

FLACI – Eine Lernumgebung für theoretische Informatik

Michael Hielscher¹, Christian Wagenknecht²

Abstract: FLACI (www.flaci.com) steht für Formal Languages and Compilers and Interpreters und ist eine modular aufgebaute, webbasierte Lernumgebung für grundlegende Inhalte der theoretischen Informatik. Die Theorie formaler Sprachen und Automaten befasst sich mit potentiell unendlichen Mengen. Die damit verbundene Begrifflichkeit in FLACI adäquat abzubilden, ist die besondere Herausforderung. Die Lernumgebung bringt deklaratives, konzeptionelles und prozedurales Wissen mit, um Definitionen, Simulationen, Transformationen und Konvertierungen für formale Grammatiken, reguläre Ausdrücke und abstrakte Automaten vornehmen zu können. Die Webapplikation FLACI stellt eine konsequente Weiterentwicklung der Lernumgebung AtoCC aus dem Jahr 2006 dar. Eine lokale Installation entfällt, wodurch diverse softwaretechnische und betriebssystemspezifische Probleme vermieden werden können. Dadurch ist der Einsatz von FLACI, neben dem akademischen Bereich, auch für den Informatikunterricht in der Schule (vorwiegend Sekundarstufe II) attraktiv. Die Applikation ist sowohl im Unterricht als auch außerhalb zeitlich und örtlich unabhängig verfügbar, speichert alle erarbeiteten Inhalte auf Wunsch automatisch und regelmäßig in der FLACI-Cloud und bietet damit Potential zur kollaborativen Arbeit. Als offene Lernumgebung bietet FLACI einen Konstruktionsraum für einfache Compiler und Interpreter als praxisrelevante Motivation für die Anwendung von Theoriekenntnissen. Hierfür bietet die Umgebung spezielle kontextfreie Sprachklassen und Transformationen an, die die Herstellung von Compilern konzeptionell unterstützen. Die Verzahnung von Theorie und praktischer Anwendung wird seit vielen Jahren im Informatikstudium umgesetzt. Im Beitrag wird FLACI im Überblick vorgestellt und zu den GI-Bildungsstandards "Informatik für die Sekundarstufe II" in Beziehung gebracht.

Keywords: Theoretische Informatik, Formale Sprachen, Abstrakte Automaten, Lernumgebung

1 Einleitung

Die theoretische Informatik (TI) ist an Universitäten und Hochschulen ein fester Bestandteil des Informatikstudiums. Zunehmend werden ausgewählte Konzepte dieses Gebietes in den Schulunterricht integriert. Die GI-Bildungsstandards für Informatik in der Sekundarstufe II [RBD16] weisen Sprachen und Automaten als einen zentralen Inhaltsbereich aus und definieren konkrete Themen, an denen sich die Lehrpläne orientieren können. Im gymnasialen Informatikunterricht ist die TI je nach Bundesland und Lehrplan bereits ein obligatorischer oder wenigstens ein möglicher Wahlbereich, s. z. B. [FPA07]. Auf die besonderen Anforderungen einer stufengerechten fach- und mediendidaktischen Konzeption für diesen Inhaltsbereich wurde hingewiesen, s. z. B. [MHL16].

¹ Medien und Schule, Pädagogische Hochschule Schwyz, Forschung, Zaystrasse 42, Goldau, CH-6410, michael.hielscher@phsz.ch

² Fakultät Elektrotechnik und Informatik, Hochschule Zittau/Görlitz, FB Informatik, Brückenstr. 1, Görlitz, 02826, c.wagenknecht@hszg.de

In den vergangenen Jahrzehnten entstanden zahlreiche Lernumgebungen für TI: JFLAP [RF06], Kara [BHN03], Exorciser [Ts04], AtoCC [HW06], [HW07] und [WH14]. Im Folgenden beschränken wir unseren Betrachtungsrahmen auf die TI-Teilgebiete “Theorie der formalen Sprachen” und die “Automatentheorie”. Diese Lernumgebungen decken jeweils einen Teil der geforderten Kompetenzen (in unterschiedlicher Ausprägung) ab und sind mehrheitlich für das universitäre Studium konzipiert. Mit AtoCC wurde ein Weg gefunden, die begriffliche Arbeit durch Definition von Instanzen und Anwendung algorithmischer Zusammenhänge zu betonen [WH14]. Dabei wird die Aktivität des Lernenden nicht nur auf das seit über 50 Jahren ausgearbeitete Gebiet der Automaten Simulationen, s. [CSK11], reduziert.

