

# **Wavelength**

**$\lambda$ -IDE**

Muhammet Guemues, Markus Himmel, Marc Huiszinga,  
Philip Klemens, Julia Schmid, Jean-Pierre von der Heydt

2018-02-07

# **Inhaltsverzeichnis**

<b>1. Einleitung</b>	<b>3</b>
<b>2. Kriterien</b>	<b>4</b>
2.1. Muss . . . . .	4
2.2. Kann . . . . .	4
2.3. Abgrenzung . . . . .	6
<b>3. Funktionale Anforderungen</b>	<b>7</b>
<b>4. Nicht-Funktionale Anforderungen</b>	<b>14</b>
<b>5. Zustandsautomat</b>	<b>15</b>
<b>6. Tests</b>	<b>16</b>
<b>A. Seitenentwürfe</b>	<b>26</b>
<b>B. Glossar</b>	<b>35</b>

## **1. Einleitung**

Ziel dieses Projektes ist die Bereitstellung einer Online-Entwicklungsumgebung für den  $\lambda$ -Kalkül. Sie kann direkt über einen kompatiblen Browser verwendet werden. Die Entwicklungsumgebung wird für die Lehre entworfen: Sie kann von Dozenten zur Demonstration eingesetzt werden, ebenso kann sie von Studenten zum Lernen verwendet werden. Mit der Anwendung können  $\lambda$ -Terme eingegeben und entsprechend einer Auswertungsstrategie ausgewertet werden. Dabei kann die Auswertung auch Schritt für Schritt verfolgt werden. Der Lernprozess wird durch ein System von Übungsaufgaben unterstützt.

## 2. Kriterien

### 2.1. Muss

M1

#### Eingabe von $\lambda$ -Terminen

Implementiert durch: F1

$\lambda$ -Terme können in Form der vereinfachten  $\lambda$ -Syntax mit der Tastatur in die Software eingegeben werden.

M2

#### Auswertung von $\lambda$ -Terminen

Implementiert durch: F2

Eingegebene  $\lambda$ -Terme können, sofern möglich, mit Hilfe der Normal-Reduktionsordnung vollständig reduziert werden. Die so bestimmte Normalform kann in vereinfachter  $\lambda$ -Syntax dargestellt werden.

M3

#### Fehlermeldung bei invalider Eingabe

Implementiert durch: F11

Beim Versuch, einen syntaktisch inkorrekteten  $\lambda$ -Term reduzieren zu lassen, wird eine Fehlermeldung ausgegeben.

M4

#### Abbruch der Reduktion

Implementiert durch: F12

Der Benutzer kann die Reduktion eines  $\lambda$ -Terms abbrechen.

### 2.2. Kann

K1

#### Weitere Auswertungsstrategien

Implementiert durch: F7

Die Anwendung kann neben der Normal-Reduktionsordnung auch Call by Name, Call by Value und Applicative Order als Auswertungsstrategie verwenden.

K2

#### Exportformate

Implementiert durch: F16

Die eingegebenen und reduzierten  $\lambda$ -Terme können neben der vereinfachten  $\lambda$ -Syntax

auch als Unicode-Text, L<sup>A</sup>T<sub>E</sub>X-Quellcode, Haskell-Quellcode und Lisp-Quellcode formatiert und exportiert werden.

### **Ausgabeformate**

**K3**

Implementiert durch: F4 F5 F6

Die eingegebenen und reduzierten  $\lambda$ -Terme können innerhalb der Applikation in Unicode-Ausgabe und als Syntaxbaum ausgegeben werden.

### **Erweiterte Fehlerdiagnostik**

**K4**

Implementiert durch: F11

Im Falle syntaktischer Fehler werden dem Nutzer die genaue Position des Fehlers, die Art des Fehlers sowie Behebungsmöglichkeiten angezeigt und die relevante Stelle im Editor hervorgehoben.

### **Intelligenter Editor**

**K5**

Implementiert durch: F1

Der Editor zur Eingabe der  $\lambda$ -Terme unterstützt Standard-Editor-Funktionen, Mehrfacheinrückungen und Smart Tabs.

### **Standardbibliothek**

**K6**

Implementiert durch: F9 F10

Es existiert eine Standardbibliothek, die Funktionen für den Umgang mit natürlichen Zahlen, Listen, Tupeln, Booleans und dem Y-Kombinator bereitstellt.

### **Übungsaufgaben**

**K7**

Implementiert durch: F17

Das Programm enthält ein Übungsaufgaben-System, welches neben Aufgaben in aufsteigendem Schwierigkeitsgrad auch Einführungen und automatisierte Tests für die Lösungen des Nutzers enthält.

### **Ausgabe von Teilschritten**

**K8**

Implementiert durch: F3 F8

Die zum Erreichen der Normalform notwendigen Reduktionsschritte gemäß der ausgewählten Auswertungsstrategie können im ausgewählten Ausgabeformat ausgegeben werden. Hierbei können entweder alle Teilschritte, die ersten und letzten Teilschritte, oder einzelne Teilschritte in periodischen Abständen angezeigt werden.

### **Schritt-für-Schritt-Modus**

**K9**

Implementiert durch: F14

Es existiert ein Schritt-für-Schritt-Modus, welcher entweder direkt von der Eingabe oder durch Pausieren einer laufenden Reduktion erreicht werden kann. Durch Klicken kann ein bestimmter Teilausdruck ausgewertet werden oder die ausgewählte Auswertungsstrategie einen einzelnen Schritt machen. Diese Auswertungsschritte können rückgängig gemacht werden.

**K10**

### **Permalinks**

Implementiert durch: F15

Es kann ein Link generiert werden, der beim Aufrufen den gesamten Zustand der Sitzung wiederherstellt.

**K11**

### **Komfortabler $\lambda$ -Code**

Implementiert durch: F13

Einem  $\lambda$ -Term kann ein Name zugewiesen werden. Dieser kann anstelle des Terms verwendet werden. Außerdem können Kommentare geschrieben werden, die bei der Auswertung nicht beachtet werden.

## **2.3. Abgrenzung**

**A1**

### **Nutzung über Mobilgeräte**

Die Anwendung muss nicht auf mobilen Endgeräten einsetzbar sein.

**A2**

### **Lokalisierung**

Die Anwendung soll nur in englischer Sprache verfügbar sein.

### 3. Funktionale Anforderungen

#### **Editor**

F1

Getestet durch: T1 Implementiert: M1 K5

Nutzer können beliebigen Text in einem Eingabefeld eingeben. Das Eingabefeld verfügt über Standard-Textfeld-Funktionalitäten mit dazugehörigen gängigen Tastenkombinationen:

- Editieren, Markieren und Navigieren von Text im Eingabefeld
- Kopieren, Ausschneiden und Einfügen
- Undo und Redo

Außerdem verfügt das Eingabefeld über Smart Tabs und Mehrfacheinrückung. Bei der Eingabe einer öffnenden Klammer wird auch direkt die schließende Klammer hinter der öffnenden Klammer eingefügt. Wird der Editor-Cursor vor oder hinter eine Klammer bewegt, so wird die zugehörige schließende Klammer angezeigt. Neben dem Eingabefeld werden Zeilenummern angezeigt.

#### **Kompletauswertung**

F2

Getestet durch: T1 Implementiert: M2

Der Nutzer hat die Möglichkeit, auf einen eingegebenen  $\lambda$ -Term automatisch  $\beta$ -Reduktionen anwenden zu lassen. Die hierzu verwendete Kompletauswertung wird durch das Drücken des Start-Knopfes (Abb. 3) gestartet. Ist die Eingabe syntaktisch falsch, so wird eine Fehlermeldung ausgegeben und die Kompletauswertung verlassen.

Es werden die Knöpfe zum Laden von Bibliotheken und Aufgaben sowie der Schritt-für-Schritt-Knopf deaktiviert. Außerdem werden das Eingabefeld, das Ausgabeformats-, das Auswertungsstrategie-Dropdown-Menü und das Ausgabeumfangs-Dropdown-Menü deaktiviert. Der Start-Knopf wird zu einem Pause-Knopf.

Die Reihenfolge der  $\beta$ -Reduktionen wird durch die derzeit aktive Auswertungsstrategie bestimmt. Die Ausgabe wird durch das derzeit aktive Ausgabeformat (Abb. 5) und den derzeitig aktiven Ausgabeumfang bestimmt.

Die Kompletauswertung übernimmt die automatische Auswertung, bis die derzeit aktive Auswertungsstrategie keine  $\beta$ -Reduktion mehr vorgibt. Wenn der  $\lambda$ -Term keine  $\beta$ -Normalform besitzt oder die Auswertungsstrategie diese nicht findet, terminiert die Kompletauswertung nicht.

Wird der Pause-Knopf (siehe Abb. 3) betätigt, so wechselt die Ausgabe in den Schritt-für-Schritt-Modus und das Auswertungsstrategie-Dropdown-Menü wird wieder aktiviert.

#### F3 **Live-Updates der Komplettauswertung**

Getestet durch: T1 Implementiert: K8

Solange die Komplettauswertung nicht terminiert, wird im Ausgabefeld der derzeitige Ergebnisstand im gewählten Ausgabeumfang ausgegeben. Insbesondere bei einer Auswahl des Ausgabeumfangs von „Full“ oder „Periodically“ werden dem Nutzer regelmäßig die derzeit berechneten  $\lambda$ -Term ausgegeben.

#### F4 **Unicode-Ausgabe**

Getestet durch: T5 Implementiert: K3

Der Nutzer hat die Möglichkeit, sich Auswertungen von  $\lambda$ -Termen in Unicode ausgeben zu lassen. Die Ausgabe wird im Ausgabefeld ausgegeben.

Jeder neu ausgegebene Auswertungsschritt wird in einer neuen Zeile ausgegeben. Dabei wird am Anfang der Zeile kenntlich gemacht, ob es sich um eine  $\alpha$ -Konversion oder  $\beta$ -Reduktion handelt (siehe Abb. 15). Wird der Mauszeiger in der letzten Zeile über eine Klammer bewegt, so wird die Klammer und die dazugehörige öffnende oder schließende Klammer markiert.

