Lógica Temporal

Manuel Alcino Cunha

Departamento de Informática
Universidade do Minho

2005/06

Motivação

- Como já vimos, existem vários tipos de propriedades desejáveis num sistema concorrente:

  Segurança *"Nada de mau poderá acontecer!"*
  Animação *"Eventualmente algo de bom irá acontecer!"*

- Em relação às propriedades de segurança já sabemos verificar uma subclasse muito simples de propriedades, designadas invariantes: *"o sistema nunca atinge um estado de erro"*.
- Mas como expressar e verificar propriedades se segurança como *"o sistema não pode reiniciar sem que antes o botão de reset seja pressionado"*?

Motivação

- Também já sabemos verificar algumas propriedades de animação, como por exemplo a invertibilidade.
- Mas como expressar e verificar propriedades de animação como *"sempre que o botão de reset é pressionado o sistema será reiniciado"*?
- Embora o tempo não seja mencionado explicitamente, estas propriedades restringem as execuções válidas do sistema estabelecendo ordens e relações causais entre ocorrências de estados.
- A complexidade destas relações exige um formalismo de especificação adequado: a *lógica temporal*.
- Nesta lógica temos operadores modais para exprimir propriedades como: *"uma proposição é sempre válida"*, *"uma proposição é eventualmente válida"*, ...

Motivação

- Na lógica temporal o tempo não é mencionado explicitamente, mas visto como possíveis sequências de estados.
- Para verificar a validade de uma fórmula é necessário saber quais as sequências de estados (*caminhos* ou *traços*) possíveis num determinado sistema.
- A lógica temporal é orientada aos estados: a sequência de acções que originou um dado caminho é irrelevante.
- Também temos que saber qual é o objecto de discurso: quais as proposições atómicas da lógica e em que estados são válidas.
- Toda esta informação será representada através de uma espécie de grafo de acessibilidade denominado *estrutura de Kripke*.

Estruturas de Kripke

Definição

Seja P um conjunto de proposições atómicas. Uma *estrutura de Kripke* é um tuplo

$$(S, i, R, L)$$

onde

- S é um conjunto finito de estados;
- $i \in S$ é o estado inicial;
- $R \subseteq S \times S$ é uma relação de transição total, ou seja, que verifica
$$\forall s \in S \cdot \exists s' \in S \cdot (s, s') \in R$$
- $L : S \to \mathcal{P}(P)$ é uma função que etiqueta cada estado com o conjunto de fórmulas atómicas válidas nesse estado.

Estruturas de Kripke e Redes de Petri

- A semântica da lógica temporal será definida sobre estruturas de Kripke.
- Desta forma, as técnicas de especificação e verificação de propriedades podem ser apresentadas independentemente do formalismo de modelação adoptado.
- No caso das redes de Petri elementares, a conversão do grafo de acessibilidade para uma estrutura de Kripke é trivial:
  - O conjunto de proposições atómicas é idêntico ao conjunto de lugares.
  - A função de etiquetagem é a função indentidade.
  - É necessário acrescentar lacetes em todos os *deadlocks* para tornar a relação de transição total.

Tempo Linear vs Tempo Ramificado

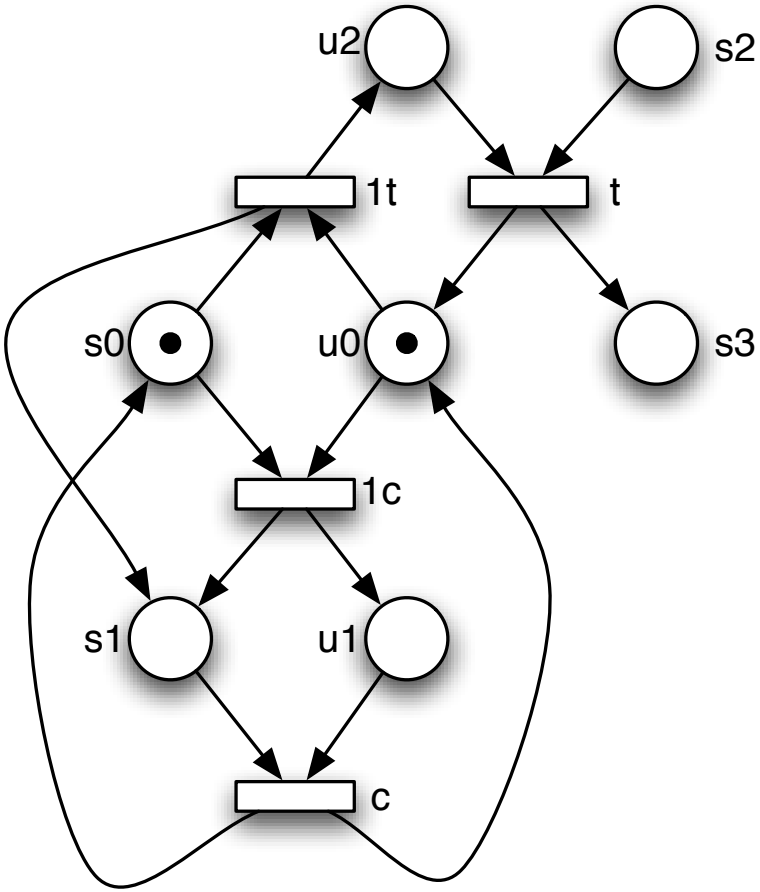
- As lógicas temporais divergem quanto ao modo como representam o tempo. Existem dois modelos básicos:

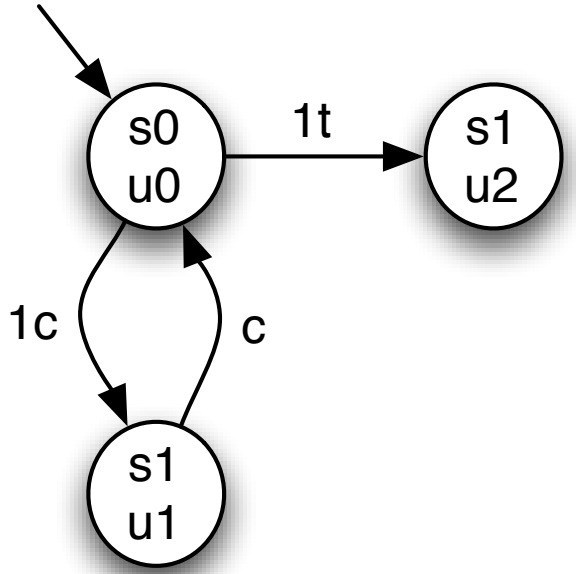
  Tempo Linear O comportamento do sistema consiste no conjunto de traços infinitos que começam no estado inicial i.

  Tempo Ramificado Todo o comportamento do sistema é capturado por uma árvore de computação de profundidade ilimitada cuja raiz é o estado inicial i.

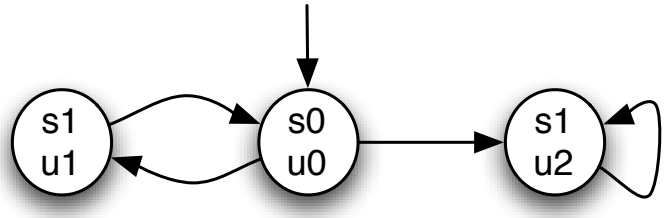
- Ambos os modelos podem ser calculados a partir da estrutura de Kripke, mas o segundo tem mais informação.
- De facto, existem propriedades que apenas podem ser verificadas usando tempo ramificado.

