Model Checking for the μ-calculus

Paolo Zuliani

15-817, Spring 2011

Outline

- What is the μ-calculus?
- Semantics
- Model Checking algorithms
- [Other fixpoint theorems]

The μ-calculus

- A language for describing properties of transition systems
- It uses least and greatest fixpoint operators
  - μ (least fixpoint)
  - ν (greatest fixpoint)
- It subsumes many temporal logics
  - CTL* can be translated into the μ-calculus

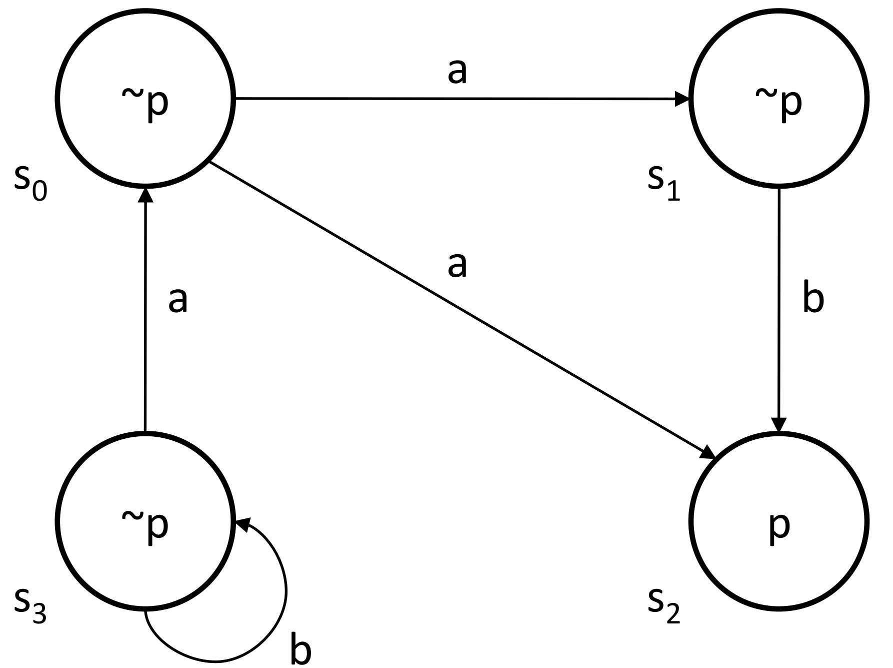
The μ-calculus

- More expressive than temporal logics
  - See last lecture on Data Flow Analysis, but also
  - *Even*(*p*) = "*p* must happen every two steps (*p* can happen or not in other steps)" along a given path (Wolper, 1981)
  - *Even*(*p*) **cannot** be expressed in temporal logics
  - *Even*(*p*) **can be** expressed in the μ-calculus (later)
- There are efficient Model Checking algorithms
- Formulae evaluate to **sets of states**

Semantics

- Given wrt **modified Kripke structures**, that is, Kripke structures with **labels on transitions**

- Example:
  - s_0, s_1, s_2, s_3 states
  - p atomic prop.
  - a, b transitions



Semantics

- A **modified Kripke structure** $M = (S,T,L)$ consists of
  - a nonempty set of states S,
  - a set of transitions T, such that for each transition $a \in T$, $a \subseteq S \times S$, and
  - a mapping $L : S \rightarrow 2^{AP}$ that gives the set of atomic propositions true in a state.
- $VAR = \{Q, Q_1, Q_2, \ldots\}$ a set of **relational variables**
- Each relational variable $Q \in VAR$ can be assigned a subset of S

Syntax

- If $p \in AP$, then p is a formula
- A relational variable is a formula
- If f, g formulas, then **$\neg f$, $f \wedge g$** and **$f \vee g$ formulas**
- If f is a formula, and $a \in T$, then **$[a]f$** and **$\langle a \rangle f$** are formulas
- For $Q \in VAR$ and formula f, then **$\mu Q.f$** and **$\nu Q.f$** are formulas
  - provided that f is **syntactically monotone** in Q, *i.e.*, all occurrences of Q within f fall under an even number of negations

Syntax

- Two **modalities** – their informal meaning is

  $[a]\, f$ = "f holds in **all states** reachable by one step of transition a"

  $\langle a \rangle\, f$ = "f holds in **a state** reachable by one step of transition a"

Syntax

- Two **modalities** – their informal meaning is

  $[a]\,f$ = "*f* holds in **all states** reachable by one step of transition *a*"

  $\langle a \rangle\, f$ = "*f* holds in **a state** reachable by one step of transition *a*"

- Example (suppose only one transition *a*):
  - $Even(p) = \nu Q.(p \wedge \langle a\rangle\langle a\rangle Q)$ (along a path)

Syntax

- Two **modalities** – their informal meaning is

  $[a]\,f$ = "*f* holds in **all states** reachable by one step of transition *a*"

  $\langle a \rangle f$ = "*f* holds in **a state** reachable by one step of transition *a*"

- Example (suppose only one transition *a*):
  - $Even(p) = \nu Q.(p \wedge \langle a \rangle \langle a \rangle Q)$ (along a path)
  - $\mathbf{E}[p\ \mathbf{U}\ q] = \mu Q.(p \wedge (q \vee \langle a \rangle Q))$ (over a Kripke str.)

