Uma ferramenta para auxílio no processo de verificação de especificações em RT-LOTOS

RICARDO FERREIRA MARTINS*
MURILO SILVA DE CAMARGO†
JEAN-MARIE FARINES*

*LCMI-EEL-CTC
†INE-CTC
Universidade Federal de Santa Catarina
Caixa Postal 476 - 88.040-900 - Florianópolis - S.C.
Tel.: +55 (48) 231-9202 - Fax: +55 (48) 231-9770
e-mail:{ricardo,farines}@lcmi.ufsc.br, murilo@inf.ufsc.br

Abstract

In this paper, we present a methodology and tools for verifying specifications of real-time systems, written in RT-LOTOS, a timed extension of the LOTOS specification language. From the model in RT-LOTOS, it is generated the related Timed Automata, over which the verification can be done, by using formulas written in the TCTL Real-Time Temporal Logic. The translator of RT-LOTOS specifications into Timed Automata that was built is described in this work. His integration with existing tools that allow the verification is also shown. Finally, case-studies allow to analyze the use of this method and the tools presented.

KEY WORDS: LOTOS, Real-Time Systems, Symbolic Model Checking.

1 Introdução

As Técnicas de Descrição Formal (TDFs) são cada vez mais aceitas e utilizadas para o desenvolvimento de sistemas complexos. Para Sistemas Distribuídos e Redes de Comunicação, em particular, as dificuldades provenientes do alto grau de paralelismo destas aplicações não encontram outra abordagem adequada para seu tratamento, a não ser o uso de técnicas formais e das metodologias e ferramentas que lhes são associadas. As linguagens de especificação padronizadas pela ISO (Estelle e LOTOS) e pelo ITU (SDL), nesta ultima década, surgiram destas necessidades e vem sendo aplicadas, tanto no campo acadêmico como industrial, com relativo sucesso. Ferramentas para análise e projeto de sistemas, a partir de especificações construídas numa dessas linguagens, já são produtos comerciais, e problemas cada vez mais complexos estão sendo tratados.

Entretanto, algumas das limitações destas técnicas se tornaram também mais visíveis, na medida do crescimento da complexidade e da diversidade das aplicações a serem desenvolvidas. Em particular, mostraram-se insuficientes para expressar as restrições temporais exigidas em diversas aplicações, tais como: sistemas multimídias, sistemas de automação industrial e sistemas tempo-real embutidos (carro, avião), entre outros.

A linguagem de especificação LOTOS, que se baseia nas álgebras de processos CCS e CSP, para a descrição comportamental e no tipo abstrato de dados ACT-One, tem despertado bastante

interesse pelo seu poder de expressão, que permite tratar a especificação em diferentes níveis de abstração; e pelo seu suporte teórico, que permite lhe associar técnicas e ferramentas de verificação baseadas em equivalências. Entretanto, sua expressividade apresenta limitações, por não ter nenhuma característica que permite a representação explícita do tempo. Nesses últimos anos, várias extensões temporais desta linguagem foram propostas [4, 6, 9, 17, 23], além de existir no momento um esforço de um comitê de padronização, na direção de um LOTOS Temporizado incluindo contribuições oriundas destas propostas. Na espera deste novo padrão, adotaremos neste trabalho a extensão proposta em [6], chamada de RT-LOTOS, sendo que as propostas e desenvolvimentos de metodologias e ferramentas serão futuramente adaptadas a este padrão.

Neste trabalho, apresenta-se uma abordagem para verificação de especificações de sistemas tempo-real, escritas na linguagem RT-LOTOS, e a ferramenta desenvolvida para auxiliar neste processo de verificação. Num primeiro tempo, os princípio básicos da verificação serão discutidos. A seguir, após uma rápida descrição das características de RT-LOTOS, serão apresentados o método de verificação de uma especificação, escrita nesta linguagem, e as ferramentas associadas. Finalmente, serão apresentados dois estudos de casos, nos quais utiliza-se um conjunto de ferramentas para validar o método de verificação. Na conclusão, este trabalho será confrontado a outros na área, mostrando suas contribuições, vantagens e limitações.

2 Os Princípios Básicos da Verificação

Neste trabalho, o método utilizado é o de verificação de modelos (*Model Checking* [1]), a partir do qual procura-se verificar se o modelo do comportamento do sistema satisfaz um conjunto de propriedades estabelecidas. Esta abordagem, dita dual, representa geralmente o comportamento do sistema a partir de um formalismo de descrição de seus estados, e as propriedades a partir de formulas de uma lógica modal.

Adota-se a álgebra de processos da linguagem LOTOS para especificar formalmente o comportamento do sistema, estendida com características temporais, que permitem a representação de sistemas tempo real. RT-LOTOS [6] é a linguagem de especificação utilizada, sendo que, em particular, ela possibilita a associação de intervalos temporais às ações do sistema. A não ocorrência destas ações dentro de seus intervalos caracteriza uma violação temporal, cujo o tratamento é feito a partir de um novo operador, chamado de *Preempção Temporal*.

Foi visto em [7] que, apesar de utilizar geralmente um Sistema de Transição Rotulado para a representação de uma especificação em álgebra de processos, no caso de RT-LOTOS, o uso deste não é adequado para a verificação de propriedades, pois implica na adição de um grande número de transições correspondentes a progressão do tempo, e consequentemente, num crescimento do número de estados do modelo do sistema. O uso de Autômatos Temporizados [2] para definir a semântica de RT-LOTOS diminui consideravelmente o espaço de estados a ser analisado, tornando viável o processo de verificação.

Por outro lado, as propriedades a serem verificadas serão descritas utilizando a Lógica Temporal Tempo Real TCTL [3], que quantifica os operadores de lógica temporal, introduzindo restrições temporais. Uma breve apresentação de TCTL será feita numa seção posterior.

A tradução das especificações RT-LOTOS para Autômatos Temporizados é obtida através do tradutor implementado pelos autores deste trabalho, sendo a verificação do autômato temporizado resultante feita através da ferramenta KRONOS [22], desenvolvida numa cooperação indústria-universidade pelo laboratório Verilog-Imag, na França. KRONOS implementa um algoritmo de verificação simbólica de modelos [15], onde os elementos de entrada são: o autômato temporizado, que descreve o sistema; e fórmulas TCTL, que estabelecem as propriedades a serem verificadas. Assim, a ferramenta KRONOS retorna como resultado as condições em que o autômato temporizado satisfaz a fórmula. A figura 1 ilustra o processo de verificação.

Figura 1: O Processo de Verificação

3 A Linguagem de Especificação RT-LOTOS

Nesta seção, é apresentado um resumo das principais características da sintaxe e da semântica de RT-LOTOS, cuja a descrição completa pode ser encontrada em [6].

RT-LOTOS é uma extensão temporal da linguagem LOTOS, que permite a representação e o tratamento de restrições temporais. Nesta linguagem, intervalos de tempo podem ser associados às ações. Estes intervalos, da forma $[t_{min}, t_{max}]$, determinam em que instante as diferentes ações podem ser oferecidas ao seu ambiente. Se nenhum intervalo é associado explicitamente à uma dada ação observável, faz-se a hipótese de que o intervalo $[0, \infty]$ está associado a esta ação, o que significa que as ações observáveis não temporizadas são consideradas como não urgentes.

