Lecture 14: Basic Fixpoint Theorems (cont.)

- Predicate Transformers
- Monotonicity and Continuity
- Existence of Fixpoints
- Computing Fixpoints
- Fixpoint Characterization of CTL Operators

Predicate Transformers

Let $M = (S, R, L)$ be an arbitrary finite Kripke structure.

$Pred(S)$ is the lattice of predicates over S. Each predicate is identified with the set of states that make it true. The ordering is set inclusion.

Thus, the least element in the lattice is the empty set, denoted by $False$, and the greatest element in the lattice is the set of all states, denoted by $True$.

A functional $F : Pred(S) \longrightarrow Pred(S)$ is called a *predicate transformer*.

- E. M. Clarke and E. A. Emerson. Synthesis of synchronization skeletons for branching time temporal logic. In *Logic of Programs: Workshop, Yorktown Heights, NY, May 1981*, volume 131 of *Lecture Notes in Computer Science*. Springer-Verlag, 1981.

Monotonicity and Continuity

Let $\tau : Pred(S) \longrightarrow Pred(S)$ be a predicate transformer, then

1. τ is *monotonic* provided that $P \subseteq Q$ implies $\tau[P] \subseteq \tau[Q]$;
2. τ is $\cup$-*continuous* provided that $P_1 \subseteq P_2 \subseteq \ldots$ implies $\tau[\cup_i P_i] = \cup_i \tau[P_i]$;
3. τ is $\cap$-*continuous* provided that $P_1 \supseteq P_2 \supseteq \ldots$ implies $\tau[\cap_i P_i] = \cap_i \tau[P_i]$.

Basic Fixpoint Theorems

If τ is monotonic, then it has a least fixpoint, **lfp** $Z\left[\tau(Z)\right]$, and a greatest fixpoint, **gfp** $Z\left[\tau(Z)\right]$.

lfp $Z\left[\tau(Z)\right] = \cap\{Z \mid \tau(Z) = Z\}$ whenever τ is monotonic.

lfp $Z\left[\tau(Z)\right] = \cup_i \tau^i(False)$ whenever τ is also $\cup$-continuous;

gfp $Z\left[\tau(Z)\right] = \cup\{Z \mid \tau(Z) = Z\}$ whenever τ is monotonic.

gfp $Z\left[\tau(Z)\right] = \cap_i \tau^i(True)$ whenever τ is also $\cap$-continuous.

Some Useful Lemmas

Let M be a finite Kripke structure and let τ be a monotonic predicate transformer on S.

1. The functional τ is both $\cup$-continuous and $\cap$-continuous.
2. For every i, $\tau^i(False) \subseteq \tau^{i+1}(False)$ and $\tau^i(True) \supseteq \tau^{i+1}(True)$.
3. There is an integer i_0 such that for every $j \geq i_0$, $\tau^j(False) = \tau^{i_0}(False)$.
   There is an integer j_0 such that for every $j \geq j_0$, $\tau^j(True) = \tau^{j_0}(True)$.
4. There is an integer i_0 such that **lfp** $Z\left[\tau(Z)\right]$ is $\tau^{i_0}(False)$.
   There is an integer j_0 such that **gfp** $Z\left[\tau(Z)\right]$ is $\tau^{j_0}(True)$.

Least Fixpoint Algorithm

As a consequence of the preceding lemmas, if τ is monotonic, its least fixpoint can be computed by the following program.

```
functionLfp(Tau: PredicateTransformer)
begin
        Q := False;
        Q' := Tau(Q);
        while (Q ≠ Q') do
        begin
            Q := Q';
            Q' := Tau(Q')
        end;
        return(Q)
end
```

Correctness of Algorithm

The invariant for the while loop is given by the assertion

$$(Q' = \tau[Q]) \wedge (Q' \subseteq \textbf{lfp}\, Z \left[\tau(Z)\right])$$

It is easy to see that at the beginning of the i-th iteration, $Q = \tau^{i-1}(False)$ and $Q' = \tau^{i}(False)$. Lemma 2 implies that

$$False \subseteq \tau(False) \subseteq \tau^2(False) \subseteq \ldots .$$

So, the number of iterations before the loop terminates is bounded by the cardinality of S.

When the loop terminates, we have $Q = \tau[Q]$ and $Q \subseteq \textbf{lfp}\, Z \left[\tau(Z)\right]$.

