CS 267: Automated Verification

Lectures 4: μ-calculus

Instructor: Tevfik Bultan

μ-Calculus

μ-Calculus is a temporal logic which consist of the following:

- Atomic properties AP
- Boolean connectives: $\neg$, $\wedge$, $\vee$
- Precondition operator: EX
- Least and greatest fixpoint operators: μ y . $\mathcal{F}$ y and ν y. $\mathcal{F}$ y
  - $\mathcal{F}$ must be syntactically monotone in y
    - meaning that all occurrences of y in within $\mathcal{F}$ fall under an even number of negations

μ-Calculus

- μ-calculus is a powerful logic
  - Any CTL* property can be expressed in μ-calculus
- So, if you build a model checker for μ-calculus you would handle all the temporal logics we discussed: LTL, CTL, CTL*
- One can write a μ-calculus model checker using the basic ideas about fixpoint computations that we discussed
  - However, there is one complication
    - Nested fixpoints!

Mu-calculus Model Checking Algorithm

eval(f : mu-calculus formula) : a set of states

case: $f \in AP$	return $\{s \mid L(s,f)=true\}$;
case: $f \equiv \neg p$	return S - eval(p);
case: $f \equiv p \wedge q$	return eval(p) $\cap$ eval(q);
case: $f \equiv p \vee q$	return eval(p) $\cup$ eval(q);
case: $f \equiv EX\ p$	return EX(eval(p));

Mu-calculus Model Checking Algorithm

```
eval(f)
    …
    case: f ≡ μ y . g(y)
        y := False;
        repeat {
            y_old := y;
            y := eval(g(y));
        } until y = y_old
        return y;
```

Mu-calculus Model Checking Algorithm

```
eval(f)
    …
    case: f ≡ ν y . g(y)
        y := True;
        repeat {
                y_old := y;
                y := eval(g(y));
        } until y = y_old
        return y;
```

Nested Fixpoints

- Here is a CTL property

  $EG\ EF\ p = \nu y\ .\ (\mu z\ .\ p \vee EX\ z) \wedge EX\ y$

  – The fixpoints are not nested.

  – Inner fixpoint is computed only once and then the outer fixpoint is computed

  – Fixpoint characterizations of CTL properties do not have nested fixpoints

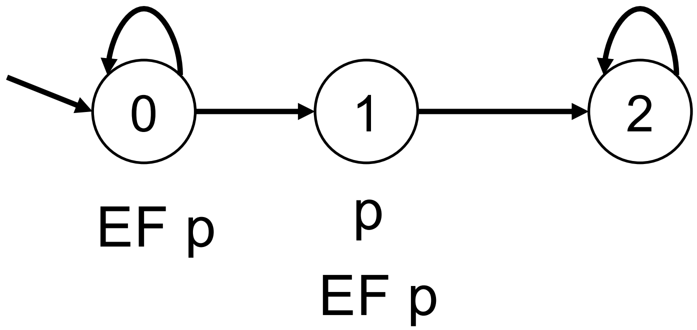
- Here is a CTL* property

  $EGF\ p = \nu y\ .\ \mu z\ .\ ((p \vee EX\ z) \wedge EX\ y)$

  – The fixpoints are nested.

  – Inner fixpoint is recomputed for each iteration of the outer fixpoint

Nested Fixpoint Example



$0 \models EG\ EF\ p$

$$EG\ EF\ p = \nu y\ .\ \overbrace{(\mu z\ .\ \underbrace{p \vee EX\ z}_{\mathcal{F}_1}) \wedge EX\ y}^{\mathcal{F}_2}$$

EF p fixpoint

$\varnothing$
$\mathcal{F}_1(\varnothing) = \{1\}$
$\mathcal{F}_1^2(\varnothing) = \{0,1\}$
$\mathcal{F}_1^3(\varnothing) = \{0,1\}$

EG {0,1} fixpoint

$S=\{0,1,2\}$
$\mathcal{F}_2(S) = \{0,1\}$
$\mathcal{F}_2^2(S) = \{0\}$
$\mathcal{F}_2^3(S) = \{0\}$

$EG\ EF\ p = \{0\}$

$0 \not\models EGF\ p$

$$EGF\ p = \nu y\ .\ \mu z\ .\ \underbrace{((p \vee EX\ z) \wedge EX\ y)}_{\mathcal{F}_3}$$

nested fixpoint

$\mathcal{F}_3$	y	z
0,0	{0,1,2}	$\varnothing$
0,1		{1}
0,2		{0,1}
0,3		{0,1}
1,0	{0,1}	$\varnothing$
1,1		$\varnothing$
2,0	$\varnothing$	$\varnothing$
2,1		$\varnothing$
3,0	$\varnothing$	

$EGF\ p = \varnothing$