Zur Methodik und Anwendung fehlerverdeckender Steuerungsrekonfiguration für eine Klasse ereignisdiskreter Systeme

Der Technischen Fakultät der Friedrich-Alexander Universität Erlangen-Nürnberg

zur Erlangung des Doktorgrades Dr.-Ing. vorgelegt von

Thomas Wittmann

aus Tirschenreuth

Als Dissertation genehmigt von der Technischen Fakultät
der Friedrich-Alexander Universität Erlangen-Nürnberg

Tag der mündlichen Prüfung:	24. Juli 2014
Vorsitzende des Promotionsorgans:	Prof. Dr.-Ing. habil. Marion Merklein
Gutachter:	Prof. Dr.-Ing. Thomas Moor
	Prof. Dr.-Ing. Georg Frey

Danksagung

Diese Arbeit hatte bis zum Ende viele Unterstützer und Wegbegleiter.

An erster Stelle möchte ich meinem Betreuer Prof. Dr.-Ing. Thomas Moor danken. Er stand mir mit Rat und Tat zur Seite und seine Inspiration war das Fundament meiner Arbeit. Unter seiner Leitung wurde nicht nur meine Einstellung zur ingenieurswissenschaftlichen Forschung entscheidend geprägt, sondern auch meine menschliche Entwicklung.

Weiterhin bedanke ich mich bei Herrn Prof. Dr.-Ing. Georg Frey für die Übernahme des Korreferats, Herrn Prof. Dr.-Ing. habil. Günter Roppenecker und Herrn Prof. Dr. Christoph Pflaum für die bereitwillige Mitwirkung im Prüfungskollegium.

Ich danke den Kolleginnen und Kollegen am Lehrstuhl für Regelungstechnik für eine Arbeitsatmosphäre, die jeder wissenschaftlichen Tätigkeit nur zuträglich sein kann. Besonders erwähnen möchte ich meinen Kollegen und Freund Dipl.-Ing. Andreas Michalka, dessen Ansichten und Ratschläge mir stets Anregung und Hilfe waren, meine Kollegin Dipl.-Ing. Christine Baier, auf deren fachliche Meinung ich zählen konnte und Dr.-Ing. Jan H. Richter, dessen Unterstützung mir bei methodischen und praktischen Problemen sicher war.

Diese Arbeit entstand im Zuge einer Forschungkooperation des Lehrstuhls für Regelungstechnik mit der Siemens AG. Ich danke Prof. Dr.-Ing. Thomas Moor und Dr.-Ing. Jan H. Richter für die Initiierung des Projekts, der Siemens AG für die finanzielle Unterstützung und Herrn Ulrich Welz für sein Interesse an meiner Arbeit und meiner Person.

Zuletzt danke ich meinen Eltern und meiner Schwester. Ohne ihre moralische Unterstützung wäre diese Arbeit nicht fertiggestellt worden.

Erlangen, im Januar 2013 Thomas Wittmann

Kurzzusammenfassung

In der gängigen Praxis der automatisierten Fertigung werden die Steuerungsprogramme von Experten geschrieben und im Versuch validiert. Nachträgliche Änderungen an bereits getesteter Software bergen Risiken, die einer nachträglichen Berücksichtigung von Komponentenfehlern entgegenstehen. Als eine spezielle Ausprägung der fehlertoleranten Regelung liefert die *fehlerverdeckende Steuerungsrekonfiguration* einen konzeptionellen Rahmen, in dem fehlerbedingte Änderungen der Prozessdynamik in den Reglerentwurf miteinbezogen werden können. Herauszustellen ist, dass ein bestehendes Steuerungssystem unverändert im rekonfigurierenden Regelkreis erhalten werden kann und bis zum Auftreten eines Fehlers die Kontrolle über den Prozess behält.

Das Konzept der fehlerverdeckenden Steuerungsrekonfiguration sieht vor, zwischen Nominalregler und fehlerbehafteter Strecke einen *Rekonfigurator* zu schalten. Seine Aufgabe ist es, den Signalfluss zwischen Nominalregler und fehlerbehafteter Strecke so zu modifizieren, dass der rekonfigurierende Regelkreis zulässiges Verhalten aufweist, während die Auswirkung eines Fehlers vor dem Nominalregler *verdeckt* wird.

Das Konzept der fehlerverdeckenden Steuerungsrekonfiguration wurde anhand von linearen zeitkontinuierlichen Systemen, siehe [Ste05], entwickelt und wird in dieser Arbeit auf ereignisdiskrete Systeme übertragen. Die Dynamik ereignisdiskreter Systeme ist durch spontane Zustandswechsel charakterisiert, die mit einem nach außen hin sichtbaren Ereignis einhergehen. Technische Systeme, die einer ereignisdiskreten Modellierung zugänglich sind, finden sich z. B. in der Fertigungsautomatisierung. Auf Basis der Supervisory Control Theory, siehe [RW87, RW89], werden der Entwurf fehlerverdeckender, ereignisdiskreter Rekonfiguratoren diskutiert und die Ergebnisse anhand einer Machbarkeitsstudie validiert.

Abstract

Common practice in manufacturing automation is that control programs are written by experts and that their functionality is validated by experiments. Thus, these programs are of considerable value and their modification is undesired. Extending an existing real-world control system by fault-tolerant control capabilities is therefore a difficult task.

Fault-hiding control reconfiguration is a specialised approach to fault-tolerant control. It provides a methodological framework in which changes in the process dynamics following a fault, can be treated, so that an existing controller remains active until the occurrence of a fault. In other words, fault-tolerant control algorithms based on the fault-hiding paradigm can be retrofitted to existing control systems, without changing them.

In principle a *reconfigurator* is placed between controller and faulty plant. Its task is to modify the information flow between controller and plant, so that the behaviour of the reconfiguring closed-loop system is admissible, while the impact of a fault is *hidden* from the controller.

The paradigm of fault-hiding was developed for linear time-continuous systems, see [Ste05]. In this work, systems, whose dynamics are characterised by spontaneously occurring state changes that go along with the occurrence events, so called discrete event systems, are considered. Based on Supervisory Control Theory, see [RW87, RW89], the design of fault-hiding reconfigurators is discussed and the outline of a proof-of-concept study from the field of manufacturing automation is provided.

Inhaltsverzeichnis

Einleitung

Lässt man die Annahme fehlerfrei arbeitender technischer Systeme fallen, stellt sich die Frage nach ihrer *Verlässlichkeit*, d. h. nach ihrer Funktionalität im Fehlerfall, vgl. [Bir14]. Die Regelungstechnik liefert mit der *fehlertoleranten Regelung* einen Beitrag zur Beantwortung dieser Frage. Sie stellt Methoden zur Analyse und Synthese von Regelkreisen bereit, in denen das Auftreten von Fehlern explizit berücksichtigt wird. Fehlertolerante Regelungssysteme weisen so eine erhöhte Verlässlichkeit auf. Zahlreiche Buchveröffentlichungen, exemplarisch [Bir14, BKL$^+$06], und Konferenzen, in jüngerer Zeit die SAFEPROCESS 2012 und die DCDS 2013, zeigen die Bedeutung der fehlertoleranten Regelung, auch im Bereich ereignisdiskreter Systeme.

In der Fertigungsautomatisierung ist es gängige Praxis, dass Steuerungssoftware von Experten erstellt wird. Solche Programme gegen einen modellbasierten Regler auszutauschen ist unerwünscht, einerseits wegen des eingeflossenen Expertenwissens, andererseits weil Änderungen an einem funktionsfähigen System grundsätzlich Risiken bergen. Die *fehlerverdeckende Steuerungsrekonfiguration* stellt den konzeptionellen Rahmen für den Entwurf einer Schnittstelle zwischen einem Steuerungsprogramm und einem modellbasierten fehlertoleranten Regler bereit. Ein bestehendes Steuerungssystem kann so zu einer fehlertoleranten Regelung erweitert werden, ohne dass das bestehende Steuerungsprogamm ersetzt werden muss.

Bemerkung 1. Der Begriff *Steuerung* wird nicht in der systemtheoretisch geprägten Bedeutung einer offenen Wirkungskette (in Abgrenzung zu einer Regelung) verwendet, sondern bezieht sich auf ein Steuergerät, beispielsweise auf eine speicherprogrammierbare Steuerung. Ein „gesteuertes System“ – oder ein „Steuerungssystem“ – bezeichnet in diesem Sinn einen physikalischen Prozess, dessen Verhalten über ein Steuergerät festgelegt wird. Über die Eigenschaften des gesteuerten Prozesses wird an dieser Stelle keine Aussage getroffen. □

Ausgangspunkt für die Betrachtungen in dieser Arbeit ist der *nominelle Regelkreis*, siehe

Abbildung 1 links, d. h. der Regelkreis unter der Annahme, dass keine Fehler auftreten. Für den Entwurf des Nominalreglers werden im Streckenmodell keine Fehler berücksichtigt. Tritt ein Fehler auf, ändert sich das Streckenverhalten und der Nominalregler kann die Entwurfsvorgaben nicht mehr umsetzen. Durch die explizite Berücksichtigung eines Fehlers im Streckenmodell und den Entwurf eines entsprechenden *fehlertoleranten Reglers* können fehlerinduzierte Verhaltensänderung der Strecke berücksichtigt und die Entwurfsvorgaben im Fehlerfall umgesetzt werden. Abbildung 1 mitte zeigt den fehlertoleranten Regelkreis. Für den Entwurf eines fehlertoleranten Reglers wird von einem freien System ausgegangen. Besteht aber schon eine nominelle Regelung, kann ein dynamisches System, der sogenannte *Rekonfigurator*, zwischen Nominalregler und fehlerbehafteter Strecke platziert werden, siehe Abbildung 1 rechts. Seine Aufgabe ist einerseits das Umsetzen der Nominalreglereingaben auf, für die fehlerbehaftete Strecke sinnvolle, Kommandos. So gesehen erfüllt der Rekonfigurator die Aufgaben eines fehlertoleranten Reglers. Andererseits müssen die Streckenausgaben so verändert werden, dass die Auswirkungen eines Fehlers vor dem Nominalregler *verdeckt* werden. Auf diese Weise verbleibt der Nominalregler im Regelkreis, während die fehlerbehaftete Strecke akzeptables Verhalten aufweist.

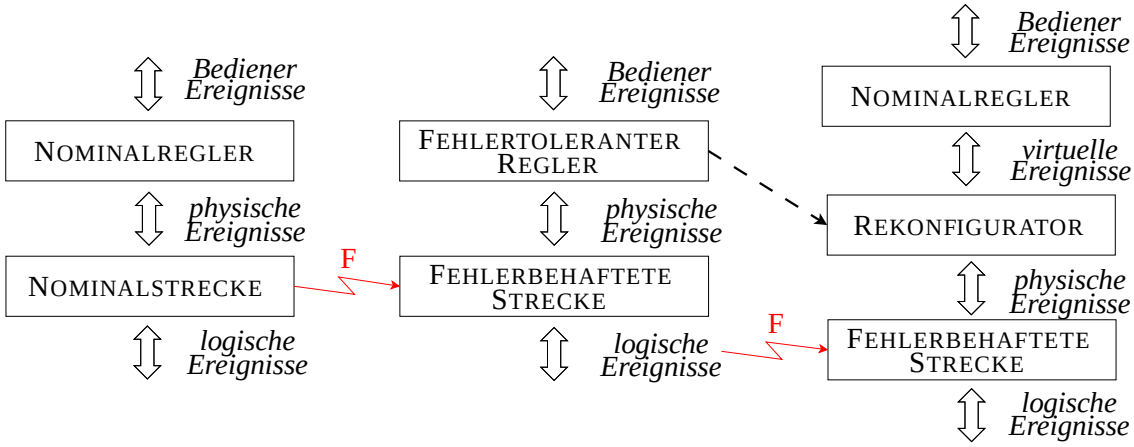


Abbildung 1: Übersicht

Die fehlerverdeckende Steuerungsrekonfiguration wurde anhand linearer, zeitkontinuierlicher Systeme entwickelt, siehe [Ste05] und soll in dieser Arbeit auf ereignisdiskrete Systeme angewandt werden. Im Folgenden werden zunächst einige Charakteristika ereignisdiskreter Systeme wiederholt, bevor eine Übersicht über bestehende Ansätze zur fehlertoleranten Regelung gegeben wird. Anschließend wird das Nominalentwurfsproblem diskutiert und ein Ansatz zur fehlertoleranten Regelung ereignisdiskreter Systeme vorgestellt. Abschließend wird auf das Problem der fehlerverdeckenden Steuerungsrekonfiguration eingegangen.

Regelung ereignisdiskreter Systeme

Ereignisdiskrete Systeme entwickeln sich über einer diskreten Zählachse mit dem spontanen Auftreten von *Ereignissen*. Die Reihenfolge und die exakten Eintrittszeitpunkte der Ereignisse bestimmen dabei unterlagerte und nicht notwendigerweise modellierte Mechanismen. Dazu zählen äußere Einflüsse, z. B. menschliche Bediener, genauso wie vernachlässigte kontinuierliche Dynamik. Technische Beispiele finden sich z. B. mit Kommunikationsprotokollen, Verkehrsleitsystemen oder automatisierten Fertigungssystemen.

Formal können ereignisdiskrete Systeme als Objekte mit einem Zustandsraum, einem Ereignisvorrat, einem Anfangszustand und einer Transitionenrelation aufgefasst werden. Ausgehend vom Anfangszustand generiert ein ereignisdiskretes System eine nach außen hin sichtbare Ereigniskette und führt gleichzeitig entsprechende Zustandswechsel aus. Abhängig vom aktuellen Zustand legt die Transitionenrelation die erlaubten Ereignisse, nebst den entsprechenden Folgezuständen, fest.

Ereignisse sind in ihrer zeitlichen Reihenfolge durch die bereits erwähnten unterlagerten Wirkzusammenhänge exakt festgelegt. Trotzdem können in ein und demselben Zustand des ereignisdiskreten Systems mehrere Ereignisse erlaubt sein. In diesem Sinn sind ereignisdiskrete Modelle *nicht-deterministisch*.

Mit formalen Sprachen und endlichen Automaten stellt die Informatik einen passenden Rahmen zur Modellierung ereignisdiskreter Systeme bereit. Jedem Automaten werden eine *generierte* und eine *markierte* Sprache zugeordnet. Sie repräsentieren das nach außen hin sichtbare Verhalten des ereignisdiskreten Systems, während die Transitionenrelation Aufschluss über dessen interne Struktur gibt. Die generierte Sprache stellt dabei ein Modell der schrittweisen Entwicklung des ereignisdiskreten Systems dar. Auf Grund der erwähnten unterlagerten Wirkzusammenhänge können *Letztendlichkeitseigenschaften*, d. h. die Tendenz ausgezeichnete Systemkonfigurationen anzustreben, bestehen. Diese können durch *gültige* Zeichenketten, d. h. in der markierten Sprache, berücksichtigt werden. In einer entsprechenden Automatenrepräsentation enden gültige Zeichenketten in *markierten* Zuständen. Repräsentieren alle Zustände eine ausgezeichnete Systemkonfiguration, dann besitzt das System lediglich *triviale* Letztendlichkeitseigenschaften.

Beispiel 1. Betrachtet wird eine einfache Maschine. Das Ereignis *a* zeigt einen Startvorgang an, während die Ereignisse *A1* und *A2* mit dem Ausstoß eines Werkstücks einhergehen.

Abbildung 2 wird dabei wie folgt interpretiert. Nachdem ein Ereignis *a* sichtbar wird,

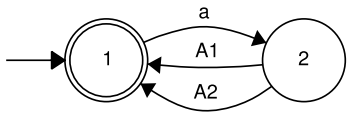


Abbildung 2: Einfache Maschine

beginnt die Maschine zu arbeiten. Sie beendet ihre Arbeit nach einer bestimmten Zeit und stößt dann ein Werkstück entweder von Typ 1 oder Typ 2 aus. Typ und Ausstoßzeitpunkt des ausgestoßenen Bauteils hängen von versteckten und nicht näher spezifizierten Mechanismen ab.

In diesem Modell sind nur Zeichenketten gültig, die mit *A1* oder *A2* enden. Damit wird eine Letztendlichkeitseigenschaft modelliert, konkret, das Bestreben der Maschine nach Beginn eines Fertigungsprozesses das fertige Bauteil auszustoßen. □

Supervisory Control Theory und Standardentwurfsproblem

Aufbauend auf formalen Sprachen und endlichen Automaten wurde mit der *Supervisory Control Theory*, siehe [RW87, RW89], ein Rahmen für den Entwurf ereignisdiskreter Regler entwickelt. Dabei wurde den Eigenheiten ereignisdiskreter Systeme Rechnung getragen.

Ereignisdiskrete Regler können keinen Einfluss auf die unterlagerten Wirkzusammenhänge nehmen. Ihre Eingriffsmöglichkeiten sind daher passiv, d. h. auf das Verbieten ausgezeichneter Ereignisse beschränkt. Alle anderen Ereignisse müssen zugelassen werden, so sie denn die Strecke erlaubt. Diese grundlegende Anforderung wird als *Steuerbarkeit* bezeichnet. Kann ein Ereignis durch Regelung unterdrückt werden, heißt es *steuerbar* und im anderen Fall *nicht-steuerbar*.

Beispiel 2. Während der Start der Maschine durchaus verboten werden kann, sind die Ereignisse *A1* und *A2* durch interne Mechanismen festgelegt; sie sind also nicht-steuerbar. Es ist also lediglich das Ereignis *a* steuerbar. □

Ereignisdiskrete Regler müssen die Letztendlichkeitseigenschaften der Strecke respektieren, d. h. die Strecke darf nicht daran gehindert werden, ausgezeichnete Systemkonfiguration zu erreichen. Ist das der Fall, dann heißen Strecke und Regler *konfliktfrei*. Ist zudem der geschlossene Regelkreis stets in der Lage, ein weiteres Ereignis zu generieren, heißt er in Übereinstimmung mit relevanter Literatur, siehe [KGM92], *vollständig*. Sind Regler und Strecke konfliktfrei und der geschlossene Regelkreis vollständig, so heißt der geschlossene Regelkreis *lebendig*.

Neben einem lebendigen und steuerbaren Regelkreis wird als zusätzliches Entwurfsziel das Einhalten einer gegebenen Sicherheitsspezifikation gefordert. In ihr sind alle *erlaubten* Ereignisfolgen hinterlegt. Erlaubt die Strecke weitere Ereignisfolgen, dann sind diese, unter Berücksichtigung der Lebendigkeit und Steuerbarkeit des geschlossenen Regelkreises, zu verbieten. Das gilt insbesondere auch dann, falls die Strecke *nicht-beobachtbare*, d. h. für den Regler nicht sichtbare, Ereignisse aufweist.

Die Synthese ereignisdiskreter Regler zielt auf einen steuerbaren und lebendigen Regelkreis, wobei die Strecke in ihrem Verhalten möglichst wenig eingeschränkt wird. Ein solcher Regler existiert eindeutig und wird als *minimal-restriktiv* bezeichnet.

Im Rahmen der Supervisory Control Theory sind Steuerbarkeit und Lebendigkeit unter Spezifikationseinschluss elementare Entwurfsziele. Entsprechend wird in dieser Arbeit ein Regelungssystem *fehlertolerant* genannt, falls der geschlossene Regelkreis steuerbar und lebendig ist, unabhängig von dem Auftreten eines Fehlers. In der Literatur sind weitere Interpretationen des Begriffs „fehlertolerant“ und Ansätze zur fehlertoleranten Regelung bekannt.

Fehlertolerante Regelung

Tritt ein *Fehler* auf, dann ändert die Strecke ihr Verhalten, sodass die Eingriffe des Nominalreglers nicht mehr zulässig sind. *Fehlertolerante Regler* berücksichtigen fehlerbedingte Verhaltensänderungen und erhöhen so die Verlässlichkeit des geregelten Systems. In der Regelungstheorie sind mit der *robusten Regelung* und der *adaptiven Regelung* bereits zwei Methoden bekannt, die für den Entwurf fehlertoleranter Regler geeignet sind, vgl. [BKL$^+$06].

Robuste Regelung. In der robusten Regelung wird der Reglerentwurf unter Berücksichtigung von Modellunsicherheiten betrachtet. Fasst man Fehler als Modellunsicherheit auf, können robuste Regler zur fehlertoleranten Regelung eingesetzt werden. Allerdings eignen sich nur wenige Beispiele für eine robuste fehlertolerante Regelung, vgl. [BKL$^+$06]. Zudem arbeitet ein fehlertoleranter, robuster Regler für die Nominalstrecke suboptimal. Allgemeine Ansätze zur robusten Regelung ereignisdiskreter Systeme werden in [BLW05, Tak00, CK99] berichtet. In [PL99] wird die Anwendung robuster Regler in der fehlertoleranten Regelung diskutiert.

Adaptive Regelung. In der adaptiven Regelung werden Strecken mit variierenden Parametern betrachtet. Durch ein Identifikationsverfahren werden die Streckenparameter fort-

laufend geschätzt und die Reglerparameter entsprechend angepasst. Äußern sich Fehler durch Parameteränderungen, können adaptive Regler zur fehlertoleranten Regelung eingesetzt werden. Ein Ansatz zur adaptiven und robusten Regelung ereignisdiskreter Systeme wird in [Lin93] berichtet.

Neben der robusten und adaptiven Regelung sind in der Literatur weitere Konzepte bekannt, die sich zur fehlertoleranten Regelung eignen.

Sprachenstabilität. In [WKHL08] wird ein System als fehlertolerant bezeichnet, falls es nach dem Auftreten eines Fehlers letztendlich wieder Nominalverhalten aufweist. Dazu wird auf den in [KGM93] eingeführten Stabilitätsbegriff zurückgegriffen. Der Entwurf von, in diesem Sinn, fehlertoleranten Reglern wird in [WHK08] diskutiert. In diesem Rahmen können nur Fehler betrachtet werden, die nach einer endlichen Anzahl von Ereignissen nicht mehr wirksam sind.

Steuerungsrekonfiguration. In [BKL$^+$06] wird vorgeschlagen, die Steuerung im Fehlerfall zu *rekonfigurieren*. Unter einer Steuerungskonfiguration wird dann eine bestimmte Kombination von Entwurfsparametern (Streckenmodell und Spezifikation) verstanden, siehe [KT12]. Der Wechsel von einer Steuerungskonfiguration in eine andere wird als *Steuerungsrekonfiguration* bezeichnet. Adaptionen dieses Konzepts für ereignisdiskrete Systeme finden sich in [Nke13, PSL11, KT12]. Daneben wird in [Sch12] der Reglerentwurf für rekonfigurierbare Werkzeugmaschinen erläutert und darauf aufbauend auch die fehlertolerante Steuerungsrekonfiguration, siehe [SS13].

Auch in dieser Arbeit wird konzeptionell auf eine Variation der fehlertoleranten Steuerungsrekonfiguration zurückgegriffen.

Fehlerverdeckende Steuerungsrekonfiguration

Die fehlerverdeckende Steuerungsrekonfiguration ist eine spezielle Ausprägung der fehlertoleranten Steuerungsrekonfiguration. Sie erlaubt es, ein bestehendes Steuerungssystem zu einer fehlertoleranten Regelung zu erweitern. Die nominelle Regelungsstrategie kann dann bis zum Auftreten eines Fehlers umgesetzt werden. Wurde die Steuerung möglicherweise von Hand programmiert, so können trotzdem formale Aussagen über den geschlossenen Regelkreis getroffen werden, solange es sich um eine zulässige Implementierung eines Nominalreglers handelt, wovon in dieser Arbeit ausgegangen wird. Aus diesen beiden Punkten ergibt sich die besondere praktische Relevanz dieses Konzepts. In der Literatur finden sich Diskussionen der fehlerverdeckenden Steuerungsrekonfiguration

für lineare Systeme [Ste05] und ausgewählte Klassen nicht-linearer [Ric11] und ereignisdiskreter Systeme [WRM13]. Als Vorbereitung auf einen Abriss der fehlerverdeckenden Steuerungsrekonfiguration soll zunächst die fehlertolerante Steuerungsrekonfiguration nach [BKL+06], bzw. ihre Umsetzung für ereignisdiskrete Systeme, siehe [PSL11], kurz beschrieben werden.

Im Vorfeld werden alle relevanten Fehlerfälle identifiziert und entsprechende Modelle der defekten Strecke erstellt. Auf dieser Basis wird eine Reglerbank berechnet, in der für jeden Fehlerfall ein geeigneter Regler vorhanden ist. Tritt im laufenden Betrieb ein Fehler auf, ist dieser zunächst mit Hilfe einer *Diagnoseeinrichtung*, siehe z. B. [SSL95], zu bestimmen. Mit dem aufgetretenen Fehler ist dann auch der entsprechende Regler bekannt. Mittels einer Umschalteinrichtung wird der Anfangszustand des ausgewählten Reglers mit dem aktuellen Streckenzustand abgeglichen und anschließend der Nominalregler ersetzt. Die entstandene Reglerbank und der Umschaltmechanismus werden zusammen als *rekonfigurierbarer Regler* oder *Rekonfigurator* bezeichnet. Der *rekonfigurierende Regelkreis* ist in Abbildung 3 dargestellt.

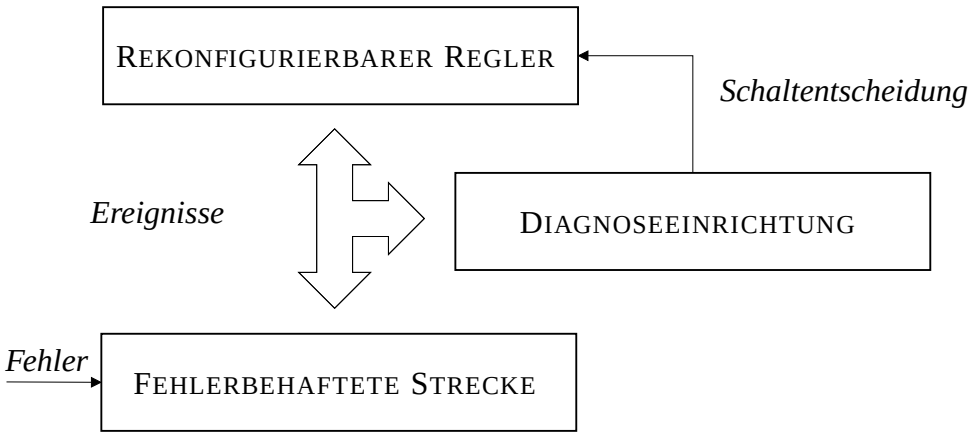


Abbildung 3: Rekonfigurierender Regelkreis mit Diagnoseeinrichtung

Bisher wurde von einem freien System ausgegangen. Liegt bereits ein geregeltes System vor, muss zur Umsetzung eines modellbasierten, fehlertoleranten rekonfigurierenden Reglers das vorliegende Steuerungsprogramm durch einen modellbasierten Regler ausgetauscht werden. Stellt das bestehende Steuerungsprogramm eine Implementierung eines modellbasiert entworfenen Nominalreglers dar, dann liefert die fehlerverdeckende Steuerungsrekonfiguration ein Konzept für den Entwurf eines *Rekonfigurators*, der zwischen Nominalregler und fehlerbehafteter Strecke geschaltet wird. Seine Aufgabe ist die Manipulation des Signalflusses, sodass ein steuerbarer und lebendiger Regelkreis entsteht. Abbildung 4 zeigt den fehlerverdeckenden Regelkreis.

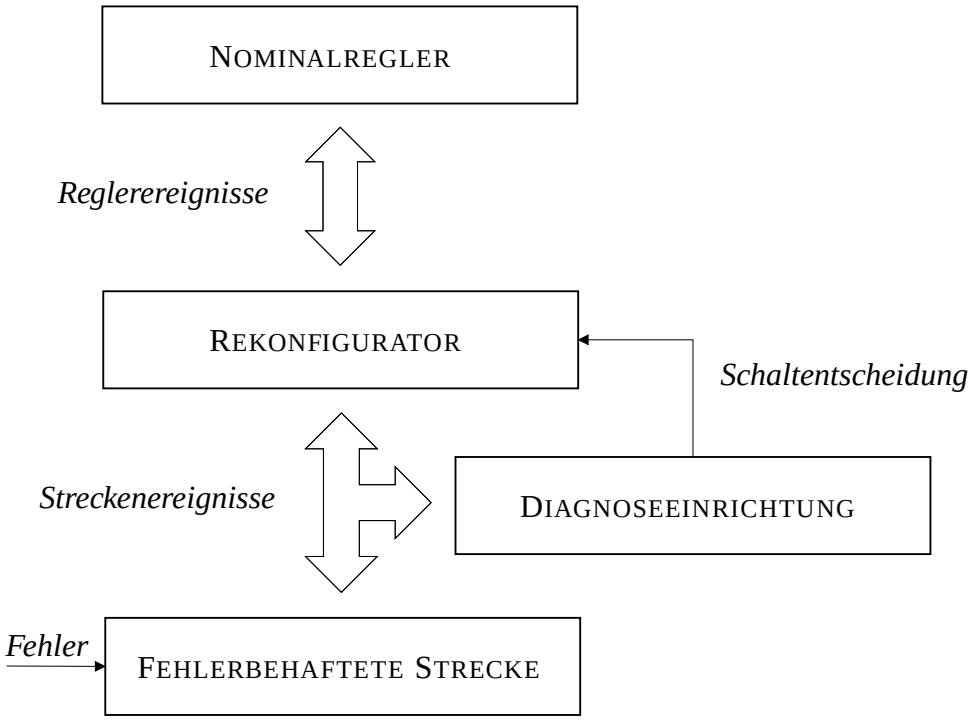


Abbildung 4: Fehlerverdeckender Regelkreis mit Diagnoseeinrichtung

Nachfolgend wird der Weg vom Nominalentwurfsproblem über einen Ansatz zur fehlertoleranten Regelung hin zur fehlerverdeckenden Steuerungsrekonfiguration beschrieben, wie er in dieser Arbeit beschritten wird.

Nominalentwurfsproblem

Zwar wird ein monolithischer Reglerentwurf angestrebt, im Gegensatz zu [Nke13, PSL11, KT12] ist allerdings der *Nominalregelkreis*, bestehend aus der *Nominalstrecke* und dem *Nominalregler*, in eine Steuerungsarchitektur eingebettet. Zur Kommunikation mit einem übergeordneten Bediener stehen ausgezeichnete *Bedienerereignisse* zur Verfügung. *Logische Ereignisse* verkoppeln einzelne Streckenkomponenten und dienen später zur Modellierung eines Fehlers. Abbildung 5 zeigt den Nominalregelkreis mit Bedienerschnittstelle.

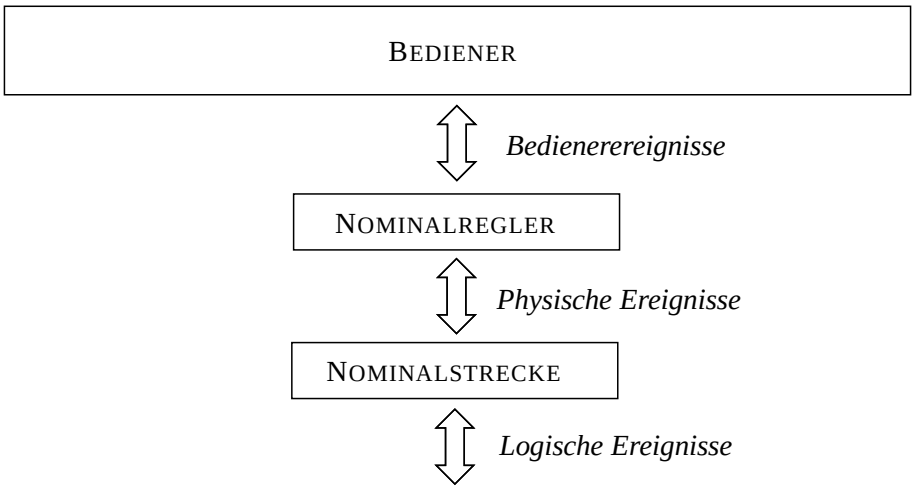


Abbildung 5: Nominalregelkreis mit Bedienerschnittstelle

Beispiel 3. Betrachtet wird eine einzelne Fertigungszelle, die von einem Leitsystem mit ihrer Umgebung koordiniert wird. Produktionsstart bzw. -abbruch werden durch die Ereignisse *a* bzw. *b* angezeigt. Erscheint das Ereignis *B*, wurde der Prozess erfolgreich beendet. Im Nominalbetrieb strebt die Maschine nach Produktionsbeginn ihren Anfangszustand an. Abbildung 6 zeigt das Nominalstreckenverhalten.

