Most popular: fixed point logics based on least and greatest fixed points:

ML	(modal logic)	$\longrightarrow$	L_μ	(modal μ-calculus)
FO	(first-order logic)	$\longrightarrow$	**LFP**	(least fixed point logic)
conjunctive queries		$\longrightarrow$	Datalog / Stratified Datalog	

Idea: Capture recursion. Any monotone relational operator

$$F_\varphi : T \mapsto \{\bar{x} : \varphi(T, \bar{x})\}$$

has a least and a greatest fixed point. Make them definable by formulae

$$[\mathbf{lfp}\, T\bar{x} \,.\, \varphi(T, \bar{x})](\bar{z}) \qquad [\mathbf{gfp}\, T\bar{x} \,.\, \varphi(T, \bar{x})](\bar{z})$$

$$\mu X \,.\, \varphi \qquad \nu X \,.\, \varphi$$

Non-montone fixed point logics

Prehistory (1970s): study of monotone and non-monotone inductive definability on abstract structures (Aczel, Kechris, Moschovakis . . .)

Non-montone fixed point logics

Prehistory (1970s): study of monotone and non-monotone inductive definability on abstract structures (Aczel, Kechris, Moschovakis . . .)

No fixed point logics in the modern sense:

- inductions over first-order formulae
- fixed points are not interleaved with other logical operations

Non-montone fixed point logics

Prehistory (1970s): study of monotone and non-monotone inductive definability on abstract structures (Aczel, Kechris, Moschovakis . . .)

No fixed point logics in the modern sense:

- inductions over first-order formulae
- fixed points are not interleaved with other logical operations

many fundamental techniques for the study of fixed point logics

Non-montone fixed point logics

Prehistory (1970s): study of monotone and non-monotone inductive definability on abstract structures (Aczel, Kechris, Moschovakis . . .)

No fixed point logics in the modern sense:

- inductions over first-order formulae
- fixed points are not interleaved with other logical operations

many fundamental techniques for the study of fixed point logics

Finite model theory and databases (since 1980s): Fixed point logics based on non-monotone operators: IFP, PFP, NFP, AFP, . . .

In this talk: logics with inflationary and deflationary fixed points:

FO	(first-order logic)	$\longrightarrow$	**IFP**	(inflationary fixed point logic)
ML	(modal logic)	$\longrightarrow$	**MIC**	(modal iteration calculus)

Outline

- Examples: greatest fixed points — deflationary fixed points
- Fixed point extensions of first-order logic:
  compare IFP with LFP
- Fixed point extensions of propositional modal logic:
  compare MIC with L_μ
- Model checking games for inflationary fixed point logics ?

Greatest fixed points (in LFP)

$[\mathbf{gfp}\ T\overline{x}\,.\,\varphi(T,\overline{x})](\overline{a})$: $\overline{a}$ contained in greatest T with $T = \{\overline{x} : \varphi(T,\overline{x})\}$

Greatest fixed points (in LFP)

$[\mathbf{gfp}\, T\overline{x}\,.\,\varphi(T,\overline{x})](\overline{a})$: $\overline{a}$ contained in greatest T with $T = \{\overline{x} : \varphi(T,\overline{x})\}$

this T exists if $F_\varphi : T \mapsto \{\overline{x} : \varphi(T,\overline{x})\}$ is monotone (preserves $\subseteq$)

to guarantee monotonicity: require that T positive in φ

Greatest fixed points (in LFP)

$[\mathbf{gfp}\, T\overline{x}\,.\,\varphi(T,\overline{x})](\overline{a})$: $\overline{a}$ contained in greatest T with $T = \{\overline{x} : \varphi(T,\overline{x})\}$

this T exists if $F_\varphi : T \mapsto \{\overline{x} : \varphi(T,\overline{x})\}$ is monotone (preserves $\subseteq$)

to guarantee monotonicity: require that T positive in φ

Inductive construction of the greatest fixed point on a structure $\mathfrak{A}$:

$$T^0 := A^k \quad \text{(all tuples of appropriate arity)}$$

$$T^{\alpha+1} := F_\varphi(T^\alpha)$$

$$T^\lambda := \bigcap_{\alpha<\lambda} T^\alpha \quad (\lambda \text{ limit ordinal})$$

$\Longrightarrow$ decreasing sequence of stages ($T^\alpha \supseteq T^{\alpha+1}$),
converges to a fixed point T^∞ of F_φ

Greatest fixed points (in LFP)

$[\mathbf{gfp}\, T\overline{x}\,.\,\varphi(T,\overline{x})](\overline{a})$: $\overline{a}$ contained in greatest T with $T = \{\overline{x} : \varphi(T,\overline{x})\}$

this T exists if $F_\varphi : T \mapsto \{\overline{x} : \varphi(T,\overline{x})\}$ is monotone (preserves $\subseteq$)
to guarantee monotonicity: require that T positive in φ

Inductive construction of the greatest fixed point on a structure $\mathfrak{A}$:

$$T^0 := A^k \quad \text{(all tuples of appropriate arity)}$$

$$T^{\alpha+1} := F_\varphi(T^\alpha)$$

$$T^\lambda := \bigcap_{\alpha<\lambda} T^\alpha \quad (\lambda \text{ limit ordinal})$$

$\Longrightarrow$ decreasing sequence of stages $(T^\alpha \supseteq T^{\alpha+1})$,
converges to a fixed point T^∞ of F_φ

Fact: $T^\infty = \mathbf{gfp}(F_\varphi)$ (Knaster, Tarski)

Example: Bisimulation

$\mathcal{K} = (V, E, P_1, \ldots, P_m)$ transition system

Bisimilarity on $\mathcal{K}$ is the greatest equivalence relation $Z \subseteq V \times V$ such that:

if $(u, v) \in Z$ then

– u and v have the same atomic properties

– from u and v there are edges into the same equivalence classes

Example: Bisimulation

$\mathcal{K} = (V, E, P_1, \dots, P_m)$ transition system

Bisimilarity on $\mathcal{K}$ is the greatest equivalence relation $Z \subseteq V \times V$ such that:

if $(u, v) \in Z$ then

– u and v have the same atomic properties

– from u and v there are edges into the same equivalence classes

Thus, bisimilarity is the greatest fixed point of the refinement operator

$$Z \quad \mapsto \quad \{(u, v) : \mathcal{K} \models \varphi(Z, u, v)\} \quad \text{where}$$

$$\varphi := \bigwedge_{i \leq m} P_i u \leftrightarrow P_i v \wedge$$
$$\forall x\,(Eux \longrightarrow \exists y(Evy \wedge Zxy)) \;\wedge\; \forall y(Evy \longrightarrow \exists x(Eux \wedge Zxy))$$

u and v are bisimilar in $\mathcal{K} \iff \mathcal{K} \models [\mathbf{gfp}\, Zuv \,.\, \varphi](u, v)$

Deflationary fixed points

$[\mathbf{dfp}\ T\overline{x}\,.\,\varphi(T,\overline{x})]$: defines the deflationary fixed point of φ.

Note: T can occur negative in φ and F_φ need not be monotone.

Deflationary fixed points

$[\mathbf{dfp}\, T\overline{x}\,.\,\varphi(T,\overline{x})]$: defines the deflationary fixed point of φ.

Note: T can occur negative in φ and F_φ need not be monotone.