#### F5 **Syntaxbaum-Ausgabe**

Getestet durch: T5 Implementiert: K3

Der Nutzer hat die Möglichkeit, sich Auswertungen von  $\lambda$ -Termen als Syntaxbaum ausgeben zu lassen. Die Ausgabe wird im Ausgabefeld ausgegeben (siehe Abb. 13).

Jeder neu ausgegebene Auswertungsschritt wird als neuer Baum unter dem letzten Auswertungsschritt ausgegeben. Hierbei wird ein  $\lambda$ -Term in folgender Weise in einen Baum übersetzt:

- $\lambda$ -Applikationen werden als Elternknoten mit dem Namen „App“ dargestellt. Der erste und zweite  $\lambda$ -Term der  $\lambda$ -Applikation werden als Kindknoten angefügt (Abb. 14b).
- $\lambda$ -Abstraktionen mit der Variable  $x$  werden als Elternknoten mit dem Namen „ $\lambda x$ “ dargestellt. Der Elternknoten besitzt einen Kindknoten, welcher den  $\lambda$ -Term der  $\lambda$ -Abstraktion enthält.
- Sollte ein Kindknoten des Baums wieder eine  $\lambda$ -Applikation oder eine  $\lambda$ -Abstraktion enthalten, wird dieser Knoten wieder als Eltern und Kindknoten dargestellt (Abb. 14a).

#### F6 **Auswahl des Ausgabeformats**

Getestet durch: T5 Implementiert: K3

Der Nutzer kann das derzeitig aktive Ausgabeformat über ein Dropdown-Menü auswählen (Abb. 5). Das Ausgabeformat entscheidet über die Visualisierung im Ausgabefeld. Das Ausgabeformat hat jedoch keine Auswirkung auf das Exportieren von Ergebnissen. Standardmäßig ist die Unicode-Ausgabe ausgewählt. Der Nutzer kann das Ausgabeformat nur im Eingabemodus ändern (vgl. Abb. 1). Das bedeutet insbesondere, dass der Nutzer das Ausgabeformat nicht während einer laufenden Auswertung ändern kann.

### Auswahl einer Auswertungsstrategie

F7

Getestet durch: T2 Implementiert: K1

Der Nutzer kann die derzeitig aktive Auswertungsstrategie über ein Dropdown-Menü auswählen (Abb. 5b)).

Die folgenden Auswertungsstrategie können gewählt werden:

- Normale-Reduktionsordnung
- Call by Name
- Call by Value
- Applicative Order

Die Auswertungsstrategie bestimmt die Reihenfolge von automatischen  $\beta$ -Reduktionen und damit die Markierung des nächsten zu reduzierenden  $\lambda$ -Terms im Ausgabefeld (Abb. 15). Standardmäßig ist die Normal-Reduktionsordnung ausgewählt. Der Nutzer kann die Auswertungsstrategie im Schritt-für-Schritt-Modus ändern, jedoch nicht während der Kompletauswertung.

### Auswahl des Ausgabeumfangs

F8

Getestet durch: T8 Implementiert: K8

Der Nutzer kann den Ausgabeumfang über ein Dropdown-Menü auswählen (Abb. 5). Standardmäßig ist der Ausgabeumfang „Result only“ ausgewählt. Der Ausgabeumfang bestimmt die Menge an Teilergebnissen einer Auswertung, die die Kompletauswertung in das Ausgabefeld schreibt. Der Nutzer kann den Ausgabeumfang während der Kompletauswertung nicht ändern (vgl. Abb. 1). Hierbei stehen dem Nutzer folgende Optionen zur Verfügung:

- Result only. Es wird nur das Ergebnis der Kompletauswertung ausgegeben.
- Shortened. Es werden nur die ersten 50 Schritte und die letzten 50 Schritte ausgegeben.
- Full. Es wird jeder einzelne Teilschritt ausgegeben, sobald er berechnet wurde.

- Periodically. Jeder hundertste Teilschritt wird ausgegeben, sobald er berechnet wurde. Der erste Schritt und das Endergebnis werden ebenfalls ausgegeben.

**F9**

### **Ganzzahlliterale**

Getestet durch: T10 Implementiert: K6

Bei Verwendung der vereinfachten  $\lambda$ -Syntax können natürliche Zahlen in Dezimaldarstellung als Literale eingegeben werden. Stellt die Applikation fest, dass eine Operation über eine als Literal eingegebene natürliche Zahl durchgeführt werden soll, wird diese Operation, sofern der Nutzer die Auswertung nicht konkret herbeiführt, im Moment der Reduktion direkt durch das Literal für das Ergebnis der Operation ersetzt.

**F10**

### **Standardbibliothek**

Getestet durch: T10 Implementiert: K6

Der Nutzer kann auf eine Bibliothek von Definitionen zurückgreifen. Die Standardbibliothek ist in Module aufgeteilt. Jedes Modul kann einzeln eingebunden werden. Die folgenden Module werden bereitgestellt:

- Basisbibliothek. Dieses Modul enthält den Y-Kombinator.
- Natürliche Zahlen. Es existieren Funktionen zur Bestimmung des Vorgängers und Nachfolgers einer natürlichen Zahl und Funktionen zur Bestimmung der Summe, der Differenz, des Produktes und des Ergebnisses der Exponentiation zweier natürlicher Zahlen.

Die Implementation erfolgt durch Church-Zahlen. Natürliche Zahlen werden durch Literale dargestellt.

- Wahrheitswerte. Es werden Definitionen für `true` und `false` bereitgestellt, sodass `true a b` zu `a` und `false a b` zu `b` ausgewertet wird.
- Tupel und Listen. Es wird eine `null`-Definition zusammen mit einer `isNull`-Funktion bereitgestellt. Es existieren Funktionen `cons`, `car` und `cdr`, mit denen ein 2-Tupel erzeugt werden kann und das erste bzw. zweite Element eines 2-Tupels extrahiert werden kann. Die gleichen Funktionen können auch zum Umgang mit einfach verketteten Listen verwendet werden: `null` ist die leere Liste, `cons a list` hängt `a` vorne an `list` an, `car` gibt das erste Element und `cdr` den Rest der Liste zurück.

**F11**

### **Fehlerdiagnostik**

Getestet durch: T7 T11 Implementiert: M3 K4

Wenn der Nutzer versucht, einen syntaktisch inkorrektenden  $\lambda$ -Term auszuwerten, wird im

Ausgabefeld eine Fehlermeldung angezeigt. Die Fehlermeldung enthält die genaue Position (Zeilen- und Spaltennummer), an der die Interpretation der Eingabe fehlschlug, eine Beschreibung des Fehlertyps (Unerwartetes Zeichen, unbalancierte Klammerung, ...) und in einfachen Fällen einen Hinweis auf mögliche Fehlerbehebung (beispielsweise wenn Einfügen einer schließenden Klammer an der problematischen Stelle zu einem syntaktisch korrekten  $\lambda$ -Term führt). Die problematische Stelle wird im Editor zusätzlich farblich markiert.

### **Abbruch der Auswertung**

F12

Getestet durch: T9 Implementiert: M4

Durch das Drücken des „Abbrechen“-Knopfes (Abb. 3) kann der Nutzer während des Schritt-für-Schritt-Modus und während der Kompletauswertung in den Eingabemodus zurückkehren.

Dabei wird die Auswertung abgebrochen und der Inhalt des Ausgabefelds erhalten, die Steuerung der Ausgabe aber deaktiviert. Alle Knöpfe, Felder und Menüs, welche beim Start der Auswertung deaktiviert wurden, werden wieder aktiviert. Anschließend kann der Nutzer das Eingabefeld wieder bearbeiten.

### **Komfortabler $\lambda$ -Code**

F13

Getestet durch: T7 Implementiert: K11

Im Editor können Kommentare verfasst werden, indem am Beginn der Zeile ein ‘//’ eingefügt wird.

$\lambda$ -Terme können Namen zugewiesen werden. Namensrekursion ist ausgeschlossen. Ein Name darf nur einmal vergeben werden. Die Syntax für die Definition findet sich im Glossar unter Namensbindung. Der Name kann in der Eingabe an Stelle des  $\lambda$ -Terms eingesetzt werden. Die Ausgabe berücksichtigt den Namen soweit möglich: Ist der konkrete  $\lambda$ -Term für die Reduktion unerheblich, so wird immer der Name verwendet. Ansonsten wird der Name aufgelöst und der  $\lambda$ -Term verwendet.

### **Schritt-für-Schritt-Modus**

F14

Getestet durch: T2 T3 Implementiert: K9

Durch Drücken des Schritt-für-Schritt-Knopfes wird zuerst die Eingabe aus dem Eingabefeld überprüft und eventuell eine Fehlermeldung ausgegeben, falls die Eingabe nicht syntaktisch korrekt ist.

Die korrekte Eingabe wird als erster Schritt in das Ausgabefeld geladen und der nächste zu reduzierende Redex gemäß der ausgewählten Auswertungsstrategie markiert.

Es werden die Knöpfe zum Laden von Bibliotheken und Aufgaben sowie der Schritt-für-Schritt-Knopf deaktiviert. Außerdem werden das Eingabefeld, das Ausgabeformats- und das Ausgabeumfangs-Dropdown-Menü deaktiviert, während der „vor“-Knopf aktiviert wird.

Beim Klick auf „vor“ wird der markierte Redex in der neuesten Zeile gemäß der ausge-

wählten Auswertungsstrategie reduziert und der reduzierte Term mit Markierung in einer neuen Zeile ausgegeben. Die Markierung in der alten Zeile bleibt hierbei erhalten.

Befindet sich mehr als eine Zeile in der Ausgabe, so kann die Ausführung zu dem Zustand der vorigen Zeile mit dem „zurück“-Knopf zurückgespult werden. Wird der „zurück“-Knopf betätigt, so wird die neueste Zeile entfernt und die Markierung der Zeile davor neu gemäß der ausgewählten Auswertungsstrategie markiert.

Wird der Mauszeiger in der neuesten Zeile auf ein nicht markiertes, zu einem Redex gehörendes  $\lambda$ -Symbol bewegt, so wird der zum  $\lambda$ -Symbol gehörige Redex markiert und die alte Markierung entfernt. Die alte Markierung wird wiederhergestellt, sobald der Mauszeiger wieder vom  $\lambda$ -Symbol weg bewegt wird.

