HERAKLIT-Fallstudie: Addierer

Peter Fettke[1,2][0000−0002−0624−4431] und Wolfgang Reisig[3][0000−0002−7026−2810]

[1] Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Saarbrücken, Deutschland
peter.fettke@dfki.de
[2] Universität des Saarlandes, Saarbrücken, Deutschland
[3] Humboldt-Universität zu Berlin, Berlin, Deutschland
reisig@informatik.hu-berlin.de

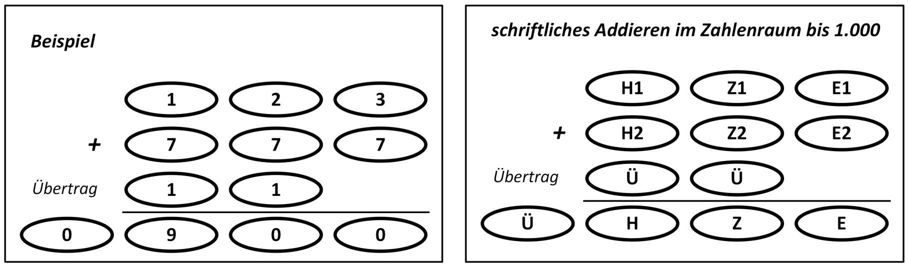
Abstract. This brief case study describes central concepts of HERAKLIT using the example of pencil-and-paper addition in the number space up to 1,000. Further sources explain how HERAKLIT can be used for modeling business application domains.

Zusammenfassung. Diese kleine Fallstudie beschreibt zentrale Konzepte von HERAKLIT am Beispiel des schriftlichen Addierens im Zahlenraum bis 1.000. Weitere Quellen erläutern, wie HERAKLIT für die Modellierung betrieblicher Anwendungsdomänen eingesetzt werden kann.

1 Einführung

Schulkinder lernen in Deutschland im Allgemeinen in der dritten Klasse das schriftliche Addieren im Zahlenraum bis 1.000. Der linke Teil der Abbildung 1 zeigt das typischerweise dabei erlernte Vorgehen anhand der Beispiels $123+777 = 900$. Im rechten Teil wird die Rechnung schematisch dargestellt; die Buchstaben E, Z und H stehen für die Einer-, Zehner- und Hunderterziffern der Summanden beziehungsweise der Summe, der Buchstabe $\ddot{U}$ für den Übertrag. Das Beispiel des schriftlichen Addierens dient in dieser Fallstudie zur Illustration der zentralen HERAKLIT-Konzepte. Dabei sind grundlegende Konzepte von Petrinetzen nützlich, wie sie in Kapitel 2 in [5] beschrieben werden.

zu zitieren als: FETTKE, P.; REISIG, W.: HERAKLIT- *Fallstudie: Addierer*. 2020. – HERAKLIT–Arbeitspapier, v1, 8. November 2020, http://www.heraklit.org



(a) Beispiel (b) Schema

Abb. 1: Schriftliches Addieren im Zahlenraum bis 1.000

2 Halbaddierer und Volladdierer

Die Grundoperation des schriftlichen Addierens besteht in der Addition zweier Zahlen kleiner als zehn. Diese Aufgabe übernimmt das Modul *Halbaddierer*, das Abbildung 2 als Petrinetz darstellt. Die beiden zu addierenden Zahlen sind als *symbolische Marken* auf die Plätze X und Y zu legen. Falls die Summe beider Zahlen kleiner zehn ist, ist Transition b aktiviert; andernfalls Transition a. Die Summe liegt nach dem Schalten der entsprechenden Transition auf dem Platz S; ein entstehender Übertrag auf $\ddot{U}$.

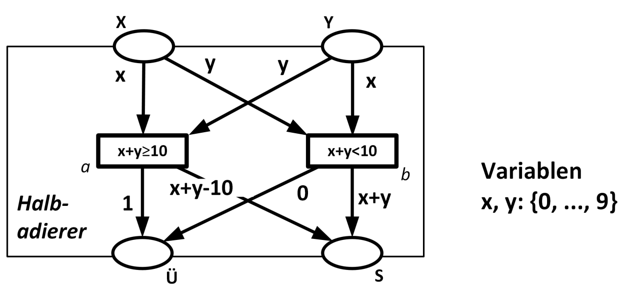


Abb. 2: Der Halbaddierer

Beim schriftlichen Addieren der Zehner- und Hunderterstellen ist gegebenenfalls ein Übertrag aus der Addition der geringerwertigen Stelle zu berücksichtigen; es werden also nicht nur zwei, sondern drei Zahlen addiert. Diese Aufgabe übernimmt der *Volladdierer*, den Abbildung 3 darstellt.