A associação de um tempo máximo, t_{max}, a uma ação não implica que se deseja forçar a urgência da ação temporizada, pois procura-se manter o paradigma das álgebras de processos, no qual a aceitação de uma ação observável depende de seu ambiente. Em RT-LOTOS, se uma ação a não pôde se realizar durante o intervalo especificado $[t_{min}, t_{max}]$, então ela não poderá o ser fora deste intervalo. Ações especiais a^* notificam as eventuais violações temporais, onde o interesse de poder dispor destas ações especiais reside na facilidade de representar os tratamentos de exceção, a serem realizados quando da ocorrência destas, a partir de um novo operador de preempção temporal.

De acordo com a descrição formal da semântica, RT-LOTOS permite a ocultação de ações observáveis temporizadas. Para manter a coerência do modelo, restrições temporais são associadas também à ação interna i, sem aumento da complexidade da semântica. A ação interna i, sem temporização explícita, é considerada como sendo temporizada por $[0, 0]$, enquanto que para $[t_{min}, t_{max}]i$, a ação interna i deve, necessariamente, realizar-se no intervalo $[t_{min}, t_{max}]$.

Os termos de RT-LOTOS são gerados pela sintaxe apresentada a seguir, que é uma extensão direta da sintaxe de LOTOS:

E	$::=$ *stop*	(* inação *)
	exit	(* terminação com sucesso *)
	$[t_{min}, t_{max}]a; E$	(* prefixação *)
	$[t_{min}, t_{max}]i; E$	(* prefixação *)
	$E[]E'$	(* escolha *)
	$E\|[L]\|E'$	(* composição paralela *)
	$hideLinE$	(* ocultação *)
	$E \gg F$	(* composição seqüencial *)
	$E[> F$	(* preempção *)
	$E < L\{a_1 : Q_1, ..., a_n : Q_n\}$	(* preempção temporal *)
	$P[a_1, ..., a_n]$	(* instanciação de processos *)

As ações de RT-LOTOS são atômicas e instantâneas, e compreendem:

- as ações clássicas de LOTOS, como as *ações observáveis*, a, pertencentes ao conjunto Act, a ação interna i e a ação de término com sucesso δ, sendo que define-se: $Act^i = Act \cup \{i\}$, $Act^\delta = Act \cup \{\delta\}$ e $Act^{i,\delta} = Act \cup \{i\} \cup \{\delta\}$;
- as ações específicas de RT-LOTOS, que são violações temporais, a^*, que pertencem ao conjunto Act^*, sendo que existe uma bijeção entre Act e Act^*.

O domínio de tempo D^w, para a temporização das ações de Act^i, pode ser esparso ou denso, mas deve ser enumerável. A semântica operacional é apresentada em [6], no estilo SOS (Structured Operational Semantics) de Plotkin, e inclui:

- um conjunto de regras de inferência para as ações clássicas de RT-LOTOS;
- um conjunto de regras de inferência para as violações temporais;
- um conjunto de regras de inferência para a progressão do tempo.

4 A Tradução de RT-LOTOS em Autômatos Temporizados

Nesta seção, apresenta-se e discute-se a metodologia utilizada para realizar o mapeamento de especificações escritas em RT-LOTOS para Autômatos Temporizados, descrevendo ainda os princípios e as etapas do tradutor automático, desenvolvido para desempenhar este mapeamento. Apesar do uso de Autômatos Temporizados para representar os modelos dos sistemas, descritos por uma álgebra de processos temporizada, ter sido apresentado em vários trabalhos existentes na literatura [7, 11, 25], julgamos importante esta apresentação e discussão do processo de tradução, por se tratar de uma extensão específica da linguagem LOTOS e por levar a implementação de uma ferramenta associada.

4.1 Autômatos Temporizados

Um autômato temporizado [1, 2] é um autômato estendido, com um conjunto finito de relógios, definidos num domínio de tempo D arbitrário. O valor de cada relógio, num dado instante, é obtido a partir do tempo decorrido desde a sua ultima reinicialização, sendo que esta ocorre como resultado do disparo de uma transição. Para cada transição, podem ser associadas condições sobre os relógios, e o disparo desta depende não somente da sua habilitação, mas também da verificação das condições sobre estes relógios.

Das definições formais disponíveis na literatura [7, 11, 25], escolhemos as definições apresentadas em [25], por corresponder ao formalismo que permitiu construir a ferramenta KRONOS, que será utilizada neste trabalho.

Definição 1 *Seja A um vocabulário de ações, no qual "a" denota um elemento de A. Seja também $\Psi(C)$ um conjunto de restrições temporais sobre um conjunto de relógios, C. Um Autômato Temporizado é uma 5-tupla $\langle S, C, L, s_I, \delta \rangle$, onde:*

- *S é o conjunto finito dos vértices, chamados localizações*
- *C é o conjunto finito dos relógios*
- *L é o conjunto finito dos arcos, chamados transições*
- *$s_I \in S$ é o vértice (ou localização) inicial*
- *$\delta : S \rightarrow \Psi(C)$ é a função que associa a cada vértice uma condição de atividade (condição de permanência no vértice)*

sendo que cada arco $e \in L$ corresponde a $\langle s, a, \psi, C', s' \rangle$, onde s, $s' \in S$ são os vértices de entrada e de saída do arco, $a \in A$ é a etiqueta para a ação ocorrida, ψ representa as condições sobre os relógios, e $C' \subseteq C$ é o conjunto dos relógios a serem inicializados.

No vértice inicial de um autômato temporizado, todos os relógios são inicializados. Se o sistema está em um determinado vértice s, então uma transição e poderá ser disparado somente se os valores dos relógios satisfazem a condição ψ, resultando na mudança para outro vértice s', e na inicialização dos relógios pertencentes ao conjunto C'. As condições dos relógios, associadas às transições, e as condições de atividade, associadas aos vértices, são combinações do tipo $x \sim c$, onde $x \in C$, $c \in \aleph$ e $\sim$ é uma relação binária pertencente ao conjunto $\{<, \leq, >, \geq, =\}$.

A determinação dos relógios a serem inicializados pelos arcos depende de uma relação entre o conjunto de vértices e o conjunto de relógios do autômato. Objetivando facilitar tal determinação, é apresentada a seguir uma extensão de autômatos temporizados [25], utilizada neste trabalho.

