Lição 6: Lógicas para Processos

Luís Soares Barbosa

Sumário

Esta Lição é dedicada ao estudo de lógicas apropriadas para exprimir e verificar propriedades dos sistemas reactivos. Começaremos por introduzir uma lógica que permite quantificar as proposições relativamente às transições entre estados, desencadeadas pela ocorrência de interacções. Trata-se essencialmente da lógica de Hennessy-Milner que historicamente acompanhou o desenvolvimento de CCS.

Na modelação de sistemas reactivos, no entanto, é necessário discutir propriedades que referenciam não apenas transições elementares, mas também os cursos do seu comportamento global. Este tipo de propriedades, ditas temporais, serão formalizadas como limites, num sentido técnico preciso, das propriedades sobre transições elementares, conduzindo-nos a uma lógica muito poderosa – o μ-calculus *modal.*

A ênfase da Lição, no entanto, será colocada, na identificação e expressão de classes de propriedades relevantes. Em particular, discutiremos uma taxonomia de propriedades temporais a partir da divisão clássica entre requisitos de segurança *e* animação.

1 Processos e Propriedades

Motivação

As diversas equivalências observacionais discutidas nas Lições anteriores para CCS e π-*calculus*, permitem relacionar modelos de sistemas reactivos expressos nessas linguagens a diferentes níveis de abstracção (recordar os casos de estudo na Lição 4). De facto, uma das características das álgebras de processos é a possibilidade de nelas se descreverem tanto especificações muito abstractas como implementações razoavelmente concretas de sistemas reais. No entanto, ao recorrer a uma relação de equivalência para discutir a correcção entre dois níveis de descrição, estamos, de certa forma a *sobre-especificar* o problema na medida em que necessitamos de dispor, em cada um desses dois níveis, de uma visão do comportamento global do sistema.

No projecto de sistemas reactivos e concorrentes é, por vezes, muito importante poder fazer *menos* que isso, *i.e.*, registar apenas certas propriedades sobre o seu comportamento cuja satisfação deve ser garantida pelos processos que os implementam. Um conjunto de propriedades constitui uma especificação (eventualmente parcial) de um comportamento que, não se preocupando com todos os detalhes da sua descrição, regista os requisitos críticos que o cliente espera ver satisfeitos. Por exemplo, no projecto de um protocolo de comunicações o contrato entre o implementador e o cliente deve especificar que toda a mensagem transmitida é recebida, sem, contudo, descrever os mecanismos que garantem a sua verificação.

Suponhamos que pretendemos especificar a propriedade seguinte sobre um processo que modele uma viatura numa rede de táxis:

$\phi_0 =$ *A viatura pode recolher um passageiro ou ser alocada pela Central a um serviço pendente.*

Notas de apoio às aulas de *Interacção e Concorrência*, Lic. em Ciências da Computação, 2014-15.

Intuitivamente esta propriedade só será válida para uma viatura em serviço, *i.e.*, que previamente saiu da Central para as ruas. De facto, a noção de validade que nos interessa não é universal, mas *relativa* a um determinado estado da vida do processo que modela a viatura. A lógica que procuramos deverá, pois, ser capaz de exprimir esta *relatividade da verdade* em função dos estados alcançados pelo evoluir dos processos.

O universo que nos interessa é obviamente o conjunto $\mathbb{P}$ dos processos. Notemos, porém, que $\mathbb{P}$ é mais que um simples conjunto. De facto, está equipado com uma família de relações de transição que especificam de que forma um processo se transforma noutro pela ocorrência de uma determinada acção em Act. Os elementos de $\mathbb{P}$ constituem, como vimos atrás, os estados de um sistema de transição etiquetado por Act. Então certas propriedades poderão ser válidas num determinado estado e deixarem de o ser após a realização de uma transição. Por exemplo, a propriedade $\phi_1 =$ *O processo pode realizar uma transição por* a é verificada pelo processo $E \triangleq a.\,\mathbf{0} + b.c.\mathbf{0}$ mas já não pelo processo $\mathbf{0}$ que resulta da ocorrência de a. Este, por sua vez, verifica a propriedade $\phi_2 =$ *O processo está inactivo*, *i.e.,* nenhuma acção pode ser realizada.

Como poderemos exprimir a propriedade $\phi_3 =$ *O processo pode evoluir pela ocorrência de uma acção* a *para um processo inactivo*? Para isso necessitamos de introduzir na lógica um conectivo que estabeleça a validade de uma propriedade, não no estado representado pelo processo E, mas num estado alcançado por uma possível transição, etiquetada por a, a partir de E. Vamos representar esse conectivo por $\langle a \rangle$. A propriedade ϕ_3 escrever-se-á, então, como

$$\langle a \rangle \phi_2$$

Consideremos, agora, uma propriedade ligeiramente diferente: $\phi_4 =$ *O processo evolui por toda a ocorrência de uma acção* a *para um processo inactivo*. I.é, qualquer transição etiquetada por a impede o processo de prosseguir a sua actividade. Reparemos que E acima verifica ϕ_4, mas já $E' \triangleq a.\,\mathbf{0} + a.c.\mathbf{0}$ o não faz, uma vez que existe uma transição por a que conduz a $c.\mathbf{0}$ que, por sua vez, pode ainda realizar c. Há, em ϕ_4, uma quantificação universal sobre as transições por a: enquanto ϕ_3 se pode ler como *é possível uma transição por* a *que conduz a um estado que verifique* ϕ_2, ϕ_4 dever-se-á ler como *é necessário que uma transição por* a *conduza a um estado que verifique* ϕ_2. Ou, dito de outro modo, *toda* a transição por a conduz a um estado que verifica ϕ_2. Vamos exprimir isto recorrendo a um novo conectivo, dito de *necessidade*, e representado por $[a]$. A propriedade ϕ_4 escrever-se-á, então, como

$$[a] \phi_2$$

Note-se que $[a]$ e $\langle a \rangle$ são conectivos *duais*, no sentido em que o são os quantificadores universal e existencial na lógica de primeira ordem [1]. Atente-se, ainda, que nem todos os processos que verificam $[a]\phi_2$ verificam igualmente $\langle a \rangle \phi_2$. É, por exemplo, o caso do processo $F \triangleq d.\,\mathbf{0} + b.c.\mathbf{0}$ que falha $\langle a \rangle \phi_2$ por não exibir nenhuma transição por a conducente a $\mathbf{0}$, mas verifica $[a]\phi_2$ (exactamente pela mesma razão pela qual todos os habitantes da Lua são licenciados em Matemática).

Não há razão para nos restringirmos a acções individuais na parametrização deste dois novos conectivos. De facto, podemos parametriza-los por qualquer subconjunto não vazio de Act. Por exemplo a propriedade $\phi_5 =$ *O processo evolui por toda a ocorrência de* a *ou* b *para um processo inactivo* será expressa pela fórmula $[\{a, b\}]\phi_2$, que, por convenção, abreviamos para $[a, b]\phi_2$.

Pensemos, agora, de que forma podemos exprimir ϕ_2. Dizer que um processo é inactivo, significa que todas as transições a partir dele conduzem a estados "impossíveis", *i.e.,* que não são representados por nenhum termo em $\mathbb{P}$. Existe, de facto, uma fórmula que não é verificada por nenhum processo: a fórmula que representa a falsidade (absoluta) e que é constituida pela constante false. Assim, um processo inactivo é aquele em que toda a transição conduz a um processo que verifica false, *i.e.*,, na notação que estamos a introduzir, $[Act]$false. Logo, um processo que verifique $[Act]$false não poderá realizar qualquer transição.

[1] Recorde que, para um predicado $p(x)$, $\forall_x\ p(x)$ sse $\neg\exists_x\ \neg p(x)$.

Dualmente, todo o processo verifica true, a fórmula composta pela constante que representa a verdade (absoluta). Logo um processo que pode realizar pelo menos uma transição (etiquetada por qualquer acção em Act) satisfaz a fórmula $\langle Act\rangle$true. De facto, o conjunto dos processos que verificam $\langle Act\rangle$true coincide com $\mathbb{P} - \{E|\ E \sim \mathbf{0}\}$, como esperaríamos. Mais uma vez, por convenção, é usual abreviar $\langle Act\rangle$ e $[Act]$ por $\langle -\rangle$ e $[-]$, respectivamente, onde $-$ representa a diferença de conjuntos. Em geral, para qualquer $K \subseteq Act$, a expressão $-K$ como argumento de qualquer dos conectivos anteriores, abrevia $Act - K$ (logo, $-$ é abreviatura de $Act - \emptyset = Act$).

Conectivos Modais

Os conectivos de possibilidade e necessidade acima introduzidos permitem-nos qualificar propriedades cuja validade se afirma em estados da vida do processo alcançados pela realização de transições. Por isso mesmo são ditos *modais*, já que quando aplicados a uma fórmula ϕ, estabelecem o *modo* em que esta é verdadeira.

Suponhamos que introduzimos estes conectivos numa linguagem proposicional com os operadores booleanos usuais. A lógica obtida é dita *modal*. Um modelo para esta lógica é construido sobre o conjunto $\mathbb{P}$ dos processos (*i.e.*,, dos termos na álgebra de processos que temos vindo a considerar) equipado com a família de relações de transição $\{\xrightarrow{x} \mid x \in Act\}$ definida pela semântica operacional da linguagem dos processos. A característica fundamental da lógica reside na *relatividade* da noção de verdade.

Recordemos que a noção de modelo em lógica proposicional clássica é baseada numa atribuição de valores lógicos às variáveis, *i.e.*,, numa função $V : Var \longrightarrow \mathbb{B}$, onde Var é um conjunto de variáveis proposicionais. Isto significa que uma proposição é verdadeira ou falsa em termos absolutos: há um único referencial, determinado quando se fixa a atribuição V, no qual a validade de uma fórmula se define indutivamente sobre a sua estrutura (por exemplo, uma conjunção será verdadeira sse as suas parcelas o forem).

Aqui, e pelo contrário, a atribuição de valores às variáveis torna-se relativa aos estados da computação, exprimindo-se por uma função $V : Var \times \mathbb{P} \longrightarrow \mathbb{B}$ ou, equivalentemente, $V : Var \longrightarrow \mathcal{P}(\mathbb{P})$, *i.e.*,, a cada variável associa-se o conjunto de estados (processos) em que o seu valor é fixado como true.

Na próxima secção vamos definir rigorosamente esta lógica baseada nas duas modalidades referidas. Somos, porém, já agora capazes de traduzir numa fórmula a propriedade com que iniciamos esta digressão. Assim, ϕ_0 escrever-se-á como

$$\langle rec, alo\rangle \mathsf{true}$$

se rec e alo representarem as acções de recolha de um passageiro e de alocação a um serviço, respectivamente. Mas mais interessante será dizer que ϕ_0 se verifica após a saída da viatura para o trabalho diário, dito de outro modo,

$$[sai]\langle rec, alo\rangle \mathsf{true}$$

Nota 1 **[Lógicas Modais]**

As *lógicas modais* são particularmente adequadas para captar propriedades de sistemas em movimento, na medida em que nos permitem exprimir e discutir as relações entre o *mundo* actual, i.é, aquele em que se coloca o observador, e outros *mundos* possíveis, em relação com ele. Não surpreende, pois, a sua popularidade nas Ciências da Computação, ciências que estudam sistemas essencialmente dinâmicos.

Temos em mãos o exemplo dos processos. Aqui partimos de um conjunto de termos de uma álgebra (*e.g.* $\mathbf{0}$ ou $a.b.\ \mathbf{0}$ $+a.d.f.\mathbf{0}$), representando estados, que, por sua vez se relacionam pela ocorrência de transições etiquetadas por elementos

de Act. Outro exemplo é fornecido pelos sistemas de transição gerados pelos programas sequenciais, onde os estados são associações de endereços da memória a valores e as transições etiquetadas por programas. As lógicas modais adequadas a este caso designam-se por *lógicas de programas* ou *dinâmicas*, entre as quais se contam a lógica de Floyd-Hoare [Hoa69] e o cálculo de *weakest preconditions* [Dij76] que foram possivelmente abordadas no estudo da programação imperativa. Note-se, de passagem, que no primeiro caso a riqueza estrutural está localizada no conjunto de estados (*cf.* a estrutura induzida pela álgebra de processos), enquanto, no segundo, reside nas etiquetas (onde se define uma álgebra de programas cujos termos são, por exemplo, '$p_1 ; p_2$', 'if b then p_1 else p_2' ou 'while b do p_1'). Em ambos, a noção computacional de *estado* é introduzida na lógica, num sentido matematicamente preciso, como patamar de interpretação das fórmulas.

A origem das lógicas modais é, porém, muito anterior e o seu campo de aplicabilidade muito mais vasto. Classicamente estas lógicas diferenciam afirmações que são necessariamente verdadeiras de outras que apenas o podem ser. Proposições sempre verdadeiras ou falsas são ditas, respectivamente, *necessárias* e *impossíveis*. Por seu lado, uma proposição *contingente* não será nem necessária nem impossível, enquanto se classificam de *possíveis* todas as proposições não impossíveis. O significado exacto desta classificação é definido de diversas formas. O quatro conceitos — *necessidade, possibilidade, impossibilidade* e *contingência* — são ditos *modais* porque em lógica medieval eram encarados como os *modos* segundo os quais se discutia a veracidade de uma proposição.

Os sistemas de transição referidos pertencem à classe das estruturas — ditas de *Kripke* — classicamente utilizadas para interpretação das lógicas modais. Estas são formadas por um conjunto W (de *mundos*) sobre o qual se toma uma relação binária, dita de *acessibilidade*. De facto, em Matemática, as lógicas modais são utilizadas como o contexto mais adequado para descrever, classificar e raciocinar sobre *estruturas relacionais* genéricas (ver [BRV95] para uma introdução razoavelmente acessível).

No nosso caso, W é o conjunto dos termos gerados pela álgebra de processos considerada. Por outro lado, generalizamos a relação de acessibilidade a uma família de relações de transição etiquetadas.

---

2 Uma Lógica Modal para Processos

A Linguagem Modal

A discussão feita na secção anterior permite-nos introduzir a seguinte linguagem modal — que designaremos por $\mathbb{L}_M$ — para exprimir propriedades comportamentais dos processos. Trata-se de uma pequena variante da *lógica de Hennessy-Milner* introduzida em [HM85] e que constitui um ingrediente clássico da teoria dos processos (o capítulo 10 de [Mil89] é-lhe dedicado). A diferença, relativamente à versão original, consiste no facto de aqui os conectivos modais serem parametrizados por conjuntos de acções, e não por acções singulares. Assim, consideremos,

$$\phi \ ::= \ \mathsf{true} \mid \mathsf{false} \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \langle K \rangle \phi \mid [K]\phi$$

onde $K \subseteq Act$.