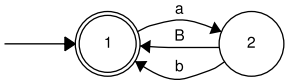


Abbildung 6: Nominalstrecke

Die Nominalspezifikation sieht das Melden des Produktionsbeginns *ah* und das Prozessende *Bh* an das Leitsystem vor, siehe Abbildung 7.

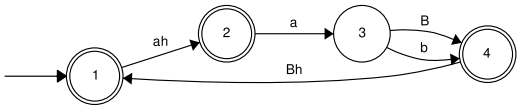


Abbildung 7: Nominalspezifikation

Es werden ansonsten keine weiteren Einschränkungen an das Streckenverhalten gestellt. In Abbildung 8 ist der minimal-restriktive Nominalregler dargestellt.

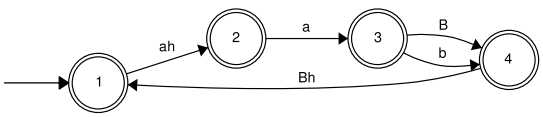


Abbildung 8: Nominalregler

Man bemerke, dass in der Nominalspezifikation die Semantik der Bedienerereignisse durch die erlaubten Abfolgen von physischen Ereignissen und Bedienerereignissen festgelegt ist. □

Nachdem ein Fehler aufgetreten ist, sind die Steuerungskommandos des Nominalreglers für die defekte Strecke nicht mehr sinnvoll. Inspiriert durch [PSL11] wird mit der *fehlerverträglichen Regelung* ein Ansatz zur fehlertoleranten Regelung vorgeschlagen.

Fehlerverträgliche Regelung

Mit *fehlerverträglichen Modellen*, siehe [WRM12], ist ein Rahmen zur systematischen Modellierung spontan auftretender Fehler gegeben. Dabei wird mit formalen Sprachen über steuerbaren und nicht-steuerbaren, bzw. beobachtbaren und nicht-beobachtbaren Ereignissen auf einen bereits bestehenden Modellierungsrahmen, siehe [RW87, RW89], zurückgegriffen.

Mit Hilfe *fehlerverträglicher Modelle* kann der Entwurf eines fehlertoleranten Reglers als Standardentwurfsproblem dargestellt werden. Im Gegensatz zu [PSL11] besteht keine Notwendigkeit für eine dedizierte Diagnoseeinrichtung und die Implementierung eines Umschaltmechanismus. Zwar werden dort keine formalen Aussagen über den geschlossenen Regelkreis getroffen, man darf aber davon ausgehen, dass ebenfalls ein steuerbarer und lebendiger geschlossener Regelkreis angestrebt wird, sodass sich die Entwurfsprobleme gleichen.

Fehlerverträgliche Modelle

Entsprechend der Vorstellung eines Fehlers als weder sichtbares noch unterdrückbares Phänomen wird dessen Auftreten als nicht-beobachtbares und nicht-steuerbares Ereignis modelliert.

Ein Fehlerereignis ist mit einer Verhaltensänderung der Strecke assoziiert. Deshalb wird neben dem *Streckennominalverhalten* ein *Streckenfehlermodell* erstellt, in dem das Auftreten eines Fehlers und dessen Konsequenzen für die Strecke nachgebildet werden. Zusammen bilden sie ein *fehlerverträgliches Modell* und ihre Vereinigung wird als *fehlerverträgliches Verhalten* bezeichnet.

Fehlerverträgliche Verhalten decken das Verhalten der Strecke im Nominalfall sowie die fehlerbedingte Verhaltensänderung der Strecke ab. Sie behalten deshalb auch im Fehlerfall ihre Gültigkeit und sind in diesem Sinne mit dem Phänomen „Fehler“ *verträglich*. Um

das Auftreten eines Fehlers in sich *schlüssig* zu modellieren, werden im Streckenfehlerverhalten lediglich diejenigen Zeichenketten erfasst, die vor dem Auftreten eines Fehlers dem Nominalstreckenverhalten genügen.

Beispiel 4. (Fortsetzung von Beispiel 3) Man gehe nun davon aus, dass die Fertigungszelle einem Fehler unterliegt. Infolgedessen zeigt das Kommando *b* keine Wirkung mehr. Zusätzlich strebt die Maschine den Arbeitszustand statt des Ruhezustands an. Ein entsprechendes Streckenfehlermodell ist in Abbildung 9 und das entsprechende fehlerverträgliche Streckenverhalten in Abbildung 10 gezeigt.

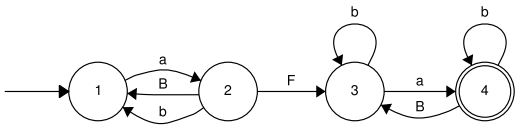


Abbildung 9: Streckenfehlerverhalten

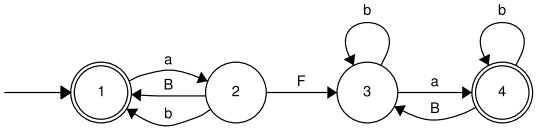


Abbildung 10: Fehlerverträgliches Streckenverhalten

Man bemerke, dass in dem fehlerverträglichen Streckenverhalten nur solche Zeichenketten auftauchen, die entweder im Nominalmodell gültig sind oder bis zum Auftreten eines Fehlers konsistent mit der Nominalstreckendynamik sind. Das heißt, der Fehlerübergang ist schlüssig modelliert. □

Fehlerverträgliches Entwurfsproblem

Fehlerverträgliche Modelle bilden den Kern der *fehlerverträglichen Regelung*. Man erinnere, dass in einem fehlerverträglichen Verhalten sowohl das Nominalverhalten als auch die fehlerbedingte Verhaltensänderung abgebildet wird. Eine Lösung des Standardentwurfsproblems mit einem fehlerverträglichen Verhalten als Eingabeparameter erzielt deshalb einen steuerbaren und lebendigen Regelkreis, unabhängig von dem Auftreten eines Fehlers.

Als Lösung des Standardentwurfsproblems benötigt ein fehlerverträglicher Regler weder eine Diagnoseeinrichtung noch einen separaten Umschaltmechanismus. Abbildung 11 zeigt den fehlerverträglichen Regelkreis.

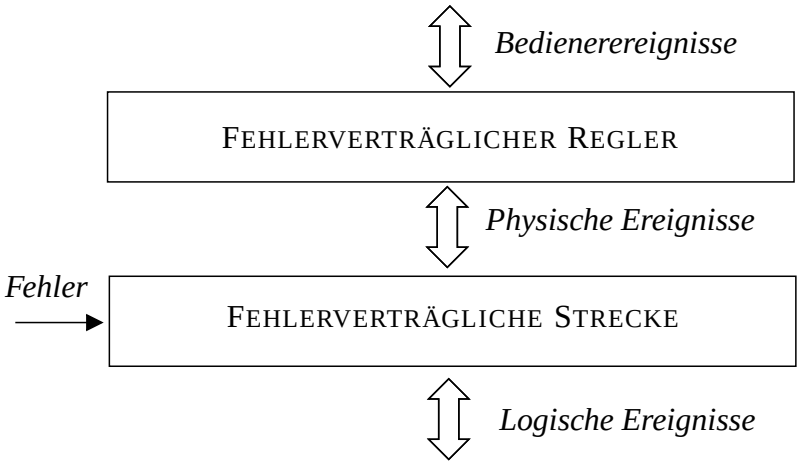


Abbildung 11: Fehlerverträglicher Regelkreis

Im Fehlerfall sind zusätzlich die Bedienereingaben in einer Art und Weise umzusetzen, die den Konsequenzen eines Fehlers gerecht werden. Vor diesem Hintergrund werden die Entwurfsvorgaben selbst in einem fehlerverträglichen Modell formuliert.

Beispiel 5 (Fortsetzung von Beispiel 4). Für den Entwurf eines fehlerverträglichen Reglers müssen die Konsequenzen des Fehlers berücksichtigt werden. In der fehlerverträglichen Spezifikation gemäß Abbildung 12 wird dies durch das Setzen entsprechender Markierungen getan. Die Semantik der Bedienerereignisse wird beibehalten.

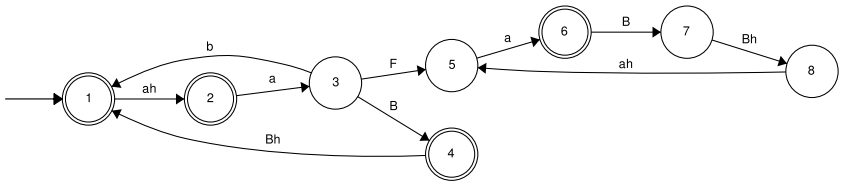


Abbildung 12: Fehlerverträgliche Spezifikation

Ein entsprechender fehlerverträglicher Regler ist in Abbildung 13 gezeigt.

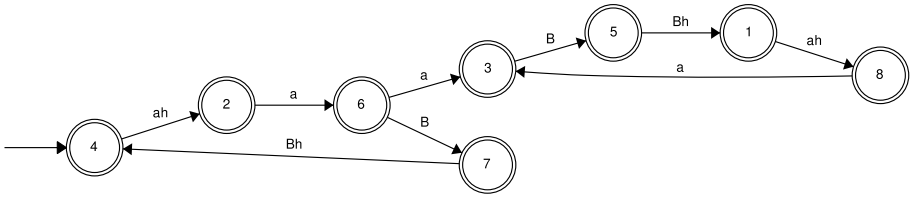


Abbildung 13: Fehlerverträglicher Regler

Der fehlerverträgliche Regler benötigt offensichtlich keine Information über das Auftreten eines Fehlers. Zudem werden die Entwurfsvorgaben für das Nominal- sowie das Fehlerverhalten ohne eine Umschaltvorrichtung umgesetzt. □

Für den Entwurf eines fehlerverträglichen Reglers wird von einem freien System ausgegangen. Liegt ein gesteuertes System vor, ersetzt die Umsetzung eines fehlerverträglichen Reglers das existierende Steuerungsprogramm. Ist dies nicht erwünscht, kann mit Hilfe der fehlerverdeckenden Steuerungsrekonfiguration eine Schnittstelle zwischen existierenden Nominalregler und fehlerverträglicher Strecke geschaffen werden.

Fehlerverdeckung durch Steuerungsrekonfiguration

Nominalregler und fehlerverträgliche Strecke sind zunächst über *physische Ereignisse* verkoppelt. Zur Manipulation des Signalflusses zwischen beiden Teilsystemen werden *virtuelle* Ereignisse eingeführt, die im Nominalregler die zu verkoppelnden physischen Ereignisse ersetzen. Durch die *Virtualisierung* des Nominalreglers entstehen zunächst zwei völlig entkoppelte Systeme, die anschließend über den Rekonfigurator synchronisiert werden, siehe Abbildung 14.

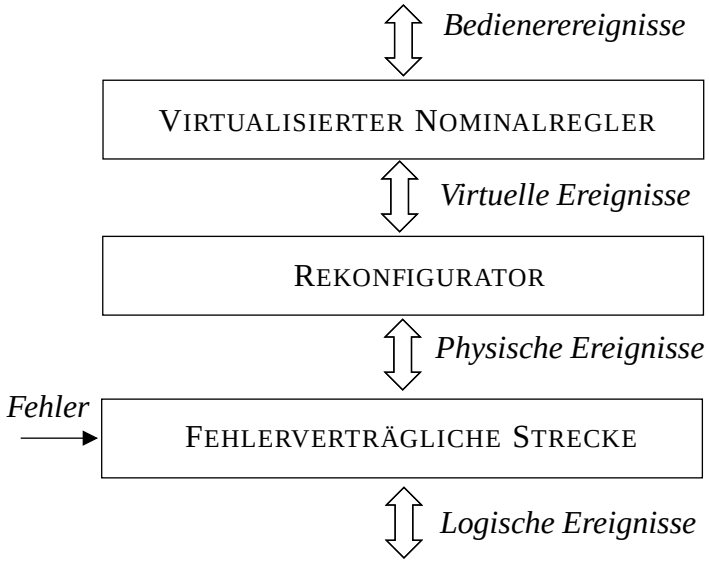


Abbildung 14: Fehlerverdeckender Regelkreis

Durch die Berücksichtigung eines Nominalreglermodells beim Rekonfiguratorentwurf entspricht die Abfolge der Reglerereignisse im rekonfigurierenden Regelkreis der Reihenfolge im Nominalregelkreis. In diesem Sinn werden die Konsequenzen eines Fehlers vor dem Nominalregler *verdeckt*.

Für den Entwurf des Rekonfigurators werden der virtualisierte Nominalregler und die fehlerverträgliche Strecke als *formale Strecke* aufgefasst. Liegt ein Modell des Nominalreglers vor, dann ist der Rekonfiguratorentwurf wiederum ein Standardentwurfsproblem.

Die Abfolge der Bedienerereignisse ist einerseits über das Modell des Nominalreglers mit den virtuellen Ereignissen verknüpft, andererseits wird durch die Entwurfsvorgaben die Semantik anhand der physischen Ereignisse festgelegt. Der Rekonfigurator stellt dann sicher, dass die Abfolge der virtuellen und der physischen Ereignisse anhand der Bedienerereignisse koordiniert wird.

Daneben können den Entwurfsvorgaben weitere Anforderungen an das Umrechnen physischer Ereignisse in virtuelle Ereignisse und umgekehrt hinzugefügt werden, z. B. dass der Nominalregler die Kontrolle über den Prozess bis zum Auftreten eines Fehlers behält. Diese Anforderung wird als *Inaktivitätsbedingung* bezeichnet, siehe [Ste05]. Für den Entwurf des Rekonfigurators werden der Nominalregler und die fehlerverträgliche Strecke als *formale Strecke* aufgefasst.

Liegt ein Steuerungsprogramm lediglich als Implementierung eines Nominalreglers vor, kann nur in Einzelfällen ein hinreichend genaues Reglermodell erstellt werden. In diesem Fall wird mit dem Nominalregler eine unbekannte Komponente in den Regelkreis eingeführt. Auf Grund dieser Unsicherheit wird gefordert, dass der Rekonfigurator für einen *beliebigen* Nominalregler einen, unter Spezifikationseinschluss, steuerbaren und lebendigen Regelkreis, erzielt. Erfüllt ein Rekonfigurator diese Anforderung, heißt er *universell*.

Nachdem der Nominalregler als Lösung des Nominalentwurfsproblems bekannt ist, liegt mit dem minimal-restriktiven Nominalregler dessen beste berechenbare Näherung vor und kann dem Entwurf zu Grunde gelegt werden. Allerdings erzielt ein Rekonfigurator auf Basis des minimal-restriktiven Reglers nicht zwangsläufig für einen beliebigen Nominalregler einen lebendigen Regelkreis.

Beispiel 6 (Fortsetzung von Beispiel 5). In einem ersten Schritt wird der minimal-restriktive Nominalregler virtualisiert, siehe Abbildung 15 .

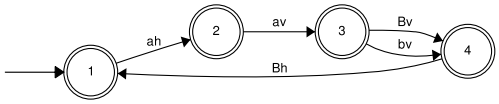


Abbildung 15: Minimal-restriktiver Nominalregler

Es kann ein Rekonfigurator gewonnen werden, der für den virtualisierten minimal-restriktiven Nominalregler einen lebendigen und implementierbaren Regelkreis, siehe Abbildung 16, erzeugt. Die Zustände 1 bis 5 und 13 legen einen Teilautomaten fest, der das Regelkreisverhalten vor dem Auftreten eines Fehlers beschreibt. Man bemerke, dass der Rekonfigurator *inaktiv* ist, d. h. physische und virtuelle Ereignisse werden 1:1

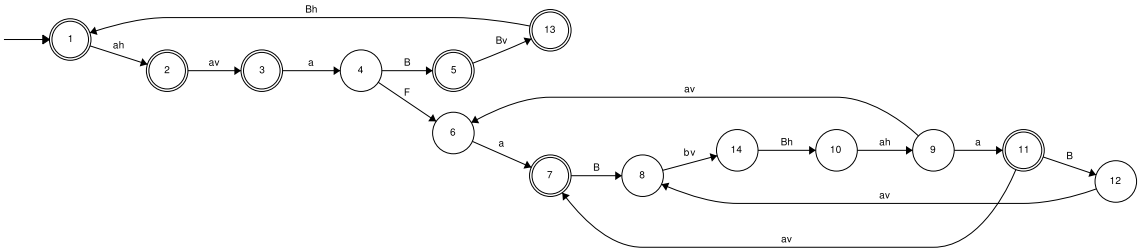


Abbildung 16: Rekonfigurierender Regelkreis für den minimal-restriktiven Nominalregler

umgesetzt. Dabei wird erzwungen, dass einem physischen steuerbaren Ereignis ein virtuelles steuerbares Ereignis und umgekehrt einem virtuellen nicht-steuerbaren Ereignis ein physisches nicht-steuerbares Ereignis vorausgeht. Implizit wird so von einer Ursache-Wirkung-Beziehung zwischen Steuergerät und Prozess ausgegangen, die das Steuergerät bzw. den Prozess als Ursache steuerbarer bzw. nicht-steuerbarer Ereignisse festlegt. Nachdem ein Fehler aufgetreten ist, werden diese *Inaktivitätsbedingungen* aufgehoben.

Weiter vergegenwärtige man sich die Synchronisation des virtualisierten Nominalreglers und der fehlerverträglichen Strecke anhand der Bedienerereignisse *ah* und *Bh*. Erst nach der Meldung des Prozessbeginns mit *ah* wird der physische Prozess gestartet und erst nach dessen Beendigung erfolgt eine Meldung an den Bediener mit *Bh*. Dazwischen werden die physischen und die virtuellen Ereignisse so ineinander umgesetzt, dass einerseits der Semantik der Bedienerereignisse genüge getan wird, andererseits die Abfolge der virtuellen Ereignisse mit den minimal-restriktiven Regler konsistent ist.

Abbildung 17 zeigt eine weitere virtualisierte Lösung des Nominalentwurfsproblems und in Abbildung 18 ist der induzierte und offensichtlich nicht lebendige, rekonfigurierende Regelkreis dargestellt.

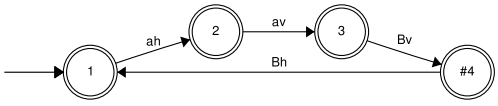


Abbildung 17: Spezieller Nominalregler

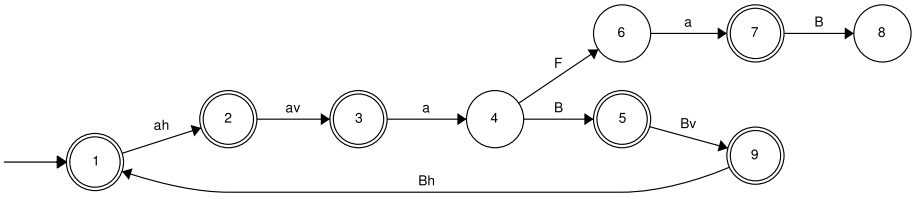


Abbildung 18: Rekonfigurierender Regelkreis für einen speziellen Nominalregler

□

Wie das obige Beispiel zeigt, stellt der Entwurf universeller Rekonfiguratoren auf Lebendigkeit ein nicht-triviales Problem dar. Ihm ist der größte Teil dieser Arbeit gewidmet.

Motiviert wurde diese Arbeit durch die Zusammenarbeit mit Vertretern der Industrie, in deren Umfeld standardmäßig von Hand programmierte Steuerungsprogramme zum Einsatz kommen. Vor dem Hintergrund einer praktischen Anwendung darf ein fehlendes Reglermodell deshalb als der Regelfall angenommen werden. Von daher steht im Hauptteil dieser Arbeit der Entwurf universeller Rekonfiguratoren im Vordergrund. Ihre praktische Anwendbarkeit wird durch eine Machbarkeitsstudie belegt.

Beiträge und Struktur dieser Arbeit

Diese Arbeit entstand aus dem Bestreben die Verlässlichkeit von automatisierten Fertigungssystemen durch steuer- und regelungstechnische Maßnahmen zu erhöhen. In einem ersten Schritt wird mit der fehlerverträglichen Regelung zunächst ein Ansatz zur fehlertoleranten Regelung ereignisdiskreter Systeme vorgestellt. Die fehlerverträgliche Regelung wird anschließend, mit dem Ziel ein bestehendes Steuerungssystem zu einem fehlertoleranten Regelungssystem zu erweitern, zur fehlerverdeckenden Steuerungsrekonfiguration weiterentwickelt.

In Kapitel 1 wird zunächst die Problemstellung formalisiert. Es wird ein geeignetes Standardentwurfsproblem formuliert, auf das der Nominalreglerentwurf und der Entwurf fehlertoleranter Regler sowie Teile des Rekonfiguratorentwurfs zurückgeführt werden können. Fehlerverträgliche Modelle liefern dann einen Modellierungsrahmen für Strecken, die spontan auftretenden Fehlern unterliegen. Weiter wird der rekonfigurierende Regelkreis und ein Rekonfigurationsproblem für ereignisdiskrete Systeme diskutiert.

Kapitel 2 ist dem Entwurf universeller Rekonfiguratoren gewidmet. Ausgehend von einem unbekannten Nominalregler werden hinreichende Bedingungen für den Entwurf eines universellen Rekonfigurators, unter der Annahme trivialer Letztendlichkeitsbedingungen, angegeben. Im zweiten Teil dieses Kapitels steht die Verifikation konfliktfreier Regelkreise im Mittelpunkt. Auch hier können hinreichende Bedingungen angegeben werden. Abschließend wird ein Entwurfsalgorithmus für universelle Rekonfiguratoren vorgestellt.

Das letzte Kapitel dieser Arbeit stellt konzeptionell die Umsetzung der fehlerverträglichen und der fehlerverdeckenden Steuerungsrekonfiguration in der industriellen Vorfeldentwicklung vor. Neben einer pragmatischen Erweiterung der Methodik auf zeitbewertete ereignisdiskrete Systeme wird die Umsetzung ereignisdiskreter Regler auf speicherpro-

Kapitel 1

Fehlerverdeckende Steuerungsrekonfiguration

In Vorbereitung auf den Entwurf fehlerverdeckender Rekonfiguratoren werden in diesem Kapitel die Elemente des rekonfigurierenden Regelkreises, siehe Abbildung 1.0.1, sowie die Entwurfsziele der fehlerverdeckenden Steuerungsrekonfiguration erläutert.

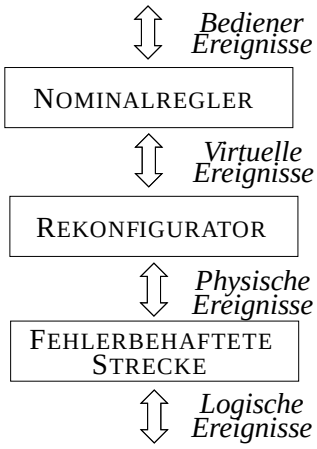


Abbildung 1.0.1: Rekonfigurierender Regelkreis

Beginnend mit einer Darstellung des formalen Rahmens wird der Nominalregler als Lösung des *Nominalentwurfsproblems* charakterisiert. Anschließend werden *fehlerverträgliche Modelle* als Modellierungswerkzeug für fehlerbehaftete Strecken eingeführt und mit der *fehlerverträglichen Regelung* ein Ansatz zur fehlertoleranten Regelung vorgestellt. Darauf aufbauend wird abschließend der *fehlerverdeckende Regelkreis* diskutiert und das *Problem der fehlerverdeckenden Steuerungsrekonfiguration* formuliert.

In der nominellen und der fehlerverträglichen Regelung sind die Regelkreiskomponenten

und die Entwurfsziele identisch, modulo der Partitionierung des Gesamtalphabets. Um die folgende Diskussion abzukürzen, wird ein entsprechend parametriertes *Standardentwurfsproblem* eingeführt. Das Nominalentwurfsproblem und das Problem der fehlerverträglichen Regelung können dann als Standardentwurfsproblem dargestellt und Teile der fehlerverdeckenden Steuerungsrekonfiguration auf das Standardentwurfsproblem zurückgeführt werden.

Die Gesamtalphabete der nominellen und fehlerverträglichen Regelung sowie der fehlerverdeckenden Steuerungsrekonfiguration sind bzgl. Mengeneinschluss direkte Erweiterungen voneinander. Trotzdem ist eine gewisse Anzahl unterschiedlicher Alphabete notwendig, um die einzelnen Entwurfsprobleme zu beschreiben. Die nachfolgende Tabelle 1.0.1 gibt bereits an dieser Stelle einen Überblick über die verwendeten Alphabete und ihre Zusammensetzung. Der Leser möge bei Bedarf auf sie zurückkommen.

	Σ_{HI}	$\Sigma_{LO,N}$	Σ_{CON}	Σ_{UCON}	$\Sigma_{CON,V}$	$\Sigma_{UCON,V}$	F
Reglerereignisse Σ_C	x		x	x			
Nominalstreckenereignisse $\Sigma_{P,N}$		x	x	x			
Nominalereignisse Σ_N	x	x	x	x			
Streckenfehlerereignisse $\Sigma_{P,F}$		x	x	x			x
Fehlerereignisse Σ_F	x	x	x	x			x
Virtuelle Reglerereignisse $\Sigma_{C,V}$	x				x	x	
Virtuelle Streckenereignisse $\Sigma_{P,V}$		x			x	x	
Virtuelle Ereignisse Σ_V	x	x			x	x	
Log. Ereignisse inkl. Fehler $\Sigma_{LO,F}$		x					x
Gesamtalphabet Σ	x	x	x	x	x	x	x

Tabelle 1.0.1: Atomare und zusammengesetzte Alphabete

1.1 Modellierung ereignisdiskreter Dynamik

Mit *formalen Sprachen* stellt die Informatik einen geeigneten mathematischen Rahmen zur Modellierung ereignisdiskreter Dynamik bereit. Es folgt eine knappe Darstellung wesentlicher Ergebnisse und grundlegender Notation. Eine Buchveröffentlichung zu formalen Sprachen und endlichen Automaten ist mit [HU94] vorhanden.

Alphabete und Zeichenketten. Unter einem *Alphabet* Σ wird eine endliche Menge von Ereignissen σ verstanden. Eine *Zeichenkette* ist eine endliche Aneinanderreihung von Er-

eignissen $s = \sigma_1\sigma_2\sigma_3\ldots\sigma_n$, mit $\sigma_i \in \Sigma$, $i \leq n$ und $n, i \in \mathbb{N}$. Weiter bezeichnet $|s| = n$ ihre *Länge*. Mit Σ^* wird die Menge aller Zeichenketten inklusive der *leeren Zeichenkette* ε, $|\varepsilon| = 0$ angegeben. Eine Teilmenge $L \subseteq \Sigma^*$ heißt *formale Sprache*.

Reguläre Sprachen. Zu zwei Zeichenketten s_1, s_2 bezeichnet $s = s_1 s_2$ ihre *Konkatenation*. Es sei s eine Zeichenkette, dann bezeichnet der Kleenesche Abschluss s^* von s die Menge aller Zeichenketten, die durch Konkatenation von s mit sich selbst hervorgehen. Per Definition seien die *leere Menge* $\emptyset$, $\{\varepsilon\}$ und $\{\sigma\}, \sigma \in \Sigma$ reguläre Sprachen. Geht eine formale Sprache L lediglich durch Vereinigung, Konkatenation und Kleenescher Abschlussbildung aus regulären Sprachen hervor, dann heißt L *regulär*. Regularität bleibt unter Komplementbildung erhalten. Weiter ist auf L die *Nerode-Äquivalenzrelation* $\equiv_L$ gegeben durch

$$(\forall s', s'' \in \Sigma^*)[s' \equiv_L s'' :\Leftrightarrow (\forall t \in \Sigma^*)[s't \in L \Leftrightarrow s''t \in L]]. \tag{1.1}$$

Man kann die Äquivalenz der folgenden Aussagen zeigen:

(i) Die Anzahl der Äquivalenzklassen $[\equiv_L]$ ist endlich.
(ii) L ist regulär.
(iii) L wird von einem endlichen Automaten akzeptiert.

Endliche Automaten. Ein *endlicher Automat* ist ein Tupel $G = (Q, \Sigma, \delta, q_0, Q_m)$ mit einer endlichen *Zustandsmenge* Q, einem Alphabet Σ, einem *Anfangszustand* q_0 und einer Menge von *markierten Zuständen* Q_m. Je nach Kontext bezeichnet das Symbol δ die Transitionenrelation $\{(q, \sigma, q') \in Q \times \Sigma \times Q \mid \delta(q, \sigma) = q'\}$ oder die partielle Funktion $\delta : Q \times \Sigma^* \to Q$. Dabei wird zunächst $\delta : Q \times \Sigma \to Q$ gesetzt und anschließend induktiv zu $\delta : Q \times \Sigma^* \to Q$, gemäß $\delta(q, \varepsilon) = q$ und $\delta(q, s\sigma) = \delta(\delta(q, s), \sigma)$, erweitert. Falls für ein $q \in Q$ und $s \in \Sigma^*$ ein $q' \in Q$ mit $\delta(q, s) = q'$ existiert, heißt δ für q und s *definiert*, in Zeichen $\delta(q, s)!$. Ein Zustand $q \in Q$ heißt *erreichbar*, falls ein $s \in \Sigma^*$ mit $\delta(q_0, s) = q$ existiert. Weiter heißt q *co-erreichbar*, falls ein $r \in \Sigma^*$ mit $\delta(q, r) \in Q_m$ existiert. Sind alle Zustände eines Automaten erreichbar bzw. co-erreichbar, so heißt der Automat erreichbar bzw. co-erreichbar. Ist ein Automat erreichbar und co-erreichbar, so heißt er *trimm*. Einem Automaten werden eine *generierte Sprache* $L(G) := \{s \in \Sigma^* \mid \delta(q_0, s)!\}$ und eine *markierte Sprache* $L_m(G) := \{s \in \Sigma^* \mid \delta(q_0, s) \in Q_m\}$ zugeordnet. Es bezeichnet $G_\emptyset$ den *leeren Automaten*, d. h. einen Automaten ohne Anfangszustand, mit $L(G) = \emptyset$ und $L_m(G) = \emptyset$.

Gegeben seien zwei Automaten $G = (Q, \Sigma, \delta, q_0, Q_m)$ und $H = (X, \Sigma, \xi, x_0, X_m)$. Es heißt H *Teilautomat* von G, in Zeichen $H \sqsubseteq G$, falls aus $s \in L(G)$ und $\xi(x_0, s)!$ die Beziehung $\xi(x_0, s) = \delta(q_0, s)$ folgt. Insbesondere gilt $L(H) \subseteq L(G)$. Für erreichbare Automa-

ten ist $H \sqsubseteq G$ genau dann, wenn $X \subseteq Q$, $x_0 = q_0$ und $\xi \subseteq \delta$ gilt. Für eine Teilmenge $P \subseteq Q$ bezeichnet $G|_{\mathrm{P}} = (P, \Sigma, \rho, q_0, Q_{\mathrm{m}} \cap P)$ die *Einschränkung von G auf P*, wobei für alle $q \in P$ und $\sigma \in \Sigma$ mit $\delta(q,\sigma) \in P$ auch $\rho(q,\sigma) \coloneqq \delta(q,\sigma)$ definiert ist. Im anderen Fall ist $\rho(q,\sigma)$ nicht definiert. Falls G der leere Automat ist oder falls $q_0 \notin P$ gilt, dann ist die Einschränkung von G auf P der leere Automat. Jede Einschränkung von G auf P ist ein Teilautomat von G. Eine Einschränkung auf alle erreichbaren Zustände hat keinen Einfluss auf die generierte Sprache von G. Eine Einschränkung von G auf alle erreichbaren und co-erreichbaren Zustände hat keinen Einfluss auf die markierte Sprache. Das *Produkt* der Automaten G und H, in Zeichen $F = G \times H$, ist definiert durch $F \coloneqq (Q \times X, \Sigma, \zeta, (q_0, x_0), Q_{\mathrm{m}} \times X_{\mathrm{m}})$ mit $\zeta((q,x),\sigma)!$ genau dann, wenn $\delta(q,\sigma)!$ und $\xi(x,\sigma)!$. Andernfalls bleibt $\zeta((q,x),\sigma)$ undefiniert. Ist einer der beiden Automaten der leere Automat, so ist es auch das Produkt. Das Automatenprodukt führt auf den Sprachenschnitt der generierten und der markierten Sprachen der beteiligten Automaten, d. h. $\mathrm{L}(G \times H) = \mathrm{L}(G) \cap \mathrm{L}(H)$ und $\mathrm{L_m}(G \times H) = \mathrm{L_m}(G) \cap \mathrm{L_m}(H)$.