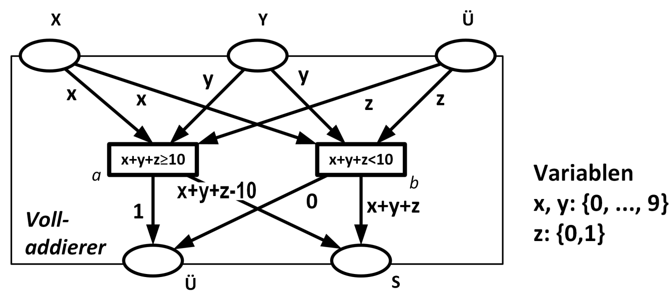


Abb. 3: Der Volladdierer

3 Module

Das Konzept des *Moduls* ist in Heraklit die Grundlage zur Strukturierung von Modellen. Ein Modul wird graphisch als Rechteck dargestellt und besteht aus einem *Inneren* sowie einer *Oberfläche*. Das Innere eines Moduls besteht im allgemeinen Fall aus seinen Namen, einer Beschreibung seines Verhaltens und weiteren Modulen. Die Elemente der Oberfläche werden als *Gates* bezeichnet und graphisch auf den Rand des Moduls gelegt. Jedes Gate eines Moduls trägt ein *Label*. Ein Modul wird *ausführbar* genannt, wenn das Innere des Moduls aus einem Petrinetz besteht.

Die Elemente der Oberfläche werden in zwei *Schnittstellen* eingeteilt, sodass ein Modul eine *linke* und eine *rechte Schnittstelle* besitzt. Zuweilen werden die Elemente der beiden Schnittstellen graphisch nicht jeweils an den linken beziehungsweise rechten Rand eines Moduls gelegt, sondern an den oberen beziehungsweise unteren Rand. Auch in diesem Fall wird trotzdem weiterhin von einer linken und rechten Schnittstelle gesprochen; bei dem in Abbildung 2 dargestellten Modul bilden also die Plätze X und Y die linke, die Plätze S und $\ddot{U}$ die rechte Schnittstelle des Moduls.

4 Abstrakte Module

Das Innere eines Moduls kann durch Anwendung des Abstraktionsoperators $[\cdot]$ auf seinen Namen reduziert werden; es bleiben nur die Elemente des Moduls übrig, welche auf dem Rand des Moduls lagen. Graphisch wird jedes Element der Oberfläche durch einen waagerechten beziehungsweise senkrechten Strich symbolisiert, an den das zugehörige *Label* notiert wird. Abbildung 4 zeigt die Module $[Halbaddierer]$ und $[Volladdierer]$, also die abstrakte Darstellung der beiden Module aus den Abbildungen 2 und 3.

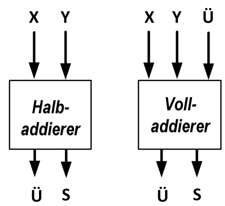
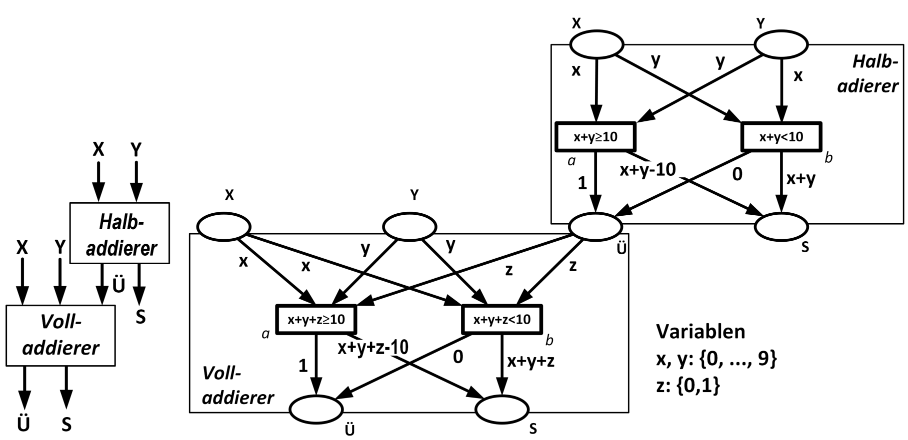