A relação fundamental entre um processo E e uma fórmula ϕ é a *relação de satisfação*, que se representa por $E \models \phi$. Esta é definida indutivamente sobre a estrutura das fórmulas. Assim,

$$\begin{array}{lll}
E \models \mathsf{true} & & \\
E \not\models \mathsf{false} & & \\
E \models \phi_1 \wedge \phi_2 & \text{sse} & E \models \phi_1 \ \wedge \ E \models \phi_2 \\
E \models \phi_1 \vee \phi_2 & \text{sse} & E \models \phi_1 \ \vee \ E \models \phi_2 \\
E \models \langle K \rangle \phi & \text{sse} & \exists_{F \in \{E' \mid E \xrightarrow{a} E' \text{ e } a \in K\}} \cdot F \models \phi \\
E \models [K]\phi & \text{sse} & \forall_{F \in \{E' \mid E \xrightarrow{a} E' \text{ e } a \in K\}} \cdot F \models \phi
\end{array}$$

As propriedades dos processos que podemos exprimir através desta lógica são essencialmente propriedades relacionadas com a necessidade ou possibilidade imediatas de se realizarem de-

terminadas acções. Vejamos alguns exemplos. Consideremos de novo a especificação de um controlador de exclusão mútua:

$$\begin{aligned} Sem &\triangleq get.put.Sem \\ P_i &\triangleq \overline{get}.c_i.\overline{put}.P_i \\ S &\triangleq \mathsf{new}\,\{get, put\}\ (Sem \mid (|_{i\in I}\ P_i)) \end{aligned}$$

Podemos facilmente verificar os factos seguintes:

- $Sem \models \langle get\rangle \mathsf{true}$ que estabelece a possibilidade do processo Sem realizar get. De facto,

  $$\exists_{F\in\{Sem' \mid\ Sem \xrightarrow{get} Sem'\}}\ .\ F \models \mathsf{true}$$

  basta tomar $F = put.Sem$.

- Contudo, $Sem \models [put]\mathsf{false}$, *i.e.*, Sem não pode realizar a acção put. De facto, o conjunto de transições $T = \{Sem' \mid\ Sem \xrightarrow{put} Sem'\}$ é vazio pelo que a expressão $\forall_{F\in T}\ .\ F \models \mathsf{false}$ se torna vacuosamente verdadeira.

- A única acção inicialmente permitida ao processo S é a acção não observável τ que resulta da sincronização de um get do semáforo com $\overline{get}$ realizado por um dos processos por ele controlados. Podemos exprimir isto pela fórmula $S \models [Act - \{\tau\}]\mathsf{false}$, ou, abreviadamente, $\models [-\tau]\mathsf{false}$.

- Após a ocorrência de τ, contudo, S pode realizar qualquer dos eventos críticos $c_1, c_2, ..., c_i$, *i.e.*,, $[\tau]\langle c_1, c_2, ..., c_i\rangle \mathsf{true}$.

- De forma análoga podemos concluir que após uma sincronização inicial com o semáforo (pela ocorrência conjunta de get em Sem e $\overline{get}$ no sub-processo P_j) e a realização do evento crítico c_j por P_j, a ocorrência de uma nova sincronização com o semáforo (agora via put e $\overline{put}$) é inevitável. Temos, portanto, $S \models [\tau][c_j](\langle -\rangle \mathsf{true} \wedge [-\tau]\mathsf{false})$.

Como vimos no último exemplo a *inevitabilidade* da ocorrência de uma acção a é expressa pela fórmula $\langle -\rangle \mathsf{true} \wedge [-a]\mathsf{false}$. A primeira parcela da conjunção indica que existe pelo menos uma transição possível, enquanto a segunda estabelece que todas as transições, excepto as etiquetadas por a, conduzem a um estado que verifica false. Como tal estado não existe, podemos concluir que, num processo que satisfaça a fórmula, apenas existem transições por a (e existem mesmo porque se força $\langle -\rangle \mathsf{true}$).

Vimos já que $\langle -\rangle \mathsf{true}$ exprime possibilidade de evolução (há pelo menos uma transição disponível) e que $[-]\mathsf{false}$ traduz terminação ou *deadlock*. Claramente, $\mathbf{0} \models [-]\mathsf{false}$. Curiosamente, as fórmulas duais — $\langle -\rangle \mathsf{false}$ e $[-]\mathsf{true}$ — são desinteressantes. A primeira é verificada por um processo E tal que

$$\exists_{F\in\{E' \mid\ E \xrightarrow{x} E' \wedge x\in Act\}}\ .\ F \models \mathsf{false}$$

Como nenhuma transição pode conduzir a um estado que não existe, $\langle -\rangle \mathsf{false}$ é equivalente à constante false. Dualmente, a segunda fórmula equivale a true. Por seu lado, a possibilidade de realização de uma sequência $a_1, a_2, a_3, ..., a_n$ de acções é captada na fórmula $\langle a_1\rangle\langle a_2\rangle\langle a_3\rangle ...\langle a_n\rangle \mathsf{true}$. Notemos, por fim, que a verificação de que um processo satisfaz uma determinada propriedade modal se faz muito simplesmente por aplicação da definição de $\models$. Em particular, não é necessário calcular o grafo de transições do processo.

Equivalência Modal e Bissimulação

A relação de satisfação em $\mathbb{L}_M$ permite-nos definir um nova equivalência entre processos — a que resulta da satisfação do mesmo conjunto de fórmulas. De facto, para qualquer conjunto Γ, finito ou infinito, de fórmulas podemos definir a seguinte equivalência em $\mathbb{P}$:

$$E \equiv_\Gamma F \quad \text{sse} \quad \forall_{\phi \in \Gamma} \, . \, E \models \phi \text{ sse } F \models \phi$$

Se, por exemplo, Γ fôr o conjunto de todas as fórmulas do tipo $\langle x_1 \rangle \langle x_2 \rangle ... \langle x_n \rangle \mathsf{true}$, a equivalência $\equiv_\Gamma$ relacionará os processos capazes de realizarem as mesmas sequências de acções. Neste caso os processos $a.b.\,\mathbf{0} + a.c.\mathbf{0}$ e $a.(b.\,\mathbf{0} + c.\mathbf{0})$, que, como bem sabemos, não são bissimilares, são identificados por esta equivalência. Já se Γ se restringisse às fórmulas do tipo $\langle x_1 \rangle \langle x_2 \rangle \langle x_3 \rangle ... \langle x_n \rangle [-] \mathsf{false}$, estaríamos a considerar equivalentes os processos que realizam as mesmas sequências *completas* de acções, *i.e.*,, aquelas que conduzem à "morte" (terminação) do seu comportamento. Os processos poderiam divergir nas restantes sequências de acções.

Dizemos, em particular, que os processos E e F são *modalmente equivalentes*, e escrevemos $E \equiv F$ quando $E \equiv_\Gamma F$ e Γ é o conjunto de *todas* as fórmulas bem formadas da lógica.

Uma questão muito importante é saber de que forma a equivalência modal se relaciona com a bissimulação (estrita) sobre a qual construímos o nosso cálculo de processos. O resultado fundamental é o seguinte

Lema 1 *Dados dois processos E e F, se $E \sim F$ então $E \equiv F$.*

Prova. A prova procede por indução sobre a estrutura das fórmulas. Assim, suponhamos que $E \sim F$ e tentemos mostrar que $E \models \phi$ sse $F \models \phi$ para toda a fórmula ϕ. Os casos de base — $\phi = \mathsf{true}$ ou $\phi = \mathsf{false}$ — são triviais. Quando ϕ é uma conjunção ou uma disjunção, o resultado decorre imediatamente da hipótese de indução. Vejamos, pois, os casos em que o conectivo principal de ϕ é uma modalidade, que são os verdadeiramente interessantes.

Seja, então, $\phi = \langle K \rangle \psi$, com $K \subseteq Act$. Se $E \models \phi$ é porque existe um $E' \in \mathbb{P}$ tal que $E \stackrel{a}{\longrightarrow} E'$, para algum $a \in K$, e $E' \models \psi$. Como $E \sim F$, sabemos que existe uma transição $F \stackrel{a}{\longrightarrow} F'$ para um $F' \in \mathbb{P}$ tal que $E' \sim F'$. Como, por hipótese de indução $E' \equiv F'$, temos que $F' \models \psi$. Então, de novo pela definição do conectivo $\langle K \rangle$, concluimos que $F \models \langle K \rangle \psi$. O argumento é similar para o caso $\phi = [K]\psi$.

□

O converso deste lema não é verdadeiro no caso geral. Por exemplo, considerem-se os processos $A \triangleq \sum_{i \geq 0} A_i$, onde $A_0 \triangleq \mathbf{0}$ e $A_{i+1} \triangleq a.A_i$, e $A' \triangleq A + \underline{fix}\,(X = a.X)$. Claramente, A e A' verificam as mesmas propriedades modais. No entanto $A \not\sim A'$ porque, caso contrário, a transição $A' \stackrel{a}{\longrightarrow} \underline{fix}\,(X = a.X)$ deveria ser correspondida por uma transição de A por a para um processo bissimilar a $\underline{fix}\,(X = a.X)$. Ora A apenas pode transitar para um A_j, para algum $j \geq 0$, e, obviamente, $A_j \not\sim \underline{fix}\,(X = a.X)$.

O resultado é contudo válido para uma classe restrita, embora na prática suficientemente ampla, de processos — os processos com *imagem finita*. Para precisar esta condição, defina-se, para um processo E e uma acção $x \in Act$, o conjunto $\{E' |\ E \stackrel{x}{\longrightarrow} E'\}$ dito dos sucessores de E por x. Sempre que os elementos do conjunto $\{F |\ \exists_{\omega \in Act^*} \, . \, E \stackrel{\omega}{\longrightarrow} F\}$ têm conjuntos finitos de sucessores para todo o $x \in Act$, E é dito um processo com *imagem finita*. Não é difícil verificar que os processos discutidos no último exemplo não estão nestas condições. Assim,

Lema 2 *Dados dois processos E e F com imagem finita, se $E \equiv F$ então $E \sim F$.*

Prova. Notemos que a condição *imagem finita* significa que todos os estados alcançáveis a partir do processo em causa têm um número finito de sucessores. Suponhamos, então, que $E \equiv F$ e procuremos encontrar

uma bissimulação que os contenha. A candidata óbvia é a própria equivalência modal $\equiv$. Verifiquemos, pois, se esta é capaz de satisfazer as duas cláusulas definidoras de uma bissimulação. Como bem sabemos essas cláusulas impõem condições simétricas, pelo que o argumento a utilizar na verificação de uma delas é também (simetricamente) válido para a segunda (o que é óptimo porque reduz a prova a metade!).

Assim, suponhamos que $E \xrightarrow{a} E'$, para algum $a \in Act$, e que $E \equiv F$. Tentemos contrariar o resultado formulando a hipótese de que não existe nenhum $F' \in \mathbb{P}$ tal que $F \xrightarrow{a} F'$ e $E' \equiv F'$.

Seja $S = \{G |\ F \xrightarrow{a} G\}$. O conjunto S é necessariamente não vazio, porque, se fosse vazio, teríamos $F \models [a]\mathsf{false}$ ao passo que $E \models \langle a \rangle \mathsf{true}$ uma vez que estamos a supôr a existência de uma transição de E por a. Nesse caso não seria válida a equivalência $E \equiv F$, que constitui a hipótese do lema. Notemos, porém, que S é um conjunto *finito* $\{G_1, G_2, ..., G_n\}$, dado que F tem imagem finita.

Se, de facto, se não tem $E' \equiv F'$, temos que, para cada $G_i \in S$, existe uma fórmula ϕ_i tal que $E' \models \phi_i$ mas $G_i \not\models \phi_i$. Segue-se, pois, que $E \models \langle a \rangle(\phi_1 \wedge \phi_2 \wedge ... \wedge \phi_n)$, o mesmo não sendo válido para F. Mas então, E não é modalmente equivalente a F, o que contradiz a hipótese. O resultado segue, pois, por contra-recíproco. Notemos, contudo, que o argumento utilizado até aqui só prova o lema porque permanece válido quando, em vez de E e F, começamos por considerar dois quaisquer processos no espaço dos processos alcançáveis a partir de E e de F, respectivamente. Da condição do lema decorre que qualquer desses processos tem um conjunto finito de sucessores para todo o $a \in Act$.

□

Reparemos em que ponto desta última prova foi usada a condição de imagem finita. Notemos que se F tivesse um número não finito de sucessores iríamos necessitar de uma conjunção infinita na definição de ψ. A lógica que introduzimos é finitária pelo que tal conjunção não faz sentido. Se eliminarmos essa restrição da lógica, o que diminui a sua operacionalidade, podemos igualmente eliminar a condição de imagem finita sobre os processos chegando a um resultado mais geral, mas (computacionalmente) menos relevante.

3 Lógica Modal e Transições Observáveis

Os conectivos modais que introduzimos exploram a estrutura das transições em $\mathbb{P}$ do tipo $\xrightarrow{x}$, para $x \in Act$, relativizando a validade das fórmulas a conjuntos de estados alcançados através de determinadas transições. Sobre $\mathbb{P}$, porém, é possível definir outras famílias de relações que são computacionalmente relevantes. Uma delas é a das relações de *transição observável*, $\stackrel{a}{\Longrightarrow} \subseteq \mathbb{P} \times \mathbb{P}$, em que as etiquetas são elementos de $L \cup \{\epsilon\}$. Recorde-se que uma $\stackrel{\epsilon}{\Longrightarrow}$-transição corresponde a zero ou mais transições elementares não observáveis (*i.e.*, etiquetadas por τ).

Vamos, pois, começar por introduzir em $\mathbb{L}_M$ dois novos conectivos modais que exprimem, respectivamente, a possibilidade e a necessidade de uma propriedade ser válida após a realização de uma quantidade arbitrária de comportamento não observável. Para definir estes operadores consideremos mais dois casos na definição da relação de satisfação $\models$.

$$E \models \langle\!\langle \rangle\!\rangle \phi \quad \text{sse} \quad \exists_{F \in \{E' |\ E \stackrel{\epsilon}{\Longrightarrow} E'\}} . F \models \phi$$
$$E \models [\![ ]\!] \phi \quad \text{sse} \quad \forall_{F \in \{E' |\ E \stackrel{\epsilon}{\Longrightarrow} E'\}} . F \models \phi$$

Por abreviatura podemos agora definir as "versões observáveis" de $\langle K \rangle$ e $[K]$, para $K \subseteq L$. Assim,

$$\langle\!\langle K \rangle\!\rangle \phi \stackrel{\text{abv}}{=} \langle\!\langle \rangle\!\rangle \langle K \rangle \langle\!\langle \rangle\!\rangle \phi$$
$$[\![K]\!] \phi \stackrel{\text{abv}}{=} [\![ ]\!] [K] [\![ ]\!] \phi$$

Vejamos alguns exemplos de propriedades exprimíveis à custa deste conectivos.