Nevertheless we can define a decreasing sequence of stages:

$$T^0 := A^k \quad \text{(all tuples of appropriate arity)}$$

$$T^{\alpha+1} := T^\alpha \cap F_\varphi(T^\alpha)$$

$$T^\lambda := \bigcap_{\alpha<\lambda} T^\alpha \quad (\lambda \text{ limit ordinal})$$

Deflationary fixed points

$[\mathbf{dfp}\ T\overline{x}\,.\,\varphi(T,\overline{x})]$: defines the deflationary fixed point of φ.

Note: T can occur negative in φ and F_φ need not be monotone.

Nevertheless we can define a decreasing sequence of stages:

$$T^0 := A^k \quad \text{(all tuples of appropriate arity)}$$

$$T^{\alpha+1} := T^\alpha \cap F_\varphi(T^\alpha)$$

$$T^\lambda := \bigcap_{\alpha<\lambda} T^\alpha \quad (\lambda \text{ limit ordinal})$$

$\Longrightarrow$ converges to a fixed point of $T \mapsto T \cap F_\varphi(T)$
called the deflationary fixed point of φ

Deflationary fixed points

$[\mathbf{dfp}\, T\overline{x}\,.\,\varphi(T,\overline{x})]$: defines the deflationary fixed point of φ.

Note: T can occur negative in φ and F_φ need not be monotone.

Nevertheless we can define a decreasing sequence of stages:

$$T^0 := A^k \quad \text{(all tuples of appropriate arity)}$$

$$T^{\alpha+1} := T^\alpha \cap F_\varphi(T^\alpha)$$

$$T^\lambda := \bigcap_{\alpha<\lambda} T^\alpha \quad (\lambda \text{ limit ordinal})$$

$\Longrightarrow$ converges to a fixed point of $T \mapsto T \cap F_\varphi(T)$
called the deflationary fixed point of φ

Inflationary fixed points: defined dually

Example: the lazy engineer revisited

consider the relativisation operator defined by $\varphi(x)$ on substructures of $\mathfrak{A}$ as an operator on subsets $Z \subseteq A$:

$$\mathrm{Rel}[\varphi] : Z \quad \longmapsto \{z \in Z : \mathfrak{A}|_Z \models \varphi(z)\} = Z \cap \{a \in A : \mathfrak{A} \models \varphi|_Z(a)\}$$

Example: the lazy engineer revisited

consider the relativisation operator defined by $\varphi(x)$ on substructures of $\mathfrak{A}$ as an operator on subsets $Z \subseteq A$:

$$\mathrm{Rel}[\varphi] : Z \quad \longmapsto \{z \in Z : \mathfrak{A}|_Z \models \varphi(z)\} = Z \cap \{a \in A : \mathfrak{A} \models \varphi|_Z(a)\}$$

Iterated relativisation is a deflationary induction via the formula $\varphi|_Z(x)$, which is the syntactic relativisation of $\varphi(x)$ to Z:

replace inductively: $\exists y \alpha \;\rightsquigarrow\; \exists y(Zy \wedge \alpha)$, $\quad \forall y \alpha \;\rightsquigarrow\; \forall y(Zy \rightarrow \alpha)$

Note: The relativisation operator is in general not monotone.

Example: the lazy engineer revisited

consider the relativisation operator defined by $\varphi(x)$ on substructures of $\mathfrak{A}$ as an operator on subsets $Z \subseteq A$:

$$\mathrm{Rel}[\varphi] : Z \longmapsto \{z \in Z : \mathfrak{A}|_Z \models \varphi(z)\} = Z \cap \{a \in A : \mathfrak{A} \models \varphi|_Z(a)\}$$

Iterated relativisation is a deflationary induction via the formula $\varphi|_Z(x)$, which is the syntactic relativisation of $\varphi(x)$ to Z:

replace inductively: $\exists y\alpha \rightsquigarrow \exists y(Zy \land \alpha)$, $\quad \forall y\alpha \rightsquigarrow \forall y(Zy \rightarrow \alpha)$

Note: The relativisation operator is in general not monotone.

Does deflation lead to depletion?

$$\mathfrak{A}^\infty \text{ non-empty} \iff \mathfrak{A} \models \exists y[\mathbf{dfp}\, Zx\,.\,\varphi|_Z(x)](y)$$

Example: the lazy engineer revisited

consider the relativisation operator defined by $\varphi(x)$ on substructures of $\mathfrak{A}$ as an operator on subsets $Z \subseteq A$:

$$\mathrm{Rel}[\varphi] : Z \longmapsto \{z \in Z : \mathfrak{A}|_Z \models \varphi(z)\} = Z \cap \{a \in A : \mathfrak{A} \models \varphi|_Z(a)\}$$

Iterated relativisation is a deflationary induction via the formula $\varphi|_Z(x)$, which is the syntactic relativisation of $\varphi(x)$ to Z:

replace inductively: $\exists y\alpha \rightsquigarrow \exists y(Zy \wedge \alpha)$, $\quad \forall y\alpha \rightsquigarrow \forall y(Zy \rightarrow \alpha)$

Note: The relativisation operator is in general not monotone.

Does deflation lead to depletion?

$$\mathfrak{A}^\infty \text{ non-empty} \iff \mathfrak{A} \models \exists y[\mathbf{dfp}\, Zx\,.\,\varphi|_Z(x)](y)$$

Question: Is this also definable in LFP ?

Logics of knowledge

multi-modal propositional logics: $[a]\varphi$ stands for "agent a knows φ"

Logics of knowledge

multi-modal propositional logics: $[a]\varphi$ stands for "agent a knows φ"

add common knowledge:

everybody knows φ, and everybody knows that everybody knows φ, and everybody knows that everybody knows that everybody knows . . .

expressible as a greatest fixed point: $C\varphi \equiv \nu X . (\varphi \wedge \bigwedge_a [a]X)$

Logics of knowledge

multi-modal propositional logics: $[a]\varphi$ stands for "agent a knows φ"

add common knowledge:

everybody knows φ, and everybody knows that everybody knows φ, and everybody knows that everybody knows that everybody knows . . .

expressible as a greatest fixed point: $C\varphi \equiv \nu X . (\varphi \wedge \bigwedge_a [a]X)$

Question: Suppose that someone (who is trusted by all agents) publicly announces φ. Has φ become common knowledge ?

Logics of knowledge

multi-modal propositional logics: $[a]\varphi$ stands for "agent a knows φ"

add common knowledge:

everybody knows φ, and everybody knows that everybody knows φ, and everybody knows that everybody knows that everybody knows . . .

expressible as a greatest fixed point: $C\varphi \equiv \nu X\,.\,(\varphi \wedge \bigwedge_a [a]X)$

Question: Suppose that someone (who is trusted by all agents) publicly announces φ. Has φ become common knowledge ?

public announcement of φ changes knowledge model $\mathcal{K}$ to $\mathcal{K}|_\varphi$:

remove states where φ does not hold.

public announcement as a logical operator:

$[\varphi!]\psi$: ψ is true after public announcement of φ

$\mathcal{K}, v \models [\varphi!]\psi \iff \mathcal{K}|_\varphi, v \models \psi$

Iterated public announcement

Note: φ is not necessarily common knowledge in $\mathcal{K}|_{\varphi}$

typical counterexamples are statements about ignorance

Iterated public announcement

Note: φ is not necessarily common knowledge in $\mathcal{K}|_\varphi$

typical counterexamples are statements about ignorance

Repeated public announcement (van Benthem, Baltag et al., Miller/Moss)

$[\varphi!^*]\psi$: by iterated announcement of φ, ψ eventually becomes true.