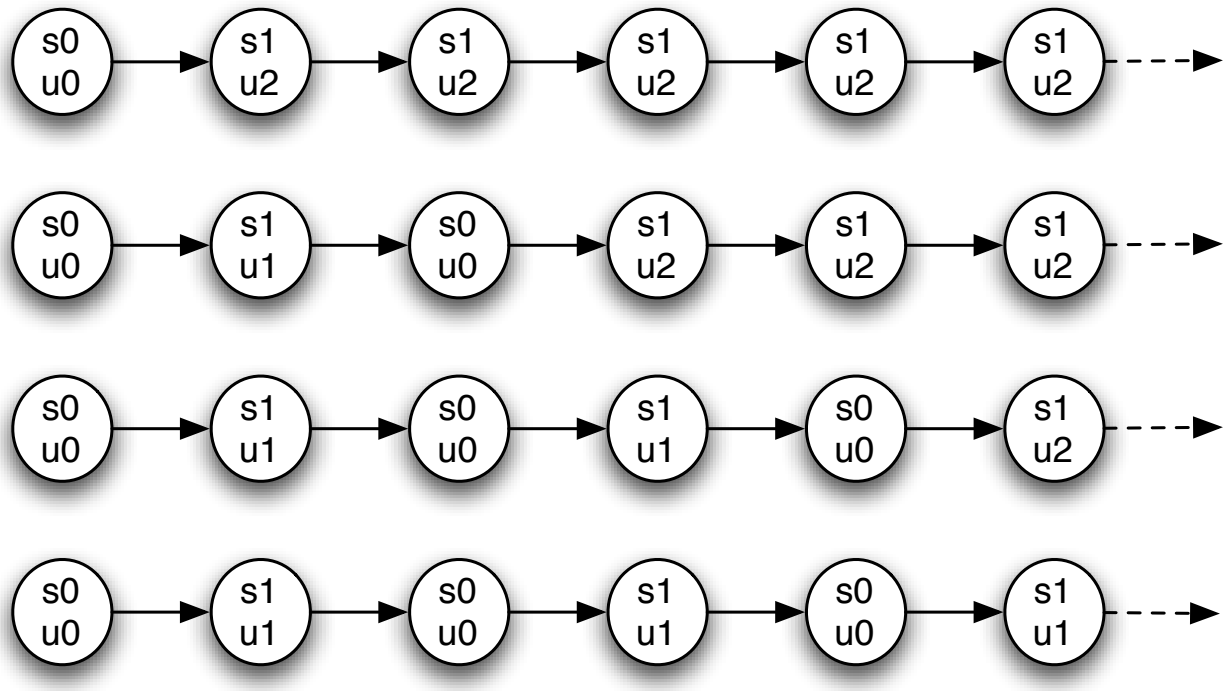
Exemplo



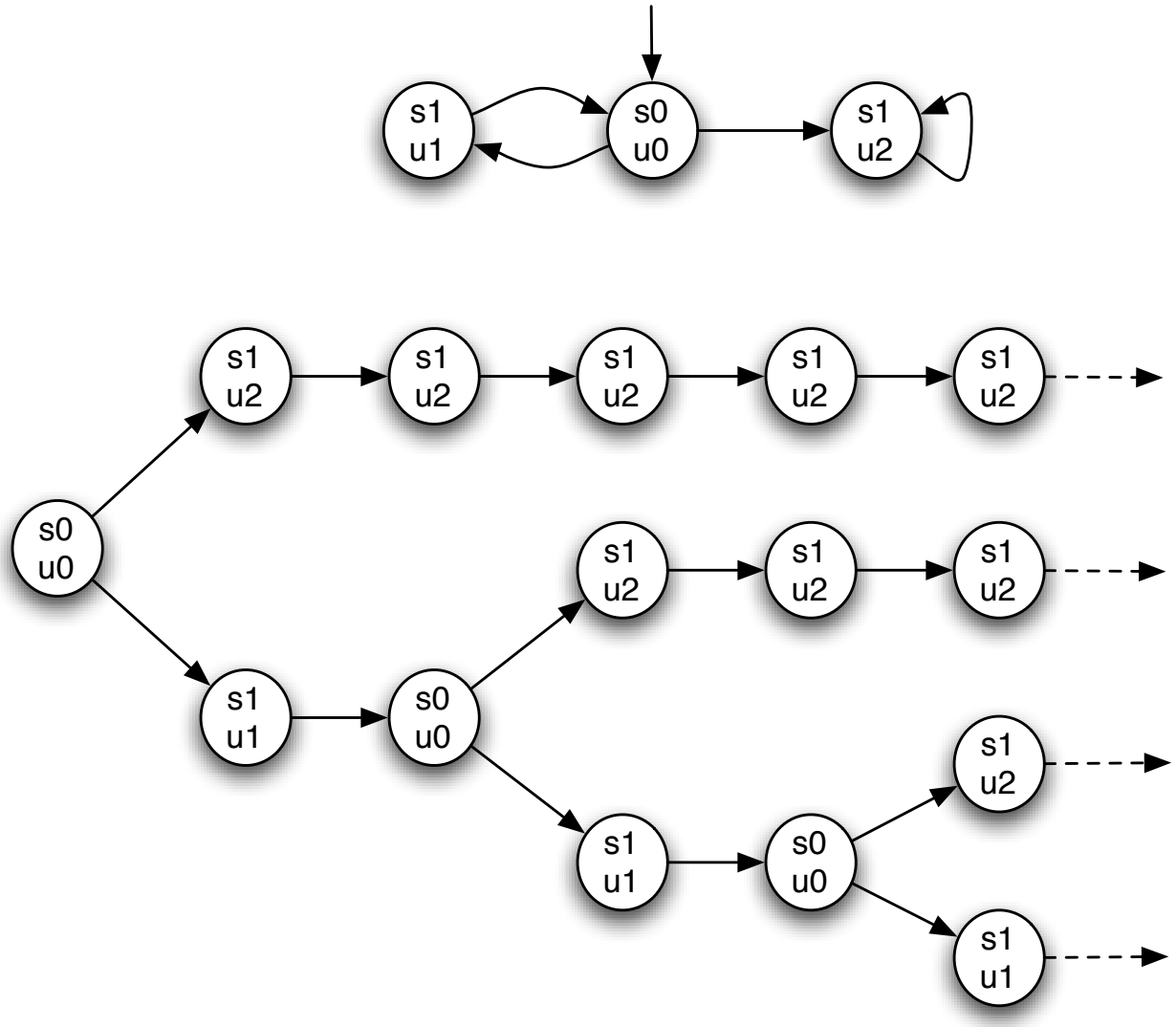


Exemplo Tempo Linear





Exemplo Tempo Ramificado



Linear Temporal Logic

- Na *linear temporal logic* (LTL) as fórmulas são interpretadas sobre traços infinitos.
- Para além das conectivas usuais da lógica proposicional temos os seguintes operadores temporais:

Next	$X\ f$ quando f é válida no *próximo* estado.
Future	$F\ f$ quando f é *eventualmente* válida.
Globally	$G\ f$ quando f é *sempre* válida.
Until	$f\ U\ g$ quando f é válida *até* que g o seja.
Release	$f\ R\ g$ quando a ocorrência de um estado onde f é válida *liberta* g de o ser.