- $\langle\!\langle a \rangle\!\rangle \mathsf{true}$ representa a capacidade de realização observável da acção a. Por exemplo, o processo S discutido anterior (que simulava uma situação de exclusão mútua) satisfaz a fórmula $\langle\!\langle c_j \rangle\!\rangle \mathsf{true}$, indicando que o processo P_j pode realizar a acção crítica respectiva.
- Similarmente a fórmula $\langle\!\langle a_1 \rangle\!\rangle \langle\!\langle a_2 \rangle\!\rangle \langle\!\langle a_3 \rangle\!\rangle ... \langle\!\langle a_n \rangle\!\rangle \mathsf{true}$ exprime a possibilidade de realização da sequência observável de acções $a_1, a_2, a_3, ..., a_n$.
- Por seu lado $[\![-]\!]\mathsf{false}$ traduz a impossibilidade de realização de comportamento observável. A propriedade é satisfeita, não apenas por $\mathbf{0}$ (que de resto verifica também $[-]\mathsf{false}$), mas igualmente por processos como $D \triangleq \tau.D + \mathbf{0}$ ou $\mathsf{new}\,\{a\}\ A \mid B$ onde $A \triangleq a.A$ e $B \triangleq \overline{a}B$, que se envolvem numa cadeia interminável de acções invisíveis.

Um caso interessante é o da propriedade que estabelece a inevitabilidade da ocorrência de a, mas agora de um ponto de vista observacional. A inevitabilidade foi já expressa como $\langle - \rangle \mathsf{true} \wedge [-a]\mathsf{false}$. Reparemos, contudo, que a versão observacional desta fórmula — $\langle\!\langle - \rangle\!\rangle \mathsf{true} \wedge [\![-a]\!]\mathsf{false}$ — é demasiado fraca. A fórmula é, por exemplo, satisfeita pelo processo $P \triangleq a.P + \tau.\mathbf{0}$, apesar de haver a possibilidade em P de a acabar por nunca ocorrer. Podemos fortalecê-la forçando a possibilidade de transições observáveis ocorrerem após a realização de comportamento invisível, *i.e.*,, $[\![\ ]\!]\langle\!\langle - \rangle\!\rangle \mathsf{true} \wedge [\![-a]\!]\mathsf{false}$. Esta fórmula já não é satisfeita por P, uma vez que $P \stackrel{\epsilon}{\Longrightarrow} \mathbf{0}$ e $\mathbf{0} \not\models \langle\!\langle - \rangle\!\rangle \mathsf{true}$. O problema permanece, porém, com o processo $P \triangleq a.P + \tau.P$, que, podendo igualmente nunca realizar a, satisfaz $[\![\ ]\!]\langle\!\langle - \rangle\!\rangle \mathsf{true}$.

O problema reside na quantidade de movimento não observável admissível na definição dos novos conectivos modais. Se desejarmos exprimir que uma acção ocorre no termo da actividade interna do processo devemos limitar finitamente essa actividade. Dito de outro modo, temos de garantir que o processo é *convergente*, *i.e.*,, incapaz de se comprometer num ciclo perpétuo de actividade interna não observável. Para isso vamos considerar um novo conectivo modal similar a $[\![\ ]\!]$, mas com informação adicional sobre convergência.

$$E \models [\![\downarrow]\!]\phi \quad \text{sse} \quad E\downarrow \wedge \forall_{F \in \{E' \mid E \stackrel{\epsilon}{\Longrightarrow} E'\}} . F \models \phi$$

onde $E\downarrow$ significa que o processo E é convergente.

Podemos, agora, definir uma fórmula para a inevitabilidade que exclua a possibilidade de divergência:

$$[\![\downarrow]\!]\langle\!\langle - \rangle\!\rangle \mathsf{true} \wedge [\![-a]\!]\mathsf{false}$$

Diversas combinações destes conectivos revelam-se úteis na especificação de processos. Por exemplo, podemos definir por abreviatura operadores que forcem a convergência do comportamento interno até determinadas transições ocorrerem ou uma dada fórmula ser satisfeita. É o caso de $[\![\downarrow K]\!] \stackrel{\text{abv}}{=} [\![\downarrow]\!][K][\![\ ]\!]$, $[\![K\downarrow]\!] \stackrel{\text{abv}}{=} [\![\ ]\!][K][\![\downarrow]\!]$ ou mesmo $[\![\downarrow K\downarrow]\!] \stackrel{\text{abv}}{=} [\![\downarrow]\!][K][\![\downarrow]\!]$. A semântica destes conectivos derivados é a que se esperaria. Por exemplo,

$$E \models [\![\downarrow K]\!]\phi \quad \text{sse} \quad E\downarrow \wedge \forall_{F \in \{E' \mid E \stackrel{a}{\Longrightarrow} E' \wedge a \in K\}} . F \models \phi$$

Como exercício, dever-se-á verificar que, apesar de $[-]\mathsf{true}$ ser equivalente a true, e, por isso válida em qualquer processo, apenas os processo convergentes satisfazem $[\![\downarrow -]\!]\mathsf{true}$. Assim, a fórmula $[\![\downarrow a_1\downarrow]\!][\![\downarrow a_2\downarrow]\!]...[\![\downarrow a_n\downarrow]\!]\mathsf{true}$ exprime a realização convergente de uma sequência $a_1a_2...a_n$ de acções observáveis.

4 Propriedades Temporais e Computações

Transições *vs* Computações

Ao longo deste curso temos sublinhado a ideia de que os *sistemas de transição* constituem uma ferramenta adequada para a definição da semântica dos processos (e, mais genericamente, dos programas e dos sistemas computacionais). Vimos, também, que sobre eles podemos interpretar lógicas — a que chamamos *modais* — que relativizam a validade das proposições aos estados que os processos vão alcançando pela realização de transições. Vimos, por fim, como estas são capazes de exprimir propriedades *locais* dos processos, nomeadamente acerca da possibilidade ou da necessidade de ocorrência de determinadas acções, que encaramos como transformadores de estados.

Na especificação de sistemas concorrentes deparamos, no entanto, com a necessidade de formular propriedades, não apenas sobre transições particulares, mas também sobre a evolução global dos processos, *i.e.*, os padrões comportamentais que estes podem ou devem exibir.

Sabemos, por exemplo, representar modalmente o facto de uma viatura, no exemplo da Rede de Táxis, poder regressar à Central, *i.e.*,

$$\phi_1 \;=\; \langle reg \rangle \mathsf{true}$$

A possibilidade deste regresso não é contudo válida em todos os estados do processo que modela a viatura. Em particular, é razoável supor que não se verifica no estado inicial do processo (no qual a viatura se encontra, precisamente, na garagem da Central). O que, em rigor, gostaríamos de exprimir é o facto de, ao longo da sua vida, a viatura acabar por atingir um (ou mais) estado(s) em que o regresso é uma possibilidade efectiva. Dito de outra forma, ϕ_1 verifica-se ou, então, existe pelo menos uma transição que conduz a um estado em que ϕ_1 se verifica ou, então, existe pelo menos uma transição que ... O que seríamos tentados a escrever como

$$\phi_2 \;=\; \langle reg \rangle \mathsf{true} \vee \langle - \rangle \langle reg \rangle \mathsf{true} \vee \langle - \rangle \langle - \rangle \langle reg \rangle \mathsf{true} \vee \langle - \rangle \langle - \rangle \langle - \rangle \langle reg \rangle \mathsf{true} \vee \;\; ...$$

Não podendo, sem comprometer o seu significado, substituir ϕ_2 por uma disjunção finita, deparamos com uma fórmula que não é bem formada na lógica que atrás introduzimos. Vamos, pois, tentar formalizar o significado que intuitivamente atribuímos às reticências.

O ponto fundamental reside no facto de estarmos, agora, interessados em propriedades, não das transições individualmente consideradas, mas das *sequências de transições* que os processos realizam. Num processo E tais sequências constituem *caminhos*, ou cursos de evolução, ao longo do sistema de transição gerado por E.

Estamos particularmente interessados em *caminhos maximais*, *i.e.*, nas sequências infinitas ou nas finitas que conduzem à terminação do processo, ou seja, a estados a partir dos quais não são possíveis quaisquer transições. No que se segue referir-nos-emos a estes caminhos como as *computações* dos processos. Formalmente, a computação de um processo E é uma sequência $\langle E_0, E_1 ... E_n \rangle$ de estados (representados por expressões no cálculo de processos) tal que

1. $E_0 = E$
2. $n = \infty$ ou a partir do termo E_n não existem mais derivações por elementos de Act
3. $\forall_{0 \leq i \leq n} \exists_{a \in Act}. \;\; E_i \xrightarrow{a} E_{i+1}$

Um outro exemplo da necessidade de raciocinar sobre as computações, e não apenas sobre as

transições individuais, é fornecido pelos processos A e A' discutidos atrás. Aí definimos

$$A \triangleq \sum_{i \geq 0} A_i \quad \text{com} \;\; A_0 \triangleq \mathbf{0} \text{ e } A_{i+1} \triangleq a.A_i$$
$$A' \triangleq A + D \quad \text{com} \;\; D \triangleq a.D$$

e argumentamos que, apesar de $A \nsim A'$, não havia nenhuma propriedade modal que os distinguisse. De facto, cada A_i falha a propriedade $\langle a \rangle^{i+1}\mathsf{true}$, que A' verifica. Mas teríamos, de novo, que recorrer a uma disjunção infinita para construir uma fórmula que qualquer A_i falhasse. O que, em verdade, distingue um processo do outro é a capacidade de A' realizar a acção a indefinidamente e essa é uma capacidade a 'longo prazo', uma capacidade *temporal*.

Propriedades Temporais

Vamos designar por *propriedades temporais* aquelas que apenas faz sentido exprimir sobre as computações dos processos. Mas que propriedades são essas? Classicamente, distinguem-se dois tipos básicos:

- As propriedades de *segurança*, que se relacionam com a invariância de uma determinada característica comportamental (por exemplo, a ausência de *deadlock*). Intuitivamente podemos dizer que uma propriedade deste tipo estabelece que 'as coisas indesejáveis não acontecem'. No já muito batido exemplo do controlador de exclusão mútua, uma propriedade de segurança seria $\psi_1 =$ *A exclusão mútua não é violada*. No domínio da programação sequencial o típico requisito de segurança é a exigência de que o programa produza resultados correctos e é conhecido como a *correcção parcial* do programa.
- As propriedades de *animação* que estabelecem a necessidade ou a possibilidade de 'coisas desejáveis acabarem por se verificar'. Voltando ao exemplo anterior notemos que o requisito de segurança ψ_1 pode ser satisfeito por um controlador que permanentemente autorize apenas um dos processos a entrar na sua região crítica, ficando o outro incapaz de prosseguir. Ou até por um semáforo autista que impeça qualquer processo de aceder à respectiva região crítica. Em qualquer dos casos, mesmo que por omissão, se garante a exclusão mútua. Assim, a especificação do controlador não pode omitir uma propriedade que estabeleça que $\psi_2 =$ *cada processo é capaz, de eventualmente ter acesso à sua região crítica*. Deste modo se garante a ausência de *livelock* — ψ_2 é, pois, uma propriedade de animação. No caso sequencial a propriedade de animação que é forçoso garantir é a *terminação* da execução do programa. A dupla garantia da correcção parcial e da terminação é referida como a *correcção total* do programa. No caso da programação concorrente, porém, a ontologia da animação é muito mais rica. Temos, por exemplo, propriedades relacionadas com o progresso dos sistemas (*e.g.* a acção reg ocorre eventualmente numa computação do sistema), propriedades recorrentes (*e.g.* a acção reg ocorre um número infinito de vezes na vida do sistema), propriedades de persistência (*e.g.* a partir de um determinado estado, a acção reg está continuamente disponível), ...

A especificação de um sistema concorrente inclui, regra geral, uma componente de animação e uma componente de segurança. Além disso qualquer propriedade sobre o comportamento de um sistema pode ser expressa como uma combinação de uma propriedade de cada um dos dois tipos referidos.

Nota 2 **[Segurança e Animação]**

A divisão das propriedades temporais em dois grandes grupos — *segurança* e *animação* — foi originalmente proposta por L. Lamport em [Lam77], onde surge também a sua ilustração em termos da ocorrência de 'coisas desejáveis' ou 'indesejáveis'. Se interpretarmos essas 'coisas' como factos verificáveis (observáveis) em tempo finito, podemos dizer as propriedades temporais se distinguem pela frequência de ocorrências de 'coisas desejáveis': sempre (segurança), pelo menos uma vez (progresso), um número infinito de vezes (recorrência), etc. Desde [Lam77] têm sido propostas diversas formulações rigorosas destas propriedades [AS85, Lam83] e outros esquemas de classificação. Em particular, existem diversas tentativas de organizar hierarquicamente as propriedades de animação (*e.g.* [MP90]).

Note-se, porém, que para especificar de requisitos de animação se torna necessária uma linguagem capaz de definir propriedades sobre comportamentos *infinitos*, enquanto para a segurança é suficiente lidar com comportamentos *finitos*, ou com prefixos finitos de comportamentos infinitos. As técnicas de prova usadas na verificação destes requisitos são, igualmente, distintas. Assim, para estabelecer uma propriedade de segurança é suficiente mostrar que esta se verifica no estado inicial do processo e que é preservada pelas transições. Consequentemente, um argumento indutivo sobre o comprimento da computação estabelece que a propriedade é sempre válida. A indução aqui é implícita: surge na justificação do método de prova e não na sua aplicação. Na área dos métodos formais de desenvolvimento de programas (por exemplo, em [Jon86]) é comum depararmos com provas deste tipo, por exemplo, na demonstração da preservação do invariante pelas operações de um modelo (ou não fosse este um típico requisito de segurança). Já a prova de propriedades de animação recorre frequentemente a um argumento indutivo explícito sobre uma observação do conjunto de estados que mede a "distância" até à verificação da propriedade desejada.

5 Interpretação das Propriedades Temporais

Como Interpretar as Propriedades Temporais?