1.2 Standardreglerentwurf

Aufbauend auf *regulären Sprachen* und *endlichen Automaten* wurde mit der *Supervisory Control Theory*, siehe [RW87, RW89] und ergänzend [CL08], eine methodische Grundlage für den Entwurf ereignisdiskreter Regler geschaffen.

Ereignisdiskrete Systeme entwickeln sich mit dem spontanen Auftreten von Ereignissen, deren Reihenfolge und Ausführungszeitpunkt durch nicht notwendigerweise modellierte Mechanismen bestimmt werden. Entsprechend beschränken sich ereignisdiskrete Modelle auf alle phänomenkompatiblen logischen Ereignisfolgen.

Nur in Ausnahmefällen sind alle möglichen Ereignisfolgen mit den Entwurfsvorgaben vereinbar. Ziel des Reglerentwurfs ist es dann, die Strecke auf die *erlaubten* Ereignisfolgen einzuschränken. Dabei sind die Wirkzusammenhänge, die die Entwicklung des betrachteten Phänomens bestimmen, geeignet zu berücksichtigen. Dies führt auf die Begriffe *Spezifikationseinschluss, Steuerbarkeit* und *Lebendigkeit*.

Beginnend mit Grundbegriffen der Supervisory Control Theory wird nachfolgend das Standardentwurfsproblem dieser Arbeit entwickelt.

1.2.1 Grundbegriffe der Supervisory Control Theory

Verkopplung

Jedem dynamischen System liegen individuelle Gesetzmäßigkeiten zu Grunde. Durch die Verkopplung dynamischer Systeme entsteht ein neues dynamisches System, das den Gesetzmäßigkeiten der verkoppelten Teilsysteme genügt. Werden ereignisdiskrete Systeme miteinander verschaltet, bleiben im verkoppelten System lediglich jene Ereignisfolgen erhalten, die den beteiligten Phänomenen genügen. Liegen ereignisdiskrete Modelle in Form von formalen Sprachen bzw. endlichen Automaten vor, ist der *Sprachenschnitt* bzw. das *Automatenprodukt* ein geeignetes Modell für das verkoppelte System.

Im Allgemeinen sind die Komponenten verkoppelter Systeme nicht über einem gemeinsamen Alphabet definiert. In diesem Fall wird die Vereinigung der Komponentenalphabete gebildet. Ist ein Ereignis ein Element aller Komponentenalphabete, heißt es *geteilt*. Während geteilte Ereignisse von allen Komponenten *synchron* ausgeführt werden, treten nicht-geteilte Ereignisse nur in der betreffenden und ungeachtet aller anderen Komponenten auf. Jede Komponente wird zunächst mit Hilfe der *inversen natürlichen Projektion* um fehlende nicht-geteilte Ereignisse ergänzt.

Natürliche Projektionen. Zu zwei Alphabeten Σ und $\Sigma_o \subseteq \Sigma$ ist die *natürliche Projektion* oder kurz *Projektion*, $\mathrm{p} : \Sigma^* \to \Sigma_o^*$ induktiv gemäß $\mathrm{p}_o\varepsilon = \varepsilon$ und $\mathrm{p}_o(s\sigma) = \mathrm{p}_o s$, falls $\sigma \notin \Sigma_o$ und $\mathrm{p}_o(s\sigma) = (\mathrm{p}_o s)\sigma$ sonst, definiert. Die natürliche Projektion entfernt aus einer Zeichenkette alle Zeichen, die nicht in Σ_o enthalten sind. Per Konvention zeigt in $\mathrm{p}_\_$ der Subskript _ die Bildmenge einer natürlichen Projektion an. Die *inverse Projektion* $\mathrm{p}_o^{-1} : \Sigma_o^* \to 2^{\Sigma^*}$ ist für ein $u \in \Sigma_o^*$ durch $\mathrm{p}_o^{-1}u := \{s \in \Sigma^* \mid \mathrm{p}_o s = u\}$ gegeben. Sowohl die Projektion als auch die inverse Projektion werden, gemäß [CL08], auf Sprachen erweitert. Dann kommutiert die Projektion mit Vereinigung und die inverse Projektion mit Vereinigung und Schnitt.

Mit Hilfe der inversen Projektion kann die *synchrone Komposition* als Modell verkoppelter Systeme über verschiedenen Alphabeten eingeführt werden.

Synchrone Komposition. Gegeben seien zwei Sprachen $L_1 \subseteq \Sigma_1^*$ und $L_2 \subseteq \Sigma_2^*$ mit $\Sigma_1 \subseteq \Sigma$ und $\Sigma_2 \subseteq \Sigma$, dann heißt $L_1 \parallel L_2 := (\mathrm{p}_1^{-1}L_1) \cap (\mathrm{p}_2^{-1}L_2)$ die synchrone Komposition von L_1 und L_2.

Lebendigkeit

Mit dem Begriff der Lebendigkeit wird die Fähigkeit eines ereignisdiskreten Systems beschrieben, sich selbst über der Zeit weiterzuentwickeln. Strebt das System auf Grund der versteckten Wirkzusammenhänge ausgezeichnete Systemkonfigurationen an, spricht man von sogenannten *Letztendlichkeitseigenschaften*. In diesem Zusammenhang unterscheidet man zwischen dem *lokalen* und dem *globalen* Systemverhalten. Während das globale Systemverhalten ausschließlich aus Zeichenketten besteht, die das System in einen ausgezeichneten Zustand führen, beschreibt das lokale Verhalten dessen schrittweise Entwicklung. Werden ereignisdiskrete Systeme durch formale Sprachen modelliert, gibt die Sprache selbst das globale Systemverhalten wieder, während mit ihrem *Vorsilbenabschluss* ein geeignetes Modell für das lokale Systemverhalten gegeben ist.

Vorsilbenabschluss. Existiert zu zwei Zeichenketten $s,t \in \Sigma^*$ eine Fortsetzung $r \in \Sigma^*$ von s, sodass $sr = t$ gilt, dann heißt s eine *Vorsilbe* von t, in Zeichen $s \leq t$. Der *Vorsilbenoperator* pre bildet eine Sprache $L \subseteq \Sigma^*$ auf die Menge all ihrer Vorsilben $\operatorname{pre} L := \{ s \in \Sigma^* \mid (\exists r \in \Sigma^*)[sr \in L] \}$ ab. Es heißt $\operatorname{pre} L$ der *Vorsilbenabschluss* von L. Enthält eine Sprache all ihre Vorsilben, in Zeichen $L = \operatorname{pre} L$, so heißt L *abgeschlossen unter Vorsilbenbildung* oder kurz *abgeschlossen*. Abgeschlossenheit unter Vorsilbenbildung bleibt unter Vereinigung erhalten, nicht jedoch unter Schnitt.

Kann ein ereignisdiskretes System durch eine unter Vorsilbenbildung abgeschlossene Sprache modelliert werden, weist es lediglich *triviale Letztendlichkeitseigenschaften* auf.

Besitzt ein ereignisdiskretes System stets die Möglichkeit zur schrittweisen Weiterentwicklung, heißt es *vollständig*. Man bemerke, dass die Vollständigkeit anhand des lokalen Systemverhaltens festgemacht werden kann.

Vollständigkeit. Ein dynamisches System L heißt *vollständig*, falls für alle $s \in \operatorname{pre} L$ eine Fortsetzung $\sigma \in \Sigma$ mit $s\sigma \in \operatorname{pre} L$ existiert, siehe [KGM92]. Eine Variation der Vollständigkeit fordert die Existenz einer einschrittigen Fortsetzung aus einem ausgezeichneten Alphabet $\Sigma_o \subseteq \Sigma$. Entsprechend heißt L Σ_o-*vollständig*, falls für alle $s \in \operatorname{pre} L$ eine Fortsetzung $t \in (\Sigma - \Sigma_o)^* \Sigma_o$ mit $st \in \operatorname{pre} L$ existiert, siehe [SMP08].

Werden verkoppelte Systeme, wie z. B. der geschlossene Regelkreis, betrachtet, müssen die Letztendlichkeitseigenschaften der jeweiligen Komponenten berücksichtigt werden. Im verkoppelten System sind nur diejenigen Ereignisfolgen gültig, die alle Komponenten in einen ausgezeichneten Zustand führen. Besteht stets für alle Komponenten die Möglichkeit simultan eine gültige Ereigniskette zu generieren, dann heißt das verkoppelte System *konfliktfrei*.

Konfliktfreiheit. Es heißen zwei dynamische Systeme $L_1 \subseteq \Sigma_1^*$ und $L_2 \subseteq \Sigma_2^*$, mit $\Sigma_1 \subseteq \Sigma$ und $\Sigma_2 \subseteq \Sigma$ *konfliktfrei*, falls $\mathrm{pre}\,(L_1 \parallel L_2) = (\mathrm{pre}\,L_1) \parallel (\mathrm{pre}\,L_2)$ gilt.

Steuerbarkeit

Im Allgemeinen werden die treibenden Mechanismen in einem ereignisdiskreten Modell nicht berücksichtigt und können von einem ereignisdiskreten Regler auch nicht unmittelbar beeinflusst werden. Vor diesem Hintergrund wird davon ausgegangen, dass Ereignisse existieren, die die Ursache weiterführender Wirkungsketten sind und die von einem ereignisdiskreten Regler *verboten* werden können. Entsprechend wird das Gesamtalphabet in *steuerbare* und *nicht-steuerbare* Ereignisse partitioniert. Reglereingriffe sind dann auf das Verbieten steuerbarer Ereignisse beschränkt. Als Beispiel für ein steuerbares Ereignis kann das Schalten eines Aktuators genannt werden. Dagegen entzieht sich das Anschlagen eines Sensors der unmittelbaren Kontrolle durch einen Regler. Mit dem Verbieten steuerbarer Ereignisse werden dem Regler nur passive Eingriffsmöglichkeiten zugestanden.

Steuerbarkeit. Die nicht-steuerbaren Ereignisse $\Sigma_{\mathrm{uc}} \subseteq \Sigma$ dürfen im geschlossenen Regelkreis $K \subseteq \Sigma^*$ nicht verboten werden, so sie denn in der Strecke erlaubt sind. Bezeichnet $L \subseteq \Sigma^*$ eine Strecke, dann heißt K in diesem Sinne *steuerbar bzgl.*$(L, \Sigma_{\mathrm{uc}})$, falls $(\mathrm{pre}\,K)\Sigma_{\mathrm{uc}} \cap (\mathrm{pre}\,L) \subseteq \mathrm{pre}\,K$ gilt.

Eingeschränkte Beobachtbarkeit

Sind in der Strecke Ereignisse vorhanden, deren Auftreten für einen Regler nicht sichtbar ist, liegt ein Regelungsproblem unter eingeschränkter Beobachtbarkeit vor. Das Gesamtalphabet wird dann in *beobachtbare* und *nicht-beobachtbare* Ereignisse partitioniert. Der Regler ist dann so zu entwerfen, dass die Entwurfsvorgaben ohne die Information über das Auftreten nicht-beobachtbarer Ereignisse umgesetzt werden.

In dieser Arbeit unterstützen nicht-beobachtbare Ereignisse eine komponentenorientierte Modellbildung. Zudem wird das Auftreten eines Fehlers als nicht-beobachtbares Ereignis aufgefasst.

Spezifikationseinschluss

Die verbalen Entwurfsvorgaben werden in einer formalen Sprache, der *Spezifikation*, hinterlegt. Von einem Regler wird dann erwartet, das Streckenverhalten auf solche Zeichen-

ketten einzuschränken, die auch in der Spezifikation erlaubt sind. Dabei sind die Steuerbarkeit und die Lebendigkeit des geschlossenen Regelkreises zu wahren.

Reglerentwurf

Ausgehend von einem Streckenmodell und einer Spezifikation ist ein ereignisdiskreter Regler so zu entwerfen, dass der geschlossene Regelkreis gewünschte Eigenschaften, hier Steuerbarkeit und Lebendigkeit, aufweist. Dazu wird ein *Kandidat* für den geschlossenen Regelkreis gesucht, dessen Eigenschaften die Ableitung eines geeigneten Reglers erlauben.

Im Kontext des Reglerentwurfs auf Konfliktfreiheit heißt zu einer Strecke $L \subseteq \Sigma^*$ ein Kandidat $K \subseteq \Sigma^*$ *relativ abgeschlossen bzgl.* L, falls $K = (\operatorname{pre} K) \cap L$ gilt. Dann erreicht K einen ausgezeichneten Zustand, falls das auch für die Strecke der Fall ist.

Ist der Regler lediglich über einer Teilmenge beobachtbarer Ereignisse $\Sigma_o \subseteq \Sigma$ definiert, muss die Zugehörigkeit einer Zeichenkette $s \in \Sigma^*$ zu den Vorsilben des geschlossenen Regelkreisverhaltens $\operatorname{pre} K$, lediglich anhand ihrer Projektion $\mathrm{p}_o s$ und ihrer Zugehörigkeit zu dem Vorsilbenabschluss der Strecke $\operatorname{pre} L$ entscheidbar sein. Gilt das für alle Vorsilben $s \in \operatorname{pre} K$, dann heißt K *vorsilbennormal bzgl.* (L, Σ_o), in Zeichen $\operatorname{pre} K = (\operatorname{pre} L) \cap \mathrm{p}_o^{-1} \mathrm{p}_o \operatorname{pre} K$.

Zudem wird ein Regler mit trivialen Letztendlichkeitseigenschaften angestrebt. Lokale Entscheidungen des Reglers können dann auf Basis seiner Vergangenheit getroffen werden und hängen nicht von dessen weiteren Entwicklung ab. In diesem Sinne ist eine *kausale Reglerimplementierung* möglich.

Liegt ein Problem unter eingeschränkter Beobachtbarkeit vor, wird der Regler durch Projektion und Vorsilbenbildung aus dem Regelkreiskandidaten gewonnen. Anderenfalls stellt der Vorsilbenabschluss des Kandidaten selbst einen geeigneten Regler dar.

1.2.2 Standardregelkreis

Das Standardentwurfsproblem ist eine Variation des in [LW88] vorgestellten Entwurfsproblems unter eingeschränkter Beobachtbarkeit, in der eine zusätzliche *Bedienerschnittstelle* vorgesehen ist.

Unter „Bediener“ fallen sowohl menschliche Anlagenbenutzer als auch in der Steuerungsarchitektur übergeordnete Objekte, wie z.B. ein übergeordnetes Leitsystem. Die Rolle der

nicht-beobachtbaren Ereignisse in [LW88] nehmen an dieser Stelle die *logischen Ereignisse* ein.

Eine *Lösung* des Standardentwurfsproblems ist ein monolithischer Regler, der konfliktfreies, vollständiges und steuerbares Regelkreisverhalten unter Spezifikationseinschluss erzielt. Aufgrund der Bedienerschnittstelle ist der induzierte geschlossene Regelkreis aber durchaus geeignet um in bereits bestehende Steuerungsarchitekturen eingegliedert zu werden.

Der *Standardregelkreis* wird aus dem *Regler H* und der *Strecke L* gebildet und ist in Abbildung 1.2.1 gezeigt. Regler und Strecke sind ausschließlich über *steuerbare Ereignisse* Σ_{CON} und *nicht-steuerbare Ereignisse* Σ_{UCON} synchronisiert. Mit den *Bedienerereignissen* Σ_{HI} ist im Regler eine Schnittstelle zu übergeordneten Teilen der Steuerungsarchitektur vorgesehen. Weiter dienen *logische Ereignisse* Σ_{LO} einer komponentenorientierten Modellbildung und der Berücksichtigung von nicht-beobachtbaren Ereignissen, insbesondere von Fehlern.

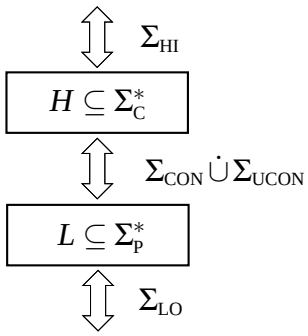


Abbildung 1.2.1: Standardregelkreis

Es bezeichnet Σ das *Gesamtalphabet*, $\Sigma_{\mathrm{P}} \subseteq \Sigma$ die Menge der *Streckenereignisse* und $\Sigma_{\mathrm{C}} \subseteq \Sigma$ die Menge der *Reglerereignisse* gemäß

$$\Sigma := \Sigma_{\mathrm{HI}} \dot{\cup} \Sigma_{\mathrm{LO}} \dot{\cup} \Sigma_{\mathrm{CON}} \dot{\cup} \Sigma_{\mathrm{UCON}}, \tag{1.2}$$

$$\Sigma_{\mathrm{C}} := \Sigma_{\mathrm{HI}} \dot{\cup} \Sigma_{\mathrm{CON}} \dot{\cup} \Sigma_{\mathrm{UCON}}, \tag{1.3}$$

$$\Sigma_{\mathrm{P}} := \Sigma_{\mathrm{LO}} \dot{\cup} \Sigma_{\mathrm{CON}} \dot{\cup} \Sigma_{\mathrm{UCON}}.$$

Sowohl der Regler $H \subseteq \Sigma_{\mathrm{C}}^*$ als auch die Strecke $L \subseteq \Sigma_{\mathrm{P}}^*$ sind dynamische Systeme. Ihre Verkopplung bildet den *Standardregelkreis* $L \parallel H$.

Der Reglerentwurf zielt auf einen Regler mit trivialen Letztendlichkeitseigenschaften, also mit der Eigenschaft

(H0) Abgeschlossenheit unter Vorsilbenbildung, d. h. $\mathrm{pre}\, H = H$,

ab.

Angestrebt wird ein geschlossener Regelkreis $L \parallel H$, in dem sich Strecke und Regler stets auf eine einschrittige Fortsetzung einigen können und die Strecke ihre Letztendlichkeitseigenschaften einlösen kann. Gefordert werden die Lebendigkeitseigenschaften

(SR1) Konfliktfreiheit, d. h. $(\operatorname{pre} L) \parallel H = \operatorname{pre}(L \parallel H)$,

(SR2) Vollständigkeit, d. h. $(\forall s \in (\operatorname{pre} L) \parallel H \, \exists \sigma \in \Sigma)[s\sigma \in (\operatorname{pre} L) \parallel H]$.

Weiter werden dem Regler mit dem Verbieten einzelner Ereignisse lediglich passive Eingriffsmöglichkeiten zugestanden. Mit den nicht-steuerbaren Ereignissen $\Sigma_{\mathrm{uc}} := \Sigma_{\mathrm{UCON}} \dot{\cup} \Sigma_{\mathrm{LO}}$ wird zusätzlich

(SR3) Steuerbarkeit bzgl. $(L, \Sigma_{\mathrm{uc}})$, d. h. $(((\operatorname{pre} L) \parallel H)\Sigma_{\mathrm{uc}}) \cap (\mathrm{p}_{\mathrm{P}}^{-1} \operatorname{pre} L) \subseteq (\operatorname{pre} L) \parallel H$,

gefordert.

Abschließend ist der geschlossene Regelkreis auf die Spezifikation $E \subseteq \Sigma^*$ einzuschränken. Es bleibt die Forderung:

(SR4) Spezifikationseinschluss, d. h. $L \parallel H \subseteq E$.

1.2.3 Standardentwurfsproblem

Die Anforderungen (H0) und (SR1) bis (SR4) stellen die Entwurfsziele des Standardreglerentwurfs dar. Sie sind durch ein gemäß Gl. (1.2) partitioniertes Alphabet, ein Streckenverhalten L und eine Spezifikation E vollständig parametriert. Dementsprechend sind Σ, L und E die Parameter des formalen *Standardentwurfsproblems*.

Definition 1.2.1. Ein *Standardentwurfsproblem* ist ein Tupel (Σ, L, E) mit einem gemäß Gl. (1.2) partitionierten Alphabet Σ, einem Streckenmodell $L \subseteq \Sigma_{\mathrm{P}}^*$ und einer Spezifikation $E \subseteq \Sigma^*$. Eine *Lösung* des Standardentwurfsproblems ist ein Regler $H \subseteq \Sigma_{\mathrm{C}}^*$ mit (H0), der im geschlossenen Regelkreis $L \parallel H$ die Eigenschaften (SR1) bis (SR4) erzielt. □

Nachfolgend wird gezeigt, dass eine Lösung H eines Standardentwurfsproblems (Σ, L, E) aus einem geeigneten Kandidaten $K \subseteq \Sigma^*$ für den geschlossenen Regelkreis abgeleitet werden kann.

Proposition 1.2.1. Gegeben sei ein Standardentwurfsproblem (Σ, L, E) zusammen mit einem Kandidaten $K \subseteq \Sigma^*$ für den geschlossenen Regelkreis. Erfüllt K mit $\Sigma_{uc} := \Sigma_{UCON} \dot{\cup} \Sigma_{LO}$ die Eigenschaften

(K1) Steuerbarkeit bzgl. (L, Σ_{uc}), d. h. $((\operatorname{pre} K)\Sigma_{uc} \cap (\mathrm{p}_P^{-1} \operatorname{pre} L) \subseteq \operatorname{pre} K$,
(K2) Vorsilbennormalität bzgl. (L, Σ_C), d. h. $\operatorname{pre} K = (\mathrm{p}_P^{-1} \operatorname{pre} L) \cap (\mathrm{p}_C^{-1} \mathrm{p}_C \operatorname{pre} K)$,
(K3) Relative Abgeschlossenheit bzgl. L, d. h. $K = (\operatorname{pre} K) \cap \mathrm{p}_P^{-1} L$,
(K4) Vollständigkeit, d. h. $(\forall s \in \operatorname{pre} K \, \exists \sigma \in \Sigma)[s\sigma \in \operatorname{pre} K]$,
(K5) Spezifikationseinschluss, d. h. $K \subseteq E$,

dann ist $H := \mathrm{p}_C \operatorname{pre} K$ eine Lösung des gegebenen Standardentwurfsproblems. Im Umkehrschluss erzielt eine Lösung $H \subseteq \Sigma_C^*$ des gegebenen Standardentwurfsproblems im geschlossenen Regelkreis $K := L \parallel H$ die Eigenschaften (K1) bis (K5). □

Man bemerke, dass die Eigenschaften (H0) und (SR1)-(SR4) unter beliebiger Vereinigung erhalten bleiben. Die Lösungsmenge eines Standardentwurfsproblems bildet also einen vollständigen oberen Halbverband bzgl. mengenwertiger Vereinigung. Entsprechend ist das supremale Element der induzierten Halbordnung bzgl. mengenwertigen Einschlusses selbst eine Lösung des Standardentwurfsproblems und wird nachfolgend als *minimal-restriktive Lösung* $H_V^{\Uparrow} \subseteq \Sigma_C^*$, des Standardentwurfsproblems bezeichnet.

Zudem kann man zeigen, dass die Eigenschaften (K1) bis (K4) unter beliebiger Vereinigung erhalten bleiben, siehe [RW89]. Obiger Argumentation folgend weist das supremale Element bzgl. Mengeneinschluss $K^{\uparrow} \subseteq \Sigma^*$ ebenfalls die Eigenschaften (K1) bis (K4) auf. Aus [MBY+12] kann eine Prozedur zur Berechnung der supremalen Teilsprache der Spezifikation E mit den Eigenschaften (K1) bis (K4) abgeleitet werden. Dazu wird an E die technische Anforderung

(E0) Relative Abgeschlossenheit bzgl. L, d. h. $E = (\operatorname{pre} E) \cap \mathrm{p}_P^{-1} L$.

gestellt. Man kann zeigen, dass eine aus $K^{\uparrow}$ abgeleitete Lösung $H^{\uparrow} := \mathrm{p}_C \operatorname{pre} K^{\uparrow}$ im geschlossenen Regelkreis das gleiche Verhalten erzielt wie die minimal-restriktive Lösung $H^{\Uparrow}$.

Proposition 1.2.2. Gegeben sei ein Standardentwurfsproblem (Σ, L, E). Es bezeichnet $K^{\uparrow} \subseteq \Sigma^*$ die supremale Teilsprache der Spezifikation E bzgl. der Eigenschaften (K1) bis (K4). Weiter sei mit $H^{\Uparrow} \subseteq \Sigma_C^*$ die supremale Lösung des Standardentwurfsproblems (Σ, L, E) gegeben. Dann sind die durch $H^{\uparrow} := \mathrm{p}_C \operatorname{pre} K^{\uparrow}$ und $H^{\Uparrow}$ erzielten Verhalten im geschlossenen Regelkreis identisch, in Zeichen $L \parallel H^{\uparrow} = L \parallel H^{\Uparrow}$. □

Ein formaler Nachweis für die obigen Propositionen 1.2.1 und 1.2.2 findet sich in Anhang A.1.

Für einen praktischen Reglerentwurf wird $K^{\uparrow}$ berechnet und daraus die Lösung $H^{\uparrow}$ gewonnen. Die Bedienerereignisse werden von der Strecke nicht geteilt und der Regler kann ihr Auftreten jederzeit verhindern. Für den Reglerentwurf werden die Bedienerereignisse als steuerbare Eigenschleifen der Strecke zugeschlagen.

Das Standardentwurfsproblem ist durch ein geeignet partitioniertes Alphabet Σ, ein Streckenmodell L und eine Spezifikation E parametriert. So können sowohl das Nominalentwurfsproblem als auch das fehlerverträgliche Entwurfsproblem und Teile der fehlerverdeckenden Steuerungsrekonfiguration auf das Standardentwurfsproblem zurückgeführt werden.

1.3 Nominelle Regelung

Befindet sich eine Strecke im *Nominalbetrieb*, werden fehlerbedingte Verhaltensänderungen ausgeschlossen. Die verbleibende Streckendynamik wird durch das *Streckennominalverhalten* abgedeckt. Der Reglerentwurf zielt in diesem Fall auf einen unter Spezifikationseinschluss konfliktfreien, vollständigen und steuerbaren geschlossenen Regelkreis ab.

Es wird davon ausgegangen, dass der Nominalregelkreis in eine bestehende Steuerungsarchitektur eingebettet ist. Der Nominalregler stellt mit einer Bedienerschnittstelle eine Kommunikationsmöglichkeit mit einer überlagerten Ebene der Steuerungshierarchie bereit.

1.3.1 Nominalregelkreis

Der Nominalregelkreis besteht aus dem *Nominalregler* H_{N} und dem *Nominalstreckenverhalten* L_{N} und ist in Abbildung 1.3.1 dargestellt. Regler und Strecke sind über steuerbare Ereignisse Σ_{CON} und nicht-steuerbare Ereignisse Σ_{UCON} synchronisiert. Der Nominalregler stellt mit den Bedienerereignissen Σ_{HI} eine Schnittstelle zu überlagerten Teilen der Steuerungsarchitektur bereit.

In den nachfolgenden Kapiteln wird das Auftreten eines Fehlers durch ein logisches Ereignis modelliert. Die nominellen Ereignisse und das Fehlerereignis sollen dabei wohlunterscheidbar sein. Entsprechend wird die Menge der *nominellen logischen Ereignisse* $\Sigma_{\mathrm{LO,N}}$ eingeführt.

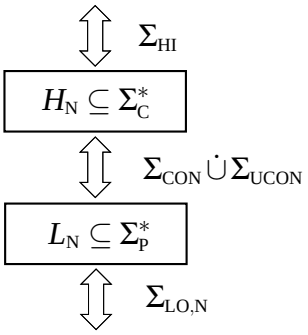


Abbildung 1.3.1: Nominalregelkreis

Es bezeichnet Σ_N die Menge aller *Nominalereignisse*, $\Sigma_{P,N}$ die Menge der *Streckennominalereignisse* und Σ_C die Menge der Reglerereignisse, gegeben durch

$$\begin{aligned} \Sigma_N &:= \Sigma_{HI} \dot\cup \Sigma_{LO,N} \dot\cup \Sigma_{CON} \dot\cup \Sigma_{UCON} \\ \Sigma_{P,N} &:= \Sigma_{LO,N} \dot\cup \Sigma_{CON} \dot\cup \Sigma_{UCON} \\ \Sigma_C &:= \Sigma_{HI} \dot\cup \Sigma_{CON} \dot\cup \Sigma_{UCON}. \end{aligned} \tag{1.4}$$

Sowohl der Nominalregler $H_N \subseteq \Sigma_C^*$ als auch die Nominalstrecke $L_N \subseteq \Sigma_{P,N}^*$ sind dynamische Systeme. Ihre Verkopplung $L_N \parallel H_N$ bildet den *Nominalregelkreis*.

1.3.2 Nominalreglerentwurf

Angestrebt wird ein Nominalregler H_N mit trivialen Letztendlichkeitsbedingungen. Er ist so auszulegen, dass der geschlossene Regelkreis $L_N \parallel H_N$ unter Spezifikationseinschluss bzgl. der *Nominalspezifikation* $E_N \subseteq \Sigma_N^*$, konfliktfrei, vollständig und steuerbar bzgl. $(L_N, \Sigma_{UCON} \dot\cup \Sigma_{LO})$ ist. Im Einzelnen werden die Eigenschaften

- (NH0) Abgeschlossenheit unter Vorsilbenbildung, d. h. $\mathrm{pre}\, H_N = H_N$,
- (NR1) Konfliktfreiheit, d. h. $(\mathrm{pre}\, L_N) \parallel H_N = \mathrm{pre}\,(L_N \parallel H_N)$
- (NR2) Vollständigkeit, d. h. $(\forall s \in (\mathrm{pre}\, L_N) \parallel H_N \, \exists \sigma \in \Sigma)[s\sigma \in (\mathrm{pre}\, L_N) \parallel H_N]$
- (NR3) Steuerbarkeit bzgl. (L_N, Σ_{uc}) mit $\Sigma_{uc} := \Sigma_{UCON} \dot\cup \Sigma_{LO,N}$, d. h. $(((\mathrm{pre}\, L_N) \parallel H_N)\Sigma_{uc} \cap (\mathrm{p}_{P,N}^{-1} \mathrm{pre}\, L_N) \subseteq (\mathrm{pre}\, L_N) \parallel H_N$
- (NR4) Spezifikationseinschluss, d. h. $L_N \parallel H_N \subseteq E_N$

gefordert.

Es ist Σ_N gemäß (1.4) partitioniert. Seine Zellen sind die Bedienerereignisse Σ_{HI}, die nominellen logischen Ereignisse $\Sigma_{LO,N}$ sowie die steuerbaren Ereignisse Σ_{CON} und die

nicht-steuerbaren Ereignisse Σ_{UCON}. Die Anforderungen (NR1) bis (NR4) sind strukturell identisch zu (SR1) bis (SR4) und durch die Angabe von Σ_N, L_N und E_N vollständig parametriert. Das Nominalreglerentwurfsproblem liegt also als Standardentwurfsproblem (Σ_N, L_N, E_N) vor.

Definition 1.3.1. Ein *Nominalentwurfsproblem* ist ein Standardentwurfsproblem (Σ_N, L_N, E_N), mit den Nominalereignissen Σ_N, partitioniert gemäß Gl. (1.4), einem Nominalstreckenverhalten $L_N \subseteq \Sigma^*_{P,N}$ und einer Nominalspezifikation $E_N \subseteq \Sigma^*_N$. Eine Lösung $H^*_N \subseteq \Sigma^*_C$ des Nominalentwurfsproblems (Σ_N, L_N, E_N) heißt *Nominallösung*.

Im Kontext der fehlerverdeckenden Steuerungsrekonfiguration wird der minimal-restriktive Regler für den Nominalfall von Bedeutung sein. Als Standardentwurfsproblem weist auch das Nominalentwurfsproblem eine minimal-restriktive Lösung $H^{\Uparrow}_N$ auf. Für praktische Anwendungen wird die supremale steuerbare, vorsilbennormale, relativ abgeschlossene und vollständige Teilsprache der Nominalspezifikation E_N berechnet und daraus die Nominallösung $H^{\uparrow}_N$ abgeleitet.

Beispiel 1.3.1. Betrachtet wird eine einfache Maschine. Mit den Ereignissen *a1*, *a2* werden zwei verschiedene Prozesse gestartet. Der erfolgreiche Abschluss eines Prozesses wird durch die Ereignisse *A1*, *A2* angezeigt, siehe Abb. 1.3.2.