Die mit AtoCC gewonnenen fachdidaktischen Erfahrungen (Studium und Sekundarstufe II) sind in die Entwicklung von FLACI eingeflossen, s. Kap. 2. Darüber hinaus wurden moderne mediendidaktische Aspekte beachtet: Der überwiegende Teil der im Einsatz befindlichen Lernumgebungen, einschl. AtoCC, basiert auf Technologien, die von mobilen Geräten (z. B. Tablets) nicht mehr unterstützt werden. Als Reaktion auf die zunehmende Verbreitung solcher Geräte in der Hand von Lernenden und Lehrpersonen wurde AtoCC grundlegend überarbeitet und FLACI als moderne Webapplikation entwickelt. Ein ähnliches Ziel wird auch in [PT18] verfolgt. Unter Berücksichtigung der Bildungsstandards ist ein didaktisches Werkzeug entstanden, das sowohl im Informatikstudium als auch im Informatikunterricht (vorwiegend Sekundarstufe II) eingesetzt werden kann. Darüber hinaus eignet sich FLACI als praxistaugliche Entwicklungsumgebung für Compiler und Interpreter, wie sie in modernen Webapplikationen Verwendung finden.

2 Struktur der Lernumgebung FLACI

Der Einsatz von FLACI ist darauf gerichtet, den Informatikunterricht durch Experimentiergelegenheiten zu ergänzen. Automatisiertes Feedback unterstützt die Lernenden etwa bei der Konstruktion von Automatenmodellen und formalen Grammatiken und weist auf allfällige Fehler oder Unvollständigheiten hin. Die Simulation von Ableitungen, von Arbeitsschritten abstrakter Automaten und von Compilationsprozessen ist dynamischer Natur und wird in FLACI mediendidaktisch passender repräsentiert als durch die Darstellung einzelner Visualisierungsschritte in gedruckten Unterrichtsmaterialien. FLACI stellt eine umfangreiche Palette an Transformations- und Konvertierungsverfahren bereit, die im Unterricht erarbeitet und von den Lernenden aufgabengeleitet und ergebnisorientiert angewandt werden. Außerdem bietet die Lernumgebung unmittelbare Hilfe zur Selbsteinschätzung vorgelegter Lösungen.

FLACI besteht aus fünf eigenständigen, aufeinander aufbauenden Modulen, die in der vorgegebenen Folge aber auch einzeln, verflochten oder unabhängig voneinander verwendet werden können. Im Folgenden werden die einzelnen Module kurz beschrieben.

2.1 Formale Sprachen

Das Theoriemodul bietet eine einfache, interaktive Umgebung, um zentrale Begriffe formaler Sprachen, wie Alphabet, Zeichen, Wort, Wortmenge und Sprache, zu explorieren. Es stehen vordefinierte Alphabete zur Auswahl. Zeichen, die aus mehreren Tastaturzeichen bestehen, bergen die große Gefahr einer typischen Fehlvorstellung vom Zeichenbegriff. Sie werden als atomare Blättchen dargestellt und verwendet. Die Lernenden können zudem eigene Alphabete definieren, s. Abbildung 1.