Iterated public announcement

Note: φ is not necessarily common knowledge in $\mathcal{K}|_\varphi$

typical counterexamples are statements about ignorance

Repeated public announcement (van Benthem, Baltag et al., Miller/Moss)

$[\varphi!^*]\psi$: by iterated announcement of φ, ψ eventually becomes true.

Note: public announcement $[\varphi!]\psi$ does not take us out of knowledge logics that are closed under syntactic relativisation (like ML or L_μ):

$$\mathcal{K}, v \models [\varphi!]\psi \iff \mathcal{K}|_\varphi, v \models \psi \iff \mathcal{K}, v \models \psi|_\varphi$$

Iterated public announcement

Note: φ is not necessarily common knowledge in $\mathcal{K}|_\varphi$

typical counterexamples are statements about ignorance

Repeated public announcement (van Benthem, Baltag et al., Miller/Moss)

$[\varphi!^*]\psi$: by iterated announcement of φ, ψ eventually becomes true.

Note: public announcement $[\varphi!]\psi$ does not take us out of knowledge logics that are closed under syntactic relativisation (like ML or L_μ):

$$\mathcal{K}, v \models [\varphi!]\psi \iff \mathcal{K}|_\varphi, v \models \psi \iff \mathcal{K}, v \models \psi|_\varphi$$

Repeated public announcement corresponds to iterated relativisation and is therefore expressible via a deflationary fixed point:

$$[\varphi!^*]\varphi \equiv (\mathbf{dfp}\, X \,.\, \varphi|_X)$$

Question: (van Benthem) Is $[\varphi!^*]\varphi$ expressible in the modal μ-calculus ?

LFP versus IFP

Question: $\text{LFP} \quad \equiv \quad \text{IFP}\ ?$

LFP versus IFP

Question: LFP $\equiv$ IFP ?

- Obviously, LFP $\leq$ IFP

  for monotone operators, ifp = lfp and dfp = gfp

LFP versus IFP

Question: LFP $\equiv$ IFP ?

- Obviously, LFP $\leq$ IFP

  for monotone operators, ifp = lfp and dfp = gfp

- Immerman/Vardi (1982):

  On ordered finite structures, LFP and IFP both capture PTIME.

LFP versus IFP

Question: LFP $\equiv$ IFP ?

- Obviously, LFP $\leq$ IFP
  for monotone operators, ifp = lfp and dfp = gfp
- Immerman/Vardi (1982):
  On ordered finite structures, LFP and IFP both capture PTIME.
- Gurevich/Shelah (1986):
  On finite structures, LFP $\equiv$ IFP.

Many believed that, in general, IFP $\not\leq$ LFP.

Evidence: On $(\omega, +, \cdot)$, non-monotone inductions over FO-formulae are stronger than monotone ones (Aczel, Moschovakis, . . .)

However, Dawar and Gurevich observed that on $(\omega, +, \cdot)$, and any other admissible structure, LFP $\equiv$ IFP.

A fundamental tool: stage comparison relations

$\bar{x}$ gets into the fixed point before $\bar{y}$ does:

$$\bar{x} \prec_\varphi \bar{y} \quad :\Longleftrightarrow \quad \text{for some ordinal } \alpha,\ \bar{x} \in T^\alpha \text{ but } \bar{y} \notin T^\alpha$$

(T^α : stages of lfp-induction via φ)

$\bar{x} \prec_\psi^{\text{inf}} \bar{y}$: similarly for ifp-induction via ψ

A fundamental tool: stage comparison relations

$\bar{x}$ gets into the fixed point before $\bar{y}$ does:

$$\bar{x} \prec_\varphi \bar{y} \quad :\Longleftrightarrow \quad \text{for some ordinal } \alpha,\ \bar{x} \in T^\alpha \text{ but } \bar{y} \notin T^\alpha$$

(T^α : stages of lfp-induction via φ)

$\bar{x} \prec_\psi^{\text{inf}} \bar{y}$: similarly for ifp-induction via ψ

Example: $\varphi(T, x, y) = Exy \wedge \exists z(Exz \wedge Tzy)$

$[\mathbf{lfp}\ Txy \,.\, \varphi]$ defines the transitive closure of E

$$(a, b) \prec_\varphi (c, d) \quad \Longleftrightarrow \quad \text{distance}(a, b) < \text{distance}(c, d)$$

Stage Comparison Theorems

results about definability of stage comparison relations $\prec_\varphi$ or $\prec_\psi^{\inf}$

Theorem. (Moschovakis (1974) For every $\varphi \in$ FO, the stage comparison relation $\prec_\varphi$ is definable by an lfp-induction over FO-formulae

Stage Comparison Theorems

results about definability of stage comparison relations $\prec_\varphi$ or $\prec_\psi^{\text{inf}}$

Theorem. (Moschovakis (1974) For every $\varphi \in$ FO, the stage comparison relation $\prec_\varphi$ is definable by an lfp-induction over FO-formulae

$\Longrightarrow$ normal form for LFP on finite structures (Immerman 86)

greatest fixed points definable via least fixed points only

the alternation hierarchy of LFP collapses on finite structures

Stage Comparison Theorems

results about definability of stage comparison relations $\prec_\varphi$ or $\prec_\psi^{\text{inf}}$

Theorem. (Moschovakis (1974) For every $\varphi \in$ FO, the stage comparison relation $\prec_\varphi$ is definable by an lfp-induction over FO-formulae

$\Longrightarrow$ normal form for LFP on finite structures (Immerman 86)

greatest fixed points definable via least fixed points only

the alternation hierarchy of LFP collapses on finite structures

Note: these consequences of the stage comparison theorem fail on infinite structures.

In general, the alternation hierarchy is strict (Moschovakis, Bradfield)

Kreutzer's Theorem

Simple observation: $\prec_{\psi}^{\text{inf}}$ is IFP-definable for all $\varphi \in$ IFP.

However, this is not useful. We need:

Inflationary Stage Comparison Theorem:

$\prec_{\psi}^{\text{inf}}$ is LFP-definable for all $\varphi \in$ LFP.

Kreutzer's Theorem

Simple observation: $\prec_{\psi}^{\inf}$ is IFP-definable for all $\varphi \in$ IFP.

However, this is not useful. We need:

Inflationary Stage Comparison Theorem:

$\prec_{\psi}^{\inf}$ is LFP-definable for all $\varphi \in$ LFP.

proved by Gurevich and Shelah for finite structures

uses existence of elements coming into the fixed point at last stage

Kreutzer (LICS 02) gives proof for general case

Kreutzer's Theorem

Simple observation: $\prec_{\psi}^{\text{inf}}$ is IFP-definable for all $\varphi \in \text{IFP}$.

However, this is not useful. We need:

Inflationary Stage Comparison Theorem:

$\prec_{\psi}^{\text{inf}}$ is LFP-definable for all $\varphi \in \text{LFP}$.

proved by Gurevich and Shelah for finite structures

uses existence of elements coming into the fixed point at last stage

Kreutzer (LICS 02) gives proof for general case

Corollary: LFP $\equiv$ IFP.

Proof. $[\mathbf{ifp}\, T\overline{x}\,.\,\psi(R,\overline{x})](\overline{x}) \equiv \psi(\{\overline{y} : \overline{y} \prec_{\psi}^{\text{inf}} \overline{x}\}, \overline{x})$

Kreutzer's Theorem

Simple observation: $\prec_{\psi}^{\text{inf}}$ is IFP-definable for all $\varphi \in$ IFP.

However, this is not useful. We need:

Inflationary Stage Comparison Theorem:

$\prec_{\psi}^{\text{inf}}$ is LFP-definable for all $\varphi \in$ LFP.

proved by Gurevich and Shelah for finite structures

uses existence of elements coming into the fixed point at last stage

Kreutzer (LICS 02) gives proof for general case

Corollary: LFP $\equiv$ IFP.