Semantics

- Given a modified Kripke structure M
- VAR = $\{Q, Q_1, Q_2, . . .\}$ a set of **relational variables**
- An **environment** $e : \mathrm{VAR} \rightarrow 2^S$
- The semantics $[\![f]\!]_M e$ of a formula f is the "**set of states in which *f* is true**"

Semantics

- Given a modified Kripke structure M
- VAR = $\{Q, Q_1, Q_2, . . .\}$ a set of **relational variables**
- An **environment** $e : VAR \rightarrow 2^S$
- The semantics $[\![f]\!]_M e$ of a formula *f* is the "**set of states in which *f* is true**"
- We denote
  - S=*True* (formula *True* holds for all states)
  - Ø=*False* (formula *False* holds for no state)
  - $e[Q \leftarrow W]$ is the environment equal to *e*, except that $(e[Q \leftarrow W])(Q) = W$

Semantics

- The order on 2^S is given by set inclusion
- The set $[\![f]\!]e$ is defined recursively as follows:
- $[\![p]\!]e = \{s \mid p \in L(s)\}$
- $[\![Q]\!]e = e(Q)$
- $[\![\neg f]\!]e = S \setminus [\![f]\!]e$
- $[\![f \wedge g]\!]e = [\![f]\!]e \cap [\![g]\!]e$
- $[\![f \vee g]\!]e = [\![f]\!]e \cup [\![g]\!]e$

Semantics

- $[\![\langle a \rangle f]\!] e = \{s \mid \exists t\ (s,t) \in a$ and $t \in [\![f]\!] e\}$
- $[\![[a] f]\!] e = \{s \mid \forall t\ (s,t) \in a$ implies $t \in [\![f]\!] e\}$

Semantics

- $[\![\langle a \rangle f]\!] e = \{s \mid \exists t\ (s,t) \in a$ and $t \in [\![f]\!] e\}$
- $[\![[a] f]\!] e = \{s \mid \forall t\ (s,t) \in a$ implies $t \in [\![f]\!] e\}$

- $[\![\mu Q.f]\!] e$ is the **least fixpoint** of the predicate transformer $\tau: 2^S \to 2^S$ defined by:

  $\tau(W) = [\![f]\!] (e[Q \leftarrow W])$

Semantics

- $[\![\langle a \rangle f]\!] e = \{s \mid \exists t\ (s,t) \in a$ and $t \in [\![f]\!] e\}$
- $[\![[a] f]\!] e = \{s \mid \forall t\ (s,t) \in a$ implies $t \in [\![f]\!] e\}$

- $[\![\mu Q.f]\!] e$ is the **least fixpoint** of the predicate transformer $\tau: 2^S \rightarrow 2^S$ defined by:

  $\tau(W) = [\![f]\!] (e[Q \leftarrow W])$

- $[\![\nu Q.f]\!] e$ is the **greatest fixpoint** of τ above

Semantics

- All logical connectives and modalities (except negation) are **monotonic**
- Example: conjunction

$$[\![f]\!]e \subseteq [\![f']\!]e \;\Rightarrow\; [\![f \wedge g]\!]e \subseteq [\![f' \wedge g]\!]e$$

$$(\, A \subseteq B \quad \Rightarrow \quad A \cap C \subseteq B \cap C \,)$$

Semantics

- Negations can be pushed down to atomic propositions by De Morgan's laws and
  - $\neg\ [a]\ f \equiv \langle a \rangle\ \neg f$
  - $\neg\ \langle a \rangle\ f \equiv [a]\ \neg f$
  - $\neg\ \mu Q.f(Q) \equiv \nu Q.\neg f(\neg Q)$
  - $\neg\ \nu Q.f(Q) \equiv \mu Q.\neg f(\neg Q)$
- Variables appear under an even number of negations
- By applying the rules above, variables will be **negation-free**

Semantics

- Therefore, in a fixpoint formula we can only define monotonic operators
- Therefore, **fixpoints exist**! (Tarski)

Semantics

- Therefore, in a fixpoint formula we can only define monotonic operators
- Therefore, **fixpoints exist**! (Tarski)
- Furthermore, we assume that **S is finite**, so we can effectively compute the fixpoints

$$[\![\mu Q.f]\!] e = \cup_i \tau^i(\mathit{False})$$

$$[\![\nu Q.f]\!] e = \cap_i \tau^i(\mathit{True})$$

- Recall that $[\![\mu Q.f]\!] e = \mathrm{lfp}(\tau)$ where $\tau(W) = [\![f]\!](e[Q \leftarrow W])$

Model Checking: a naïve algorithm

function eval(f, e)

if $f = p$ **then return** $\{s \mid p \in L(s)\}$;
if $f = Q$ **then return** $e(Q)$;
if $f = g_1 \wedge g_2$ **then return** $\text{eval}(g_1, e) \cap \text{eval}(g_2, e)$;
if $f = g_1 \vee g_2$ **then return** $\text{eval}(g_1, e) \cup \text{eval}(g_2, e)$;

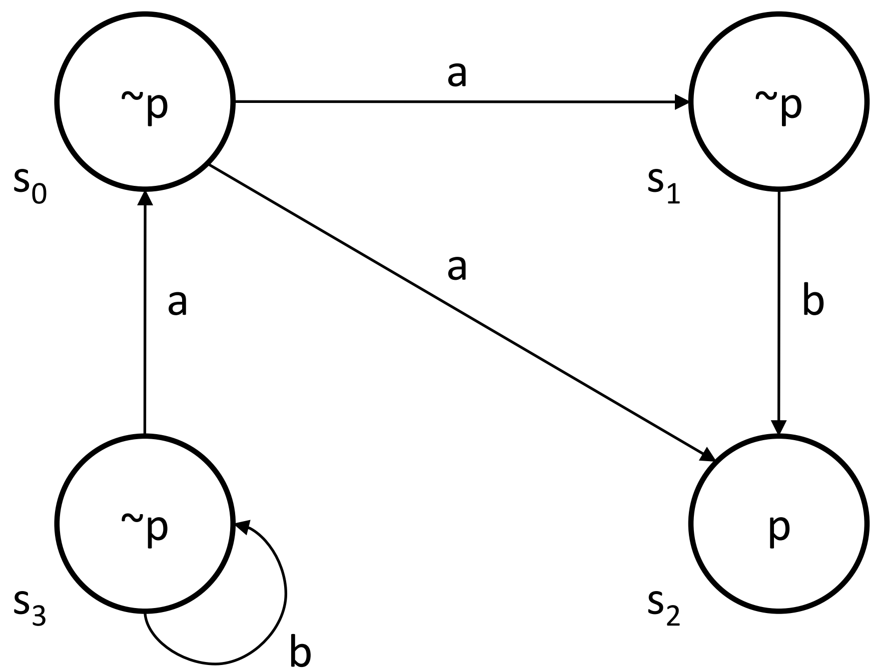
if $f = \langle a \rangle g$ **then return** $\{\, s \mid \exists t\, [(s,t) \in a \text{ and } t \in \text{eval}(g, e)]\,\}$;
if $f = [a]g$ **then return** $\{\, s \mid \forall t\, [(s,t) \in a \text{ implies } t \in \text{eval}(g, e)]\,\}$;

if $f = \mu Q.g(Q)$ **then**
    $Q_{\text{val}} := \mathit{False}$;
    repeat
        $Q_{\text{old}} := Q_{\text{val}}$;
        $Q_{\text{val}} := \text{eval}(g, e\,[Q \leftarrow Q_{\text{val}}])$;
    until $Q_{\text{val}} = Q_{\text{old}}$;
    return Q_{val};
end if;

end function

Model Checking: example

- Calculate $[\![\nu Q.(p \vee \langle b \rangle Q)]\!] e$ on the Kripke structure