Vimos já que o que fundamentalmente distingue as propriedades temporais das propriedades modais anteriormente consideradas é o facto de as primeiras serem formuladas sobre o conjunto das *computações* dos processos. Assim, parece razoável supor que as mesmas fórmulas modais, se interpretadas sobre sequências de transições elementares, poderiam ser 're-utilizadas' para exprimir propriedades temporais. Por exemplo, poderíamos tomar, como universo de interpretação, um sistema de transição cujos estados continuassem a ser os termos de $\mathbb{P}$, mas em que, na definição das transições, se tomasse $\xrightarrow{\omega} \subseteq \mathbb{P} \times \mathbb{P}$, com $\omega \in Act^*$ (em vez de $\xrightarrow{a} \subseteq \mathbb{P} \times \mathbb{P}$, com $a \in Act$). Poderíamos, ainda, ser um pouco mais ambiciosos e restringir esta relação às sequências que efectivamente representam computações válidas (de acordo com a definição acima).

Em qualquer caso, obtemos uma estrutura de interpretação muito mais rica que, por si só, sugere a introdução de novos operadores na lógica: operadores que realizem uma "quantificação" sobre as computações (da mesma forma que $[-]$ e $\langle - \rangle$ 'quantificam' sobre as transições elementares). As lógicas resultantes são classificadas como *lógicas temporais*.

Nota 3

Transições etiquetadas por sequências de acções, ou por determinados tipos destas, introduzem o elemento de *temporalidade* que pretendemos dominar. Podemos, mesmo, *codificar* estas relações em estruturas relacionais que modelam discretamente o tempo como um conjunto de instantes e uma ordem parcial entre eles.

De facto, originalmente, as lógicas temporais definem-se como lógicas modais sobre sistemas de transição não etiquetados onde as relações de transição são simples ordens parciais. Se olharmos para essas ordens como a relação familiar de precedência temporal e o conjunto de estados como um conjunto de instantes, as duas famílias básicas de operadores modais colapsam em dois únicos conectivos (uma vez que omitimos a etiquetagem por acções). Temos, então, $\langle\,\rangle$ e $[\,]$ que exprimem, respectivamente, *eventualidade* e *invariância*. Diferentes tipos de lógicas temporais são obtidas considerando diferentes maneiras de ordenar o tempo. A usual ordem total leva-nos aos modelos *lineares*. Se, porém, admitirmos não linearidade à esquerda ou à direita obtemos os modelos *ramificados* que permitem considerar distintos cenários futuros

ou diferentes histórias passadas, respectivamente. Recorde-se que uma relação de ordem $\mathcal{R}$ é linear à direita se $(a\mathcal{R}c \wedge b\mathcal{R}c) \Rightarrow (a = b \vee a\mathcal{R}b \vee b\mathcal{R}a)$. Dualmente, diz-se linear à esquerda se $(c\mathcal{R}a \wedge c\mathcal{R}b) \Rightarrow (a = b \vee a\mathcal{R}b \vee b\mathcal{R}a)$. Existem inúmeras variantes desta aproximação e das suas aplicações às ciências da computação (ver, por exemplo, [Sti92]).

A abordagem que vamos seguir neste curso é, contudo, um pouco distinta. Assim, em vez de interpretarmos as fórmulas modais directamente sobre uma estrutura relacional mais geral, vamos encarar as propriedades temporais como *limites* de propriedades modais sobre transições elementares. Esta visão das propriedades tem uma analogia, que queremos explorar, com a visão das computações de um processo como *limites* das transições elementares. Por exemplo, a possibilidade de um processo realizar numa determinada computação a transição etiquetada por a, pode ser vista como o limite das possibilidades de, em cada estado dessa computação ser possível a realização de a.

Porém, para desenvolvermos esta perspectiva, necessitamos de voltar a reflectir sobre as propriedades modais que atrás introduzimos. O que motiva

Um Segundo Olhar Sobre a Lógica Modal

A cada fórmula modal ϕ podemos associar o conjunto dos processos que a verificam, *i.e.*, $\{E \in \mathbb{P} \mid E \models \phi\}$, conjunto que representaremos por $\|\phi\|$. Notemos que ϕ e $\|\phi\|$ são duas maneiras distintas de nos referirmos à *mesma* propriedade, ainda que ϕ o faça de forma mais conveniente uma vez que, regra geral, $\|\phi\|$ não é um conjunto finito.

Uma qualquer propriedade ϕ divide, pois, $\mathbb{P}$ em dois subconjuntos disjuntos: $\|\phi\|$ e o seu complemento (*i.e.*, $\mathbb{P} - \|\phi\|$). Este último corresponderia a $\|\neg\phi\|$, se tivéssemos introduzido na lógica o conectivo usual de negação ($\neg$). Mesmo não o tendo feito, podemos sempre referirmo-nos ao *complemento* de ϕ, que representaremos por ϕ^{c}, como a fórmula correspondente à propriedade $\mathbb{P} - \|\phi\|$. De facto, esta noção de fórmula complementar está presente na lógica mesmo sem a introdução explícita da negação. Podemos defini-la estruturalmente. Assim,

$$\begin{aligned}
\mathsf{true}^{\mathsf{c}} &= \mathsf{false} \\
\mathsf{false}^{\mathsf{c}} &= \mathsf{true} \\
(\phi_1 \wedge \phi_2)^{\mathsf{c}} &= \phi_1^{\mathsf{c}} \vee \phi_2^{\mathsf{c}} \\
(\phi_1 \vee \phi_2)^{\mathsf{c}} &= \phi_1^{\mathsf{c}} \wedge \phi_2^{\mathsf{c}} \\
(\langle a \rangle \phi)^{\mathsf{c}} &= [a]\phi^{\mathsf{c}}
\end{aligned}$$

Esta correspondência entre fórmulas modais e conjuntos de processos fornece-nos uma definição alternativa da semântica da lógica que temos vindo a considerar. Para isso, porém, torna-se necessário definir directamente os conjuntos $\|\phi\|$, por indução sobre a estrutura das fórmulas, e não em termos da relação $\models$. É fácil concluir que

$$\begin{aligned}
\|\mathsf{true}\| &= \mathbb{P} \\
\|\mathsf{false}\| &= \emptyset \\
\|\phi_1 \wedge \phi_2\| &= \|\phi_1\| \cap \|\phi_2\| \\
\|\phi_1 \vee \phi_2\| &= \|\phi_1\| \cup \|\phi_2\|
\end{aligned}$$

As propriedades modais, vistas como conjuntos de processos, formam uma álgebra booleana (basta verificar que são fechadas para as operações de intersecção e complementação e que satisfazem os axiomas usuais, cuja verificação é sugerida como exercício).

Mas como vamos interpretar $[K]\phi$ ou $\langle K \rangle \phi$? Tal como a conjunção é interpretada pela intersecção e a disjunção pela reunião, também os conectivos modais dão origem a funções (unárias) entre

conjuntos de processos. Vamos designar essas funções por $\|[K]\|$ e $\|\langle K\rangle\|$, respectivamente. Por exemplo, $\|[K]\|$ faz corresponder a um conjunto X de processos um outro conjunto de processos cujas derivadas imediatas por acções em K constituem X. Formalmente,

$$\|[K]\| \;=\; \lambda_{X\subseteq\mathbb{P}} \,.\, \{F \in \mathbb{P} |\text{ se } F \xrightarrow{a} F' \text{ e } a \in K \;\text{ então }\; F' \in X\}$$

De forma similar se define $\|\langle K\rangle\|$,

$$\|\langle K\rangle\| \;=\; \lambda_{X\subseteq\mathbb{P}} \,.\, \{F \in \mathbb{P} |\; \exists_{F'\in X, a\in K} \,.\, F \xrightarrow{a} F'\}$$

Note-se que o efeito destas funções é operar, em cada processo em X, uma redução ao 'estado anterior' indexada pelas acções em K. Tal como acontece com as interpretações dos conectivos booleanos usuais, estas são igualmente funções *monótonas* relativamente à inclusão de conjuntos, *i.e.*,

$$\text{Se } \; X_1 \subseteq X_2 \;\; \text{ então } \;\; \|[K]\|(X_1) \subseteq \|[K]\|(X_2) \;\text{ e }\; \|\langle K\rangle\|(X_1) \subseteq \|\langle K\rangle\|(X_2)$$

Além disso não é difícil mostrar que as funções $\|[K]\|$ e $\|\langle K\rangle\|$ verificam as seguintes igualdades:

$$\begin{aligned}
\|[K]\|(\mathbb{P}) &= \mathbb{P} \\
\|\langle K\rangle\|(\emptyset) &= \emptyset \\
\|[K]\|(X_1 \cap X_2) &= \|[K]\|(X_1) \cap \|[K]\|(X_2) \\
\|\langle K\rangle\|(X_1 \cup X_2) &= \|\langle K\rangle\|(X_1) \cup \|\langle K\rangle\|(X_2)
\end{aligned}$$

Podemos, agora, completar a tabela anterior, de interpretação das fórmulas modais, definido,

$$\begin{aligned}
\|[K]\phi\| &= \|[K]\|(\|\phi\|) \\
\|\langle K\rangle\phi\| &= \|\langle K\rangle\|(\|\phi\|)
\end{aligned}$$

Nota 4 **[Álgebra Modal]**

Dissemos acima que a correspondência entre fórmulas e conjuntos de processos fornecia uma definição alternativa da semântica da lógica modal. Notemos, porém, que, em rigor, a definição estrutural de $\|\ \|$ para o caso dos conectivos modais só é válida assumindo que trabalhamos num universo de processos fechado para a relação de transição, o que é o caso de $\mathbb{P}$.

Feito este último comentário, o que é interessante é que obtivemos um novo *modelo* para a lógica de sabor mais algébrico, em alternativa ao que já tínhamos (*i.e.*, os sistemas de transição ou, genericamente, as estruturas relacionais). De facto, e seguindo um procedimento usual em Matemática, podemos olhar estes conjuntos de processos e operadores sobre eles (*e.g.* $\cap$, $\emptyset$, $\|[K]\|$, ...) como objectos e operadores abstractos que respeitam determinados axiomas equacionais. Podemos, assim, definir a noção de *álgebra modal*:

Definição. Uma álgebra modal é um tuplo $(P, 1, \sqcap, \neg, \{\|[x]\| \mid x \in Act\})$ onde $(P, 1, \sqcap, \neg)$ forma uma álgebra booleana e, para cada $x \in Act$, o operador $\|[x]\| : P \longrightarrow P$ verifica $\|[x]\|(1) = 1$ e $\|[x]\|(m \sqcap n) = \|[x]\|(m) \sqcap \|[x]\|(n)$.

Note-se que os restantes operadores — 0, $\sqcup$ e $\|\langle x\rangle\|$ — são definidos por abreviatura; em particular $\sqcup \stackrel{\text{abv}}{=} \neg \sqcap \neg$ e $\|\langle x\rangle\| \stackrel{\text{abv}}{=} \neg\|[x]\|\neg$.

Esta linha de raciocínio é ilustrativa de uma estragégia mais geral, e muito bem sucedida, em Lógica. Trata-se de traduzir conceitos e problemas da lógica em termos algébricos, aplicar, de seguida, os métodos da Álgebra Universal na sua resolução e traduzir a solução de novo para a linguagem da Lógica. Esta estratégia corresponde a uma disciplina própria — a *Lógica Algébrica* — que se iniciou com De Morgan, no final do século XIX, quando a procura de estruturas

correspondentes à lógica proposicional clássica conduziu à formulação das familiares álgebras booleanas. As álgebras modais, que acabamos de introduzir, desempenham um papel análogo relativamente à lógica modal [BRV95].

Historicamente, as álgebras modais constituiram um dos primeiros modelos para a lógica modal. De facto, a ponte entre ambas surgiu num artigo, hoje clássico, de Jónsson e Tarski em 1951 [JT51], cerca de uma década antes do desenvolvimento da semântica relacional que temos vindo a adoptar, por Kripke e outros [Kri63].

Tal como os sistemas de transição, as álgebras modais podem ser classificadas de acordo com certas condições adicionais que verificam. No caso dos sistemas de transição estas surgem como propriedades da família de relações que os definem (*e.g.* simetria, serialidade, etc.). Nas álgebras modais tais condições tomam usualmente a forma de um conjunto de equações — um exemplo típico é o caso das *álgebras dinâmicas* de V. Pratt [Pra92] que constituem a contrapartida algébrica de certas lógicas (modais) dos programas imperativos.

A correspondência entre estes dois modelos semânticos (relacional e algébrico) é muito útil quando queremos raciocinar sobre propriedades exprimíveis neste tipo de lógicas. Recordemos que esta nossa incursão algébrica se tornou necessária para tentar definir o significado das propriedades temporais dos processos. De facto, cada sistema de transição determina uma álgebra modal sobre a potência do respectivo conjunto de estados, como vimos, de forma particular, para o caso do sistema $(\mathbb{P}, \{\xrightarrow{x} \mid x \in Act\})$. Dualmente, as álgebras modais determinam sistemas de transição, segundo uma construção similar à do teorema da representação das álgebras booleanas [Sto36].

Propriedades Temporais Como Limites

Como dissemos, pretendemos ver as propriedades temporais como *limites* de propriedades modais sobre as transições elementares dos processos. Esta visão das propriedades tem uma analogia com a semântica operacional dos processos. De facto, as computações, que constituem o alvo deste tipo de propriedades, resultam do encadeamento, potencialmente não finito, dessas transições.

Raciocinemos, pois, por analogia. Quando queremos exprimir padrões comportamentais complexos, e também eles potencialmente não finitos, o mecanismo que utilizamos é a recursão sobre $\mathbb{P}$, introduzida através da igualdade definicional ($\triangleq$). Que sentido fará introduzir o mesmo mecanismo nas fórmulas modais? Certamente escrever

$$X \triangleq \langle a \rangle \mathsf{true}$$

não levanta qualquer dificuldade: X é apenas uma fórmula verificável exactamente pelos mesmos processos que verificam $\langle a \rangle \mathsf{true}$. Mas que diremos, por exemplo, de

$$X \triangleq \langle a \rangle X$$

Sabemos que uma maneira de interpretar as fórmulas modais é associa-las aos conjuntos de processos que as satisfazem. Em particular, a interpretação de $\langle a \rangle X$ é uma função em $\mathcal{P}(\mathbb{P})$ que ao conjunto de processos que satisfazem X, qualquer que ele seja, faz corresponder o conjunto $\|\langle a \rangle\|(X)$, do processos que através da realização de pelo menos uma transição por a conduzem a processos que satisfazem X. Assim, podemos interpretar a fórmula $X \triangleq \langle a \rangle X$ como uma *equação* em $\mathcal{P}(\mathbb{P})$ cujas soluções são os pontos fixos da função

$$\lambda_{X \subseteq \mathbb{P}} \,.\, \|\langle a \rangle\|(X)$$

Note-se que, por um ligeiro abuso de notação, identificamos a variável lógica X com uma variável sobre $\mathbb{P}$ (em rigor, deveríamos ter escrito $\|X\|$ na função acima).

