Sistemas interativos de prova

Carlos Henrique Cardonha

Universidade de São Paulo
Instituto de Matemática e Estatística
Departamento de Ciência da Computação

Elementos

- Verificador: Máquina de Turing de capacidade limitada.
- Provador: Máquina de Turing de capacidade ilimitada.
- L: Linguagem num alfabeto finito Σ.
- x: Palavra em Σ^*.

Objetivo

- O verificador quer saber se $x \in L$.
- Deve consumir tempo polinomial em $|x|$.
- Decisão pode não ser feita trivialmente.
- Interação com provador para fazer a decisão.

Ferramentas de interação

- A interação entre as máquinas ocorre por meio de fitas com políticas específicas de uso.
- Em um sistema interativo, são utilizadas 4 fitas:

Ferramentas de interação

- A interação entre as máquinas ocorre por meio de fitas com políticas específicas de uso.
- Em um sistema interativo, são utilizadas 4 fitas:
  - X: Read-only para as duas máquinas. Contém a palavra x.

Ferramentas de interação

- A interação entre as máquinas ocorre por meio de fitas com políticas específicas de uso.
- Em um sistema interativo, são utilizadas 4 fitas:
  - X: Read-only para as duas máquinas. Contém a palavra x.
  - F_P: Read-only para o verificador.

Ferramentas de interação

- A interação entre as máquinas ocorre por meio de fitas com políticas específicas de uso.
- Em um sistema interativo, são utilizadas 4 fitas:
  - X: Read-only para as duas máquinas. Contém a palavra x.
  - F_P: Read-only para o verificador.
  - F_V: Read-only para o provador.

Ferramentas de interação

- A interação entre as máquinas ocorre por meio de fitas com políticas específicas de uso.
- Em um sistema interativo, são utilizadas 4 fitas:
  - X: Read-only para as duas máquinas. Contém a palavra x.
  - F_P: Read-only para o verificador.
  - F_V: Read-only para o provador.
  - τ: Fita de bits aleatórios utilizada pelo verificador.

Questões relevantes

- O verificador vai trocar mensagens com o provador.

Questões relevantes

- O verificador vai trocar mensagens com o provador.
- O provador tem capacidade ilimitada (responde qualquer questão do verificador em tempo constante).

Questões relevantes

- O verificador vai trocar mensagens com o provador.
- O provador tem capacidade ilimitada (responde qualquer questão do verificador em tempo constante).
- Perguntas:

Questões relevantes

- O verificador vai trocar mensagens com o provador.
- O provador tem capacidade ilimitada (responde qualquer questão do verificador em tempo constante).
- Perguntas:
  - Por que o verificador não pergunta imediatamente se $x \in L$?

Questões relevantes

- O verificador vai trocar mensagens com o provador.
- O provador tem capacidade ilimitada (responde qualquer questão do verificador em tempo constante).
- Perguntas:
  - Por que o verificador não pergunta imediatamente se $x \in L$?
  - Para que uma fita de bits aleatórios?

Questões relevantes

- O verificador vai trocar mensagens com o provador.
- O provador tem capacidade ilimitada (responde qualquer questão do verificador em tempo constante).
- Perguntas:
  - Por que o verificador não pergunta imediatamente se $x \in L$?
  - Para que uma fita de bits aleatórios?
- Resumindo: Qual a graça do problema?

Resposta para uma das questões relevantes

- Existem provadores não-confiáveis.
- Afirmam que $x \in L$ mesmo quando isso não é verdade.
- Ou seja, perguntar diretamente se a pertinência é válida não vai resolver o problema do verificador...

Resposta para uma das questões relevantes (cont.)

- Verificador deve fazer perguntas especiais para o provador.
- Intuitivamente, o ideal seria o verificador fazer perguntas extremamente “inteligentes”.
- Porém, o provador é ilimitadamente inteligente. Logo, a inteligência, sozinha, não resolve o problema do verificador...

Resposta para a outra questão relevante

- Se a inteligência não é suficiente, o que nos resta?

Resposta para a outra questão relevante

- Se a inteligência não é suficiente, o que nos resta?
- A aleatoriedade :).

Resposta para a outra questão relevante

- Se a inteligência não é suficiente, o que nos resta?
- A aleatoriedade :).
- O uso de τ é essencial para que o verificador evite truques do provador (ainda que nem sempre possa evitá-los).

Classe IP

- Seja $q(n)$ uma função de $\mathbb{N}$ em $\mathbb{N}$.
- Seja $n = |x|$.
- Dizemos que L pertence a $\mathbf{IP}(q(n))$ se:

Classe IP

- Seja $q(n)$ uma função de $\mathbb{N}$ em $\mathbb{N}$.
- Seja $n = |x|$.
- Dizemos que L pertence a $\mathbf{IP}(q(n))$ se:
  - O número de mensagens trocadas entre P e V é $O(q(n))$.

Classe IP

- Seja $q(n)$ uma função de $\mathbb{N}$ em $\mathbb{N}$.
- Seja $n = |x|$.
- Dizemos que L pertence a $\mathbf{IP}(q(n))$ se:
  - O número de mensagens trocadas entre P e V é $O(q(n))$.
  - Se $x \in L$, existe um provador P que sempre convence V disso.

Classe IP

- Seja $q(n)$ uma função de $\mathbb{N}$ em $\mathbb{N}$.
- Seja $n = |x|$.
- Dizemos que L pertence a $\mathbf{IP}(q(n))$ se:
  - O número de mensagens trocadas entre P e V é $O(q(n))$.
  - Se $x \in L$, existe um provador P que sempre convence V disso.
  - Se $x \notin L$, verificador é enganado com probabilidade limitada por alguma constante em $(0, 1)$.