$[\![\nu Q.(p \vee \langle b \rangle Q)]\!] e$ is gfp of $\tau(W) = [\![p \vee \langle b \rangle Q]\!](e[Q \leftarrow W])$

- Start iterating τ from *True* (the entire state space S)

$\tau^1(\mathit{True}) = [\![p \vee \langle b \rangle Q]\!](e[Q \leftarrow \mathit{True}])$

$[\![\nu Q.(p \vee \langle b \rangle Q)]\!] e$ is gfp of $\tau(W) = [\![p \vee \langle b \rangle Q]\!](e[Q \leftarrow W])$

- Start iterating τ from *True* (the entire state space S)

$$\tau^1(True) = [\![p \vee \langle b \rangle Q]\!](e[Q \leftarrow True])$$
$$= [\![p]\!](e[Q \leftarrow S]) \cup [\![\langle b \rangle Q]\!](e[Q \leftarrow S])$$

$[\![\nu Q.(p \vee \langle b \rangle Q)]\!]e$ is gfp of $\tau(W) = [\![p \vee \langle b \rangle Q]\!](e[Q \leftarrow W])$

- Start iterating τ from *True* (the entire state space S)

$$\tau^1(\mathit{True}) = [\![p \vee \langle b \rangle Q]\!](e[Q \leftarrow \mathit{True}])$$
$$= [\![p]\!](e[Q \leftarrow S]) \cup [\![\langle b \rangle Q]\!](e[Q \leftarrow S])$$
$$= \{s_2\} \cup \{s \mid \exists t\ (s,t) \in b \text{ and } t \in ([\![Q]\!]e[Q \leftarrow S])\}$$

$[\![\nu Q.(p \vee \langle b\rangle Q)]\!]e$ is gfp of $\tau(W) = [\![p \vee \langle b\rangle Q]\!](e[Q \leftarrow W])$

- Start iterating τ from *True* (the entire state space S)

$$
\begin{aligned}
\tau^1(\mathit{True}) &= [\![p \vee \langle b\rangle Q]\!](e[Q \leftarrow \mathit{True}]) \\
&= [\![p]\!](e[Q \leftarrow S]) \cup [\![\langle b\rangle Q]\!](e[Q \leftarrow S]) \\
&= \{s_2\} \cup \{s \mid \exists t\ (s,t) \in b \ \text{ and } \ t \in ([\![Q]\!]e[Q \leftarrow S])\} \\
&= \{s_2\} \cup \{s \mid \exists t\ (s,t) \in b \ \text{ and } \ t \in S\}
\end{aligned}
$$

$[\![\nu Q.(p \vee \langle b \rangle Q)]\!] e$ is gfp of $\tau(W) = [\![p \vee \langle b \rangle Q]\!](e[Q \leftarrow W])$

- Start iterating τ from *True* (the entire state space S)

$$
\begin{aligned}
\tau^1(\mathit{True}) &= [\![p \vee \langle b \rangle Q]\!](e[Q \leftarrow \mathit{True}]) \\
&= [\![p]\!](e[Q \leftarrow S]) \cup [\![\langle b \rangle Q]\!](e[Q \leftarrow S]) \\
&= \{s_2\} \cup \{s \mid \exists t\ (s,t) \in b \text{ and } t \in ([\![Q]\!] e[Q \leftarrow S])\} \\
&= \{s_2\} \cup \{s \mid \exists t\ (s,t) \in b \text{ and } t \in S\} \\
&= \{s_2\} \cup \{s_1, s_3\} = \{s_1, s_2, s_3\}
\end{aligned}
$$

$[\![\nu Q.(p \vee \langle b \rangle Q)]\!] e$ is gfp of $\tau(W) = [\![p \vee \langle b \rangle Q]\!](e[Q \leftarrow W])$

- Start iterating τ from *True* (the entire state space S)

$$\tau^1(True) = [\![p \vee \langle b \rangle Q]\!](e[Q \leftarrow True])$$
$$= [\![p]\!](e[Q \leftarrow S]) \cup [\![\langle b \rangle Q]\!](e[Q \leftarrow S])$$
$$= \{s_2\} \cup \{s \mid \exists t\ (s,t) \in b \ \text{ and } \ t \in ([\![Q]\!] e[Q \leftarrow S])\}$$
$$= \{s_2\} \cup \{s \mid \exists t\ (s,t) \in b \ \text{ and } \ t \in S\}$$
$$= \{s_2\} \cup \{s_1, s_3\} = \{s_1, s_2, s_3\}$$

$$\tau^2(True) = \tau(\tau(True)) = \tau(\{s_1, s_2, s_3\})$$

$[\![\nu Q.(p \vee \langle b \rangle Q)]\!]e$ is gfp of $\tau(W) = [\![p \vee \langle b \rangle Q]\!](e[Q \leftarrow W])$

- Start iterating τ from *True* (the entire state space S)

$$\tau^1(True) = [\![p \vee \langle b \rangle Q]\!](e[Q \leftarrow True])$$
$$= [\![p]\!](e[Q \leftarrow S]) \cup [\![\langle b \rangle Q]\!](e[Q \leftarrow S])$$
$$= \{s_2\} \cup \{s \mid \exists t\ (s,t) \in b \text{ and } t \in ([\![Q]\!]e[Q \leftarrow S])\}$$
$$= \{s_2\} \cup \{s \mid \exists t\ (s,t) \in b \text{ and } t \in S\}$$
$$= \{s_2\} \cup \{s_1, s_3\} = \{s_1, s_2, s_3\}$$

$$\tau^2(True) = \tau(\tau(True)) = \tau(\{s_1, s_2, s_3\})$$
$$= \{s_2\} \cup \{s_1, s_3\} = \{s_1, s_2, s_3\}$$