A questão interessante é, agora, saber como caracterizar os pontos fixos da função em causa, *i.e.*, os subconjuntos $S \subseteq \mathbb{P}$ que verificam

$$S = \|\langle a \rangle\|(S)$$

A resolução deste tipo de equações não levanta problemas de maior. Contudo, o domínio em que as temos de resolver (*i.e.*, o espaço $\mathcal{P}(\mathbb{P})$) é consideravelmente menos rico em propriedades algébricas que, por exemplo, o corpo dos reais. Não disporemos, portanto, de algoritmos expeditos comparáveis.

Vimos atrás que a função $\lambda_{X \subseteq \mathbb{P}} \,.\, \|\langle a \rangle\|(X)$ era monótona relativamente à inclusão de conjuntos, propriedade que, de resto, é partilhada pela interpretação em $\mathcal{P}(\mathbb{P})$ dos outros conectivos modais. Isto basta para podermos, no cálculo das soluções que procuramos, recorrer a um resultado clássico sobre a existência de pontos fixos de funções monótonas em recticulados completos — o *teorema de Knaster-Tarski*, já recordado numa Lição anterior deste curso.

Nota 5

O teorema de Knaster-Tarski [Tar55] é um instrumento importante em semântica da computação. Recordemos a sua formulação.

Teorema 5.1 *Seja $(U, \sqsubseteq)$ um recticulado completo e $f : U \longrightarrow U$ uma função monótona,* i.e.*, tal que se $x \sqsubseteq y$, então $f(x) \sqsubseteq f(y)$. Então, o menor e o maior ponto fixo de f são dados, respectivamente, por*

$$m = \bigsqcap \{x \in U |\ f(x) \sqsubseteq x\}$$
$$M = \bigsqcup \{x \in U |\ x \sqsubseteq f(x)\}$$

Prova. Comecemos por provar que m é o menor ponto fixo de f. Seja $X = \{x \in U |\ f(x) \sqsubseteq x\}$ e tomemos um qualquer $x \in X$. Claramente, $m \sqsubseteq x$ e, sendo f monótona, $f(m) \sqsubseteq f(x)$. Por outro lado $f(x) \sqsubseteq x$, uma vez que $x \in X$. Logo podemos concluir que, para todo o $x \in X$, $f(m) \sqsubseteq x$. Isto significa que m é o *menor* pré-ponto fixo de f. Em particular $f(m) \sqsubseteq m$, o que, de novo pela monotonia de f, nos leva a $f(f(m)) \sqsubseteq f(m)$. Daqui concluimos que $f(m) \in X$ e, portanto, $m \sqsubseteq f(m)$. Mas então, $f(m) = m$ como queríamos. A segunda parte do teorema decorre desta: basta notar que se f é monótona em $(U, \sqsubseteq)$, então é ainda monótona no recticulado completo $(U, \sqsupseteq)$ formado pela ordem inversa. Se M é o menor ponto fixo de f em $(U, \sqsupseteq)$, será o *maior* ponto fixo da mesma função f em $(U, \sqsubseteq)$.

□

Instanciando este resultado para o recticulado $(\mathcal{P}(\mathbb{P}), \subseteq)$, concluimos que o *menor* dos pontos fixos de uma função monótona pode ser calculado como a intersecção generalizada (*i.e.*, o ínfimo no recticulado em causa) do conjunto dos respectivos pré-pontos fixos. Estes, por sua vez, são aqueles elementos S de $\mathcal{P}(\mathbb{P})$ tais que $f(S) \subseteq S$. Dito de outro modo, e para o caso que nos interessa, são aqueles conjuntos de processos que verificam

$$\|\langle a \rangle\|(S) \ \subseteq \ S$$

Ora, conhecendo nós a definição do operador $\|\langle a \rangle\|$, podemos re-escrever a condição acima sob a forma de um predicado sobre os processos que fazem parte desses pré-pontos fixos. Diremos, assim, que S é um pré-ponto fixo de $\lambda_{X \subseteq \mathbb{P}} \,.\, \|\langle a \rangle\|(X)$ sse

$$\text{(PRE)} \qquad \text{Se } \ E \in \mathbb{P} \ \wedge \ E' \in S \ \wedge \ E \xrightarrow{a} E' \ \text{ então } \ E \in S$$

Vejamos que conjuntos S satisfazem esta condição. Claramente, o conjunto $\{A \triangleq a.A\}$ verifica-a. Mas o mesmo se passa com o conjunto vazio (de facto se a premissa é falsa podemos concluir o que quisermos acerca do consequente da implicação). Chegamos, deste modo, à nossa primeira decepção: a fórmula, que parecia tão promissora, equivale a uma fórmula que é satisfeita pelo conjunto vazio de processos, *i.e.*, à fórmula false!

Felizmente, porém, esta não é a única solução da equação em mãos. Por verificação directa encontramos outra – o conjunto $\{A \triangleq a.A\}$. O teorema de Knaster-Tarski não nos diz como obter

uma solução arbitrária para a equação (foi um 'golpe de vista' que nos conduziu a $\{A \triangleq a.A\}$ ...), mas, contudo, fornece um método para calcular o outro ponto fixo extremo, *i.e.,* o *maior* ponto fixo. De acordo com o teorema, este é calculado pela reunião generalizada (*i.e.,* o supremo no recticulado em causa) do conjunto dos post-pontos fixos da função, *i.e.,* daqueles subconjuntos S de $\mathbb{P}$ tais que

$$S \subseteq \|\langle a \rangle\|(S)$$

De novo podemos traduzir esta condição sob a forma de um predicado (POST) de acordo com,

$$\text{(POST)} \quad \text{Se } E \in S \text{ então } E \xrightarrow{a} E' \text{ para algum } E' \in S$$

Qual será o maior subconjunto de $\mathbb{P}$ que verifica esta condição? A condição afirma que, se um processo pertence a S, então pode realizar a acção a e o processo que resulta dessa realização pertence igualmente a S. Podemos concluir que o maior subconjunto de $\mathbb{P}$ que verifica esta condição é o conjunto dos processos que apresentam pelo menos uma computação infinita da forma

$$E \xrightarrow{a} E_1 \xrightarrow{a} E_3 \xrightarrow{a} E_3 \xrightarrow{a} ...$$

Conseguimos, pois, exprimir à custa do maior ponto fixo da equação modal, a seguinte propriedade temporal:

$\phi =$ *Em qualquer ponto da sua evolução, o processo tem sempre a possibilidade de realizar a acção* a

O Caso Geral

Como este exemplo ilustrou, propriedades temporais podem ser formuladas como pontos fixos de determinadas equações. Equações que, como vimos, correspondem à interpretação em $\mathcal{P}(\mathbb{P})$ de fórmulas do tipo $X \triangleq \phi$, sendo X uma variável proposicional e ϕ uma fórmula modal onde X ocorre livre. A forma geral da interpretação em $\mathcal{P}(\mathbb{P})$ destas equações é

$$X = f(X)$$

onde $=$ é a igualdade entre conjuntos e f é uma função que transforma conjuntos de processos em conjuntos de processos de acordo com a 'especificação em ϕ', *i.e.,*

$$f = \lambda_{S \subseteq \mathbb{P}} . \{S/X\}\|\phi\|$$

onde $\|.\|$ é estendido a fórmulas com variáveis fazendo $\|X\| = X$ (*i.e.,*, a interpretação da variável lógica X é a variável em $\mathcal{P}(\mathbb{P})$ igualmente designada por X). Com auxílio do teorema de Knaster-Tarski podemos identificar a menor e a maior solução de uma equação deste tipo, ou seja o menor e o maior ponto fixo de f.

Nota 6

Como atrás se referiu, para garantir a existência de pontos fixos extremos é necessária a monotonia de f relativamente a $\subseteq$, o que se pode verificar para a interpretação em $\mathcal{P}(\mathbb{P})$ de todos os conectivos de $\mathbb{L}_M$. Aqui reside, porém, a razão pela qual se omitiu de $\mathbb{L}_M$ o conectivo usual de negação. De facto, na sua presença nem todas as equações modais exibem pontos fixos extremos. Repare-se que, por exemplo em $X = \neg X$, f não é monótona. É possível, contudo, introduzir $\neg$ desde que se imponha que em qualquer equação, toda a ocorrência livre da variável X esteja guardada por um número par de negações.

Por vezes as duas soluções são coincidentes (logo, únicas), mas no caso geral são distintas e existem, não raro, outras soluções intermédias. Como não possuímos um processo sistemático de cálculo destas soluções intermédias, seremos forçados a viver sem elas. No entanto vamos frequentemente recorrer às soluções extremas. Temos, pois, necessidade de introduzir alguma notação para a elas nos referirmos. Assim, vamos representar por

$$\mu X \,.\, \phi$$

o menor ponto fixo de f acima e por

$$\nu X \,.\, \phi$$

o maior ponto fixo. No exemplo em mãos, mostramos que $\mu X \,.\, \langle a \rangle \mathsf{true}$ é equivalente a false, enquanto $\nu X \,.\, \langle a \rangle \mathsf{true}$ garante a existência de pelo menos uma computação infinita de as. De modo análogo, a fórmula $\nu X \,.\, \langle \tau \rangle \mathsf{true}$ caracteriza os processos que se podem comprometer numa sequência infinita de acções não observáveis, ou seja, os processos *divergentes*.

Se o conectivo modal fôr parametrizado por um conjunto de acções $K \subseteq Act$, o processo que verifique $\nu X \,.\, \langle K \rangle \mathsf{true}$ pode realizar uma computação infinita composta por uma sequência qualquer de acções em K. Em particular, fazendo K coincidir com o próprio Act, a fórmula $\nu X \,.\, \langle - \rangle \mathsf{true}$ representa os processos que possuem pelo menos uma computação (qualquer) infinita, *i.e.*, que têm a possibilidade de exibir um comportamento perpétuo.

Vimos, também, que no cálculo destes pontos fixos é útil exprimir as condições $f(X) \subseteq X$ e $X \subseteq f(X)$ em termos dos elementos de X, *i.e.*,

$$\begin{array}{rl} \text{(PRE)} & \text{Se} \quad E \in \mathbb{P} \text{ e } E \in f(X) \quad \text{então} \quad E \in X \\ \text{(POST)} & \text{Se} \quad E \in X \quad \text{então} \quad E \in f(X) \end{array}$$

respectivamente.

Exemplo: $X \triangleq \phi \vee \langle a \rangle X$

Vamos investigar novas equações modais, aplicando esta estratégia para tentar caracterizar outras propriedades temporais. Comecemos, pois, por considerar a equação

$$X \triangleq \phi \vee \langle a \rangle X$$

A função implícita é

$$f \triangleq \lambda_{X \subseteq \mathbb{P}} \,.\, \|\phi\| \cup \|\langle a \rangle\|(X)$$

Vamos desenvolver a condição que define os seus pré-pontos fixos. Assim,

$$\begin{array}{rl} \text{(PRE)} & \text{Se} \quad E \in \mathbb{P} \text{ e } E \in f(X) \quad \text{então} \quad E \in X \\ \equiv & \text{Se} \quad E \in \mathbb{P} \text{ e } E \in (\|\phi\| \cup \|\langle a \rangle\|(X)) \quad \text{então} \quad E \in X \\ \equiv & \text{Se} \quad E \in \mathbb{P} \text{ e } E \in \{F |\, F \models \phi\} \cup \{F \in \mathbb{P} |\, \exists_{F' \in X} \,.\, F \xrightarrow{a} F'\} \quad \text{então} \quad E \in X \\ \equiv & \text{Se} \quad E \in \mathbb{P} \text{ e } E \models \phi \vee \exists_{E' \in X} \,.\, E \xrightarrow{a} E' \quad \text{então} \quad E \in X \end{array}$$

O menor conjunto de processos que verifica esta condição é formado por processos que possuem pelo menos uma computação ao longo da qual são capazes de realizar a acção a *até* a propriedade ϕ se verificar. Note-se, porém, e aqui está um ponto fundamental, que existe mesmo, nessa computação, um estado em que ϕ se verifica. E isto porque, ao tomar a intersecção destes conjuntos, ficamos com aqueles que atingem o estado em que ϕ se verifica num número *finito* de passos.

Vejamos, agora, qual será o maior dos pontos fixos de f. Para isso vamos 'desenrolar' a condição que caracteriza os respectivos post-pontos fixos.

$$\begin{array}{rlll} \text{(POST)} & \text{Se} \quad E \in X & \text{então} & E \in f(X) \\ \equiv & \text{Se} \quad E \in X & \text{então} & E \in (\|\phi\| \cup \|\langle a \rangle\|(X)) \\ \equiv & \text{Se} \quad E \in X & \text{então} & E \in \{F |\ F \models \phi\} \cup \{F \in X |\ \exists_{F' \in X} \,.\, F \xrightarrow{a} F'\} \\ \equiv & \text{Se} \quad E \in X & \text{então} & E \models \phi \vee \exists_{E' \in X} \,.\, E \xrightarrow{a} E' \end{array}$$

O maior ponto fixo é formado tomando a reunião de todos os post-pontos fixos. Isto significa, neste exemplo, que nele vamos incluir não apenas os processos que podem realizar a até à verificação de ϕ, mas também aqueles que mantêm a possibilidade de realizar a indefinidamente, sem nunca chegar a atingir um estado que verifique ϕ.

Em resumo, diremos que tanto $\mu X \,.\, \phi \vee \langle a \rangle X$ como $\nu X \,.\, \phi \vee \langle a \rangle X$ exprimem propriedades do tipo "qualquer coisa acontece *até* se atingir um estado em que outra propriedade se verifica". No primeiro caso, porém, este *até* é *estrito*, no sentido em que o estado em causa acaba por ser alcançado num número finito de passos. No segundo o estado que satifaz ϕ pode nunca ser atingido.

Dois casos particulares desta propriedade são particularmente relevantes na especificação de sistemas concorrentes. O primeiro consiste em substituir a por τ. Neste caso,

$$\mu X \,.\, \phi \vee \langle \tau \rangle X$$

indica que, após a eventual realização de alguma actividade interna, o processo que satisfaz este requisito alcança um estado que verifica ϕ. Note-se que este é precisamente o significado da fórmula $\langle\!\langle \rangle\!\rangle \phi$ discutido atrás e que em $\mathbb{L}_M$ não podíamos introduzir por abreviatura.