Classe IP

- Seja $q(n)$ uma função de $\mathbb{N}$ em $\mathbb{N}$.
- Seja $n = |x|$.
- Dizemos que L pertence a $\mathbf{IP}(q(n))$ se:
  - O número de mensagens trocadas entre P e V é $O(q(n))$.
  - Se $x \in L$, existe um provador P que sempre convence V disso.
  - Se $x \notin L$, verificador é enganado com probabilidade limitada por alguma constante em $(0, 1)$.
  - O provador não tem acesso à τ.

Classe IP

- Seja $q(n)$ uma função de $\mathbb{N}$ em $\mathbb{N}$.
- Seja $n = |x|$.
- Dizemos que L pertence a $\mathbf{IP}(q(n))$ se:
  - O número de mensagens trocadas entre P e V é $O(q(n))$.
  - Se $x \in L$, existe um provador P que sempre convence V disso.
  - Se $x \notin L$, verificador é enganado com probabilidade limitada por alguma constante em $(0, 1)$.
  - O provador não tem acesso à τ.
- A classe **IP** (Goldwasse, Micali e Rackoff) é definida da seguinte forma:

$$\mathbf{IP} = \bigcup_{k \geq 0} \mathbf{IP}(n^k)$$

Classe AM

- Removendo a última condição, temos a definção da classe **AM**$(q(n))$. A definição de **AM** (Babai) é análoga.
- Proibir o provador de olhar o conteúdo de τ não é algo tão importante quanto parece.
- Goldwasser e Sipser provaram a seguinte relação entre **IP(q(n))** e **AM(q(n))**:

Teorema

IP(q(n)) = **AM(q(n)+2)**.

Dúvidas?

coNP ⊆ IP

- Warm-up para o resultado principal.
- Resultado de Lund, Fortnow, Karloff e Nisan.
- Estratégia de prova: Mostrar que **N3C**, que é **coNP**-completo, está em **IP**.
- **N3C**: Dado um grafo G, decidir se $\chi(G) > 3$.

Aritmetização de instâncias

- Transformar a instância num polinômio de grau limitado.
- As mensagens enviadas pelo provador são esses polinômios.
- Algumas propriedades desses polinômios ajudam o verificador na tarefa de detectar afirmações erradas do provador.

Aritmetização do N3C

- Instância: Grafo G.
  Vamos assumir que $V(G) = [n]$, onde $n = |V(G)|$.

Aritmetização do N3C

- Instância: Grafo G.
  Vamos assumir que $V(G) = [n]$, onde $n = |V(G)|$.
- Bloco básico: $p(x) = \frac{5x^2}{4} + \frac{x^4}{4}$.
- $p(x) = 0$ se $x = 0$ e $p(x) = 1$ se $x = -2, -1, 1, 2$.
- Polinômio associado à instância:

$$q(X_1, \ldots, X_n) = \prod_{ij \in E(G)} p(X_i - X_j)$$

- X_i pertence a um corpo finito $F = \{0, 1, \ldots, N-1\}$.

Relação entre $q(x)$ e colorações de G

- O grau d de $q(X_1, \ldots, X_n)$ é limitado por $4|E(G)|$.

Relação entre $q(x)$ e colorações de G

- O grau d de $q(X_1, \ldots, X_n)$ é limitado por $4|E(G)|$.
- Se $X \in \{0, 1, 2\}^n$, X é uma 3-coloração de G.
- X é válida sse dois vizinhos não possuem a mesma cor.

Relação entre $q(x)$ e colorações de G

- O grau d de $q(X_1, \ldots, X_n)$ é limitado por $4|E(G)|$.
- Se $X \in \{0, 1, 2\}^n$, X é uma 3-coloração de G.
- X é válida sse dois vizinhos não possuem a mesma cor.
- Fatos essenciais:

Relação entre $q(x)$ e colorações de G

- O grau d de $q(X_1, \ldots, X_n)$ é limitado por $4|E(G)|$.
- Se $X \in \{0, 1, 2\}^n$, X é uma 3-coloração de G.
- X é válida sse dois vizinhos não possuem a mesma cor.
- Fatos essenciais:
  - Se X é válida, $q(x) = 1$.
  - Se X não é válida, $q(x) = 0$.

Relação entre $q(x)$ e colorações de G

- O grau d de $q(X_1, \ldots, X_n)$ é limitado por $4|E(G)|$.
- Se $X \in \{0,1,2\}^n$, X é uma 3-coloração de G.
- X é válida sse dois vizinhos não possuem a mesma cor.
- Fatos essenciais:
    - Se X é válida, $q(x) = 1$.
    - Se X não é válida, $q(x) = 0$.
- Pergunta: Quando G não admite uma 3-coloração válida?

Relação entre $q(x)$ e colorações de G

- O grau d de $q(X_1, \ldots, X_n)$ é limitado por $4|E(G)|$.
- Se $X \in \{0, 1, 2\}^n$, X é uma 3-coloração de G.
- X é válida sse dois vizinhos não possuem a mesma cor.
- Fatos essenciais:
  - Se X é válida, $q(x) = 1$.
  - Se X não é válida, $q(x) = 0$.
- Pergunta: Quando G não admite uma 3-coloração válida?
- Resposta: Quando o valor da expressão abaixo é zero:

$$q_0 = \sum_{x_1 \in \{0,1,2\}} \sum_{x_2 \in \{0,1,2\}} \cdots \sum_{x_n \in \{0,1,2\}} q(x_1, x_2, \ldots, x_n).$$

Papel do provador

- O número de termos da expressão é exponencial em n.
- Verificador não calcula q_0 de maneira trivial.
- Verificador quer decidir se q_0 é 0 ou 1.
- Para isso, ele interage com provador.
- Provador vai tentar convencê-lo que $q_0 = 0$.