Proof. $[\mathbf{ifp}\, T\overline{x} \,.\, \psi(R, \overline{x})](\overline{x}) \;\equiv\; \psi(\{\overline{y} : \overline{y} \prec_{\psi}^{\text{inf}} \overline{x}\}, \overline{x})$

Corollary: Iterated relativisation is LFP-definable

(It is not clear how to do this directly, without using Kreutzer's Theorem)

LFP versus IFP

Although LFP $\equiv$ IFP, the two logics have different structures

- the translation from IFP to LFP makes formulae much more complicated (length, arity, nesting depth)
  $[\mathbf{ifp}\ T\overline{x} \quad \ldots] \quad \rightsquigarrow \quad [\mathbf{lfp}\ R\overline{xy} \quad \ldots [\mathbf{gfp}\ S\overline{xy} \quad \ldots] \ldots]$
- the alternation hierarchy for LFP is strict
  the alternation hierarchy for IFP collapses
- nested gfp can be collapsed to a single one (of larger arity)
  nested dfp cannot be collapsed
- IFP is a more robust logic
  (remains well-defined when other operators are added)

monotone versus non-monotone fixed points in modal logic

L_μ: propositional modal logic **ML** + least and greatest fixed points

Examples:

- $\mathcal{K}, w \models \nu X \,.\, \langle a \rangle X \iff$ there is an infinite a-path from w in $\mathcal{K}$

  $\mathcal{K}, w \models \mu X \,.\, P \lor [a]X \iff$ every infinite a-path from w eventually hits P

monotone versus non-monotone fixed points in modal logic

L_μ: propositional modal logic **ML** + least and greatest fixed points

Examples:

- $\mathcal{K}, w \models \nu X . \langle a \rangle X \quad \Longleftrightarrow$ there is an infinite a-path from w in $\mathcal{K}$

  $\mathcal{K}, w \models \mu X . P \vee [a]X \quad \Longleftrightarrow$ every infinite a-path from w eventually hits P

- $\mathcal{K}, w \models \nu X \, \mu Y . \Diamond((P \wedge X) \vee Y) \quad \Longleftrightarrow$

  on some path from w, P occurs infinitely often

monotone versus non-monotone fixed points in modal logic

L_μ: propositional modal logic **ML** + least and greatest fixed points

Examples:

- $\mathcal{K}, w \models \nu X\,.\,\langle a \rangle X \iff$ there is an infinite a-path from w in $\mathcal{K}$
  $\mathcal{K}, w \models \mu X\,.\,P \vee [a]X \iff$ every infinite a-path from w eventually hits P
- $\mathcal{K}, w \models \nu X\,\mu Y\,.\,\Diamond((P \wedge X) \vee Y) \iff$
  on some path from w, P occurs infinitely often

One can also permit simultaneous inductions over several formulae without changing the expressive power.

Bekic principle:

$$\nu\, XY\,.\,(\psi(X,Y), \varphi(X,Y)) \equiv \nu X\,.\,\psi(X, \nu Y.\varphi(X,Y))$$

Importance of the modal μ-calculus

- encompasses popular logics used in hardware verification and other fields: LTL, CTL, CTL*, PDL, game logic, description logics, . . .

Importance of the modal μ-calculus

- encompasses popular logics used in hardware verification and other fields: LTL, CTL, CTL*, PDL, game logic, description logics, . . .
- well developed theory
  interesting connection to automata and theory of infinite games

Importance of the modal μ-calculus

- encompasses popular logics used in hardware verification and other fields: LTL, CTL, CTL*, PDL, game logic, description logics, ...
- well developed theory
  interesting connection to automata and theory of infinite games
- L_μ is a modal logic with nice model-theoretic properties:
  - invariant under bisimulation, tree model property
  - finite model property
  - L_μ is the bisimulation-invariant fragment of MSO

Importance of the modal μ-calculus

- encompasses popular logics used in hardware verification and other fields: LTL, CTL, CTL*, PDL, game logic, description logics, . . .
- well developed theory
  interesting connection to automata and theory of infinite games
- L_μ is a modal logic with nice model-theoretic properties:
  - invariant under bisimulation, tree model property
  - finite model property
  - L_μ is the bisimulation-invariant fragment of MSO
- reasonably good algorithmic properties:
  - satisfiability problem decidable (Exptime-complete)
  - efficient model checking for practically important fragments of L_μ
  - algorithms based on automata and games
  - model checking for L_μ is in NP $\cap$ Co-NP. Open: is it in P?

Inflationary fixed points in modal logic

$$\text{LFP} \quad : \quad \text{IFP} \quad = \quad L_\mu \quad : \quad ????$$

Inflationary fixed points in modal logic

$$\text{LFP} \quad : \quad \text{IFP} \quad = \quad L_\mu \quad : \quad \text{MIC}$$

MIC: modal iteration calculus (Dawar, G. , Kreutzer)

ML + simultaneous inflationary and deflationary fixed points

The modal iteration calculus

Syntax: given formulae $\varphi_1, \dots, \varphi_m$, and a system of rules

$$S := \begin{cases} X_1 \leftarrow \varphi_1(X_1, \dots, X_m) \\ \vdots \\ X_m \leftarrow \varphi_m(X_1, \dots, X_m) \end{cases}$$

build new formulae $(\mathbf{dfp}\, X_i : S)$ and $(\mathbf{ifp}\, X_i : S)$

The modal iteration calculus

Syntax: given formulae $\varphi_1, \dots, \varphi_m$, and a system of rules

$$S := \begin{cases} X_1 \leftarrow \varphi_1(X_1, \dots, X_m) \\ \vdots \\ X_m \leftarrow \varphi_m(X_1, \dots, X_m) \end{cases}$$

build new formulae $(\mathbf{dfp}\, X_i : S)$ and $(\mathbf{ifp}\, X_i : S)$

Semantics: S defines decreasing sequence of stages $\overline{X}^\alpha = (X_1^\alpha, \dots, X_m^\alpha)$, via

$$X_i^0 = V \quad \text{(all states)}$$

$$X_i^{\alpha+1} := X_i^\alpha \cap \{v : (\mathcal{K}, \overline{X}^\alpha), v \models \varphi_i\}$$

$$X_i^\lambda := \bigcap_{\alpha<\lambda} X_i^\alpha \quad (\lambda \text{ limit ordinal})$$

converging to a fixed point $\overline{X}^\infty = (X_1^\infty, \dots, X_m^\infty)$.

$$\mathcal{K}, v \models (\mathbf{dfp}\, X_i : S) \quad :\Longleftrightarrow \quad v \in X_i^\infty$$

Inflationary fixed points in modal logic

$$\text{LFP} \quad : \quad \text{IFP} \quad = \quad L_\mu \quad : \quad \text{MIC}$$

MIC: modal iteration calculus (Dawar, G., Kreutzer)

ML + simultaneous inflationary and deflationary fixed points

Inflationary fixed points in modal logic

$$\text{LFP} \quad : \quad \text{IFP} \quad = \quad L_\mu \quad : \quad \text{MIC}$$

MIC: modal iteration calculus (Dawar, G. , Kreutzer)

ML + simultaneous inflationary and deflationary fixed points

How is this analogy of definitions reflected by expressive power and complexity ?

$$\text{LFP} \quad : \quad \text{IFP} \quad = \quad L_\mu \quad : \quad \text{MIC}$$

MIC: modal iteration calculus (Dawar, G. , Kreutzer)
ML + simultaneous inflationary and deflationary fixed points

How is this analogy of definitions reflected by expressive power and complexity ?

