


Análise Formal de Sistemas

Uma abordagem orientada a linguagem

Christiano Braga
Instituto de Computação, UFF

E. Hermann Haeusler
Depto. de Informática, PUC-Rio

Escola de Verão, Depto. de Matemática, Unb, 2006

Agradecimentos

- Prof. Maurício Ayala pelo convite.
- Dr. Carlos Bazílio pela "consultoria" em Promela e XSB/LMC.

Importante

- Este trabalho se encontra em desenvolvimento.
- Formalização de uma técnica que os autores aplicavam em contextos diferentes e com formalismos diferentes.
- A expressão "análise de sistemas" *nada* tem a ver com análise estruturada de software, ou orientada a objetos.
- "Análise" aqui se refere, essencialmente, a raciocinar sobre o modelo de um sistema.

Objetivo

- Apresentar uma abordagem formal para análise sistemas orientada a linguagem.

- Ilustrar como matemática pode e deve ser utilizada em computação.

- Esta apresentação foi construída sobre um exemplo (o protocolo de eleição de líder) para auxiliar o entendimento das idéias a serem discutidas.

    - Deve ficar claro, no entanto, que as técnicas apresentadas aqui *não* se restringem somente ao domínio de aplicações dos protocolos de comunicação.


O que é análise formal de sistemas?

- O que é análise de sistemas? No escopo deste trabalho, análise de sistemas significa *raciocinar* sobre um sistema através do seu *modelo*.
- O que significa ser formal? Ser formal significa ter *suporte computacional* para a realização de uma dada tarefa utilizando ferramental *matematicamente* definido. Por ferramental queremos dizer técnicas e software que implementam estas técnicas.


O que é análise formal de sistemas?

- Então, o que é, análise formal de sistemas? É o raciocínio sobre sistemas, através de seu modelo, utilizando ferramental, teórico e de software, matematicamente definido.


Por quê analisar sistemas?

- Exemplo: eleição de líder num anel assíncrono. Algoritmos distribuídos formam exemplos importantes em computação, uma área com bastante investimento em análise formal.

- O que é eleição de líder? Eleição de líder é um algoritmo distribuído ou protocolo de comunicação que visa, através de *trocas de mensagens* entre *nós* de uma rede (*grafo*), eleger um (nó) líder pela comparação de seus números identificadores.

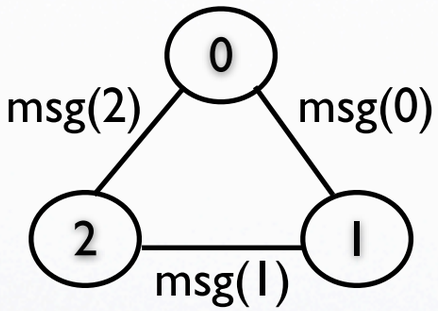
- Para que serve? Por exemplo, dado um *pool* de servidores onde um deles deve distribuir tarefas, caso ocorra uma falha eles devem eleger um novo distribuidor de tarefas.


Por quê analisar sistemas?

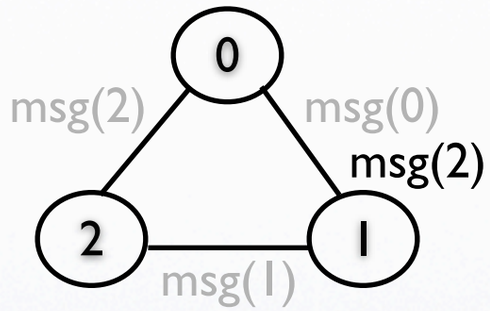
- Como funciona? Inicialmente os nós enviam para seus nós adjacentes seus próprios identificadores.
    - Se um nó recebe um identificador menor que o seu ele desconsidera a mensagem. (Ele é, potencialmente, um líder.)
    - Caso contrário, o nó repassa a seus vizinhos o identificador maior.
    - Quando um nó recebe uma mensagem com seu próprio identificador ele sabe que é o líder.
- Vejamos um exemplo concreto.

Por quê analisar sistemas?