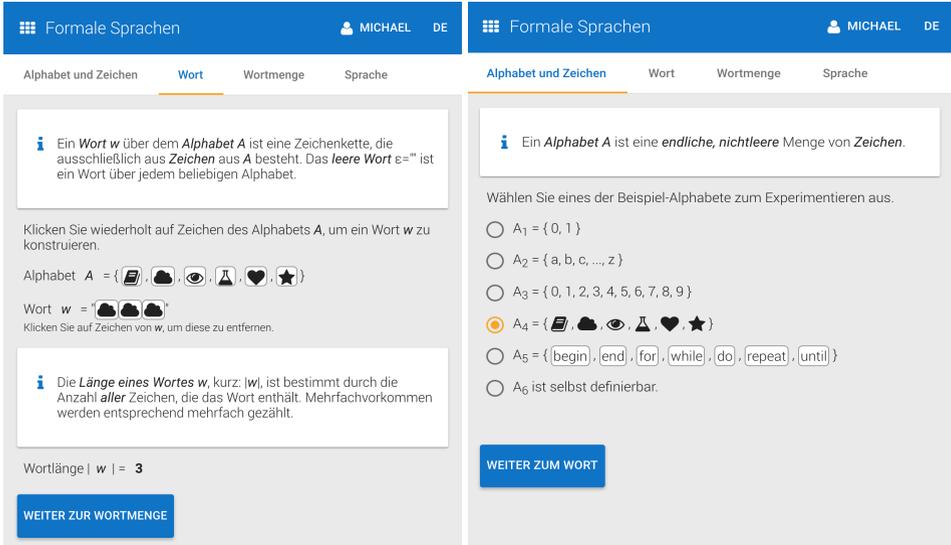


Abbildung 1: Alphabet- und Zeichenbegriff (links) und Wortbegriff (rechts)

Die mögliche Eigenschaft von Wortmengen und Sprachen abzählbar unendliche Mengen zu sein, wird durch eine „aufklappbare“ Punktnotation, für beliebig viele weitere Elemente der betrachteten Mengen, erfahrbar. Die in FLACI eingebauten Definitionen der o. g. Begriffe dienen als verbindliche Implementierung der durch die Lehrperson eingeführten Begrifflichkeit, s. Abbildung 1.

2.2 Reguläre Ausdrücke

Sprachen durch manuelle Auswahl einzelner Wörter aus der jeweils zugrunde liegenden Wortmenge zu definieren erschöpft sich schnell und führt zur Notwendigkeit geeigneter Beschreibungsmittel für potenziell unendliche Mengen. Reguläre Ausdrücke sind eine kompakte und mächtige Definitionsform regulärer Sprachen. Die Syntax regulärer Ausdrücke wird in FLACI an konkreten Beispielen im Dialog illustriert. Zu jedem angegebenen regulären Ausdruck wird ein entsprechendes Syntaxdiagramm generiert. Die Betrachtung

tung von Kfz-Kennzeichen und E-Mail-Adressen werden zur abschließenden selbstständigen Übung angeboten. In einem Experimentierbereich lassen sich beliebige reguläre Ausdrücke und Suchtexte eingeben, Zufallswörter aus der beschriebenen Sprache erzeugen, äquivalente Grammatiken und endliche Automaten generieren, s. Abbildung 2.

Abbildung 2: Reguläre Ausdrücke mit interaktiven Erklärungen (links) und Beispielaufgaben (rechts)

Ergänzend kann die praktische Bedeutung regulärer Ausdrücke etwa bei Such- und Ersetzungsaufgaben in gängigen Texteditoren oder Werkzeugen wie `grep` auf der Kommandozeile thematisiert werden. Deshalb wurde in FLACI eine praxisrelevante Notation regulärer Ausdrücke gewählt und auf die formale Betrachtung regulärer Mengen und deren Überführung in zugehörige Sprachen verzichtet, was im Unterricht jedoch thematisiert werden muss, s. [WH14].

2.3 Kontextfreie Grammatiken

Schnell stößt man an die Grenzen regulärer Sprachen. Praktische Einsatzzwecke motivieren die Betrachtung formaler Sprachen, die nicht mit regulären Ausdrücken definiert werden können. Folglich stellt sich die Frage nach der Definition und passenden Beschreibungsmitteln für Sprachklassen, die über reguläre Sprachen hinausgehen.

Abstrakte Automaten und formale Grammatiken leisten das Gewünschte. Das Grammatik-Modul in FLACI erlaubt sowohl eine textuelle (BNF: Backus-Naur-Form) als auch visuelle Eingabe (Syntax-Diagramm) von Grammatiken, s. Abbildungen 3.