Complexity of Model Checking

- Calculate $[\![\mu X.\ \mu Y.\ \tau\ (X,Y)]\!]e$ (τ is ∪-cont.)
- Define $\zeta(X) = \mu Y.\ \tau\ (X,Y)$ so that

$$[\![\mu X.\ \mu Y.\ \tau\ (X,Y)]\!]e = [\![\mu X.\ \zeta(X)]\!]e$$

Complexity of Model Checking

- Calculate $[\![\mu X.\ \mu Y.\ \tau\ (X,Y)]\!]e$ (τ is ∪-cont.)
- Define $\zeta(X) = \mu Y.\ \tau\ (X,Y)$ so that

$$[\![\mu X.\ \mu Y.\ \tau\ (X,Y)]\!]e = [\![\mu X.\ \zeta(X)]\!]e$$

- Now: iterate $\zeta(\mathit{False})$ until $\zeta^{i}(\mathit{False}) = \zeta^{i+1}(\mathit{False})$

$$\zeta^{i+1}(\mathit{False}) = \mu Y.\ \tau\ (\zeta^{i}(\mathit{False}),Y)$$

Complexity of Model Checking

- Calculate $[\![\mu X.\ \mu Y.\ \tau\,(X,Y)]\!]e$ (τ is ∪-cont.)
- Define $\zeta(X) = \mu Y.\ \tau\,(X,Y)$ so that

$$[\![\mu X.\ \mu Y.\ \tau\,(X,Y)]\!]e = [\![\mu X.\ \zeta(X)]\!]e$$

- Now: iterate $\zeta(\mathit{False})$ until $\zeta^i(\mathit{False}) = \zeta^{i+1}(\mathit{False})$

$$\zeta^{i+1}(\mathit{False}) = \mu Y.\ \tau\,(\zeta^i(\mathit{False}),Y)$$

- Iterate $\tau(\zeta^i(\mathit{False}),\mathit{False})$ until

$$\tau^j(\zeta^i(\mathit{False}),\mathit{False}) = \tau^{j+1}(\zeta^i(\mathit{False}),\mathit{False})$$

Complexity of Model Checking

- Calculate $[\![\mu X.\ \mu Y.\ \tau\ (X,Y)]\!]e$ (τ is ∪-cont.)
- Define $\zeta(X) = \mu Y.\ \tau\ (X,Y)$ so that

$$[\![\mu X.\ \mu Y.\ \tau\ (X,Y)]\!]e = [\![\mu X.\ \zeta(X)]\!]e$$

- Now: iterate $\zeta(\mathit{False})$ until $\zeta^i(\mathit{False}) = \zeta^{i+1}(\mathit{False})$

$$\zeta^{i+1}(\mathit{False}) = \mu Y.\ \tau\ (\zeta^i(\mathit{False}),Y)$$

- Iterate $\tau(\zeta^i(\mathit{False}),\mathit{False})$ until

$$\tau^j(\zeta^i(\mathit{False}),\mathit{False}) = \tau^{j+1}(\zeta^i(\mathit{False}),\mathit{False})$$

- Overall, we need **$O(|S|^2)$** iterations of τ
  - A formula with **k** nested fixpoint operators needs **$O(|S|^k)$** iterations of the innermost fixpoint transformer

Faster Model Checking

- **Key idea**: nested fixpoints of the **same type** do not need re-initialization to *False* (or *True*)
- Need to define **alternation depth** of a formula
  - "number of alternations of μ and ν operators"

Faster Model Checking

- **Key idea**: nested fixpoints of the **same type** do not need re-initialization to *False* (or *True*)
- Need to define **alternation depth** of a formula
  - "number of alternations of μ and ν operators"
- A **top-level ν-subformula** of f is a subformula $\nu Q.g$ of f not contained in any other ν-subformula of f
- Example: $f = \mu Q.(\nu Q_1.g_1 \lor \nu Q_2.g_2)$
  - $\nu Q_1.g_1$ and $\nu Q_2.g_2$ are ν-subformulae of f

Alternation Depth

- If f contains **subsentences $w_1, \ldots, w_n$** then
  - AD(f) = max(AD(w_1), …, AD(w_n), AD(f')) where f' is obtained from f by substitution new constants $c_1, \ldots, c_n$ for $w_1, \ldots, w_n$
- The AD of **atomic propositions** or **relational variables** is 0
- The AD of $f \wedge g$, $f \vee g$, $\langle a \rangle f$, $[a]f$ is the maximum AD of subformulae f and g
- The AD of **$\mu Q.f$** is
  - max (AD(f), 1 + max (AD(f_1), …, AD(f_n)) where $f_1, \ldots, f_n$ are the top-level ν-subformulae of f

What is the Alternation Depth of

$\mu Q. (p \vee [a]Q) = 1$

What is the Alternation Depth of

$\mu Q. (p \vee [a]Q) = 1$

$\mu Q.(\nu Q_1.(p \vee \langle a \rangle Q_1) \vee [a]Q)$

What is the Alternation Depth of

$\mu Q. (p \vee [a]Q) = 1$

$\mu Q.(\nu Q_1.(p \vee \langle a \rangle Q_1) \vee [a]Q)$

What is the Alternation Depth of

$\mu Q.\ (p \vee [a]Q)\ = 1$

$\mu Q.(\nu Q_1.(p \vee \langle a \rangle Q_1) \vee [a]Q)$

max $(\boldsymbol{\nu Q_1.(p \vee \langle a \rangle Q_1)}, \mu Q.(\mathbf{X} \vee [a]Q),) = 1$

What is the Alternation Depth of

$\mu Q.\ (p \vee [a]Q)\ = 1$

$\mu Q.(\nu Q_1.(p \vee \langle a \rangle Q_1)) \vee [a]Q)$

max ($\boldsymbol{\nu Q_1.(p \vee \langle a \rangle Q_1)}$, $\mu Q.(\mathbf{X} \vee [a]Q),$) = 1