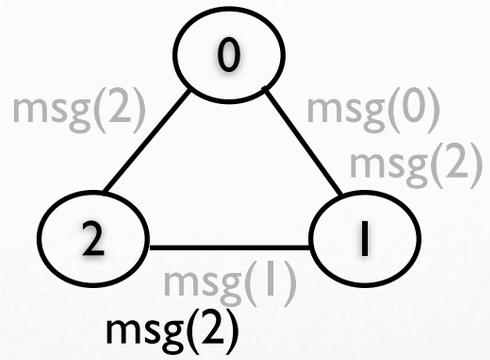
- Eleição de líder: *topologia* (grafo) com três nós



Início



Encaminhamento das mensagens



Identificação do líder


Por quê analisar sistemas?

- Como se analisa eleição de líder? Analisar é pensar ou raciocinar sobre o modelo de um sistema. Para o algoritmos de eleição de líder o que, em geral, *se deseja saber é se um único líder é eleito, se for*.

- O raciocínio ocorre sobre as diversas possíveis seqüências de trocas de mensagens que podem ocorrer.

- Note que a seqüência mostrada na figura anterior é *uma* possível seqüência: existem diferentes ordens de envio e processamento das mensagens.


Por quê analisar sistemas?

- Como se analisa eleição de líder formalmente? Para que um computador possa "responder a perguntas" como "se um líder for eleito, será somente um" é preciso formalizar o sistema e as perguntas sobre o sistema, isto é, descrevê-las num formato que o computador possa entender.

- Em outras palavras, é necessário e*specificar* o algoritmo para que um *modelo* formal possa ser construído.

- Com o modelo construído pode-se *caminhar* sobre ele procurando responder as perguntas.


Como analisar sistemas formalmente?

- O que é mesmo ser formal? Ser formal é ser matematicamente preciso, mas de tal maneira que um computador possa entender.
    - Lamport diz que ao sermos formais percebemos quão imprecisa ("sloppy") nossa matemática é.
- Ser formal é necessário?
    - Ser algorítmico é bom para uma análise manual.
    - O suporte computacional é muito importante: sistemas ditos *mission critical* são, em geral, submetidos a análise formal.


Como analisar sistemas formalmente?

- Uma técnica formal de verificação que se tornou muito conhecida é a de *model-checking* ou verificação de modelo.

- É uma técnica *automática* de verificação: diferentemente de prova de teoremas não é necessária a intervenção humana.

- Essencialmente o algoritmo de *model-checking* combina o modelo da aplicação sendo analisada com o modelo associado a uma fórmula (em geral em lógica temporal) que representa uma pergunta sendo feita ao sistema, procurando pelo *menor ponto fixo* que satisfaça aquela pergunta.


Como analisar sistemas formalmente?

- Para analisarmos formalmente o algoritmo de eleição de líder precisamos primeiro *especificá-lo*. A especificação em geral diz "o que" a aplicação deve fazer, e não "como" deve-se fazer o que tem que ser feito.

- Uma forma de especificar é através de um cálculo de processos ou de um autômato finito.

- A abordagem por autômato: descreve abstratamente cada uma das três situações (estados) possíveis ilustradas anteriormente e como o sistema muda (transições) de uma situação para a outra. (Veremos exemplos concretos a frente.)

- Um modelo é formado por *configurações* de estados que satisfazem a especificação.


Como analisar sistemas formalmente?

- Especificar é, em última análise, entender um problema. Uma questão importante em especificação é como nos convencermos de que o que especificamos é de fato o que se deseja.

- Por isto análise é importante. É importante também um processo incremental que alterne entre a especificação e a análise para que o modelo seja aos poucos enriquecido de forma fazer valer certas propriedades, assim como *não permitir* outras.


Por quê a abordagem orientada a linguagem é interessante?

- Ponto chave: permitir a especificação e análise no nível do *domínio da aplicação* ⇒ *encapsular* a linguagem e o formalismo de análise.

- Duas abstrações interessantes em protocolos de comunicação:

  - Topologia do protocolo: por exemplo, eleição de líder não precisa ser somente num anel.

  - Separação entre a especificação do protocolo e o processo de análise.


Por quê a abordagem orientada a linguagem é interessante?