Definição 2 *Um Autômato Temporizado Estendido é representado pela 6-tupla* $\langle S, C, L, SI, \delta, F \rangle$, *onde* $\langle S, C, L, SI, \delta \rangle$ *é um Autômato Temporizado, sendo* $F \subseteq S \times C$ *um conjunto de extensões, associando os relógios pertencentes ao conjunto* C *com os vértices do conjunto* S. *Para todo* $s \in S$, *é definido:*

$$F(s) = \{x \in C \mid (s, x) \in F\}$$

Definição 3 *Seja um autômato temporizado estendido* $T = \langle S, C, L, SI, \delta, F \rangle$, *define-se:*

$$Reach(T) = \langle \bar{S}, C, \bar{L}, SI, \bar{\delta}, \bar{F} \rangle$$

onde:
$SI \in \bar{S}$
Se $s \in \bar{S}$ *e* $\langle s, a, \psi, C', s' \rangle \in L$, *então* $s' \in \bar{S} (\bar{S} \subseteq S)$
$\bar{L} = L \cap (\bar{S} \times A \times \psi(C) \times 2^C \times \bar{S})$
$\bar{\delta} = \delta \cap (\bar{S} \times \psi)$
$\bar{F} = F \cap (\bar{S} \times C)$

$Reach(T)$ é um autômato temporizado estendido, onde os vértices são unicamente os vértices acessíveis a partir do vértice inicial SI do autômato T, ou seja, eliminando os vértices inacessíveis e as transições que nunca serão disparadas.

4.2 A Tradução de RT-LOTOS para Autômatos Temporizados

É mostrado a seguir como se constrói um autômato temporizado estendido, denotado por $T[P]$, a partir de expressões de comportamento P, escritas em RT-LOTOS. Esta construção é efetuada através do tratamento de cada subexpressão contida em P, e é conhecida como construção guiada pela sintaxe. Serão apresentadas neste artigo apenas as translações de alguns operadores, sendo que todas elas podendo ser encontradas detalhadamente em [19].

Porém, existem certas limitações quanto a este mapeamento, discutidas em [7, 12], estabelecendo que será possível construir um autômato temporizado, a partir de uma expressão P, somente se esta especificação levar a um sistema de transições finito, gerando assim um autômato com um número finito de vértices. Em [12], observou-se que especificações LOTOS que apresentam uma recursividade contida em um dos operandos do operador de composição paralela, ou contida no operando esquerdo dos operadores de composição seqüencial e preempção, podem gerar um sistema de transições infinito, classificando estas recursividades como proibidas, do ponto de vista da geração do autômato.

Além do conjunto de ações de RT-LOTOS, definidos anteriormente, é necessário utilizar na tradução uma ação especial ε, para representar o instante a partir do qual uma ação a ser sincronizada esteja disponível em um dos processos, envolvidos numa composição paralela. Esta ação é normalmente encontrada nos métodos de construções composicionais [12].

Serão considerados a seguir dois processos genéricos, P e Q, para a representação dos operadores, onde os respectivos autômatos temporizados estendidos são:

$$T[P] \equiv \langle S_1, C_1, L_1, s_{I_1}, \delta_1, F_1 \rangle \text{ e } T[Q] \equiv \langle S_2, C_2, L_2, s_{I_2}, \delta_2, F_2 \rangle$$

Inação: O autômato para o processo *stop* consiste apenas de um vértice. Ele não possui nenhuma transição, e a condição de atividade deste vértice é sempre verdadeira (*true*), possibilitando uma permanência indefinida neste vértice.

$$T[stop] = \langle \{s\}, 0, 0, s, \{(s, true)\}, 0 \rangle$$

Terminação com sucesso: O autômato para o processo *exit* consiste de dois vértices e uma transição etiquetada com a ação não urgente δ, caracterizada pela condição de atividade do vértice $so(true)$, e pela condição da transição (*true*). Este fato dispensa a atribuição de um novo relógio.

$$T[exit] = \langle \{so, s1\}, 0, e, so, \{(so, true), (s1, true)\}, 0 \rangle$$

onde: $e = \langle so, \delta, true, 0, s1 \rangle$

Prefixação: A prefixação será representada de duas formas: $[t_1, t_2]i; P$, correspondente a prefixação de um processo pela ação interna i, e $[t_1, t_2]a; P$, correspondente a prefixação de um processo por uma ação observável a. Esta diferenciação se deve ao fato de que, no instante $t = t_2$, a ação i se torna urgente e incontrolável, enquanto que a ação a não ocorrerá mais, gerando uma violação temporal.

A representação da urgência de uma ação se faz através da condição de atividade, associada ao vértice de entrada do arco, ou seja, quando o relógio t associado ao vértice, atingir um valor $t_{lim} = t_2$, e se este vértice possuir como condição de atividade $t \leq t_{lim}$, a transição que possuir suas condições satisfeitas será, obrigatoriamente, disparada.

i. Para $[t_1, t_2]i; P$, com $t_2 \geq t_1$, tem-se:

$$T[[t_1, t_2]i; P] = \langle S_1 \cup \{so\}, C_1 \cup \{t\}, L_1 \cup \{e\}, so, \delta_1 \cup \delta', F_1 \cup F' \rangle$$

onde:
$so \notin S_1, t \notin C_1$ e
$e = \langle so, i, t_1 \leq t \leq t_2, F_1(s_{I_1}) \cap C_1, s_{I_1} \rangle$
$\delta' = \{(so, t \leq t_2)\}$
$F' = \{(so, t)\}$

Assim, quando o relógio t, atingir o valor t_2, no vértice so, a ação i será disparada de maneira urgente e incontrolável, caso não tenha ocorrido durante o intervalo $[t_1, t_2]$. A expressão $F_1(s_{I_1}) \cap C_1$ estabelece o conjunto de relógios a serem inicializados pelo arco, que correspondem aos relógios relacionados ao vértice inicial do processo P.

ii. Para $[t_1, t_2]a; P$, com $t_2 \geq t_1$ e $a \in Act$, tem-se:

$$T[[t_1, t_2]a; P] = \langle S_1 \cup \{so, s1\}, C_1 \cup \{t\}, L_1 \cup L', so, \delta_1 \cup \delta', F_1 \cup F' \rangle$$

onde:
$\{so, s1\} \notin S_1, t \notin C_1$ e:
$L' = \{\langle so, a, t_1 \leq t \leq t_2, F_1(s_{I_1}) \cap C_1, s_{I_1} \rangle, \langle so, a^*, t = t_2, 0, s1 \rangle\}$
$\delta' = \{(so, t \leq t_2), (s1, true)\}$
$F' = \{(so, t)\}$

Neste caso, a partir do momento em que se tem $t = t_2$, a^* torna-se urgente e incontrolável. Para uma prefixação do tipo $a; P$, ou seja, uma ação que pode acontecer em qualquer instante, tem-se:

$$T[a; P] = \langle S_1 \cup \{so\}, C_1, L_1 \cup \{e\}, so, \delta_1 \cup \delta', F_1 \rangle$$

onde: $so \notin S_1, e = \langle so, a, true, F_1(si_1) \cap C_1, si_1 \rangle$ e $\delta' = \{(so, true)\}$