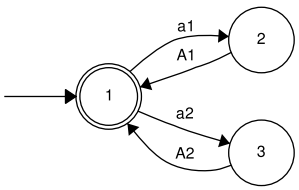


Abbildung 1.3.2: Nominalstreckenverhalten L_N

Während des Nominalbetriebs sollen jeweils das Starten, *ah*, und Beenden, *Ah*, an den Bediener gemeldet werden, siehe Abb. 1.3.3

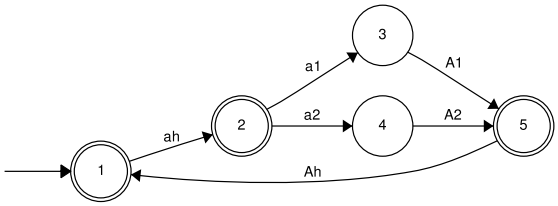


Abbildung 1.3.3: Nominalspezifikation E_N

Der aus der bzgl. (K1) bis (K5) supremalen Teilsprache der Spezifikation $K^{\uparrow}$ resultierende Regler $H_{\mathrm{N}}^{\uparrow}$ ist in Abb. 1.3.4 gezeigt.

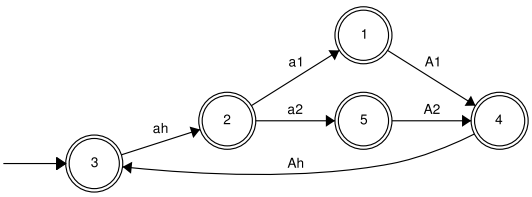


Abbildung 1.3.4: Nominalregler $H_{\mathrm{N}}^{\uparrow}$

Man bemerke, dass der Nominalregler die Bedienereingaben gemäß der Nominalspezifikation umsetzt.

□

1.4 Fehlerverträgliche Regelung

Das Modell einer Strecke heißt *fehlerverträglich*, falls es seine Gültigkeit im Fehlerfall nicht verliert. Dazu bildet ein *fehlerverträgliches Streckenverhalten* sowohl den Nominalfall als auch fehlerinduzierte Verhaltensänderungen korrekt nach. Analog heißt ein Regler *fehlerverträglich*, falls er, trotz des Auftretens eines Fehlers, die Entwurfsziele Konfliktfreiheit, Vollständigkeit und Steuerbarkeit unter Spezifikationseinschluss sichert. Dabei ist das Auftreten eines Fehlers für den Regler weder sichtbar noch unterdrückbar und wird deshalb als nicht-beobachtbares und nicht-steuerbares Ereignis aufgefasst.

Fehlerverträgliche Modelle sind durch [PSL11] motiviert, worin das Verhalten einer fehlerbehafteten Strecke durch das Streckennominalverhalten und durch Modelle der bereits defekten Strecke beschrieben wird. Für jedes einzelne Modell wird dann ein Regler entworfen, auf den nach der Diagnose eines Fehlers umgeschaltet wird. Dabei muss der Anfangszustand des ausgewählten Reglers zum Schaltzeitpunkt abgeglichen werden. Entsprechend sind in diesem Ansatz zusätzliche Diagnose- und Umschaltmechanismen vorgesehen.

Im Gegensatz zu [PSL11] werden in einem *fehlerverträglichen Verhalten* die unterschiedlichen Fehlerfälle in einem Streckenmodell integriert. Dazu wird für jeden Fehlerfall ein Modell entwickelt, das die fehlerbedingte Verhaltensänderung der Strecke nachbildet. Die Vereinigung all dieser Modelle mit dem Nominalverhalten stellt, als ein integriertes Modell der Strecke, das angestrebte fehlerverträgliche Streckenverhalten dar.

Für den Reglerentwurf können die Entwurfsvorgaben ebenfalls in einem fehlerverträglichen Verhalten formalisiert werden. Zusammen mit einem geeignet partitionierten Alphabet formen eine fehlerverträgliche Strecke und eine fehlerverträgliche Spezifikation ein Standardentwurfsproblem, dessen Lösung, ohne Information über das Auftreten eines Fehlers, die Entwurfsvorgaben umsetzen kann. Insbesondere wird im Gegensatz zu [PSL11] weder eine Diagnoseeinrichtung noch ein entsprechender Umschaltmechanismus benötigt.

Im Folgenden werden zunächst fehlerverträgliche Modelle eingeführt, auf Aktuator/Sensor-Systeme angewandt und ihre automatisierte Konstruktion diskutiert. Abschließend werden die Elemente des fehlerverträglichen Regelkreises und das Problem der fehlerverträglichen Regelung abgehandelt.

1.4.1 Fehlerverträgliche Modellierung und fehlerverträglicher Regelkreis

Angelehnt an [PSL11] wird das Auftreten eines Fehlers durch ein nicht-steuerbares und für den Regler nicht-beobachtbares Ereignis F, $F \notin \Sigma_{\mathrm{P,N}}$ modelliert. Im Rahmen des Standardentwurfsproblems nach Abschnitt 1.2 wird das Ereignis F als logisches Ereignis eingestuft und die nominellen logischen Ereignisse $\Sigma_{\mathrm{LO,N}}$ zu $\Sigma_{\mathrm{LO,F}} := \Sigma_{\mathrm{LO,N}} \dot\cup \{F\}$ erweitert. Entsprechend ergibt sich die Menge der *Streckenfehlerereignisse* $\Sigma_{\mathrm{P,F}} := \Sigma_{\mathrm{P,N}} \dot\cup \{F\}$ zu

$$\Sigma_{\mathrm{P,F}} = \Sigma_{\mathrm{LO,N}} \dot\cup \Sigma_{\mathrm{CON}} \dot\cup \Sigma_{\mathrm{UCON}} \dot\cup \{F\} = \Sigma_{\mathrm{LO,F}} \dot\cup \Sigma_{\mathrm{CON}} \dot\cup \Sigma_{\mathrm{UCON}}.$$

Neben dem Streckennominalverhalten L_{N} wird ein *Streckenfehlerverhalten* $L_{\mathrm{D}} \subseteq \Sigma^*_{\mathrm{P,F}}$ erstellt. In L_{D} ist die gesamte Streckenhistorie, die möglicherweise zu dem Auftreten eines Fehlers führen kann, nebst der fehlerinduzierten Streckenverhaltensänderung, abzudecken. Die Vereinigung von L_{N} und L_{D} beschreibt dann das angestrebte *fehlerverträgliche Streckenverhalten*

$$L_{\mathrm{F}} := L_{\mathrm{N}} \cup L_{\mathrm{D}}.$$

Um die fehlerbedingte Verhaltensänderung der Strecke in L_{D} schlüssig nachzubilden, muss für alle Zeichenketten $s' \in L_{\mathrm{D}}$, die ein Fehlerereignis F enthalten, eine Zerlegung $s' = rFt$ mit $r \in \mathrm{pre}\, L_{\mathrm{N}}$ und $t \in \Sigma^*_{\mathrm{P,N}}$ existieren. Dann wird die zu einem Fehler führende Streckenhistorie r durch die nachfolgende Streckenentwicklung t schlüssig in L_{D} fortgesetzt. Es wird also ein fehlerverträgliches Verhalten L_{F} gemäß

$$L_{\mathrm{F}} = L_{\mathrm{N}} \cup \big((\mathrm{pre}\, L_{\mathrm{N}}) F \Sigma^*_{\mathrm{P,N}} \cap L_{\mathrm{D}} \big). \tag{1.5}$$

angestrebt, siehe Abb. 1.4.1.

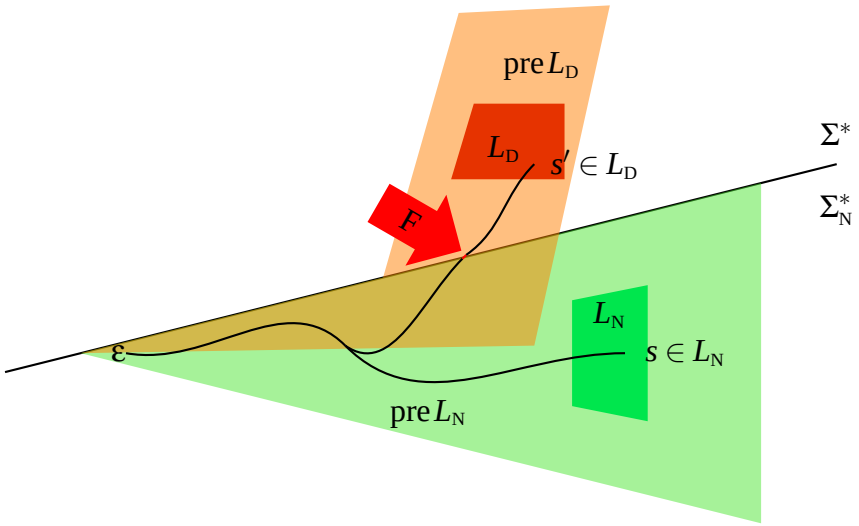


Abbildung 1.4.1: Fehlerverträgliches Verhalten

Um ein fehlerverträgliches Verhalten gemäß Gl. (1.5) zu sichern, muss der Nominalteil jeder Zeichenkette $s' \in L_D$ konsistent mit den Gesetzen des Nominalstreckenverhaltens L_N sein, sowohl das lokale Verhalten als auch die Letztendlichkeitsbedingungen betreffend, d. h.

$$(\operatorname{pre} L_D) \cap \Sigma_N^* \subseteq \operatorname{pre} L_N \tag{1.6}$$

$$L_D \cap \Sigma_N^* \subseteq L_N. \tag{1.7}$$

Definition 1.4.1. Ein *fehlerverträgliches Modell* ist ein Paar (L_N, L_D) mit einem Nominalstreckenverhalten $L_N \subseteq \Sigma_{P,N}^*$ und einem Streckenfehlerverhalten $L_D \subseteq \Sigma_{P,F}^*$. Das fehlerverträgliche Modell (L_N, L_D) heißt *zulässig*, falls die Gln. (1.6) und (1.7) erfüllt sind. □

Gemäß der nachfolgenden Proposition führt ein zulässiges fehlerverträgliches Modell (L_N, L_D) tatsächlich auf ein fehlerverträgliches Streckenverhalten mit der gewünschten Struktur nach Gl. (1.5).

Proposition 1.4.1. Es bezeichnet (L_N, L_D) ein zulässiges fehlerverträgliches Modell. Dann ergibt sich das fehlerverträgliche Verhalten $L_F := L_N \cup L_D$ gemäß Gl. (1.5). □

Beweis. Zu zeigen ist, dass

$$L_N \cup L_D = L_N \cup ((\operatorname{pre} L_N) F \Sigma_{P,F}^* \cap L_D) \tag{1.8}$$

gilt. Die rechtsseitige Einschließung in Gl. (1.8) folgt sofort aus $(\mathrm{pre}\, L_\mathrm{N})F\Sigma^*_{\mathrm{P,F}} \cap L_\mathrm{D} \subseteq L_\mathrm{D}$. Um die linksseitige Einschließung zu zeigen, wird zunächst der Zusammenhang

$$L_\mathrm{D} \subseteq (\mathrm{pre}\, L_\mathrm{N})F\Sigma^*_{\mathrm{P,F}} \cup L_\mathrm{N} \tag{1.9}$$

aus den Gln. (1.6) und (1.7) hergeleitet. Man wähle ein beliebiges $s \in L_\mathrm{D}$. Besteht s nur aus Nominalereignissen, d.h. $s \in \Sigma^*_\mathrm{N}$, dann folgt aus Gl. (1.7), dass $s \in L_\mathrm{N}$ gilt. Falls s ein Fehlerereignis enthält, d. h. $s \notin \Sigma^*_\mathrm{N}$, dann wähle man für s eine Zerlegung $s = rFt, r \in \Sigma^*_\mathrm{N}, t \in \Sigma^*_{\mathrm{P,N}}$. Es ist $r \in \mathrm{pre}\, L_\mathrm{D}$ und mit Gl. (1.6) folgt $r \in \mathrm{pre}\, L_\mathrm{N}$. Daraus folgt dann $s = rFt \in L_\mathrm{N}F\Sigma^*_{\mathrm{P,N}} \cup L_\mathrm{N}$. Damit ist der Beweis von Gl. (1.9) vollständig und die linksseitige Einschließung in Gl. (1.8) leitet sich wie folgt ab:

$$L_\mathrm{F} = L_\mathrm{N} \cup L_\mathrm{D} \subseteq L_\mathrm{N} \cup (\,((\mathrm{pre}\, L_\mathrm{N})F\Sigma^*_{\mathrm{P,N}} \cup L_\mathrm{N}) \cap L_\mathrm{D}\,) \subseteq L_\mathrm{N} \cup (\,(\mathrm{pre}\, L_\mathrm{N})F\Sigma^*_{\mathrm{P,N}} \cap L_\mathrm{D}\,).$$

q.e.d.

Beispiel 1.4.1 (Fortsetzung von Bsp. 1.3.1). In Beispiel 1.3.1 wurde eine einfache Maschine vorgestellt. Abbildung 1.4.2 zeigt das entsprechende Nominalstreckenverhalten L_N.

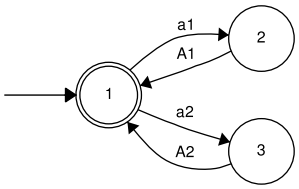


Abbildung 1.4.2: Nominalstreckenverhalten L_N

Es wird nun angenommen, dass Prozess 1 fehlerbehaftet ist. Tritt der Fehler auf, wird anstatt des ersten Prozesses der Prozess 2 ausgeführt und beendet. Nach einem Fehler steht der Maschine lediglich Prozess 2 zur Verfügung. In Abbildung 1.4.3 ist das resultierende Streckenfehlerverhalten L_D gezeigt.

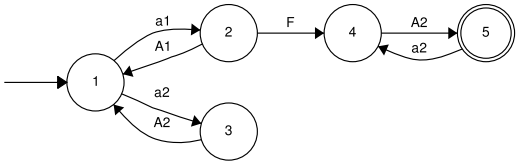


Abbildung 1.4.3: Streckenfehlerverhalten L_D

Man bemerke, dass die Zustände 1, 2, 3 in Abbildung 1.4.3 dem Nominalstreckenverhalten entsprechen. Alle Nominalzeichenketten, die in den Vorsilben von L_D enthalten sind,

sind auch Elemente von preL_N, d. h. es gilt Gl. (1.6). Zudem sind die Letztendlichkeitsbedingungen im Sinne von Gl. (1.7) konsistent. Das fehlerverträgliche Modell (L_N, L_D) ist also zulässig und das fehlerverträgliche Verhalten, $L_F := L_N \cup L_D$, siehe Abb. 1.4.4, weist die in Gl. (1.5) geforderte Struktur auf. □

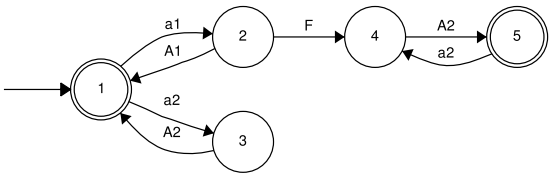


Abbildung 1.4.4: Fehlerverträgliches Streckenverhalten

Die obigen Überlegungen können ohne weiteres von einem Streckenverhalten auf eine Spezifikation übertragen werden. Zu einer Nominalspezifikation $E_N \subseteq \Sigma_N^*$ sei zusätzlich eine *Fehlerspezifikation* $E_D \subseteq \Sigma_F^*$ gegeben. Erfüllt E_D die Anforderungen

$$(\text{pre}\, E_D) \cap \Sigma_N^* \subseteq \text{pre}\, E_N$$
$$E_D \cap \Sigma_N^* \subseteq E_N,$$

so ist für die fehlerverträgliche Spezifikation $E_F := E_N \cup E_D$ eine Struktur gemäß

$$E_F = E_N \cup ((\text{pre}\, E_N) F \Sigma_N^* \cap E_D)$$

zu erwarten.

Automatisierte Konstruktion fehlerverträglicher Modelle für Systeme mit Aktuatoren und Sensoren

Nachfolgend werden technische Systeme betrachtet, die einer Regelung ausschließlich über Sensoren und Aktuatoren zugänglich sind. Unter einem *Aktuator* bzw. einem *Sensor* wird im Folgenden eine Stell- bzw. Messeinrichtung verstanden, deren Zustand durch eine Anzahl endlicher Werte beschrieben werden kann. In dem zu Beginn des Abschnitts 1.4.1 abgesteckten Rahmen werden Aktuatorzustandsänderungen als steuerbare und beobachtbare Ereignisse $\sigma \in \Sigma_{CON}$ aufgefasst, während Sensorzustandsänderungen als nicht-steuerbare, aber beobachtbare Ereignisse $\sigma \in \Sigma_{UCON}$ eingestuft werden. Zudem wird angenommen, dass ein Fehler an das Auftreten eines *kritischen Ereignisses* $\sigma \in \Sigma_{krit} \subseteq \Sigma_{P,N}$ gebunden ist.

Typischerweise besteht in diesem Zusammenhang der Regelkreis aus einer speicherprogrammierbaren Steuerung oder einem vergleichbaren Steuergerät sowie einem Prozess,

der Eingriffe über Aktuatoren erlaubt bzw. Informationen über Sensoren liefert. Reaktionen in der Strecke können durch den Regler dann nur mittelbar, über Änderungen im Prozessabbild der speicherprogrammierbaren Steuerung, erzielt werden. Ebenso verhält es sich mit Streckeninformationen, die dem Regler nur über das Prozessabbild zur Verfügung stehen. Als Beispiel betrachte man das Einschalten eines Motors. Auf Prozessabbildebene entspricht das Ereignis „Motor ein“ dem Setzen einer „1“ in einer Speicherzelle; im Prozess dagegen entspricht dieses Ereignis dem Motorhochlauf, also einer Drehzahländerung von Null auf Sollwert.

Aus Sicht des Reglerentwurfs kann also das reale Prozessverhalten nur mittelbar über das Prozessabbild beeinflusst werden. Vor diesem Hintergrund erfolgt die Modellbildung auf der Ebene des Prozessabbildes. Hier sind lediglich Änderungen in Speicherzellen sichtbar, nicht aber ihre physikalische Ursache bzw. ihre Wirkung. Beispielsweise ist auf Prozessabbildebene ein erneutes Motoreinschalten durchaus möglich. Es hat aber bei einem bereits laufenden Motor keine physikalische Konsequenz. Tatsächlich ist in einem Streckenmodell auf Prozessabbildebene das Setzen/Rücksetzen eines beliebigen Aktuators jederzeit zu berücksichtigen. Analog muss ein Reglermodell sämtliche Sensorereignisse jederzeit erlauben.

In Systemen mit Aktuatoren und Sensoren können bestimmte Fehlertypen identifiziert werden. Jedem Fehlertyp wird ein bestimmtes *Ausfallmuster* zugeordnet. Anhand der verschiedenen Ausfallmustern kann einerseits eine Klassifikation fehlerverträglicher Modelle vorgenommen werden, andererseits lassen sich aus ihnen Prozeduren zur automatisierten Konstruktion des Streckenfehlerverhaltes auf Basis des Streckennominalverhaltens angeben. Für Aktuatoren werden die Ausfallmuster

- Vollständiges Versagen,
- Versagen mit Regeneration und
- Uneingeschränkter Betrieb

betrachtet und für Sensoren werden

- Vollständiger Informationsverlust,
- Informationsverlust mit Regeneration und
- Zufälliges Auslösen

als mögliche Ausfallmuster herangezogen.

Beispiel 1.4.2. Betrachtet wird ein System mit redundanter Aktorik, siehe Abb. 1.4.5, unter dem Ausfallmuster „Vollständiges Versagen“ mit dem kritischen Ereignis *ACT1*.

Algorithmus 1 zeigt ein mögliches Konstruktionsverfahren für ein dem Ausfallmuster

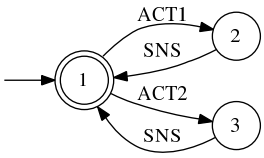


Abbildung 1.4.5: Streckennominalmodell bei redundanter Aktorik

"Vollständiges Versagen" entsprechendes Streckenfehlerverhalten. Zunächst wird die Nominaltransitionenstruktur inklusive der Markierungen dupliziert. Anschließend werden Fehlertransitionen eingefügt und aus den duplizierten Transitionen diejenigen mit kritischen Ereignissen entfernt. Abschließend werden alle nicht erreichbaren und nicht coerreichbaren Zustände entfernt.

Mit den Eingabeparametern L_{N} und $\Sigma_{\mathrm{krit}} = \{ACT1\}$ liefert Algorithmus 1 das in Abb. 1.4.6 dargestellte Streckenfehlerverhalten.

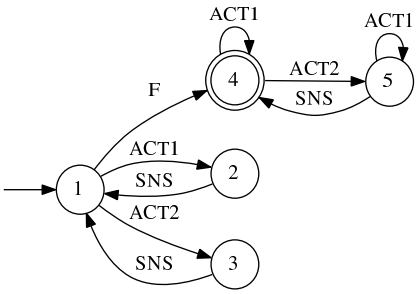


Abbildung 1.4.6: Fehlerverhalten bei redundanter Aktorik und vollständigem Versagen

Der Teilautomat $G|_{\{1,2,3\}}$ spiegelt das Nominalverhalten wieder, während alle Zeichenketten, die in den Zuständen 4 oder 5 enden, das Streckenverhalten nach dem Auftreten des Fehlers beschreiben. Das Fehlerereignis *F* ist an das kritische Ereignis *ACT1* gebunden. Es ist deshalb nur in Zustand 1 erlaubt. Dem Ausfallmuster "Dauerhaftes Versagen" entsprechend werden alle Ereignisketten, die von *ACT1* abhängen, aus dem Modell entfernt. Die nach dem Fehler im Modell verbleibenden Ereignisketten werden durch die Zustände 4,5 repräsentiert. Ihnen ist das Ereignis *ACT1* als Eigenschleife hinzugefügt. Obwohl eine entsprechende Änderung im Prozessabbild stets möglich ist, kann keine physische Änderung in der Strecke erzielt werden. □

Eine detaillierte Abhandlung der automatisierten Konstruktion fehlerverträglicher Modelle findet sich in Anhang B.

Algorithmus 1 L_D-Konstruktion für das Ausfallmuster "Dauerhaftes Versagen"

$G := (Q, \Sigma, \delta, q_0, Q_m), L(G) = L_N$

$H := (X, \Lambda, \xi, x_0, X_m), X = Q, \xi = \delta, \Lambda = \Sigma, x_0 = q_0, X_0 = \emptyset$

$\Lambda_{krit} := \Sigma_{krit}$, $X_0 := X$, $\xi_0 := \xi$

$X_1 := \mathrm{h}X_0$, $\mathrm{h} : X_0 \to X_1$ bijektiv

// Füge Fehlerereignis ein

$\Lambda \leftarrow \Lambda \dot{\cup} \{F\}$

// Dupliziere Nominaltransitionenstruktur

$X \leftarrow X \cup X_1$

for all $(x_1, \alpha, x_2) \in \xi_0$ **do**

$\xi \leftarrow \xi \cup (\mathrm{h}x_1, \alpha, \mathrm{h}x_2)$

end for

for all $x \in X_0$ **do**

if $x \in X_m$ **then**

$X_m \leftarrow X_m \cup \mathrm{h}x$

end if

end for

// Füge Fehlertransitionen ein

for all $(x_1, \alpha, x_2) \in \xi_0$ **do**

if $\alpha \in \Lambda_{krit}$ **then**

$\xi \leftarrow \xi \cup (x_1, F, \mathrm{h}x_2)$

end if

end for

// Entferne Transitionen mit kritischen Ereignissen

for all $(x_1, \alpha, x_2) \in \xi - \xi_0$ **do**

if $\alpha \in \Lambda_{krit}$ **then**

$\xi \leftarrow \xi - (x_1, \alpha, x_2)$

end if

end for

// Trimm

for all $x \in X$ **do**

if $\neg(\exists s \in \Lambda^*)[\xi(x_0, s) = x \ \& \ \neg(\exists t \in \Lambda^*)[\xi(x, t) \in X_m]$ **then**

$X \leftarrow X - \{x\}$

end if

end for

return H

Fehlerverträglicher Regelkreis

Die fehlerverträgliche Steuerungsrekonfiguration ist eine Erweiterung der nominellen Regelung. Entsprechend verfügt der fehlerverträgliche Regler H_{F} mit den Bedienerereignissen Σ_{HI} über eine Bedienerschnittstelle, siehe Abbildung 1.4.7.

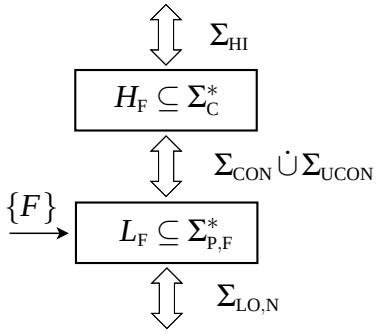


Abbildung 1.4.7: Fehlerverträglicher geschlossener Regelkreis

Wie bereits erwähnt wird das Auftreten eines Fehlers als logisches Ereignis F modelliert. Das Fehlerereignis F und die nominellen logischen Ereignisse sind dabei wohlunterscheidbar, d. h. $F \notin \Sigma_{\mathrm{LO,N}}$. Zur Berücksichtigung von Fehlern beim Reglerentwurf werden die Nominalereignisse Σ_{N} bzw. die Nominalstreckenereignisse $\Sigma_{\mathrm{P,N}}$ um das Fehlerereignis F zu den Alphabeten Σ_{F} und $\Sigma_{\mathrm{P,F}}$ erweitert, während die Reglerereignisse unverändert bleiben, d. h. es gilt:

$$\begin{aligned}
&\Sigma_{\mathrm{F}} := \Sigma_{\mathrm{HI}} \dot\cup \Sigma_{\mathrm{LO,F}} \dot\cup \Sigma_{\mathrm{CON}} \dot\cup \Sigma_{\mathrm{UCON}}, \quad \Sigma_{\mathrm{LO,F}} := \Sigma_{\mathrm{LO,N}} \dot\cup \{F\} \\
&\Sigma_{\mathrm{P,F}} := \Sigma_{\mathrm{LO,F}} \dot\cup \Sigma_{\mathrm{CON}} \dot\cup \Sigma_{\mathrm{UCON}} \dot\cup \{F\} \\
&\Sigma_{\mathrm{C}} := \Sigma_{\mathrm{HI}} \dot\cup \Sigma_{\mathrm{CON}} \dot\cup \Sigma_{\mathrm{UCON}}, \quad \text{vgl. Gl. (1.3).}
\end{aligned} \tag{1.10}$$

Der fehlerverträgliche Regler $H_{\mathrm{F}} \subseteq \Sigma_{\mathrm{C}}^*$ und die fehlerverträgliche Strecke $L_{\mathrm{F}} \subseteq \Sigma_{\mathrm{P,F}}^*$ sind dynamische Systeme. Ihre Verkopplung $L_{\mathrm{F}} \parallel H_{\mathrm{F}}$ bildet den *fehlerverträglichen Regelkreis*.

Ein fehlerverträglicher Regler H_{F} ist so zu entwerfen, dass der fehlerverträgliche Regelkreis trotz des Auftretens eines Fehlers konfliktfrei, vollständig und steuerbar unter Spezifikationseinschluss bzgl. $E_{\mathrm{F}} \subseteq \Sigma_{\mathrm{F}}^*$ ist, im Einzelnen

(FH0) Abgeschlossenheit unter Vorsilbenbildung, d. h. $\mathrm{pre}\, H_{\mathrm{F}} = H_{\mathrm{F}}$,

(FR1) Konfliktfreiheit, d. h. $(\mathrm{pre}\, L_{\mathrm{F}}) \parallel H_{\mathrm{F}} = \mathrm{pre}\,(L_{\mathrm{F}} \parallel H_{\mathrm{F}})$,

(FR2) Vollständigkeit, d. h. $(\forall s \in (\mathrm{pre}\, L_{\mathrm{F}}) \parallel H_{\mathrm{F}}\, \exists \sigma \in \Sigma)[s\sigma \in (\mathrm{pre}\, L_{\mathrm{F}}) \parallel H_{\mathrm{F}}]$,

(FR3) Steuerbarkeit bzgl. $(L_{\mathrm{F}}, \Sigma_{\mathrm{uc}})$, mit $\Sigma_{\mathrm{uc}} := \Sigma_{\mathrm{UCON}} \dot\cup \Sigma_{\mathrm{LO,N}} \dot\cup \{F\}$ d. h.
$((\mathrm{pre}\, L_{\mathrm{F}}) \parallel H_{\mathrm{F}})\, \Sigma_{\mathrm{uc}}) \cap (\mathrm{p}_{\mathrm{P,F}}^{-1}\, \mathrm{pre}\, L_{\mathrm{F}}) \subseteq (\mathrm{pre}\, L_{\mathrm{F}}) \parallel H_{\mathrm{F}}$,

(FR4) Spezifikationseinschluss, d. h. $L_{\mathrm{F}} \parallel H_{\mathrm{F}} \subseteq E_{\mathrm{F}}$.

1.4.2 Entwurf fehlerverträglicher Regler

Man bemerke, dass Σ_F ein gemäß Gl. (1.2) partitioniertes Alphabet mit den Zellen $\Sigma_{HI}, \Sigma_{LO,F}, \Sigma_{CON}$ und Σ_{UCON} ist. Weiter sind die Anforderung (FR1) bis (FR4) strukturell identisch zu (SR1) bis (SR4). Entsprechend formen Σ_F, L_F und E_F ein Standardentwurfsproblem.

Definition 1.4.2. Das *Problem der fehlerverträglichen Regelung* ist ein Standardentwurfsproblem (Σ_F, L_F, E_F) mit Σ_F gemäß Gl. (1.10), einem fehlerverträglichen Streckenmodell $L_F \subseteq \Sigma_{P,F}^*$ und einer fehlerverträglichen Spezifikation $E_F \subseteq \Sigma_F^*$. Eine Lösung $H_F \subseteq \Sigma_C^*$ des fehlerverträglichen Entwurfsproblems (Σ_F, L_F, E_F) heißt *fehlerverträgliche* Lösung. □

Als Standardentwurfsproblem weist auch das fehlerverträgliche Rekonfigurationsproblem eine minimal-restriktive Lösung $H_F^{\Uparrow}$ auf. Praktisch wird die supremale, steuerbare, vorsilbennormale, relativ abgeschlossene und vollständige Teilsprache der fehlerverträglichen Spezifikation E_N berechnet und daraus die fehlerverträgliche Lösung $H_F^{\uparrow}$ abgeleitet.

Beispiel 1.4.3 (Fortsetzung von Bsp. 1.4.1). In Bsp. 1.4.1 wurde für die einfache Maschine aus Bsp. 1.3.1 ein fehlerverträgliches Streckenverhalten erstellt, siehe Abb. 1.4.8.

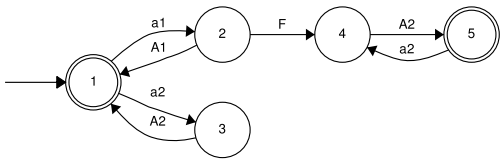


Abbildung 1.4.8: Fehlerverträgliches Streckenverhalten L_F

Ein fehlerverträglicher Regler soll an einen Bediener Prozessbeginn und -ende melden. Im Falle eines Fehlers soll zunächst Prozess 2 neu gestartet werden, bevor eine Meldung erfolgt, siehe Abb. 1.4.9.

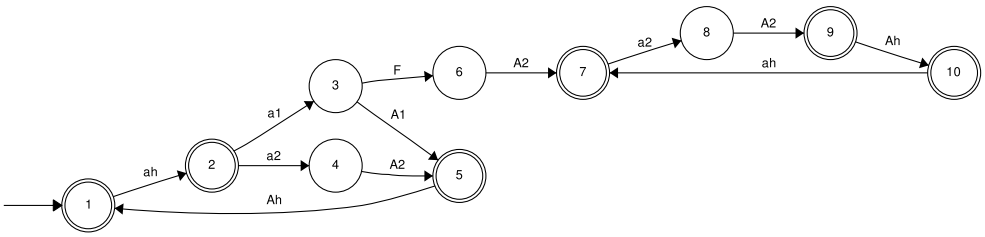


Abbildung 1.4.9: Fehlerverträgliche Spezifikation E_F

Der resultierende minimal-restriktive Regler ist in Abb. 1.4.10 dargestellt.