Sintaxe

- Dado um conjunto P de proposições atómicas, o conjunto das fórmulas LTL sintacticamente válidas é definido indutivamente da seguinte forma:
  - true e false são fórmulas LTL.
  - Qualquer $p \in P$ é uma fórmula LTL.
  - Se f é uma fórmula LTL então $\neg f$, $X\ f$, $F\ f$ e $G\ f$ são fórmulas LTL.
  - Se f e g são fórmulas LTL então $f \vee g$, $f \wedge g$, $f\ U\ g$ e $f\ R\ g$ são fórmulas LTL.
- As conectivas X, F e G são por vezes definidas usando, respectivamente, os símbolos $\bigcirc$, $\Diamond$ e $\Box$.

Caminhos

- Dada uma estrutura de Kripke $M = (S, i, R, L)$ um *caminho* de M iniciado em $s_0 \in S$ é uma sequência infinita de estados $\pi = s_0, s_1, s_2, \ldots$ onde $\forall i \geq 0 \cdot (s_i, s_{i+1}) \in R$.
- Dado um caminho π, o seu primeiro estado será denotado por π_0.
- O sufixo do caminho $\pi = s_0, s_1, s_2, \ldots$ que começa na posição i será denotado por $\pi^i = s_i, s_{i+1}, s_{i+2}, \ldots$.

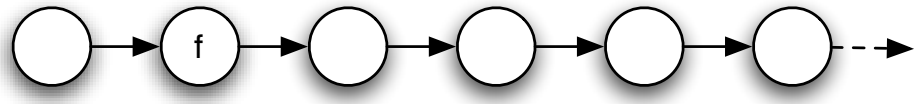
Semântica

- O facto de uma fórmula LTL f se verificar num caminho π da estrutura de Kripke M será denotado por $M, \pi \models f$. Diz-se que π é um *modelo* de f.
- A relação $\models$ é definida indutivamente da seguinte forma, onde $p \in P$ representa uma fórmula atómica e f e g são fórmulas LTL arbitrárias.

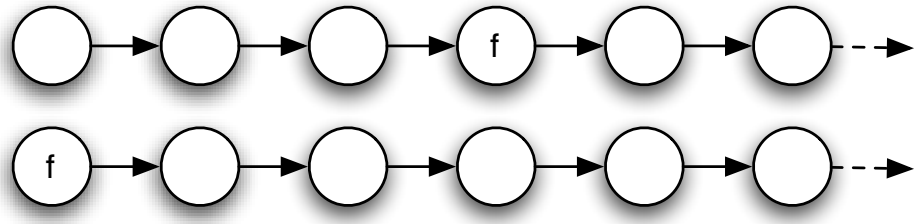
$$
\begin{array}{lcl}
M, \pi \models \text{true} & & \\
M, \pi \not\models \text{false} & & \\
M, \pi \models p & \Leftrightarrow & p \in L(\pi_0) \\
M, \pi \models \neg f & \Leftrightarrow & M, \pi \not\models f \\
M, \pi \models f \vee g & \Leftrightarrow & M, \pi \models f \quad \text{ou} \quad M, \pi \models g \\
M, \pi \models f \wedge g & \Leftrightarrow & M, \pi \models f \quad \text{e} \quad M, \pi \models g
\end{array}
$$

Semântica

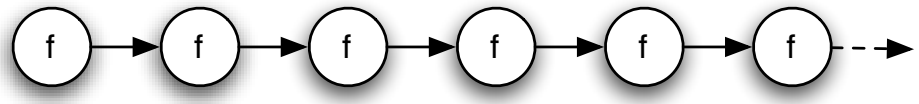
$$M, \pi \models X\ f \quad \Leftrightarrow \quad M, \pi^1 \models f$$



$$M, \pi \models F\ f \quad \Leftrightarrow \quad \exists i \geq 0 \cdot M, \pi^i \models f$$

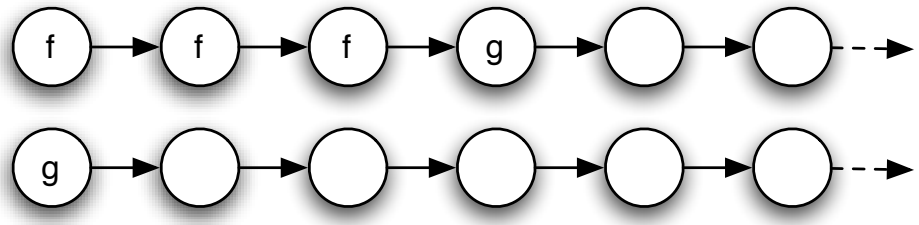


$$M, \pi \models G\ f \quad \Leftrightarrow \quad \forall i \geq 0 \cdot M, \pi^i \models f$$



Semântica

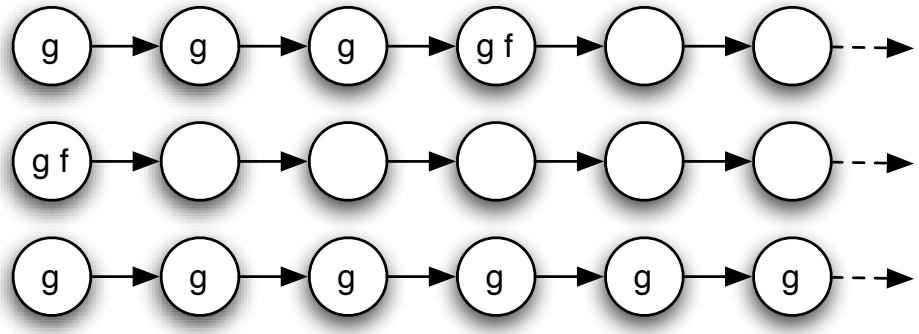
$$M, \pi \models f \ U \ g \quad \Leftrightarrow \quad \exists i \geq 0 \cdot M, \pi^i \models g \text{ e } \forall 0 \leq j < i \cdot M, \pi^j \models f$$



$$M, \pi \models f \ R \ g$$

$$\Leftrightarrow$$

$$\forall i \geq 0 \cdot \text{ se } \forall 0 \leq j < i \cdot M, \pi^j \not\models f \text{ então } M, \pi^i \models g$$



Validade

- Uma fórmula LTL f é válida num estado s de uma estrutura de Kripke M (denotado $M, s \models f$) sse para todos os caminhos π de M onde $\pi_0 = s$ se verifica $M, \pi \models f$.
- Uma fórmula LTL f é válida numa estrutura de Kripke M (denotado $M \models f$) sse $M, i \models f$.