Escolha: O método utilizado para obter o autômato temporizado de um comportamento $P[]Q$ consiste em construí-lo a partir dos autômatos P e Q, onde seus vértices são os pares $(s1, s2)$, com $s1 \in S_1$ e $s2 \in S_2$. Quando ocorrer uma ação $a \in Act^{i,\delta}$ de qualquer um dos processos, a partir de um vértice $(s1, s2) \in S_1 \times S_2$, uma transição estará direcionada para a continuação daquele processo, resolvendo a escolha. Já a violação temporal e a ação especial ε não resolvem a escolha, ou seja, tem-se um vértice $(s1', s2)$ ou $(s1, s2')$ como vértice de saída para tais transições. A representação de $P[]Q$ é dada por:

$$T[P[]Q] = Reach(\langle S, C, L, si, \delta, F \rangle)$$

onde: $S = S_1 \cup S_2 \cup (S_1 \times S_2), C = C_1 \cup C_2$, $si = (si_1, si_2)$

$$\begin{aligned} L = & L_1 \cup L_2 \\ & \cup \{\langle (s1, s2), a, \psi, C', s \rangle \mid \langle s1, a, \psi, C', s \rangle \in L_1 \wedge a \in Act^{i,\delta}\} \\ & \cup \{\langle (s1, s2), a, \psi, C', s \rangle \mid \langle s2, a, \psi, C', s \rangle \in L_2 \wedge a \in Act^{i,\delta}\} \\ & \cup \{\langle (s1, s2), a^*, \psi, 0, (s1', s2) \rangle \mid \langle s1, a^*, \psi, C', s1' \rangle \in L_1 \wedge a^* \in Act^{\varepsilon,*}\} \\ & \cup \{\langle (s1, s2), a^*, \psi, 0, (s1, s2') \rangle \mid \langle s2, a^*, \psi, C', s2' \rangle \in L_2 \wedge a^* \in Act^{\varepsilon,*}\} \end{aligned}$$

$F = F_1 \cup F_2 \cup \{((s1, s2), f) \mid (s1, f) \in F_1 \vee (s2, f) \in F_2\}, si \in S_i, i = 1, 2$
e para cada $s \in S$:

$$\delta[s] : \begin{cases} \delta_i[s] & \text{se } s \in S_i, i = 1, 2 \\ \delta_1[s1] \wedge \delta_2[s2] & \text{se } s = (s1, s2) \in S_1 \times S_2 \end{cases}$$

Ressalta-se que a ação especial ε será adicionada a um determinado autômato somente através do operador de composição paralela. No entanto, os operadores binários devem considerar a possibilidade de seus operandos possuírem tal ação.

Composição Paralela: O operador de composição paralela impõe urgência às ações a serem sincronizadas entre dois processos somente quando ambos estão prontos para isto. A ação especial ε representa a mudança do vértice de uma ação que deve ser sincronizada, mas esta sem a condição dos relógios satisfeita, para o vértice onde esta condição se torna satisfeita.

Para transições onde o limite mínimo da condição dos relógios é igual a zero, não é necessário utilizar a ação ε, pois ela já estará habilitada a partir do vértice de entrada do arco. Este é o caso da ação δ, gerada pelo operador de terminação com sucesso. O autômato temporizado estendido para o operador $P \mid [J] \mid Q$ é dado então por:

$$T[P \mid [J] \mid Q] = Reach(\langle S, C, L, si, \delta, F \rangle)$$

onde: $S = (S_1 \times S_2) \cup (S_1 \times S_q) \cup (S_2 \times S_p), C = C_1 \cup C_2$, $si = (si_1, si_2)$

$$\begin{aligned} L = & \{\langle (s1, s2), a, \psi, C', (s1', s2) \rangle \mid \langle s1, a, \psi, C', s1' \rangle \in L_1 \wedge a \in Act^i \wedge a \notin J\} \\ & \cup \{\langle (s1, s2), a, \psi, C', (s1, s2') \rangle \mid \langle s2, a, \psi, C', s2' \rangle \in L_2 \wedge a \in Act_i \wedge a \notin J\} \\ & \cup \{\langle (s1, s2), \varepsilon, Min(\psi_1), 0, (sp, s2) \rangle \mid \langle s1, a, \psi, C', s1' \rangle \in L_1 \wedge a \in J \wedge Min(\psi_1) \neq 0\} \\ & \cup \{\langle (s1, s2), \varepsilon, Min(\psi_2), 0, (s1, sq) \rangle \mid \langle s2, a, \psi, C', s2' \rangle \in L_2 \wedge a \in J \wedge Min(\psi_2) \neq 0\} \\ & \cup \{\langle (sp, sq), a, 0, C_1' \cup C_2', (s1', s2') \rangle \mid \langle si, a, \psi_i, C_i', si' \rangle \in L_i \wedge a \in J \cup \{\delta\}, i = 1, 2\} \\ & \cup \{\langle (s1, s2), a^*, \psi_1, C_1', (s1', s2) \rangle \mid \langle s1, a^*, \psi_1, C_1', s1' \rangle \in L_1\} \\ & \cup \{\langle (s1, s2), a^*, \psi_2, C_2', (s1, s2') \rangle \mid \rangle s2, a^*, \psi_2, C_2', s2' \rangle \in L_2\} \end{aligned}$$

e para $s = (s1, s2) \in S$:
$F = \{((s1, s2), f) \mid (s1, f) \in F_1 \vee (s2, f) \in F_2\}$ e

$$\delta[s] : \begin{cases} \delta_1[s1] \wedge \delta_2[s2] & se\langle si, a, \psi_i, C_i', si'\rangle \notin L_i \wedge a \in J, i = 1,2 \\ (\delta_1(s1) \wedge Clock(\psi_2) \leq Min(\psi_2)) \vee (\delta_2(s2) \wedge Clock(\psi_1) \leq Min(\psi_1)) & \text{caso contrário} \end{cases}$$

onde $Clock(\psi)$ é definida como sendo a função que retorna o valor do relógio utilizado na condição ψ. O autômato resultante possui como conjunto de vértices o produto cartesiano dos vértices de P e Q, unidos com o produto cartesiano dos vértices de P com os vértices que representam a habilitação das ações a serem sincronizadas em $Q(S_q)$, e com o produto cartesiano dos vértices de Q com os vértices que representam a habilitação das ações a serem sincronizadas em $P(S_p)$.

A relação entre os estados de um sistema e os vértices de um autômato temporizado, representando o modelo deste sistema, está no fato de que cada vértice do autômato temporizado representa um ou mais estados do sistema.

4.3 O Tradutor RT-LOTOS / Autômatos Temporizados

A implementação direta do mapeamento de especificações RT-LOTOS para autômatos temporizados apresentada em [7], apesar de ser realizável, não é a melhor solução do ponto de vista prático, pois esta abordagem compromete o desempenho da tradução, conforme discutido em [25]. Portanto, escolheu-se neste trabalho, adotar uma tradução realizada em duas etapas, seguindo a proposta feita em [25]: a primeira etapa consiste na simplificação do modelo a ser manipulado, utilizando uma extensão das redes de Petri, a ser gerada a partir de uma especificação RT-LOTOS; a segunda etapa consiste na geração do autômato temporizado, obtido através da simulação da rede de Petri. A arquitetura do tradutor pode ser vista na figura 2.