Does MIC have similar model-theoretic and algorithmic properties as the modal μ-calculus ?

some simple observations on MIC

- $L_\mu \;\leq\;$ MIC

  $\mu X \,.\, \varphi \;\equiv (\mathbf{ifp}\, X \leftarrow \varphi)$ and $\nu X \,.\, \varphi \;\equiv (\mathbf{dfp}\, X \leftarrow \varphi)$

  (if X positive in φ)

some simple observations on MIC

- $L_\mu \leq$ MIC
  $\mu X . \varphi \equiv (\textbf{ifp}\, X \leftarrow \varphi)$ and $\nu X . \varphi \equiv (\textbf{dfp}\, X \leftarrow \varphi)$
  (if X positive in φ)

- MIC is a modal logic
  - invariant under bisimulation
  - tree model property

  unravel the fixed points: MIC $\leq$ ML$^\infty$ on any class of structures of bounded cardinality

Specific questions on MIC

- What is the expressive power of MIC ?

  MIC $\equiv L_\mu$? MIC $\leq$ MSO ?

Specific questions on MIC

- What is the expressive power of MIC ?
  MIC $\equiv L_\mu$? MIC $\leq$ MSO ?
- Does MIC have the finite model property ?
  every satisfiable $\psi \in L_\mu$ has a model of size $2^{O(|\psi|)}$

Specific questions on MIC

- What is the expressive power of MIC ?
  MIC $\equiv L_\mu$? MIC $\leq$ MSO ?

- Does MIC have the finite model property ?
  every satisfiable $\psi \in L_\mu$ has a model of size $2^{O(|\psi|)}$

- Is Sat(MIC) decidable ?
  $\mathrm{Sat}(L_\mu) \in \textsc{Exptime}$

Specific questions on MIC

- What is the expressive power of MIC ?
  MIC $\equiv L_\mu$? MIC $\leq$ MSO ?
- Does MIC have the finite model property ?
  every satisfiable $\psi \in L_\mu$ has a model of size $2^{O(|\psi|)}$
- Is Sat(MIC) decidable ?
  Sat(L_μ) $\in$ Exptime
- Complexity of model checking for MIC ?
  model checking for L_μ is in NP $\cap$ Co-NP, and possibly in P

Specific questions on MIC

- What is the expressive power of MIC ?
  MIC $\equiv L_\mu$? MIC $\leq$ MSO ?

- Does MIC have the finite model property ?
  every satisfiable $\psi \in L_\mu$ has a model of size $2^{O(|\psi|)}$

- Is Sat(MIC) decidable ?
  $\text{Sat}(L_\mu) \in \textsc{Exptime}$

- Complexity of model checking for MIC ?
  model checking for L_μ is in NP $\cap$ Co-NP, and possibly in P

- MIC $\equiv$ bisimulation-invariant PTIME ?
  multi-dimensional L_μ captures bisimulation-invariant PTIME

Specific questions on MIC

- What is the expressive power of MIC ?
  MIC $\equiv L_\mu$? MIC $\leq$ MSO ?
- Does MIC have the finite model property ?
  every satisfiable $\psi \in L_\mu$ has a model of size $2^{O(|\psi|)}$
- Is Sat(MIC) decidable ?
  Sat(L_μ) $\in$ EXPTIME
- Complexity of model checking for MIC ?
  model checking for L_μ is in NP $\cap$ Co-NP, and possibly in P
- MIC $\equiv$ bisimulation-invariant PTIME ?
  multi-dimensional L_μ captures bisimulation-invariant PTIME
- Epistemic logic with repeated public announcement is a fragment of MIC. What are its properties?
  Recall van Benthem's question: is it a fragment of L_μ ?

Example: MIC on words

Consider words $cxdy$ with $x, y \in \{a, b\}^*$

$$c \quad b \quad b \quad a \quad a \quad d \quad b \quad a \quad a \quad a$$

$$\psi := \mathbf{ifp}\, X \leftarrow (c \wedge \text{eventually}(\varphi)) \vee (\neg c \wedge \Box(X \vee d))$$

$$\text{eventually}(\alpha) \;:=\; \mathbf{ifp}\, X \leftarrow (\alpha \vee \Diamond X)$$

$$\text{frontier}(X) \;:=\; \neg X \wedge \Box(X \vee d)$$

$$\begin{aligned}\varphi := \text{frontier}(X) \wedge \big(\, & (a \wedge \text{eventually}(\text{frontier}(X) \wedge \neg a)) \\ & \vee (b \wedge \text{eventually}(\text{frontier}(X) \wedge \neg b)) \\ & \vee (c \wedge \text{eventually}(\text{frontier}(X) \wedge (a \vee b))\big)\end{aligned}$$

Claim: $cxdy \models \psi \iff x \neq y$

Example: MIC on words

Consider words $cxdy$ with $x, y \in \{a, b\}^*$

c	b	b	a	a	d	b	a	a	a
				X					X

$$\psi := \mathbf{ifp}\, X \leftarrow (c \wedge \text{eventually}(\varphi)) \vee (\neg c \wedge \Box(X \vee d))$$

$$\text{eventually}(\alpha) \;:=\; \mathbf{ifp}\, X \leftarrow (\alpha \vee \Diamond X)$$

$$\text{frontier}(X) \;:=\; \neg X \wedge \Box(X \vee d)$$

$$\begin{aligned}\varphi := \text{frontier}(X) \wedge \Big(& (a \wedge \text{eventually}(\text{frontier}(X) \wedge \neg a)) \\ & \vee (b \wedge \text{eventually}(\text{frontier}(X) \wedge \neg b)) \\ & \vee (c \wedge \text{eventually}(\text{frontier}(X) \wedge (a \vee b))\Big)\end{aligned}$$

Claim: $cxdy \models \psi \iff x \neq y$

Example: MIC on words

Consider words $cxdy$ with $x, y \in \{a, b\}^*$

c	b	b	a	a	d	b	a	a	a
			X	X				X	X

$$\psi := \mathbf{ifp}\, X \leftarrow (c \wedge \text{eventually}(\varphi)) \vee (\neg c \wedge \Box(X \vee d))$$

$$\text{eventually}(\alpha) \;:=\; \mathbf{ifp}\, X \leftarrow (\alpha \vee \Diamond X)$$