Conectivas Mínimas

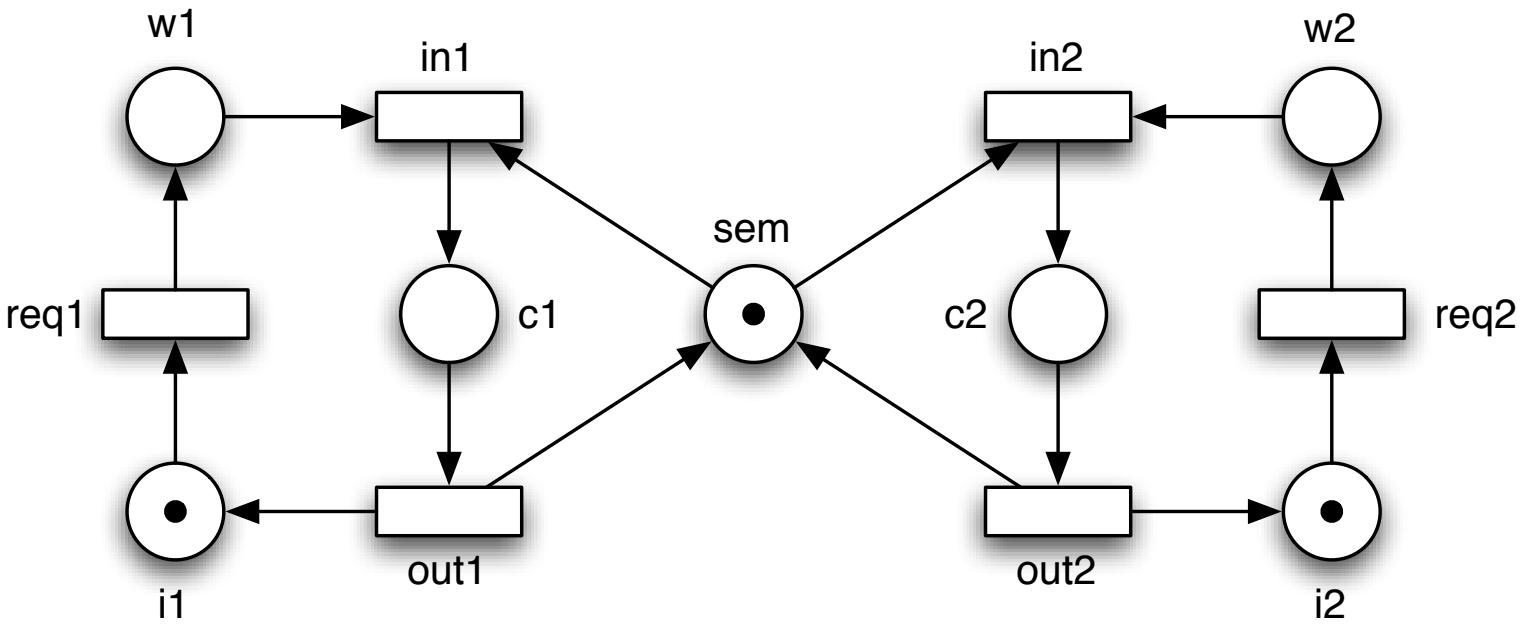
- É relativamente fácil de demonstrar que as conectivas $\neg$, $\vee$, X e U são suficientes para expressar qualquer fórmula LTL.

$$\begin{aligned} f \wedge g &\equiv \neg(\neg f \vee \neg g) \\ F\ f &\equiv \text{true}\ U\ f \\ G\ f &\equiv \neg F\ \neg f \\ f\ R\ g &\equiv \neg(\neg f\ U\ \neg g) \end{aligned}$$

- As seguintes equivalências também são úteis.

$$\begin{aligned} f \supset g &\equiv \neg f \vee g \\ G\ f &\equiv \text{false}\ R\ f \\ f\ R\ g &\equiv g\ U\ (f \wedge g) \vee G\ g \end{aligned}$$

Exemplo



Exemplo

- Exclusão mútua: os processos nunca estão simultaneamente na região crítica.

$$G\ \neg(c_1 \wedge c_2)$$

- Evolução: sempre que um processo está espera eventualmente estará na região crítica.

$$G\ (w_1 \supset F\ c_1) \wedge \ldots$$

- Prioridade: o primeiro processo a requisitar o acesso é o primeiro a aceder à região crítica.

$$G\ (w_1 \wedge i_2 \supset (\neg c_2\ U\ c_1)) \wedge \ldots$$

Exemplo

- Alternância: os processos alternam no acesso à região crítica.

$$G\ (c_1 \supset c_1\ U\ (c_2\ R\ \neg c_1)) \wedge \ldots$$

- Se o primeiro processo está um número ilimitado de vezes na sua região crítica então também está um número ilimitado de vezes à espera.

$$GF\ c_1 \supset GF\ w_1$$

- Há um momento a partir do qual o primeiro processo fica inactivo.

$$FG\ i_1$$

Padrões

- Usados para exprimir requisitos de existência ou ausência de determinadas condições (padrões) em períodos bem definidas da execução.
- Os períodos especificam o contexto da ocorrência do padrão e correspondem a segmentos dos caminhos delimitados por certos eventos.
- Os padrões mais frequentes são:

  Ausência Quando no contexto se pretende que não ocorram certos eventos ou estados.

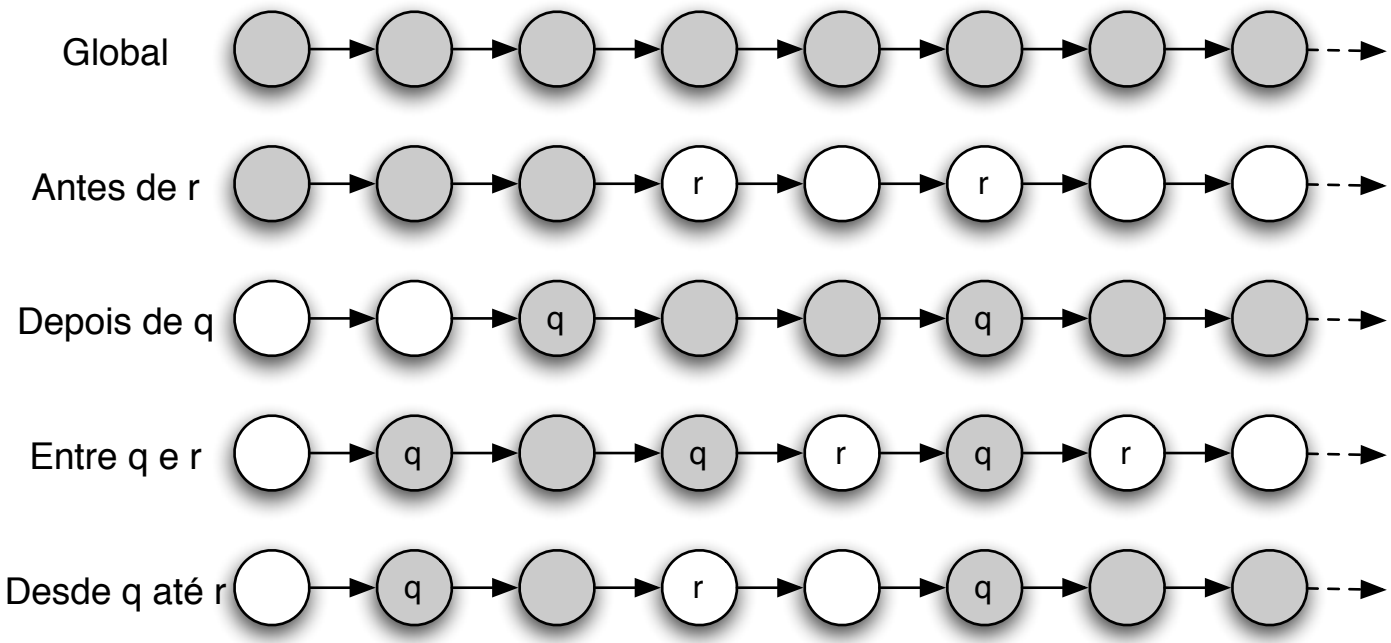
  Universalidade Quando se pretende que em todo o contexto certa propriedade se verifique.