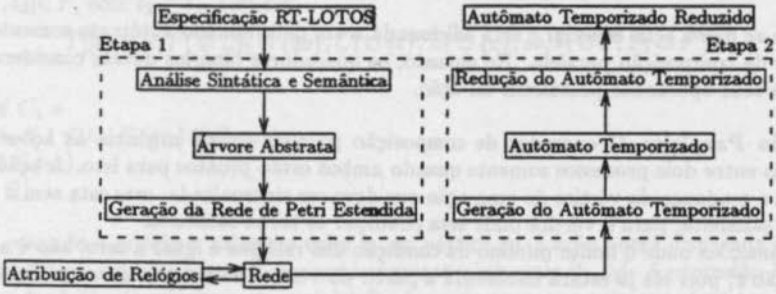

Figura 2: Tradutor RT-LOTOS / Autômato Temporizado

Análise Sintática e Semântica: Esta é a primeira fase do tradutor, desenvolvido neste trabalho, que consiste basicamente na verificação sintática e semântica da especificação. Foi utilizada como ferramenta auxiliar o sistema SYNTAX [5], que consiste de um conjunto de ferramentas que facilitam a concepção e a realização de tradutores, principalmente no domínio da compilação. Com esta ferramenta, e utilizando a teoria de Sintaxe Abstrata [20], foi construída a árvore abstrata da especificação.

Geração da Rede de Petri Estendida: O tradutor gera a rede de Petri estendida a partir da árvore abstrata da especificação, que contém apenas as informações essenciais à montagem desta, ou seja, os operadores associados aos seus respectivos operandos. A definição formal da rede de Petri estendida é baseada na forma apresentada em [25], sendo que o processo de geração desta rede para todos os operadores é apresentado detalhadamente em [19].

Atribuição de Relógios: A atribuição de relógios é feita apenas para as transições onde o disparo ocorrerá em um determinado instante, ou dentro de um intervalo finito. O algoritmo faz uma otimização sobre todas as transições que disparam instantaneamente, pois para tais transições é associado apenas um único relógio, que representará sempre o mesmo comportamento durante o percorrer da rede.

Geração do Autômato Temporizado: Após a geração da Rede de Petri e a atribuição de relógios às transições, é realizado o mapeamento da rede para um autômato temporizado. O algoritmo utilizado para este mapeamento, apresentado também em [25], inicia a montagem a partir da marcação inicial da Rede, M_I. Os vértices do autômato são obtidos a partir de todas as marcações acessíveis a partir de M_I. Para cada marcação, são identificadas todas as transições sensibilizadas pela mesma, gerando assim os arcos que partem do vértice relacionado à marcação. As condições relacionadas aos vértices são obtidas a partir das transições disparadas pela marcação analisada, observando os relógios e os limites associados a cada transição, colocando assim as restrições necessárias para cada relógio. Os relógios a serem inicializados por cada arco são obtidos a partir das marcações finais geradas pelas transições disparadas, analisando todas as transições sensibilizadas por estas marcações.

Redução do Autômato Temporizado: Esta fase tem como objetivo a eliminação de vértices repetidos, gerados pelo método apresentado, já que no momento da geração, ocorre a identificação distinta de vértices com as mesmas características, ou seja, as mesmas condições de atividade e arcos. Para que dois vértices sejam semelhantes, é necessário que a relação deles com o conjunto de relógios seja a mesma.

5 A Verificação

A linguagem TCTL

As propriedades a serem verificadas sobre o modelo do sistema serão descritas por fórmulas escritas na lógica temporal TCTL, construídas a partir de proposições e restrições sobre os relógios e vértices do modelo do sistema.

A lógica TCTL [3] (*Temporal Computation Tree Logic*) é uma extensão da lógica CTL [8], estendendo os operadores temporais $\exists u$ (existe uma execução) e $\forall u$ (em todas execuções), utilizando restrições temporais que permitam um tratamento temporal quantitativo.

As fórmulas TCTL seguem a seguinte gramática:

$$< fórmula >:=< predicado > | < fórmula > \wedge < fórmula > | < fórmula > \vee < fórmula > |$$
$$\neg < fórmula > | < fórmula > \exists u_I < fórmula > | < fórmula > \forall u_I < fórmula >$$
$$< predicado >:= init \mid enable(a) \mid after(a) \mid x \in I$$

sendo $init$ o vértice inicial do modelo do sistema, x um relógio pertencente ao conjunto de relógios do modelo do sistema, $enable(a)$ o conjunto de vértices do modelo do sistema onde a transição etiquetada com a é habilitada, e $after(a)$ o conjunto de vértices do modelo do sistema que são vértices de saída das transições etiquetadas com a. O intervalo de valores positivos inteiros é chamado I.

A fórmula $< fórmula1 > \exists u_I < fórmula2 >$ significa que existe uma execução do sistema com uma prefixação finita onde $< fórmula2 >$ é verdadeira no último vértice, com um tempo $t \in I$, sendo que $< fórmula1 >$ é continuamente verdadeira nos vértices anteriores. A fórmula $< fórmula1 > \forall u_I < fórmula2 >$ se diferencia no fato de que todas as execuções do sistema são verdadeiras. Através destas duas fórmulas, é possível obter algumas abreviações típicas, tais como:

$true \exists u_I <$ *fórmula* $>$ denotada por $\exists \circ_I <$ *fórmula* $>$
$true \forall u_I <$ *fórmula* $>$ denotada por $\forall \circ_I <$ *fórmula* $>$
$\neg \forall \circ_I \neg <$ *fórmula* $>$ denotada por $\exists \Box_I <$ *fórmula* $>$
$\neg \exists \circ_I <$ *fórmula* $>$ denotada por $\forall \Box_I <$ *fórmula* $>$

A ferramenta de verificação adotada neste trabalho (KRONOS) utiliza esta gramática conforme apresentado em [22], bem como as abreviações citadas.

A verificação de modelos (*"Model Checking"*)

A verificação de modelos, através de fórmulas TCTL, é baseada num algoritmo de verificação simbólica de modelos [15], que consiste em calcular o conjunto dos vértices do autômato temporizado do modelo do sistema que satisfazem à fórmula TCTL, definido como conjunto característico de uma fórmula TCTL. O algoritmo calcula este conjunto como sendo uma disjunção das restrições temporais sobre os relógios do modelo, representado por um conjunto finito de matrizes de pontos fixos.

6 Estudo de casos

6.1 Algoritmo de Exclusão Mútua

O algoritmo de exclusão mútua foi descrito por Lamport [16], sendo um caso clássico no estudo dos métodos formais de verificação de sistemas dependentes do tempo [18]. O objetivo é garantir a exclusão mútua num sistema concorrente, consistindo de vários processos que utilizam uma variável compartilhada para o auxílio no acesso a uma seção crítica, ou seja, somente um processo pode ter acesso à seção crítica em um determinado tempo. É assumido que cada processo tem um identificador distinto. Uma descrição abstrata do algoritmo pode ser feita da seguinte forma:

Processo *i*:
start: wait for x = 0
x := i
delay
if x $\neq$ i then goto *start*
x := 0

Assim, cada processo que solicita o acesso a seção crítica deve primeiramente esperar até que a variável compartilhada seja inicializada com o valor zero, indicando que nenhum outro processo está utilizando a seção crítica. Logo após, o processo atribui o valor do seu identificador à variável compartilhada. Desde que vários processos podem estar competindo pela seção crítica, o processo deve esperar até que o valor da variável compartilhada estabilize, verificando em seguida se o valor desta é ainda o valor de seu identificador. O processo no qual o valor do identificador for igual ao valor da variável compartilhada, ou seja, o último a inicializá-la, terá o direito a acessar a seção crítica, sendo que os demais devem esperar até que a variável seja reinicializada novamente, reiniciando o procedimento de escolha. Quando um processo termina a utilização da seção crítica, o valor da variável compartilhada é retornado a zero. Para evitar a violação da exclusão mútua, devido a possibilidade de uma diferença na velocidade dos processos, são adotadas restrições que estabelecem que todo processo deve esperar um tempo suficiente para que os outros processos verifiquem o novo valor da variável compartilhada.