Definições e propriedades envolvendo $q(x)$

- Notação mais sucinta:

$$q_i(X_1,\ldots,X_i) = \sum_{x_{i+1}\in\{0,1,2\}} \cdots \sum_{x_n\in\{0,1,2\}} q(X_1,\ldots,X_i,x_{i+1},\ldots,x_n).$$

Definições e propriedades envolvendo $q(x)$

- Notação mais sucinta:

$$q_i(X_1, \ldots, X_i) = \sum_{x_{i+1} \in \{0,1,2\}} \cdots \sum_{x_n \in \{0,1,2\}} q(X_1, \ldots, X_i, x_{i+1}, \ldots, x_n).$$

- Propriedade envolvendo tais polinômios:

$$q_{i-1}(X_1, \ldots, X_{i-1}) = \sum_{x_i \in \{0,1,2\}} q_i(X_1, \ldots, X_{i-1}, x_i)$$

Algoritmo do Provador

Cria o polinômio $\tilde{q}_0 = 0$
Escreve $\tilde{q}_0$ em F_P
Para i de 1 até n faça
    Escolhe o polinômio $\tilde{q}_i(X_i)$
    Escreve $\tilde{q}_i(X_i)$ em F_P
    Lê de F_V um inteiro ρ_i

Algoritmo do Verificador

Lê de F_P o polinômio $\tilde{q}_0$
Para i de 1 até n faça
  Lê de F_P o polinômio $\tilde{q}_i(X_i)$
  Se $\tilde{q}_i(X_i)$ tem grau maior que d ou $\tilde{q}_{i-1}(\rho_{i-1}) \neq \sum_{x \in \{0,1,2\}} \tilde{q}_i(x)$
    Rejeita a prova
  $\rho_i \leftarrow rand(F)$
  Escreve ρ_i em F_V
Se $\tilde{q}_n(\rho_n) = q(\rho_1, \ldots, \rho_n)$
  Aceita a prova
Senão rejeita a prova

Caso 1: $q_0 = 0$

- G não admite 3-coloração válida, e portanto o provador sempre deve ser capaz de convencer o verificador desse fato.

Caso 1: $q_0 = 0$

- G não admite 3-coloração válida, e portanto o provador sempre deve ser capaz de convencer o verificador desse fato.
- Estratégia do provador: $\tilde{q}_i(x) = q_i(\rho_1, \ldots, \rho_{i-1}, x)$.

Caso 1: $q_0 = 0$

- G não admite 3-coloração válida, e portanto o provador sempre deve ser capaz de convencer o verificador desse fato.
- Estratégia do provador: $\tilde{q}_i(x) = q_i(\rho_1, \ldots, \rho_{i-1}, x)$.
- Devido a essa propriedade, tais polinômios sempre passam pelo passo 4 do verificador.

Caso 1: $q_0 = 0$

- G não admite 3-coloração válida, e portanto o provador sempre deve ser capaz de convencer o verificador desse fato.
- Estratégia do provador: $\tilde{q}_i(x) = q_i(\rho_1, \ldots, \rho_{i-1}, x)$.
- Devido a essa propriedade, tais polinômios sempre passam pelo passo 4 do verificador.
- A igualdade do passo 8 do verificador será verdadeira.

Esclarecendo alguns pontos...

- Em cada iteração, o número de variáveis dos polinômios deveria aumentar.
- Verificador não lê mensagens exponenciais em n.
- Para cada variável inserida, um valor em F é sorteado.
- Ao checar essa propriedade, o verificador "força" o provador a ter uma certa consistência na escolha dos polinômios.

Caso 2: $q_0 \neq 0$

- É claro que $\tilde{q}_1 \neq q_1$, pois $\tilde{q}_1(0) + \tilde{q}_1(1) + \tilde{q}_1(2) = \tilde{q}_0 = 0$ e $q_1(0) + q_1(1) + q_1(2) = q_0 \neq 0$.

Caso 2: $q_0 \neq 0$

- É claro que $\tilde{q}_1 \neq q_1$, pois $\tilde{q}_1(0) + \tilde{q}_1(1) + \tilde{q}_1(2) = \tilde{q}_0 = 0$ e $q_1(0) + q_1(1) + q_1(2) = q_0 \neq 0$.
- Se $\tilde{q}_k(x) \neq q_k(\rho_1, \ldots, \rho_{i-1}, x)$, $1 \leq k \leq n$, só há aceitação se ocorrer igualdade no passo 8 do verificador.

Caso 2: $q_0 \neq 0$

- É claro que $\tilde{q}_1 \neq q_1$, pois $\tilde{q}_1(0) + \tilde{q}_1(1) + \tilde{q}_1(2) = \tilde{q}_0 = 0$ e $q_1(0) + q_1(1) + q_1(2) = q_0 \neq 0$.
- Se $\tilde{q}_k(x) \neq q_k(\rho_1, \ldots, \rho_{i-1}, x)$, $1 \leq k \leq n$, só há aceitação se ocorrer igualdade no passo 8 do verificador.
- Tal evento ocorre com probabilidade $\frac{d}{N}$.

Caso 2: $q_0 \neq 0$

- É claro que $\tilde{q}_1 \neq q_1$, pois $\tilde{q}_1(0) + \tilde{q}_1(1) + \tilde{q}_1(2) = \tilde{q}_0 = 0$ e $q_1(0) + q_1(1) + q_1(2) = q_0 \neq 0$.
- Se $\tilde{q}_k(x) \neq q_k(\rho_1, \ldots, \rho_{i-1}, x)$, $1 \leq k \leq n$, só há aceitação se ocorrer igualdade no passo 8 do verificador.
- Tal evento ocorre com probabilidade $\frac{d}{N}$.
- Limitar d e escolher adequadamente N é essencial.

Caso 2: A sorte sorri para o provador...

- Eventualmente, o provador é capaz de ludibriar o verificador.

Caso 2: A sorte sorri para o provador...