- Dois exemplos em *model-checkers* importantes.
- Especificação do algoritmo de eleição de líder que acompanha a distribuição de:
  - Promela, linguagem de especificação do *model-checker* Spin, desenvolvido no Bell Labs;
  - XSB/LMC, linguagem lógica que implementa pi-calculus, desenvolvida em SUNY em Stony Bruck.


Por quê a abordagem orientada a linguagem é interessante?

- Em Promela:

```
proctype node (chan in, out; byte mynumber)
{
    ...
    :: in?winner,nr ->
            if
            :: nr != mynumber ->
            printf("MSC: LOST\n");
            :: else ->
                printf("MSC: LEADER\n");
                nr_leaders++;
               assert(nr_leaders == 1)
            fi;
            if
            :: know_winner
            :: else -> out!winner,nr
            fi;
    ...
}
```

Amarra a topologia.

Anotação para o analisador.


Por quê a abordagem orientada a linguagem é interessante?

- Em XSB/LMC:

```
...
nodeActive(Right, Left, N, Maxi, Nbr) ::=
    Left ? first(NId);
    if (NId \== Maxi)
    then{Right ! second
         nodeActive(Right,                  , NId) }
    else
    if (Maxi is N-1)
    then{action(leader);
         nodeActive(Right, Left, N, Maxi, NId)}
    e{ action(fail) }
```

Amarra a topologia.

Anotação para o analisador.

Anotação para o analisador.


Por quê a abordagem orientada a linguagem é interessante?

- Trecho de especificação que amarra a topologia: define um canal de comunicação de leitura de informação do nó anterior e um canal de comunicação de escrita com o próximo nó.
- A "anotação" na especificação faz com que o traço de execução produzido por aquele trecho da especificação tenha a marca, por exemplo, de que um líder foi eleito. Com estas marcas o *model-checker* pode, por exemplo, perguntar ao modelo se o líder foi eleito.


Por quê a abordagem orientada a linguagem é interessante?

- É importante enfatizar que estas duas abstrações (independência de topologia e separação da especificação da análise) são características do domínio de protocolos de comunicação, mas que a idéia de se *abstrair* conceitos, de *encapsular* o formalismo através de linguagens específicas de domínio, é geral e pode ser aplicada a *qualquer* domínio.


O que é a abordagem orientada a linguagem?

- O objetivo da técnica é permitir a especificação e análise no nível do domínio da aplicação.

- Alguns exemplos de abstração: utilizando o exemplo do algoritmo de eleição de líder, uma especificação do algoritmo não deveria se preocupar com a organização da rede. Um dos conceitos importantes é a idéia de *vizinhança*.

- Linguisticamente, este conceito poderia ser capturado, utilizando notação de cálculo de processos, da seguinte forma: `neighbours!id` representando o envio do identificador do nó aos seus vizinhos.


O que é a abordagem orientada a linguagem?

- Outro conceito interessante é o de *broadcasting*, utilizado por exemplo em algoritmos de consenso.
- No algoritmo de eleição de líder pode-se, ao final, avisar a todos os nós que um líder foi eleito: `everyone!leader` onde `everyone` representa todos os nós da rede e `leader` é uma mensagem representando que um líder foi eleito.


O que é a abordagem orientada a linguagem?

- As propriedades a serem verificadas também devem ser especificadas utilizando elementos léxicos do domínio. Por exemplo: `<>someone!leader`, onde o operador `<>` significa "eventualmente", e a fórmula (em lógica temporal) especifica que eventualmente algum nó anuncia que um líder foi eleito.


O que é a abordagem orientada a linguagem?

- "Especificações incrementais": alternância entre modelagem e análise, para facilitar o entendimento do problema. Podemos ter mente o modelo comportamental do sistema, propriedades que valem no modelo e traços que não valem.

- Exemplo: imagine uma primeira especificação da eleição de líder sem a especificação do que acontece quando o identificador recebido por um nó é menor do que o seu próprio.

  - Isso poderia "estourar" os buffers dos nós.

  - Análise poderia identificar esta situação.


O que é a abordagem orientada a linguagem?