Como o tradutor desenvolvido trata de especificações escritas em RT-LOTOS Básico, não é possível especificar o comportamento da variável compartilhada de forma genérica, sendo que para tal seria necessário utilizar a componente de RT-LOTOS que trata de tipos de dados. Assim, a especificação do algoritmo é apresentada a seguir, para um número de processos limitado (3), mas

observa-se que o acréscimo de processos pode ser feito de uma maneira trivial. A definição dos pontos de comunicação do processo foi omitida apenas para tornar a leitura mais simples.

```
specification Mutual_Exclusion_Algorithm[...]
behaviour
    (P[...] ||| P[...] ||| P[...]) |[vo, v, v1, v2, v3, s1, s2, s3]| Shared_Var[...]
where
    process P[...] :=
        i; v; [0, a]i; vi; [b, c]si; start; i; end; vo; stop <si] {si:  P[...]}
    endproc
    process Shared_Var[...] :=
        v; ((Cmp[...] >> vo; Shared_Var[...]) [] Shared_Var[...])
    where
        process Cmp[...] :=
            (v1; (s1; exit [] Cmp[...])) [] (v2; (s2; exit [] Cmp[...]))
            [] (v3; (s3; exit [] Cmp[...]))
        endproc
    endproc
endspec
```

Para ilustrar o método de tradução apresentado e analisar o resultado obtido pelo tradutor automático, os autômatos temporizados das especificações Shared_Var e P são representados graficamente pela figura 3.

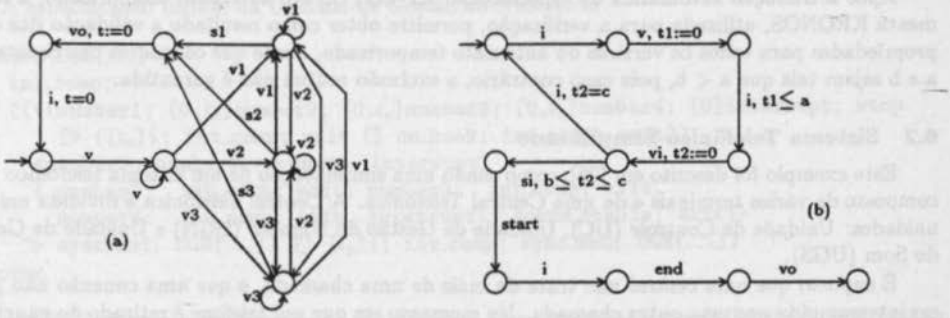

Figura 3: a)Processo Shared_Var; b)Processo P

A especificação Shared_Var representa o comportamento da variável compartilhada, onde é possível observar que não existem restrições temporais associadas aos arcos do autômato. Isto se deve ao fato da especificação P, que representa o comportamento dos processos que necessitam acessar a seção crítica, estabelece as restrições sobre o acesso à variável compartilhada. Assim, após um processo verificar, por meio da ação "v", indicando que a variável possui o valor zero, que a seção crítica não está sendo utilizada, o processo deve atribuir o valor de seu identificador "vi" à variável, num limite de tempo definido pelo intervalo [0, a]. A espera pela estabilização do valor da variável ocorrerá dentro do limite [b, c], sendo que se a ação "si" não ocorrer, indicando que a variável não possui o mesmo valor do identificador do processo, o processo terá que esperar até que a ação "v" fique disponível novamente para outra tentativa de acesso. A ação "vo" representa a reinicialização da variável compartilhada, executada quando um processo libera a seção crítica.

Verificação de Propriedades

A primeira verificação feita consistiu em garantir a propriedade de exclusão mútua. Esta propriedade estabelece que após o acesso a seção crítica por um processo, nenhum outro processo poderá acessá-la até que esta se torne disponível. Esta verificação não envolve nenhuma restrição temporal, mas consiste na verificação clássica do problema. A propriedade é estabelecida pela seguinte fórmula TCTL:

$$\texttt{init} \Rightarrow (\texttt{after(start)} \Rightarrow ((\forall\Box(\neg\texttt{enable(vo)}) \; \forall u \; (\forall\diamond \; (\neg(\texttt{enable(start)})))))$$

Como este trabalho objetiva verificar propriedades que envolvam limites temporais, foi considerada também a propriedade apresentada em [18], estabelecendo que a partir do momento em que qualquer processo tente acessar a seção crítica, existe um limite máximo, 2c + 5a, para que a seção crítica seja acessada por qualquer processo. Este limite é estabelecido através da análise das três fases mais importantes do algoritmo. A primeira estabelece que algum processo alcançará a seção crítica em, no máximo, c + a. A segunda diz que, com a seção crítica disponível, o primeiro evento importante será quando a variável compartilhada passar do valor zero para o valor do identificador de um processo, que ocorrerá até "a". E a última estabelece que a variável se tornará estável em até c + 3a, totalizando o limite apresentado acima. Esta propriedade é representada pela seguinte fórmula TCTL:

$$\texttt{after(v)} \Rightarrow (\forall\diamond_{\leq 2c+5a} \; (\texttt{enable(start)}))$$

Após a tradução automática da especificação RT-LOTOS em autômato temporizado, a ferramenta KRONOS, utilizada para a verificação, permitiu obter como resultado a validação das duas propriedades para todos os vértices do autômato temporizado, desde que os valores das constantes a e b sejam tais que a < b, pois caso contrário, a exclusão mútua não é garantida.

6.2 Sistema Telefônico Simplificado

Este exemplo foi descrito em [25], como sendo uma simplificação de um sistema telefônico real, composto de vários terminais e de uma Central Telefônica. A Central Telefônica é dividida em três unidades: Unidade de Controle (UC), Unidade de Gestão do Número (UGN) e Unidade de Gestão do Som (UGS).

É suposto que uma central não trata de mais de uma chamada, e que uma conexão não pode ser interrompida por uma outra chamada. No momento em que um telefone é retirado do gancho, a central está pronta para gerar uma composição de um número de chamada, realizada pela Unidade de Gestão do Número. Após o final da composição do número do terminal a ser chamado, a Central Telefônica inicia o tom de chamada através da Unidade de Gestão do Som. A duração da comunicação é limitada, ou seja, se o chamador não finalizar a comunicação até um determinado instante, t_m (seg.), a comunicação é interrompida.

Admite-se ainda que os números telefônicos são compostos de quatro algarismos, e que a sua composição deve obedecer as seguintes restrições:

- O terminal chamador deve discar o primeiro número até t_p (seg.), logo após o tom de discar.
- O tempo entre os algarismos não deve ultrapassar o limite de t_s (seg.).
- A composição de um número não deve ultrapassar o limite de t_c (seg.).