Abb. 4: Abstrakte Versionen des Halb- und des Volladdierers, algebraisch geschrieben als: $[Halbaddierer]$ und $[Volladdierer]$

5 Die Komposition von Modulen

Um zwei zweistellige Zahlen zu addieren, liegt es nahe, mit einem Halbaddierer die Einer- und mit einem Volladdierer die Zehner-Stellen zusammen mit dem Übertrag des Halbaddierers zu addieren. Dazu muss der Übertrag in der rechten Schnittstelle des Halbaddierers (in den Abbildungen 2 und 4 unten links dargestellt) verschmolzen werden mit dem Übertrag in der linken Schnittstelle des Volladdierers (in den Abbildungen 3 und 4 oben rechts dargestellt). Der Halbaddierer wird mit dem Volladdierer zu einem neuen Modul *komponiert* wie Abbildung 5 zeigt; in HERAKLIT schreiben wir dafür $Halbaddierer \bullet Volladdierer$ beziehungsweise $[Halbaddierer] \bullet [Volladdierer]$ für die abstrakte Fassung. Die allgemeine Definition des Kompositionsoperators $\bullet$ erläutern weiterführende Texte [4,1].



(a) abstrakt (b) ausführbar

Abb. 5: Komposition des Halb- und Volladdierers

Das komponierte Modul in Abbildung 5 hat noch einen Mangel: In der linken Schnittstelle kommen die Label X und Y und in der rechten Schnittstelle das Label S jeweils doppelt vor. Es fehlt also eine eindeutige Kennzeichnung der Gates für die Einer- und die Zehner-Werte der beiden zu addierenden Zahlen und des Ergebnisses. Dies wird erreicht mit einer *Umbenennung* der Gates, also mit einer Ersetzung der Label durch andere. Mit den naheliegenden Notationen E, Z und $\ddot{U}$ für Einer, Zehner und Übertrag zeigt Abbildung 6 eine Umbenennung der Gates des Halbaddierers, die ihn zum Addierer $Z10$ macht, der also Zahlen addiert, die kleiner als 10 sind.

Solche Umbenennungen werden zuweilen gebraucht. Wie sie gebildet werden und wie man damit das Modul $Z10$ konstruiert, zeigt der folgende Abschnitt.

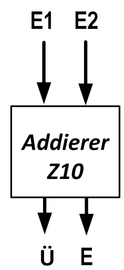


Abb. 6: umbenannter Halbaddierer

6 Umbenennungsmodule

Gates von HERAKLIT-Modulen werden zuweilen umbenannt; technisch besorgen spezielle, einfach geformte Module das Umbenennen. Abbildung 7 zeigt beispielhaft das Umbenennungsmodul, das das Label *alt* in das Label *neu* umbenennt.

Die Abbildung zeigt auch einige graphische Konventionen, die Umbenennungsmodule graphisch und notationell vereinfachen. Das in der Abbildung gezeigte Abbildungsmodul wird algebraisch geschrieben als: $alt \rhd neu$.

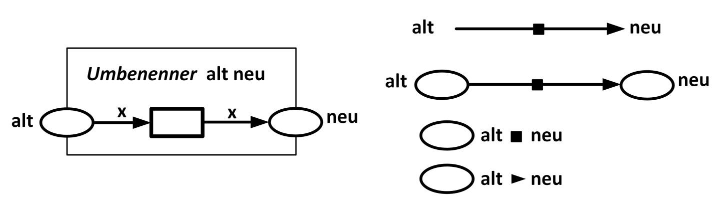


Abb. 7: Ein Umbenennungsmodul und seine Kurzschreibweisen

Als Beispiel komponiert die Abbildung 8 die Umbenennungsmodule $E1 \rhd X$, $E2 \rhd Y$ und $S \rhd E$ sowie den $Halbaddierer$ zum Modul $Z10$.