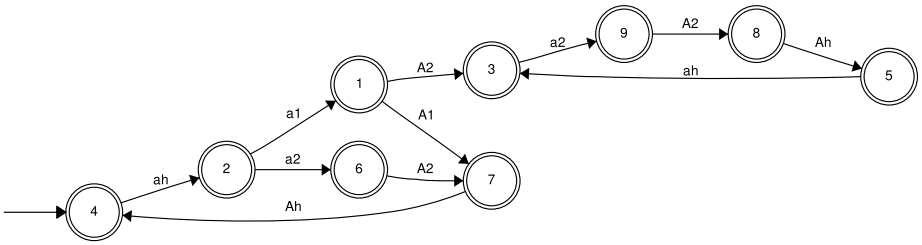


Abbildung 1.4.10: Fehlerverträglicher Regler H_{F}

Man bemerke, dass sich der Nominalteil von H_{F}, d. h. der Teilautomat $G|_{\{1,2,4,6,7\}}$ mit dem Nominalregler aus Beispiel 1.4.1 identisch ist. Zusätzlich wird in Zustand 1 das Auftreten eines Fehlers, angezeigt durch das Ereignis *A2*, berücksichtigt. Nach dem Auftreten eines Fehlers steht Prozess 1 nicht mehr zur Verfügung und der Regler weicht auf Prozess 2 aus. Die Bedienerereignisse werden gemäß der fehlerverträglichen Spezifikation umgesetzt.□

Generell wird von E_{F} nicht die Struktur eines fehlerverträglichen Verhaltens gefordert. Formuliert man allerdings die Spezifikation als fehlerverträgliches Modell $(E_{\mathrm{N}}, E_{\mathrm{D}})$, können die Entwurfsvorgaben für den Nominal- und den Fehlerfall separat und damit systematisch angegeben werden. In $E_{\mathrm{D}} \subseteq \Sigma_{\mathrm{F}}^{*}$ werden dann Entwurfsvorgaben an den Übergang von Nominal- zu fehlerbehafteten Streckenverhalten sowie an das Verhalten der Strecke nach dem Auftreten eines Fehlers abgedeckt. Insbesondere kann die Semantik der Bedienerereignisse Σ_{HI} für die defekte Strecke neu definiert werden.

1.5 Fehlerverdeckende Steuerungsrekonfiguration

Ziel der fehlerverdeckenden Steuerungsrekonfiguration ist es, fehlerbedingte Verhaltensänderungen der Strecke bezüglich eines übergeordneten Bedieners zu verdecken. Im Gegensatz zur fehlerverträglichen Regelung sollen die Bedienerereignisse nicht ausschließlich nach Spezifikationsvorgabe auftreten können, sondern nur dann, falls der Nominalregler das auch erlaubt. Dazu muss der Nominalregler im geschlossenen Regelkreis verbleiben.

Im Allgemeinen sind die Steuerkommandos des Nominalreglers für die fehlerbehaftete Strecke nach dem Auftreten eines Fehlers nicht mehr sinnvoll. Es wird deshalb zwischen Nominalregler und fehlerbehafteter Strecke ein *Rekonfigurator* geschaltet, der die Bedienerereignisse gemäß einer fehlerverträglichen Spezifikation auf die fehlerbehaftete Strecke umsetzt, während der geschlossene Regelkreis konfliktfrei, vollständig und steuerbar, unter Spezifikationseinschluss bleibt.

Im Folgenden wird das Problem der fehlerverdeckenden Steuerungsrekonfiguration für ereignisdiskrete Systeme konkretisiert. Zunächst werden die Elemente des selbstrekonfigurierenden Regelkreises vorgestellt und die Entwurfsziele spezifiziert. Anschließend wird eine kompakte, formale Zusammenfassung des dargelegten Rekonfigurationsproblems angegeben.

1.5.1 Rekonfigurierender Regelkreis

Mit Bezug auf die bisherigen Ergebnisse sei der Nominalregler eine beliebige Lösung $H_{\mathrm{N}} \subseteq \Sigma_{\mathrm{C}}^*$ des Nominalentwurfsproblems $(\Sigma_{\mathrm{N}}, L_{\mathrm{N}}, E_{\mathrm{N}})$ gemäß Abschnitt 1.3. Weiter sei mit dem zulässigen fehlerverträglichen Modell $(L_{\mathrm{N}}, L_{\mathrm{D}})$ bzw. dem induzierten fehlerverträglichen Verhalten $L_{\mathrm{F}} \subseteq \Sigma_{\mathrm{P,F}}^*$, gemäß Abschnitt 1.4.1, ein Modell der fehlerbehafteten Strecke bekannt. Die Entwurfsvorgaben für den Nominal- und Fehlerfall, insbesondere die Semantik der Bedienerereignisse in Termen der Streckenfehlerereignisse $\Sigma_{\mathrm{P,F}}$, seien in $E_{\mathrm{F}} \subseteq \Sigma_{\mathrm{F}}^*$ hinterlegt.

Der Nominalregler H_{N} und die fehlerbehaftete Strecke L_{F} sind über die Ereignisse $\Sigma_{\mathrm{C}} \cap \Sigma_{\mathrm{P,F}} = \Sigma_{\mathrm{CON}} \dot{\cup} \Sigma_{\mathrm{UCON}}$ verkoppelt. Soll der Nominalregler im geschlossenen Regelkreis verbleiben, ist deshalb im Allgemeinen die Umsetzung unterschiedlicher Entwurfsvorgaben für den Nominalbetrieb und den Fehlerbetrieb durch einen fehlerverträglichen Regler H_{F}, auf Grund der Verkopplung über $\Sigma_{\mathrm{CON}} \dot{\cup} \Sigma_{\mathrm{UCON}}$ nicht möglich. Die dazu notwendigen Freiheitsgrade können aber durch das Entkoppeln des Nominalreglers und der fehlerbehafteten Strecke geschaffen werden. Zu diesem Zweck werden die verkoppelnden Ereignisse $\Sigma_{\mathrm{CON}} \dot{\cup} \Sigma_{\mathrm{UCON}}$ schlicht in gedachte – oder *virtuelle* – Entsprechungen umbenannt. In Abgrenzung zu virtuellen Ereignissen werden Σ_{CON} und Σ_{UCON} als *physische Ereignisse* bezeichnet.

Formal werden zu diesem Zweck *virtuelle steuerbare Ereignisse* $\Sigma_{\mathrm{CON,V}}$ und *virtuelle nicht-steuerbare Ereignisse* $\Sigma_{\mathrm{UCON,V}}$ eingeführt; beide disjunkt mit $\Sigma_{\mathrm{P,F}}$ und über eine eineindeutige Abbildung ρ mit ihren physischen Entsprechungen verknüpft, in Zeichen

$$\Sigma_{\mathrm{CON,V}} := \rho\, \Sigma_{\mathrm{CON}}$$
$$\Sigma_{\mathrm{UCON,V}} := \rho\, \Sigma_{\mathrm{UCON}}.$$

Die *virtuellen Ereignisse* Σ_{V} bzw. die *virtuellen Reglerereignisse* $\Sigma_{\mathrm{C,V}}$ ergeben sich dann gemäß

$$\Sigma_{\mathrm{V}} := \Sigma_{\mathrm{HI}} \dot{\cup} \Sigma_{\mathrm{LO}} \dot{\cup} \Sigma_{\mathrm{CON,V}} \dot{\cup} \Sigma_{\mathrm{UCON,V}},$$
$$\Sigma_{\mathrm{C,V}} := \Sigma_{\mathrm{HI}} \dot{\cup} \Sigma_{\mathrm{CON,V}} \dot{\cup} \Sigma_{\mathrm{UCON,V}}.$$

Als Erweiterung der Abbildung ρ wird über dem Gesamtalphabet des Rekonfigurationsproblems

$$\Sigma := \Sigma_{\mathrm{HI}} \dot\cup \Sigma_{\mathrm{LO,N}} \dot\cup \{F\} \dot\cup \Sigma_{\mathrm{CON}} \dot\cup \Sigma_{\mathrm{UCON}} \dot\cup \Sigma_{\mathrm{CON,V}} \dot\cup \Sigma_{\mathrm{UCON,V}}, \tag{1.11}$$

der Operator $\mathrm{h} : \Sigma^* \to \Sigma_{\mathrm{V}}^*$ induktiv gemäß

$$\mathrm{h}\ \varepsilon := \varepsilon$$
$$(\mathrm{h}\ s\sigma) = \begin{cases} (\mathrm{h}s)\rho\sigma, & \text{falls } \sigma \in \Sigma_{\mathrm{CON}} \dot\cup \Sigma_{\mathrm{UCON}}, \\ (\mathrm{h}s)\sigma, & \text{sonst,} \end{cases}$$

definiert. Man bemerke, dass h alle steuerbaren und nicht-steuerbaren Ereignisse $\sigma \in \Sigma_{\mathrm{CON}} \dot\cup \Sigma_{\mathrm{UCON}}$ einer Zeichenkette $s \in \Sigma^*$ durch ihre virtuellen Entsprechungen $\rho\,\sigma \in \Sigma_{\mathrm{CON,V}} \dot\cup \Sigma_{\mathrm{UCON,V}}$ ersetzt. Es kann h in einer zu [CL08] analogen Weise auf Sprachen erweitert werden. Man definiere den *virtualisierten Nominalregler* H_{V} als das Bild des Nominalreglers H_{N} unter *h*, d. h.

$$H_{\mathrm{V}} := \mathrm{h}\, H_{\mathrm{N}} \subseteq \Sigma_{\mathrm{C,V}}^*.$$

Beispiel 1.5.1 (Fortsetzung von Beispiel 1.4.3). Für den minimal-restriktiven Nominalregler für die einfache Maschine aus Beispiel 1.3.1 ergibt sich der virtualisierte minimal-restriktive Nominalregler $H_{\mathrm{V}}^{\uparrow} = \mathrm{h} H_{\mathrm{V}}$ gemäß Abbildung 1.5.1.

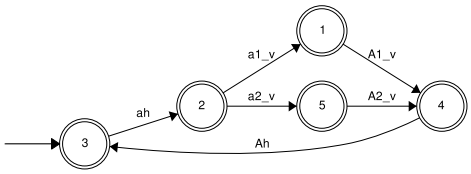


Abbildung 1.5.1: Virtualisierter minimal-restriktiver Nominalregler $H_{\mathrm{V}}^{\uparrow}$

□

Weiter bezeichnen

$$L_{\mathrm{V}} := \mathrm{h}\ L_{\mathrm{N}} \subseteq \Sigma_{\mathrm{P,V}}^*$$
$$E_{\mathrm{V}} := \mathrm{h}\ E_{\mathrm{N}} \subseteq \Sigma_{\mathrm{V}}^*$$

die *virtuelle Nominalstrecke* L_{V} und die *virtuelle Nominalspezifikation* E_{V}. Da *h* nur Ereignisse umbenennt, folgt sofort, dass H_{V} das Standardentwurfsproblem $(\Sigma_{\mathrm{V}}, L_{\mathrm{V}}, E_{\mathrm{V}})$ genau dann löst, wenn H_{N} das Standardentwurfsproblem $(\Sigma_{\mathrm{N}}, L_{\mathrm{N}}, E_{\mathrm{N}})$ löst. Zudem ist $H_{\mathrm{V}}^{\Uparrow} := \mathrm{h}\ H_{\mathrm{N}}^{\Uparrow}$ die minimal-restriktive Lösung des Standardentwurfsproblems $(\Sigma_{\mathrm{V}}, L_{\mathrm{V}}, E_{\mathrm{V}})$.

Die *Virtualisierung* des Nominalreglers führt auf zwei dynamische Systeme über disjunkten Alphabeten, d. h. auf zwei vollständig entkoppelte Systeme, H_V und L_F, siehe Abb. 1.5.2. In Abbildungen werden nicht gekoppelte Systeme mit dem Zeichen $\cdot \parallel \cdot$ gekennzeichnet.

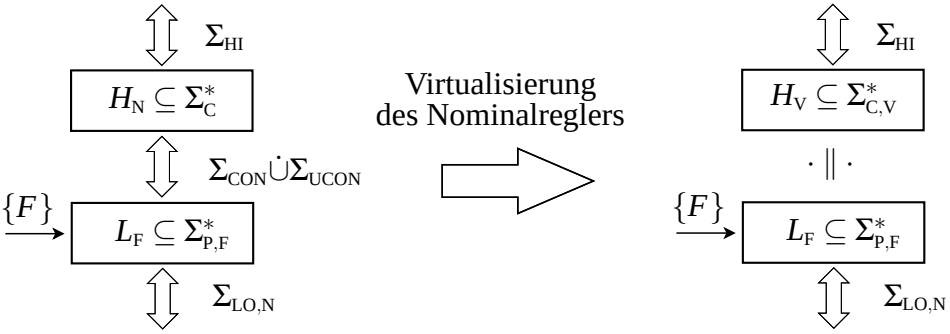


Abbildung 1.5.2: Entkopplung des Nominalreglers und der fehlerbehafteten Strecke

Beziehen sich die Entwurfsvorgaben für die fehlerbehaftete Strecke nicht auf die Bedienerschnittstelle, so kann durch einen fehlerverträglichen Regler, der nicht über den Bedienerereignissen definiert ist, ein zulässiges Regelkreisverhalten erzielt werden. Geht man von einer Spezifikation $\hat{E}_F \subseteq \Sigma_{P,F}^*$ aus, dann sichert eine Lösung $\hat{H}_F \subseteq (\Sigma_C - \Sigma_{HI})^*$ des Entwurfsproblems $(\Sigma_{P,F}, L_F, \hat{E}_F)$ die Entwurfsziele Konfliktfreiheit, Vollständigkeit und Steuerbarkeit unter Spezifikationseinschluss und zwar unabhängig des virtualisierten Nominalreglers, siehe Abb. 1.5.3. Offensichtlich ist unter diesen Voraussetzungen eine Verkopplung des Nominalreglers mit der fehlerbehafteten Strecke über die Bedienerereignisse nicht möglich.

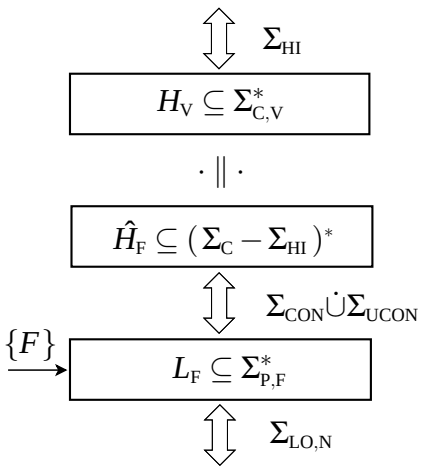


Abbildung 1.5.3: Fehlerbehafteter Regelkreis mit Nominalregler

Um nun den Nominalregler und die fehlerbehaftete Strecke anhand der Bedienerereig-

nisse zu synchronisieren, gehe man von einer Spezifikation $E_F \subseteq \Sigma_F^*$ aus. Man bemerke, dass die Bedienerereignisse Σ_{HI} von der Spezifikation E_F und einem virtuellen Regler H_V wegen $\Sigma_F \cap \Sigma_{C,V} = \Sigma_{HI}$ geteilt werden. Die Spezifikation E_F fordert damit über die Bedienerschnittstelle eine Verkopplung des virtualisierten Nominalreglers und der fehlerbehafteten Strecke. Effektiv werden dadurch im geschlossenen Regelkreis die Bedienerereignisse durch den Nominalregler vorgegeben und durch den Rekonfigurator gemäß der in E_F definierten Semantik auf die fehlerbehaftete Strecke umgesetzt. In diesem Sinne wird der Fehler bzgl. eines Bedieners verdeckt.

Zur Umsetzung der Verkopplung von Nominalregler und fehlerverträglicher Strecke durch die Spezifikation E_F, müssen virtuelle und physische Ereignisse geeignet ineinander umgerechnet werden. Zwischen dem virtualisierten Nominalregler H_V und der fehlerverträglichen Strecke L_F wird deshalb ein *Rekonfigurator* $R \subseteq \Sigma_R^*$ mit

$$\Sigma_R := \Sigma_{CON} \dot\cup \Sigma_{UCON} \dot\cup \Sigma_{CON,V} \dot\cup \Sigma_{UCON,V}$$

platziert, siehe Abb.1.5.4.

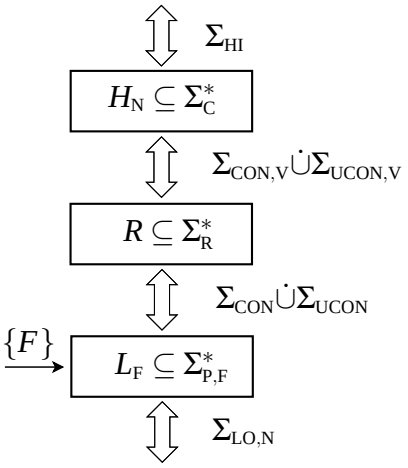


Abbildung 1.5.4: Rekonfigurierender Regelkreis

Mit dem Rekonfigurator R ist die gewollte Verkopplung von Nominalregler und fehlerbehafteter Strecke erreicht, und der *rekonfigurierende Regelkreis* ist durch $L_F \parallel R \parallel H_V$ gegeben. Die Synchronisation über die Bedienerereignisse kann jetzt als Entwurfsvorgabe eingefordert werden.

1.5.2 Entwurfsziele

Angestrebt wird ein Rekonfigurator, der durch eine kausale Rückführung implementierbar ist. Es wird

(R0) Abgeschlossenheit unter Vorsilbenbildung, d. h.
$\mathrm{pre}\, R = R$

gefordert.

Der rekonfigurierende Regelkreis $L_\mathrm{F} \parallel R \parallel H_\mathrm{V}$ soll konfliktfrei und vollständig sein, d. h. es werden die Anforderungen

(R1) Konfliktfreiheit, d. h. $(\mathrm{pre}\, L_\mathrm{F}) \parallel R \parallel H_\mathrm{V} = \mathrm{pre}\,(L_\mathrm{F} \parallel R \parallel H_\mathrm{V})$

(R2) Vollständigkeit, d. h. $(\forall s \in (\mathrm{pre}\, L_\mathrm{F}) \parallel R \parallel H_\mathrm{V}\, \exists \sigma \in \Sigma)[s\sigma \in (\mathrm{pre}\, L_\mathrm{F}) \parallel R \parallel H_\mathrm{V}]$

gestellt. Die Eingriffe des Rekonfigurators sind lediglich passiv. Das Auftreten von nichtsteuerbaren Ereignissen Σ_UCON und virtuellen steuerbaren Ereignisse $\Sigma_\mathrm{UCON,V}$ kann vom Rekonfigurator nicht verhindert werden. Es wird daher

(R3) Steuerbarkeit bzgl. $(L_\mathrm{F} \parallel H_\mathrm{V}, \Sigma_\mathrm{uc})$ mit $\Sigma_\mathrm{uc} := \Sigma_\mathrm{HI} \dot\cup \Sigma_\mathrm{LO} \dot\cup \Sigma_\mathrm{UCON} \dot\cup \Sigma_\mathrm{CON,V}$, d. h.
$((\mathrm{pre}\, L_\mathrm{F}) \parallel R \parallel H_\mathrm{V})\, \Sigma_\mathrm{uc} \cap ((\mathrm{pre}\, L_\mathrm{F}) \parallel H_\mathrm{V}) \subseteq (\mathrm{pre}\, L_\mathrm{F}) \parallel R \parallel H_\mathrm{V}$

gefordert. Abschließend soll der Rekonfigurator die Entwurfsvorgaben in Form einer Spezifikation $E \subseteq \Sigma^*$ sichern. Es soll also

(R4) Spezifikationseinschluss, d. h. $L_\mathrm{F} \parallel R \parallel H_\mathrm{V} \subseteq E$,

gelten, wobei sich $E = E_\mathrm{F} \parallel E_\mathrm{R}$ aus einer fehlerverträglichen Spezifikation $E_\mathrm{F} \subseteq \Sigma_\mathrm{F}^*$ und einer zusätzlichen *Rekonfigurationsspezifikation* $E_\mathrm{R} \subseteq \Sigma^*$ zusammensetzt. In dieser Arbeit wird E_R dazu verwendet, die physischen Ereignisse 1:1 in virtuelle Ereignisse umzusetzen und umgekehrt. In [Ric11] wird diese Anforderung als *Inaktivitätsbedingung* bezeichnet. Formal lässt sich die Inaktivitätsbedingung als regulärer Ausdruck formulieren:

$$E_\mathrm{R} = \mathrm{pre}\,(\,\mathrm{pre}\,(\,\{(\rho\sigma)\sigma \mid \sigma \in \Sigma_\mathrm{CON}\} \cup \{(\sigma\rho\sigma) \mid \sigma \in \Sigma_\mathrm{UCON}\}\,)F\Sigma^*\,).$$

Beispiel 1.5.2 (Fortsetzung von Beispiel 1.4.10). In Abbildung 1.5.5 ist die Umsetzung der Inaktivitätsbedingung für $\Sigma_\mathrm{CON} = \{a1, a2\}$ und $\Sigma_\mathrm{UCON} = \{A1, A2\}$ gezeigt. □

1.5.3 Rekonfigurationsproblem

Durch die Angabe von L_N, E_N und Σ_N wird ein Nominalentwurfsproblem festgelegt und damit auch die Menge aller Nominalregler H_N. Sind zusätzlich L_F und E_F zusammen mit der Rekonfigurationsspezifikation E_R gegeben, so sind die Anforderungen (R1) bis (R4) vollständig parametriert. Demnach legen Σ, L_N, E_N, L_F, E_F und E_R ein Rekonfigurationsproblem fest.

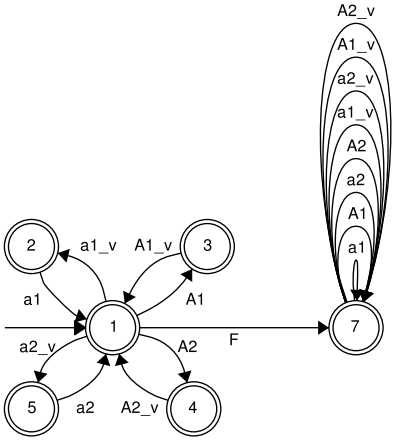


Abbildung 1.5.5: Inaktivitätsbedingungen E_{R}

Definition 1.5.1. Ein *Rekonfigurationsproblem* ist ein Tupel $(\Sigma, L_{\mathrm{N}}, E_{\mathrm{N}}, L_{\mathrm{F}}, E_{\mathrm{F}}, E_{\mathrm{R}})$, in dem das Alphabet Σ gemäß Gl. (1.11) partitioniert ist. Es bezeichnet $L_{\mathrm{N}} \subseteq \Sigma_{\mathrm{P,N}}^*$ ein Nominalstreckenverhalten, $E_{\mathrm{N}} \subseteq \Sigma_{\mathrm{N}}^*$ eine Nominalspezifikation und $L_{\mathrm{F}} \subseteq \Sigma_{\mathrm{P,F}}^*$ ein fehlerverträgliches Streckenverhalten. Weiter ist mit $E_{\mathrm{F}} \subseteq \Sigma_{\mathrm{F}}^*$ eine fehlerverträgliche Spezifikation und mit $E_{\mathrm{R}} \subseteq \Sigma_{\mathrm{R}}^*$ eine Rekonfigurationsspezifikation gegeben. Zu einer festen virtualisierten Lösung $H_{\mathrm{V}} \subseteq \Sigma_{\mathrm{C,V}}^*$ des Nominalentwurfsproblems $(\Sigma_{\mathrm{N}}, L_{\mathrm{N}}, E_{\mathrm{N}})$ heißt ein Rekonfigurator $R \subseteq \Sigma_{\mathrm{R}}^*$ *Lösung* des Rekonfigurationsproblems, falls R unter Vorsilbenbildung abgeschlossen ist (R0) und der rekonfigurierende Regelkreis $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}$ die Eigenschaften (R1) bis (R4) aufweist. Weiter heißt ein unter Vorsilbenbildung abgeschlossener Rekonfigurator $R \subseteq \Sigma_{\mathrm{R}}^*$ *universelle* Lösung, falls für eine beliebige virtualisierte Lösung $H_{\mathrm{V}} \subseteq \Sigma_{\mathrm{C,V}}^*$ des Nominalentwurfsproblems $(\Sigma_{\mathrm{N}}, L_{\mathrm{N}}, E_{\mathrm{N}})$ der rekonfigurierende Regelkreis $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}$ die Eigenschaften (R1) bis (R4) besitzt. □

Ergebnisstand. Als Vorbereitung auf den Entwurf universeller Rekonfiguratoren wurden die Elemente des rekonfigurierenden Regelkreises, der Nominalregler, der Rekonfigurator und die fehlerverträgliche Strecke, diskutiert sowie die entsprechenden Entwurfsziele formuliert. Ausgehenden von formalen Sprachen und endlichen Automaten und der Supervisory Control Theory als formalen Rahmen wurde der Nominalregler als Lösung eines Standardentwurfsproblems charakterisiert. Zur Modellierung fehlerbehafteter Strecken wurden fehlerverträgliche Modelle herangezogen und mit der fehlerverträglichen Regelung ein Ansatz zur fehlertoleranten Regelung bereitgestellt. Dieser wurde dann zur fehlerverdeckenden Steuerungsrekonfiguration erweitert. In deren Mittelpunkt steht der Entwurf eines Rekonfigurators, der für einen beliebigen Nominalregler die Eingaben eines übergeordneten Bedieners geeignet auf die fehlerbehaftete Strecke umsetzt und gleichzeitig einen steuerbaren und lebendigen rekonfigurierenden Regelkreis sicherstellt.

Kapitel 2

Entwurf universeller Rekonfiguratoren

Ausgehend von einem Rekonfigurationsproblem $(\Sigma, L_N, E_N, L_F, E_F, E_R)$ wird in diesem Kapitel der Entwurf entsprechender universeller Lösungen diskutiert. Von einer, im Sinne der Definition 1.5.1, universellen Lösung wird erwartet, dass sie die Entwurfsziele Konfliktfreiheit (R1), Vollständigkeit (R2) und Steuerbarkeit (R3) unter Spezifikationseinschluss (R4), für einen beliebigen virtualisierten Nominalregler erzielt. Effektiv resultiert die Berücksichtigung einer beliebigen Nominallösung in einer All-Quantifizierung über die Nominalregler bzgl. der Eigenschaften (R1) bis (R4). Diese stellt die wesentliche Herausforderung für den Entwurf universeller Rekonfiguratoren dar. Diese Einführung soll das Konzept für den Entwurf universeller Rekonfiguratoren anhand des Sonderfalls eines bekannten Nominalreglers verdeutlichen. Die besagte All-Quantifizierung stürzt dann auf den bereits bekannten Nominalregler zusammen.

Ist der virtualisierte Nominalregler H_V bekannt, dann ist die verbleibende Unbekannte der Rekonfigurator R, also der Entwurfsgegenstand selbst. Man kann dann die fehlerverträgliche Strecke L_F und den virtualisierten Regler H_V formal als Strecke auffassen, während R die Rolle eines Reglers übernimmt. Der rekonfigurierende Regelkreis, wie in Abbildung 1.5.4 gezeigt, geht dann in die Darstellung aus Abbildung 2.0.1 über.

Aus diesem Blickwinkel besteht die Aufgabe des Rekonfigurators darin, die fehlerverträgliche Strecke L_F und den virtualisierten Nominalregler H_V zu koordinieren. Mit der fehlerverträglichen Spezifikation E_F kann dabei die Semantik der Bedienerereignisse neu definiert werden, während mit der Rekonfigurationsspezifikation E_R die Umsetzung physischer und virtueller Ereignisse, z. B. durch die Inaktivitätsbedingungen gemäß Gl. (2.1), vorgegeben werden kann.

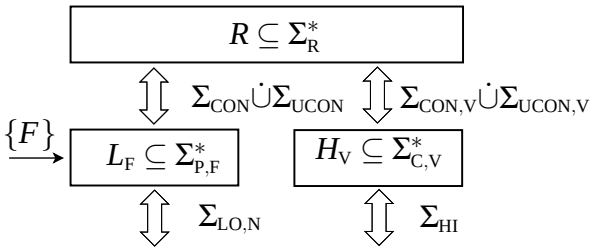


Abbildung 2.0.1: Rekonfigurierender Regelkreis aus Sicht des Rekonfiguratorentwurfs

Fasst man L_F und H_V zur *formalen Strecke* $L_F \parallel H_V$ zusammen, so stellt sich der Rekonfiguratorentwurf als ein Standardentwurfsproblem gemäß Abschnitt 1.2 dar und der in Abb. 2.0.1 gezeigte Regelkreis geht in Abb. 2.0.2 über.

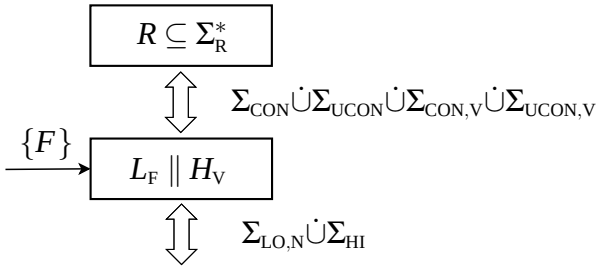


Abbildung 2.0.2: Selbst-Rekonfigurierender Regelkreis - Entwurf

Man setze $L := L_F \parallel H_V$ und $H := R$ und bemerke, dass sich die Anforderungen (R0) und (R1) bis (R4) sowie (H0) bzw. (SR1) bis (SR4) decken. Weiter unterteile man Σ gemäß

$$\Sigma = \{\} \dot\cup (\Sigma_{LO,N} \dot\cup \Sigma_{HI} \dot\cup \{F\}) \dot\cup (\Sigma_{CON} \dot\cup \Sigma_{UCON,V}) \dot\cup (\Sigma_{CON,V} \dot\cup \Sigma_{UCON}) \tag{2.1}$$

und bemerke, dass Σ ein gemäß Gl. (1.2) partitioniertes Alphabet mit den Zellen $\{\}$, $\Sigma_{LO,N} \dot\cup \Sigma_{HI} \dot\cup \{F\}$, $\Sigma_{CON} \dot\cup \Sigma_{UCON,V}$ und $\Sigma_{CON,V} \dot\cup \Sigma_{UCON}$ ist. Es liegt also ein Standardentwurfsproblem (Σ, L, E) mit $L = L_F \parallel H_V$ und $E = E_F \parallel E_R$ vor.

Korollar 2.0.1. Gegeben ist ein Rekonfigurationsproblem $(\Sigma, L_N, E_N, L_F, E_F, E_R)$ zusammen mit einer virtualisierten Lösung $H_V \subseteq \Sigma_{C,V}^*$ des Nominalentwurfsproblems (Σ_N, L_N, E_N). Eine Lösung $R \subseteq \Sigma_R^*$ des Standardentwurfsproblems $(\Sigma, L_F \parallel H_V, E_R \parallel E_F)$ ist eine Lösung des gegebenen Rekonfigurationsproblems und umgekehrt. □

Lässt man die Annahme eines bekannten Nominalreglers fallen, so kommt die Quantifizierung über alle Nominalregler wieder zum Tragen. Man bemerke, dass zu einem nicht näher bekannten virtualisierten Nominalregler H_V mit der virtualisierten minimalrestriktiven Lösung $H_V^\uparrow$ des Nominalentwurfsproblems (Σ_N, L_N, E_N) die beste, bekannte

und berechenbare Abschätzung für H_{V} gegeben ist. Es wird dann $L_{\mathrm{F}} \parallel H_{\mathrm{V}}^{\uparrow}$ als formale Strecke aufgefasst. Ähnlich dem Vorgehen zur Lösung des Standardentwurfsproblems wird wieder ein Kandidat $K \subseteq \Sigma^*$ für das Verhalten des selbst-rekonfigurierenden Regelkreises gesucht, aus dem durch Projektion und Bildung des Vorsilbenabschlusses eine universelle Lösung R des gegebenen Rekonfigurationsproblems abgeleitet werden kann. Dazu werden schrittweise Bedingungen an K gestellt.