- Eventualmente, o provador é capaz de ludibriar o verificador.
- Se existe i tal que $\tilde{q}_{i-1}(\rho_{i-1}) = \sum_{x \in \{0,1,2\}} q_i(\rho_1, \ldots, \rho_{i-1}, x)$, bastará escolher $\tilde{q}_k(x) = q_k(\rho_1, \ldots, \rho_{k-1}, x)$, $i \leq k \leq n$.

Caso 2: A sorte sorri para o provador...

- Eventualmente, o provador é capaz de ludibriar o verificador.
- Se existe i tal que $\tilde{q}_{i-1}(\rho_{i-1}) = \sum_{x \in \{0,1,2\}} q_i(\rho_1, \ldots, \rho_{i-1}, x)$, bastará escolher $\tilde{q}_k(x) = q_k(\rho_1, \ldots, \rho_{k-1}, x)$, $i \leq k \leq n$.
- Pergunta: Por que o provador não escolhe um $\tilde{q}_{i-1}(x)$ tal que a igualdade acima sempre é verdadeira?

Caso 2: A sorte sorri para o provador...

- Eventualmente, o provador é capaz de ludibriar o verificador.
- Se existe i tal que $\tilde{q}_{i-1}(\rho_{i-1}) = \sum_{x \in \{0,1,2\}} q_i(\rho_1, \ldots, \rho_{i-1}, x)$, bastará escolher $\tilde{q}_k(x) = q_k(\rho_1, \ldots, \rho_{k-1}, x)$, $i \leq k \leq n$.
- Pergunta: Por que o provador não escolhe um $\tilde{q}_{i-1}(x)$ tal que a igualdade acima sempre é verdadeira?
- Resposta: A escolha de $\tilde{q}_{i-1}(x)$ pelo provador ocorre antes do sorteio de ρ_{i-1}. Logo, ele depende da sorte.

Caso 2: A sorte sorri para o provador...

- Eventualmente, o provador é capaz de ludibriar o verificador.
- Se existe i tal que $\tilde{q}_{i-1}(\rho_{i-1}) = \sum_{x \in \{0,1,2\}} q_i(\rho_1, \ldots, \rho_{i-1}, x)$, bastará escolher $\tilde{q}_k(x) = q_k(\rho_1, \ldots, \rho_{k-1}, x)$, $i \leq k \leq n$.
- Pergunta: Por que o provador não escolhe um $\tilde{q}_{i-1}(x)$ tal que a igualdade acima sempre é verdadeira?
- Resposta: A escolha de $\tilde{q}_{i-1}(x)$ pelo provador ocorre antes do sorteio de ρ_{i-1}. Logo, ele depende da sorte.
- Estratégia do provador: Escolher polinômios $\tilde{q}_i(x)$ capazes de passar pelos testes de consistência.

Conclusão

- O provador depende da sorte para ludibriar o verificador.
- Na iteração i, a probabilidade de o verificador sortear um ρ_i que ajuda o provador é $\left(1 - \frac{d}{N}\right)^{i-1} \frac{d}{N}$.
- Logo, a probabilidade de o verificador ser enganado é

$$\sum_{i=1}^{n+1} \left(1 - \frac{d}{N}\right)^{i-1} \frac{d}{N} \leq \frac{d(n+1)}{N}$$

- Como $d \leq 4|E(G)|$ e $n = |V(G)|$, basta escolher $N = 8n^3$ para que o limite acima seja $\frac{1}{4}$.

Dúvidas?

PSPACE = IP

- Resultado mais importante envolvendo a classe **IP**.
- Resultado obtido originalmente por Shamir.
- Estratégia de prova:
  - **IP** $\subseteq$ **PSPACE**: simulação do sistema interativo.
  - **PSPACE** $\subseteq$ **IP**: Mostrar que o problema **QBF**, que é **PSPACE**-completo, está em **IP**.

IP ⊆ PSPACE

- Se uma linguagem está em **IP**, então existe um verificador limitado por um polinômio $p(n)$.
- Além do consumo de tempo, $p(n)$ também limita

IP $\subseteq$ PSPACE

- Se uma linguagem está em **IP**, então existe um verificador limitado por um polinômio $p(n)$.
- Além do consumo de tempo, $p(n)$ também limita
  - o número de bits aleatórios utilizados pelo verificador.

IP $\subseteq$ PSPACE

- Se uma linguagem está em **IP**, então existe um verificador limitado por um polinômio $p(n)$.
- Além do consumo de tempo, $p(n)$ também limita
  - o número de bits aleatórios utilizados pelo verificador.
  - a soma dos comprimentos das mensagens trocadas.

IP ⊆ PSPACE

- Se uma linguagem está em **IP**, então existe um verificador limitado por um polinômio $p(n)$.
- Além do consumo de tempo, $p(n)$ também limita
  - o número de bits aleatórios utilizados pelo verificador.
  - a soma dos comprimentos das mensagens trocadas.
- $2^{p(n)}$ configurações possíveis de τ.

IP $\subseteq$ PSPACE

- Se uma linguagem está em **IP**, então existe um verificador limitado por um polinômio $p(n)$.
- Além do consumo de tempo, $p(n)$ também limita
  - o número de bits aleatórios utilizados pelo verificador.
  - a soma dos comprimentos das mensagens trocadas.
- $2^{p(n)}$ configurações possíveis de τ.
- Número de seqüências que contêm a concatenação das mensagens trocadas é limitado por $|\Sigma|^{p(n)}$.

IP ⊆ PSPACE

- Se uma linguagem está em **IP**, então existe um verificador limitado por um polinômio $p(n)$.
- Além do consumo de tempo, $p(n)$ também limita
  - o número de bits aleatórios utilizados pelo verificador.
  - a soma dos comprimentos das mensagens trocadas.
- $2^{p(n)}$ configurações possíveis de τ.
- Número de seqüências que contêm a concatenação das mensagens trocadas é limitado por $|\Sigma|^{p(n)}$.
- A simulação consome espaço polinomial.