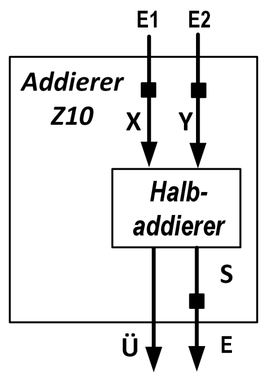


Abb. 8: Der Halbaddierer mit den Umbenennungsmodulen

7 Das Gesamtsystem

Mit den oben beschriebenen Techniken können wir nun leicht das Modul $Z10$ zum Modul $Z100$ und dieses wiederum zum Modul $Z1000$ erweitern. Für den $Z100$ verwenden wir die Komposition eines Halbaddierers und eines Volladdierers wie in Abbildung 5 und benennen die Gates geeignet um. Abbildung 9 zeigt das Resultat. Für das Modul $Z1000$ ergänzen wir das Modul $Z100$ um einen weiteren Volladdierer sowie geeigneter Umbenennungsmodule und es entsteht Abbildung 10. Das Innere des auf diese Weise komponierten Gesamtsystems zeigt Abbildung 11.

Die gebildeten Module können nicht nur graphisch, sondern auch algebraisch konstruiert werden:

$$\begin{aligned}
Z10 &:= (E1 \rhd X) \bullet (E2 \rhd Y) \bullet Halbaddierer \bullet (S \rhd E)\\
Z100 &:= Z10 \bullet (Z1 \rhd X) \bullet (Z2 \rhd Y) \bullet Volladdierer \bullet (S \rhd Z)\\
Z1000 &:= Z100 \bullet (H1 \rhd X) \bullet (H2 \rhd Y) \bullet Volladdierer \bullet (S \rhd H).
\end{aligned}$$

