

Checkpoint 3 de Compiladores

Gabriel Hendrix
Franco Schmidt Rossi
Vinius de Oliveira Risso

29 de setembro de 2021

Perguntas específicas não foram definidas no enunciado para o *Checkpoint 3*, como foram no *Checkpoint 1*. Portanto, as perguntas do enunciado do *Checkpoint 1* serão respondidas aqui, de forma equivalente, referentes aos esforços e resultados do desenvolvimento do *Checkpoint 3*.

- **Instruções de compilação e teste:** Na pasta raiz, o comando `sudo make && sudo chmod x comptests.sh && sudo ./comptests.sh` gera o executável `trabalho` (nosso compilador) e, esse mesmo comando, gera todos os arquivos `.asm`, a partir dos exemplos fornecidos no diretório `tests3/dev`. O arquivo `comptests.sh` lê todos os arquivos dentro de cada pasta filha da pasta `tests3`, no caso `tests3/dev`, e gera arquivos nos formatos `asm` e `dot` dentro dos diretórios `asms` e `dots`, respectivamente. O comando `make clean` limpa os arquivos gerados na compilação do compilador. Para executar os arquivos `.asm` que foram gerados no diretório `asms`, baixe o simulador através deste link: <http://courses.missouristate.edu/kenvollmar/mars/download.htm>.
- **O compilador está completo?** Não. Existem requisitos mínimos faltando, especificamente:
 - O tipo básico `bool` é substituído por `int`, de acordo com a especificação da linguagem;
 - A declaração e execução correta de chamadas de funções não foi implementada;
 - O sistema de tipos não trata adequadamente de todos os tipos permitidos, como por exemplo qualificadores de tipos (`const`, `extern`, etc) e declaração de ponteiros (exceto como vetores);

Como a lista acima é relativamente extensa, é válido listar quais requisitos mínimos *funcionam*:

- Operações aritméticas e de comparação básicas;
- Comandos de atribuição;
- Uma estrutura de escolha (`if` e `else`), e uma estrutura de repetição (`while`);
- Declaração e manipulação de tipos básicos;
- Declaração e manipulação de um tipo composto, no caso vetores.
- As operações de IO básicas (`printf` e `scanf` em C) foram feitas de forma desleixada. Existem restrições para o uso dessas duas operações, sendo elas:
 - * Só é possível ler ou imprimir uma variável por vez
 - * Não é possível imprimir de maneira composta como por exemplo `printf("Alguma coisa %d outra coisa", variavel);`
 - * Só é possível executar escrita ou leitura sobre um valor, de um array, passando um número para o índice e não uma variável. Exemplo: `printf("%d", vetor[2]);` e não `printf("%d", vetor[i]);`
- **O analisador semântico está funcionando?** Não completamente. O analisador semântico gera a AST do código dado, respeitando as restrições ditas no tópico acima, mas ele faz poucas análises semânticas propriamente ditas do código. Dois exemplos do que funciona são: ele impede que uma variável inteira receba um valor real, e ele quebra ao encontrar uma chamada de variável que não tenha sido declarada; mas a implementação de regras semânticas foi muito baseada nos laboratórios dados, e não é de forma alguma extensiva o suficiente.
- **Quais foram as principais dificuldades encontradas até aqui?** A falta de uma documentação completa, a interferência de outras atividades de matérias e empregos de todos os participantes, e a dificuldade/complexidade da LP escolhida.

- **Como essas dificuldades foram resolvidas?** A falta de documentação foi resolvida fazendo este *Checkpoint* usando como base documentações do MIPS usado pelo MARS, para que fosse possível facilitar o processo de geração de código do compilador. As ocupações, referente aos empregos, dos integrantes não foram resolvidas, o que levou impediu que o trabalho ficasse melhor. A dificuldade de resolver dúvidas foram resolvidas através testes, contornos e também de pesquisas em fóruns e livros.
- **Há algum caso de teste que vocês gostariam de chamar atenção?** O caso de teste `tests3/dev/019.c` demonstra uma limitação, importante a ser ressaltada, do `printf`. Essa limitação é descrita no tópico 2.
- **Vocês conseguiram tratar a linguagem fonte toda?** Não. Apenas alguns dos requisitos mínimos foram cumpridos.
- **Quais foram as simplificações necessárias?** Todas as permitidas, e um pouco mais. O segundo tópico deste relatório entra em detalhe sobre o que foi feito e o que não foi.
- **Observações sobre os testes:** É possível notar que, ao se tentar usar o comando `printf` antes de qualquer `scanf`, acaba por gerar um problemas em todos os `scanf`'s subsequentes. Este erro não foi corrigido, porém foi verificada a sua existência.