Für beliebige Eingabewörter, die zur Sprache gehören, können Ableitungen als Ableitungsbäume und tabellarische Satzformlisten dargestellt werden. Der Ableitungsprozess lässt sich schrittweise animieren und zwischen Links- und Rechtsableitung umschalten. Bei Mehrdeutigkeiten werden für ein betrachtetes Wort mehrere Ableitungsbäume zur

Auswahl angeboten. Das Modul stellt Algorithmen bereit, die Grammatiken z. B. zum Zwecke der Vereinfachung umformen oder beispielsweise überprüfen, ob die grammatischen Regeln syntaktisch denen einer regulären Grammatik entsprechen oder ob es sich um eine LL(1)-Sprache handelt. Außerdem lassen sich reguläre Grammatiken in reguläre Ausdrücke oder endliche Automaten konvertieren. Kontextfreie Grammatiken (kfG) können in Kellerautomaten (s. Abschnitt 2.4) konvertiert werden.

The screenshot shows the FLACI software interface. On the left, a list of grammar rules is displayed in a table format:

1	Stylesheet	→ Ruleset Ruleset Stylesheet
2	Ruleset	→ R1 R3
3	R1	→ Selector Selector , R1
4	R2	→ Declaration Declaration ; R2
5	R3	→ { R2 }
6	Selector	→ Identifier
7	Declaration	→ Property : Expression
8	Property	→ Identifier
9	Expression	→ Number Identifier Number %
10	Identifier	→ Buchstabe Buchstabe Identifier2
11	Identifier2	→ Buchstabe Ziffer - Buchstabe Identifier2
12	Number	→ Zahl Zahl . Zahl
13	Ziffer	→ 0 1 2 3 4 5 6 7 8 9

Below the rules, a section titled "So definiert man eine Grammatik in FLACI:" provides instructions:

- Man notiert nur die Produktionsregeln.
- Nach Vereinbarung ist das Nichtterminal auf der linken Seite der zuerst angegebenen Regel das Spitzensymbol der Grammatik.
- Für ε in ε-Regeln ist EPSILON zu schreiben: ε → a | EPSILON
- Die Pünktchennotation hilft den Schreibaufwand zu reduzieren:

On the right, the "Ableitungsbaum von 1:" (Parse Tree) is shown. The root node is "Stylesheet", which derives "Ruleset". "Ruleset" derives "R1" and "R3". "R1" derives "Selector", which further derives "Identifier" (Buchstabe 'h') and "Identifier2" (Ziffer '1'). "R3" derives "Declaration", which derives "Property" (Identifier: Buchstabe 'c'), "Expression" (Identifier: Buchstabe 'o'), and "Expression" (Identifier: Buchstabe 'b'). The "Expression" nodes further derive "Identifier" nodes, which then derive "Buchstabe" and "Identifier2" nodes, eventually leading to the terminal symbols 'c', 'o', and 'b'.

Abbildung 3: Formale Grammatiken in BNF (links) und Ableitungsbaum (rechts)

2.4 Abstrakte Automaten

Als äquivalente Beschreibungsmittel formaler Grammatiken können abstrakte Automaten mit Hilfe des entsprechenden Moduls in FLACI eingeführt werden. Das Modul umfasst deterministische und nichtdeterministische endliche Automaten (DEA, NEA), Kellerautomaten (DKA, NKA) und deterministische Turingmaschinen (DTM), s. Abbildung 4. Die Konstruktion erfolgt primär grafisch mittels Übergangsgraph des Automaten und einer vorherigen Definition der verwendeten Alphabete. Endliche Automaten lassen sich zusätzlich über eine editierbare Übergangstabelle erstellen und bearbeiten. Die Arbeitsweise der Automaten lässt sich für beliebige Eingabewörter simulieren und bietet die Möglichkeit einzelne Schritte animiert nachvollziehen. Nichtdeterminismus wird in Form geklonter Maschinen, deren Arbeitsweise einzeln beobachtet werden kann, erkennbar. Es stehen Verfahren bereit, mit denen endliche Automaten in äquivalente reguläre Ausdrücke oder reguläre Grammatiken, Kellerautomaten in kontextfreie Grammatiken konvertiert werden können. Zu jedem endlichen Automaten kann der äquivalente Minimal-Automat erzeugt werden. Auch dieses Modul bietet einen fließenden Übergang zu den Themen eines Informatikstudiums. Vielfältige Assistenzsysteme unterstützen bei der Konstruktion von Automaten (z. B. automatische Vervollständigung bei endlichen Automaten und farbliche Hervorhebungen bei fehlenden Übergängen).