O outro caso com interesse surge quando se toma o conjunto Act como parâmetro do operador modal. De facto

$$\mu X \,.\, \phi \vee \langle - \rangle X$$

significa que existe uma computação em que se alcança um estado que verifica a propriedade ϕ após um número finito de transições. Repare-se que esta é uma propriedade de *animação fraca*. *Animação* porque afirma a ocorrência de qualquer coisa (no caso, da evolução para um estado que verifica ϕ) algures num determinado ponto da vida do processo, mas *fraca* porque essa garantia é dada, apenas, relativamente a uma computação. De facto, o processo poderia evoluir através de outra computação que nunca alcançasse o tal estado que valida ϕ.

Exemplo: $X \triangleq \phi \wedge \langle a \rangle X$

Consideremos, agora, a seguinte equação

$$X \triangleq \phi \wedge \langle a \rangle X$$

A condição (PRE) toma a forma

$$\text{(PRE)} \quad \text{Se} \quad E \in \mathbb{P} \text{ e } E \models \phi \wedge \exists_{E' \in X} \,.\, E \xrightarrow{a} E' \quad \text{então} \quad E \in X$$

Fazendo $X = \emptyset$ tornamos falso o antecedente da implicação, pelo que (PRE) resulta verdadeira. Daqui concluimos que $\emptyset$ é um pré-ponto fixo da função que resulta da interpretação em $\mathcal{P}(\mathbb{P})$ do lado direito da equação modal. Claramente a intersecção do conjunto dos pré-pontos fixos

será também ela vazia. Assim, $\mu X \,.\, \phi \wedge \langle a \rangle X$ é equivalente a false. A condição (POST), porém, re-escreve em

$$\text{(POST)} \quad \text{Se} \;\; E \in X \;\; \text{então} \;\; E \models \phi \wedge \exists_{E' \in X} \,.\, E \xrightarrow{a} E'$$

o que nos faz concluir que o maior ponto fixo, *i.e.*,

$$\nu X \,.\, \phi \wedge \langle a \rangle X$$

será o conjunto de processos que verificam ϕ e, simultaneamente, exibem uma computação infinita da forma

$$E \xrightarrow{a} E_1 \xrightarrow{a} E_2 \xrightarrow{a} E_3 \xrightarrow{a} E_4 \xrightarrow{a} \ldots$$

Dito de outro modo, um processo que verifique esta fórmula exibe a capacidade de realização perpétua da acção a ao longo da qual a propriedade ϕ permanece válida. Este é claramente um requisito de *segurança*.

Poderíamos ser um pouco mais exigentes e requerer ainda que a propriedade ϕ permanecesse válida ao longo de uma computação por a infinita ou *finita*, *i.e.*, conducente a um estado terminal do processo. Esta variante toma a forma

$$\nu X \,.\, \phi \wedge (\langle a \rangle X \vee [a]\mathsf{false})$$

No caso geral, para um conjunto $K \subseteq Act$, exigir que ϕ se verifique durante uma realização *maximal* de acções em K (*i.e.*, uma computação infinita ou finita desde que terminal), constitui uma propriedade de *segurança fraca* e exprime-se, como esperávamos, por

$$\nu X \,.\, \phi \wedge (\langle K \rangle X \vee [K]\mathsf{false})$$

ou, ainda, para $K = Act$, por

$$\nu X \,.\, \phi \wedge (\langle - \rangle X \vee [-]\mathsf{false})$$

Exemplo: $X \triangleq [-]X$

Vamos terminar com um exemplo um pouco surpreendente. Consideremos a equação

$$X \triangleq [-]X$$

O maior ponto fixo coincide com o próprio $\mathbb{P}$. De facto,

$$\begin{aligned} &\text{(POST)} \quad \text{Se} \;\; E \in X \;\; \text{então} \;\; E \in \|[-]\|(X) \\ &\quad \equiv \quad \text{Se} \;\; E \in X \;\; \text{então} \;\; (\text{se} \;\; E \xrightarrow{x} E' \;\text{e}\; x \in Act \;\; \text{então} \;\; E' \in X) \end{aligned}$$

Dito de outro modo, o resultado de uma transição qualquer em qualquer processo que pertença a um post-ponto fixo X, ainda pertence a X. É imediato que o maior subconjunto de $\mathbb{P}$ que verifica esta condição é ele próprio.

Mais curiosa é a análise dos pré-pontos fixos. Temos,

$$\text{(PRE)} \quad \text{Se} \;\; E \in \mathbb{P} \;\text{e}\; (\text{se} \;\; E \xrightarrow{x} E' \;\text{e}\; x \in Act \;\; \text{então} \;\; E' \in X) \;\; \text{então} \;\; E \in X$$

Claramente, o processo $\mathbf{0}$ pertence a qualquer conjunto X que verifique (PRE). Por outro lado, um processo que realize uma qualquer transição e derive em $\mathbf{0}$ pertence igualmente a X, uma

vez que $\mathbf{0} \in X$. Mas agora todo o processo que derive por uma qualquer transição noutro que, por sua vez, derive em $\mathbf{0}$, deve ser ainda considerado em X. E assim sucessivamente. O menor X que verifica a condição é então o conjunto de todos os processos finitos. Assim, a propriedade $\mu X \,.\, [-]X$ estabelece, para um processo que a verifique, a impossibilidade de este realizar uma computação infinita. Versões menos fortes deste requisito são obtidas substituindo Act por um seu subconjunto K. Nesse caso a propriedade exprime a ausência de computações infinitas com acções em K.

6 Taxonomia das Propriedades Temporais

Segurança e Animação

Acabamos de ver como que é possível exprimir propriedades de animação e segurança através, respectivamente, do menor e do maior ponto fixo de determinadas equações. Nos exemplos em causa, tratavam-se de propriedades de animação e segurança *fracas*, na medida em que formulavam uma eventualidade (no primeiro caso) e uma invariância (no segundo) sobre apenas, pelo menos, uma computação. Quer dizer, um processo que verifique qualquer dessas fórmulas pode evoluir realizando computações que, de facto, não validam os requisitos pretendidos.

Recordemos que uma propriedade de animação fraca pode ser expressa pela fórmula

$$\mu X \,.\, \phi \ \vee \ \langle - \rangle X$$

Intuitivamente isto significa que existe pelo menos uma computação em que um estado que verifique ϕ pode ser alcançado[2].

Qual será o complementar desta propriedade? Se um processo E não a verifica, então é porque todos os estados, alcançáveis ao longo de qualquer uma das suas computações, verificam a propriedade complementar de ϕ, *i.e.*, ϕ^{c}. Se, por mera cosmética, fizermos $\psi = \phi^{\mathsf{c}}$, tal requisito escreve-se como

$$\nu X \,.\, \psi \wedge [-]X$$

como se pode facilmente verificar. Propriedades deste tipo são ditas de *segurança forte* (repare-se que agora são considerdas *todas* as computações).

Temos, pois, que propriedades de *segurança forte* e de *animação fraca* são complementares. Atrás definimos, estruturalmente, a noção de fórmula complementar. Podemos, agora, alargar essa definição às formulas do tipo $\mu X \,.\, \phi$ e $\nu X \,.\, \phi$. De facto, se uma propriedade é o menor (maior) ponto fixo da equação modal

$$X \ \triangleq \ \phi$$

então a propriedade complementar é o maior (menor) ponto fixo da equação

$$X \ \triangleq \ \phi^{\mathsf{c}}$$

Formalmente,

$$(\mu X \,.\, \phi)^{\mathsf{c}} = \nu X \,.\, \phi^{\mathsf{c}}$$
$$(\nu X \,.\, \phi)^{\mathsf{c}} = \mu X \,.\, \phi^{\mathsf{c}}$$

[2]Como de costume, substituindo Act por um qualquer $a \in Act$ ou $K \subseteq Act$, obtemos versões relaxadas da propriedade na medida em que as computações consideradas não são quaisquer mas as geradas por a ou pelas acções em K. Geralmente iremos concentrar a nossa atenção no caso em que $K = Act$, mas tudo o que afirmarmos permanece válido para as versões relaxadas.

Em exemplo anteriores já deparamos com este tipo de situação. Mostramos, por exemplo, que, enquanto $\nu X \,.\, \langle\tau\rangle X$ exprimia *divergência*, a fórmula complementar garantia *convergência* (*i.e.*, comportamento não observável finito). De facto, $(\nu X \,.\, \langle\tau\rangle X)^{\mathsf{c}} = \mu X \,.\, (\langle\tau\rangle X)^{\mathsf{c}} = \mu X \,.\, [\tau] X$.

Vamos aplicar o mesmo raciocínio à fórmula que na acima usamos para exprimir a segurança fraca. Devemos chegar à formulação de uma propriedade que traduza um requisito de *animação* forte. De facto, o complementar de uma garantia de que existe uma computação ao longo da qual ϕ se verifica, é a possibilidade de, em todas as computações, ser possível alcançar um estado que verifique o complementar de ϕ. O complemento de

$$\nu X \,.\, \phi \wedge (\langle -\rangle X \vee [-]\mathsf{false})$$

será

$$\mu X \,.\, \psi \vee ([-] X \wedge \langle -\rangle \mathsf{true})$$

com $\psi = \phi^{\mathsf{c}}$. Esta última fórmula afirma, precisamente, que será alcançado (finitamente) um estado que verifique ψ. Trata-se de uma típica propriedade de animação.

Sumariando, temos que *segurança* e *animação* são propriedades complementares. Não é motivo de surpresa que a primeira recorra ao maior ponto fixo (dada a necessidade de postular uma garantia ao longo de toda uma vida do processo) e a segundo ao menor (porque as "coisas desejáveis" devem ser observáveis em tempo finito).

Acções e Estados

Há, no entanto, uma outra distinção em que é necessário pensar. Consideremos a propriedade de animação forte sobre a Rede de Táxis:

$\phi_0 \;=\;$ *A viatura acaba por regressar à Central*

Se *regressar à Central* fôr modelado pela acção reg, podemos escrever

$$\mu X \,.\, \langle reg\rangle \mathsf{true} \vee ([-] X \wedge \langle -\rangle \mathsf{true})$$

O que aqui se diz, de facto, é que todas as computações do processo viatura alcançam um estado em que é possível realizar a acção reg, facto expresso na subfórmula $\langle reg\rangle\mathsf{true}$. Mas isto é *diferente* de afirmar que reg efectivamente ocorre.

O que está em causa é a possibilidade de formularmos as propriedades de animação (forte) sobre os *estados* ou sobre a *ocorrência das acções*. No primeiro caso garantimos que um estado com certas características é alcançado. No segundo, gostaríamos de dizer que uma determinada acção ocorre. No exemplo em mãos diríamos, então,

$$\mu X \,.\, [-reg] X \wedge \langle -\rangle \mathsf{true}$$

A forma geral desta propriedade será, para garantir a realização de uma acção em $K \subseteq Act$,

$$\mu X \,.\, [-K] X \wedge \langle -\rangle \mathsf{true}$$

Por seu lado, a fórmula complementar desta é

$$\nu X \,.\, \langle -K\rangle X \vee [-]\mathsf{false}$$

que, naturalmente, representa uma propriedade de segurança fraca dirigida, não aos estados, como a que acima consideramos, mas à ocorrência das acções.

A relevância desta distinção *estados-acções* pode ser ilustrada pelo seguinte exemplo. Consideremos os processos

$$A_0 \triangleq a.\sum_{i\geq 0} A_i \quad \text{com} \;\; A_{i+1} \triangleq b.A_i$$
$$B_0 \triangleq a.\sum_{i\geq 0} B_i + \sum_{i\geq 0} B_i \quad \text{com} \;\; B_{i+1} \triangleq b.B_i$$

Não é difícil verificar que o processo $(A_k \mid A_k)$, para um $k > 0$, verifica a propriedade

$$\mu X \,.\, [-a]X \wedge \langle - \rangle \mathsf{true}$$

i.e., garantidamente a acaba por ocorrer. No entanto, falha a propriedade

$$\mu X \,.\, (\langle - \rangle \mathsf{true} \wedge [-a]\mathsf{false}) \;\vee\; (\langle - \rangle \mathsf{true} \wedge [-]X)$$

que afirma ser alcançável um estado em que a seja inevitável. De facto, os dois processos podem evoluir de forma que, em qualquer momento, um deles, pelo menos, ofereça a possibilidade de realizar b. Já o processo $(B_k \mid B_k)$, para um $k > 0$, falha as duas propriedades anteriores, mas verifica

$$\mu X \,.\, \langle a \rangle \mathsf{true} \;\vee\; (\langle - \rangle \mathsf{true} \wedge [-]X)$$

um requisito de animação que afirma ser sempre alcançável um estado em que a é possível (podendo, contudo, nunca ocorrer e, muito menos, surgir como inevitável).

Notemos, por fim, que esta distinção entre propriedades sobre os estados e propriedades sobre as acções não faz sentido quando discutimos propriedades de segurança forte. De facto, dizer que um conjunto K de acções nunca ocorre é equivalente a dizer que todos os estados em qualquer computação verificam $[K]\mathsf{false}$, *i.e.*,

$$\nu X \,.\, [K]\mathsf{false} \wedge [-]X$$

Verifica-se imediatamente, por complementação, que a mesma distinção é irrelevante no caso das propriedades de animação fraca.