A composição de um número é interrompida se as restrições acima não forem respeitadas ou se o chamador retorna o telefone ao gancho.

Além disto, a Unidade de Gestão do Som provoca um toque, de duração t_n (seg.), no terminal chamado. Se este não responder até t_r (seg.), a chamada será cancelada, caso contrário, faz-se a sinalização de conexão para a Unidade Central.

A especificação formal da Unidade Central, escrita em RT-LOTOS, é dada a seguir:

```
process UC[...] :=
    off_hook; ini_comp;
    (int_comp; sync_end; UC[...]
     [] sound_enable; sync_end; ini_sound;
        (int_sound; UC[...]
         [] connect; (on_hook; UC[...]
                      [] (([t_m]i; [0]int_comm; exit
                           <int_comm] {int_comm:  exit} ) >> UC[...]))))
endproc
```

A ação sync_end é utilizada para sincronizar a inicialização das unidades da Central Telefônica, sendo assim uma ação interna do sistema. A ação int_comm representa a interrupção da comunicação feita pela Unidade Central, quando o tempo de conexão atingir o limite estabelecido pelo dispositivo de timeout ($[t_m]$i), inicializando a Central Telefônica. As ações ini_comp e int_comp representam, respectivamente, o início e a interrupção da composição do número chamado. As ações ini_sound e int_sound representam, respectivamente, o início e a interrupção do toque do terminal relativo ao número chamado. A ação sound_enable indica que a composição de um número foi encerrada e que a central está pronta para iniciar o toque do número chamado, e a ação connect indica o estabelecimento de uma conexão.

A especificação formal da Unidade de Gestão do Número é:

```
process UGN[...] :=
    ini_comp;
    ((((number1; [0,t_s]number2; [0,t_s]number3; [0,t_s]number4; [0]interrupt; stop
        [> ([t_c]i; int_comp; exit [] on_hook; int_comp; exit))
      <number2, number3, number4, interrupt]
      { number2:  int_comp; exit, number3:  int_comp; exit,
        number4:  int_comp; exit, interrupt:  sound_enable; exit})
     >> sync_end; UGN[...]) [] [t_p]i; int_comp; sync_end; UGN[...])
endproc
```

A ação interrupt também é uma ação interna, que representa a possibilidade de interrupção do início da chamada, após finalizar a composição do número.

Para cada digito do número chamado, existe a restrição sobre o tempo máximo para que o mesmo seja obtido. Este comportamento é expressado através do operador de tratamento de violações temporais. Existe também a restrição sobre o tempo máximo para a composição do número completo, expressa pelo dispositivo de watchdog. Um dispositivo de timeout checa se o primeiro digito será obtido até o limite de tempo permitido, caso contrário, a composição do número é cancelada.

A especificação formal da Unidade de Gestão do Som é:

```
process UGS[...] :=
    ini_sound; (Make_new_sound[...] [> ([t_r]i; int_sound; UGS[...]
                                        [] ready; connect; UGS[...]
                                        [] on_hook; int_sound; UGS[...]))
where
```

```
    process Make_new_sound[...] :=
            [0]ring; [t_n]i; Make_new_sound[...]
    endproc
endproc
```

Um dispositivo de timeout é utilizado para a intermitência do toque da campainha, e um dispositivo de watchdog para o encerramento do toque, caso o número chamado não responda.

A especificação formal da Central Telefônica é a composição paralela de suas unidades.

```
specification Central[...]
behaviour
   (hide [sync_end, interrupt] in
      (UC[...] |[ini_comp, int_comp, sound_enable, sync_end]|
       UGN[...])
   ) |[ini_sound, int_sound, connect]| UGS[...]
endspec
```

Verificação de Propriedades

A primeira propriedade a ser verificada estabelece que o usuário deverá finalizar a conversação até t_m segundos, caso contrário, a Central Telefônica se encarregará de encerrar a conexão, tornando-se disponível, representada pela seguinte fórmula TCTL:

$$\texttt{after(connect)} \Rightarrow \forall\circ_{\leq t_m}\ \texttt{init}$$

Após a demanda de comunicação, a Central Telefônica deverá estar livre no mais tardar depois de $t_c + t_r + t_m$ segundos. Esta propriedade é representada pela seguinte fórmula:

$$\texttt{init} \Rightarrow \forall\Box(\texttt{after(off\_hook)} \Rightarrow \forall\circ_{\leq t_c+t_r+t_m}\ \texttt{init})$$

O primeiro digito deverá ser obtido em até tp segundos, e o número completo deverá ser obtido em até tc segundos, caso contrário, a Central encerra a composição do número e torna-se disponível novamente. As duas fórmulas abaixo expressam estas propriedades.

$$\texttt{after(off\_hook)} \Rightarrow \forall\circ_{\leq t_p}\ (\texttt{after(number1)} \vee \texttt{init})$$
$$\texttt{after(off\_hook)} \Rightarrow \forall\circ_{\leq t_c}\ (\texttt{after(number4)} \vee \texttt{init})$$

A última propriedade estabelece que se o número chamado não responder em até tr segundos, após o início da chamada, então a Central Telefônica encerra esta demanda, tornando-se livre novamente. A fórmula correspondente é:

$$\texttt{after(ini\_sound)} \Rightarrow \forall\circ_{\leq t_r}\ (\texttt{after(connect)} \vee \texttt{init})$$

Essas propriedades a serem verificadas foram definidas no trabalho já referenciado [25]. Adotando valores de acordo com o funcionamento de um sistema real, o uso do processo de verificação através do tradutor RT-LOTOS/Autômato Temporizado e da ferramenta KRONOS para as especificações descritas acima nos permitiu verificar a satisfação de todas as fórmulas acima.

7 Conclusões

Neste artigo, foi apresentado uma abordagem para a verificação de especificações de sistemas tempo-real, escritas em RT-LOTOS, extensão temporizada de LOTOS. Esta abordagem consiste na tradução de uma especificação RT-LOTOS para um Autômato Temporizado, a partir do qual pode ser realizada a verificação de formulas da Lógica Temporal Tempo-Real TCTL.

Para este efeito, um tradutor de RT-LOTOS para Autômato Temporizado foi desenvolvido e implementado neste trabalho. Ele foi integrado com uma ferramenta de verificação de modelos já existente (KRONOS [22]). A abordagem de verificação citada e a ferramenta tradutor-verificador obtida foram testadas em vários casos, sendo que dois deles foram apresentados neste artigo. Entretanto, outros exemplos podem ser encontrados em [19], tais como: Protocolo Tick-Tock [11], Protocolo do Bit Alternante [12], Sincronização de Lábios [24], e o cruzamento generalizado Rodovia-Ferrovia [13].