- Todo o ferramental teórico (formalismo de análise) e computacional (ferramentas de análise) que discutimos até agora podem ser construídos especificamente para cada domínio (e.g. protocolos de comunicação). Obviamente um esforço muito grande...

- A abordagem orientada a linguagem busca relacionar, de maneira *matematicamente* precisa, o domínio da aplicação com um formalismo de análise de forma a *reaproveitar* o ferramental existente.

- Por matematicamente preciso queremos dizer que esta relação deve *preservar* a semântica do domínio da aplicação no formalismo de análise.


O que é a abordagem orientada a linguagem?

- Do ponto de vista sintático, a técnica se utiliza de:
  - Duas linguagens formais (isto é, inteligíveis por computadores).
    - Uma representando o domínio da aplicação (ex. protocolos de comunicação).
    - Outra representando o formalismo de análise (ex. cálculo de processos ou lógica de reescrita).
- Modelo computacional: uma especificação, no formalismo de análise, que *capture os principais conceitos* do domínio da aplicação (ex. processos, mensagens, ou buffers).


O que é a abordagem orientada a linguagem?

- Do ponto de vista sintático, a técnica se utiliza de (cont.):
  - Uma função de transformação que dada uma especificação no domínio da aplicação produza uma outra no formalismo de análise que faça uso do modelo computacional do item anterior.
  - Suporte a especificação incremental: funções de *pretty-printing* do formalismo de análise para o domínio da aplicação para:
    - a especificação do sistema;
    - para as propriedades e
    - para os traços.


O que é a abordagem orientada a linguagem?

- Do ponto de vista semântico, é necessário demonstrar que a função de transformação preserva a semântica do domínio da aplicação.
- Significado de um especificação incremental: é uma tripla onde ...
  - Primeira projeção é o sistema de transição associado ao modelo;
  - Segunda projeção são as fórmulas que valem naquele modelo;
  - Terceira projeção são os traços de propriedades que não valem naquele modelo.


O que é a abordagem orientada a linguagem?

- A demonstração de preservação da semântica: dadas duas especificações incrementais, a primeira no domínio da aplicação e a segunda no formalismo de análise, cada um dos componentes das duas especificações devem ser, par a par, simulações um do outro, isto é, *bisimilares*.

- Bisimilação entre dois modelos: a cada transição entre dois estados do primeiro, deve existir uma transição entre dois estados *bisimilares* do segundo. (Mais formalmente, o que se busca numa bisimulação é o *maior ponto* fixo da relação.)

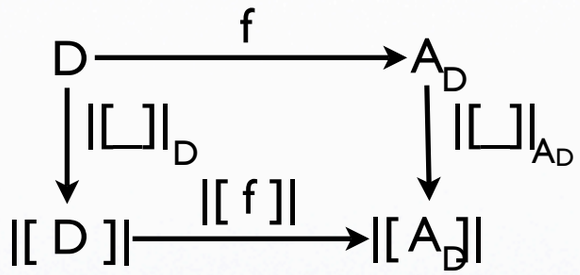

O que é a abordagem orientada a linguagem?

- No exemplo de eleição de líder: para cada dois estados relacionados pela relação de transição no sistema de transição associado ao protocolo de comunicação (dois grafos como no exemplo) devem existem dois estados (bisimilares) no sistema de transição associado a especificação em, por exemplo, pi-calculus ou lógica de reescrita.

- Com relação as funções de *pretty-printing* elas devem relacionar elementos das álgebras iniciais das duas especificações.

O que é a abordagem orientada a linguagem?

- A abordagem orientada a linguagem pode ser sumarizada através do seguinte diagrama:



onde D é a linguagem formal que representa o domínio da aplicação; A_D é linguagem do formalismo de análise restrita aos conceitos do domínio D; f é a função de transformação de D para A_D; $|[\_]|_D$ e $|[\_]|_{A_D}$ são as funções semânticas para especificações em D e A_D, respectivamente; |[D]| é a semântica do domínio da aplicação D; |[A_D]| é a semântica do formalismo de análise A restrito aos conceitos de D; e |[f]| é a semântica de f.

O que é a abordagem orientada a linguagem?