It follows directly that $Q = \textbf{lfp}\, Z \left[\tau(Z)\right]$ and that the value returned is the least fixpoint.

Greatest Fixpoint Algorithm

The greatest fixpoint of τ may be computed in a similar manner. Essentially the same argument can be used to show that the procedure terminates and that the value it returns is **gfp** $Z\left[\tau(Z)\right]$.

```
functionGfp(Tau: PredicateTransformer)
begin
        Q := True;
        Q' := Tau(Q);
        while (Q ≠ Q') do
        begin
            Q := Q';
            Q' := Tau(Q')
        end;
        return (Q)
end
```

Fixpoint Characterizations for CTL

Each CTL operator can be characterized as a least or greatest fixpoint of a predicate transformer:

- $\mathbf{A}[f_1 \mathbin{\mathbf{U}} f_2] = \mathbf{lfp}\, Z \left[f_2 \vee (f_1 \wedge \mathbf{AX}\, Z)\right]$
- $\mathbf{E}[f_1 \mathbin{\mathbf{U}} f_2] = \mathbf{lfp}\, Z \left[f_2 \vee (f_1 \wedge \mathbf{EX}\, Z)\right]$
- $\mathbf{AF}\, f_1 = \mathbf{lfp}\, Z \left[f_1 \vee \mathbf{AX}\, Z\right]$
- $\mathbf{EF}\, f_1 = \mathbf{lfp}\, Z \left[f_1 \vee \mathbf{EX}\, Z\right]$
- $\mathbf{AG}\, f_1 = \mathbf{gfp}\, Z \left[f_1 \wedge \mathbf{AX}\, Z\right]$
- $\mathbf{EG}\, f_1 = \mathbf{gfp}\, Z \left[f_1 \wedge \mathbf{EX}\, Z\right]$

We will only prove the characterization for **EU**.

Fixpoint Characterization of EU

Lemma

$\mathbf{E}[f_1 \;\mathbf{U}\; f_2]$ is the least fixpoint of the functional $\tau(Z) = f_2 \vee (f_1 \wedge \mathbf{EX}\, Z)$.

Proof:

It is straightforward to prove that $\mathbf{E}[f_1 \;\mathbf{U}\; f_2]$ is a fixpoint of $\tau(Z)$.

Additional steps are required to show that $\mathbf{E}[f_1 \;\mathbf{U}\; f_2]$ is the least such fixpoint.

1. Prove that $\tau(Z) = f_2 \vee (f_1 \wedge \mathbf{EX}\, Z)$ is monotonic.
2. Observe that τ is $\cup$-continuous and that $\mathbf{lfp}\, Z \left[\tau(Z)\right] = \cup_i \tau^i(False)$.
3. Show that $\mathbf{E}[f_1 \;\mathbf{U}\; f_2] = \cup_i \tau^i(False)$. See next page.
4. Conclude from steps 2 and 3 that $\mathbf{E}[f_1 \;\mathbf{U}\; f_2]$ is the least fixpoint of $\tau(Z) = f_2 \vee (f_1 \wedge \mathbf{EX}\, Z)$. $\square$

Characterization of EU (Cont.)

Next, we show that $\mathbf{E}[f_1 \mathbf{U} f_2] = \cup_i \tau^i(False)$. We break this step into two parts:

- First, show that $\cup_i \tau^i(False) \subseteq \mathbf{E}[f_1 \mathbf{U} f_2]$.

  Hint: Prove by induction that for all i, $\tau^i(False) \subseteq \mathbf{E}[f_1 \mathbf{U} f_2]$. Use the fact that $\mathbf{E}[f_1 \mathbf{U} f_2]$ is a fixpoint of $\tau(Z)$.

- Next, show that $\mathbf{E}[f_1 \mathbf{U} f_2] \subseteq \cup_i \tau^i(False)$.

  Hint: If $s_1 \models \mathbf{E}[f_1 \mathbf{U} f_2]$, then there is a path $\pi = s_1, \ldots, s_j, \ldots$ such that $s_j \models f_2$ and for all $l < j$, $s_l \models f_1$. Show that $s_1 \in \tau^j(False)$.