QBF

- Sigla para *Quantified Boolean Formula*.
- Uma instância do **QBF** tem o formato $(Q_1x_1)(Q_2x_2)\ldots(Q_nx_n)B(x_1, x_2, \ldots, x_n)$. Ex:

$$(\exists x_1)(\forall x_2)(\exists x_3)((x_1 \vee \overline{x_2})(\overline{x_1} \vee x_2 \vee \overline{x_3}))$$

QBF

- Sigla para *Quantified Boolean Formula*.
- Uma instância do **QBF** tem o formato $(Q_1x_1)(Q_2x_2)\ldots(Q_nx_n)B(x_1, x_2, \ldots, x_n)$. Ex:

$$(\exists x_1)(\forall x_2)(\exists x_3)((x_1 \vee \overline{x_2})(\overline{x_1} \vee x_2 \vee \overline{x_3}))$$

- Cada Q_i representa $\exists$ ou $\forall$.
- $B(x_1, x_2, \ldots, x_n)$ é uma fórmula booleana na forma normal conjuntiva.
- Se a informação descrita por uma instância do **QBF** é verdadeira, dizemos que ela é um teorema.

Aritmetização do QBF

- São aplicadas as seguintes substituições em $B(x_1, \ldots, x_n)$:
  - $x \wedge y \rightarrow xy$.
  - $x \vee y \rightarrow x \star y = x + y - xy$.
  - $\overline{x} \rightarrow 1 - x$.

Aritmetização do QBF

- São aplicadas as seguintes substituições em $B(x_1, \ldots, x_n)$:
  - $x \wedge y \rightarrow xy$.
  - $x \vee y \rightarrow x \star y = x + y - xy$.
  - $\overline{x} \rightarrow 1 - x$.
- Exemplo: $B(x_1, x_2, x_3) = x_1 \wedge (\overline{x}_2 \vee x_3) \wedge (x_2 \vee \overline{x}_3)$:

$$P_0(x_1, x_2, x_3) \quad = \quad x_1((1 - x_2) \star x_3)(x_2 \star (1 - x_3))$$

Aritmetização do QBF

- São aplicadas as seguintes substituições em $B(x_1, \ldots, x_n)$:
  - $x \wedge y \rightarrow xy$.
  - $x \vee y \rightarrow x \star y = x + y - xy$.
  - $\overline{x} \rightarrow 1 - x$.
- Exemplo: $B(x_1, x_2, x_3) = x_1 \wedge (\overline{x}_2 \vee x_3) \wedge (x_2 \vee \overline{x}_3)$:

$$P_0(x_1, x_2, x_3) \quad = \quad x_1((1 - x_2) \star x_3)(x_2 \star (1 - x_3))$$

- Trocando verdadeiro por 1 e falso por 0, temos:
  - Se a fórmula é verdadeira, polinômio vale 1.
  - Se a fórmula é falsa, polinômio vale 0.

Avaliando uma instância do QBF

- Para avaliar uma instância do **QBF**, utilizamos os quantificadores.

Avaliando uma instância do QBF

- Para avaliar uma instância do **QBF**, utilizamos os quantificadores.
- Transformações definidas para um polinômio:

$$
\begin{aligned}
(\forall x_i P)(x_1, \ldots, x_i) &= P(x_1, \ldots, x_{i-1}, 0) P(x_1, \ldots, x_k, 1), \\
(\exists x_i P)(x_1, \ldots, x_i) &= P(x_1, \ldots, x_{i-1}, 0) \star P(x_1, \ldots, x_k, 1), \\
(R x_i P)(x_1, \ldots, x_i) &= P(x_1, \ldots, x_i) \mod (x_i^2 - x_i).
\end{aligned}
$$

Avaliando uma instância do QBF

- Para avaliar uma instância do **QBF**, utilizamos os quantificadores.
- Transformações definidas para um polinômio:

$$
\begin{aligned}
(\forall x_i P)(x_1,\ldots,x_i) &= P(x_1,\ldots,x_{i-1},0)P(x_1,\ldots,x_k,1),\\
(\exists x_i P)(x_1,\ldots,x_i) &= P(x_1,\ldots,x_{i-1},0) \star P(x_1,\ldots,x_k,1),\\
(R x_i P)(x_1,\ldots,x_i) &= P(x_1,\ldots,x_i) \mod (x_i^2 - x_i).
\end{aligned}
$$

- As duas primeiras envolvem a aplicação dos quantificadores.

Avaliando uma instância do QBF

- Para avaliar uma instância do **QBF**, utilizamos os quantificadores.
- Transformações definidas para um polinômio:

$$
\begin{aligned}
(\forall x_i P)(x_1, \ldots, x_i) &= P(x_1, \ldots, x_{i-1}, 0) P(x_1, \ldots, x_k, 1), \\
(\exists x_i P)(x_1, \ldots, x_i) &= P(x_1, \ldots, x_{i-1}, 0) \star P(x_1, \ldots, x_k, 1), \\
(R x_i P)(x_1, \ldots, x_i) &= P(x_1, \ldots, x_i) \mod (x_i^2 - x_i).
\end{aligned}
$$

- As duas primeiras envolvem a aplicação dos quantificadores.
- A terceira transforma potências x_i^t em x_i.

Avaliando uma instância do QBF(cont.)

- Utilizando as transformações acima, podemos avaliar uma instância ϕ de **QBF**.

Avaliando uma instância do QBF(cont.)

- Utilizando as transformações acima, podemos avaliar uma instância ϕ de **QBF**.
- Seqüência de instruções para avaliação de ϕ:

$$
\begin{array}{l}
Rx_1, Rx_2, \ldots, Rx_n \\
Q_n x_n, \\
Rx_1, Rx_2, \ldots, Rx_{n-1} \\
Q_{n-1} x_{n-1}, \\
\ldots \\
Rx_1, \\
Q_1 x_1.
\end{array}
$$

Avaliando uma instância do QBF(cont.)