- O significado de uma especificação em |[D]| é uma tripla $< m, \Delta, k >$, onde m é sistema de transição que modela a especificação em D, Δ são as propriedades válidas em m, e k são os contra-exemplos para propriedades inválidas em m.

- Dizer que uma especificação em A_D preserva a semântica de de uma especificação em D significa provar que:

$$\forall\ d \in D\ |[\ f\ ]|\ |[d]|_D \overset{\overline{bis}}{\leftrightarrow} |[\ f(d)\ ]|_{A_D}$$

onde $\overline{bis}$ é a relação de bisimulação para a par entre entre os elementos de $|[\ f\ ]|\ |[d]|_D$ e $|[\ f(d)\ ]|_{A_D}$.


Como utilizar a abordagem orientada a linguagem?

- LEP: linguagem para especificação de protocolos definida por Carlos Bazílio e o segundo autor deste trabalho.

- LEP se encaixa perfeitamente nesta discussão: é uma *linguagem específica de domínio* para definição de protocolos de comunicação.

- O protocolo de eleição de líder tem uma descrição bastante elegante em LEP.


Como utilizar a abordagem orientada a linguagem?

- LEP: uso de pronomes, tanto a especificação quanto a análise do protocolo ficam simplificadas.

- Os pronomes representam exatamente as abstrações as quais nos referimos anteriormente, como neighbours, everyone e someone, capturando os conceitos necessários ao algoritmo de eleição de líder, por exemplo.

- Outra consequência do uso de pronomes: a especificação independente de topologia.

- Na linguagem existem também construções para definir a topologia, como por exemplo Ring(5) que define um anel de cinco nós.


Como utilizar a abordagem orientada a linguagem?

- Junto com LEP existe uma arquitetura que permite mapear a especificação LEP para diferentes ferramentas de análise que implementam a técnica de model-checking, como Spin ou SMV, através da tradução de LEP para as linguagens de especificação dos model checkers.

- Para tanto foi definido um modelo de comunicação (ou modelo computacional) que define os mecanismos básicos de comunicação.


Como utilizar a abordagem orientada a linguagem?

- Eleição de líder em LEP:

```
topology is Ring(5), directed, reliable;
module candidate
    vars my, p, count : int;
    init ->this; neighbours!msg(my, count);
    this?win -> stop;
    this?msg(p, count) ->
        if ((p > my) or
                ((p == my) and (count < topology.size))) then
            my = p;  count = count + 1;
            neighbours!msg(my, count);
        else
            if ((p == this) and (count > topology.size)) then
                everyone!win; stop;
            endif
        endif
endmodule

Propriedade: [](any!msg → <> someone!win)
```


Como utilizar a abordagem orientada a linguagem?

- Maude MSOS Tool (MMT): este segundo exemplo reforça que a técnica de análise orientada a linguagem não serve somente para o domínio de protocolos de comunicação.

- Vamos apresentar uma simulação da aplicação da técnica de análise orientada a linguagem utilizando a linguagem de especificação MSDF (implementada em MMT) representando a linguagem do domínio de aplicação e Maude, uma implementação de lógica de reescrita, como exemplo de linguagem de análise.

- Este trabalho contou com a colaboração de Fabrício Chalub, José Meseguer e Peter Mosses.


Como utilizar a abordagem orientada a linguagem?

- MSDF é na realidade uma linguagem de espectro amplo e não específica de domínio, podendo então especificar qualquer tipo de sistema formal.

- No entanto, o paradigma declarativo e suas características sintáticas permitem descrever de maneira bastante sucinta o protocolo de eleição de líder numa especificação mais simples do que aquelas que apresentamos anteriormente.

- Não explorarmos aqui a idéia de pronomes, que pode no entanto ser definida através de módulos na linguagem.


Como utilizar a abordagem orientada a linguagem?

- Aplicamos as idéias de análise orientada a linguagem para permitir a análise de especificações MSDF em Maude:
  - foi definida uma especificação em Maude que captura os principais conceitos de MSDF;
  - foi definida uma função de transformação de MSDF para Maude assim como uma função de pretty-printing ;
  - e finalmente provamos a existência de bisimulação entre especificações MSDF e aquelas produzidas pela função de transformação.