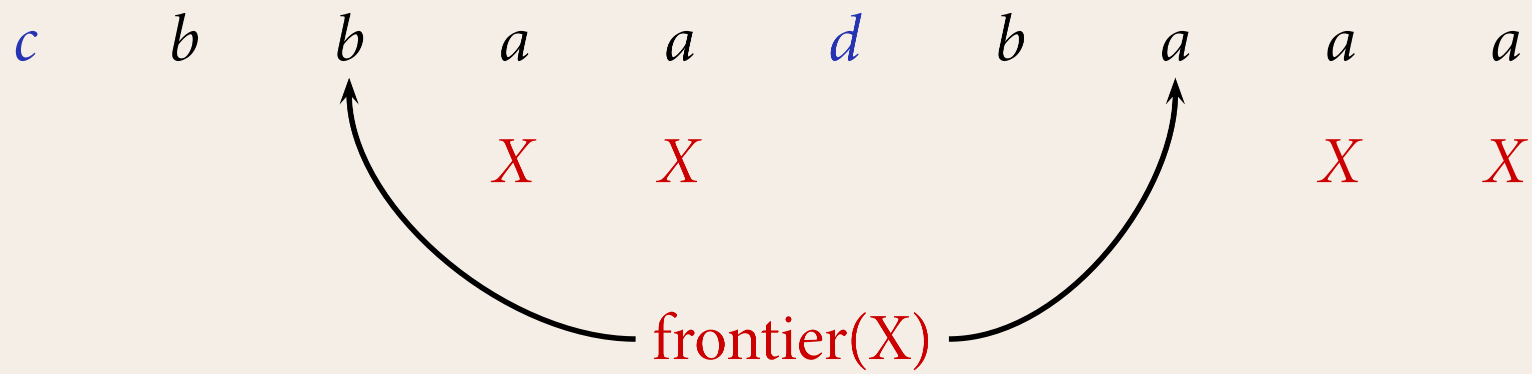
$$\text{frontier}(X) \;:=\; \neg X \wedge \Box(X \vee d)$$

$$\varphi := \text{frontier}(X) \wedge \big((a \wedge \text{eventually}(\text{frontier}(X) \wedge \neg a)) \vee (b \wedge \text{eventually}(\text{frontier}(X) \wedge \neg b)) \vee (c \wedge \text{eventually}(\text{frontier}(X) \wedge (a \vee b)) \big)$$

Claim: $cxdy \models \psi \iff x \neq y$

Example: MIC on words

Consider words $cxdy$ with $x, y \in \{a, b\}^*$



$$\psi := \mathbf{ifp}\, X \leftarrow (c \wedge \text{eventually}(\varphi)) \vee (\neg c \wedge \Box(X \vee d))$$

$$\text{eventually}(\alpha) \;:=\; \mathbf{ifp}\, X \leftarrow (\alpha \vee \Diamond X)$$

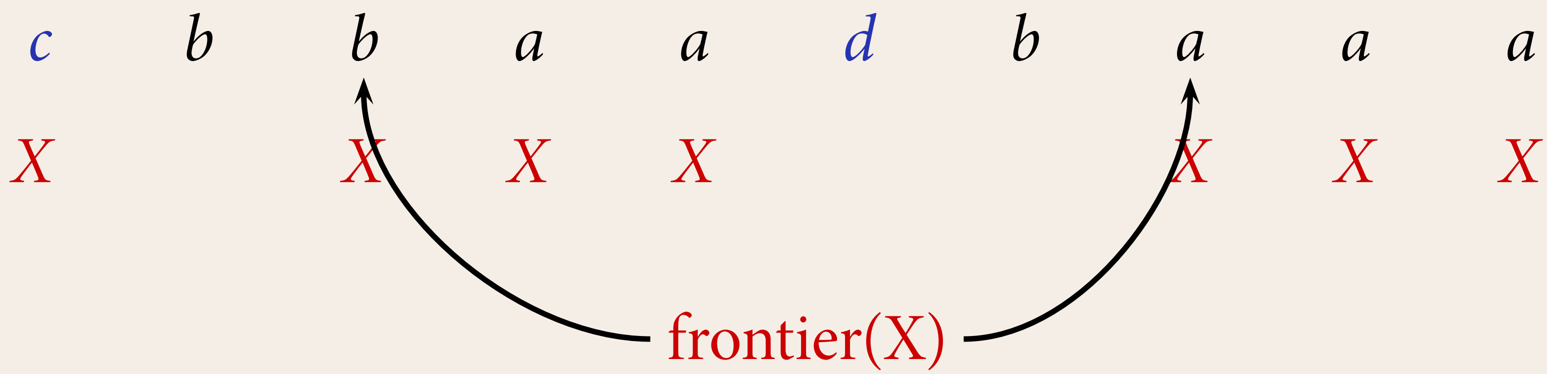
$$\text{frontier}(X) \;:=\; \neg X \wedge \Box(X \vee d)$$

$$\begin{aligned}\varphi := \text{frontier}(X) \wedge \Big(& (a \wedge \text{eventually}(\text{frontier}(X) \wedge \neg a)) \\ & \vee (b \wedge \text{eventually}(\text{frontier}(X) \wedge \neg b)) \\ & \vee (c \wedge \text{eventually}(\text{frontier}(X) \wedge (a \vee b)) \Big)\end{aligned}$$

Claim: $cxdy \models \psi \iff x \neq y$

Example: MIC on words

Consider words $cxdy$ with $x, y \in \{a, b\}^*$



$$\psi := \mathbf{ifp}\, X \leftarrow (c \wedge \text{eventually}(\varphi)) \vee (\neg c \wedge \Box(X \vee d))$$

$$\text{eventually}(\alpha) \;:=\; \mathbf{ifp}\, X \leftarrow (\alpha \vee \Diamond X)$$

$$\text{frontier}(X) \;:=\; \neg X \wedge \Box(X \vee d)$$

$$\begin{aligned}\varphi := \text{frontier}(X) \wedge \big(& (a \wedge \text{eventually}(\text{frontier}(X) \wedge \neg a)) \\ & \vee (b \wedge \text{eventually}(\text{frontier}(X) \wedge \neg b)) \\ & \vee (c \wedge \text{eventually}(\text{frontier}(X) \wedge (a \vee b))\big)\end{aligned}$$

Claim: $cxdy \models \psi \iff x \neq y$

What does this example prove?

$L = \{cwdw : w \in \{a, b\}^*\}$ is MIC-definable.

But L is not regular, indeed not even context-free.

Hence L is not MSO-definable.

Corollary.

- MIC is strictly more expressive than L_μ
- MIC $\not\leq$ MSO

What does this example prove?

$L = \{cwdw : w \in \{a, b\}^*\}$ is MIC-definable.

But L is not regular, indeed not even context-free.

Hence L is not MSO-definable.

Corollary.

- MIC is strictly more expressive than L_μ
- MIC $\not\leq$ MSO

A similar construction on trees shows that iterated relativisation $(\mathbf{dfp}\, X \leftarrow \varphi|_X)$ is not MSO-definable (even with $\varphi \in$ ML).

This solves van Benthem's problem.

Theorem. Epistemic logic with repeated public announcement cannot be embedded into the μ-calculus.

Infinity axioms in MIC

Theorem MIC does not have the finite model property

Infinity axioms in MIC

Theorem MIC does not have the finite model property

Proof. Axiomatise (up to bisimulation) the trees of height ω
(the height of a node is the supremum of the heights of its children)

$$\text{Deflationary induction via } S := \begin{cases} X \leftarrow \Diamond X \vee (\Diamond \mathit{true} \wedge \Box \neg Y) \\ Y \leftarrow X \end{cases}$$

stages $i < \omega$: $X^i = \{v : \text{height}(v) \geq i\}$

$Y^i = X^{i-1}$

stage ω: $X^\omega = Y^\omega = \{v : \text{height}(v) \geq \omega\}$

stage $\omega + 1$: $X^{\omega+1} = Y^{\omega+1} = X^\omega$

The formula

$$\omega\text{-height} := (\mathbf{dfp}\, X : S) \wedge \neg\Diamond(\mathbf{dfp}\, X : S)$$

has only infinite models.

Satisfiability

Full arithmetic on heights of well-founded trees can be interpreted in MIC:

take tree T of height ω, represent $s \in \omega$ by $S = \{v \in T : \text{height}(v) < s\}$.

construct MIC-formulae $\text{plus}(S, T)$ and $\text{times}(S, T)$ to encode arithmetic on heights

Theorem. The first-order theory of $(\omega, +, \cdot)$ reduces to Sat(MIC).
Hence Sat(MIC) is undecidable, in fact Σ^1_1-hard.