$\nu Q.\mu Q_1.\langle a \rangle(\nu Q_2.\mu Q_3.(\langle a \rangle(p \wedge Q_2) \vee Q_3)) \wedge Q) \vee Q_1)$

What is the Alternation Depth of

$\mu Q.\ (p \vee [a]Q)\ = 1$

$\mu Q.(\nu Q_1.(p \vee \langle a \rangle Q_1)) \vee [a]Q)$

max ($\boldsymbol{\nu Q_1.(p \vee \langle a \rangle Q_1)}$, $\mu Q.(\mathbf{X} \vee [a]Q),)$ = 1

$\nu Q.\mu Q_1.\langle a \rangle(\nu Q_2.\mu Q_3.(\langle a \rangle(p \wedge Q_2) \vee Q_3)) \wedge Q) \vee Q_1)$

What is the Alternation Depth of

$\mu Q.\ (p \vee [a]Q) = 1$

$\mu Q.(\nu Q_1.(p \vee \langle a \rangle Q_1)) \vee [a]Q)$

$\max(\boldsymbol{\nu Q_1.(p \vee \langle a \rangle Q_1)},\ \mu Q.(\mathbf{X} \vee [a]Q),) = 1$

$\nu Q.\mu Q_1.\langle a \rangle(\nu Q_2.\mu Q_3.(\langle a \rangle(p \wedge Q_2) \vee Q_3)) \wedge Q) \vee Q_1)$

$\max(\boldsymbol{\nu Q_2.\mu Q_3.(\langle a \rangle(p \wedge Q_2) \vee Q_3)},$
$\nu Q.\mu Q_1.\langle a \rangle(\mathbf{Y} \wedge Q) \vee Q_1) = 2$

Faster Model Checking

- E.A. Emerson and C.-L. Lei, LICS 1986
- Reset relational variables to *True* (*False*) only **when fixpoint operators alternate**
- Thus, need only **$O(|S|^d)$** iterations of the innermost fixpoint transformer, where d=AD(*f*)

Emerson and Lei’s algorithm

- ***Lemma***: Let $\tau: 2^S \to 2^S$ be monotonic (thus $\bigcup$- and $\bigcap$-continuous, since S finite). Then:

Emerson and Lei's algorithm

- ***Lemma***: Let $\tau: 2^S \rightarrow 2^S$ be monotonic (thus $\bigcup$- and $\bigcap$-continuous, since S finite). Then:
  - If $\mathbf{X} \subseteq \mu Q.\tau(Q)$ then $\mu Q.\tau(Q) = \bigcup_i \tau^i(\mathbf{X})$

Emerson and Lei's algorithm

- ***Lemma***: Let $\tau: 2^S \rightarrow 2^S$ be monotonic (thus $\bigcup$- and $\bigcap$-continuous, since S finite). Then:
  - If **X** $\subseteq \mu Q.\tau(Q)$ then $\mu Q.\tau(Q) = \bigcup_i \tau^i($**X**$)$
  - If **Y** $\supseteq \nu Q.\tau(Q)$ then $\nu Q.\tau(Q) = \bigcap_i \tau^i($**Y**$)$

Emerson and Lei's algorithm

- *Lemma*: Let $\tau: 2^S \rightarrow 2^S$ be monotonic (thus $\bigcup$- and $\bigcap$-continuous, since S finite). Then:
  - If **X** $\subseteq \mu Q.\tau(Q)$ then $\mu Q.\tau(Q) = \bigcup_i \tau^i($**X**$)$
  - If **Y** $\supseteq \nu Q.\tau(Q)$ then $\nu Q.\tau(Q) = \bigcap_i \tau^i($**Y**$)$
- "we can iterate from **any approximation** known to be below (above) the fixpoint"

Emerson and Lei's algorithm

- *Lemma*: Let $\tau: 2^S \rightarrow 2^S$ be monotonic (thus $\bigcup$- and $\bigcap$-continuous, since S finite). Then:
  - If **X** $\subseteq \mu Q.\tau(Q)$ then $\mu Q.\tau(Q) = \bigcup_i \tau^i($**X**$)$
  - If **Y** $\supseteq \nu Q.\tau(Q)$ then $\nu Q.\tau(Q) = \bigcap_i \tau^i($**Y**$)$
- "we can iterate from **any approximation** known to be below (above) the fixpoint"
- In particular

  $\tau(\mathit{False}) \subseteq \ldots \subseteq \tau^j(\mathit{False}) \subseteq \ldots \subseteq \bigcup_i \tau^i(\mathit{False}) = \mu Q.\tau(Q)$

Emerson and Lei's algorithm

- *Lemma*: Let $\tau: 2^S \to 2^S$ be monotonic (thus $\bigcup$- and $\bigcap$-continuous, since S finite). Then:
  - If **X** $\subseteq \mu Q.\tau(Q)$ then $\mu Q.\tau(Q) = \bigcup_i \tau^i($**X**$)$
  - If **Y** $\supseteq \nu Q.\tau(Q)$ then $\nu Q.\tau(Q) = \bigcap_i \tau^i($**Y**$)$
- "we can iterate from **any approximation** known to be below (above) the fixpoint"
- In particular

  $\tau(\mathit{False}) \subseteq \ldots \subseteq \boxed{\tau^j(\mathit{False})} \subseteq \ldots \subseteq \bigcup_i \tau^i(\mathit{False}) = \mu Q.\tau(Q)$

Emerson and Lei's algorithm

- Example: $\mu X.\mu Y.\ \tau\ (X,Y)$ (τ is monotonic)
- Let $\zeta(X) = \mu Y.\ \tau\ (X,Y)$ so $\mu X.\mu Y.\ \tau\ (X,Y) = \mu X.\ \zeta(X)$

Emerson and Lei's algorithm

- Example: $\mu X.\mu Y.\ \tau\ (X,Y)$ (τ is monotonic)
- Let $\zeta(X) = \mu Y.\ \tau\ (X,Y)$ so $\mu X.\mu Y.\ \tau\ (X,Y) = \mu X.\ \zeta(X)$