Wird in der neuesten Zeile auf ein zu einem Redex gehörendes  $\lambda$ -Symbol geklickt, so wird dieser reduziert und der reduzierte Term mit Markierung in einer neuen Zeile ausgegeben.

Die Auswertungsstrategie kann im Schritt-für-Schritt-Modus geändert werden, was die Markierung der alten Auswertungsstrategie durch die der neuen ersetzt.

Wird der „Start“-Knopf betätigt, so wird die Komplettauswertung mit der ausgewählten Auswertungsstrategie gestartet und das Ausgabefeld sowie die „vor“- und „zurück“-Knöpfe deaktiviert.

## F15

### Permalinks

Getestet durch: T3 Implementiert: K10

Die in der Adresszeile des Browsers angezeigte URL kodiert den Zustand der Sitzung. Der Zustand besteht aus:

- Inhalt des Eingabefeldes.
- Inhalt des Ausgabefeldes.
- Ausgewähltes Ausgabeformat, Auswertungsstrategie, und Ausgabeumfang.
- Geladenen Bibliotheken
- Geladene Übungsaufgabe
- Der aktuelle Zustand der IDE (siehe Abb. 1).

Nach jeder Aktion wird die angezeigte URL angepasst, sofern sich der Zustand der IDE verändert.

Aktionen umfassen sowohl Eingaben des Nutzers, Interaktionen mit der Benutzeroberfläche als auch die automatischen Ausgaben während der Komplettauswertung.

Wird das Programm durch einen Link geöffnet, so wird der von der Link-URL codierte Zustand wiederhergestellt.

### **Export**

**F16**

Getestet durch: T6 Implementiert: K2

Beim Klicken auf den „Export“-Knopf kann eines der folgenden Exportformate gewählt werden: (siehe auch Abb. 11)

- vereinfachte  $\lambda$ -Syntax: Stellt das  $\lambda$ -Symbol durch ‘\’ dar.
- Unicode-Text: Stellt das  $\lambda$ -Symbol als Unicode-Symbol dar.
- $\text{\LaTeX}$ -Quellcode: Formatiert den Text als  $\text{\LaTeX}$ -Quellcode. Enthält auch die Markierungen der Ausgabe.
- Lisp-Quellcode: Formatiert den Text als Lisp-Quellcode mit Lisp-Lambdas. Analoge Ausführung in Lisp ist nicht garantiert.
- Haskell-Quellcode: Formatiert den Text als Haskell-Quellcode mit Haskell-Lambdas. Analoge Ausführung in Haskell sowie Typisierbarkeit sind nicht garantiert.

Nach der Auswahl öffnet sich ein Fenster. Dieses enthält den aktuellen Inhalt des Ausgabefelds im ausgewählten Format. Der Inhalt kann nun kopiert werden.

### **Übungsaufgaben**

**F17**

Getestet durch: T4 Implementiert: K7

Das Programm enthält eine Sammlung von Übungsaufgaben zum  $\lambda$ -Kalkül.

Im Optionsmenü wird eine Liste aller Übungsaufgaben (Abb. 6) angezeigt.

Wählt der Nutzer eine Übung aus, so wird der Nutzer über ein Fenster darauf hingewiesen, dass durch einen Wechsel in den Übungsmodus seine aktuelle Eingabe und Ausgabe gelöscht wird. In diesem Fenster kann der Nutzer auswählen, ob er mit dem Moduswechsel fortfahren oder diesen abbrechen möchte. Wird „fortfahren“ ausgewählt, so wechselt das Programm in den Übungsmodus und Eingabe und Ausgabe werden geleert.

Die Aufgabenstellung wird in einem neuen Aufgabenfeld neben dem Eingabefeld angezeigt. Eventuell vorgegebener Code wird in das Eingabefeld geladen (siehe Abb. 7). Die Aufgabenstellung definiert eventuell bereits gebundene Namen, welche Eingaben, die von der IDE bereitgestellt werden, zum Testen der Aufgabenstellung enthalten.

Der Nutzer kann anschließend die Aufgabe bearbeiten.

Wenn der Nutzer seine Lösung auswerten möchte, werden für die vorher festgelegten Platzhalter konkrete  $\lambda$ -Terme eingesetzt. Die Platzhalter können möglicherweise bei jeder Ausführung mit anderen Werten ersetzt werden. Sobald die Auswertung die  $\beta$ -Normalform erreicht, wird dem Nutzer ausgegeben, ob sein Ergebnis mit dem erwarteten Ergebnis übereinstimmt (siehe Abb. 8). Abhängig von der Aufgabe wird dem Nutzer außerdem die erwartete  $\beta$ -Normalform ausgegeben.

Wird ein Schritt rückgängig gemacht, so werden auch die Informationen über das Ergebnis des Tests entfernt, bis der Nutzer wieder die  $\beta$ -Normalform erreicht.

Weiterhin hat der Nutzer die Möglichkeit, sich durch Drücken eines Knopfes die Musterlösung anzeigen zu lassen (siehe Abb. 9). Durch Drücken des selben Knopfes wird diese wieder ausgeblendet.

Solange sich das Programm im Übungsmodus befindet, kann der Nutzer durch Drücken des „Übungsmodus abbrechen“-Knopfes in den Eingabemodus zurückkehren.

Bevor der Übungsmodus verlassen wird, öffnet sich ein Fenster, das den Nutzer darauf aufmerksam macht, dass seine Eingaben und Ausgaben im Übungsmodus durch das Verlassen des Übungsmodus verloren gehen. In diesem Fenster kann der Nutzer auswählen, ob er mit dem Moduswechsel fortfahren oder diesen abbrechen möchte.

Beim letztendlichen Wechsel in den Eingabemodus werden Eingabefeld und Ausgabefeld geleert.

## 4. Nicht-Funktionale Anforderungen

### N1 Benutzeroberfläche

Die Benutzeroberfläche ist während der Auswertung noch verwendbar, d.h. die Oberfläche blockiert nicht vollständig während der Ausführung.

### N2 Lizenz

Die IDE unterliegt der BSD-3-Klausel.

### N3 Öffentlichkeit

Der zur IDE gehörige Programmcode, die zugehörigen Dokumente, etc sind öffentlich und ohne Anmeldung zugänglich unter <https://github.com/wavelength-ide/wavelength-ide>.

### N4 Implementierung

Die Implementierung muss in einer statisch typisierten, objektorientierten Programmiersprache erfolgen.

## Benutzbarkeit

N5

Nach Aufruf der Seite kann ein Nutzer in drei Schritten einen  $\lambda$ -Term nach der normale Reduktionsordnung auswerten:

- Eingabe eines  $\lambda$ -Terms
- Drücken des „Start“-Knopfes
- Warten auf das Ende der Auswertung

## 5. Zustandsautomat

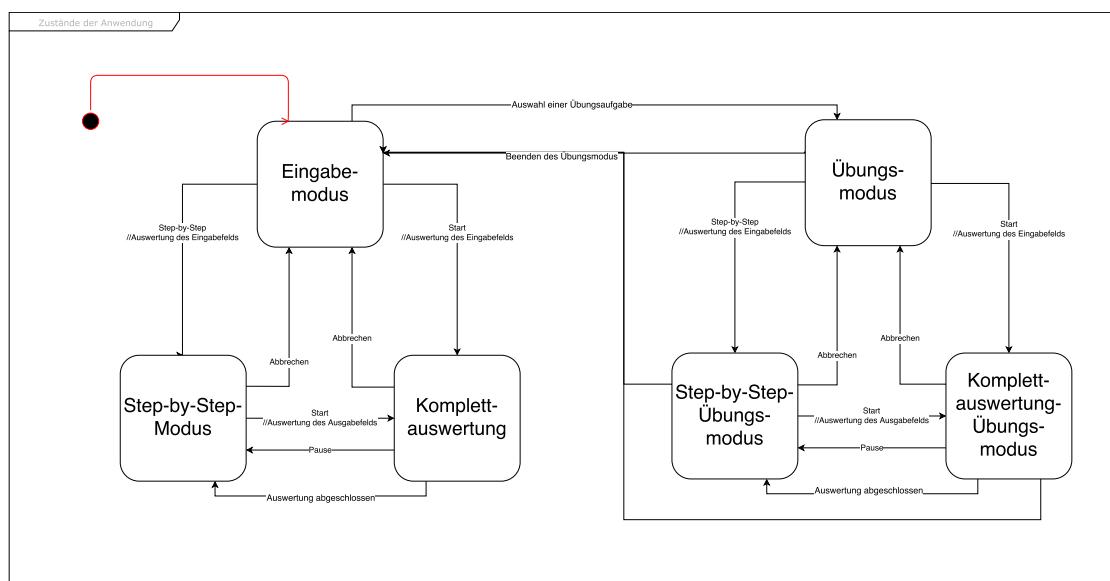


Abbildung 1: Zustandsautomat der Anwendung

Die Anwendung kann sich in drei verschiedenen Zuständen befinden, wobei jeder Zustand ein Gegenstück im Übungsmodus besitzt. Der Übungsmodus wird durch Auswahl einer Übungsaufgabe betreten und durch Beendigung des Übungsmodus verlassen.

Beim Start der Anwendung befindet sich der Nutzer im Eingabemodus. In diesem Modus kann der Nutzer einen  $\lambda$ -Term in das Eingabefeld eingeben. Durch Drücken des Schritt-

für-Schritt- oder Start-Knopfes (Abb. 3) gelangt der Nutzer in den jeweiligen Auswertungsmodus und es beginnt die Auswertung des eingegebenen  $\lambda$ -Terms. Zeitgleich wird das Eingabefeld deaktiviert. Wenn noch Ausgaben einer früheren Auswertung im Ausgabefeld stehen, werden diese gelöscht.

Durch Drücken des Start- oder Pause-Knopfes kann der Nutzer zwischen den entsprechenden Auswertungsmodi wechseln. Beim Wechsel vom Schritt-für-Schritt-Modus zur Kompletauswertung wird der zuletzt ausgegebene  $\lambda$ -Term des Ausgabefeldes weiter ausgewertet. Die Auswertungsmodi können erst durch das Drücken des Abbrechen-Knopfes verlassen werden. Es erscheint eine entsprechende Ausgabe im Ausgabefeld. Erst jetzt kann der Nutzer neue Eingaben im Eingabefeld eingeben.