- Utilizando as transformações acima, podemos avaliar uma instância ϕ de **QBF**.
- Seqüência de instruções para avaliação de ϕ:

$$
\begin{aligned}
&Rx_1, Rx_2, \ldots, Rx_n\\
&Q_n x_n,\\
&Rx_1, Rx_2, \ldots, Rx_{n-1}\\
&Q_{n-1} x_{n-1},\\
&\ldots\\
&Rx_1,\\
&Q_1 x_1.
\end{aligned}
$$

- Aplicando as transformações, obtemos os polinômios $p_1, \ldots, p_k$, onde $k = \frac{n^2+3n}{2}$.
- Se ϕ é teorema, $p_k = 1$. Senão, $p_k = 0$.

Aplicando as tranformações

- Seja $P_1(x_1, x_2, x_3) = x_1 - x_1x_2 - x_1x_3 + 2x_1x_2x_3$.

Aplicando as tranformações

- Seja $P_1(x_1, x_2, x_3) = x_1 - x_1x_2 - x_1x_3 + 2x_1x_2x_3$.
- Aplicando $(\exists x_3)$, temos:

$$\begin{aligned} P_2(x_1, x_2) &= (x_1 - x_1x_2 - x_1 1 + 2x_1x_2 1) + \\ &\quad (x_1 - x_1x_2 - x_1 0 + 2x_1x_2 0) - \\ &\quad (x_1 - x_1x_2 - x_1 1 + 2x_1x_2 1) \\ &\quad (x_1 - x_1x_2 - x_1 0 + 2x_1x_2 0) \\ &= x_1 - x_1^2x_2 + x_1^2x_2^2. \end{aligned}$$

Aplicando as tranformações

- Seja $P_1(x_1, x_2, x_3) = x_1 - x_1x_2 - x_1x_3 + 2x_1x_2x_3$.
- Aplicando $(\exists x_3)$, temos:

$$\begin{aligned} P_2(x_1, x_2) &= (x_1 - x_1x_2 - x_1 1 + 2x_1x_2 1) + \\ &\quad (x_1 - x_1x_2 - x_1 0 + 2x_1x_2 0) - \\ &\quad (x_1 - x_1x_2 - x_1 1 + 2x_1x_2 1) \\ &\quad (x_1 - x_1x_2 - x_1 0 + 2x_1x_2 0) \\ &= x_1 - x_1^2x_2 + x_1^2x_2^2. \end{aligned}$$

- Aplicando (Rx_1) e (Rx_2), temos:

$$P_4(x_1, x_2) = x_1 - x_1x_2 + x_1x_2 = x_1.$$

Aplicando as tranformações(cont.)

- Aplicando $(\forall x_2)$, temos:

$$P_5(x_1) \quad = \quad (x_1)(x_1) = x_1^2.$$

Aplicando as tranformações(cont.)

- Aplicando $(\forall x_2)$, temos:

$$P_5(x_1) = (x_1)(x_1) = x_1^2.$$

- Aplicando (Rx_1), temos $P_6(x_1) = x_1$.

Aplicando as tranformações(cont.)

- Aplicando $(\forall x_2)$, temos:

$$P_5(x_1) = (x_1)(x_1) = x_1^2.$$

- Aplicando (Rx_1), temos $P_6(x_1) = x_1$.
- Se $Q_1 x_1$ é $\forall x_1$, a instância é falsa.
- Se $Q_1 x_1$ é $\exists x_1$, a instância é um teorema.

Importância do provador

- O número de termos pode dobrar após cada transformação.
- Uma avaliação trivial pelo verificador consome tempo $\mathcal{O}(2^n)$.
- Para estimar o valor, o verificador interage com o provador.
- O provador pode tentar convencê-lo que ϕ é um teorema mesmo quando isso não é verdade.

Resumo da interação

- Provador envia uma seqüência $\tilde{p}_k, \tilde{p}_{k-1}, \ldots, \tilde{p}_1$ de polinômios.
- Supostamente, tais polinômios são esses aqui.

Resumo da interação

- Provador envia uma seqüência $\tilde{p}_k, \tilde{p}_{k-1}, \ldots, \tilde{p}_1$ de polinômios.
- Supostamente, tais polinômios são esses aqui.
- Polinômios enviados na ordem inversa. Ou seja, o número de variáveis deveria crescer conforme as iterações vão passando.
- Mas o verificador é limitado polinomialmente.

Resumo da interação

- Provador envia uma seqüência $\tilde{p}_k, \tilde{p}_{k-1}, \ldots, \tilde{p}_1$ de polinômios.
- Supostamente, tais polinômios são esses aqui.
- Polinômios enviados na ordem inversa. Ou seja, o número de variáveis deveria crescer conforme as iterações vão passando.
- Mas o verificador é limitado polinomialmente.
- Verificador sorteia valores em F para tais variáveis.
- Verificador checa a consistência dos polinômios que recebe.

Trabalho do verificador

- A checagem do verificador depende da transformação.
- Cada operação é realizada em função de uma variável.
- Após cada operação, o verificador sorteia um valor para essa variável, que deve ser enviado para o provador.
- Para checar a consistência, o verificador guarda:

Trabalho do verificador

- A checagem do verificador depende da transformação.
- Cada operação é realizada em função de uma variável.
- Após cada operação, o verificador sorteia um valor para essa variável, que deve ser enviado para o provador.
- Para checar a consistência, o verificador guarda:
  - O último polinômio enviado pelo provador.
  - O valor do penúltimo polinômio aplicado no valor sorteado.

Controle de tempo

- Pergunta: Que horas são?