  Existência Quando se pretende que uma propriedade ocorra alguma vez no contexto.

  Resposta Dentro do contexto a ocorrência de certo evento (causa) deve ser seguida da ocorrência de outro (efeito).

Contextos

- Os contextos são especificados definindo uma condição de arranque e outra de paragem, possivelmente opcionais.
- Se ambas estiverem definidas, o contexto engloba todos os estados começando no primeiro que verifique a condição de arranque até (mas não incluindo) um que verifique a condição de paragem.



Universalidade

- A propriedade p deve verificar-se em todo o contexto.
- Muito semelhante ao padrão de ausência.

Global	$G\ p$
Antes de r	$(F\ r) \supset (p\ U\ r)$
Depois de q	$G\ (q \supset G\ p)$
Entre q e r	$G\ ((q \wedge \neg r \wedge F\ r) \supset (p\ U\ r))$
Desde q até r	$G\ ((q \wedge \neg r) \supset ((G\ p) \vee (p\ U\ r)))$

Existência

- Este padrão determina que p deve ser verdade algures no contexto.

Global	$F\ p$
Antes de r	$p\ R\ \neg r$
Depois de q	$(G\ \neg q) \vee F\ (q \wedge F\ p)$
Entre q e r	$G\ ((q \wedge \neg r) \supset (p\ R\ \neg r))$
Desde q até r	$G\ ((q \wedge \neg r) \supset (\neg r\ U\ (\neg r \wedge p)))$

- Ver mais padrões e respectivas implementações em

`http://patterns.projects.cis.ksu.edu`

Axiomas

- Leis de distributividade:

$$X\ (f \vee g) \equiv X\ f \vee X\ g$$
$$X\ (f \wedge g) \equiv X\ f \wedge X\ g$$
$$X\ \neg f \equiv \neg X\ f$$
$$F\ (f \vee g) \equiv F\ f \vee F\ g$$
$$\neg F\ f \equiv G\ \neg f$$
$$G\ (f \wedge g) \equiv G\ f \wedge G\ g$$
$$\neg G\ f \equiv F\ \neg f$$
$$(f \wedge g)\ U\ h \equiv (f\ U\ h) \wedge (g\ U\ h)$$
$$f\ U\ (g \vee h) \equiv (f\ U\ g) \vee (f\ U\ h)$$

Axiomas

- Leis de idempotência:

$$FF\ f \equiv F\ f$$
$$GG\ f \equiv G\ f$$
$$FGF\ f \equiv GF\ f$$
$$GFG\ f \equiv FG\ f$$
$$f\ U\ (f\ U\ g) \equiv f\ U\ g$$

- Leis de expansão:

$$F\ f \equiv f \vee XF\ f$$
$$G\ f \equiv f \wedge XG\ f$$
$$f\ U\ g \equiv g \vee (f \wedge X\ (f\ U\ g))$$
$$f\ R\ g \equiv g \wedge (f \vee X\ (f\ R\ g))$$

Full Computation Tree Logic

- Muitas propriedades interessantes de sistemas concorrentes não podem ser especificadas usando o modelo de tempo linear.
- Por exemplo, como especificar a invertibilidade?
- A *full computation tree logic* (CTL*) acrescenta à LTL os quantificadores de caminho A (universal) e E (existencial).
- Sendo f uma fórmula de caminho, $A\ f$ será válida num estado s se f for válida em todos os caminhos que começam em s.
- $E\ f$ é válida num estado s se f for válida em pelo menos um caminho que comece em s.
- Qualquer fórmula LTL f é equivalente à fórmula $A\ f$ em CTL*.

Sintaxe

- A introdução destes quantificadores implica a distinção de dois tipos de fórmulas em CTL*: *fórmulas de estado* e *fórmulas de caminho*.
- Dado um conjunto de proposições atómicas P, uma fórmula de estado pode ser:
  - true, false ou qualquer $p \in P$;
  - $\neg f$, $f \vee g$ ou $f \wedge g$, onde f e g são fórmulas de estado; ou
  - $A\ f$ ou $E\ f$, onde f é uma fórmula de caminho.
- Uma fórmula de caminho pode ser:
  - Qualquer fórmula de estado f; ou
  - $\neg f$, $f \vee g$, $f \wedge g$, $X\ f$, $F\ f$, $G\ f$, $f\ U\ g$ ou $f\ R\ g$, onde f e g são fórmulas de caminho.

Semântica

- O facto de uma fórmula de estado CTL* f ser válida num estado s de uma estrutura de Kripke M será denotado por $M, s \models f$.
- Analogamente, temos $M, \pi \models g$ para uma fórmula de caminho g.
- Uma fórmula de estado f é válida numa estrutura de Kripke M (denotado $M \models f$) sse $M, i \models f$.

Validade de Fórmulas de Estado

- Para fórmulas de estado, a relação $\models$ é definida indutivamente da seguinte forma, onde $p \in P$ representa uma fórmula atómica, f e g são fórmulas de estado, e h uma fórmula de caminho.

$$
\begin{array}{lcl}
M, s \models \text{true} & & \\
M, s \not\models \text{false} & & \\
M, s \models p & \Leftrightarrow & p \in L(s) \\
M, s \models \neg f & \Leftrightarrow & M, s \not\models f \\
M, s \models f \vee g & \Leftrightarrow & M, s \models f \text{ ou } M, s \models g \\
M, s \models f \wedge g & \Leftrightarrow & M, s \models f \text{ e } M, s \models g \\
M, s \models A\ h & \Leftrightarrow & \forall \pi \in M \cdot \text{ se } \pi_0 = s \text{ então } M, \pi \models h \\
M, s \models E\ h & \Leftrightarrow & \exists \pi \in M \cdot \text{ se } \pi_0 = s \text{ então } M, \pi \models h
\end{array}
$$

Validade das Fórmulas de Caminho

- A semântica das fórmulas de caminho é idêntica à da LTL. Sendo f e g fórmulas de caminho e h uma fórmula de estado temos

$$
\begin{array}{lcl}
M, \pi \models h & \Leftrightarrow & M, \pi_0 \models h \\
M, \pi \models \neg f & \Leftrightarrow & M, \pi \not\models f \\
M, \pi \models f \vee g & \Leftrightarrow & M, \pi \models f \text{ ou } M, \pi \models g \\
M, \pi \models f \wedge g & \Leftrightarrow & M, \pi \models f \text{ e } M, \pi \models g \\
M, \pi \models X\ f & \Leftrightarrow & M, \pi^1 \models f \\
M, \pi \models F\ f & \Leftrightarrow & \exists i \geq 0 \cdot M, \pi^i \models f \\
M, \pi \models G\ f & \Leftrightarrow & \forall i \geq 0 \cdot M, \pi^i \models f \\
M, \pi \models f\ U\ g & \Leftrightarrow & \exists i \geq 0 \cdot M, \pi^i \models g \text{ e } \forall 0 \leq j < i \cdot M, \pi^j \models f \\
M, \pi \models f\ R\ g & \Leftrightarrow & \\
 & & \forall i \geq 0 \cdot \text{ se } \forall 0 \leq j < i \cdot M, \pi^j \not\models f \text{ então } M, \pi^i \models g
\end{array}
$$

Conectivas Mínimas e Expressividade

- As conectivas $\neg$, $\vee$, X, U, e A são suficientes para exprimir qualquer fórmula CTL*. Temos mais uma dualidade:

$$E\ f \equiv \neg A\ \neg f$$

- Alguns exemplos de propriedades que podem ser expressas em CTL* e que não podem ser expressas em LTL.
  - Invertibilidade: é possivel chegar ao estado i a partir de qualquer estado.

$$AG\ (EF\ i)$$

  - Em qualquer estado é possível prosseguir com a execução sem nunca visitar o estado s.

$$AG\ (EG\ \neg s)$$

  - Eventualmente será atingido um estado onde f se verifica em todos os estados sucessores.

