

5 Conclusão

Neste trabalho estudamos dois métodos que podem ser utilizados para compactar provas lógicas. Em suma, o trabalho consistiu em apresentar dois métodos de compactação para provas proposicionais, que podem ser utilizados durante a construção da prova, para dois casos específicos.

O primeiro método apresentado, que se aplica a provas que possuem ramos semelhantes, consegue reduzir o tamanho da prova quando elimina das provas estes ramos semelhantes. A semelhança que tratamos nesta tese envolve basicamente renomeamento de variáveis. Este método corresponde a introdução de fórmulas máximas na prova, e tem garantida a diminuição de tamanho da prova quando a prova é representada como um grafo acíclico direto conforme mostrado na seção 3.3.2, em dedução natural a redução é alcançada quando a prova é representada usando o estilo devido a Fitch.

O segundo método apresentado, que se aplica em provas com \perp_c encadeados, reduz o tamanho da prova empregando axiomas de extensão.

As principais contribuições do trabalho são:

- Apresentamos métodos para reduções de provas proposicionais.

Sabemos que provas curtas são importantes tanto para Ciência da Computação quanto para Teoria da Prova e muito pouco se sabe sobre como obtê-las e o nosso trabalho aponta alternativas neste sentido.

- Os métodos apresentados, são implementáveis e podem ser utilizados durante a execução de um provador de teoremas sem a intervenção do usuário.

Os provadores automáticos de teoremas normalmente são feitos para realizar provas normais (ou livres da regra do corte), pois fazem uso do princípio de subfórmulas. Para permitir a inserção de cortes, o implementador deve considerar uma grande quantidade de alternativas em cada passo da prova o que torna a implementação pouco eficiente.

Extensões naturais deste trabalho incluem:

- Um refinamento do método horizontal no que se refere a busca por fórmulas semelhantes.

Isto implica o desenvolvimento de um algoritmo de busca em grafo acíclicos de provas. Na apresentação do método consideramos não-determinística a escolha de fórmulas possivelmente semelhantes que seriam aplicadas ao algoritmo 1. O não-determinismo na escolha não nos permite eliminar completamente a intervenção humana durante a construção da prova.

- A elaboração dos métodos para lógica de predicados.

Os resultados apresentados para SEQ_0 do método horizontal são naturalmente estendidos para lógica de predicados SEQ . Como a grande diferença de complexidade entre eliminação de corte proposicional e eliminação de corte em lógica de predicados é devida a repetições "desnecessárias" de instâncias idênticas da regra do \forall . Tais instâncias são unificáveis por substituições apropriadas, e assim equivalentes modulo WS . Portanto, usar $SEQ + WS$ pode reduzir sensivelmente o peso da regra de eliminação do corte.

- A implementação de um provador automático de teoremas que utilize os métodos apresentados. Para comprovar o que foi apresentado teoricamente.