Ao visto dos resultados obtidos, consideramos que a abordagem apresentada neste artigo configura-se em uma boa alternativa para analisar os sistemas tempo-real, descritos por extensões temporais de LOTOS, sendo complementar a abordagens por simulação, tal como a apresentada em [10]. As limitações atuais dessa abordagem dizem respeito sobretudo ao tratamento da variável @t, utilizada de forma clássica nas álgebras de processos temporizadas para guardar o valor do tempo de ocorrência de uma ação. Este problema pode levar em muitos casos a não-decidabilidade, em termos do processo de verificação. Entretanto, para um conjunto razoável de aplicações de sistemas tempo-real e de sistemas multimídias, o uso da abordagem de verificação apresentada pode ser considerada como uma boa opção para validar especificações.

Este trabalho se situa na linha da proposta feita em [25]. Em particular, o uso de Autômatos Temporizados para representar modelos dos sistemas dependentes do tempo, descritos por uma álgebra de processos temporizada, e a verificação de sistemas a partir destes, têm sido apresentados em vários trabalhos existentes na literatura [10, 11, 25]. Em [25], foi desenvolvido um tradutor de especificações escritas através da álgebra de processos ATP [21] para autômatos temporizados, porém, a linguagem ATP possui limitações quanto a representação da ocorrência de ação num intervalo de tempo, e do tratamento de exceções temporais. Em [11], é apresentada uma proposta de mapeamento para uma extensão temporal de LOTOS, diferente da deste artigo e chamada ET-LOTOS, porém, nenhum mecanismo de tradução automático foi desenvolvido nem implementado, deixando incompleto uma validação da abordagem para as extensões temporais de LOTOS. No trabalho apresentado neste artigo, além da verificação da adequação da abordagem apresentada, para o caso das extensões temporais de LOTOS, foi desenvolvido, implementado e testado um tradutor de especificações RT-LOTOS em Autômatos Temporizados, para ser associado a uma ferramenta de verificação já existente, no caso KRONOS. A adaptação deste tradutor a um futuro padrão para a extensão temporizada de LOTOS, e a integração a outras ferramentas de verificação baseadas em Autômatos Temporizadas (HyTech [14]) fazem parte das perspectivas de continuação deste trabalho. Enfim, a utilização desta abordagem e da ferramenta associada estão previstas no projeto aprovado PROTEM-CC (fase 3), denominado DAMD - Design de Aplicações Multimídias Distribuídas.

Referências

[1] R. Alur, C. Courcoubetis and D. Dill. *Model Checking for Real-Time Systems. In Procs. of the 5th IEEE Symposium on Logic in Computer Science*, 1990.

[2] R. Alur and D. Dill. *The theory of Timed Automata*. Proccedings of the REX Workshop, Lecture Notes in Computer Science No. 600, Mook, The Netherlands, June, 1991.

[3] R. Alur and T. A. Henzinger. *Logics and Models for Real Time: A Survey. In Procs. of the REX Workshop*, Mook, The Netherlands, June, 1991, Lecture Notes in Computer Science No. 600, Springer-Verlag, 1992.

[4] T. Bolognesi, F. Lucidi and S. Trigila. *Converging Towards a Timed-LOTOS Standard.* Reserarch Report, CNUCE/C.R.N., Pisa, Itália, 1993.

[5] P. Boullier et P. Deschamp. *Le Système SYNTAX: Manuel d'Utilisation et de Mise en Oeuvre sous UNIX.* Project Langages et Traducteurs, INRIA, setembro, 1988.

[6] M. S. de Camargo. *Tornando a Linguagem LOTOS Apta para Especificar Sistemas Dependentes do Tempo.* Tese de Doutorado LCMI/UFSC, Florianópolis, 1995.

[7] M. S. de Camargo e J. -M. Farines. *Uma abordagem para especificação e verificação de sistemas dependentes do tempo.* IX SBES, 1995.

[8] E. M. Clarke, E. A. Emerson and A. P. Sistla. *Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Transactions on Programming Languages and Systems,* 8(2):244-263, 1986.

[9] J. -P. Courtiat and R. C. Oliveira. *About time nondeterminism and exception handling in a temporal extension of LOTOS. In Protocol Specification, Testing and Verification XIV,* pages 37-52, Canada, 1994.

[10] J. -P. Courtiat and R. C. Oliveira. *RT-LOTOS and its application to multimedia protocol specification and validation.* MnNet'95 - International Conference on Multimedia and Networking, Aizu, Japan, September, 1995.

[11] C. Daws, A. Olivero and S. Yovine. *Verifying ET-LOTOS programs with KRONOS.* In Proceedings of the FORTE94, Berne, Switzerland, Outubro, 1994.

[12] H. Garavel. *Compilation et Vérification of LOTOS Programs.* Thèse de Docteur de lUniversité Joseph Fourier - Grenoble I, França, Novembro, 1989.

[13] C. Heitmeyer and N. Lynch. *The Generalized RailRoad Crossing: A case study in formal verification of real-time systems.* Laboratory for Computer Science - MIT, Cambridge, Massachusetts, November, 1994.

[14] T. A. Henzinger and P. -H. Ho. *HyTech: The Cornell HYbrid TECHnology Tool. In Hybrid Systems II, Lecture Notes in Computer Science 999,* Springer-Verlag, 1995, pp. 265-294.

[15] T. A. Henzinger, X. Nicollin, J. Sifakis and S. Yovine . *Symbolic model-checking for real-time systems. In Procs. of 7th LICS,* IEEE Computer Society Press, 1992.

[16] L. Lamport. *A fast mutual exclusion algorithm.* ACM Transactions on Computer Systems, 5(1):1-11, fevereiro, 1987.

[17] G. Leduc and L. Léonard. *A Formal Definition of Time in LOTOS.* Research Report, Université de Liège, Institut d'Electricité Montefiori, Bélgica, 1994.

[18] V. Luchangco, E. Söylemez, S. Garland and N. Lynch . *Verifying Timing Properties of Concurrent Algorithms. In Proc. FORTE'94,* Berne, Swtzerland, outubro, 1994.

[19] R. F. Martins. *Verificação de sistemas dependentes do tempo a partir de especificações escritas em RT-LOTOS.* Dissertação de Mestrado LCMI/UFSC, Florianópolis, Junho, 1996.

[20] B. Meyer. *Introduction to the Theory of Programming Languages.* International Series in Computer Science, Prentice-Hall, 1990.

[21] X. Nicollin. *ATP: une algèbre pour la spécification et l'analyse des systèmes temps reel.* Thèse, Institut National Polytechnique de Grenoble, France, maio, 1992.

[22] A. Olivero and S. Yovine. *KRONOS: A Tool for Verifying Real-Time Systems, User's Guide and Reference Manual.* Montbonnot Soint Martin, França, agosto, 1993.

[23] T. Regan. *Multimidia in Temporal LOTOS.* In IFIP - Protocol Specification, Testing and Verification, XIII (C-16), Dantine, A.; Leduc, G.; Wolper, P. (Editors), Elsevier Science Publishers B.V. (North-Holland), pp. 127-143, 1993.

[24] J. -B. Stefani, L. Hazard and F. Horn. *Computational model for distributed multimedia applications based on a synchronous programming language.* Computer Communications, 15(2):114-128, March, 1992.

[25] S. Yovine. *Méthodes et Outils pour la Vérification Symbolique de Systèmes Temporisés.* Thèse de Docteur de l'Institut National Polytechnique de Grenoble - Grenoble, França, Maio, 1993.