- The naïve algorithm:
- Iterate $\zeta(\mathit{False})$ until $\zeta^{i}(\mathit{False}) = \zeta^{i+1}(\mathit{False})$

  $$\zeta^{i+1}(\mathit{False}) = \mu Y.\ \tau\ (\zeta^{i}(\mathit{False}),Y)$$

- Iterate $\tau(\zeta^{i}(\mathit{False}),\mathit{False})$ until

  $$\tau^{j}(\zeta^{i}(\mathit{False}),\mathit{False}) = \tau^{j+1}(\zeta^{i}(\mathit{False}),\mathit{False})$$

- Need **$O(|S|^2)$** iterations of τ

Emerson and Lei's algorithm

- Example: $\mu X.\mu Y.\ \tau(X,Y)$ (τ is monotonic)
- $\zeta(X) = \mu Y.\ \tau(X,Y)$ $\mu X.\mu Y.\ \tau(X,Y) = \mu X.\ \zeta(X)$

Emerson and Lei's algorithm

- Example: $\mu X.\mu Y.\ \tau(X,Y)$ (τ is monotonic)
- $\zeta(X) = \mu Y.\ \tau(X,Y)$ $\mu X.\mu Y.\ \tau(X,Y) = \mu X.\ \zeta(X)$
- Note that $\zeta^{i-1}(\mathit{False}) \subseteq \zeta^{i}(\mathit{False})$ and τ $\bigcup$-continuous

Emerson and Lei's algorithm

- Example: $\mu X.\mu Y.\ \tau(X,Y)$ (τ is monotonic)
- $\zeta(X) = \mu Y.\ \tau(X,Y)$ $\mu X.\mu Y.\ \tau(X,Y) = \mu X.\ \zeta(X)$
- Note that $\zeta^{i-1}(\mathit{False}) \subseteq \zeta^{i}(\mathit{False})$ and τ $\bigcup$-continuous

$$\mu Y.\ \tau(\zeta^{i-1}(\mathit{False}),Y) \subseteq \mu Y.\ \tau(\zeta^{i}(\mathit{False}),Y) \quad (*)$$

Emerson and Lei's algorithm

- Example: $\mu X.\mu Y.\ \tau(X,Y)$ (τ is monotonic)
- $\zeta(X) = \mu Y.\ \tau(X,Y)$ $\mu X.\mu Y.\ \tau(X,Y) = \mu X.\ \zeta(X)$
- Note that $\zeta^{i-1}(\mathit{False}) \subseteq \zeta^{i}(\mathit{False})$ and τ $\bigcup$-continuous

$$\mu Y.\ \tau(\zeta^{i-1}(\mathit{False}),Y) \subseteq \mu Y.\ \tau(\zeta^{i}(\mathit{False}),Y) \quad (*)$$

$\zeta^{i+1}(\mathit{False}) = \mu Y.\ \tau(\zeta^{i}(\mathit{False}),Y)$

Emerson and Lei's algorithm

- Example: $\mu X.\mu Y.\ \tau\,(X,Y)$ (τ is monotonic)
- $\zeta(X) = \mu Y.\ \tau\,(X,Y)$ $\mu X.\mu Y.\ \tau\,(X,Y) = \mu X.\ \zeta(X)$
- Note that $\zeta^{i-1}(\mathit{False}) \subseteq \zeta^{i}(\mathit{False})$ and τ $\bigcup$-continuous

$$\mu Y.\ \tau\,(\zeta^{i-1}(\mathit{False}),Y) \subseteq \mu Y.\ \tau\,(\zeta^{i}(\mathit{False}),Y) \quad (*)$$

$$\zeta^{i+1}(\mathit{False}) = \mu Y.\ \tau\,(\zeta^{i}(\mathit{False}),Y) = \cup_j\ \tau^{j}(\zeta^{i}(\mathit{False}),\mathit{False})$$

Emerson and Lei's algorithm

- Example: $\mu X.\mu Y.\ \tau\,(X,Y)$ (τ is monotonic)
- $\zeta(X) = \mu Y.\ \tau\,(X,Y)$ $\mu X.\mu Y.\ \tau\,(X,Y) = \mu X.\ \zeta(X)$
- Note that $\zeta^{i-1}(\mathit{False}) \subseteq \zeta^{i}(\mathit{False})$ and τ $\bigcup$-continuous

$$\mu Y.\ \tau\,(\zeta^{i-1}(\mathit{False}),Y) \subseteq \boxed{\mu Y.\ \tau\,(\zeta^{i}(\mathit{False}),Y)} \quad (*)$$

$$\zeta^{i+1}(\mathit{False}) = \boxed{\mu Y.\ \tau\,(\zeta^{i}(\mathit{False}),Y)} = \cup_j\ \tau^{j}(\zeta^{i}(\mathit{False}),\mathit{False})$$

Emerson and Lei's algorithm

- Example: $\mu X.\mu Y.\ \tau\ (X,Y)$ (τ is monotonic)
- $\zeta(X) = \mu Y.\ \tau\ (X,Y)$ $\mu X.\mu Y.\ \tau\ (X,Y) = \mu X.\ \zeta(X)$
- Note that $\zeta^{i-1}(\mathit{False}) \subseteq \zeta^{i}(\mathit{False})$ and τ $\bigcup$-continuous

$$\mu Y.\ \tau\ (\zeta^{i-1}(\mathit{False}),Y) \subseteq \boxed{\mu Y.\ \tau\ (\zeta^{i}(\mathit{False}),Y)} \quad (*)$$

$$\zeta^{i+1}(\mathit{False}) = \boxed{\mu Y.\ \tau\ (\zeta^{i}(\mathit{False}),Y)} = \bigcup_j \tau^j(\zeta^{i}(\mathit{False}),\mathit{False})$$

by (*) and *Lemma*

Emerson and Lei's algorithm

- Example: $\mu X.\mu Y.\ \tau\ (X,Y)$ (τ is monotonic)
- $\zeta(X) = \mu Y.\ \tau\ (X,Y)$ $\mu X.\mu Y.\ \tau\ (X,Y) = \mu X.\ \zeta(X)$
- Note that $\zeta^{i-1}(\mathit{False}) \subseteq \zeta^{i}(\mathit{False})$ and τ $\bigcup$-continuous

$$\mu Y.\ \tau\ (\zeta^{i-1}(\mathit{False}),Y) \subseteq \boxed{\mu Y.\ \tau\ (\zeta^{i}(\mathit{False}),Y)} \quad (*)$$

$$\zeta^{i+1}(\mathit{False}) = \boxed{\mu Y.\ \tau\ (\zeta^{i}(\mathit{False}),Y)} = \cup_j\ \tau^{j}(\zeta^{i}(\mathit{False}),\mathit{False})$$

by (*) and <u>*Lemma*</u>

$$= \cup_j\ \tau^{j}(\zeta^{i}(\mathit{False}),\ \mu Y.\ \tau\ (\zeta^{i-1}(\mathit{False}),Y))$$