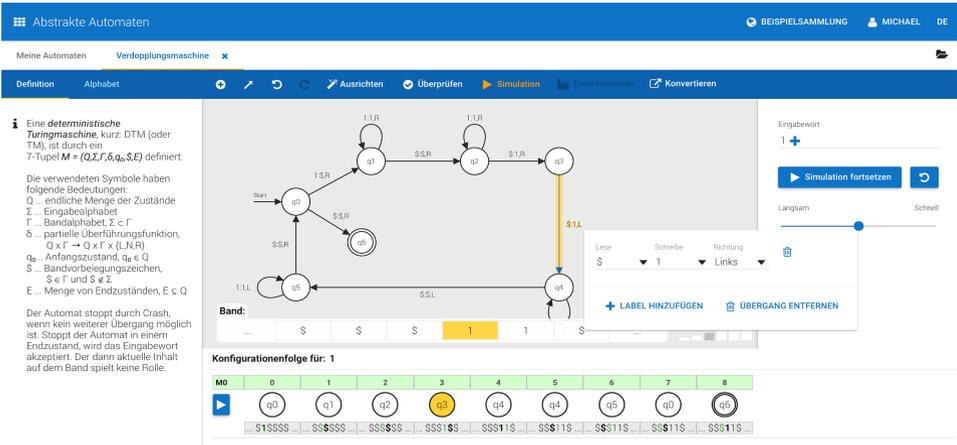


Abbildung 4: Simulation einer deterministischen Turingmaschine in FLACI

2.5 Compiler und Interpreter

Als Motivation für die Behandlung formaler Sprachen, abstrakter Automaten und zugehöriger Konzepte haben sich praktische Anwendungen im Compilerbau als tragfähig erwiesen. FLACI bietet in diesem Modul einen Editor zur Modellierung von Übersetzungsprozessen mit Hilfe von T-Diagrammen, s. Abbildung 5, sowie einen Editor zur Entwicklung von Compilern und Interpretern mit Hilfe einer formalen Beschreibungssprache (TDL: Translator Description Language). Compiler werden dabei nicht per Hand codiert, sondern mit Compilergeneratoren (Compiler Compiler) aus der formalen Definition automatisiert generiert. Als Ausgangspunkt kann modulübergreifend eine bereits erstellte formale Grammatik direkt verwendet werden, wobei automatisch passende Tokenklassen für den Scanner und Regeln für den Parser generiert werden. Für die Lernenden bleibt die Aufgabe, synthetische Attribute (S-Attribute) für jede Regel des Parsers als JavaScript-Code zu definieren, s. Abbildung 6. Der Scanner arbeitet mit regulären Ausdrücken, die zuvor erarbeitet werden müssen. Der Parser verwendet die gleiche Darstellung editierbarer Syntaxdiagramme wie das Grammatik-Modul von FLACI. Sowohl Scanner als auch Parser können simuliert und Zustandstabellen des generierten Compilers angezeigt werden. Enthält die Grammatik Mehrdeutigkeiten wird durch entsprechende Meldungen darauf hingewiesen. Alle erstellten Compiler und Interpreter lassen sich über eine URL als eigenständiges JavaScript-Programm in beliebige Web-Applikationen einbinden.