Propriedades Condicionais

O poder expressivo da lógica modal com pontos fixos é surpreendentemente elevado. Muitas propriedades podem, de facto, ser formuladas combinando as formas básicas estudadas atrás. Por exemplo, suponhamos que queremos exprimir

ϕ_1 = *Após a recolha de um passageiro (icr), o táxi tem de o conduzir ao destino (fcr)*

A segunda parte de ϕ_1 é um requisito de animação forte que exprimimos na forma

$$\mu X \,.\, [-fcr]X \wedge \langle - \rangle \mathsf{true}$$

Porém, esta propriedade só tem de se verificar após o início de uma corrida. Se escrevermos apenas

$$[icr](\mu X \,.\, [-fcr]X \wedge \langle - \rangle \mathsf{true})$$

estamos a dizer muito pouco — apenas que todas as transições por icr a partir do estado inicial conduzem a estados que verificam o referido requisito de animação. O que, em verdade, queremos é especificar que isso mesmo acontece irrelevantemente do estado de evolução do processo Viatura. Ou seja, estamos perante um requisito de segurança forte. Mais exactamente, um requisito de animação *embebido* num requisito de segurança forte. A propriedade ϕ_1 traduz-se, pois, por

$$\nu Y \,.\, [icr](\mu X \,.\, [-fcr]X \wedge \langle - \rangle \mathsf{true}) \ \wedge \ [-]Y$$

Podemos designar este tipo de propriedades por *propriedades condicionais*. Neste caso trata-se de animação condicional, mas podemos igualmente pensar em segurança condicional. Um exemplo é a propriedade que garante que, sempre que uma dada propriedade ϕ se torna verdadeira, outra propriedade ψ não pode deixar de o ser (ou seja, nunca é possível ter-se ψ^{c}). Diríamos, pois,

$$\nu Y \,.\, (\phi^{\mathsf{c}} \vee (\phi \wedge \nu X \,.\, \psi \wedge [-]X)) \wedge [-]Y$$

Outra possibilidade consiste em fixar um requisito de segurança após a eventual ocorrência de uma acção. Por exemplo, e de novo no processo Viatura da rede de táxis, podemos especificar que

$\phi_2 \ = \ \textit{É possível alcançar estados a partir dos quais o regresso à garagem } (reg) \textit{ não pode ocorrer.}$

O requisito de segurança é aqui a impossibilidade de regresso que formalizamos como

$$\nu X \,.\, [reg]\mathsf{false} \wedge [-]X$$

Vamos, agora, combina-lo com a propriedade de animação (fraca) que traduz a possibilidade desta situação se verificar. Assim,

$$\mu Y \,.\, (\nu X \,.\, [reg]\mathsf{false} \wedge [-]X) \ \vee \ \langle - \rangle Y$$

Um outro tipo de propriedades condicionais que já encontramos são as que se formulam em termos de um *até*. Por exemplo,

$$\mu X \,.\, \psi \vee (\phi \wedge \langle - \rangle \mathsf{true} \wedge [-]X)$$

i.e., em todas as computações ϕ verifica-se até ψ se verificar. Este *até* é estrito, o que significa que ψ acaba por se verificar em todas as computações (note-se que esta situação é distinta da analisada atrás, onde consideramos um *até* estrito mas *fraco*). Esta restrição pode ser removida complementado o ponto fixo, *i.e.*,

$$\nu X \,.\, \psi \vee (\phi \wedge \langle - \rangle \mathsf{true} \wedge [-]X)$$

Se retirarmos a parcela $\langle - \rangle \mathsf{true}$, que força o progresso do sistema, obtemos

$$\nu X \,.\, \psi \vee (\phi \wedge [-]X)$$

Ou seja, em todas as computações (incluindo as finitas) ϕ é válido *a menos que* ψ se verifique. Aqui não faz sentido forçar a verificação de ψ.

Propriedades Cíclicas

O tipo mais comum de propriedades que afirmam a recorrência de determinados padrões comportamentais é a classe das propriedades cíclicas. Por exemplo,

ϕ_4 = *Em toda a computação de um determinado processo de gestão de memória, out ocorre em segundo, quarto, sexto, ..., $2n$-ésimo lugar.*

traduz-se por

$$\nu X \,.\, [-]([-out]\mathsf{false} \wedge [-]X)$$

Ou, mais realisticamente, poderíamos requerer que a cada in se seguisse um out, embora outras acções pudessem ocorrer entre estas. Diríamos, então,

$$\mu X \,.\, [out]\mathsf{false} \wedge [in](\mu Y \,.\, [in]\mathsf{false} \wedge [out]X \wedge [-out]Y) \wedge [-in]X$$

Note-se que o uso de menores pontos fixos estabelece a condição adicional de ser finita a quantidade de transições que podem ocorrer entre um in e um out.

Um pouco mais complexas, mas muito frequentes na especificação de sistemas concorrentes são propriedades do tipo

ϕ_5 = *Um estado em que in é possível é alcançado um número infinito de vezes*

ou

ϕ_6 = *A acção in pode ocorrer um número infinito de vezes*.

As fórmulas que exprimem estas condições são, respectivamente,

$$\nu X \,.\, \mu Y \,.\, (\langle in\rangle\mathsf{true} \vee \langle -\rangle Y) \;\wedge\; ([-]X \;\wedge\; \langle -\rangle\mathsf{true})$$

e

$$\nu X \,.\, \mu Y \,.\, [-in]Y \wedge [-]X \wedge \langle -\rangle\mathsf{true}$$

Trocando os operadores de ponto fixo no último exemplo, e substituindo $[-]X \wedge \langle -\rangle\mathsf{true}$ por $[in]X$ obtemos

$$\mu X \,.\, \nu Y \,.\, [-in]Y \wedge [in]X$$

que exprime a possibilidade de in ocorrer um número finito de vezes. Note-se que todas estas propriedades são *fortes*, na medida em que realizam uma quantificação implícita sobre todas as computações.

7 O μ-calculus Modal

Acabamos de ver que as propriedades temporais dos processos são exprimíveis numa extensão à lógica modal $\mathbb{L}_M$. Tal extensão consiste simplesmente em introduzir dois novos (mas poderosos) operadores que modelam os pontos fixos extremos da interpretação em $\mathcal{P}(\mathbb{P})$ de equações modais. Uma vantagem deste procedimento reside no facto de não termos tido necessidade de introduzir uma nova lógica, nem sequer um novo modelo para a lógica que vínhamos utilizando, surgindo esta como um fragmento bem caracterizado do caso geral.

O 'caso geral' é aqui precisamente a lógica $\mathbb{L}_M$ enriquecida com pontos fixos e um conjunto X de variáveis proposicionais. Esta constitui o *μ-calculus modal* [Koz83], que aqui vamos designar por $\mathbb{L}_{\mu M}$. Juntemos, pois, as peças num esforço de sistematização. Assim,

- As fórmulas bem formadas da lógica são-no de acordo com o seguinte BNF:

$$\phi \ ::= \ X \ \mid \ \phi_1 \wedge \phi_2 \ \mid \ \phi_1 \vee \phi_2 \ \mid \ \langle K \rangle \phi \ \mid \ [K]\phi \ \mid \ \mu X \,.\, \phi \ \mid \ \nu X \,.\, \phi$$

onde $K \subseteq \mathit{Act}$ e X é um conjunto de variáveis.

Note-se que as constantes true e false podem definir-se por abreviatura como

$$\mathsf{true} \stackrel{\text{abv}}{=} \nu X \,.\, X \quad \text{e} \quad \mathsf{false} \stackrel{\text{abv}}{=} \mu X \,.\, X$$

Da mesma forma, e como vimos atrás, as modalidades observáveis, antes introduzidas primitivamente em $\mathbb{L}_M$, surgem, agora, como simples abreviaturas de limites. De facto,

$$\begin{aligned} \langle\!\langle \rangle\!\rangle \phi &\stackrel{\text{abv}}{=} \mu X \,.\, \phi \vee \langle \tau \rangle X \\ [\![\,]\!] \phi &\stackrel{\text{abv}}{=} \nu X \,.\, \phi \wedge [\tau] X \\ [\![\downarrow]\!] \, \phi &\stackrel{\text{abv}}{=} \mu X \,.\, \phi \wedge [\tau] X \end{aligned}$$

Notemos que a definição de $[\![\,]\!]\phi$ é complementar da de $\langle\!\langle \rangle\!\rangle \phi$. Por outro lado, $[\![\downarrow]\!] \, \phi$ define-se como o *menor* ponto fixo da mesma fórmula usada na definição do operador $[\![\,]\!]\phi$. Esta escolha garante que nesse caso, ao contrário do que sucede com $[\![\,]\!]\phi$, é excluída a capacidade de realização infinita de acções não observáveis.

- Tal como a sintaxe, a semântica é apenas uma extensão da noção de modelo que já tínhamos para a lógica modal simples. Seguindo a abordagem atrás esboçada, vamos considerar que a relação de satisfação $\models$ é definida indirectamente em termos dos conjuntos $\|\phi\|$ (*i.e.*, os conjuntos de processos que validam a fórmula ϕ).

  Tendo introduzido variáveis na linguagem, é forçoso considerar modelos parametrizados por *valorações*, ou seja, funções que a cada variável atribuem o conjunto de processos em que esta toma o valor true. Formalmente,

$$V : X \longrightarrow \mathcal{P}(\mathbb{P})$$

  Obviamente, o cálculo da semântica das fórmulas é relativo à valoração das variáveis livres considerada. Assim, definimos,

$$\begin{aligned} \|X\|_V &= V(X) \\ \|\phi_1 \wedge \phi_2\|_V &= \|\phi_1\|_V \cap \|\phi_2\|_V \\ \|\phi_1 \vee \phi_2\|_V &= \|\phi_1\|_V \cup \|\phi_2\|_V \\ \|[K]\phi\|_V &= \|[K]\|(\|\phi\|_V) \\ \|\langle K \rangle \phi\|_V &= \|\langle K \rangle\|(\|\phi\|_V) \end{aligned}$$

  Até aqui nada de novo: apenas indexamos os conjuntos que interpretam as fórmulas pela valoração V e incluimos uma cláusula própria para dizer que a interpretação de uma variável é o conjunto que a valoração considerada lhe associa. Para o caso das fórmulas cujo conectivo principal é um operador de ponto fixo, teremos, como seria de esperar à luz da discussão anterior,

$$\begin{aligned} \|\nu X \,.\, \phi\|_V &= \bigcup \{S \in \mathbb{P} \mid S \subseteq \|\phi\|_{\{S/X\}V}\} \\ \|\mu X \,.\, \phi\|_V &= \bigcap \{S \in \mathbb{P} \mid \|\phi\|_{\{S/X\}V} \subseteq S\} \end{aligned}$$

  De facto, toda a fórmula ϕ determina uma função *monótona* entre conjuntos de processos, que a cada $S \subseteq \mathbb{P}$ faz corresponder o conjunto $\|\phi\|_{\{S/X\}V}$, relativa à valoração V e à variável

X. Assim, as interpretações das fórmulas $\nu X \,.\, \phi$ e $\mu X \,.\, \phi$ traduzem, respectivamente, e de acordo com o teorema de Knaster-Tarski, o maior e o menor ponto fixo dessa função.

Notemos, por fim, que o conjunto das fórmulas que não incluem variáveis livres é fechado para a complementação.

A noção de equivalência modal introduzida no início do capítulo permanece válida em $\mathbb{L}_{\mu M}$. Mas o que é muito mais interessante é o facto de os dois resultados sobre a relação entre ela e a bissimulação continuarem a verificar-se. O segundo resultado — *processos com imagem finita, modalmente equivalentes, são bissimilares* — decorre directamente uma vez que a única coisa que fizemos foi estender a lógica. Infelizmente, e ao contrário do que diria alguma (falsa) intuição, a restrição aos processos com imagem finita não pode ser relaxada. O primeiro resultado — *processos bissimilares são modalmente equivalentes* — está exaustivamente demonstrado em [Sti95]. À luz da discussão acima obtemos imediatamente o facto de estes resultados permanecerem válidos para o fragmento da lógica baseado nas modalidades observáveis.

Nota 7

Terminamos aqui o estudo das lógicas para processos. Foi nossa opção não dar uma panorâmica das inúmeras abordagens disponíveis, mas estudar uma delas com alguma profundidade.

A lógica estudada — o μ-calculus modal — foi inicialmente proposto em [Koz83], aplicando ao caso modal uma extensão que se tem revelado fecunda em outro tipo de lógicas: a inclusão de operadores de ponto fixo. Trata-se de uma lógica finitária e proposicional que apresenta duas vantagens importantes:

- é *decidível* e
- dotada de um grande poder expressivo (estritamente mais expressiva que lógicas como PDL e CTL* muito populares nas ciências da computação).

Entre os vários tópicos que não puderam ser abordados neste curso, encontra-se um que é de vital importância: o estudo dos métodos de prova para o μ-calculus modal. De facto, o cálculo de pontos fixos pelo teorema de Knaster-Tarski torna-se pouco expedito, e de difícil compreensão, para fórmulas em que surgem diversos conectivos μ e ν. O seu cálculo mais sistemático pode ser feito calculando os limites de cadeias de Kleene, uma vez que as funções que interpretam as fórmulas modais são, não apenas monótonas, mas também, em certas condições, *contínuas*. O processo, exemplificado com detalhe em [Sti95], é instrutivo (na medida em que o próprio cálculo das aproximações aumenta a nossa compreensão intuitiva do significado da fórmula), embora para processos não finitos, os índices das cadeias tenham que se estender para lá do ordinal ω.

Referências

[AS85] B. Alpern and F. B. Schneider. Defining liveness. *Information Processing Letters*, (21):181–185, 1985.

[BRV95] P. Blackburn, M. de Rijke, and I. Venema. The algebra of modal logic. CWI Tech. Rep., CWI, Amsterdam, 1995.

[Dij76] E. Dijkstra. *A Discipline of Programming*. Prentice-Hall, 1976.

[HM85] M. C. Hennessy and A. J. R. G. Milner. Algebraic laws for non-determinism and concurrency. *Journal of ACM*, 32(1):137–161, 1985.

[Hoa69] C. A. R. Hoare. An axiomatic basis for computer programming. *Comm. of ACM*, (12):576–580, 1969.

[Jon86] C. B. Jones. *Systematic Software Development Using* VDM. Series in Computer Science. Prentice-Hall International, 1986.

[JT51] B. Jónsson and A. Tarski. Boolean algebras with operators. *American Journal of Mathematics*, 78:891–939, 1951.

[Koz83] D. Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, (27):333–353, 1983.

[Kri63] S. Kripke. Semantical analysis of modal logic I: normal propositional calculi. *Zeitschrift fur mathematische Logik und Grundlagen der Mathematik*, (9):67–96, 1963.

[Lam77] L. Lamport. Proving the correctness of multiprocess programs. *IEEE Trans. Software Engineering*, (3):125–143, 1977.

[Lam83] L. Lamport. What good is temporal logic? In R. E. A. Mason, editor, *Proc. IFIP 9th World Congress*, pages 657–668. North Holland, 1983.

[Mil89] R. Milner. *Communication and Concurrency*. Series in Computer Science. Prentice-Hall International, 1989.

[MP90] Z. Manna and A. Pnueli. A hierarchy of temporal properties. In *Proc. 9th ACM Symp. on Principles of Distributed Computing*, pages 377–408, 1990.

[Pra92] V. Pratt. Origins of the calculus of binary relations. In *Proc. 7th Annual IEEE Symp. on Logic in Computer Science*, pages 248–254. Santa Cruz, CA, 1992.