Emerson and Lei's algorithm

- Example: $\mu X.\mu Y.\ \tau\ (X,Y)$ (τ is monotonic)
- $\zeta(X) = \mu Y.\ \tau\ (X,Y)$ $\mu X.\mu Y.\ \tau\ (X,Y) = \mu X.\ \zeta(X)$
- Note that $\zeta^{i-1}(\mathit{False}) \subseteq \zeta^{i}(\mathit{False})$ and τ $\bigcup$-continuous

$$\mu Y.\ \tau\ (\zeta^{i-1}(\mathit{False}),Y) \subseteq \mu Y.\ \tau\ (\zeta^{i}(\mathit{False}),Y) \quad (*)$$

$$\zeta^{i+1}(\mathit{False}) = \mu Y.\ \tau\ (\zeta^{i}(\mathit{False}),Y) = \cup_j\ \tau^{j}(\zeta^{i}(\mathit{False}),\mathit{False})$$

by (*) and <u>*Lemma*</u>

$$= \cup_j\ \tau^{j}(\zeta^{i}(\mathit{False}),\ \mu Y.\ \tau\ (\zeta^{i-1}(\mathit{False}),Y))$$

Emerson and Lei's algorithm

- Example: $\mu X.\mu Y.\ \tau(X,Y)$ (τ is monotonic)
- $\zeta(X) = \mu Y.\ \tau(X,Y)$ $\mu X.\mu Y.\ \tau(X,Y) = \mu X.\ \zeta(X)$
- Note that $\zeta^{i-1}(\mathit{False}) \subseteq \zeta^{i}(\mathit{False})$ and τ $\bigcup$-continuous

$$\mu Y.\ \tau(\zeta^{i-1}(\mathit{False}),Y) \subseteq \mu Y.\ \tau(\zeta^{i}(\mathit{False}),Y) \quad (*)$$

$$\zeta^{i+1}(\mathit{False}) = \mu Y.\ \tau(\zeta^{i}(\mathit{False}),Y) = \bigcup_j \tau^j(\zeta^{i}(\mathit{False}),\mathit{False})$$

by (*) and *Lemma*

$$= \bigcup_j \tau^j(\zeta^{i}(\mathit{False}), \mu Y.\ \tau(\zeta^{i-1}(\mathit{False}),Y))$$

No need to use Y=*False*! Only **O(|S|)** iterations of τ.

Emerson and Lei's algorithm

function eval(f, e)

if $f = p$ **then return** $\{s \mid p \in L(s)\}$;
if $f = Q$ **then return** $e(Q)$;
if $f = g_1 \wedge g_2$ **then return** eval$(g_1, e) \cap$ eval(g_2, e);
if $f = g_1 \vee g_2$ **then return** eval$(g_1, e) \cup$ eval(g_2, e);

if $f = \langle a \rangle g$ **then return** $\{ s \mid \exists t\, [(s,t) \in a \text{ and } t \in \text{eval}(g, e)]\}$;
if $f = [a]g$ **then return** $\{ s \mid \forall t\, [(s,t) \in a \text{ implies } t \in \text{eval}(g, e)]\}$;

if $f = \mu Q_i . g(Q_i)$ **then**
  forall top-level greatest fixpoint subformulas $\nu Q_j . g'(Q_j)$ of g
    do $A[j] := True$;
  repeat
    $Q_{\text{old}} := A[i]$;
    $A[i] := \text{eval}(g, e\,[Q_i \leftarrow A[i]])$;
  until $A[i] = Q_{\text{old}}$;
  return $A[i]$;
end if;

end function

Emerson and Lei's algorithm

function eval(f, e)

Same as before →

if $f = p$ **then return** $\{s \mid p \in L(s)\}$;
if $f = Q$ **then return** $e(Q)$;
if $f = g_1 \wedge g_2$ **then return** $\text{eval}(g_1, e) \cap \text{eval}(g_2, e)$;
if $f = g_1 \vee g_2$ **then return** $\text{eval}(g_1, e) \cup \text{eval}(g_2, e)$;

if $f = \langle a \rangle g$ **then return** $\{\, s \mid \exists t\, [(s,t) \in a \text{ and } t \in \text{eval}(g, e)]\, \}$;
if $f = [a]g$ **then return** $\{\, s \mid \forall t\, [(s,t) \in a \text{ implies } t \in \text{eval}(g, e)]\, \}$;

if $f = \mu Q_i . g(Q_i)$ **then**
    forall top-level greatest fixpoint subformulas $\nu Q_j . g'(Q_j)$ of g
        do $A[j] := \mathit{True}$;
    repeat
        $Q_{\text{old}} := A[i]$;
        $A[i] := \text{eval}(g, e\,[Q_i \leftarrow A[i]])$;
    until $A[i] = Q_{\text{old}}$;
    return $A[i]$;
end if;

end function

Complexity

- Let d=AD(*f*)
- Since we need to start from *False* (*True*) only when μ and ν alternates, the complexity is **$O((|f|\cdot|S|)^d)$**