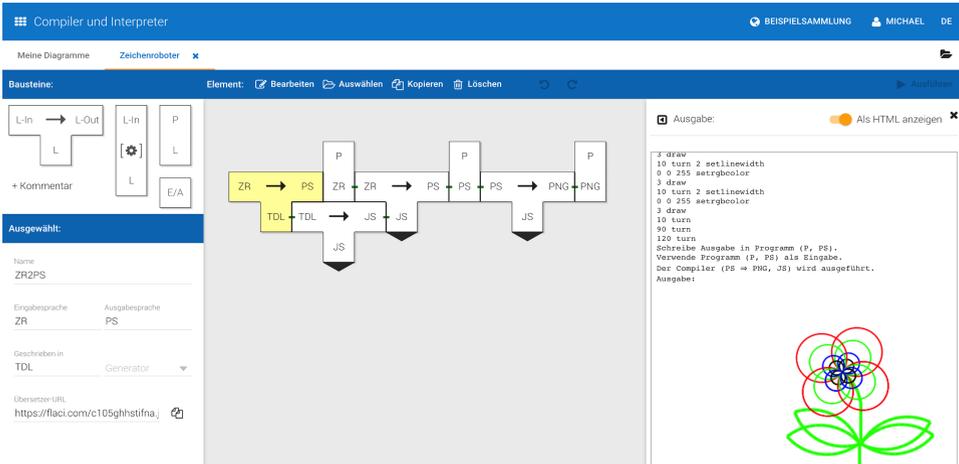


Abbildung 5: T-Diagramm für einen Übersetzungsprozess über mehrere Teilschritte und Ausführung (rechts)

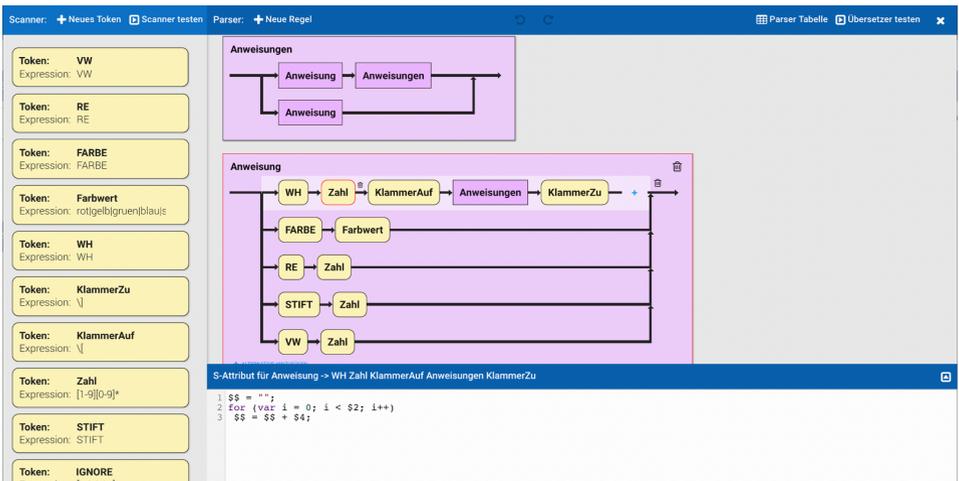


Abbildung 6: Compilerdefinition mit Scanner und Parser inklusive S-Attribute (unten)

2.6 Zusammenwirken der Module

Die erarbeiteten Grammatiken, Automaten und Compiler werden wahlweise mit Hilfe des lokalen Webbrowsers oder über den FLACI-Server (bei vorheriger Anmeldung) gespeichert. Die erstellten Produkte können auch als JSON-Dateien heruntergeladen, über Web-links auf dem FLACI-Server mit anderen Lernenden ausgetauscht oder von Lehrpersonen

Tabelle 1: Auszug aus Bildungsstandards für die Sek. II und passende Module aus FLACI

Grundlegendes und erhöhtes Anforderungsniveau	Modul in FLACI
<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ● vergleichen formale mit natürlichen Sprachen, ● untersuchen den Zusammenhang zwischen einer Grammatik und ihrer Sprache, leiten Wörter einer Sprache ab und stellen Ableitungsbäume dar, ● verwenden Sprachdefinitionen (z. B. Grammatiken, Syntaxdiagramme) zur Analyse, Beschreibung und Entwicklung formaler Sprachen, ● überführen Grammatiken in endliche Automaten und umgekehrt. 	<p>“Formale Sprachen”</p> <p>“Formale Grammatiken”</p> <p>“Reguläre Ausdrücke” “Formale Grammatiken”</p> <p>“Formale Grammatiken” “Abstrakte Automaten”</p>
<p>Erhöhtes Anforderungsniveau</p> <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ● erläutern den Zusammenhang zwischen Grammatiken, Sprachen und Automaten, ● analysieren und implementieren Programme zu Problemstellungen auf Kellerautomaten, Turingmaschinen oder Registermaschinen, ● erläutern prinzipielle und praktische Grenzen der Berechenbarkeit. 	<p>“Formale Grammatiken” “Abstrakte Automaten” “Abstrakte Automaten” “Compiler und Interpreter”</p>