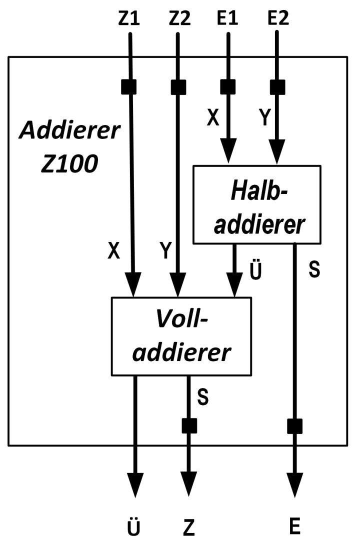


Abb. 9: Der Addierer Z100

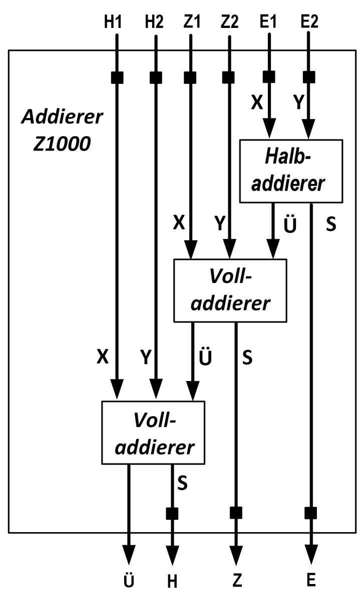


Abb. 10: Der Addierer Z1000

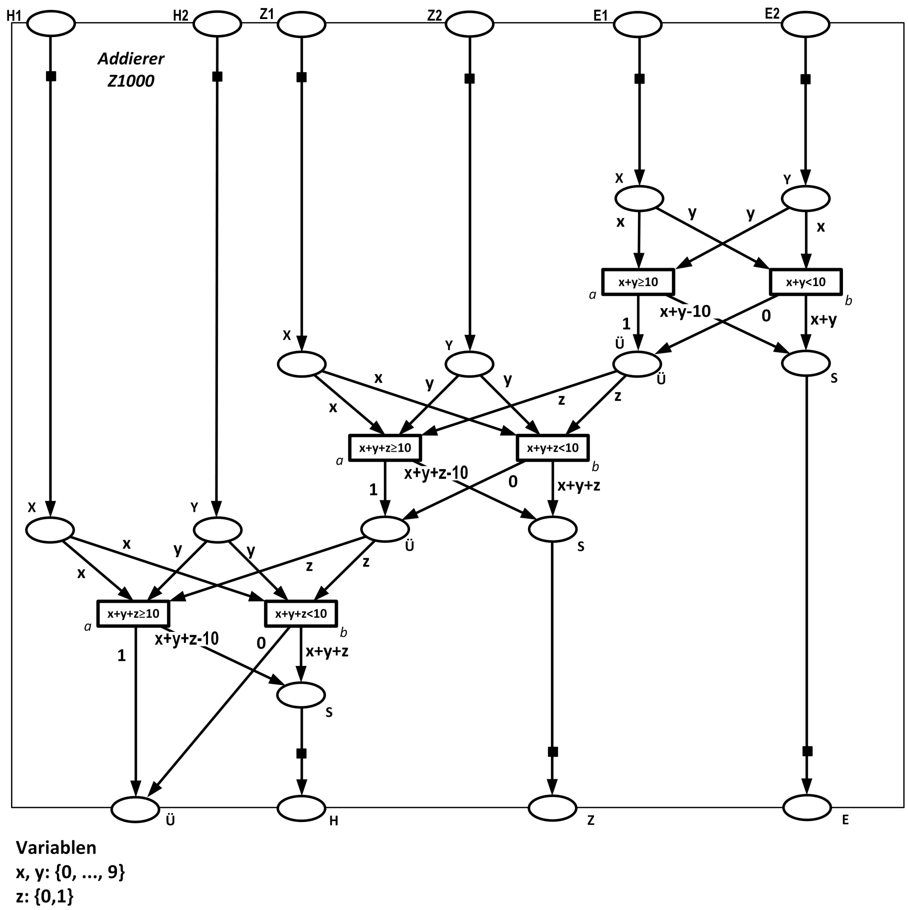


Abb. 11: Das Innere des Gesamtsystems

8 Ein verteilter Ablauf

Abbildung 12 zeigt einen *verteilten Ablauf* für die eingangs angeführte Berechnung 123 + 777, wiederum dargestellt als ein Petrinetz. Die Ziffern der beiden Summanden werden auf die entsprechenden Plätze der linken Schnittstelle des Moduls *Addierer* $Z1000$ gelegt, die Plätze an der rechten Schnittstelle enthalten dann das Ergebnis der Berechnung. Der in der Abbildung dargestellte verteilte Ablauf besteht aus verschiedenen *Aktionen*. Unter einer Aktion wird eine Transition und der zugehörige Vor- und Nachbereich verstanden. Beispiel: Die Transition b des Halbaddierers mit den Plätzen X und Y als Vor- und den Plätzen $\ddot{U}$ und S als Nachbereich bildet eine Aktion.

Aus der Struktur des verteilten Ablaufs ist ersichtlich, dass erst die Einer-, dann die Zehner- und schließlich die Hunderterziffern addiert werden. Diese Ordnung zwischen den entsprechenden Aktionen wird bei allen Additionen eingehalten, indem über die Plätze für die Überträge eine Synchronisation der einzelnen Addierer erfolgt. Falls eine andere Ordnung gewählt wird, ist die Korrektheit des Ergebnisses nicht gesichert, sodass das Einhalten dieser Ordnung konstitutiv für diesen Ablauf ist. Wie ein Addierer konstruiert werden kann, der eine schwächere Ordnung zwischen den Aktionen erlaubt, wird in [3] erläutert.