Simple Example for $\mathbf{E}[p\,\mathbf{U}\,q]$

The next four figures show how $\mathbf{E}[p\,\mathbf{U}\,q]$ may be computed for a simple Kripke structure.

In this case the functional τ is given by

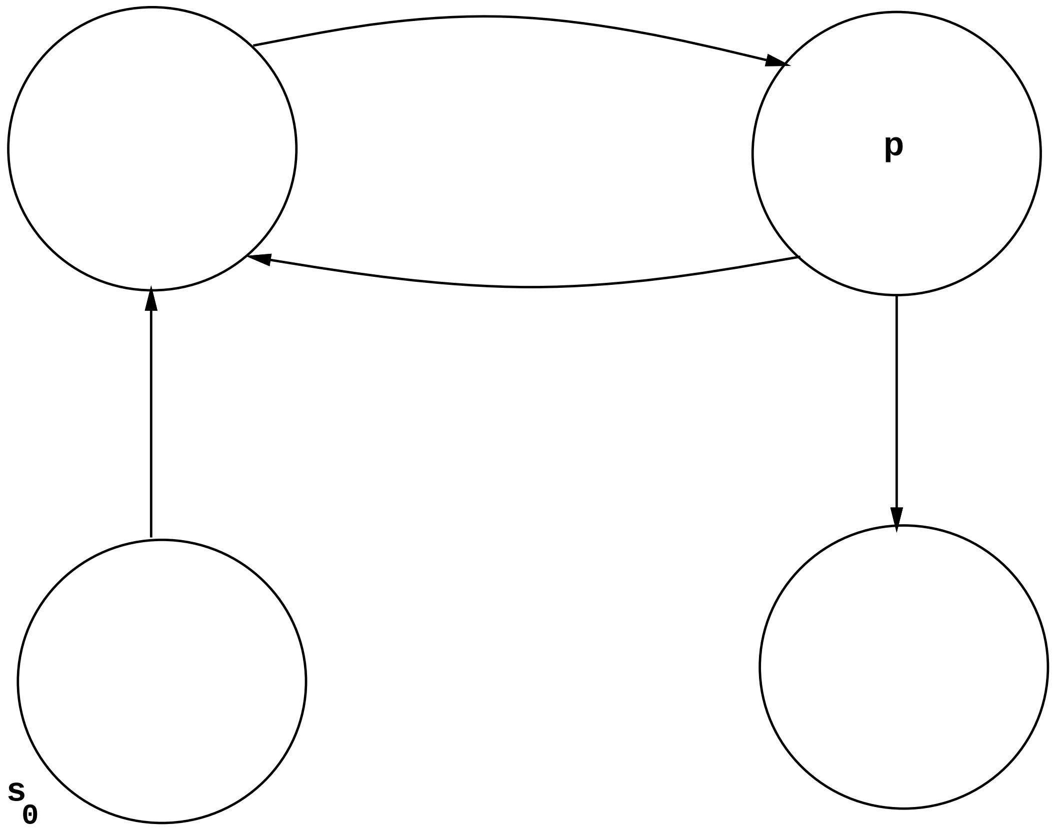
$$\tau(Z) = q \vee (p \wedge \mathbf{EX}\,Z).$$

The figures demonstrate that the sequence of approximations $\tau^i(False)$ converges to $\mathbf{E}[p\,\mathbf{U}\,q]$.

$\mathbf{E}[p\,\mathbf{U}\,q] = \tau^3(False)$ since $\tau^3(False) = \tau^4(False)$.