## 6. Tests

T1

### Kompletauswertung

Testet: F1 F2 F3

T1.1 **Stand:** Nutzer „Stephen Cook“ hat einen Browser geöffnet.

**Aktion:** Stephen navigiert zur URL des Wavelength-Projekts.

**Reaktion:** Die IDE öffnet sich wie in Abbildung 2.

T1.2 **Stand:** Die IDE ist fertig geladen.

**Aktion:** Stephen gibt im Eingabefeld einen gültigen  $\lambda$ -Term mit Klammern ein, dessen Reduktion durch die normale Reduktionsordnung terminiert. Hierbei navigiert er über den Text und macht eine vermeintlich fehlerbehaftete Änderung rückgängig. Stephen merkt, dass die Änderung doch korrekt war und springt wieder zum aktuellen Stand. Er kopiert etwas Text und fügt diesen an einer anderen Stelle wieder ein. Der  $\lambda$ -Term ist mehrere Zeilen lang und Stephen rückt mehrere Zeilen des  $\lambda$ -Terms gleichzeitig ein. Stephen fügt eine neue Zeile hinter einer bereits eingerückten Zeile ein und schreibt in diese Zeile einen Kommentar. Stephen bewegt seinen Mauszeiger über eine Klammer, um die dazugehörige schließende Klammer zu finden. Er ändert den Ausgabeumfang auf die Ausgabe aller Teilschritte und die Auswertungsstrategie zur Call by Name-Reduktionsordnung.

**Reaktion:** Das Textfeld verhält sich wie erwartet.

T1.3 **Stand:** Als Ausgabeformat ist die Unicode-Ausgabe, als Auswertungsstrategie die normale Reduktionsordnung und als Ausgabenumfang die Ausgabe aller Teilschritte vorausgewählt.  
**Aktion:** Stephen betätigt den „Start“-Knopf.  
**Reaktion:** Das Ausgabefeld füllt sich graduell mit den Zwischenschritten, welche gemäß Call by Name auftreten. Zum Schluss wird die Normalform des eingangs eingegebenen Terms ausgegeben.

### Schritt-für-Schritt-Modus

T2

Testet: F7 F14

T2.1 **Stand:** Benutzer „Donald Knuth“ befindet sich auf der Wavelength-Seite.  
**Aktion:** Er gibt einen syntaktisch korrekten  $\lambda$ -Term in das Eingabefeld ein und betätigt den „Schritt-für-Schritt“-Knopf.  
**Reaktion:** Der eingegebene Term wird in das Ausgabefeld übernommen und der Vor-Knopf wird aktiv. Im Term wird der nächste  $\beta$ -Reduktionsschritt gemäß der normalen Reduktionsordnung markiert.

T2.2 **Stand:** Die IDE befindet sich im Schritt-für-Schritt-Modus.  
**Aktion:** Don betätigt den „Vor“-Knopf.  
**Reaktion:** Das Resultat der markierten Reduktion erscheint als neue Zeile im Ausgabefeld. In der neuen Zeile ist der nächste Reduktionsschritt markiert.

T2.3 **Stand:** Die IDE ist bereit für die nächste Eingabe.  
**Aktion:** Don betätigt den „Zurück“-Knopf.  
**Reaktion:** Der zweite Reduktionsschritt wird aus dem Ausgabefeld entfernt.

T2.4 **Stand:** Die IDE ist bereit für die nächste Eingabe.  
**Aktion:** Don schwebt mit der Maus über einem nicht markierten, zu einem Redex gehörenden  $\lambda$ -Symbol in der aktiven Zeile im Ausgabefeld.  
**Reaktion:** Die alten Markierungen in der aktiven Zeile verschwinden. Neue Markierungen für den Redex, über dem sich die Maus befindet, erscheinen.

T2.5 **Stand:** Der Mauszeiger befindet sich über einem Redex.  
**Aktion:** Don bewegt die Maus von diesem Redex weg.  
**Reaktion:** Die Markierungen verschwinden und die Markierungen gemäß der normalen Reduktionsordnung werden wieder angezeigt.

T2.6 **Stand:** Die IDE ist bereit für die nächste Eingabe.

**Aktion:** Don klickt mit der Maus auf ein zu einem Redex gehörendes  $\lambda$ -Symbol.

**Reaktion:** Es erscheint eine neue Zeile im Ausgabefeld, die das Ergebnis der angeklickten Reduktion enthält. In der neuen Zeile erscheinen die Markierungen gemäß der ausgewählten Auswertungsstrategie.

T2.7 **Stand:** Die IDE ist bereit für die nächste Eingabe.

**Aktion:** Don wählt das Dropdown-Menü für die Auswertungsstrategie an. Er wählt „Call by Name“.

**Reaktion:** Die Markierungen für die vorher ausgewählte Auswertungsstrategie werden ausgeblendet. Stattdessen wird der nächste Schritt gemäß Call by Name angezeigt.

T2.8 **Stand:** Die IDE ist bereit für die nächste Eingabe.

**Aktion:** Don betätigt den „Vor“-Knopf.

**Reaktion:** Das Resultat der markierten Reduktion (gemäß Call by Name) erscheint als neue Zeile im Ausgabefeld. In der neuen Zeile wird der nächste Reduktionsschritt markiert.

T2.9 **Stand:** Die IDE ist bereit für die nächste Eingabe.

**Aktion:** Don wiederholt die vorangegangenen beiden Schritte für „Call by Value“ und „Applicative Order“.

**Reaktion:** Die Applikation verhält sich wie in den vorherigen beiden Schritten beschrieben, verwendet aber die jeweils ausgewählte Auswertungsstrategie.

### T3

#### Erstellen vom Permalinks

Testet: F15 F14

T3.1 **Stand:** Benutzer „Edsger Dijkstra“ befindet sich auf der Wavelength-Seite.

**Aktion:** Edsger gibt einen gültigen  $\lambda$ -Term in die Software ein, wählt eine Auswertungsstrategie und ein Ausgabeformat aus, begibt sich in den Schritt-für-Schritt-Modus und durchläuft einige Schritte der Auswertung.

**Reaktion:** Die Anwendung verhält sich wie spezifiziert.

T3.2 **Stand:** Die IDE ist bereit für die nächste Aktion.

**Aktion:** Edsger verwendet den „Teilen“-Knopf im unteren Bereich der Anwendung.

**Reaktion:** Neben dem Knopf erscheint eine URL wie in Abbildung 12 abgebildet.

T3.3 **Stand:** Edsger hat die URL in seine Zwischenablage kopiert.  
**Aktion:** Edsger wählt die URL in einem neuen Browserfenster an.  
**Reaktion:** Die IDE öffnet sich im neuen Fenster und Eingabefeld, Ausgabefeld und alle Optionen befinden sich im gleichen Zustand wie im alten Fenster.

T3.4 **Stand:** Die IDE ist im neuen Fenster geöffnet.  
**Aktion:** Edsger betätigt den „Zurück“-Knopf.  
**Reaktion:** Der letzte im alten Browserfenster durchgeführte Schritt wird im neuen Browserfenster rückgängig gemacht.

## Übungsmodus

T4

Testet: F17

T4.1 **Stand:** Benutzerin „Ada Lovelace“ befindet sich auf der Wavelength-Seite.  
**Aktion:** Ada wählt im Menü (vgl. Abb. 7) eine Übungsaufgabe aus.  
**Reaktion:** Es öffnet sich ein Fenster, das Ada darauf hinweist, dass ihre Eingabe und Ausgabe verloren geht. Ada wählt „fortfahren“ aus. Der Übungsmodus wird aktiviert. Das Eingabefeld wird aufgeteilt in ein schmaleres Eingabefeld und ein Aufgabenfeld. Im Aufgabenfeld erscheint eine Beschreibung der gewählten Übungsaufgabe. Der „Beenden“- und der „zeige Lösung“-Knopf erscheinen.

T4.2 **Stand:** Die IDE ist bereit für die nächste Aktion.  
**Aktion:** Ada gibt eine syntaktisch korrekte Lösung ein, die kein korrektes Ergebnis liefert. Ada drückt den „Start“-Knopf.  
**Reaktion:** Die IDE wertet die Eingabe wie spezifiziert aus. Nach dem letzten Schritt wird angezeigt, dass die Lösung inkorrekt ist, sowie das erwartete Ergebnis.

T4.3 **Stand:** Die IDE ist bereit für die nächste Aktion.  
**Aktion:** Ada gibt die korrekte Lösung ein. Ada drückt den „Start“-Knopf.  
**Reaktion:** Die IDE wertet die Eingabe wie spezifiziert aus. Nach dem letzten Schritt wird angezeigt, dass die Lösung korrekt ist.

T4.4 **Stand:** Die IDE ist bereit für die nächste Aktion.  
**Aktion:** Ada wählt eine andere Übungsaufgabe aus und bestätigt, dass Eingabe- und Ausgabe verloren gehen.  
**Reaktion:** Eingabe und Ausgabe werden entfernt. Im Aufgabenfeld erscheinen eine Beschreibung der Übungsaufgabe und ein Stück Code.

T4.5 **Stand:** Die IDE ist bereit für die nächste Aktion.

**Aktion:** Ada vervollständigt den Code zu einer korrekten Lösung, verwendet dabei die in der Aufgabestellung genannten vordefinierten Namen für die Eingabe und durchläuft den Schritt-für-Schritt-Modus komplett von Hand.

**Reaktion:** Die IDE verhält sich wie spezifiziert. Beim Starten des Schritt-für-Schritt-Modus werden die vordefinierten Namen durch Testwerte ersetzt. Als letzter Schritt wird angezeigt, dass die eingegebene Lösung korrekt ist.

T4.6 **Stand:** Die IDE ist bereit für die nächste Aktion.

**Aktion:** Ada drückt auf den „zeige Lösung“-Knopf.

**Reaktion:** Die Musterlösung wird im Aufgabenfeld angefügt. Der „zeige Lösung“-Knopf wird zum „verberge Lösung“-Knopf.

T4.7 **Stand:** Im Aufgabenfeld wird die Musterlösung angezeigt.

**Aktion:** Ada kopiert die Musterlösung, überschreibt ihre Eingabe damit und drückt den „Start“-Knopf.

**Reaktion:** Die IDE verhält sich wie spezifiziert. Als letzter Schritt wird angezeigt, dass die eingegebene Lösung korrekt ist.