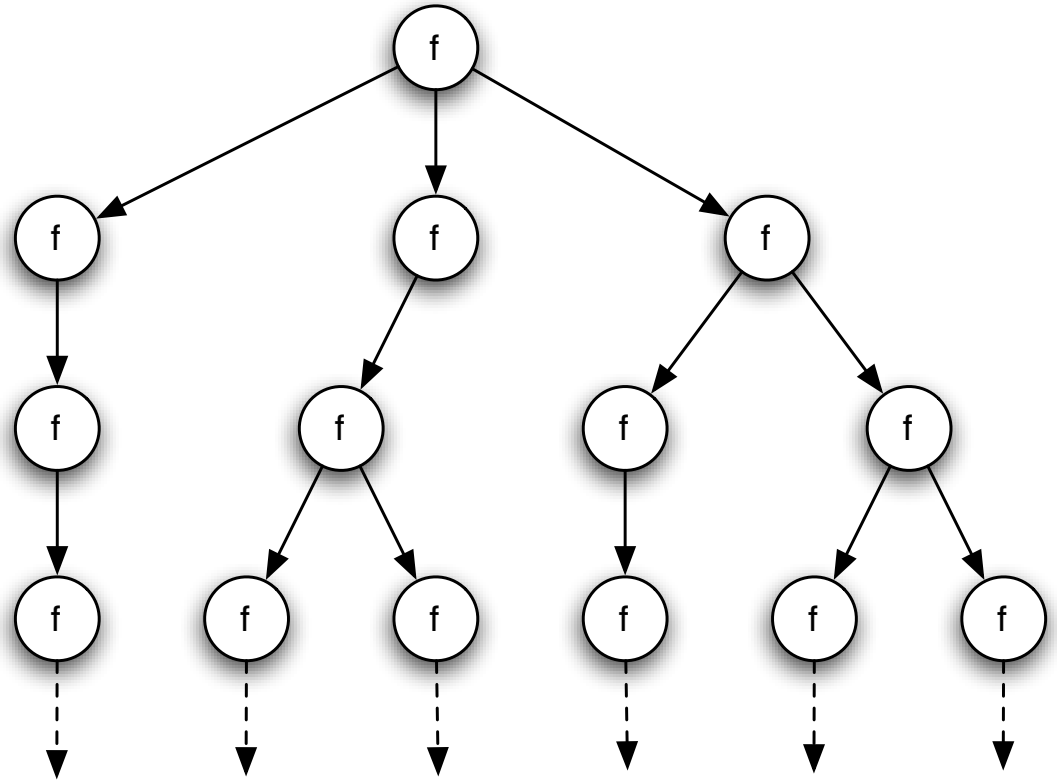
$$AF\ (AX\ f)$$

Computation Tree Logic

- A lógica temporal CTL é um subconjunto da lógica CTL*, onde um operador temporal tem que ser imediatamente precedido por um quantificador de caminho.
- Por exemplo, a fórmula $AGF\ p$ não é válida em CTL.
- A semântica define-se de forma idêntica à lógica CTL*.
- Esta lógica é interessante porque possui um mecanismo de verificação de modelos muito eficiente.
- Nomeadamente, é possível estabelecer a validade de uma fórmula em tempo linear no tamanho da mesma.
- No caso das lógicas LTL e CTL* esse tempo é exponencial no tamanho da fórmula.