[Sti92] C. Stirling. Modal and temporal logics. In Maibaum Abramsky, Gabbay, editor, *Handbook of Logic in Computer Science (vol. 2)*, pages 478–551. Oxford Science Publications, 1992.

[Sti95] C. Stirling. Modal and temporal logics for processes. *Springer Lect. Notes Comp. Sci. (715)*, pages 149–237, 1995.

[Sto36] M. Stone. The Theory of Representation for Boolean Algebras. *Trans. Amer. Math. Soc.*, (40):37–111, 1936.

[Tar55] A. Tarski. A lattice–theoretic fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–309, 1955.

A Exercícios

Exercício 1

Formule cada uma das seguintes propriedades na lógica modal $\mathbb{L}_M$. Repare que algumas são propositadamente vagas e podem, portanto, ser formalizadas de diversos modos.

1. A ocorrência de um a e um b é impossível.
2. A ocorrência de um a seguido de um b é impossível.
3. Apenas a ocorrência de a é possível.
4. Após a ocorrência de a é possível a ocorrência de b ou c.
5. Após a ocorrência de a ou b é possível a ocorrência de c.
6. Se a ocorrer, é possível a ocorrência de b ou c, mas não de ambos.
7. a não pode ocorrer antes de b.
8. Existe apenas uma transição etiquetada por a.

Exercício 2

Considere os processos seguintes e indique, para cada um deles, quais das propiedades acima são verificadas:

1. $E_1 \triangleq a.b.\mathbf{0}$
2. $E_2 \triangleq a.c.\mathbf{0}$
3. $E \triangleq E_1 + E_2$
4. $F \triangleq a.(b.\ \mathbf{0} + c.\mathbf{0})$
5. $G \triangleq E + F$

Exercício 3

Considere a seguinte especificação de um dispositivo de corte de madeira.

$$\begin{aligned} Start &\triangleq fw.Go + stop.\mathbf{0} \\ Go &\triangleq fw.bk.bk.Start + right.left.bk.Start \end{aligned}$$

Escreva em $\mathbb{L}_M$ as seguintes propriedades:

1. Após fw outro fw é imediatamente possível.
2. Após um fw seguido de um $right$, $left$ é possível e bk não.
3. A acção fw é, inicialmente, a única possível.
4. A terceira acção de $Start$ é distinta de fw.

Exercício 4

Seja $P \triangleq \mathsf{new}\ \{a\}\ (A \mid B) + a.\mathbf{0}$, onde $A \triangleq a.A$ e $B \triangleq \overline{a}.B$. Verifique ou refute as seguintes afirmações:

1. $P \models \langle a \rangle \mathsf{true} \vee \langle \overline{a} \rangle \mathsf{true}$
2. $P \models [\![a]\!][-]\mathsf{false}$

3. $P \models [\![\downarrow a]\!][-]\mathsf{false}$
4. $P \models [\![a \downarrow]\!][-]\mathsf{false}$
5. $P \models [\![\downarrow]\!]\mathsf{true}$

Exercício 5

A descrição que se segue modela um sistema de comunicação entre um processo S, que aceita pedidos para enviar mensagens, e um processo R, que as recebe. O meio de comunicação é simulado pelo processo M e está longe de ter um comportamento fiável.

$$\begin{aligned} S &\triangleq send.\overline{s}.S \\ R &\triangleq r.\overline{receive}.S \\ M &\triangleq s.\overline{r}.M + s.\overline{r}.\overline{r}.M + s.M \end{aligned}$$

$$P \triangleq \mathsf{new}\,\{e, r\}\ (S \mid M \mid R)$$

1. Explique porque razão o meio de comunicação não é fiável, identificando os problemas que podem surgir quando é solicitada a transmissão de uma mensagem.
2. Mostre que $\mathsf{new}\,\{e, r\}\ (\overline{s}.S \mid s.M \mid R) \xrightarrow{\tau} P$ é uma transição possível.
3. Verifique que $P \models [send]\langle\!\langle\overline{receive}\rangle\!\rangle\,\mathsf{true}$.
4. Para lidar com um meio de comunicação tão pouco fiável seria necessário especificar um protocolo de comunicação muito mais robusto. Tal protocolo deveria, em particular, garantir que *toda a mensagem enviada fosse recebida*. Escreva uma fórmula em $\mathbb{L}_M$ que registe formalmente essa propriedade.

Exercício 6

Seja E um processo. Uma fórmula ϕ diz-se a *fórmula característica* de E sse

$$\forall_{F \in \mathbb{P}}\ .\ F \models \phi \ \ sse \ \ F \sim E$$

Repare que, por definição, um processo verifica a *fórmula característica* de E sse fôr estritamente equivalente a E. Determine a *fórmula característica* do processo $x.\mathbf{0}$.

Exercício 7

Considere os processos $E \triangleq a.(b.\ \mathbf{0} + c.\mathbf{0})$ e $F \triangleq a.b.\ \mathbf{0} + a.c.\mathbf{0}$. Escreva uma fórmula ϕ em $\mathbb{L}_M$ que seja válida em E mas não em F.

Exercício 8

Considere os processos seguintes e escreva uma fórmula em $\mathbb{L}_M$ que seja válida para o proceso R e falsa para S.

$$E \triangleq b.c.\ \mathbf{0} + b.d.\mathbf{0} \tag{1}$$

$$F \triangleq E + b.(c.\ \mathbf{0} + d.\mathbf{0}) \tag{2}$$

$$R \triangleq a.E + a.F \tag{3}$$

$$S \triangleq a.F \tag{4}$$

Exercício 9

Defina em $\mathbb{L}_M$, por abreviatura, um operador (K), com $K \subseteq Act$, de forma que $E \models (K)\phi$ sse as acções em K forem as acções iniciais de E, todas elas conduzindo a estados que validam ϕ.

Exercício 10

Como sabe, a composição paralela não é, no caso geral, idempotente.

1. Fazendo $E \triangleq a.b.E$, formule uma propriedade em $\mathbb{L}_M$ que distinga E de $E \mid E$.
2. Em certos casos, porém, a idempotência verifica-se. Construa uma bissimulação que demonstre a equivalência $E \sim E \mid E$ quando E é da forma $E \triangleq \sum_{x \in K} x.E$, para um qualquer $K \subseteq L$. Será a conclusão anterior válida para todo o $K \subseteq Act$?

Exercício 11

Seja P um processo, a uma acção e ϕ uma fórmula em $\mathbb{L}_M$. Assumindo que $P \models \langle\!\langle a \rangle\!\rangle \phi$, indique justificando quais das asserções seguintes são verdadeiras.

1. $a.P \models \langle\!\langle a \rangle\!\rangle \phi$
2. $\tau.P \models \langle\!\langle a \rangle\!\rangle \phi$
3. $P[b/a] \models \langle\!\langle b \rangle\!\rangle \phi$, para qualquer acção b diferente de τ
4. $P \mid Q \models \langle\!\langle a \rangle\!\rangle \phi$, para qualquer processo Q
5. $P + Q \models \langle\!\langle a \rangle\!\rangle \phi$, para qualquer processo Q

Exercício 12

Determine

1. $\|[a][b]\langle c, d\rangle \mathsf{true}\|$
2. $\|\langle a\rangle\langle -\rangle \mathsf{true}\|$
3. $\|[a]\langle -\rangle \mathsf{true} \wedge [b][-]\mathsf{false}\|$
4. $\|[a]\langle -\rangle \mathsf{true} \vee [b][-]\mathsf{false}\|$

Exercício 13

O reconhecimento de linguagens regulares baseia-se, como sabe, na noção de *autómato*. Basicamente, este consiste num conjunto Σ de símbolos (o alfabeto), um conjunto finito S de estados, um estado onde se inicia o reconhecimento, uma relação de transição $\delta \subseteq S \times \Sigma \times S$ e um conjunto F de estados finais correspondentes às sequências de símbolos aceites pelo autómato. Por exemplo, o autómato especificado por

$$\begin{aligned} \Sigma &= \{a, b\} \quad S = \{1, 2\} \quad s_0 = 1 \quad F = \{2\} \\ \delta &= \{(1, a, 1), (1, b, 2), (2, a, 2), (2, b, 1)\} \end{aligned}$$

reconhece a linguagem formada por sequências de a e b, com um número ímpar de bs.

1. Codifique o autómato descrito acima numa expressão da linguagem $\mathbb{P}$ de processos que estudou.
2. Mostre ou refute (fornecendo contra-exemplos adequados) que

- Dois autómatos distintos que reconhecem a mesma linguagem regular são observacionalmente equivalentes.
- Dois autómatos que verificam um mesmo conjunto de propriedades expressas em $\mathbb{L}_M$, são estritamente equivalentes.
- Dois autómatos que verificam um mesmo conjunto de propriedades expressas em $\mathbb{L}_{\mu M}$, são estritamente equivalentes.

Exercício 14

Um sistema de segurança residencial é suposto fazer soar um alarme (acção modelada por alm) logo que detecta a presença de um intruso (situação modelada por int).

1. Será que a fórmula $[int](\langle alm\rangle \mathsf{true} \wedge [-alm]\mathsf{false})$ em $\mathbb{L}_M$ representa adequadamente essa propriedade comportamental?
2. Caso pense que não, represente-a correctamente (porventura em $\mathbb{L}_{\mu M}$)

Exercício 15

Suponha que num processo que especifica o comportamento de uma máquina de azar a acção $ganha(x)$ modela o facto do jogador ganhar uma quantia de x moedas. Alguém sugeriu que o processo deveria satisfazer uma das seguintes propriedades:

$$\begin{aligned}\phi_1 &= \nu X\,.\,(\mu Y\,.\,(\langle ganha(1000)\rangle \mathsf{true} \vee \langle -\rangle Y) \wedge [-]X)\\ \phi_2 &= \nu X\,.\,\mu Y\,.\,(\langle ganha(1000)\rangle X \vee \langle -\rangle Y\end{aligned}$$

Alguém, porém, argumentou que ϕ_1 e ϕ_2 eram equivalentes.

1. Explique o significado destas propriedades e discuta se serão ou não equivalentes.
2. Recorde a classificação das propriedades modais e temporais. Em que classe incluiría ϕ_1? E ϕ_2? Justifique.

Exercício 16

Formule em $\mathbb{L}_{\mu M}$ as seguintes propriedades sobre o comportamento de uma rede de táxis e classifique-as:

1. Toda a viatura acaba por regressar à Central.
2. Toda a viatura em serviço acaba por regressar à Central.
3. Para toda a viatura em serviço existe uma computação que a conduz a um estado em que pode regressar à Central.
4. Após a recolha de um passageiro, a corrida tem de terminar.
5. Uma vez alocada a um serviço, a viatura não pode regressar sem que este seja anulado ou completado com sucesso.
6. Qualquer viatura pode detectar um cliente na rua.

Exercício 17

Formule em $\mathbb{L}_{\mu M}$ a propriedade seguinte sobre o comportamento de uma máquina de venda automática de bebidas: *O depósito de uma ou duas fichas conduz à aquisição de um café ou um chá*.

Exercício 18

Uma propriedade importante em sistemas que controlam linhas de montagem indutriais é a garantia de que

ϕ = *sempre que uma situação de erro grave ocorre, o sistema pára*.

Note, porém, que, regra geral, não pára instantaneamente: por exemplo, pode ser necessário que, antes de parar, o sistema desligue certos circuitos, active indicadores luminosos num painel, etc.

1. Supondo que a acção $erro$ modela a ocorrência de um erro grave, codifique a propriedade ϕ em $\mathbb{L}_{\mu M}$.
2. Recorde a classificação das propriedades modais e temporais. Em que classe incluiría ϕ? Justifique.

Exercício 19

Quando um número limitado de recursos é partilhado por um número elevado de processos existe a possibilidade de alguns desses processos clientes terem de aguardar permissão para utilizar um recurso até que este seja libertado pelo processo que o está correntemente a usar. A alocação de recursos aos clientes é um problema típico, por exemplo, na concepção de sistemas operativos, onde é conhecido como o problema de *scheduling*.

Para que a alocação seja feita com sucesso é necessário saber quais os processos que aguardam permissão de acesso. Assim, o acesso não pode ser modelado por uma única acção atómica, mas antes por um par de acções: $request$ (para requerer o acesso) e $permission$ (para indicar que este foi concedido). Quando o recurso é libertado o processo realiza a acção $release$ para sinalizar que terminou a tarefa e que o recurso está de novo disponível.

Um algoritmo de alocação muito popular é similar ao usado em certas lojas em que cada utilizador tira um *ticket* com um número, emitido, por ordem estritamente crescente, por um dispositivo próprio. Os recursos partilhados são, neste exemplo, os funcionários da loja e, usualmente, existe um número fixo de funcionários que podem atender clientes simultaneamente. Sempre que um deles está livre chama o cliente com o número mais baixo e que ainda aguarda ser servido.

1. Suponha que se pretende modelar uma situação em que existem R recursos (similares) disponíveis, o que significa que R processos podem ser "atendidos" ao mesmo tempo. Complete a especificação do processo:

$$\begin{aligned} Sch &\triangleq Sch(0,0,0) \\ Sch(p,t,r) &\triangleq \text{if} \quad ... \quad \text{then } \overline{permission_t}\,. \quad ... \\ &\qquad \text{else } \overline{request(p)}\,. \quad ... + \quad release(x)\,. \quad ... \end{aligned}$$

   que realiza a alocação de processos a recursos de acordo com o protocolo acima descrito. O processo tem 3 parâmetros:

   - p é o número a atribuir ao próximo processo que solicite acesso (coincidente com o número de processos que já realizaram $request$);
   - t é o número do próximo processo a ser atendido (coincidente com o número de processos que já realizaram $permission$);
   - r é o número de processos que já foram servidos e libertaram o recurso utilizado realizando $release$.

2. Os 3 contadores — p, t e r — podem, eventualmente, acabar por atingir o mesmo valor $n > 0$. Diga quando é que tal situação pode ocorrer e modifique a definição de Sch de modo a que, nessas circunstâncias, os contadores sejam re-inicializados com o valor 0.
3. Um dos objectivos deste protocolo é garantir que nunca existe, simultaneamente, um processo à espera de permissão de acesso e um recurso livre. Especifique essa propriedade em $\mathbb{L}_{\mu M}$.
4. Outra propriedade que é desejável garantir é a ausência de *esperas infinitas*, *i.e.*,, todo o processo cliente que pede permissão acaba por conseguir aceder ao recurso. Especifique essa propriedade em $\mathbb{L}_{\mu M}$ e diga como poderia demonstrar que Sch a verifica.
