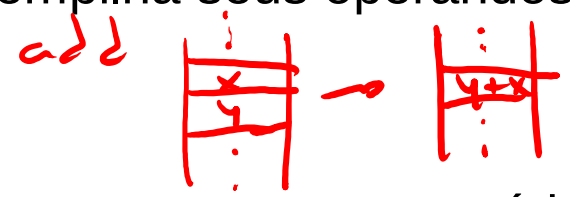
Compiladores – Geração de Código

Fabio Mascarenhas – 2015.2

http://www.dcc.ufrj.br/~fabiom/comp

Máquinas de Pilha

- Uma máquina de pilha é um tipo de processador em que todos os valores temporários são armazenados em uma pilha
  - Não são usados registradores
- Toda operação em uma máquina de pilha desempilha seus operandos, faz a operação e empilha o resultado



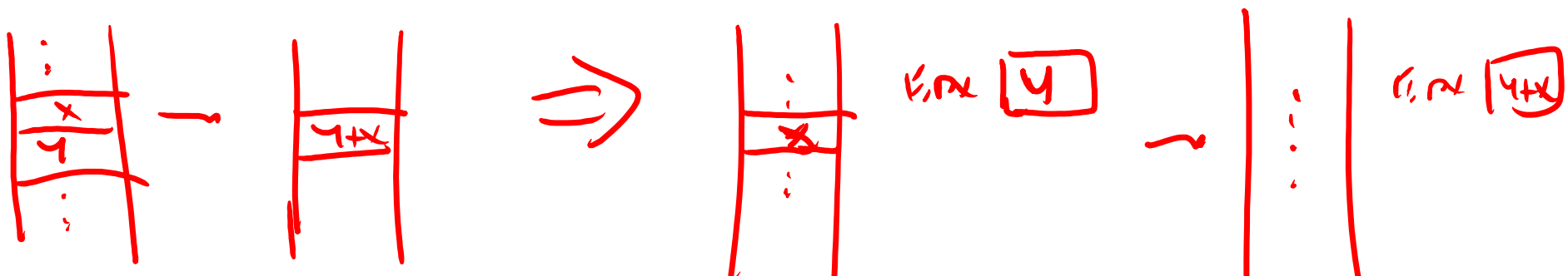
- Instruções também podem empilhar valores constantes, ou o conteúdo de variáveis locais e endereços da memória (variáveis globais)
- Compilar para máquinas de pilha é bem fácil, mas menos eficiente que usar registradores

Pilha + Acumulador

- Uma otimização da máquina de pilha é manter o topo da pilha sempre em um registrador, o *acumulador*