T4.8 **Stand:** Die IDE ist bereit für die nächste Aktion.

**Aktion:** Ada drückt auf den „verberge Lösung“-Knopf.

**Reaktion:** Die Musterlösung wird aus dem Aufgabenfeld entfernt.

T4.9 **Stand:** Die IDE ist bereit für die nächste Aktion.

**Aktion:** Ada drückt den „Beenden“-Knopf.

**Reaktion:** Es öffnet sich ein Fenster, das Ada darauf hinweist, dass ihre Eingabe und Ausgabe verloren geht. Ada wählt „fortfahren“ aus. Der Übungsmodus wird beendet, der „Beenden“-Knopf verschwindet, Eingabe und Ausgabe werden entfernt. Das Aufgabenfeld wird entfernt.

## T5

### Ausgabeformate

Testet: F6 F4 F5

T5.1 **Stand:** Benutzer „Richard Hamming“ befindet sich auf der Wavelength-Seite.

**Aktion:** Richard gibt einen syntaktisch korrekten  $\lambda$ -Term ein und drückt den „Start“-Knopf.

**Reaktion:** Der eingegebene  $\lambda$ -Term wird ausgewertet und im Unicode-Format im Ausgabefeld angezeigt.

T5.2 **Stand:** Im Ausgabefeld wird das Ergebnis der vorherigen Schritts angezeigt.  
**Aktion:** Richard bewegt den Mauszeiger auf eine öffnende Klammer in der letzten Zeile der Ausgabe.  
**Reaktion:** Die öffnende Klammer und die zugehörige schließende Klammer werden markiert.

T5.3 **Stand:** Die IDE ist bereit für die nächste Aktion  
**Aktion:** Richard wählt im Dropdown-Menü (vgl. Abb. 5a) das Ausgabeformat „tree“ und drückt den „Start“-Knopf.  
**Reaktion:** Der eingegebene  $\lambda$ -Term wird ausgewertet und wie spezifiziert (vgl. 3) ausgegeben.

## Export

Testet: F16

T6

T6.1 **Stand:** Benutzer „Robert Tarjan“ befindet sich auf der Wavelength-Seite.  
**Aktion:** Robert gibt einen gültigen  $\lambda$ -Term ein, wählt als Ausgabeumfang „full“, drückt auf den „Start“-Knopf, wartet, bis das finale Ergebnis im Ausgabefeld zu sehen ist und drückt auf den „Exportieren“-Knopf.  
**Reaktion:** Es öffnet sich eine Liste mit allen Exportformaten.

T6.2 **Stand:** Die IDE erwartet die Auswahl eines Exportformats.  
**Aktion:** Robert wählt ein beliebiges Exportformat aus der Liste aus.  
**Reaktion:** Die Liste mit den Exportformaten verschwindet. Es öffnet sich ein Fenster (vgl. Abb. 11) mit der im gewählten Format formatierten Ausgabe, die inhaltlich mit dem Ausgabefenster übereinstimmt.

T6.3 **Stand:** Der Fokus befindet sich auf dem Fenster mit der formatierten Ausgabe.  
**Aktion:** Robert kopiert die formatierte Eingabe in die Zwischenablage und schließt das Fenster.  
**Reaktion:** Das Fenster mit der formatierten Eingabe verschwindet. Eingabe und Ausgabe sind unverändert.

## Namensbindung

Testet: F13 F11

T7

T7.1 **Stand:** Benutzer „Haskell Curry“ befindet sich auf der Wavelength-Seite.

**Aktion:** Haskell gibt einen gültigen  $\lambda$ -Term ein und nennt ihn „curry“. Haskell verwendet „curry“ in einem weiteren gültigen  $\lambda$ -Term und drückt den „Start“-Knopf.

**Reaktion:** Die Eingabe wird gemäß Spezifikation ausgewertet (vgl. Namensbindung).

T7.2 **Stand:** Die IDE ist bereit für die nächste Aktion.

**Aktion:** Haskell gibt einen gültigen  $\lambda$ -Term ein. Er benennt ihn mit einem gemäß Spezifikation (vgl. Namensbindung) ungültigen Namen. Er drückt den „Start“-Knopf.

**Reaktion:** Es wird eine Fehlermeldung ausgegeben. Haskell wird darauf hingewiesen, dass er einen ungültigen Namen gewählt hat.

T7.3 **Stand:** Die IDE ist bereit für die nächste Aktion.

**Aktion:** Haskell bindet zwei gültige  $\lambda$ -Terme an den gleichen Namen und drückt den „Start“-Knopf.

**Reaktion:** Es wird eine Fehlermeldung ausgegeben. Haskell wird darauf hingewiesen, dass Namen nur einmal gebunden werden dürfen.

## T8

### Auswahl des Ausgabeumfangs

Testet: F8

T8.1 **Stand:** Benutzer „Wilhelm Ackermann“ befindet sich auf der Wavelength-Seite.

**Aktion:** Wilhelm gibt einen gültigen  $\lambda$ -Term ein und drückt auf den „Start“-Knopf.

**Reaktion:** Die Eingabe wird ausgewertet und nur das Ergebnis wird angezeigt (vgl. Standardeinstellung).

T8.2 **Stand:** Die IDE ist bereit für die nächste Aktion.

**Aktion:** Wilhelm wählt im Ausgabeumfang-Dropdown-Menü einen anderen Ausgabeumfang aus (vgl. Abb. 5) und drückt auf den „Start“-Knopf.

**Reaktion:** Die IDE verhält sich wie spezifiziert (vgl. 3).

T8.3 **Stand:** Die IDE ist bereit für die nächste Aktion.

**Aktion:** Wilhelm wählt in der „Display“-Box das Ausgabeformat „Tree“ und drückt den „Start“-Knopf.

**Reaktion:** Die IDE verhält sich wie spezifiziert (vgl. 3). Im Ausgabefeld erscheint ein Syntaxbaum gemäß des ausgewählten Ausgabeumfangs (vgl. 3).

## Wechsel des Auswertungsmodus

T9

Testet: F12

T9.1 **Stand:** Die Kompletauswertung eines korrekten  $\lambda$ -Terms läuft.

**Aktion:** Nutzer „Bob“ drückt den „Pause“-Knopf.

**Reaktion:** Die Auswertung wird angehalten, der aktuelle  $\lambda$ -Term wird in das Ausgabefeld geschrieben, die IDE wechselt in den Schritt-für-Schritt-Modus.

T9.2 **Stand:** Die IDE befindet sich im Schritt-für-Schritt-Modus.

**Aktion:** Bob drückt den „Start“-Knopf.

**Reaktion:** Die IDE wechselt in den Kompletauswertungs-Modus, der eingegebene  $\lambda$ -Term wird ohne Aktionen des Nutzers ausgewertet.

T9.3 **Stand:** Die Kompletauswertung eines  $\lambda$ -Terms läuft.

**Aktion:** Bob drückt den „Abbrechen“-Knopf.

**Reaktion:** Die Auswertung wird abgebrochen, das Ausgabefeld wird erhalten und deaktiviert, das Eingabefeld kann bearbeitet werden.

## Standardbibliothek

T10

Testet: F9 F10

T10.1 **Stand:** Benutzer „Alan Turing“ befindet sich auf der Wavelength-Seite.

**Aktion:** Alan wählt im Menü die Y-Kombinator-Bibliothek aus.

**Reaktion:** Die gewählte Bibliothek wird geladen.

T10.2 **Stand:** Die Basisbibliothek ist eingebunden.

**Aktion:** Alan benutzt den Y-Kombinator in einem gültigen  $\lambda$ -Term und drückt den „Start“-Knopf.

**Reaktion:** Die IDE verhält sich wie spezifiziert.

T10.3 **Stand:** Die natürlichen Zahlen sind eingebunden.

**Aktion:** Alan definiert eine natürliche Zahl durch ein Literal. Für diese Zahl benutzt er die Funktionen, um Vorgänger und Nachfolger zu bestimmen. Dann drückt er den „Start“-Knopf.

**Reaktion:** Die IDE verhält sich wie spezifiziert.

T10.4 **Stand:** Die natürlichen Zahlen sind eingebunden.

**Aktion:** Alan berechnet zwei natürliche Zahlen mit den gegebenen Operationen (siehe 3).

**Reaktion:** Die IDE verhält sich wie spezifiziert.

T10.5 **Stand:** Die natürlichen Zahlen sind eingebunden.

**Aktion:** Alan gibt den Term „ $(\lambda x.2x)(3)$ “ ein und drückt den „Start“-Knopf.

**Reaktion:** Es wird „6“ ausgegeben.

T10.6 **Stand:** Die Wahrheitswerte sind eingebunden.

**Aktion:** Alan definiert zwei gültige  $\lambda$ -Terme a und b. Alan gibt ein: „true a b“ und drückt den „Start“-Knopf.

**Reaktion:** Es wird „a“ ausgegeben.

T10.7 **Stand:** Die Wahrheitswerte sind eingebunden.

**Aktion:** Alan definiert zwei gültige  $\lambda$ -Terme a und b. Alan gibt ein: „false a b“ und drückt den „Start“-Knopf.

**Reaktion:** Es wird „b“ ausgegeben.

T10.8 **Stand:** Tupel und Listen sind eingebunden.

**Aktion:** Alan erschafft ein Tupel und benutzt die Funktionen car bzw. cdr (siehe 3). Er drückt den „Start“-Knopf.

**Reaktion:** Es wird das erste bzw. das zweite Element des Tupels ausgegeben.

T10.9 **Stand:** Tupel und Listen sind eingebunden.

**Aktion:** Alan erschafft eine leere Liste *list* mit *null*. Mit dem Befehl *cons a list* hängt er das Element a an den Anfang von *list* und bindet das Resultat an den Namen *list2*. Mit dem Befehl *cons b list2* hängt er das Element b an den Anfang von *list2* und bindet das Resultat an den Namen *list3*. Alan benutzt die Funktionen *car* und *cdr* (siehe 3), um das erste bzw. zweite Element von *list3* zu erhalten und drückt den „Start“-Knopf.

**Reaktion:** Es wird a bzw. b ausgegeben.

## T11

### Fehlerdiagnostik

Testet: F11

T11.1 **Stand:** Benutzer „Richard Bellman“ befindet sich auf der Wavelength-Seite.