Controle de tempo

- Pergunta: Que horas são?
- Se já estiver tarde, pular....

Exemplo: Teste para quantificador ∃

- Descrição da primeira interação:

Exemplo: Teste para quantificador $\exists$

- Descrição da primeira interação:
  - Provador **escreve** $\tilde{p}_6(x_1) = x_1$ em F_P.

Exemplo: Teste para quantificador $\exists$

- Descrição da primeira interação:
  - Provador **escreve** $\tilde{p}_6(x_1) = x_1$ em F_P.
  - Verificador checa que $(\exists x_1 \tilde{p}_6) = 0 + 1 - 0.1 = e = 1$.

Exemplo: Teste para quantificador $\exists$

- Descrição da primeira interação:
  - Provador **escreve** $\tilde{p}_6(x_1) = x_1$ em F_P.
  - Verificador checa que $(\exists x_1 \tilde{p}_6) = 0 + 1 - 0.1 = e = 1$.
  - Verificador sorteia $c_1 = rand(F)$.

Exemplo: Teste para quantificador $\exists$

- Descrição da primeira interação:
  - Provador **escreve** $\tilde{p}_6(x_1) = x_1$ em F_P.
  - Verificador checa que $(\exists x_1 \tilde{p}_6) = 0 + 1 - 0.1 = e = 1$.
  - Verificador sorteia $c_1 = rand(F)$.
  - Verificador altera e para $\tilde{p}_6(c_1)$.

Exemplo: Teste para quantificador $\exists$

- Descrição da primeira interação:
  - Provador **escreve** $\tilde{p}_6(x_1) = x_1$ em F_P.
  - Verificador checa que $(\exists x_1 \tilde{p}_6) = 0 + 1 - 0.1 = e = 1$.
  - Verificador sorteia $c_1 = rand(F)$.
  - Verificador altera e para $\tilde{p}_6(c_1)$.
  - Verificador **escreve** c_1 em F_V.

Exemplo: Teste para quantificador $\exists$

- Descrição da primeira interação:
  - Provador **escreve** $\tilde{p}_6(x_1) = x_1$ em F_P.
  - Verificador checa que $(\exists x_1 \tilde{p}_6) = 0 + 1 - 0.1 = e = 1$.
  - Verificador sorteia $c_1 = rand(F)$.
  - Verificador altera e para $\tilde{p}_6(c_1)$.
  - Verificador **escreve** c_1 em F_V.
- Antes da primeira interação, $e = 1$ (verificador começa acreditando no provador...).

Exemplo: Teste para operação de aumento de grau

- Descrição da segunda interação:

Exemplo: Teste para operação de aumento de grau

- Descrição da segunda interação:
  - Provador **escreve** $\tilde{p}_5(x_1) = x_1^2$ em F_P.

Exemplo: Teste para operação de aumento de grau

- Descrição da segunda interação:
  - Provador **escreve** $\tilde{p}_5(x_1) = x_1^2$ em F_P.
  - Verificador checa que $(Rx_1\tilde{p}_5) = 0 + (1-0)c_1 = e$.

Exemplo: Teste para operação de aumento de grau

- Descrição da segunda interação:
  - Provador **escreve** $\tilde{p}_5(x_1) = x_1^2$ em F_P.
  - Verificador checa que $(Rx_1\tilde{p}_5) = 0 + (1-0)c_1 = e$.
  - Verificador sorteia $c_1 = rand(F)$.

Exemplo: Teste para operação de aumento de grau

- Descrição da segunda interação:
  - Provador **escreve** $\tilde{p}_5(x_1) = x_1^2$ em F_P.
  - Verificador checa que $(Rx_1\tilde{p}_5) = 0 + (1-0)c_1 = e$.
  - Verificador sorteia $c_1 = rand(F)$.
  - Verificador altera e para $\tilde{p}_5(c_1)$.

Exemplo: Teste para operação de aumento de grau

- Descrição da segunda interação:
  - Provador **escreve** $\tilde{p}_5(x_1) = x_1^2$ em F_P.
  - Verificador checa que $(Rx_1\tilde{p}_5) = 0 + (1-0)c_1 = e$.
  - Verificador sorteia $c_1 = rand(F)$.
  - Verificador altera e para $\tilde{p}_5(c_1)$.
  - Verificador **escreve** c_1 em F_V.

Exemplo: Teste para operação de aumento de grau

- Descrição da segunda interação:
  - Provador **escreve** $\tilde{p}_5(x_1) = x_1^2$ em F_P.
  - Verificador checa que $(Rx_1\tilde{p}_5) = 0 + (1-0)c_1 = e$.
  - Verificador sorteia $c_1 = rand(F)$.
  - Verificador altera e para $\tilde{p}_5(c_1)$.
  - Verificador **escreve** c_1 em F_V.
- Novamente, a transformação ocorreu em função de x_1.

Exemplo: Teste para operação de aumento de grau

- Descrição da segunda interação:
  - Provador **escreve** $\tilde{p}_5(x_1) = x_1^2$ em F_P.
  - Verificador checa que $(Rx_1\tilde{p}_5) = 0 + (1-0)c_1 = e$.
  - Verificador sorteia $c_1 = rand(F)$.
  - Verificador altera e para $\tilde{p}_5(c_1)$.
  - Verificador **escreve** c_1 em F_V.
- Novamente, a transformação ocorreu em função de x_1.
- Novamente, foi sorteado um valor para x_1.

Exemplo: Teste para operação $\forall$

- Descrição da terceira interação:

Exemplo: Teste para operação $\forall$

- Descrição da terceira interação:
  - Provador **escreve** $\tilde{p}_4(x_2) = c_1$ em F_P.

Exemplo: Teste para operação $\forall$

- Descrição da terceira interação:
  - Provador **escreve** $\tilde{p}_4(x_2) = c_1$ em F_P.
  - Verificador checa que $(\forall x_2 \tilde{p}_5)(c_1) = c_1 c_1 = e$.