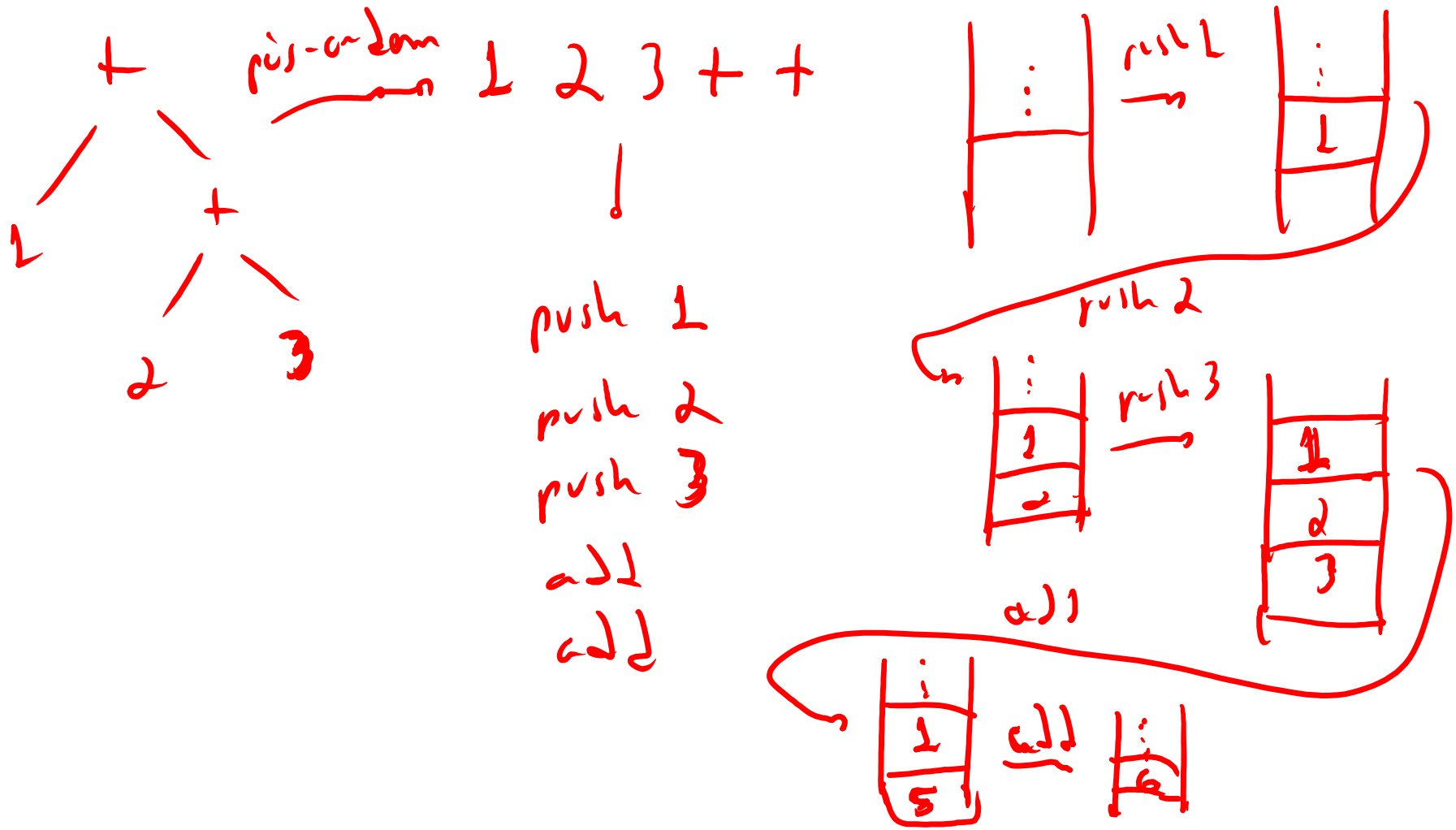
EAX

- Algumas operações da máquina ficam mais eficientes, mas podemos usar as mesmas operações de uma máquina de pilha comum



Compilando expressões

- Para ter uma intuição de como a geração de código funciona para uma máquina de pilha, vamos gerar código para `1 + (2 + 3)`:



Geração de Código para TINY em x86

- Vamos usar um modelo de máquina de pilha para gerar código para TINY com procedimentos para x86

- As instruções de nossa máquina de pilha serão implementadas por instruções de x86, usando o registrador EAX como acumulador e a pilha do processador como o resto da pilha

- Para simplificar, vamos tratar apenas de variáveis inteiras e booleanos

- Nossa máquina de pilha terá 15 instruções: getglobal, putglobal, icload, iload, istore, iadd, isub, imul, idiv, invoke, if_icmpneq, if_icmpgeq, jmp, read, write, pop

- A organização e nomes lembram os de máquinas virtuais de pilha, como a JVM

Contexto de Geração de Código

- Vamos criar uma classe para ser o *contexto* de geração de código
- O contexto implementa as instruções da máquina de pilha, gerando código x86 para elas em um buffer
- Vamos usar um contexto para cada procedimento, e depois costurar o código dos procedimentos junto com o código do corpo principal do programa e o código que declara variáveis globais
- Ele gerencia também os *labels* do programa, usados nas instruções de salto
- Os métodos de geração de código da AST só vão precisar de preocupar em chamar os métodos do contexto que correspondem às instruções da máquina