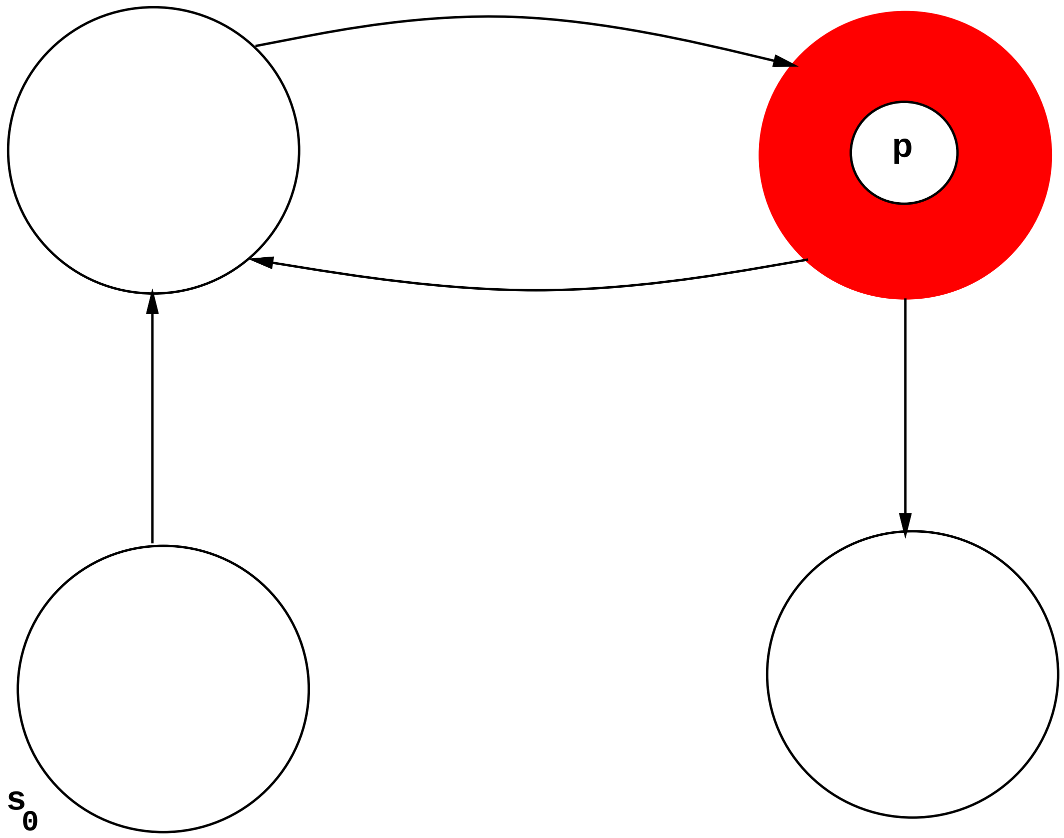
Simple Example for $\mathbf{E}[p\,\mathbf{U}\,q]$ (Cont.)

$M, s_0 \models \mathbf{E}[p\,\mathbf{U}\,q]$?



Simple Example for $\mathbf{E}[p\ \mathbf{U}\ q]$ (Cont.)

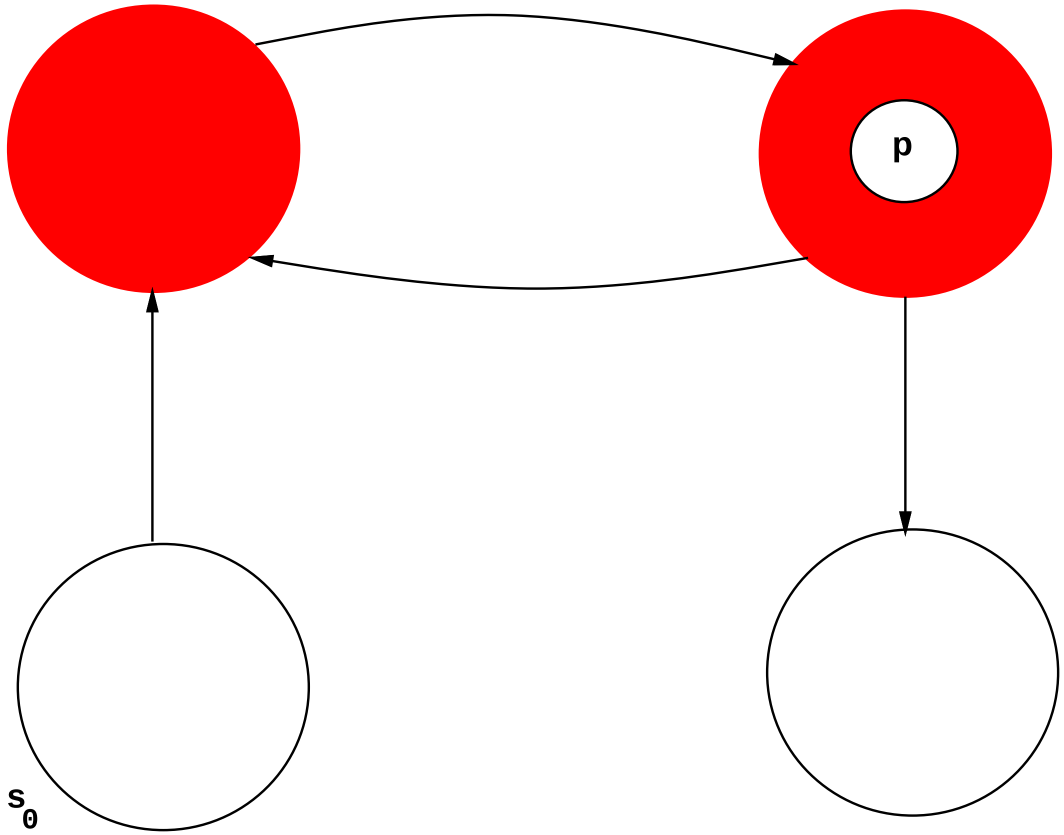
$M, s_0 \models \mathbf{E}[p\ \mathbf{U}\ q]?$