Exemplo: Teste para operação $\forall$

- Descrição da terceira interação:
  - Provador **escreve** $\tilde{p}_4(x_2) = c_1$ em F_P.
  - Verificador checa que $(\forall x_2 \tilde{p}_5)(c_1) = c_1 c_1 = e$.
  - Verificador sorteia $c_2 = rand(F)$.

Exemplo: Teste para operação $\forall$

- Descrição da terceira interação:
  - Provador **escreve** $\tilde{p}_4(x_2) = c_1$ em F_P.
  - Verificador checa que $(\forall x_2 \tilde{p}_5)(c_1) = c_1 c_1 = e$.
  - Verificador sorteia $c_2 = rand(F)$.
  - Verificador altera e para $\tilde{p}_4(c_2)$.

Exemplo: Teste para operação $\forall$

- Descrição da terceira interação:
  - Provador **escreve** $\tilde{p}_4(x_2) = c_1$ em F_P.
  - Verificador checa que $(\forall x_2 \tilde{p}_5)(c_1) = c_1 c_1 = e$.
  - Verificador sorteia $c_2 = rand(F)$.
  - Verificador altera e para $\tilde{p}_4(c_2)$.
  - Verificador **escreve** c_2 em F_V.

Exemplo: Teste para operação $\forall$

- Descrição da terceira interação:
  - Provador **escreve** $\tilde{p}_4(x_2) = c_1$ em F_P.
  - Verificador checa que $(\forall x_2 \tilde{p}_5)(c_1) = c_1 c_1 = e$.
  - Verificador sorteia $c_2 = rand(F)$.
  - Verificador altera e para $\tilde{p}_4(c_2)$.
  - Verificador **escreve** c_2 em F_V.
- Uma nova variável foi introduzida: x_2.

Exemplo: Teste para operação $\forall$

- Descrição da terceira interação:
  - Provador **escreve** $\tilde{p}_4(x_2) = c_1$ em F_P.
  - Verificador checa que $(\forall x_2 \tilde{p}_5)(c_1) = c_1 c_1 = e$.
  - Verificador sorteia $c_2 = rand(F)$.
  - Verificador altera e para $\tilde{p}_4(c_2)$.
  - Verificador **escreve** c_2 em F_V.
- Uma nova variável foi introduzida: x_2.
- A variável x_1 supostamente foi substituída por c_1 em $\tilde{p}_4$.

Estratégias do provador

- Se a instância do **QBF** é um teorema, enviar $\tilde{p}_i(x_j) = p_i(c_1, c_2, \ldots, c_{j-1}, x_j)$.

Estratégias do provador

- Se a instância do **QBF** é um teorema, enviar $\tilde{p}_i(x_j) = p_i(c_1, c_2, \ldots, c_{j-1}, x_j)$.
- Senão, enviar polinômios que passam nos testes de consistência.

Estratégias do provador

- Se a instância do **QBF** é um teorema, enviar $\tilde{p}_i(x_j) = p_i(c_1, c_2, \ldots, c_{j-1}, x_j)$.
- Senão, enviar polinômios que passam nos testes de consistência.
- Se o verificador sortear um valor c_j tal que $\tilde{p}_i(c_j) = p_i(c_1, c_2, \ldots, c_{j-1}, c_j)$, o provador será capaz de ludibriá-lo.

Chances do provador

- A probabilidade de o verificador sortear um valor que vai atrapalhá-lo na i-ésima iteração é $\left(1 - \frac{d}{|F|}\right)^{i-1} \frac{d}{|F|}$.

Chances do provador

- A probabilidade de o verificador sortear um valor que vai atrapalhá-lo na i-ésima iteração é $\left(1 - \frac{d}{|F|}\right)^{i-1} \frac{d}{|F|}$.
- Como o número de transformações é $O(n^2)$, a probabilidade de o verificador ser ludibriado é de:

$$\sum_{i=1}^{O(n^2)} \left(1 - \frac{d}{|F|}\right)^{i-1} \frac{d}{|F|} \leq \frac{O(dn^2)}{|F|}$$

Chances do provador

- A probabilidade de o verificador sortear um valor que vai atrapalhá-lo na i-ésima iteração é $\left(1 - \frac{d}{|F|}\right)^{i-1} \frac{d}{|F|}$.
- Como o número de transformações é $O(n^2)$, a probabilidade de o verificador ser ludibriado é de:

$$\sum_{i=1}^{O(n^2)} \left(1 - \frac{d}{|F|}\right)^{i-1} \frac{d}{|F|} \leq \frac{O(dn^2)}{|F|}$$

- Se $|F| = 4n^3$, a probabilidade de o verificador aceitar incorretamente o argumento do provador é limitada superiormente por $\frac{1}{4}$.

Dúvidas?

[1] L. Babai.
Trading group theory for randomness.
Proc. 17th Ann. ACM Symp. on Theory of Computing., 41(1):421–429, 1994.

[2] S. Goldwasser, S. Micali, and C. Rackoff.
The knowledge complexity of interactive proof-systems.
Proc. 17th Ann. ACM Symp. on Theory of Computing, 291–304, 1985.

[3] S. Goldwasser, and M. Sipser.
Private coins versus public coins in interactive proof-systems.
Proc. 18th Ann. ACM Symp. on Theory of Computing, 291–304, 1986.

[4] R. M. Karp.
On the computational complexity of combinatorial problems.
Networks, 5(1):45–68, 1975.

[5] C. Lund, L. Fortnow, H. Karloff and N. Nisan.
Algebraic methods for interactive proof systems.
Journal of the ACM, 39:859–868, 1992.

[6] A. Shamir.
IP = PSPACE.
Journal of the ACM, 39:869–877, 1992.