Como utilizar a abordagem orientada a linguagem?

- Eleição de líder em MSDF:

```
(msos LEADER-ELECTION is
 see PROTOCOL .

 ProcState ::= start | waiting | leader .

 Process ::= prc (Int, Int, ProcState) .

 Message ::= m Int to Int .

 prc (Int, Int', start) : System -->
               prc (Int, Int', waiting) (m Int to Int') .

    Int'' > Int
 -- ------------------------------------------------------
 prc (Int, Int', waiting) (m Int'' to Int) : System -->
            prc (Int, Int', waiting) (m Int'' to Int') .

    Int'' < Int
 -- ------------------------------------------------------
 prc (Int, Int', ProcState) (m Int'' to Int) : System -->
            prc (Int, Int', ProcState) .

    Int'' == Int
 -- ------------------------------------------------------
 prc (Int, Int', waiting) (m Int'' to Int) : System -->
                                  prc (Int, Int', leader) .
sosm)
```


Como utilizar a abordagem orientada a linguagem?

- Note que não existem anotações no código para ajudar o analisador. As propriedades a serem definidas são feitas externamente a especificação do protocolo.

- Podemos analisar esta especificação utilizando as técnicas de programação em lógica e *model-checking* disponíveis em MMT, built-in no sistema Maude.


Como utilizar a abordagem orientada a linguagem?

```
--- Busca por estados com um processo eleito como líder.

(search init =>* < S:System ::: 'System, R:Record >
                              s.t. leaders(S:System) == 1 .)
Solution 1
R:Record --> {null}; S:System -->
(m 0 to 1)(m 1 to 2)
prc(0,1,waiting)prc(1,2,waiting)prc(2,0,leader)

Solution 2
R:Record --> {null}; S:System -->
(m 1 to 2)prc(0,1,waiting)prc(1,2,waiting)prc(2,0,leader)

Solution 3
R:Record --> {null}; S:System -->
(m 0 to 1)prc(0,1,waiting)prc(1,2,waiting)prc(2,0,leader)

Solution 4
R:Record --> {null}; S:System --> prc(0,1,waiting)prc(1,2,waiting)prc(2,0,leader)

No more solutions.
rewrites: 7633 in 740ms cpu (771ms real) (10314 rewrites/second)
```


Como utilizar a abordagem orientada a linguagem?

```
--- Busca por estados finais.

(search init =>! < S:System ::: 'System,R:Record > .)

Solution 1
R:Record --> {null}; S:System --> prc(0,1,waiting)prc(1,2,waiting)prc(2,0,leader)

No more solutions.
rewrites: 9907 in 740ms cpu (792ms real) (13387 rewrites/second)


--- Eventualmente um líder sempre é encontrado.

reduce in CHECK : modelCheck(init,<>[]oneLeader)
result Bool : true

rewrites: 9747 in 720ms cpu (772ms real) (13537 rewrites/second)
```


Como utilizar a abordagem orientada a linguagem?

```
--- Não é verdade que sempre não se acha um líder.

reduce in CHECK : modelCheck(init,~[]noLeader)
result Bool : true

rewrites: 10349 in 760ms cpu (791ms real) (13617 rewrites/second)

--- Nenhum líder é encontrado até que se encontre um líder.
--- (Ou seja, só um líder é encontrado.)

reduce in CHECK : modelCheck(init,[](noLeader U oneLeader))
result Bool : true
```


Conclusão

- Apresentamos uma abordagem orientada a linguagem para análise formal de sistemas.
- Discutimos o que é análise formal, através do protocolo de eleição de líder.
- Mostramos como a linguagem específica para o domínio de protocolos de comunicação LEP pode auxiliar a descrição e análise destes protocolos.
- Mostramos como o ambiente Maude MSOS Tool foi construído utilizando esta técnica.

Conclusão

- A abordagem orientada a linguagem parece ser uma técnica interessante para análise formal de sistemas.

- Trabalhos futuros incluem representar a semântica descrita aqui na forma de uma sintaxe concreta para permitir a especificação de ambientes formais.

- Aplicar a técnica sobre outros exemplos relacionando outros formalismos.


Obrigado!

Perguntas?