Satisfiability

Full arithmetic on heights of well-founded trees can be interpreted in MIC:

take tree T of height ω, represent $s \in \omega$ by $S = \{v \in T : \text{height}(v) < s\}$.

construct MIC-formulae plus(S, T) and times(S, T) to encode arithmetic on heights

Theorem. The first-order theory of $(\omega, +, \cdot)$ reduces to Sat(MIC). Hence Sat(MIC) is undecidable, in fact Σ^1_1-hard.

A stronger result:

Theorem. (Miller, Moss)
Epistemic logic with repeated public announcement is Σ^1_1-hard, even in the absence of common knowledge.

Model checking for MIC

Given finite $\mathcal{K}$ and $\psi \in \text{MIC}$, decide whether $\mathcal{K}, v \models \psi$

naive bottom evaluation:

if $\mathcal{K}$ has n nodes,

ψ has d nested simultaneous fixed points of width $\leq k$

then $(kn)^d$ iterations and time $O(|\psi| \cdot \|\mathcal{K}\|)$ per iteration suffice

Model checking for MIC

Given finite $\mathcal{K}$ and $\psi \in \text{MIC}$, decide whether $\mathcal{K}, v \models \psi$

naive bottom evaluation:

if $\quad \mathcal{K}$ has n nodes,

$\quad \psi$ has d nested simultaneous fixed points of width $\leq k$

then $(kn)^d$ iterations and time $O(|\psi| \cdot \|\mathcal{K}\|)$ per iteration suffice

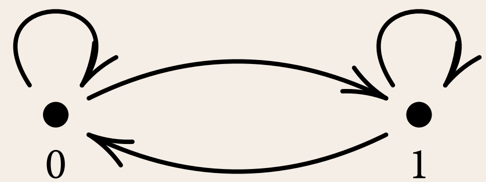
Complexity of MIC model checking:

- time $O((kn)^d \cdot |\psi| \cdot \|\mathcal{K}\|)$ and space $O(n \cdot |\psi|)$
- combined complexity: Pspace
- data complexity (ψ fixed): Ptime and linear space

Model checking for MIC

we can't really do much better than this naive algorithm

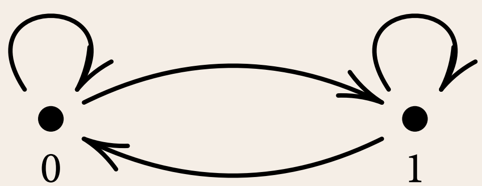
Theorem. The model checking problem for MIC is PSPACE-complete, even for formulae without simultaneous inductions, and on the fixed structure



Model checking for MIC

we can't really do much better than this naive algorithm

Theorem. The model checking problem for MIC is PSPACE-complete, even for formulae without simultaneous inductions, and on the fixed structure



Recall.

The model checking problem for L_μ is known to be in NP $\cap$ Co-NP, and conjectured to be in PTIME.

MIC versus PTIME

Theorem. (Otto)

multi-dimensional L_μ captures bisimulation-invariant PTIME

MIC versus PTIME

Theorem. (Otto)

multi-dimensional L_μ captures bisimulation-invariant PTIME

Theorem. MIC $\subsetneq$ bisimulation-invariant PTIME

MIC versus PTIME

Theorem. (Otto)
multi-dimensional L_μ captures bisimulation-invariant PTIME

Theorem. MIC $\subsetneq$ bisimulation-invariant PTIME

Proof. Let $S = \{T : T$ finite tree, all children of root are bisimilar$\}$

Clearly, $S \in$ PTIME, and S invariant under bisimulation (on trees).

for every $\varphi \in$ MIC define equivalence relation on trees:

$T_1 \sim_\varphi T_2$: at all stages, same subformulae are true on both trees

MIC versus PTIME

Theorem. (Otto)
multi-dimensional L_μ captures bisimulation-invariant PTIME

Theorem. MIC $\subsetneq$ bisimulation-invariant PTIME

Proof. Let $S = \{T : T$ finite tree, all children of root are bisimilar$\}$

Clearly, $S \in$ PTIME, and S invariant under bisimulation (on trees).

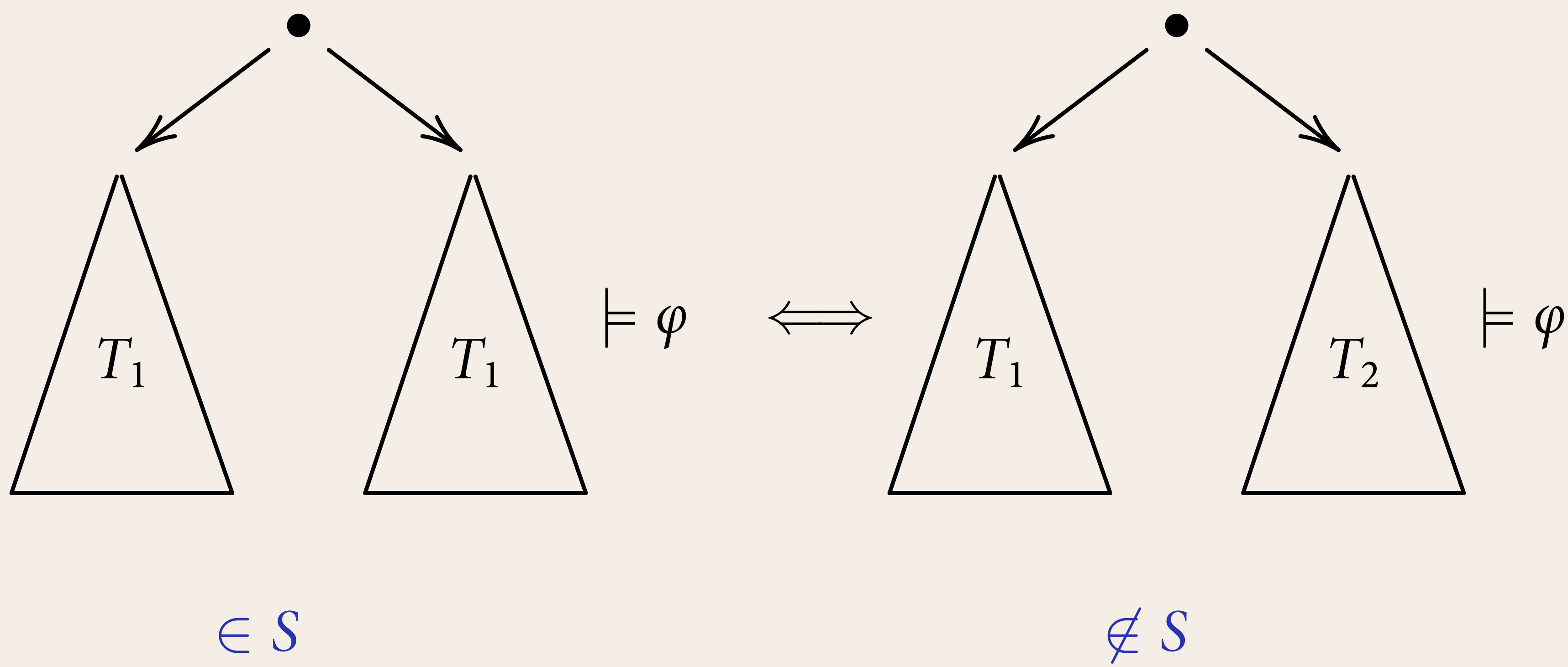
for every $\varphi \in$ MIC define equivalence relation on trees:

$T_1 \sim_\varphi T_2$: at all stages, same subformulae are true on both trees

On trees of height n, we have an exponential number of $\sim_\varphi$-types, but a non-elementary number of bisimulation types.