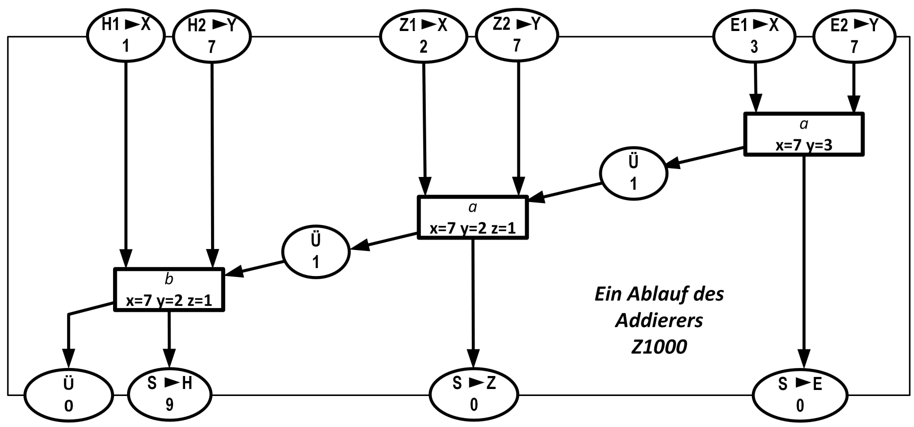


Abb. 12: Verteilter Ablauf für die Addition 123 + 777

9 Eine Modulvariante

Das Modulkonzept ermöglicht neben der Abstraktion und der Verfeinerung auch alternative Realisierungen des Inneren eines Moduls. Abbildung 13 zeigt eine solche Alternative in abstrakter und konkreter Form. Die Modulvariante beruht auf der Idee, dass ein Halbaddierer als ein Volladdierer verstanden werden kann, wobei kein Übertrag vorhanden ist, also der Wert 0 verwendet wird. Folglich kann ein Halbaddierer aus dem Modul des Volladdierers und des Moduls leerer Übertrag komponiert werden. Das Modul *leerer Übertrag* hat nur die Aufgabe, an seine Oberfläche die Ziffer Null abzulegen.

Die hier dargestellte Variante hat dieselbe Oberfläche, aber ein unterschiedliches Inneres wie das in Abbildung 2 beschriebene Modul. Darüber hinaus produzieren beide Module bei gleicher Startbelegung der Plätze X und Y die gleiche Endbelegung auf den Plätzen S und $\ddot{U}$ und sind in diesem Sinne äquivalent.

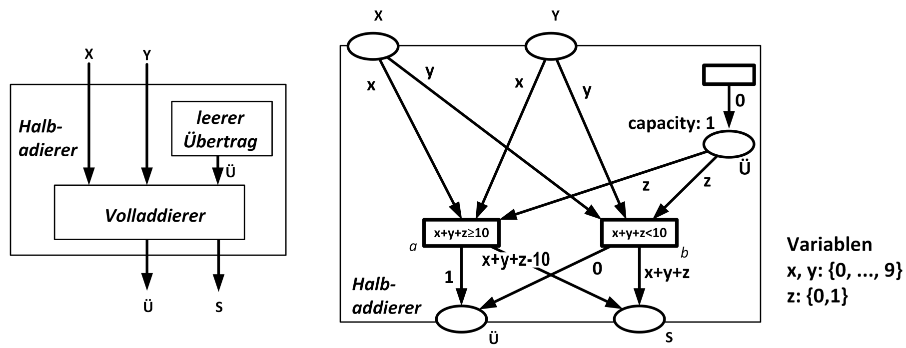


Abb. 13: Eine Variante des Halbaddierers

10 Zusammenfassung und Ausblick

Diese Fallstudie zeigt die zentralen Heraklit-Konzepte an einem kleinen, wenig typischen Beispiel. Intendierte Anwendungen von Heraklit liegen unter anderem im Bereich rechnerintegrierter betrieblicher Systeme realer Größenordnung. Im Rahmen der Heraklit-Infrastruktur entstehen fortlaufend neue Beispiele und Fallstudien; dazu gehören:

- Heraklit-Webseite (`http://www.heraklit.org`),
- Heraklit-Handbuch [4],
- Fallstudie zur Modellierung eines Service-Systems [2],
- Fallstudie zur Modellierung eines Handelsbetriebs [1].

Literatur

1. Fettke, P. ; Resig, W. : Modellieren mit Heraklit – Tutorium und Fallstudie (im Erscheinen). (2020)
2. Fettke, P. ; Resig, W. : Modelling service-oriented systems and cloud services with Heraklit. In: *CoRR* abs/2009.14040 (2020). `http://arxiv.org/abs/2009.14040`. – presented at the 16th International Workshop on Engineering Service-Oriented Applications and Cloud Services, Heraklion, Greece, September 28-30, 2020
3. Fettke, P. ; Resig, W. : Ein nebenläufiger Addierer (in Vorbereitung). (2020)
4. Fettke, P. ; Resig, W. : Modellieren mit Heraklit – Handbuch (in Vorbereitung). (2021)
5. Reisig, W. : *Petrinetze – Modellierungstechnik, Analysemethoden, Fallstudien.* Vieweg+Teubner, 2010