**Aktion:** Richard gibt einen syntaktisch inkorrekteten  $\lambda$ -Term ein, der eine schließende Klammer zu wenig enthält und wählt den „Start“-Knopf an.

**Reaktion:** Im Ausgabefeld wird eine Nachricht angezeigt, die auf unbalancierte Klammerung hinweist. Die Nachricht enthält einen Vorschlag für einen korrigierten Term. Im Eingabefenster wird die Klammer, die keinen Partner besitzt, markiert.

T11.2 **Stand:** Die IDE ist bereit für die nächste Aktion.

**Aktion:** Richard löscht die aktuelle Eingabe aus dem Eingabefenster und gibt einen inkorrekten Term ein, der strukturell weder  $\lambda$ -Applikation, noch  $\lambda$ -Abstraktion, noch Variable ist. Dann wählt er den „Start“-Knopf an.

**Reaktion:** Im Ausgabefeld erscheint eine Nachricht, die darauf hinweist, dass der eingegebene Term strukturell inkorrekt ist.

T11.3 **Stand:** Die IDE ist bereit für die nächste Aktion.

**Aktion:** Richard leert das Eingabefeld und wählt den „Start“-Knopf an.

**Reaktion:** Im Ausgabefeld erscheint eine Nachricht, die darauf hinweist, dass die Eingabe leer war.

T11.4 **Stand:** Die IDE ist bereit für die nächste Aktion.

**Aktion:** Richard gibt „ $//(\lambda x.x) (5)$ “ in das Eingabefeld ein und drückt den „Start“-Knopf.

**Reaktion:** Im Ausgabefeld erscheint eine Nachricht, die darauf hinweist, dass die Eingabe leer war.

T11.5 **Stand:** Die IDE ist bereit für die nächste Aktion.

**Aktion:** Richard gibt zwei syntaktisch korrekte  $\lambda$ -Terme, die beide keine Namensbindung sind, nacheinander in das Eingabefeld ein und wählt den „Start“-Knopf an.

**Reaktion:** Im Ausgabefeld erscheint eine Nachricht, die darauf hinweist, dass eine gültige Eingabe aus beliebig vielen Namensbindungen gefolgt von genau einem freistehendem  $\lambda$ -Term besteht.

## A. Seitenentwürfe

Bei Aufruf der Website wird eine leere IDE angezeigt. Das obere Textfeld ist das Eingabefeld, das untere ist das Ausgabefeld. Das Größenverhältnis der Felder kann mittels einer Schaltfläche angepasst werden.

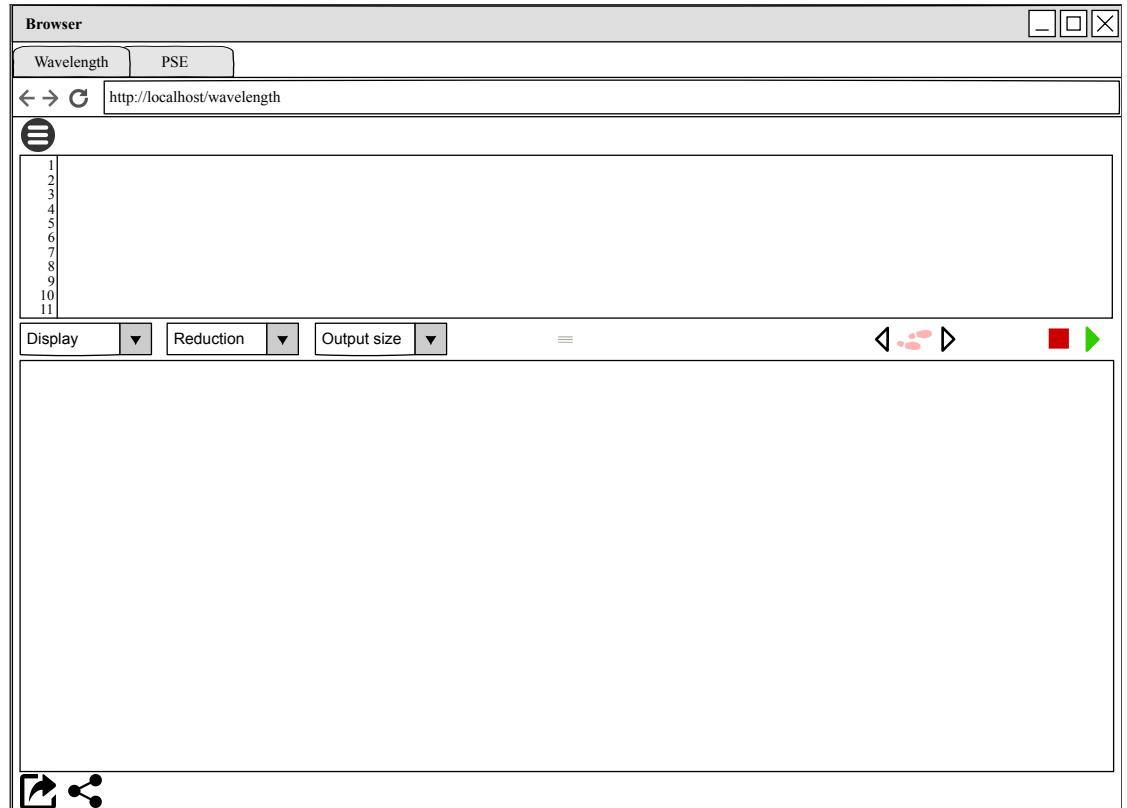


Abbildung 2: Die Startseite der IDE bei Aufruf der Website.

Der „Start“-Knopf löst die Auswertung zum Endergebnis aus. Durch Drücken des „Abbrechen“-Knopfes wird die Auswertung abgebrochen. Beim Drücken des Schritt-für-Schritt-Knopfes wird der Schritt-für-Schritt-Modus gestartet, in welchem der Nutzer mit den beiden Knöpfen daneben navigieren kann.



Abbildung 3: Schritt-zurück, Schritt-für-Schritt-Modus, Schritt-vor, Start und Abbrechen (von links nach rechts)



Abbildung 4: Wurde die Auswertung mittels „Start“ gestartet, wird der „Start“-Knopf durch einen „Pause“-Knopf ersetzt, um in den Schritt-für-Schritt-Modus wechseln zu können.

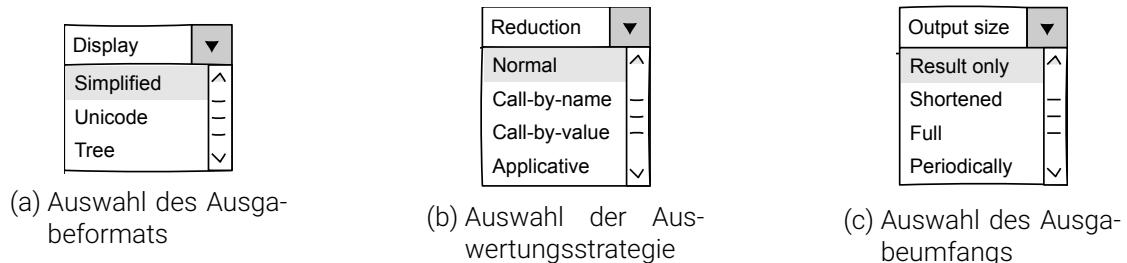


Abbildung 5: Verschiedene Ausgabeoptionen können festgelegt werden.

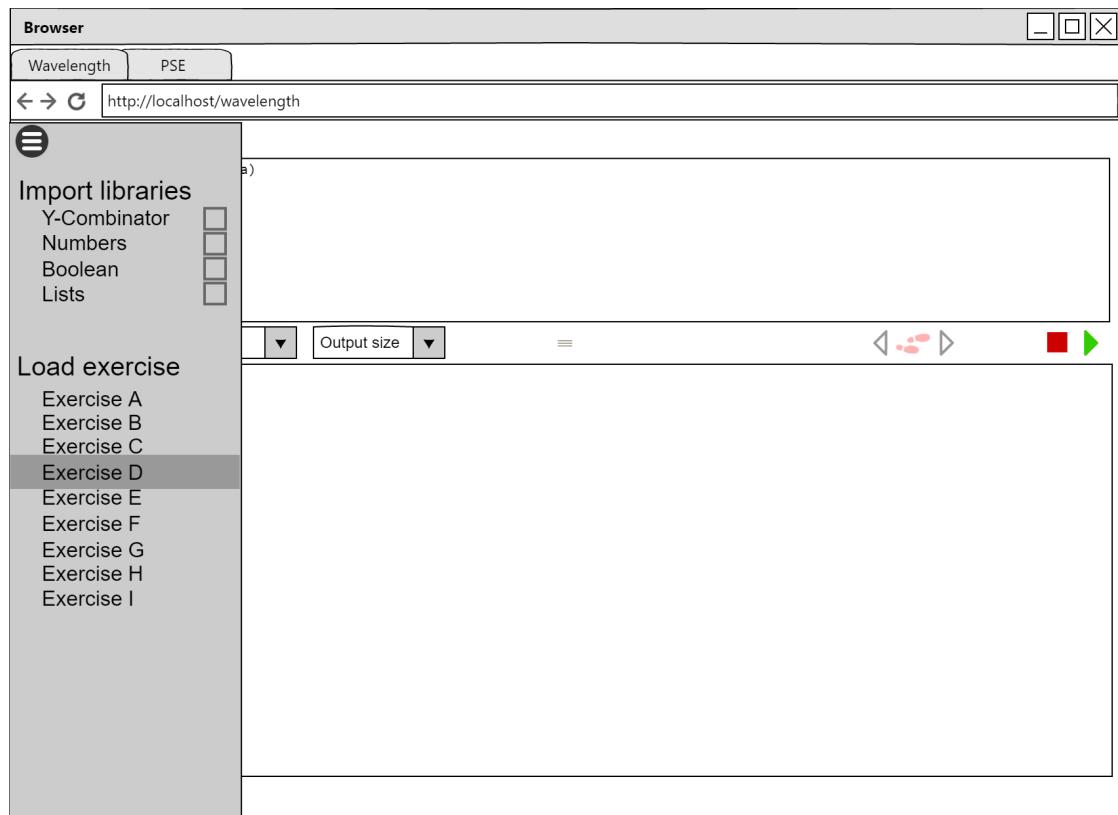


Abbildung 6: Im Optionsmenü hat der Nutzer die Möglichkeit, Bibliotheken einzubinden oder eine Übungsaufgabe auszuwählen.