Es wird sich zeigen, dass die Steuerbarkeitsbedingung (R3) und der Spezifikationseinschluss (R4) für einen beliebigen Nominalregler ohne weitere Anforderungen an K erfüllt werden kann. Das gilt jedoch nicht für die Lebendigkeitseigenschaften Konfliktfreiheit (R1) und Vollständigkeit (R2). Im ersten Teil dieses Kapitels werden (R2) und (R3) diskutiert. Am Ende stehen hinreichende Bedingungen an einen Kandidaten K, sodass der abgeleitete Rekonfigurator R eine universelle Lösung des gestellten Rekonfigurationsproblems ist. Allerdings werden dabei lediglich triviale Letztendlichkeitsbedingungen berücksichtigt. In einem zweiten Schritt werden, aufbauend auf den Resultaten für abgeschlossene Sprachen, hinreichende Bedingungen zur Verifikation der Konfliktfreiheit (R1) angegeben. Dazu wird auf Resultate des abstraktionsbasierten Reglerentwurfs, siehe [TM], zurückgegriffen. Anschließend wird der Entwurf optimaler Rekonfiguratoren diskutiert und für reguläre Sprachen eine Entwurfsprozedur abgeleitet. Mit [MBY+12] steht dazu ein passender Rahmen bereit.

2.1 Reglerentwurf für triviale Letztendlichkeitsbedingungen

Hauptergebnis dieses Abschnitts sind hinreichende Bedingungen, formuliert in Termen eines Regelkreiskandidaten $K \subseteq \Sigma^*$, sodass ein Rekonfigurator

$$R := \mathrm{p}_{\mathrm{R}} \operatorname{pre} K \tag{2.2}$$

im rekonfigurierenden Regelkreis die Eigenschaften (R1) bis (R4) für einen beliebigen Nominalregler sicherstellt. Effektiv sind die Ergebnisse dieses Kapitels auf Systeme mit trivialen Letztendlichkeitsbedingungen beschränkt.

Die Diskussionen in diesem Abschnitt beschränken sich im Wesentlichen auf die Vollständigkeit (R2) und die Steuerbarkeitsbedingung (R3). Man bemerke, dass diese Eigenschaften ausschließlich in Termen der Vorsilben des rekonfigurierenden Regelkreises formuliert sind. Es ist daher nicht zwingend notwendig und zudem ungünstig, mit Blick

auf eine spätere Integration mit Ergebnissen zur Konfliktfreiheit bei nicht-trivialen Letztendlichkeitsbedingungen, L_F und K von vornherein als abgeschlossen anzunehmen. Man bemerke allerdings, dass für eine unter Vorsilbenbildung abgeschlossene fehlerverträgliche Strecke der rekonfigurierende Regelkreis ohnehin konfliktfrei und (R1) damit trivial erfüllt ist.

Für die weitere Diskussion definiere man die formale Strecke

$$L := L_F \parallel H_V^{\uparrow}. \tag{2.3}$$

und bemerke, dass $\mathrm{pre}\, L = (\mathrm{pre}\, L_F) \parallel H_V^{\uparrow}$ ist.

Um sicherzustellen, dass ein Rekonfigurator R gemäß Gl. 2.2 tatsächlich den minimalrestriktiven rekonfigurierenden Regelkreis $L_F \parallel R \parallel H_V^{\uparrow}$ erzielt, d. h.

$$\mathrm{pre}\, K = (\mathrm{pre}\, L_F) \parallel R \parallel H_V^{\uparrow} \tag{2.4}$$

$$K = L_F \parallel R \parallel H_V^{\uparrow}, \tag{2.5}$$

fordere man von K

(M1) Vorsilbennormalität bzgl. (L, Σ_R), d. h. $\mathrm{pre}\, K = (\mathrm{pre}\, L) \cap \mathrm{p}_R^{-1}\, \mathrm{p}_R\, \mathrm{pre}\, K$,
(M2) Relative Abgeschlossenheit bzgl. L, d. h. $K = (\mathrm{pre}\, K) \cap L$,

und folge der anschließenden Ableitung zum Nachweis der Gleichungen (2.4) und (2.5).

$$\begin{aligned}
\mathrm{pre}\, K &= (\mathrm{p}_{P,F}^{-1}\, \mathrm{pre}\, L_F) \cap (\mathrm{p}_{C,V}^{-1} H_V^{\uparrow}) \cap (\mathrm{p}_R^{-1}\, \mathrm{p}_R\, \mathrm{pre}\, K) && [\text{aus (M1) und Gl. (2.3)}] \\
&= (\mathrm{p}_{P,F}^{-1}\, \mathrm{pre}\, L_F) \cap (\mathrm{p}_R^{-1} R) \cap (\mathrm{p}_{C,V}^{-1} H_V^{\uparrow}) && [\text{aus Gl. (2.2)}] \\
&= (\mathrm{pre}\, L_F) \parallel R \parallel H_V^{\uparrow}, \\
K &= (\mathrm{pre}\, K) \cap (L_F \parallel H_V^{\uparrow}) && [\text{aus (M2)}] \\
&= ((\mathrm{pre}\, L_F) \parallel R \parallel H_V^{\uparrow}) \cap (L_F \parallel H_V^{\uparrow}) && [\text{aus Gl. 2.4}] \\
&= L_F \parallel R \parallel H_V^{\uparrow}.
\end{aligned}$$

Steuerbarkeit. Mit Blick auf die Steuerbarkeitsbedingung (R3), erinnere man zunächst die nicht-steuerbaren Ereignisse

$$\Sigma_{uc} := \Sigma_{HI} \dot\cup \Sigma_{LO} \dot\cup \Sigma_{UCON} \dot\cup \Sigma_{CON,V}.$$

Man bemerke, dass H_V und $H_V^{\uparrow}$ als Lösung des Standardentwurfsproblems (Σ_V, L_V, H_V) steuerbar bzgl. $(L_V, \Sigma_{UCON,V})$ sind. Lässt also $H_V^{\uparrow}$ ein $\sigma \in \Sigma_{CON,V}$ zu, so gilt das auch für H_V. Alle anderen nicht-steuerbaren Ereignisse sind keine Elemente von $\Sigma_{C,V}$ und können von H_V deshalb auch nicht unterdrückt werden. Bezüglich der Steuerbarkeit (R3) wird deshalb von K zusätzlich

(M3) Steuerbarkeit bzgl. (L, Σ_{uc}), d. h. $(\operatorname{pre} K)\,\Sigma_{uc} \cap (\operatorname{pre} L) \subseteq \operatorname{pre} K$

gefordert, wobei gemäß Gl. (2.3) $L = L_F \parallel H_V^{\uparrow}$ gilt.

Lemma 2.1.1. Gegeben sei ein Rekonfigurationsproblem $(\Sigma, L_N, E_N, L_F, E_F, E_R)$, die virtualisierte minimal-restriktive Lösung $H_V^{\uparrow} \subseteq \Sigma_{C,V}^*$ des Nominalentwurfsproblems (Σ_N, L_N, E_N) und eine Sprache $K \subseteq \Sigma^*$ mit den Eigenschaften (M1) bis (M3). Für eine beliebige virtualisierte Lösung $H_V \subseteq \Sigma_{C,V}^*$ des Nominalentwurfsproblems (Σ_N, L_N, E_N) sichert der Rekonfigurator $R := p_R \operatorname{pre} K$ die Steuerbarkeitsbedingung (R3) des rekonfigurierenden Regelkreises $L_F \parallel R \parallel H_V$. □

Beweis. Zu einer beliebigen virtualisierten Lösung $H_V \subseteq \Sigma_{C,V}^*$ des Nominalentwurfsproblems (Σ_N, L_N, E_N), wähle man $s \in (\operatorname{pre} L_F) \parallel R \parallel H_V$ und $\sigma \in \Sigma_{uc}$ mit $s\sigma \in (\operatorname{pre} L_F) \parallel H_V$ beliebig. Es ist $H_V^{\uparrow}$ minimal-restriktiv, sodass auch $s\sigma \in (\operatorname{pre} L_F) \parallel H_V^{\uparrow}$ und $s \in (\operatorname{pre} L_F) \parallel R \parallel H_V \subseteq (\operatorname{pre} L_F) \parallel R \parallel H_V^{\uparrow}$ gilt. Aus der letzten Aussage, zusammen mit Gl. (2.4), folgt $s \in \operatorname{pre} K$, und gemäß der Steuerbarkeitsbedingung (M3) erhält man $s\sigma \in \operatorname{pre} K$. Mit Gl. (2.4) folgt dann $s\sigma \in \operatorname{pre} K = (\operatorname{pre} L_F) \parallel R \parallel H_V^{\uparrow}$. Um den Beweis der Steuerbarkeitsbedingung (R3) abzuschließen, erinnere man, dass $s\sigma \in (\operatorname{pre} L_F) \parallel H_V$ angenommen wurde. Folglich ist $s\sigma \in (\operatorname{pre} L_F) \parallel R \parallel H_V$. q.e.d.

Vollständigkeit. Die tatsächlich vorliegende formale Strecke $L_F \parallel H_V$ ist wegen $H_V \subseteq H_V^{\uparrow}$ lediglich eine Teilmenge der dem Entwurf zu Grunde liegenden Strecke $L_F \parallel H_V^{\uparrow}$. Tatsächlich kann der virtualisierte Nominalregler H_V Zeichenketten, die ausschließlich aus Bedienerereignissen bestehen, verbieten und so die Vollständigkeit von K gefährden.

Mit der Forderung nach

(M4) $\Sigma - \Sigma_{HI}$ Vollständigkeit, d. h. $(\forall s \in \operatorname{pre} K\, \exists t \in \Sigma_{HI}^*, \sigma \in (\Sigma - \Sigma_{HI}))[st\sigma \in \operatorname{pre} K]$

wird für ein beliebiges $s \in \operatorname{pre} K$ eine alternative Fortsetzung $t \in \Sigma_{HI}^*$ zu s in $\operatorname{pre} K$ gefordert, die mit einem $\sigma \notin \Sigma_{HI}$ endet. Die Vollständigkeit von K kann folglich durch verbotene Bedienerereignisse nicht mehr gefährdet werden.

Eine weitere kritische Situation entsteht, falls im Nominalregelkreis neben einem nicht-steuerbaren Ereignis auch ein steuerbares Ereignis erlaubt ist. Im rekonfigurierenden Regelkreis kann dann der Rekonfigurator das virtuelle nicht-steuerbare Ereignis verbieten, während ein virtualisierter Nominalregler das virtuelle steuerbare Ereignis verbietet und stattdessen auf das virtuelle nicht-steuerbare Ereignis wartet.

Diese Situation wird mit der Anforderung

(M5) Schwache Sensorkonsistenz, d. h.
$$(\forall s \in \operatorname{pre} K)\ [(\mathrm{p}_{\mathrm{P,V}}\, s)\Sigma_{\mathrm{UCON,V}} \cap (\operatorname{pre} L_{\mathrm{V}}) \neq \emptyset \Rightarrow s(\Sigma - \Sigma_{\mathrm{C,V}})^* \Sigma_{\mathrm{UCON,V}} \cap \operatorname{pre} K \neq \emptyset]$$

adressiert. Man betrachte ein beliebiges $s \in \operatorname{pre} K$, das durch ein virtuelles nicht-steuerbares Ereignis $\sigma \in \Sigma_{\mathrm{UCON,V}}$ fortgesetzt werden kann, in Zeichen $(\mathrm{p}_{\mathrm{P,V}}\, s)\Sigma_{\mathrm{UCON,V}} \cap (\operatorname{pre} L_{\mathrm{V}}) \neq \emptyset$. Anforderung (M5) fordert für s eine Fortsetzung im Kandidaten K, die ebenfalls mit einem virtuellen nicht-steuerbaren Ereignis endet, aber deren Auftreten von einem beliebigen virtualisierten Nominalregler H_{V} nicht beeinflusst werden kann, in Zeichen $s(\Sigma^* - \Sigma^*_{\mathrm{C,V}})\Sigma_{\mathrm{UCON,V}} \cap \operatorname{pre} K \neq \emptyset$. Die Gefährdung der Vollständigkeit durch das Verbieten eines nicht-steuerbaren virtuellen Ereignisses durch den Rekonfigurator ist damit ausgeschlossen.

Abschließend wird mit

(M6) Nominalstreckeneinschluss, d. h. $\operatorname{pre} K \subseteq \mathrm{p}^{-1}_{\mathrm{P,V}} \operatorname{pre} L_{\mathrm{V}}$,

gefordert, dass die Abfolge virtueller Streckenereignisse $\Sigma_{\mathrm{LO}} \dot\cup \Sigma_{\mathrm{CON,V}} \dot\cup \Sigma_{\mathrm{UCON,V}}$ im rekonfigurierenden Regelkreis konsistent mit den dynamischen Gesetzen des virtualisierten Nominalstreckenverhaltens ist.

Lemma 2.1.2. Gegeben sei ein Rekonfigurationsproblem $(\Sigma, L_{\mathrm{N}}, E_{\mathrm{N}}, L_{\mathrm{F}}, E_{\mathrm{F}}, E_{\mathrm{R}})$, die virtualisierte minimal-restriktive Lösung $H^{\uparrow}_{\mathrm{V}} \subseteq \Sigma^*_{\mathrm{C,V}}$ des Nominalentwurfsproblems $(\Sigma_{\mathrm{N}}, L_{\mathrm{N}}, E_{\mathrm{N}})$ und eine Sprache $K \subseteq \Sigma^*$ mit den Eigenschaften (M1) bis (M6). Für eine beliebige virtualisierte Lösung $H_{\mathrm{V}} \subseteq \Sigma^*_{\mathrm{C,V}}$ des Nominalentwurfsproblems $(\Sigma_{\mathrm{N}}, L_{\mathrm{N}}, E_{\mathrm{N}})$ erfüllt der Rekonfigurator $R := \mathrm{p}_{\mathrm{R}} \operatorname{pre} K$ die Lebendigkeitseigenschaft (R2) des rekonfigurierenden Regelkreises $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}$. □

Beweis. Man wähle einen beliebigen virtualisierten Nominalregler H_{V} und ein beliebiges $s \in (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}$ und zeige die Existenz eines $\sigma \in \Sigma$ mit $s\sigma \in (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}$. Um den nachfolgenden Beweis zu strukturieren, wird die Menge aller von H_{V} erlaubten Ereignisse $\gamma_{\mathrm{H}} := \{\sigma \in \Sigma_{\mathrm{C,V}} \mid (\mathrm{p}_{\mathrm{C,V}}\, s)\sigma \in H_{\mathrm{V}}\}$ und die Menge aller von L_{F} und R gleichzeitig erlaubten Ereignisse $\gamma_{\mathrm{LR}} := \{\sigma \in \Sigma_{\mathrm{R}} \cup \Sigma_{\mathrm{P,F}} \mid (\mathrm{p}_{\mathrm{P,F}}\, s)\sigma \in \operatorname{pre} L_{\mathrm{F}} \text{ und } (\mathrm{p}_{\mathrm{R}}\, s)\sigma \in R\}$ eingeführt. Daneben erinnere man, dass H_{V} das Standardentwurfsproblem $(\Sigma_{\mathrm{V}}, L_{\mathrm{V}}, H_{\mathrm{V}})$ löst, sodass $L_{\mathrm{V}} \parallel H_{\mathrm{V}}$ vollständig ist. Weiter folgt aus Gl. (2.4) und der Vollständigkeit (M4) von K, dass $\gamma_{\mathrm{LR}} \neq \emptyset$ gilt. Man unterscheide die folgenden Fälle:

Fall 1a: $\gamma_{\mathrm{H}} \cap \Sigma_{\mathrm{HI}} \neq \emptyset$, H_{V} und L_{F} entwickeln sich unabhängig voneinander. Man wähle $\sigma \in \gamma_{\mathrm{H}} \cap \Sigma_{\mathrm{HI}}$. Aus $(\Sigma_{\mathrm{R}} \cup \Sigma_{\mathrm{P,F}}) \cap \Sigma_{\mathrm{HI}} = \emptyset$ folgt, dass $s\sigma \in (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}$ gilt, woraus (R2)

folgt. Daher wird für das Folgende $\gamma_H \subseteq \Sigma_{CON,V} \dot{\cup} \Sigma_{UCON,V}$ angenommen.

Fall 1b: $\gamma_{LR} \cap \Sigma_{P,F} \neq \emptyset$*,* $L_F \parallel R$ *kann sich unabhängig von* H_V *entwickeln.* Man wähle $\sigma \in \gamma_{LR} \cap \Sigma_{P,F}$ und bemerke, dass aus $\Sigma_{P,F} \cap \Sigma_{C,V} = \emptyset$ sofort $s\sigma \in (\operatorname{pre} L_F) \parallel R \parallel H_V$ folgt, was (R2) impliziert. Man nehme deshalb im Weiteren $\gamma_{LR} \subseteq \Sigma_{CON,V} \dot{\cup} \Sigma_{UCON,V}$ an.

Zusammenfassend zeigt Fall 1 die Vollständigkeit des rekonfigurierenden Regelkreises dann, wenn H_V und $L_F \parallel R$ sich unabhängig voneinander entwickeln können. Es bleibt der Fall zu diskutieren, in dem $L_F \parallel R$ und H_V sich synchron entwickeln, d. h. $\gamma_{LR} \cap \gamma_H$ ist nicht leer.

Fall 2a: $\gamma_H \cap \Sigma_{CON,V} \neq \emptyset$. Angenommen es gibt ein $\sigma \in \gamma_H \cap \Sigma_{CON,V}$, dann gilt $p_{P,F}(s\sigma) = p_{P,F}(s) \in \operatorname{pre} L_F$ und $p_{C,V}(s\sigma) \in H_V^{\uparrow}$, letzteres wegen $H_V \subseteq H_V^{\uparrow}$. Deshalb gilt $s\sigma \in (\operatorname{pre} L_F) \parallel H_V^{\uparrow}$ und aus der Steuerbarkeit (M3) folgt $s\sigma \in \operatorname{pre} K$, woraus man $s\sigma \in (\operatorname{pre} L_F) \parallel R \parallel H_V$ schließt, d. h. (R2) ist erfüllt. Der verbleibende Fall ist durch $\gamma_H \subseteq \Sigma_{UCON,V}$ charakterisiert, d. h. der Nominalregler wartet auf ein virtuelles nicht-steuerbares Ereignis.

Fall 2b: $\gamma_H \cap \Sigma_{UCON,V} \neq \emptyset$. Man beziehe sich auf Gl. (2.4) und wegen $H_V \subseteq H_V^{\uparrow}$ folgt $s \in \operatorname{pre} K$. Aus (M6) erhält man $s \in p_{P,V}^{-1} \operatorname{pre} L_V$ und deshalb $p_{P,V} s \in \operatorname{pre} L_V$. Es löst H_V das Standardentwurfsproblem (Σ_V, L_V, H_V), sodass aus der Vollständigkeit von $L_V \parallel H_V$ (SR2) die Existenz eines σ mit $p_V(s\sigma) \in (\operatorname{pre} L_V) \parallel H_V$ folgt. Insbesondere muss $\sigma \in \Sigma_{UCON,V}$ gelten, d. h. der Regler wartet auf ein nicht-steuerbares Ereignis. Man bemerke, dass schwache Sensorkonsistenz (M5) impliziert, dass ein $t \in (\Sigma - \Sigma_{C,V})^*$ und ein $\sigma' \in \Sigma_{UCON,V}$ mit $st\sigma' \in \operatorname{pre} K$ existiert. Man beziehe sich auf Gl. (2.4) und schließe $(p_{P,F}\, st\sigma') \in \operatorname{pre} L_F$ sowie $(p_R\, st\sigma') \in R$. Dann folgt aus $\gamma_{LR} \subseteq \Sigma_{CON,V} \dot{\cup} \Sigma_{UCON,V}$ sofort $t = \varepsilon$ und man erhält $\sigma' \in \gamma_{LR}$. Es ist $p_V s \in (\operatorname{pre} L_V) \parallel H_V$ und $L_V \parallel H_V$ ist steuerbar bzgl. $(L_V, \Sigma_{UCON,V})$, und man erhält $p_V s\sigma \in (\operatorname{pre} L_V) \parallel H_V$. Insbesondere ist $s\sigma \in p_{C,V}^{-1} H_V$. Das schließt den Beweis der Vollständigkeit (R2) ab. q.e.d.

Spezifikationseinschluss. Man bemerke, dass für einen beliebigen virtualisierten Nominalregler H_V auch $H_V \subseteq H_V^{\uparrow}$ gilt, woraus dann $L_F \parallel R \parallel H_V \subseteq L_F \parallel R \parallel H_V^{\uparrow}$ folgt. Zur Sicherung des Spezifikationseinschlusses (R4) ist deshalb

(M7) Spezifikationseinschluss, d. h. $K \subseteq E$

ausreichend.

Lemma 2.1.3. Gegeben sei ein Rekonfigurationsproblem $(\Sigma, L_N, E_N, L_F, E_F, E_R)$, die virtualisierte minimal-restriktive Lösung $H_V^{\uparrow} \subseteq \Sigma_{C,V}^*$ des Nominalentwurfsproblems (Σ_N, L_N, E_N) und eine Sprache $K \subseteq \Sigma^*$ mit den Eigenschaften (M2), (M3) und (M7). Für eine beliebige virtuelle Lösung $H_V \subseteq \Sigma_{C,V}^*$ des Nominalentwurfsproblems (Σ_N, L_N, E_N)

sichert der Rekonfigurator $R := \mathrm{p}_{\mathrm{R}}\,\mathrm{pre}\,K$ den Spezifikationseinschluss (R4) des rekonfigurierenden Regelkreises $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}$. □

Beweis. Man wähle eine beliebige virtualisierte Lösung H_{V} des Nominalentwurfsproblems $(\Sigma_{\mathrm{N}}, L_{\mathrm{N}}, E_{\mathrm{N}})$. Es ist $H_{\mathrm{V}}^{\uparrow}$ minimal-restriktiv, sodass aus Gl. (2.4) sofort $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}} \subseteq L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}^{\uparrow} = K \subseteq E$ folgt, was den Nachweis des Spezifikationseinschlusses (R4) abschließt. q.e.d.

Weist also ein Kandidat K die Eigenschaften (M1) bis (M7) auf, dann genügt der rekonfigurierende Regelkreis $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}$ mit dem Rekonfigurator R gemäß Gl. 2.2 den Anforderung (R2) bis (R4) und zwar universell für alle virtualisierten Nominalregler H_{V}. Nimmt man zusätzlich das fehlerverträgliche Streckenverhalten L_{F} als abgeschlossen an, so ist (R1) trivial erfüllt. Der nachfolgende Satz fasst die Ergebnisse dieses Abschnitts zusammen.

Satz 2.1.1. Gegeben sei ein Rekonfigurationsproblem $(\Sigma, L_{\mathrm{N}}, E_{\mathrm{N}}, L_{\mathrm{F}}, E_{\mathrm{F}}, E_{\mathrm{R}})$ und die virtuelle minimal-restriktive Lösung $H_{\mathrm{V}}^{\uparrow} \subseteq \Sigma_{\mathrm{C,V}}^{*}$ des Nominalentwurfsproblems $(\Sigma_{\mathrm{N}}, L_{\mathrm{N}}, E_{\mathrm{N}})$. Betrachtet wird eine Sprache $K \subseteq \Sigma^{*}$ mit den Eigenschaften (M1) bis (M7). Falls die fehlerverträgliche Strecke L_{F} unter Vorsilbenbildung abgeschlossen ist, in Zeichen $\mathrm{pre}\,L_{\mathrm{F}} = L_{\mathrm{F}}$, dann ist der Rekonfigurator $R := \mathrm{p}_{\mathrm{R}}\,\mathrm{pre}\,K$ eine universelle Lösung des gegebenen Rekonfigurationsproblems. □

Beweis.

Ad [R0]: (R0) folgt direkt aus der Definition von R.

Ad [R1]: (R1) ist für abgeschlossene Sprachen trivial erfüllt.

Ad [R2]: (R2) folgt direkt aus Lemma 2.1.2.

Ad [R3]: (R3) folgt direkt aus Lemma 2.1.1.

Ad [R4]: (R4) folgt direkt aus Lemma 2.1.3. q.e.d.

Im folgenden Abschnitt wird die Verifikation konfliktfreier rekonfigurierender Regelkreise diskutiert und damit der Entwurf universeller Rekonfiguratoren abgeschlossen. Dementsprechend findet sich die Fortsetzung von Beispiel 1.4.10 am Ende des nächsten Abschnitts.

2.2 Reglerentwurf für nicht-triviale Letztendlichkeitseigenschaften

Im obigen Abschnitt 2.1 wurden unter der Voraussetzung trivialer Letztendlichkeitseigenschaften hinreichende Kriterien für den Entwurf universeller Rekonfiguratoren erarbeitet. Weist die fehlerverträgliche Strecke allerdings nicht-triviale Letztendlichkeitseigenschaften auf, sind für einen konfliktfreien rekonfigurierenden Regelkreis (R4) zusätzliche Anforderungen an einen Kandidaten K zu stellen.

Zur Verifikation von Konfliktfreiheit gehe man von einem gegebenen Rekonfigurator R aus. Man erinnere, dass zu einem Kandidaten $K \subseteq \Sigma^*$ mit (M1) und (M2) ein Rekonfigurator $R := \mathrm{p}_{\mathrm{R}}\,\mathrm{pre}\,K$ den rekonfigurierenden Regelkreis $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}^{\uparrow}$ induziert. Interpretiert man $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}^{\uparrow}$ als bekannte Strecke und den virtualisierten Regler H_{V} als Regler, ergibt sich der Regelkreis $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}^{\uparrow} \parallel H_{\mathrm{V}}$, dargestellt in Abbildung 2.2.1.

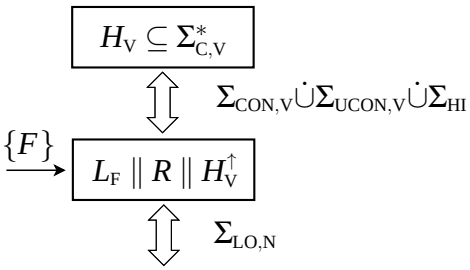


Abbildung 2.2.1: Rekonfigurierender Regelkreis - Abstraktionsbasierte Sicht

Während also die Strecke $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}^{\uparrow}$ exakt bekannt ist, ist H_{V} lediglich als Lösung des Standardentwurfsproblems $(\Sigma_{\mathrm{V}}, L_{\mathrm{V}}, E_{\mathrm{V}})$, d. h. ohne konkretes Modell, gegeben. Es bleibt dann zu klären, ob der bereits entworfene Rekonfigurator R für ein fehlerverträgliches Streckenverhalten L_{F} und eine beliebige virtualisierte Lösung H_{V} des Nominalentwurfsproblems $(\Sigma_{\mathrm{N}}, L_{\mathrm{N}}, E_{\mathrm{N}})$ einen konfliktfreien geschlossenen Regelkreis $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}$ garantiert. Mit dem *abstraktionsbasierten Reglerentwurf*, siehe z. B. [TM], ist in der Literatur ein verwandtes Problem bekannt.

2.2.1 Fehlerverdeckende Steuerungsrekonfiguration und abstraktionsbasierter Reglerentwurf

Im Kontext des abstraktionsbasierten Reglerentwurfs gemäß [TM] wird das Gesamtalphabet Σ in steuerbare Ereignisse Σ_{C} und nicht-steuerbare Ereignisse Σ_{UC} partitioniert,

d. h. $\Sigma = \Sigma_C \dot\cup \Sigma_{UC}$. Zu einem Streckenverhalten $L \subseteq \Sigma^*$ wird anhand der Projektion $p_o : \Sigma^* \to \Sigma_o^*$, $\Sigma_o \subseteq \Sigma_o$ eine *Streckenabstraktion* $L_o := p_o L$ berechnet. Für die Streckenabstraktion wird schließlich ein, unter Vorsilbenbildung abgeschlossener, Regler $H_o \subseteq \Sigma_o^*$, entworfen, sodass der *abstrahierte Regelkreis* $L_o \parallel H_o$ die Eigenschaften

(O1) Konfliktfreiheit, d. h. $(\operatorname{pre} L_o) \parallel H_o = \operatorname{pre}(L_o \parallel H_o)$

(O2) Vollständigkeit, d. h. $(\forall s \in (\operatorname{pre} L_o) \parallel H_o \exists \sigma \in \Sigma_o)[s\sigma \in (\operatorname{pre} L_o) \parallel H_o]$

(O3) Steuerbarkeit bzgl. (L_o, Σ_{UC}), d. h. $((\operatorname{pre} L_o) \parallel H_o)\Sigma_{UC} \cap \operatorname{pre} L_o \subseteq (\operatorname{pre} L) \parallel H_o$

aufweist. Der Entwurf eines Reglers H_o ist ein bereits bekanntes und gelöstes Problem, ähnlich dem Standardentwurfsproblem in Abschnitt 1.2.

Betrachtet wird dann der *abstraktionsbasierte Regelkreis* $L \parallel H_o$, dargestellt in Abb. 2.2.2.

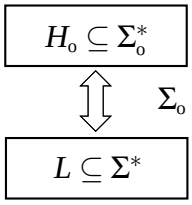


Abbildung 2.2.2: Abstraktionsbasierter Regelkreis

Gesucht werden Bedingungen für L, sodass ein *beliebiger*, aber abgeschlossener, abstraktionsbasierter Regler H_o mit (O1) bis (O3) auch für die tatsächliche Strecke L die Eigenschaften

(P1) Konfliktfreiheit, d. h. $(\operatorname{pre} L) \parallel H_o = \operatorname{pre}(L \parallel H_o)$

(P2) Vollständigkeit, d. h. $(\forall s \in (\operatorname{pre} L) \parallel H_o \exists \sigma \in \Sigma)[s\sigma \in (\operatorname{pre} L) \parallel H_o]$

(P3) Steuerbarkeit bzgl. (L, Σ_{UC}), d. h. $((\operatorname{pre} L) \parallel H_o)\Sigma_{UC} \cap (\operatorname{pre} L) \subseteq (\operatorname{pre} L) \parallel H_o$,

sichert.

Im Kontext der fehlerverdeckenden Steuerungsrekonfiguration ist der virtualisierte Nominalregler H_V eine Lösung des Standardentwurfsproblems (Σ_V, L_V, E_V) und sichert deshalb für L_V konfliktfreies, vollständiges und steuerbares Regelkreisverhalten. Andererseits kann $L_F \parallel R \parallel H_V^{\uparrow}$ als Strecke aufgefasst werden. Analogien zur Steuerungsrekonfiguration zeigen sich also für $L := L_F \parallel R \parallel H_V$, $\Sigma_o := \Sigma_{CON,V} \dot\cup \Sigma_{UCON,V}$ und $H_o := H_V$. Allerdings entspricht die Projektion von $L_F \parallel R \parallel H_V^{\uparrow}$ nicht dem virtualisierten Nominalstreckenverhalten L_V, d. h. $p_{P,V}(L_F \parallel R \parallel H_V^{\uparrow}) \neq L_V$. Stattdessen gilt lediglich $p_{P,V}((\operatorname{pre} L_F) \parallel R \parallel H_V^{\uparrow}) \subseteq \operatorname{pre} L_V$. Das fehlerverdeckende Rekonfigurationsproblem ist deshalb kein Problem des abstraktionsbasierten Reglerentwurfs. Nichtsdestotrotz sind (P1) und (R1) für $L := L_F \parallel R$

und $H_o := H_V$ strukturell identisch. Es ist also zu vermuten, dass sich Ergebnisse aus dem abstraktionsbasierten Reglerentwurf auf die Steuerungsrekonfiguration übertragen lassen.

In der Literatur wird im Rahmen der hierarchischen Regelung ereignisdiskreter Systeme der Begriff des *natürlichen Beobachters* vorgestellt. Man ziehe [WW96] für eine allgemeine Diskussion des Beobachterkonzepts und [TM] für eine knappe Darstellung im Rahmen des abstraktionsbasierten Reglerentwurfs heran. In [TM] wurde gezeigt, dass die Verifikation von Konfliktfreiheit auch auf Basis einer *Rückwärtserreichbarkeitsanalyse* in Termen spezieller Erreichbarkeitsoperatoren möglich ist. Die Rückwärtserreichbarkeitsanalyse erweist sich dabei als weniger restriktiv als der natürliche Beobachter.