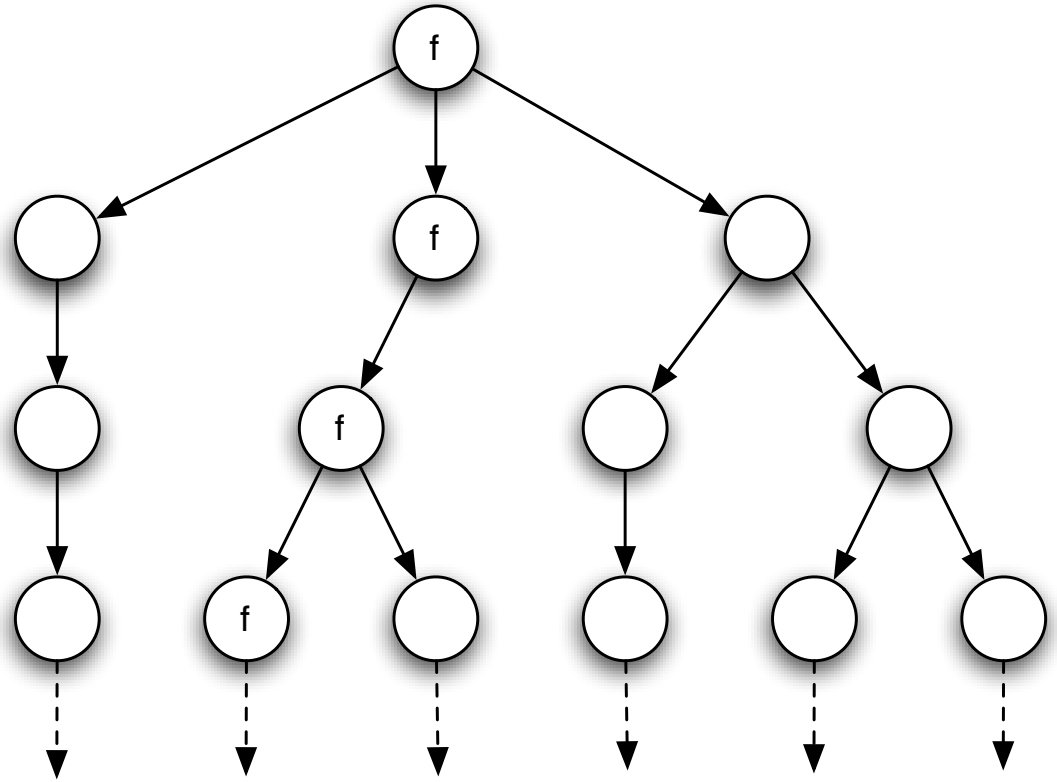
Combinações Típicas

$AG\ f$



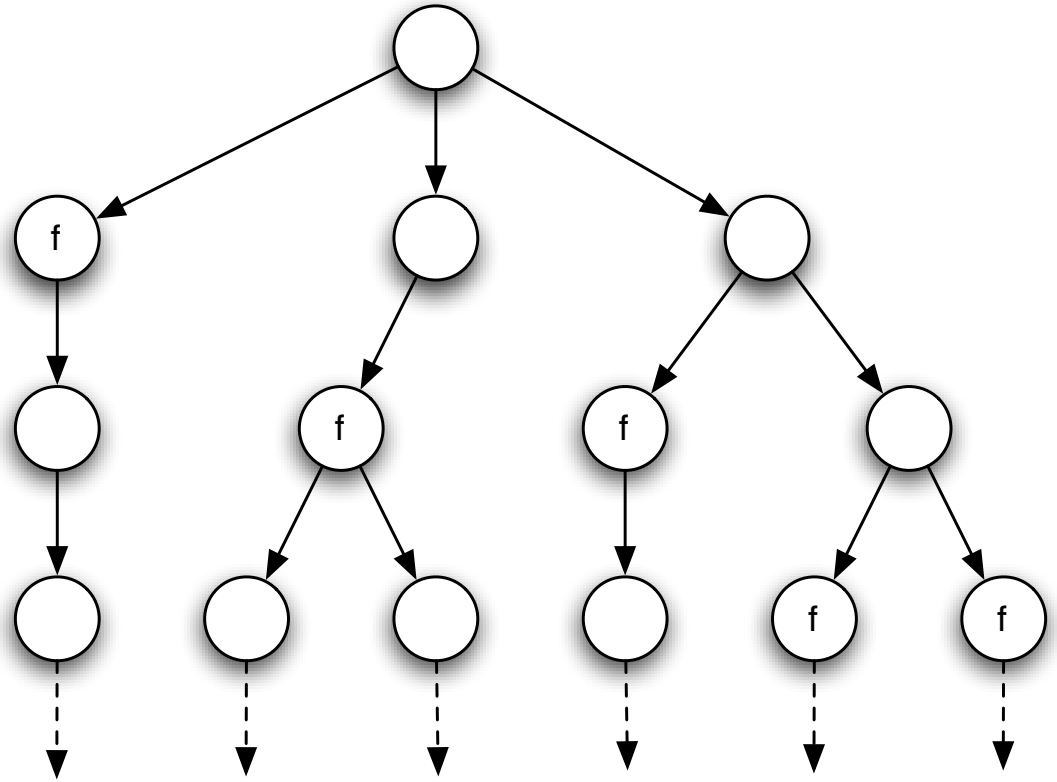
Combinações Típicas

$EG\ f$



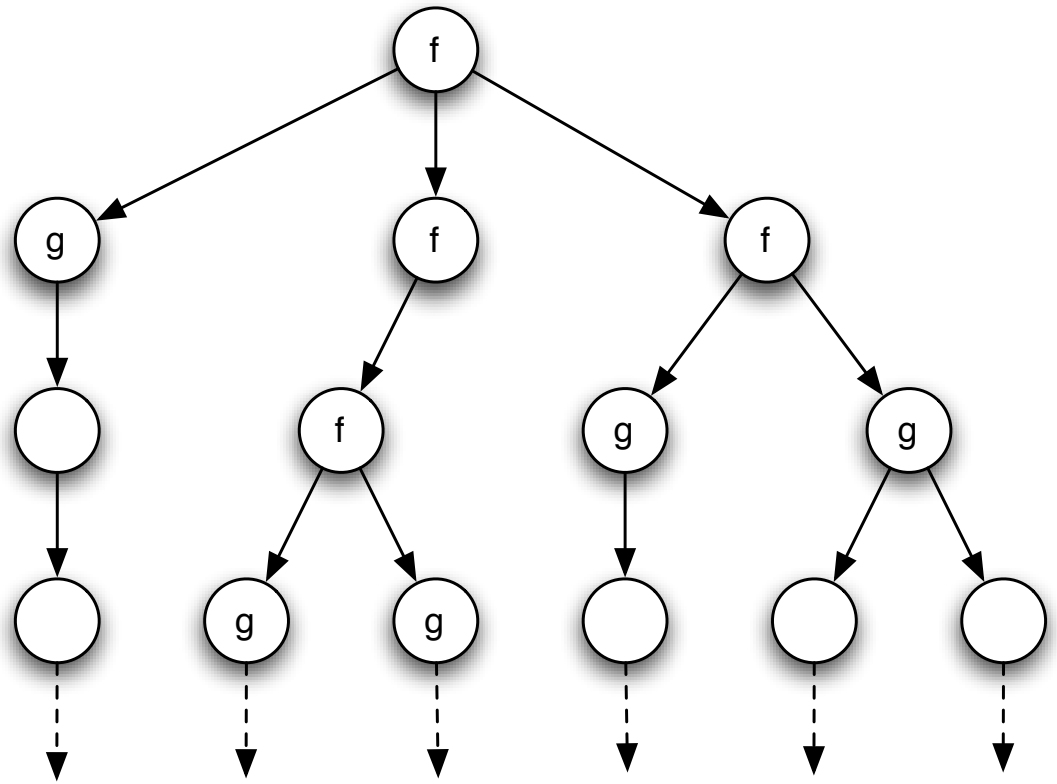
Combinações Típicas

$AF\ f$



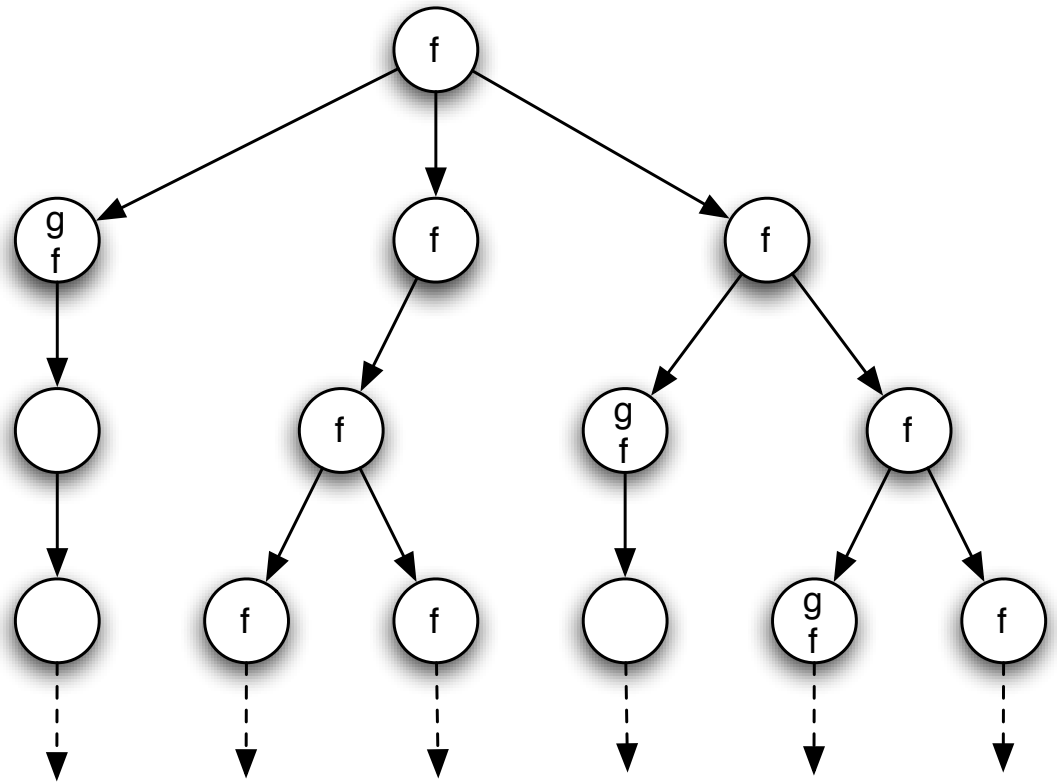
Combinações Típicas

$$A(f \ U \ g)$$

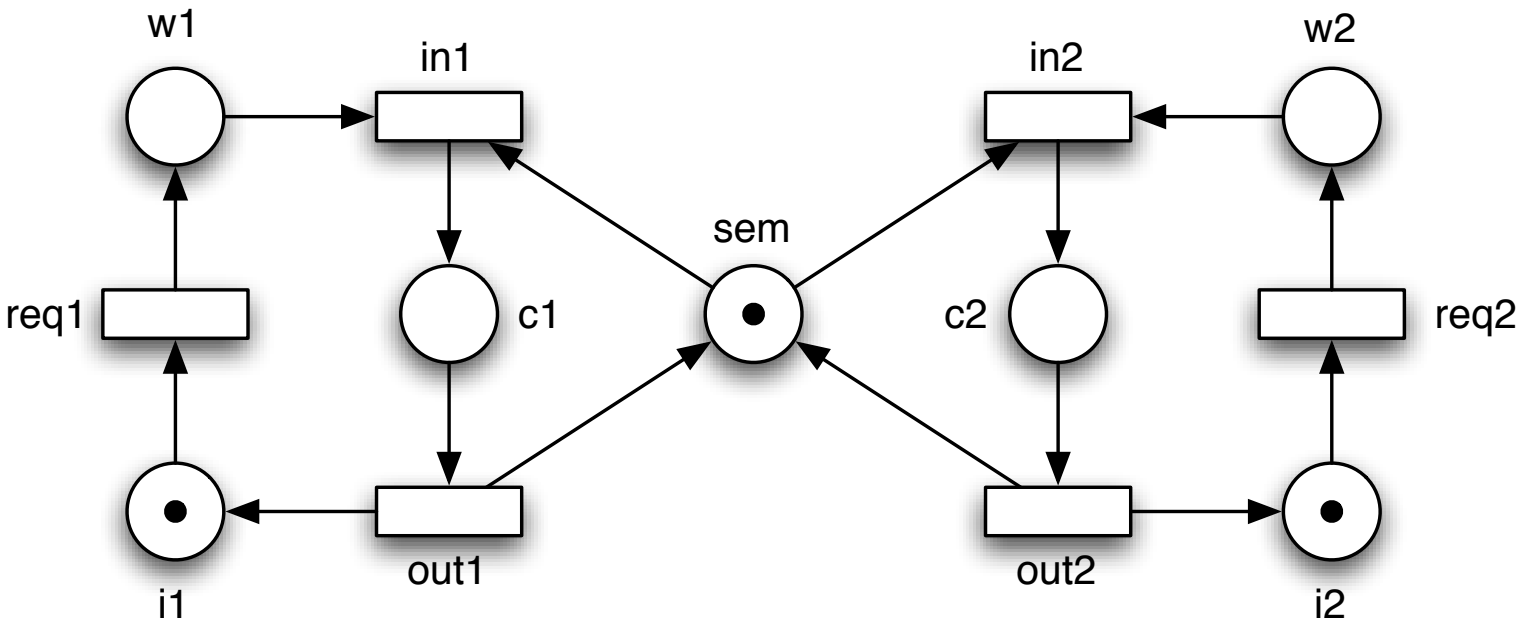


Combinações Típicas

$$A(g \ R \ f)$$



Exemplo



Exemplo

- Exclusão mútua: os processos nunca estão simultaneamente na região crítica.

$$AG\ \neg(c_1 \wedge c_2)$$

- Evolução: sempre que um processo está espera posteriormente estará na região crítica.

$$AG\ (w_1 \supset AF\ c_1) \wedge \ldots$$

- Sempre que um processo está espera tem a hipótese de no estado seguinte estar na região crítica.

$$AG\ (w_1 \supset EX\ c_1) \wedge \ldots$$

Exemplo

- Invertibilidade: há sempre a hipótese de regressar ao estado inicial.

$$AG\ EF\ (i_1 \wedge i_2 \wedge \ldots)$$

- O primeiro processo pode ficar bloqueado à espera de entrar na região crítica.

$$EF\ EG\ w_1$$

- Existe a hipótese de o primeiro processo a requisitar o acesso à região crítica ser o último a entrar.