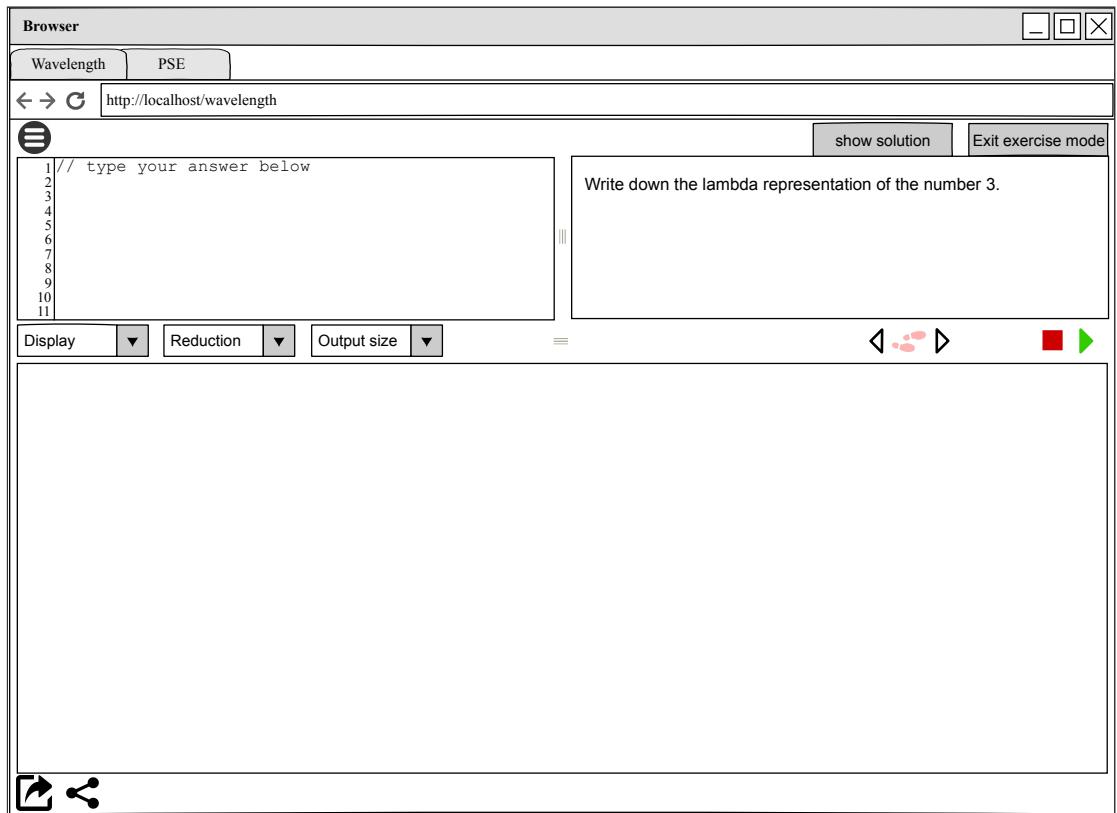


Abbildung 7: Wurde eine Übungsaufgabe ausgewählt, so wird die Aufgabenstellung im neu erscheinenden Aufgabenfeld angezeigt. Eventuell vorgegebene Eingaben werden in das Eingabefeld geladen. Der Nutzer hat die Möglichkeit, das Größenverhältnis dieser beiden Felder anzupassen. Die Musterlösung der Aufgabe wird durch Klicken eines entsprechenden Knopfes angezeigt oder auch wieder ausgeblendet. Mit Klick auf einen weiteren Knopf kann der Übungsmodus beendet werden.

The screenshot shows a browser window titled "Browser" with the URL <http://localhost/wavelength>. The interface is designed for a programming exercise. On the left, there is a code editor with the following content:

```
// type your answer below
s.\z.s(s z)
```

Below the code editor are three dropdown menus: "Display", "Reduction", and "Output size". To the right of the code editor is a large text input field containing the lambda expression  $\lambda s. \lambda z. s(s z)$ . Above this input field are two buttons: "hide solution" and "Exit exercise mode". To the right of the input field is the instruction: "Write down the lambda representation of the number 3.". At the bottom right of the input field are four small icons: a left arrow, a right arrow, a red square, and a green triangle.

Abbildung 8: Lässt der Nutzer seine Eingabe prüfen, so wird im Ausgabefeld angezeigt, ob diese korrekt ist.

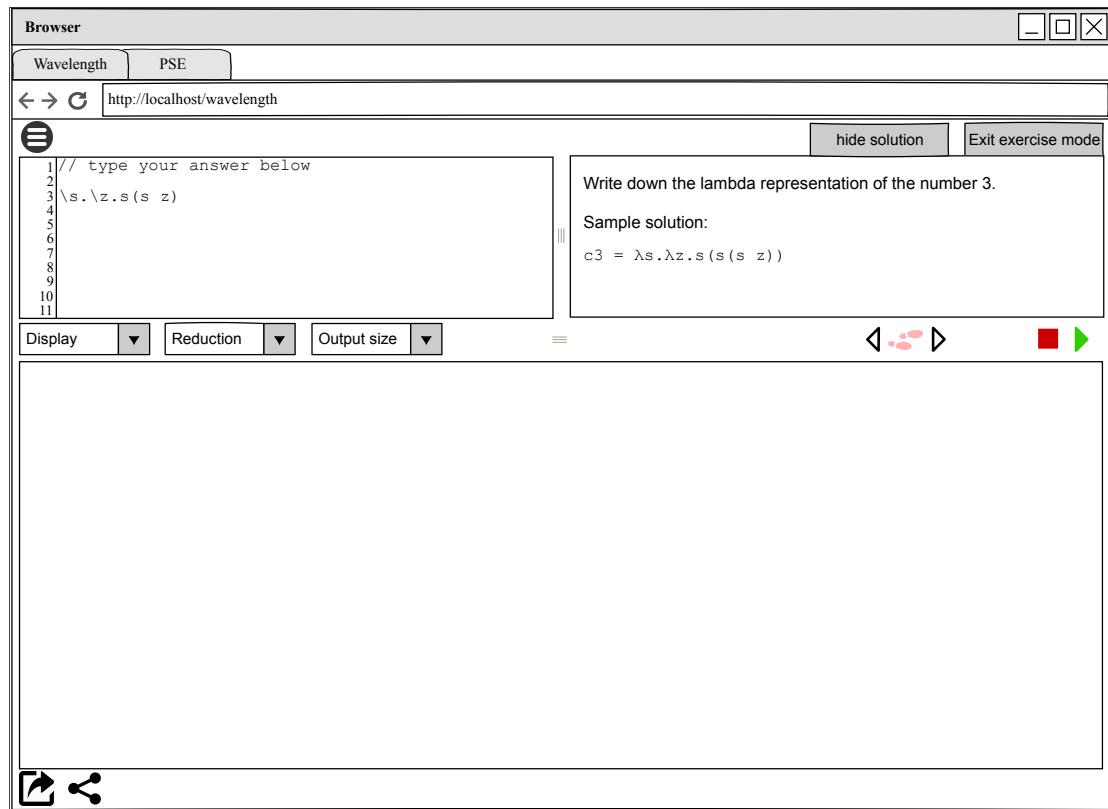


Abbildung 9: Wird die Musterlösung angezeigt, so hat der Nutzer die Möglichkeit, diese auch wieder auszublenden.

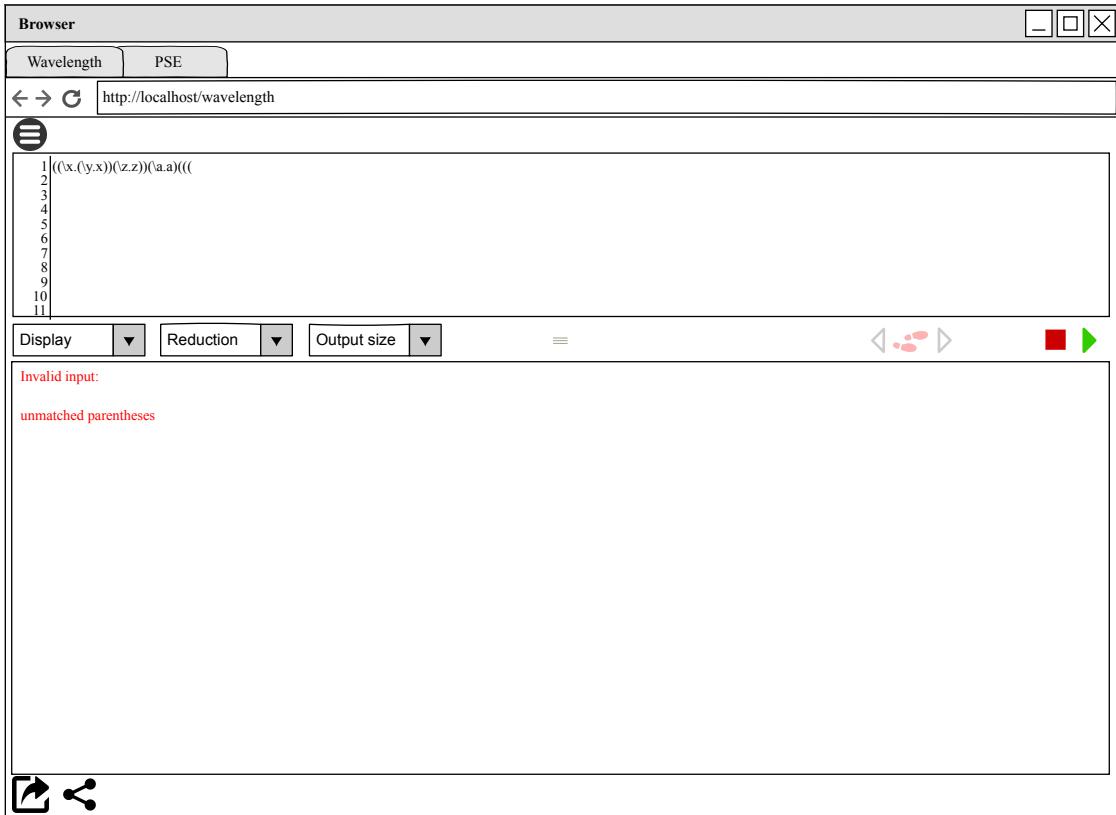


Abbildung 10: Bei fehlerhafter Eingabe wird eine Fehlermeldung im Ausgabefeld ausgegeben.

