

Kamil Breczko

# Instrukcja - Proof Checker

22 stycznia 2017

## 1. Kompilacja projektu

Aby móc prawidłowo skompilować projekt należy posiadać pliki:

- *dataTypes.ml*;
- *expression.ml*;
- *main.ml*;
- *lexer.mll*;
- *parser.mly*;
- *proofChecker.ml*;
- *verification.ml*;
- Reguły wnioskowania w katalogu */rules*:
  - *eliminationRules.ml*;
  - *introductionRules.ml*;
  - *predicateLogicRules.ml*;

W celu zbudowania projektu należy wykonać poniższą komendę w katalogu, w którym znajdują się wyżej wymienione pliki.

```
ocamlbuild -I rules -use-menhir -tag thread -use-ocamlfind -quiet -pkg core main.native
```

## 2. Testowanie

Program wczytuje dane z pliku tekstowego, w którym znajduje się ciąg dowodów. Aby sprawdzić poprawność dowodu, należy dostarczyć plik z zapisanym dowodem do programu *main.native*.

```
./main.native test.txt
```

Tak uruchomiony program na wyjściu zwraca informację o poprawności dowodu wraz z komunikatami o błędach. Aby oddzielić strumień wyjścia od strumienia błędów należy wykonać podaną komendę:

```
./main.native test.tx » log_file 2» err_file
```

## 3. Format dowodu

Formułę dowodu można zapisać na dwa sposoby. W pierwszy rozpoczynając od słowa kluczowego **goal** oraz podanie identyfikatora i formuły, która zostanie udowodniona między słowami kluczowymi **proof** i **end**. Jeśli dowód zostanie zapisany w podanym formacie i jest poprawny, można wykorzystać podaną formułę w innych dowodach bez konieczności ponownego dowodzenia. Zapis dowodu w pierwszym formacie:

```
goal nazwa_dowodu: formuła atomowa
proof
dowód
end.
```

Drugi sposób polega na zapisanie samego dowodu między słowami kluczowymi **proof** i **end**, bez nadawania identyfikatora. Użycie podanego formatu uniemożliwia korzystanie z udowodnionej formuły w innych dowodach. Zapis dowodu w drugim formacie:

```
proof
dowód
end.
```

## 4. Elementy dowodu

Dowód zapisywany jest w notacji liniowej. Taka reprezentacja dowodu składa się z elementów rozdzielonych średnikiem. Każdy element dowodu jest albo pojedynczą formułą albo ramką. Wystąpienie pojedynczej formuły w dowodzie jest poprawne, jeśli jest konkluzją jednej z reguł wnioskowania, której wszystkie przesłanki są spełnione. Jeśli program napotka formułę, której nie da się wywieść w ten sposób, dowód uznany jest za niepoprawny.

Zapis ramki z użyciem świeżej zmiennej w rachunku kwantyfikatorów:  
**[[x]: dowód ];**

Zapis ramki z użyciem świeżej zmiennej oraz założenia w rachunku kwantyfikatorów:  
**[[x], założenie: dowód ];**

Zapis ramki z użyciem świeżej zmiennej w rachunku kwantyfikatorów wraz z określeniem typu zmiennej:  
**[[x: Typ]: dowód ];**

## 5. Aksjomaty i określenie typu

W pliku z dowodem można zapisać także aksjomaty i łączyć funkcje 0-argumentowe z konkretnym typem, jaki ma przyjąć w kolejnych dowodach.

Zapis aksjomatu:  
**axiom** formuła .

Nadanie typu funkcji:  
**type** nazwa() = Typ.

Symbole relacyjne i termy, należy zapisać zaczynając od małej litery. W przypadku określenia typu termu, należy zacząć od dużej litery.

## 6. Formuły rachunku kwantyfikatorów

Formuły rachunku kwantyfikatorów budowane są z formuł atomowych w sposób podobny, jak w rachunku zdań. Istnieje możliwość związania wystąpienia danej zmiennej w formule używając kwantyfikatorów: **V** (kwantyfikator ogólny) i **E** (kwantyfikator egzystencjalny). Przy związaniu zmiennej możliwe jest określenie typu zmiennej.

Przykład formuły rachunku kwantyfikatorów z użyciem kwantyfikatora ogólnego oraz związania zmiennej  $x$  z typem liczb naturalnych:  
**V x:Nat. (plus(zero(),x) = x)**

Przykład formuły rachunku kwantyfikatorów z użyciem kwantyfikatora egzystencjalnego:  
**E x. p(x)**

## 7. Słowa kluczowe

Zapis dowodu powinien składać się z podanych symboli:

- Implikacja:  $\Rightarrow$
- Prawda: **T**
- Fałsz: **F**
- Negacja:  $\sim$
- Równoważność:  $\Leftrightarrow$
- Symbolu równości:  $=$
- Koniunkcja:  $\wedge$
- Alternatywa:  $\vee$
- Kwantyfikator egzystencjalny: **E**
- Kwantyfikator ogólny: **V**
- Aksjom: **axiom**
- Typ termu: **type**
- Rozpoczęcie i zakończenie dowodu: **proof** i **end**
- Zdefiniowanie dowodu: **goal**