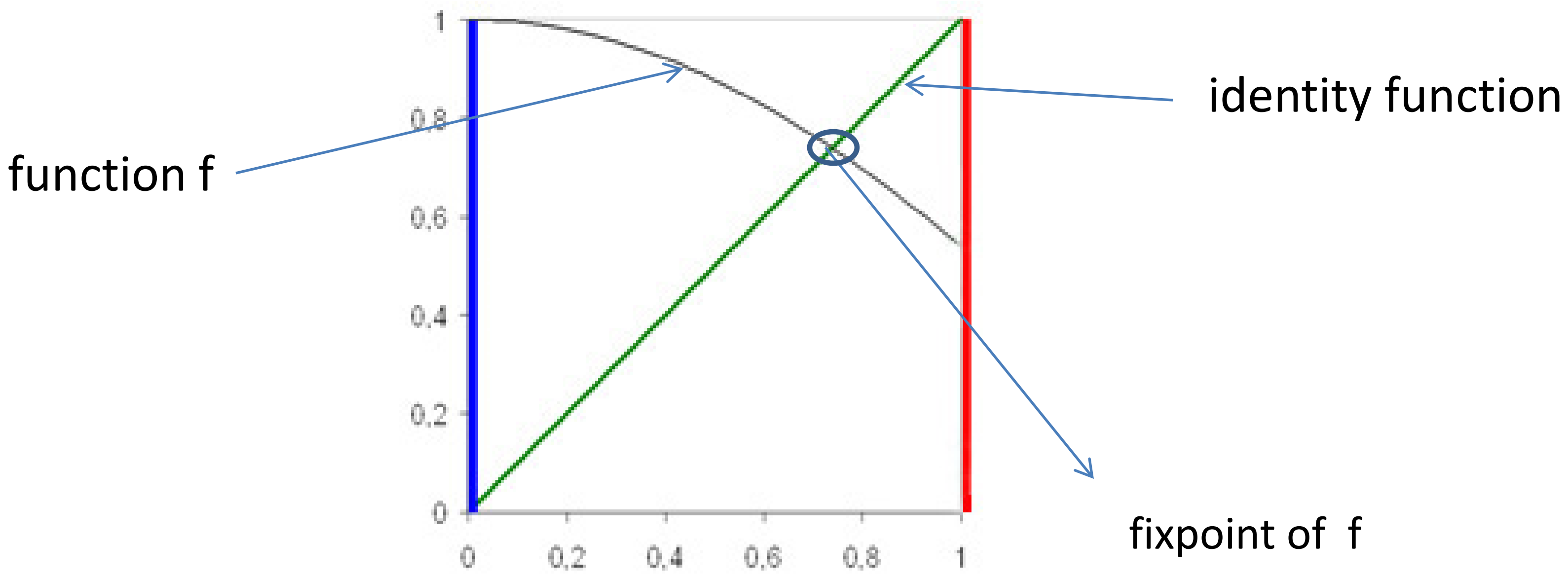
Complexity

- Let d=AD(*f*)
- Since we need to start from *False* (*True*) only when μ and ν alternates, the complexity is **$O((|f|\cdot|S|)^{d})$**
- Clarke *et al*. (CAV 1994) presented an algorithm with complexity **$O((|f|\cdot|S|)^{d/2+1})$**
- The Model Checking problem for the μ-calculus is in NP ⋂ co-NP

Other fixpoint theorems

Brouwer fixpoint theorem (one-dimensional case)

Every continuous f :[a,b] ⟶ [a,b] has a fixpoint



Other fixpoint theorems

Brouwer fixpoint theorem (one-dimensional case)

Every continuous $f : [a,b] \longrightarrow [a,b]$ has a fixpoint

Other fixpoint theorems

Brouwer fixpoint theorem (one-dimensional case)

Every continuous f :[a,b] $\longrightarrow$ [a,b] has a fixpoint

Proof:

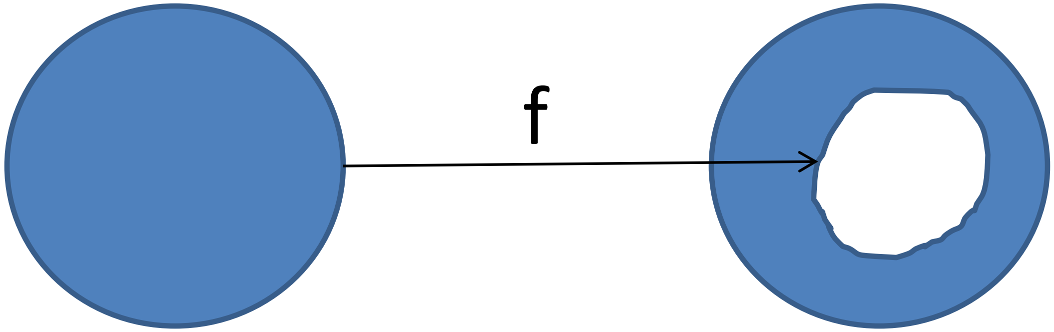
Define g(x)=f(x)-x. Then g(a)$\geqslant$0 and g(b)$\leqslant$0. By the intermediate value theorem, there is a point ξ in [a,b] such that $g(\xi) = 0 = f(\xi) - \xi$.

Thus ξ is a fixpoint for f.

Other fixpoint theorems

Brouwer fixpoint theorem (generalizations)

- Every continuous function from a *closed* disk to itself has a fixpoint



Other fixpoint theorems

Brouwer fixpoint theorem (generalizations)

- Every continuous function from a *closed* ball of an Euclidean space to itself has a fixpoint

Other fixpoint theorems

Brouwer fixpoint theorem (generalizations)

- Every continuous function from a *closed* ball of an Euclidean space to itself has a fixpoint

- Every continuous function from a *convex compact* subset K of an Euclidean space to K itself has a fixpoint

Other fixpoint theorems

Banach Contraction Principle

Say $f:\mathbb{R}^n \longrightarrow \mathbb{R}^n$ and $d(x,y) = ||x-y||$ for $x,y \in \mathbb{R}^n$. Suppose $\exists \alpha<1$ such that $d(f(x),f(y)) \leqslant \alpha \cdot d(x,y)$ for all $x,y \in \mathbb{R}^n$ (f is said to be a **contraction**). Then:

- f has a **unique fixpoint** u, and
- $\lim_{i\to\infty} f^i(y) = u$ for each $y \in \mathbb{R}^n$.