$$AG\ (w_1 \wedge i_2 \supset E(c_2\ R\ \neg c_1))$$

CTL vs LTL

- A maior parte das propriedades pode ser expressa quer em CTL quer em LTL.

$$AG\ p \quad \text{vs} \quad G\ p$$

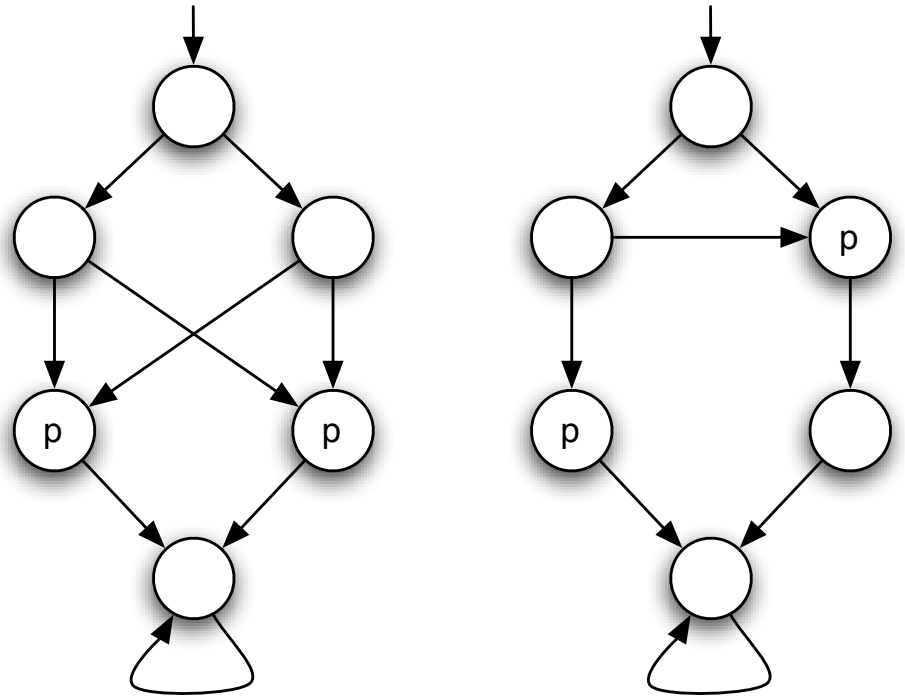
$$AG\ (p \supset AF\ q) \quad \text{vs} \quad G\ (p \supset F\ q)$$

- Mas a expressividade das duas é incomparável: cada uma das lógicas consegue exprimir propriedades que a outra não consegue.
- Por exemplo, a invertibilidade apenas pode ser expressa em CTL.

$$AG\ EF\ i$$

CTL vs LTL

- A propriedade $AF\ AX\ p$ também não pode ser expressa em LTL com semântica equivalente.
- Esta propriedade distingue as duas estruturas de Kripke seguintes, enquanto que a propriedade $FX\ p$ é válida em ambas.



CTL vs LTL

- Embora uma árvore de computação inclua todos os possíveis traços de uma estrutura de Kripke, a forma como a semântica está definida faz com que haja propriedades que apenas podem ser expressas em LTL.
- Por exemplo, a propriedade correspondente à fórmula $FG\ p$ não pode ser expressa em CTL.
- A seguinte estrutura de Kripke mostra que esta fórmula não é equivalente a $AF\ AG\ p$.