MIC versus PTIME

For every $\varphi \in \text{MIC}$, we can find trees $T_1 \not\sim T_2$ with $T_1 \sim_\varphi T_2$



Hence φ does not define S.

Further results on MIC

- Simultaneous inductions in MIC cannot be eliminated without losing expressive power (the Bekic principle fails in both directions)

Further results on MIC

- Simultaneous inductions in MIC cannot be eliminated without losing expressive power (the Bekic principle fails in both directions)
- Nevertheless, MIC without simultaneous inductions has no better algorithmic properties than full MIC.

Further results on MIC

- Simultaneous inductions in MIC cannot be eliminated without losing expressive power (the Bekic principle fails in both directions)
- Nevertheless, MIC without simultaneous inductions has no better algorithmic properties than full MIC.
- Expressive power on words:
  Every language in $\mathrm{DTIME}(O(n))$ is MIC-definable

Model checking games

Reduce model checking problem $\mathfrak{A} \models \psi$ to strategy problem for model checking game $G(\mathfrak{A}, \psi)$, played by

- Falsifier (also called Player 1), and
- Verifier (also called Player 0), such that

$$\mathfrak{A} \models \psi \iff \text{Verifier has winning strategy for } G(\mathfrak{A}, \psi)$$

Model checking games

Reduce model checking problem $\mathfrak{A} \models \psi$ to strategy problem for model checking game $G(\mathfrak{A}, \psi)$, played by

- Falsifier (also called Player 1), and
- Verifier (also called Player 0), such that

$$\mathfrak{A} \models \psi \iff \text{Verifier has winning strategy for } G(\mathfrak{A}, \psi)$$

$\Longrightarrow$ Model checking via construction of winning strategies

Model checking games

Reduce model checking problem $\mathfrak{A} \models \psi$ to strategy problem for model checking game $G(\mathfrak{A}, \psi)$, played by

- Falsifier (also called Player 1), and
- Verifier (also called Player 0), such that

$$\mathfrak{A} \models \psi \iff \text{Verifier has winning strategy for } G(\mathfrak{A}, \psi)$$

$\Longrightarrow$ Model checking via construction of winning strategies

Model checking games for L_μ (and LFP) are parity games

Parity games

$G = (V, E, \Omega)$, $V = V_0 \cup V_1$, $\Omega : V \longrightarrow \mathbb{N}$

$\Omega(v)$ is the priority of position v

Parity games

$G = (V, E, \Omega)$, $\quad V = V_0 \cup V_1$, $\quad \Omega : V \longrightarrow \mathbb{N}$

$\Omega(v)$ is the priority of position v

Play: finite or infinite sequence $\pi = v_0 v_1 v_2 \cdots$ with $(v_i, v_{i+1}) \in E$

Parity games

$G = (V, E, \Omega)$, $\quad V = V_0 \cup V_1$, $\quad \Omega : V \longrightarrow \mathbb{N}$

$\Omega(v)$ is the priority of position v

Play: finite or infinite sequence $\pi = v_0 v_1 v_2 \cdots$ with $(v_i, v_{i+1}) \in E$

Winning condition:

– finite plays: who cannot move, loses

– infinite plays: least priority seen infinitely often determines winner

$$\text{Player 0 wins } \pi \iff \min\{k : (\exists^\infty i)\Omega(v_i) = k\} \text{ is even}$$

Parity games

$G = (V, E, \Omega), \qquad V = V_0 \cup V_1, \qquad \Omega : V \longrightarrow \mathbb{N}$

$\Omega(v)$ is the priority of position v

Play: finite or infinite sequence $\pi = v_0 v_1 v_2 \cdots$ with $(v_i, v_{i+1}) \in E$

Winning condition:

– finite plays: who cannot move, loses

– infinite plays: least priority seen infinitely often determines winner

$$\text{Player 0 wins } \pi \iff \min\{k : (\exists^\infty i)\Omega(v_i) = k\} \text{ is even}$$

Reduce model checking problem $\mathfrak{A} \models \psi$ (for $\psi \in$ LFP)

to parity game $\mathcal{G}(\mathfrak{A}, \psi)$.

Positions: $\varphi(\overline{a})$ where $\varphi(\overline{x})$ subformula of ψ, $\quad \overline{a} \in \mathfrak{A}$

Priorities: Alternation depth of fixed points

Complexity of parity games

Theorem. The problem, which player wins from a given position in a parity game, is in NP $\cap$ Co-NP (in fact, in UP $\cap$ Co-UP).

The best known deterministic algorithms are polynomial in the size of the game graph, but exponential in the number of priorities.

Conjecture. The following problems are solvable in polynomial time:

(1) computing winning regions in parity games

(2) model checking for modal μ-calculus

(3) model checking for LFP-formulae of width k (for any $k \geq 2$)

Complexity of parity games

Theorem. The problem, which player wins from a given position in a parity game, is in NP $\cap$ Co-NP (in fact, in UP $\cap$ Co-UP).

The best known deterministic algorithms are polynomial in the size of the game graph, but exponential in the number of priorities.

Conjecture. The following problems are solvable in polynomial time:

(1) computing winning regions in parity games

(2) model checking for modal μ-calculus

(3) model checking for LFP-formulae of width k (for any $k \geq 2$)

If one of these problems admits a polynomial time algorithm, then all of them do.

What are the model checking games for MIC and IFP ?

Trivial solutions: games for corresponding infinitary formulae make explicit use of ordinals.

What are the model checking games for MIC and IFP ?

Trivial solutions: games for corresponding infinitary formulae
make explicit use of ordinals.

Drawback: Explosion of game graph

We want to use (essentially) the same game graph as for LFP- or L_μ-games,
but modify the winning condition.

What are the model checking games for MIC and IFP ?

Trivial solutions: games for corresponding infinitary formulae make explicit use of ordinals.

Drawback: Explosion of game graph

We want to use (essentially) the same game graph as for LFP- or L_μ-games, but modify the winning condition.

Parity games with backtracking and counting.

- Backtracking. Under certain conditions, a player can backtrack from current position v to a previously seen position of the same priority.

- Winning condition. After backtracking at priority p, a counting condition on number of nodes with priority p applies.

The winner of infinite plays is determined by the parity condition.

Conclusion

- IFP and LFP have the same expressive power, and similar algorithmic properties

  The translation from IFP to LFP is not simple

  Different structural properties (alternation hierarchy, nesting, . . .)

Conclusion

- IFP and LFP have the same expressive power, and similar algorithmic properties

  The translation from IFP to LFP is not simple

  Different structural properties (alternation hierarchy, nesting, . . .)

- MIC is much more expressive than L_μ

  Different structural properties (simultaneous vs. simple inductions)

  Interesting fragment of MIC for reasoning about knowledge.

  MIC is algorithmically much less manageable than L_μ:
  not applicable in areas where fast algorithms are needed

Conclusion

- IFP and LFP have the same expressive power, and similar algorithmic properties

  The translation from IFP to LFP is not simple

  Different structural properties (alternation hierarchy, nesting, . . .)

- MIC is much more expressive than L_μ

  Different structural properties (simultaneous vs. simple inductions)

  Interesting fragment of MIC for reasoning about knowledge.

  MIC is algorithmically much less manageable than L_μ:
  not applicable in areas where fast algorithms are needed

- Is there an interesting game theory for inflationary fixed points ?