4 Fazit

Beim Einsatz der Lernumgebung AtoCC wurden in mehr als 10 Jahren im Informatikstudium und -unterricht (Sekundarstufe II) positive didaktische Erfahrungen gemacht. Mehr als 25'000 Downloads der AtoCC-Software von Lehrenden und Lernenden an Schulen und Hochschulen lassen ein beachtliches Interesse erkennen. Die Weiterentwicklung zu FLACI als Webapplikation stellt einen wichtigen Schritt zur Fortführung des Ansatzes dar: FLACI berücksichtigt zukunftsfähige mobile Geräte und reduziert den Installationsaufwand in Schulen sowie auf privaten Geräten der Lernenden und der Lehrpersonen. Nach den überaus positiven Erfahrungen mit FLACI im Informatikstudium ermöglicht die Passfähigkeit mit den Rahmenvorgaben für den Informatikunterricht den Einsatz dieser

Lernumgebung auch im Schulkontext. Zu einer wissenschaftlichen Evaluation wird aufgerufen. FLACI steht unter www.flaci.com zur kostenlosen Nutzung zur Verfügung. Der gesamte Quellcode des Projekts wurde unter einer MIT-Lizenz auf [github](https://github.com) veröffentlicht.

5 Literatur

- [BHN03] Brändle, M.; Hartmann, W.; Nievergelt, J.; Reichert, R.; Schlatter, T.: Kara: Ein theoriebasierter Ansatz für Lernumgebungen zu fundamentalen Konzepten der Informatik. In INFOS 2003 (S. 201-210).
- [CSK11] Chakraborty, P.; Saxena, P. C.; Katti, C. P.: Fifty Years of Automata Simulation: A Review. In: *acm inroads*, 2011, Vol. 2, No. 4.
- [FPA07] Friedrich, S., Puhmann, H., Altdorf, L. G.: Bildungsstandards Informatik - von Wünschen zu Maßstäben für eine informatische Bildung. In: INFOS 2007, S. 21-32.
- [HW06] Hielscher, M., Wagenknecht, C.: AtoCC: learning environment for teaching theory of automata and formal languages. In *ITiCSE 2006*, Vol. 6, pp. 26-28.
- [HW07] Hielscher, M., Wagenknecht, C.: AtoCC-didaktischer Ort und erste Erfahrungen. In *INFOS 2007*, S. 159-170.
- [MHL16] Mane, D. T.; Howal, S. S.; Lokare, V. T.: Problem-based Learning using Simulation Tools for Automata Theory. In: *Journal of Engineering Education Transformations*, Special Issue, 2016.
- [PT18] Pereira, C. H., Terra, R.: A Mobile App For Teaching Formal Languages and Automata. In *Wiley Online Library*, Special Issue Article, 2018.
- [RBD16] Röhner, G.; Brinda, T.; Denke, V. u.a.: Bildungsstandards Informatik für die Sekundarstufe II. Beilage zu *LOG IN*, 36. Jg. 2016, Heft Nr. 183/184.
- [RF06] Rodger, S. H., Finley, T. W.: *JFLAP: An Interactive Formal Languages and Automata Package*. Jones & Bartlett Learning, 2006.
- [Ts04] Exorciser: Automatic Generation and Interactive Grading of Structured Exercises in the Theory of Computation. Doctoral dissertation, ETH Zurich, 2004.
- [WH14] Wagenknecht, C., Hielscher, M.: *Formale Sprachen, abstrakte Automaten und Compiler*. 2. Aufl., 2014, Springer Fachmedien Wiesbaden.