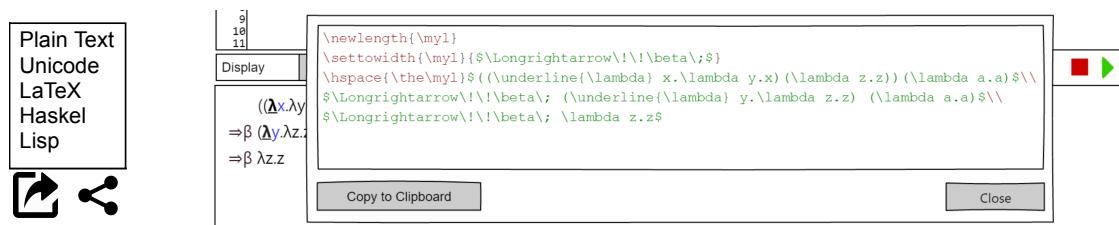


Abbildung 11: Durch Klick auf den Export-Knopf kann der aktuelle Inhalt des Ausgabefelds in das gewünschte Format exportiert werden (Hier L<sup>A</sup>T<sub>E</sub>X-Quellcode).

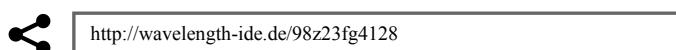


Abbildung 12: Bei Betätigen des Teilen-Knopfes wird ein entsprechender Link erstellt.

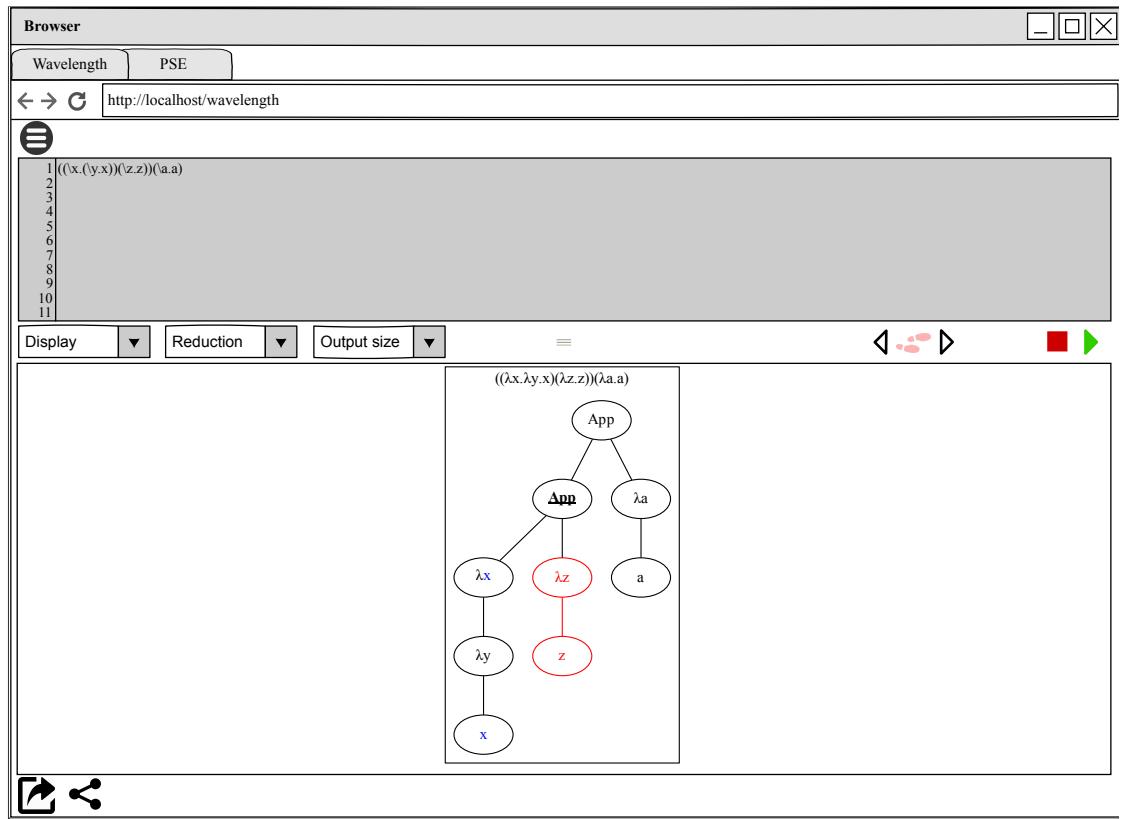
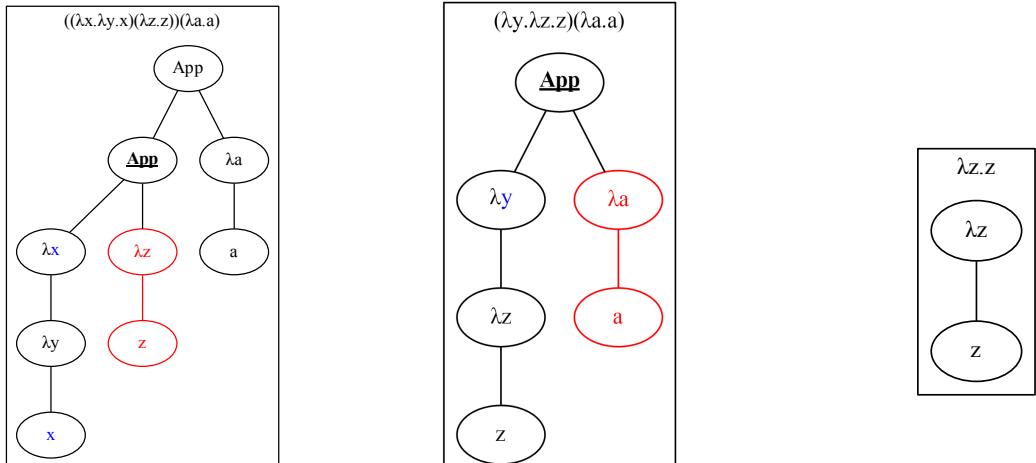


Abbildung 13: Wird das Ausgabeformat „Tree“ ausgewählt (siehe Abb. 5), so werden einzelne Schritte der Ausführung als Syntaxbaum im Ausgabefeld angezeigt.



(a) Syntaxbaum mit ausgewählter  $\lambda$ -Applikation (unterstrichen).

(b) Neue Ausgabe beim Anklicken der ausgewählten  $\lambda$ -Applikation.

(c) Endausgabe bei Auswahl der Wurzel.

Abbildung 14: Wird ein „App“-Symbol ( $\lambda$ -Applikation) angeklickt, so wird dieser Teilterm reduziert und der neue Syntaxbaum angezeigt.

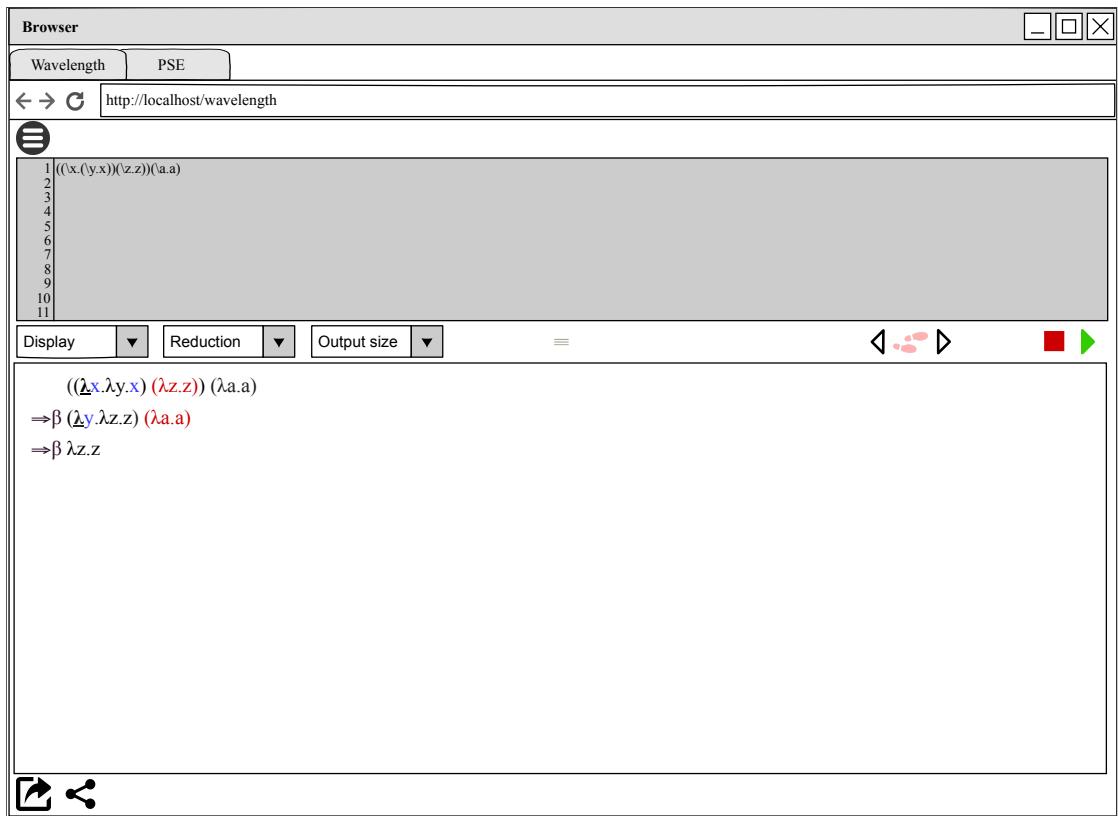


Abbildung 15: Wird das Ausgabeformat „Unicode“ ausgewählt (siehe Abb. 5), so wird der eingegebene  $\lambda$ -Term mit Unicode- $\lambda$ -Zeichen im Ausgabefeld angezeigt.

## **B. Glossar**