Nachfolgend wird zunächst eine für die Steuerungsrekonfiguration angepasste Version des natürlichen Beobachters, der *virtuelle Beobachter*, vorgestellt. Auf dieser Basis kann bereits die Konfliktfreiheit des rekonfigurierenden Regelkreises im Sinne von (R1) nachgewiesen werden. Anschließend wird, [TM] folgend, die Verifikation von Konfliktfreiheit auf Basis einer geeigneten Rückwärtserreichbarkeitsanalyse untersucht. Dazu werden mit *universellen Rückwärtserreichbarkeitsoperatoren* alle Zeichenketten erfasst, die durch ein einzelnes Zeichen oder durch eine geeignete Zeichenkette, unabhängig von H_V, zu einer gültigen Zeichenkette erweitert werden können. Falls der Vorsilbenabschluss eines Regelkreiskandidaten $\operatorname{pre} K$, mit den Eigenschaften (M1) bis (M7), ein Fixpunkt eines universellen Rückwärtserreichbarkeitsoperators ist, dann ist der Rekonfigurator $R := \mathrm{p}_R \operatorname{pre} K$ eine universelle Lösung des Rekonfigurationsproblems.

2.2.2 Virtueller Beobachter

In dem oben aufgespannten Rahmen zur abstraktionsbasierten Regelung heißt eine Projektion $\mathrm{p}_o : \Sigma^* \to \Sigma_o^*$ ein natürlicher Beobachter für L, falls

$$(\forall s \in \operatorname{pre} L, u \in \Sigma_o^*)[\mathrm{p}_o su \in \mathrm{p}_o L \quad \Rightarrow \quad (\exists t \in \mathrm{p}_o^{-1} u)[st \in L]] \tag{2.6}$$

gilt.

Vor dem Hintergrund der Konfliktfreiheit des abstraktionsbasierten Regelkreises $L \parallel H_o$, betrachte man ein beliebiges $s \in (\operatorname{pre} L) \parallel H_o$. Für Konfliktfreiheit ist dann ein $t \in \Sigma^*$ zu finden, sodass $st \in L \parallel H_o$ gilt. Dazu bemerke man, dass $\mathrm{p}_o s \in (\operatorname{pre} L_o) \parallel H_o$ gilt. Wegen (O1) existiert eine Fortsetzung $u \in \Sigma_o^*$ zu einer im abstrahierten Regelkreis gültigen Zeichenkette $\mathrm{p}_o su \in L_o \parallel H_o$. Ist die Projektion p_o ein natürlicher Beobachter für L, so existiert für alle Fortsetzungen von $\mathrm{p}_o s$ zu einer in $L_o \parallel H_o$ gültigen Zeichenkette, auch eine Fortsetzung $t \in \Sigma^*$ von s zu einer in dem abstraktionsbasierten Regelkreis gültigen Zeichenkette

$st \in L \parallel H_o$, die unter Projektion u gleicht. Effektiv folgt also aus der Konfliktfreiheit des abstrahierten Regelkreises auch die Konfliktfreiheit von tatsächlicher Strecke L und abstraktionsbasierten Regler H_o.

Analog lässt sich im Kontext der Steuerungsrekonfiguration aus der Konfliktfreiheit des virtuellen minimal-restriktiven Regelkreises $L_V \parallel H_V^{\uparrow}$ die Konfliktfreiheit des rekonfigurierenden Regelkreises $L_F \parallel R \parallel H_V$ für ein beliebiges H_V ableiten. Das Analogon des natürlichen Beobachters im Kontext der fehlerverdeckenden Steuerungsrekonfiguration wird als *virtueller* Beobachter bezeichnet.

Definition 2.2.1. Gegeben sei ein Rekonfigurationsproblem $(\Sigma, L_N, E_N, L_F, E_F, E_R)$ und die virtualisierte minimal-restriktive Lösung $H_V^{\uparrow} \subseteq \Sigma_{C,V}^*$ des Nominalreglerentwurfsproblems (Σ_N, L_N, E_N). Zu einer Sprache $K \subseteq \Sigma^*$ heißt die Projektion $p_V : \Sigma^* \to \Sigma_V^*$ *virtueller Beobachter* bzgl. (K, Σ_V), falls

$$(\forall s \in \mathrm{pre}\, K, u \in \Sigma_V^*)[p_V su \in L_V \parallel H_V^{\uparrow} \quad \Rightarrow \quad (\exists t \in p_V^{-1} u)[st \in K]]$$

gilt. □

Tatsächlich kann nun für einen Regelkreiskandidaten mit den Eigenschaften (M2), (M3) und (M6) unter der zusätzlichen Voraussetzung

(M8.1) die Projektion $p_V : \Sigma^* \to \Sigma_V^*$ ist ein virtueller Beobachter bzgl. (K, Σ_V),

die Konfliktfreiheit des rekonfigurierenden Regelkreises im Sinne von (R1) nachgewiesen werden.

Proposition 2.2.1. Gegeben sei ein Rekonfigurationsproblem $(\Sigma, L_N, E_N, L_F, E_F, E_R)$ und die virtualisierte minimal-restriktive Lösung $H_V^{\uparrow} \subseteq \Sigma_{C,V}^*$ des Nominalentwurfsproblems (Σ_N, L_N, E_N), zusammen mit einer vorsilbennormalen (M2), relativ abgeschlossenen (M3) und von der virtuellen Strecke eingeschlossenen (M6) Sprache $K \subseteq \Sigma^*$. Ist die Projektion $p_V : \Sigma^* \to \Sigma_V^*$ ein virtueller Beobachter bzgl. (K, Σ_V) (M8.1), dann ist für eine beliebige virtualisierte Lösung $H_V \subseteq \Sigma_{C,V}^*$ des Nominalreglerentwurfsproblems (Σ_N, L_N, E_N) und für den Rekonfigurator $R := p_R \,\mathrm{pre}\, K$ der rekonfigurierenden Regelkreis $L_F \parallel R \parallel H_V$ konfliktfrei (R1). □

Beweis. Für eine beliebige virtualisierte Lösung H_V und ein beliebiges $s \in (\mathrm{pre}\, L_F) \parallel R \parallel H_V$ ist die Existenz eines $t \in \Sigma^*$ mit $st \in L_F \parallel R \parallel H_V$ zu zeigen. Es ist $H_V^{\uparrow}$ minimal-restriktiv, sodass $s \in (\mathrm{pre}\, L_F) \parallel R \parallel H_V \subseteq (\mathrm{pre}\, L_F) \parallel R \parallel H_V^{\uparrow}$ gilt. Zudem gilt gemäß Gl. (2.4): $\mathrm{pre}\, K = (\mathrm{pre}\, L_F) \parallel R \parallel H_V^{\uparrow}$, woraus sofort $s \in \mathrm{pre}\, K$ folgt. Die Voraussetzung (M6) impliziert dann

$s \in \mathrm{p}_\mathrm{V}^{-1}\,\mathrm{pre}\,L_\mathrm{V}$ und gemäß Proposition A.2.1 hat man $\mathrm{p}_\mathrm{V}\,s \in \mathrm{p}_\mathrm{V}\,\mathrm{p}_{\mathrm{P,V}}^{-1}\,L_\mathrm{V} = \mathrm{p}_{\mathrm{P,V}}^{-V}\,L_\mathrm{V}$[1]. Zudem gilt $s \in \mathrm{pre}\,K \subseteq \mathrm{p}_{\mathrm{C,V}}^{-1}\,H_\mathrm{V}^{\uparrow}$ und gemäß Prop. A.2.1 auch $\mathrm{p}_\mathrm{V}\,s \in \mathrm{p}_\mathrm{V}\,\mathrm{p}_{\mathrm{C,V}}^{-1}\,H_\mathrm{V} = \mathrm{p}_{\mathrm{C,V}}^{-V}\,H_\mathrm{V}$. Aus $s \in \mathrm{p}_{\mathrm{P,V}}^{-V}\,L_\mathrm{V}$ und $s \in \mathrm{p}_{\mathrm{C,V}}^{-V}\,H_\mathrm{V}$ schließe man auf $s \in L_\mathrm{V} \parallel H_\mathrm{V}^{\uparrow}$. Es löst H_V das Standardentwurfsproblem $(\Sigma_\mathrm{V}, L_\mathrm{V}, E_\mathrm{V})$, sodass der virtuelle Regler H_V und die virtuelle Strecke L_V konfliktfrei sind. Folglich existiert ein $u \in \Sigma_\mathrm{V}^*$ mit $\mathrm{p}_\mathrm{V}\,su \in L_\mathrm{V} \parallel H_\mathrm{V} \subseteq L_\mathrm{V} \parallel H_\mathrm{V}^{\uparrow}$. Aus Eigenschaft (M8.1) folgere man dann die Existenz eines $t \in \mathrm{p}_\mathrm{V}^{-1}\,u$ mit $st \in K$. Man bemerke, dass $(\mathrm{p}_\mathrm{V}\,st) \in L_\mathrm{V} \parallel H_\mathrm{V}$ gilt und deshalb gilt auch $st \in \mathrm{p}_\mathrm{V}^{-1}\,(L_\mathrm{V} \parallel H_\mathrm{V})$. Aus $st \in \mathrm{p}_\mathrm{V}^{-1}\,(L_\mathrm{V} \parallel H_\mathrm{V}) \subseteq \mathrm{p}_\mathrm{V}^{-1}\,H_\mathrm{V}$ und $st \in K = L_\mathrm{F} \parallel R \parallel H_\mathrm{V}^{\uparrow}$, siehe Gl. (2.5), erhält man schließlich $st \in L_\mathrm{F} \parallel R \parallel H_\mathrm{V}$. q.e.d.

Zieht man zusätzlich die Ergebnisse für triviale Letztendlichkeitseigenschaften aus dem letzten Abschnitt 2.1 heran, dann kann aus dem Regelkreiskandidaten K mit (M1) bis (M8.1) eine universelle Lösung R des fehlerverdeckenden Rekonfigurationsproblems gewonnen werden. Der nachfolgende Satz ist eine direkte Konsequenz der Lemmata 2.1.1 bis 2.1.3 und 2.2.1.

Satz 2.2.1. Gegeben sei ein Rekonfigurationsproblem $(\Sigma, L_\mathrm{N}, E_\mathrm{N}, L_\mathrm{F}, E_\mathrm{F}, E_\mathrm{R})$ und die virtualisierte minimal-restriktive Lösung $H_\mathrm{V}^{\uparrow} \subseteq \Sigma_{\mathrm{C,V}}^*$ des Nominalentwurfsproblems $(\Sigma_\mathrm{N}, L_\mathrm{N}, E_\mathrm{N})$. Falls eine Sprache $K \subseteq \Sigma^*$ die Eigenschaften (M1) bis (M8.1) aufweist, dann ist der Rekonfigurator $R := \mathrm{p}_\mathrm{R}\,\mathrm{pre}\,K$ eine universelle Lösung des gegebenen Rekonfigurationsproblems. □

2.2.3 Rückwärtserreichbarkeitsanalyse

Definitionsgemäß fordert Konfliktfreiheit, dass von einer beliebigen Vorsilbe des rekonfigurierenden Kreises $s \in (\mathrm{pre}\,L_\mathrm{F}) \parallel R \parallel H_\mathrm{V}$ die Möglichkeit zur Einlösung einer Letztendlichkeitseigenschaften besteht. Mit anderen Worten: es muss eine Zeichenkette $t \in \Sigma^*$ mit $st \in (\mathrm{pre}\,L_\mathrm{F}) \parallel R \parallel H_\mathrm{V}$ existieren. In diesem Abschnitt soll eine solche Zeichenkette t mit Hilfe einer geeigneten Rückwärtserreichbarkeitsanalyse konstruiert werden. Ausgehend von der *Zielsprache* $L_\mathrm{F} \parallel R \parallel H_\mathrm{V}^{\uparrow}$ werden alle Zeichenketten $s \in (\mathrm{pre}\,L_\mathrm{F} \parallel R \parallel H_\mathrm{V}^{\uparrow},)$ identifiziert, für die eine Fortsetzung t mit $st \in L_\mathrm{F} \parallel R \parallel H_\mathrm{V}^{\uparrow}$ existiert, die von keinem virtuellen Nominalregler H_V unterdrückt werden kann. Wird diese Menge mit $(\mathrm{pre}\,L_\mathrm{F}) \parallel R \parallel H_\mathrm{V}$ gefunden, dann ist der rekonfigurierende Regelkreis im Sinne von (R1) konfliktfrei. Wie bereits im vorhergehenden Abschnitt wird dabei wegen einer leichteren Ergebnisintegration eine zusätzliche Anforderung an einen Regelkreiskandidaten K angestrebt.

[1] Für $\Sigma_\mathrm{A} \subseteq \Sigma_\mathrm{B} \subseteq \Sigma$ wird die inverse Projektion $\mathrm{p}_\mathrm{A}^{-B} := \{s \in \Sigma_\mathrm{B}^* \mid \mathrm{p}_\mathrm{o}\,s = u\}$ betrachtet.

Mit Blick auf eine spätere iterative Rückwärtserreichbarkeitsanalyse werden nachfolgend die entsprechenden Rückwärtserreichbarkeitsoperatoren anhand einer allgemeinen Zielsprache $M \subseteq \operatorname{pre} K$ eingeführt. Tatsächlich behält man dabei aber die Strecke $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}^{\uparrow}$ im Auge.

Definition 2.2.2. Gegeben sei ein Rekonfigurationsproblem $(\Sigma, L_{\mathrm{N}}, E_{\mathrm{N}}, L_{\mathrm{F}}, E_{\mathrm{F}}, E_{\mathrm{R}})$, die virtualisierte minimal-restriktive Lösung $H_{\mathrm{V}}^{\uparrow} \subseteq \Sigma_{\mathrm{C,V}}^{*}$ des Nominalentwurfsproblems $(\Sigma_{\mathrm{N}}, L_{\mathrm{N}}, E_{\mathrm{N}})$ und eine Sprache $K \subseteq \Sigma^{*}$. Ein Operator Ω_{*} über $\operatorname{pre} K$ heißt *universeller Rückwärtserreichbarkeitsoperator*, falls für eine beliebige Zielsprache $M \subseteq \operatorname{pre} K$ und eine beliebige virtualisierte Lösung $H_{\mathrm{V}} \subseteq \Sigma_{\mathrm{C,V}}^{*}$ des Nominalentwurfsproblems $(\Sigma_{\mathrm{N}}, L_{\mathrm{N}}, E_{\mathrm{N}})$

$$(\forall s \in (\Omega_{*} M) \parallel H_{\mathrm{V}}\, \exists t \in \Sigma^{*})[st \in M \parallel H_{\mathrm{V}}] \tag{2.7}$$

gilt. □

Erfüllt ein Regelkreiskandidat $K = L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}^{\uparrow}$ für einen universellen Rückwärtserreichbarkeitsoperation Ω_{*} die Eigenschaft

(M8.2) $\operatorname{pre} K = \Omega_{*} K$,

dann existiert definitionsgemäß, siehe Gl. 2.7 mit $M = K = L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}$, für einen beliebigen virtualisierten Nominalregler H_{V} und ein beliebiges $s \in \operatorname{pre} K = (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}$ tatsächlich eine Fortsetzung $t \in \Sigma^{*}$ mit $st \in L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}^{\uparrow} \parallel H_{\mathrm{V}} = L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}$. Entsprechend ist der rekonfigurierende Regelkreis im Sinne von (R1) konfliktfrei.

Proposition 2.2.2. Gegeben sei ein Rekonfigurationsproblem $(\Sigma, L_{\mathrm{N}}, E_{\mathrm{N}}, L_{\mathrm{F}}, E_{\mathrm{F}}, E_{\mathrm{R}})$ und die virtualisierte minimal-restriktive Lösung $H_{\mathrm{V}}^{\uparrow} \subseteq \Sigma_{\mathrm{C,V}}^{*}$ des Nominalentwurfsproblems $(\Sigma_{\mathrm{N}}, L_{\mathrm{N}}, E_{\mathrm{N}})$. Falls eine Sprache $K \subseteq \Sigma^{*}$ die Eigenschaft (M8.2) erfüllt, dann ist für eine beliebige virtualisierte Lösung $H_{\mathrm{V}} \subseteq \Sigma_{\mathrm{C,V}}^{*}$ des Nominalentwurfsproblems $(\Sigma_{\mathrm{N}}, L_{\mathrm{N}}, E_{\mathrm{N}})$ und den Rekonfigurator $R := \mathrm{p}_{\mathrm{R}} \operatorname{pre} K$ der rekonfigurierende Regelkreis $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}$ konfliktfrei (R1). □

Beweis. Man wähle eine beliebige virtualisierte Lösung H_{V} des Nominalentwurfproblems $(\Sigma_{\mathrm{N}}, L_{\mathrm{N}}, E_{\mathrm{N}})$ zusammen mit einem beliebigen $s \in (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}$. Es ist $H_{\mathrm{V}}^{\uparrow}$ minimal-restriktiv, sodass $(\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}} \subseteq (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}^{\uparrow}$ gilt. Gemäß Gl. (2.4) ist $\operatorname{pre} K = (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}^{\uparrow}$, sodass daraus auch $s \in \operatorname{pre} K$ folgt. Die Voraussetzung (M8.2) fordert, dass $\operatorname{pre} K = \Omega_{*} K$ gilt, weshalb aus $s \in \operatorname{pre} K$ auch $s \in \Omega_{*} K$ folgt. Man bemerke, dass $s \in \Omega_{*} K$ und $s \in (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}} \subseteq \mathrm{p}_{\mathrm{C,V}}^{-1} H_{\mathrm{V}}$ zusammen $s \in (\Omega_{*} M) \parallel H_{\mathrm{V}}$ implizieren. Folglich existiert gemäß der Definition des Operators Ω_{*}, siehe Gl. 2.7, ein $t \in \Sigma^{*}$ mit

$st \in K \parallel H_V$. Wegen Gl. (2.5) ist $K = L_F \parallel R \parallel H_V^{\uparrow}$, sodass aus $st \in K \parallel H_V$ auch $st \in L_F \parallel R \parallel H_V$ folgt. q.e.d.

Bisher wurden Rückwärtserreichbarkeitsoperatoren lediglich abstrakt definiert. Nachfolgend sollen konkrete universelle Rückwärtserreichbarkeitsoperatoren angegeben werden.

Bezüglich der Rückwärtserreichbarkeitsoperatoren werden mit Streckenfehlerereignissen $\Sigma_{P,F}$ und $\Sigma_{UCON,V}$ Ereignisse identifiziert, deren Auftreten ein virtualisierter Nominalregler H_V nicht verhindern kann. Erstere, weil ihr Auftreten für H_V nicht sichtbar ist, letztere, weil H_V als Lösung des Standardentwurfsproblems (Σ_V, L_V, E_V) ein bzgl. $(L_V, \Sigma_{UCON,V})$ steuerbares Regelkreisverhalten erzeugt. Daneben wird die Vollständigkeit des rekonfigurierenden Regelkreises (R2) ausgenützt. Diesbezüglich werden Ereignisketten als zulässig erkannt, falls alle möglichen Erweiterungen in die Zielsprache M führen.

Proposition 2.2.3. Gegeben sei ein Rekonfigurationsproblem $(\Sigma, L_N, E_N, L_F, E_F, E_R)$, die virtualisierte minimal-restriktive Lösung $H_V^{\uparrow}$ des Nominalentwurfproblems (Σ_N, L_N, E_N) und eine Sprache $K \subseteq \Sigma^*$ mit den Eigenschaften (M1) bis (M6). Zu einer beliebigen Zielsprache $M \subseteq \mathrm{pre}K$ betrachte man die Operatoren Ω_{ID}, $\Omega_{1,A}$, $\Omega_{1,B}$ und $\Omega_{1,C}$ über $\mathrm{pre}K$, gegeben durch

$$\Omega_{ID} M := M$$
$$\Omega_{1,A} M := \{s \in \mathrm{pre}K \mid (\exists \sigma \in \Sigma_{P,F})[s\sigma \in M]\},$$
$$\Omega_{1,B} M := \{s \in \mathrm{pre}K \mid (\exists \sigma \in \Sigma_{UCON,V})[s\sigma \in M]\} \text{ und}$$
$$\Omega_{1,C} M := \{s \in \mathrm{pre}K \mid (\forall \sigma \in \Sigma)[s\sigma \in \mathrm{pre}K \Rightarrow s\sigma \in M]\}.$$

Die Operatoren Ω_{ID}, $\Omega_{1,A}$, $\Omega_{1,B}$ und $\Omega_{1,C}$ sind universelle Rückwärtserreichbarkeitsoperatoren. □

Beweis. Man wähle eine beliebige virtualisierte Lösung $H_V \subseteq \Sigma_{C,V}^*$ des Nominalentwurfproblems (Σ_N, L_N, E_N) zusammen mit einer beliebigen Zielsprache $M \subseteq \mathrm{pre}K$.

Zu Ω_{ID}: Zu einem beliebigen $s \in (\Omega_{ID} M) \parallel H_V$ wird $t = \varepsilon$ gesetzt und man erhält $st = s \in (\Omega_{ID} M) \parallel H_V = M \parallel H_V$.

Zu $\Omega_{1,A}$: Zu einem beliebigen $s \in (\Omega_{1,A} M) \parallel H_V$ existiert, gemäß der Definition des Operators $\Omega_{1,A}$, ein $\sigma \in \Sigma_{P,F}$ mit $s\sigma \in M$. Die Streckenfehlerereignisse $\Sigma_{P,F}$ und die virtuellen Reglerereignisse $\Sigma_{C,V}$ sind disjunkte Alphabete, in Zeichen $\Sigma_{P,F} \cap \Sigma_{C,V} = \emptyset$. Entsprechend sind die fehlerverträgliche Strecke L_F und der virtuelle Nominalregler H_V entkoppelte Systeme. Deshalb erhält man aus $s \in (\Omega_{1,A} M) \parallel H_V \subseteq \mathrm{p}_{C,V}^{-1} H_V$ auch $s\sigma \in \mathrm{p}_{C,V}^{-1} H_V$ und zu-

sammen mit $s\sigma \in M$ folgt letztlich, dass $s\sigma \in M \parallel H_V$ gilt. Es ist also $\Omega_{1,A}$ ein universeller Rückwärtserreichbarkeitsoperator.

Ad $\Omega_{1,B}$*:* Zu einem beliebigen $s \in (\Omega_{1,B} M) \parallel H_V$ existiert gemäß der Definition des Operators $\Omega_{1,B}$ ein $\sigma \in \Sigma_{UCON,V}$ mit $s\sigma \in M$. Aus (M6) schließe man $s\sigma \in M \subseteq \mathrm{pre}\, K \subseteq \mathrm{p}_{P,V}^{-1} \mathrm{pre}\, L_V$. Folglich ist $\mathrm{p}_V s\sigma \in \mathrm{p}_V \mathrm{p}_{P,V}^{-1} L_V = \mathrm{p}_{P,V}^{-V} L_V$. Weiter gilt wegen $M \subseteq \mathrm{pre}\, K \subseteq (\mathrm{pre}\, L_F) \parallel H_V^{\uparrow}$ auch $\mathrm{p}_V s \in \mathrm{p}_V \mathrm{p}_{C,V}^{-1} H_V \subseteq \mathrm{p}_{C,V}^{-V} H_V^{\uparrow}$, sodass man $s \in L_V \parallel H_V$ erhält. Da H_V das Entwurfsproblem (Σ_V, L_V, H_V) löst, ist $L_V \parallel H_V$ steuerbar bzgl. $(L_V, \Sigma_{UCON,V} \dot{\cup} \Sigma_{LO})$. Folglich ist $s\sigma \in L_V \parallel H_V$. Aus $s\sigma \in M$ und $s\sigma \in L_V \parallel H_V$ ergibt sich $s\sigma \in M \parallel H_V$. Also ist $\Omega_{1,B}$ ein universeller Rückwärtserreichbarkeitsoperator.

Ad $\Omega_{1,C}$*:* Man wähle ein beliebiges $s \in (\Omega_{1,C} M) \parallel H_V$. Es erfüllt K die Eigenschaften (M1) bis (M6), sodass gemäß Lemma 2.1.2 der rekonfigurierende Regelkreis $L_F \parallel R \parallel H_V$ vollständig ist. Folglich existiert ein $\sigma \in \Sigma$ mit $s\sigma \in (\mathrm{pre}\, L_F) \parallel R \parallel H_V$. Damit gilt auch: $s\sigma \subseteq \mathrm{p}_{C,V}^{-1} H_V$. Daneben ist $H_V^{\uparrow}$ minimal-restriktiv, sodass $(\mathrm{pre}\, L_F) \parallel R \parallel H_V \subseteq (\mathrm{pre}\, L_F) \parallel R \parallel H_V^{\uparrow}$ gilt. Zudem gilt gemäß Gl. (2.4): $\mathrm{pre}\, K = (\mathrm{pre}\, L_F) \parallel R \parallel H_V^{\uparrow}$, weshalb aus $s\sigma \in (\mathrm{pre}\, L_F) \parallel R \parallel H_V$ auch $s\sigma \in \mathrm{pre}\, K$ folgt. Daraus und aus $s \in \Omega_{1,C} M$ folgere man, gemäß der Definition von $\Omega_{1,C}$, $s\sigma \in M$. Letztlich implizieren $s\sigma \subseteq \mathrm{p}_{C,V}^{-1} H_V$ und $s\sigma \in M$ zusammen $s\sigma \in M \parallel H_V$. Folglich ist $\Omega_{1,C}$ ein universeller Rückwärtserreichbarkeitsoperator. q.e.d.

Es bietet sich weiter an, die Anforderung des virtuellen Beobachters in einen Rückwärtserreichbarkeitsoperator zu überführen.

Proposition 2.2.4. Gegeben sei ein Rekonfigurationsproblem $(\Sigma, L_N, E_N, L_F, E_F, E_R)$, die virtualisierte minimal-restriktive Lösung $H_V^{\uparrow}$ des Nominalentwurfsproblems (Σ_N, L_N, E_N) und eine Sprache $K \subseteq \Sigma^*$. Zu einer beliebig gewählten Zielsprache $M \subseteq \mathrm{pre}\, K$ betrachte man den Operator

$$\Omega_{*,D} M := \{ s \in \mathrm{pre}\, K \mid (\forall u \in \Sigma_V^*)[\mathrm{p}_V su \in L_V \parallel H_V^{\uparrow} \Rightarrow (s \mathrm{p}_V^{-1} \mathrm{pre}\, u) \cap M \neq \emptyset] \}.$$

Der Operator $\Omega_{*,D}$ ist ein universeller Rückwärtserreichbarkeitsoperator. □

Beweis. Man wähle eine beliebige virtualisierte Lösung $H_V \subseteq \Sigma_{C,V}^*$ des Nominalentwurfsproblems (Σ_N, L_N, E_N) zusammen mit einer beliebigen Zielsprache $M \subseteq \mathrm{pre}\, K$ und einem beliebigen $s \in (\Omega_{*,D} M) \parallel H_V$. Es ist $\Omega_{*,D} M \subseteq \mathrm{pre}\, K$, sodass auch $s \in \mathrm{pre}\, K$ gilt. Aus (M6) folgere man $s \in \mathrm{p}_{P,V}^{-1} \mathrm{pre}\, L_V$ und aus Prop. A.2.1 folgt $\mathrm{p}_V s \in \mathrm{p}_{P,V}^{-V} \mathrm{pre}\, L_V$. Aus $s \in M \parallel H_V$ ergibt sich $s \in \mathrm{p}_{C,V}^{-1} H_V$ und, gemäß Prop. A.2.1, gilt auch $\mathrm{p}_V s \in \mathrm{p}_{C,V}^{-V} H_V$. Folglich hat man $s \in (\mathrm{pre}\, L_V) \parallel H_V$. Da H_V das Entwurfsproblem (Σ_V, L_V, H_V) löst, sind der virtuelle Regler H_V und die virtuelle Strecke L_V konfliktfrei. Folglich existiert ein $u \in \Sigma_V^*$

mit $\mathrm{p_V}\, su \in L_\mathrm{V} \parallel H_\mathrm{V}$. Nachdem $s \in \Omega_{*,\mathrm{D}} M$ gilt, kann man auf $(s\mathrm{p}_\mathrm{V}^{-1}\,\mathrm{pre}\,u) \cap M \neq \emptyset$ schließen. Folglich existiert ein $t \in \Sigma^*$ mit $\mathrm{p_V}\, st = \mathrm{p_V}\, su \in L_\mathrm{V} \parallel H_\mathrm{V}$ und $st \in M$. Letztlich ist $s \in M \parallel L_\mathrm{V} \parallel H_\mathrm{V} \subseteq M \parallel H_\mathrm{V}$. q.e.d.

Auch die Vereinigung universeller Rückwärtserreichbarkeitsoperatoren selbst ist wieder ein universeller Rückwärtserreichbarkeitsoperator, d. h.

$$\Omega := \Omega_{1,\mathrm{A}} \cup \Omega_{1,\mathrm{B}} \cup \Omega_{1,\mathrm{C}} \cup \Omega_{*,\mathrm{D}} \cup \Omega_{\mathrm{ID}} \tag{2.8}$$

ist selbst ein universeller Rückwärtserreichbarkeitsoperator.

Mit Blick auf die Verifikation der Konfliktfreiheit des rekonfigurierenden Regelkreises erscheint die Forderung, mit einer einzelnen Rückwärtserreichbarkeitsanalyse sämtliche Vorsilben eines Regelkreiskandidaten abzudecken, als restriktiv. Durch die Iteration universeller Rückwärtserreichbarkeitsoperatoren können die bisherigen Anforderungen gelockert werden. Man betrachte dazu

$$\begin{aligned} &M_0 := M, \\ &M_{\mathrm{i}+1} := \Omega M_\mathrm{i}, i \in \mathbb{N}_0. \end{aligned} \tag{2.9}$$

Proposition 2.2.5. Gegeben sei ein Rekonfigurationsproblem $(\Sigma, L_\mathrm{N}, E_\mathrm{N}, L_\mathrm{F}, E_\mathrm{F}, E_\mathrm{R})$, die virtualisierte minimal-restriktive Lösung $H_\mathrm{V}^{\uparrow}$ des Nominalentwurfsproblems $(\Sigma_\mathrm{N}, L_\mathrm{N}, E_\mathrm{N})$ und eine Sprache $K \subseteq \Sigma^*$. Zu einer beliebig gewählten Zielsprache $M \subseteq \mathrm{pre}\, K$, wähle man $n \in \mathbb{N}_0$ und die Iteration (2.9). Dann sind die Operatoren

$$\begin{aligned} &\Omega^\mathrm{n} M := M_\mathrm{n} \\ &\Omega_\infty M := \bigcup_{i \in \mathbb{N}} M_\mathrm{i}, \end{aligned}$$

universelle Rückwärtserreichbarkeitsoperatoren. □

Beweis. Man wähle eine beliebige virtualisierte Lösung $H_\mathrm{V} \subseteq \Sigma_{\mathrm{C,V}}^*$ des Nominalentwurfsproblems $(\Sigma_\mathrm{N}, L_\mathrm{N}, E_\mathrm{N})$ zusammen mit einer beliebigen Zielsprache $M \subseteq \mathrm{pre}\, K$. Zu zeigen ist für ein beliebiges $s \in (\Omega^\mathrm{n} M) \parallel H_\mathrm{V}$ die Existenz eines $t \in \Sigma^*$ mit $st \in M \parallel H_\mathrm{V}$. Für einen Induktionsbeweis betrachte man zunächst den Fall $n = 0$. Es gilt $\Omega^0 = M_0 = M$. Man wähle ein $s_0 \in (\Omega^0 M) \parallel H_\mathrm{V} = M \parallel H_\mathrm{V}$ und bemerke, dass mit $t_0 = \varepsilon \in \Sigma^*$ die Behauptung für $n = 0$, d. h. $s_0 t_0 \in M \parallel H_\mathrm{V}$ erfüllt ist. Man nehme nun an, für ein $i \in \mathbb{N}_0$ existiere für ein beliebiges $s_\mathrm{i} \in (\Omega^\mathrm{i} M) \parallel H_\mathrm{V}$ ein $t_\mathrm{i} \in \Sigma^*$ mit $s_\mathrm{i} t_\mathrm{i} \in M \parallel H_\mathrm{V}$. Für den Induktionsschluss

wähle man ein $s_{i+1} \in (\Omega^{i+1}M) \parallel H_V$. Falls $\Omega_*^{i+1}M = \Omega^i M$ gilt, dann folgt mit $t_{i+1} = \varepsilon$ sofort $s_{i+1}t_{i+1} \in \Omega^i M$. Andernfalls bemerke man, dass $\Omega^{i+1}M = M_{i+1} = \Omega M_i$ gilt. Gemäß der Definition universeller Rückwärtserreichbarkeitsoperatoren existiert dann ein $t_{i+1} \in \Sigma^*$ mit $s_{i+1}t_{i+1} \in M_i \parallel H_V$. Weiter ist $M_i = \Omega^i M$, sodass $s_{i+1}t_{i+1} \in (\Omega^i M) \parallel H_V$. Gemäß der Induktionsannahme existiert dann ein t_i mit $s_{i+1}t_{i+1}t_i \in M \parallel H_V$. Man bemerke dass mit $\Omega_\infty M$ eine Vereinigung universeller Rückwärtserreichbarkeitsoperatoren vorliegt, d. h. $\Omega_\infty M$ ist ebenfalls ein universeller Rückwärtserreichbarkeitsoperator. q.e.d.