$\tau^1(False)$

Simple Example for $\mathbf{E}[p \ \mathbf{U} \ q]$ (Cont.)

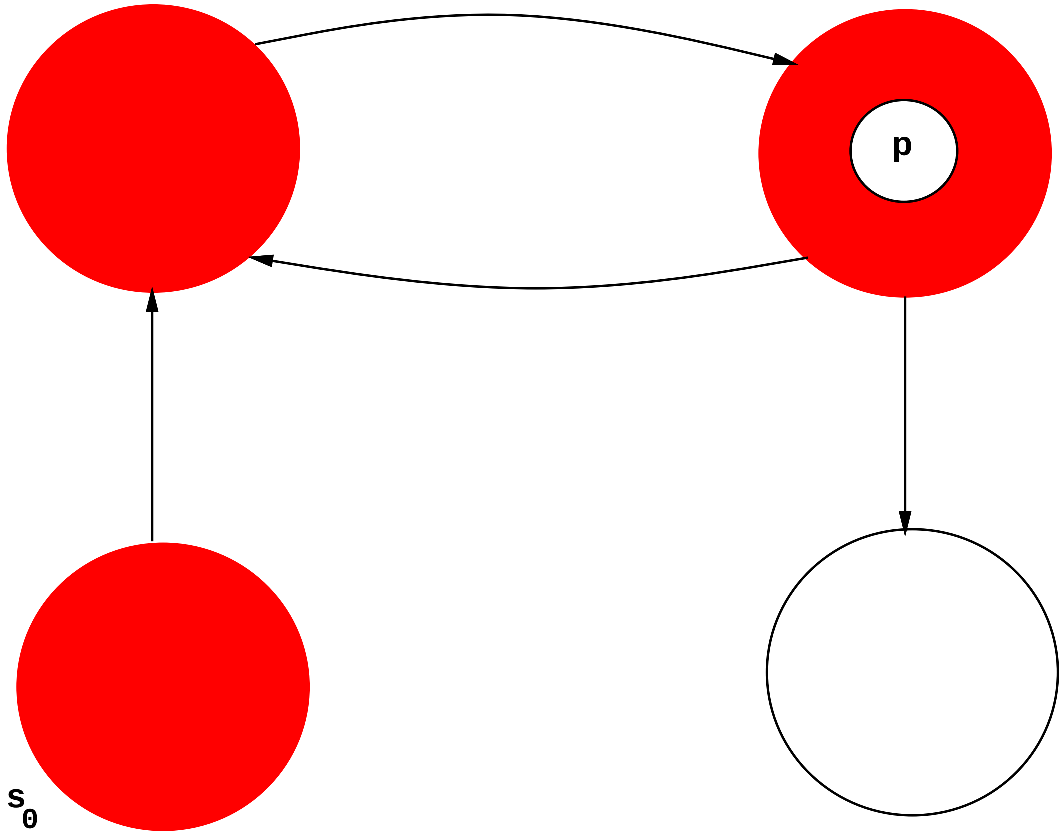
$M, s_0 \models \mathbf{E}[p \ \mathbf{U} \ q]?$



$\tau^2(False)$

Simple Example for $\mathbf{E}[p\ \mathbf{U}\ q]$ (Cont.)

$M, s_0 \models \mathbf{E}[p\ \mathbf{U}\ q]$?



$\tau^3(False)$