Tatsächlich lässt sich zeigen, dass falls die Vorsilben eines Regelkreiskandidaten $\mathrm{pre}\,K$ die Eigenschaft

(M8) Universelle Erreichbarkeit, d. h. $\mathrm{pre}\,K = \Omega_\infty K$

erfüllen, der von $R := \mathrm{p_R}\,\mathrm{pre}\,K$ induzierte rekonfigurierende Regelkreis für beliebige H_V konfliktfrei ist.

Satz 2.2.2. Gegeben sei ein Rekonfigurationsproblem $(\Sigma, L_N, E_N, L_F, E_F, E_R)$ und die virtualisierte minimal-restriktive Lösung $H_V^{\uparrow}$ des Nominalentwurfsproblems (Σ_N, L_N, E_N). Betrachtet wird eine Sprache $K \subseteq \Sigma^*$ mit den Eigenschaften (M1) bis (M8), dann ist der Rekonfigurator $R := \mathrm{p_R}\,\mathrm{pre}\,K$ eine universelle Lösung des gegebenen Rekonfigurationsproblems. □

Für den praktisch interessanten Fall regulärer Parameter kann die Konvergenz der Folge $(M_i)_{i\in\mathbb{N}_0}$, induziert durch Iteration 2.9, nach einer endlichen Anzahl von Schritten nachgewiesen werden.

Im Folgenden wird eine Spezialisierung Ω' des Operators Ω, gemäß

$$\Omega' := \Omega_{1,A} \cup \Omega_{1,B} \cup \Omega_{1,C}$$

betrachtet.

In Anlehnung an [MBY+12] wird mit der nächsten Proposition eine universelle untere Schranke für die Mächtigkeit eines Zustandsraumes angegeben, auf dem die Iterate M_i der Iteration 2.9 für $\Omega = \Omega'$ realisiert werden können.

Proposition 2.2.6. Gegeben sei ein Rekonfigurationsproblem $(\Sigma, L_N, E_N, L_F, E_F, E_R)$ und ein Regelkreiskandidat $K \subseteq \Sigma^*$. Zu einer beliebigen Zielsprache $M \subseteq \mathrm{pre}\,K$ werden die Operatoren $\Omega_{1,A}$, $\Omega_{1,B}$ und $\Omega_{1,C}$ gemäß Prop. 2.2.3 betrachtet. Steht Ω stellvertretend für

einen der genannten Operatoren, dann gilt für beliebige $s', s'' \in \Sigma^*$

$$s' \equiv_{\mathrm{pre\,K}} s'' \quad \wedge \quad s' \equiv_{\mathrm{M}} s'' \quad \Rightarrow \quad s' \equiv_{\Omega\mathrm{M}} s''.$$

□

Beweis. Man wähle $M \subseteq \mathrm{pre}\,K$ und $s', s'' \in \Sigma^*$ mit $s' \equiv_{\mathrm{pre\,K}} s''$ und $s' \equiv_{\mathrm{M}} s''$ sowie ein beliebiges $t \in \Sigma^*$ mit $s't \in \Omega M$. Zu zeigen bleibt dann, dass $s''t \in \Omega M$ gilt. Der umgekehrte Schluss folgt dann aus Symmetriegründen.

Zu $\Omega_{1,\mathrm{A}}$. Aus $s't \in \Omega_{1,\mathrm{A}}\, M$ folgt sofort die Existenz eines $\sigma \in \Sigma_{\mathrm{P,F}}$, sodass $s't\sigma \in M$ gilt. Wegen $s' \equiv_{\mathrm{M}} s''$ folgt auch $s''t\sigma \in M$ und deshalb ist $s''t \in \Omega_{1,\mathrm{A}}\, M$.

Zu $\Omega_{1,\mathrm{B}}$. Mit $\sigma \in \Sigma_{\mathrm{CON,V}}$ im Wortlaut wie im Beweis zu $\Omega_{1,\mathrm{A}}$.

Zu $\Omega_{1,\mathrm{C}}$. Man wähle zu $s''t \in \mathrm{pre}\,K$ ein beliebiges $\sigma \in \Sigma$ mit $s''t\sigma \in \mathrm{pre}\,K$. Wegen $s' \equiv_{\mathrm{pre\,K}} s''$ ist auch $s't\sigma \in \mathrm{pre}\,K$. Es ist $s't \in \Omega_{1,\mathrm{C}}$, sodass sofort $s't\sigma \in M$ folgt. Wegen $s' \equiv_{\mathrm{M}} s''$ folgt daraus auch $s''t\sigma \in M$ und deshalb ist $s''t \in \Omega_{1,\mathrm{C}}\, M$. q.e.d.

Bezogen auf die Iteration 2.9 sichert Proposition 2.2.6 die Regularität der einzelnen Iterate M_{i}. Die endliche Konvergenz der Folge $(M_{\mathrm{i}})_{\mathrm{i}\in\mathbb{N}_0}$ ist dann eine Konsequenz ihrer Monotonieeigenschaften.

Proposition 2.2.7. Zu einem Rekonfigurationsproblem $(\Sigma, L_{\mathrm{N}}, E_{\mathrm{N}}, L_{\mathrm{F}}, E_{\mathrm{F}}, E_{\mathrm{R}})$ und einem Regelkreiskandidaten K betrachte man den Operator

$$\Omega' := \Omega_{1,\mathrm{A}} \cup \Omega_{1,\mathrm{B}} \cup \Omega_{1,\mathrm{C}}$$

über einer Zielsprache $M \subseteq \mathrm{pre}\,K$. Sind alle Parameter des gegebenen Rekonfigurationsproblems regulär, so erreicht die Iteration 2.9 für $\Omega = \Omega'$ nach einer endlichen Schrittzahl einen Fixpunkt. □

Beweis. Gemäß Proposition 2.2.6 gilt für eine beliebiges Iterat $M_{\mathrm{i}} \subseteq \mathrm{pre}\,K$, mit $i \in \mathbb{N}_0$

$$s' \equiv_{\mathrm{pre\,K}} s'' \quad \wedge \quad s' \equiv_{\mathrm{M_i}} s'' \quad \Rightarrow \quad s' \equiv_{\mathrm{M_{i+1}}} s''.$$

Nach Prop. [WR87], Lemma 3.1., gilt für die Anzahl der Nerodezellen von $M_{\mathrm{i+1}}$

$$|M_{\mathrm{i+1}}| \leq |M_{\mathrm{i}}||\mathrm{pre}\,K| + 1$$

und insbesondere

$$|M_{\mathrm{i}}| \leq |M||\mathrm{pre}\,K| + 1$$

für alle $i \in \mathbb{N}_0$.

Weiter bemerke man, dass $M_i \subseteq M_{i+1} = \Omega M$ gilt, sodass $(M_i)_{i\in\mathbb{N}_0}$ eine monoton steigende und durch pre$K$ beschränkte Folge ist. Aus dieser Monotonieeigenschaft folgt schließlich die endliche Konvergenz der Folge $(M_i)_{i\in\mathbb{N}_0}$ gegen einen Fixpunkt. q.e.d.

Für die Zwecke dieser Arbeit wurden die Erreichbarkeitsanalyse lediglich auf Basis der Operatoren $\Omega_{1,A}$, $\Omega_{1,B}$ und $\Omega_{1,C}$ implementiert.

Man kann zeigen, dass zu einer Spezifikation E die supremale Teilsprache bzgl. der Eigenschaften (M1) bis (M6) existiert. Im nächsten Abschnitt wird ihre Berechnung für reguläre Parameter im Zustandsraum diskutiert. Die Fortsetzung von Beispiel 1.4.10 wird an dieser Stelle vorgezogen, passend zum Abschluss des Entwurfs universeller Rekonfiguratoren.

Beispiel 2.2.1 (Fortsetzung von Bsp. 1.4.10). Zusammen mit der fehlerverträglichen Strecke aus Beispiel 1.4.1 bildet der virtualisierte minimal-restriktive Nominalregler aus Beispiel 1.5.1 die formale Strecke $L_F \parallel H_V^{\uparrow}$. Die Spezifikation besteht aus der fehlerverträglichen Spezifikation, Beispiel 1.4.3, und der Inaktivitätsbedingung aus Beispiel 1.5.2.

Abbildung 2.2.3 stellt die supremale Teilsprache $K^{\uparrow}$ der Spezifikation $E_F \parallel E_R$ bzgl. (M1) bis (M6) dar. Außerdem kann für $K^{\uparrow}$ die Eigenschaft (M8) verifiziert werden.

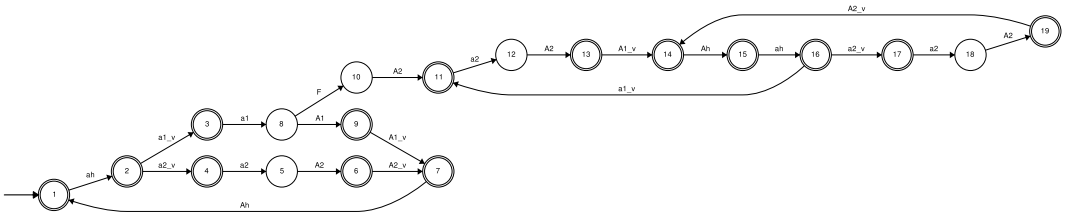


Abbildung 2.2.3: Supremale Teilsprache der Spezifikation bgzl. (M1) bis (M6)

Abbildung 2.2.4 zeigt den aus $K^{\uparrow}$ abgeleiteten Rekonfigurator $R^{\uparrow}$.

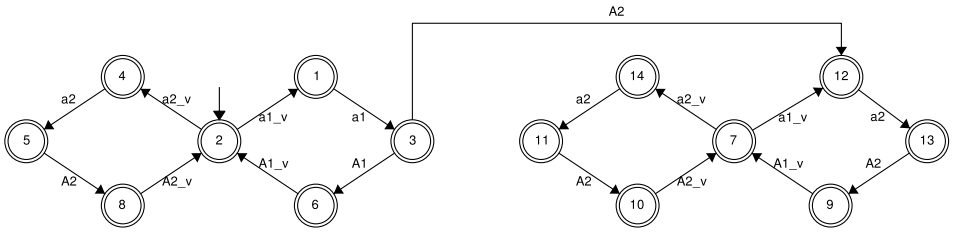


Abbildung 2.2.4: Supremale Teilsprache der Spezifikation bgzl. (M1) bis (M6)

Man bemerke, dass mit dem Ereignis *A2* in Zustand 3 das Auftreten eines Fehlers angezeigt wird. Vor dem Fehler werden die Inaktivitätsbedingungen umgesetzt. Nach dem

Fehler greift die fehlertolerante Steuerungsstrategie, d. h. der Regler weicht auf Prozess 2 aus, obwohl Prozess 1 angefordert wurde. Die Abfolge virtueller Ereignisse entspricht dabei dem Nominalfall. Der Fehler ist vor dem Nominalregler verdeckt.

Man bemerke weiter, dass für $H_{\mathrm{V}}^{\uparrow}$ der Rekonfigurator den Regelkreis $K^{\uparrow}$ induziert und dieser offensichtlich konfliktfrei und vollständig ist. Es bleibt die Lebendigkeit für spezielle Lösungen des Nominalentwurfsproblems zu prüfen. Aus dem minimal-restriktiven Regler $H_{\mathrm{V}}^{\uparrow}$ können zwei lebendige Lösungen abgeleitet werden, siehe Abbildungen 2.2.5 und 2.2.6,

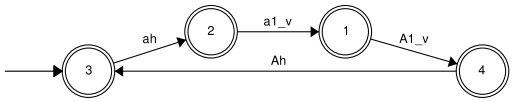


Abbildung 2.2.5: Spezieller virutalisierter Nominalregler H_{V1}

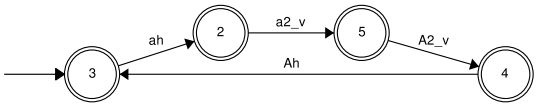


Abbildung 2.2.6: Spezieller virutalisierter Nominalregler H_{V2}

Die nachfolgenden Abbildungen 2.2.7 und 2.2.8 zeigen die rekonfigurierende Regelkreise für H_{V1} und H_{V1}.

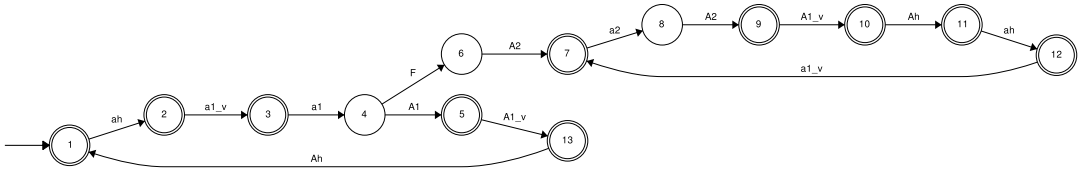


Abbildung 2.2.7: Rekonfigurierender Regelkreis für H_{V1}

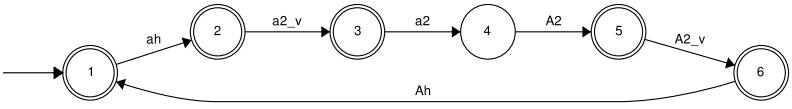


Abbildung 2.2.8: Rekonfigurierender Regelkreis für H_{V2}

Offensichtlich sichert der Rekonfigurator $R^{\uparrow}$ für alle gezeigten virutalisierten Nominalregler einen steuerbaren und lebendigen Regelkreis. □

2.3 Rekonfiguratorentwurf

Gegenstand dieses Abschnitts ist der Entwurf einer universellen Lösung $R^{\uparrow} \subseteq \Sigma_{\mathrm{R}}^{*}$ des Rekonfigurationsproblems $(\Sigma, L_{\mathrm{N}}, E_{\mathrm{N}}, L_{\mathrm{F}}, E_{\mathrm{F}}, E_{\mathrm{R}})$, die die formale Strecke $L_{\mathrm{F}} \parallel H_{\mathrm{V}}^{\uparrow}$ in ihrem Verhalten möglichst wenig einschränkt.

Bemerkung 2.3.1. Es sei an dieser Stelle darauf hingewiesen, dass die Eigenschaften (R1)-(R4) unter beliebiger Vereinigung erhalten bleiben. Zu einem gegebenen Rekonfigurationsproblem $(\Sigma, L_{\mathrm{N}}, E_{\mathrm{N}}, L_{\mathrm{F}}, E_{\mathrm{F}}, E_{\mathrm{R}})$ existiert also eine supremale universelle Lösung $R^{\Uparrow} \subseteq \Sigma_{\mathrm{R}}^{*}$. Allerdings sind die in den Abschnitten 2.1 und 2.2 angegebenen Bedingungen zur Ableitung einer universellen Lösung nur hinreichend und nicht notwendig. Insbesondere sind die geschlossenen Regelkreisverhalten, erzielt durch $R^{\uparrow}$ und $R^{\Uparrow}$, dann nicht zwingend identisch, d. h. für eine beliebige Lösung H_{V} des Nominalentwurfsproblems $(\Sigma_{\mathrm{N}}, L_{\mathrm{N}}, E_{\mathrm{N}})$ gilt lediglich $L_{\mathrm{F}} \parallel R^{\uparrow} \parallel H_{\mathrm{V}} \subseteq L_{\mathrm{F}} \parallel R^{\Uparrow} \parallel H_{\mathrm{V}}$, nicht jedoch die umgekehrte Einschließung. □

Als Kandidat zur Ableitung von $R^{\uparrow}$ dient die supremale Teilsprache $K^{\uparrow}$ der formalen Spezifikation $E_{\mathrm{F}} \parallel E_{\mathrm{R}}$ bzgl. der Eigenschaften (M1) bis (M7). Ist zusätzlich (M8) erfüllt, dann sichert, gemäß Satz 2.2.2, der Rekonfigurator $R^{\uparrow} := \mathrm{p}_{\mathrm{R}} \operatorname{pre} K^{\uparrow}$ konfliktfreies (R1), vollständiges (R2) und steuerbares (R3) Regelkreisverhalten unter Spezifikationseinschluss (R4). Umgekehrt existiert für $K^{\uparrow} = \emptyset$ oder, falls (M8) nicht erfüllt ist, keine nicht-triviale universelle Lösung des Rekonfigurationsproblems, die im rekonfigurierenden Regelkreis die Eigenschaften (M1) bis (M8) erzielt.

Ist $K^{\uparrow}$ bekannt, kann gemäß Satz 2.1.1 durch Vorsilbenbildung und Projektion der Rekonfigurator $R^{\uparrow}$ bestimmt werden. Sowohl Vorsilbenbildung als auch Projektion sind bereits bekannte Operationen. Effektiv reduziert sich damit die Berechnung von $R^{\uparrow}$ auf das Überführen der formalen Spezifikation E in $K^{\uparrow}$.

Von $K^{\uparrow}$ wird Supremalität simultan bzgl. der Eigenschaften (M1) bis (M7) gefordert. Die Formulierung eines entsprechenden Algorithmus ist deshalb kein triviales Problem. Neben der Angabe eines geeigneten Berechnungsschemas ist für den praktisch relevanten Fall regulärer Sprachen dessen Konvergenz nach einer endlichen Zahl von Schritten nachzuweisen. In der Literatur ist mit [MBY+12] ein passender Rahmen bekannt, der nachfolgend kurz umrissen wird.

Zu einer Spezifikation E seien $n \in \mathbb{N}$ verschiedene Regelkreiseigenschaften $A_1 \ldots A_{\mathrm{n}}$ gegeben. Weiter bezeichnen $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{\mathrm{n}}$ die Mengen aller Sprachen, die den jeweiligen Eigenschaften genügen. Mit $\mathcal{A} = \cap_{i \leq n} \mathcal{A}_{\mathrm{i}}$, $i \in \mathbb{N}$ wird die Menge aller Sprachen bezeichnet, die simultan alle Eigenschaften A_{i} erfüllt.

Jede beliebige Eigenschaft A_i mit $i \leq n$ bleibe unter beliebiger Vereinigung erhalten, formal

(A1) $\quad \mathcal{B} \subseteq A_\mathrm{i} \quad \Rightarrow \quad \bigcup_{K \in \mathcal{B}} K \in \mathcal{A}_\mathrm{i}$,

und sei anhand ihrer Vorsilben charakterisierbar, d. h.

(A2) $\quad K \in A_\mathrm{i} \quad \Leftrightarrow \quad \mathrm{pre}K \in \mathcal{A}_\mathrm{i}$.

Weiter seien mit $<\cdot>^{\uparrow(\mathcal{A}_\mathrm{i})}$, $i \leq n$ Operatoren bekannt, die E in ihre supremale Teilsprache bzgl. $\mathcal{A}_\mathrm{i}$

$$<E>^{\uparrow(\mathcal{A}_\mathrm{i})} := \{K \subseteq E \mid K \text{ erfüllt } \mathcal{A}_\mathrm{i}\}$$

überführen.

Die gesuchte supremale Teilsprache $<E>^{\uparrow(\mathcal{A})}$ wird in [MBY$^+$12] als Fixpunkt der Verkettung aller Operatoren $<\cdot>^{\uparrow(\mathcal{A}_\mathrm{i})}$

$$\Omega(K) = [<\cdot>^{\uparrow(\mathcal{A}_1)} \circ <\cdot>^{\uparrow(\mathcal{A}_2)} \circ \ldots \circ <\cdot>^{\uparrow(\mathcal{A}_\mathrm{i})}](\mathrm{pre}K) \cap E \tag{2.10}$$

charakterisiert. Man bemerke, dass die Berechnung der einzelnen supremalen Teilsprachen über abgeschlossene Sprachen durchgeführt werden kann. Zudem kann eine rechnergestützte Auswertung des Operators Ω durch ein einfaches Iterationsschema erfolgen.

In [MBY$^+$12] wird zudem die endliche Konvergenz der entsprechenden Iteration für reguläre Sprachen, realisiert durch endliche Automaten, diskutiert. Fokussiert werden dabei Algorithmen, die eine Realisierung der gesuchten supremalen Sprache lediglich durch Entfernen von Zuständen und Transitionen erreichen. Da es zu einem Automaten nur endlich viele unterscheidbare Teilautomaten gibt und lediglich Transitionen und Zustände entfernt, nicht jedoch hinzugefügt werden, ist ihre Konvergenz in endlicher Zeit offensichtlich. Es bleibt die Wahl eines hinreichend reichhaltigen Zustandsraums, auf dem die gesuchte supremale Teilsprache realisiert werden kann. Formal werden die beiden genannten Anforderungen in der Bedingung

(A3) Für eine beliebige Realisierung H der Spezifikation E, existiert ein Generator G_i mit $\mathrm{L}(G_\mathrm{i}) = \Sigma^*$, sodass

$$(\forall F \sqsubseteq G_\mathcal{A} \times H \; \exists F^\uparrow \sqsubseteq G_\mathcal{A} \times H)[\mathrm{L}(F^\uparrow) = <\mathrm{L}(F)>^{\uparrow(\mathcal{A})}]$$

gilt.

zusammengefasst.

Zur Realisierung der supremalen Teilsprache bei Berücksichtigung mehrerer Regelkreiseigenschaften, muss der initiale Zustandsraum reichhaltig genug sein, um die supremale Teilsprache bzgl. aller gewünschten Regelkreiseigenschaften darzustellen. Es wird zusätzlich die Forderung

(A4) Für eine beliebige Realisierung H der Spezifikation E existiert ein Generator $G_{\mathcal{A}}$ mit $\mathrm{L}(G_{\mathcal{A}}) = \Sigma^*$, sodass für jede Eigenschaft $\mathcal{A}_\mathrm{i}$, $i \leq n$

$$(\forall C \sqsubseteq G_{\mathcal{A}} \times H \; \exists C^{\uparrow} \sqsubseteq G_{\mathcal{A}} \times H)[\mathrm{L}(C^{\uparrow}) = < \mathrm{L}(C) >^{\uparrow(\mathcal{A}_\mathrm{i})}]$$

gilt.

gestellt.

Im Kontext der fehlerverdeckenden Steuerungsrekonfiguration ist die supremale Teilsprache der formalen Spezifikation E bezüglich der Eigenschaften (M1) bis (M7) zu berechnen. Für jede einzelne dieser Eigenschaften sind also die Anforderungen (A1) bis (A3) und für ihre Verknüpfung die Anforderung (A4) zu verifizieren.

Die Steuerbarkeitsbedingung (M1) und Vorsilbennormalität (M2) erfüllen diese Anforderungen bereits, siehe [RW87] und respektive [CL98]. Schlägt man L_V der Spezifikation E zu, d. h. man setze $E = E_\mathrm{F} \parallel E_\mathrm{R} \parallel (\mathrm{pre}\, L_\mathrm{V})$, dann ist (M6) offensichtlich erfüllt. Zudem sichern die Ergebnisse in [MBY+12] die relative Abgeschlossenheit (M3), falls E die Eigenschaft

(RE) Relative Abgeschlossenheit bzgl. $L_\mathrm{F} \parallel H_\mathrm{V}^{\uparrow}$, d. h. $E = (\mathrm{pre}\, E) \cap L_\mathrm{F} \parallel H_\mathrm{V}^{\uparrow}$.

aufweist. Obwohl Vollständigkeit in der Literatur diskutiert wird, ist dem Autor keine Diskussion der Σ_o-Vollständigkeit bekannt, die Entwurfsalgorithmen umfasst. Es verbleibt also die Diskussion für „Σ_o-Vollständigkeit“ und „Schwache Sensorkonsistenz“.

Sowohl für „Schwache Sensorkonsistenz“ als auch für „Σ_o-Vollständigkeit“ müssen Operatoren gefunden werden, die die Spezifikation E auf die entsprechenden supremalen Teilsprache abbildet. Die dazu vorgeschlagene Vorgehensweise lehnt sich konzeptionell an [WR87] an, worin die Autoren die Berechnung supremaler Teilsprachen in Termen spezieller Operatoren Ω auf Sprachen über Σ diskutieren. Ziel dieser Operatoren ist die Entfernung aller Zeichenketten, die mit der betrachteten Eigenschaft konfligieren.

Nachfolgend wird die Berechnung der supremalen schwach sensorkonsistenten Teilsprache und der Σ_o-vollständigen Teilsprache als Fixpunktiteration formuliert. Für reguläre Sprachen wird die Konvergenz nach endlich vielen Schritten nachgewiesen. Weiter werden für beide Eigenschaften Zustandsräume vorgestellt, auf denen die entsprechende supremale Teilsprache realisiert werden kann. Letztlich wird ein Zustandsraum angegeben,

der zur Berechnung der gesuchten supremale Teilsprache $K^{\uparrow}$ geeignet ist. Eine Prozedur zur Berechnung der supremalen Teilsprache bzgl. der Eigenschaften (M1) bis (M7) ergibt sich dann aus den Ergebnissen in [MBY+12].

2.3.1 Supremale schwach-sensorkonsistente Teilsprache

Zu einem Rekonfigurationsproblem $(\Sigma, L_{\mathrm{N}}, E_{\mathrm{N}}, L_{\mathrm{F}}, E_{\mathrm{F}}, E_{\mathrm{R}})$ und $L_{\mathrm{V}} := \mathrm{h} L_{\mathrm{N}} \subseteq \Sigma_{\mathrm{P,V}}^{*}$, heißt eine Sprache $K \subseteq \Sigma^{*}$ schwach-sensorkonsistent, falls

$$(\forall s \in \mathrm{pre}\, K)[(\mathrm{p}_{\mathrm{P,V}}\, s)\Sigma_{\mathrm{UCON,V}} \cap (\mathrm{pre}\, L_{\mathrm{V}}) \neq \emptyset \quad \Rightarrow \quad s(\Sigma - \Sigma_{\mathrm{C,V}})^{*}\Sigma_{\mathrm{UCON,V}} \cap \mathrm{pre}\, K \neq \emptyset]$$

gilt, vgl. (M5) in Abschnitt 2.1, und es bezeichnet

$$\mathcal{A}_{\mathrm{WSC}} := \{K \subseteq \Sigma^{*} \mid K \text{ ist schwach sensorkonsistent bzgl. } L_{\mathrm{V}}\} \tag{2.11}$$

die Menge aller bzgl. L_{V} schwach sensorkonsistenten Sprachen.

Schwache Sensorkonsistenz supremaler Elemente

Das Supremum einer beliebigen Teilmenge $\mathcal{B} \in \mathcal{A}_{\mathrm{WSC}}$ ist durch die Vereinigung all ihrer Elemente $K_{\cup} := \cup_{K \in \mathcal{B}} K$ gegeben. Die Eigenschaft

(A1) $\quad \mathcal{B} \subseteq \mathcal{A}_{\mathrm{WSC}} \quad \Rightarrow \quad \cup_{K \in \mathcal{B}} K \in \mathcal{A}_{\mathrm{WSC}}$

fordert, dass mit allen Elementen $K \in \mathcal{B}$ auch $K_{\cup}$ selbst schwach sensorkonsistent ist.

Um sicherzustellen, dass (A1) tatsächlich erfüllt ist, wähle man eine beliebige Teilmenge $\mathcal{B} \subseteq \mathcal{A}_{\mathrm{WSC}}$. Um zu zeigen, dass $K_{\cup}$ selbst schwach sensorkonsistent ist, wähle man ein beliebiges $s \in \mathrm{pre}\, K_{\cup}$ mit $(\mathrm{p}_{\mathrm{P,V}}\, s)\Sigma_{\mathrm{UCON,V}} \cap (\mathrm{pre}\, L_{\mathrm{V}}) \neq \emptyset$. Wegen $s \in \mathrm{pre}\, K_{\cup} = \cup_{K \in \mathcal{B}} \mathrm{pre}\, K$ existiert ein $K \in \mathcal{B}$ mit $s \in \mathrm{pre}\, K$. Es ist K selbst schwach sensorkonsistent, weshalb auch ein $r \in (\Sigma - \Sigma_{\mathrm{C,V}})^{*}\Sigma_{\mathrm{UCON,V}}$ mit $sr \in \mathrm{pre}\, K \subseteq \mathrm{pre}\, K_{\cup}$ existiert. Folglich ist $K_{\cup}$ selbst schwach sensorkonsistent und man hat $K_{\cup} \in \mathcal{B}$.

Das Supremum einer beliebigen Teilmenge $\mathcal{B} \in \mathcal{A}_{\mathrm{WSC}}$ ist also selbst schwach sensorkonsistent. Damit formt $(\mathcal{A}_{\mathrm{WSC}}, \cup)$ einen vollständigen oberen Halbverband. Folglich ist die Existenz der supremalen schwach-sensorkonsistenten Teilsprache einer gegebenen Spezifikation $E \subseteq \Sigma^{*}$, in Zeichen

$$< E >^{\uparrow(\mathrm{WSC})} := \sup\{K \subseteq E \mid K \in \mathcal{A}_{\mathrm{WSC}}\},$$

gesichert.

Vorsilbencharakterisierung

Die Forderung nach der Charakterisierbarkeit schwacher Sensorkonsistenz anhand der Vorsilben einer Sprache wird durch die Anforderung

$$\text{(A2)} \qquad K \in \mathcal{A}_{\text{WSC}} \quad \Leftrightarrow \quad \text{pre}\, K \in \mathcal{A}_{\text{WSC}}\,.$$

ausgedrückt.

Man bemerke, dass sich die schwache Sensorkonsistenz einer Sprache K ausschließlich auf ihre Vorsilben $\text{pre}\, K$ bezieht. Wegen $\text{pre}\,\text{pre}\, K = \text{pre}\, K$ wird die Bedingung von K erfüllt, falls sie für $\text{pre}\, K$ gilt. Daraus folgt (A2).

Fixpunktcharakterisierung des supremalen Elements

Zur Berechnung der supremalen schwach sensorkonsistenten Teilsprache zu einer Sprache $K \subseteq \Sigma^*$ wird der Operator Ω_{WSC}, gemäß

$$\begin{aligned}&\Omega_{\text{WSC}} K :=\\ &\{ t \in K \mid (\forall s \leq t)[s\Sigma_{\text{UCON,V}} \cap (\text{pre}\, \text{p}_{\text{P,V}}^{-1} L_{\text{V}}) \neq \emptyset \Rightarrow s(\Sigma - \Sigma_{\text{C,V}})^* \Sigma_{\text{UCON,V}} \cap \text{pre}\, K \neq \emptyset] \}\,,\end{aligned} \tag{2.12}$$

gewählt. Der Operator Ω_{WSC} reduziert die Sprache K auf alle Zeichenketten $t \in K$, die nicht mit schwacher Sensorkonsistenz konfligieren.

Aus der Definition des Operators Ω_{WSC} folgt sofort, dass das Bild einer Sprache K unter Ω_{WSC} vollständig in K enthalten ist, d. h. $\Omega_{\text{WSC}} K \subseteq K$. Dabei gilt Gleichheit genau dann, wenn K selbst schwach sensorkonsistent ist, wenn also $K \in \mathcal{A}_{\text{WSC}}$ gilt. Die Folge $(K_i)_{i \in \mathbb{N}_0}$, gegeben durch

$$\begin{aligned}&K_0 := E\\ &K_{\text{i+1}} := \Omega_{\text{WSC}} K_i,\end{aligned} \tag{2.13}$$

ist wegen $\Omega_{\text{WSC}} K_{\text{i}} \subseteq K_{\text{i}}$, für $i \in \mathbb{N}_0$, monoton fallend und strebt definitionsgemäß gegen den Grenzwert $K_\infty := \cap_{i \in \mathbb{N}_0} K_i$. Dieser stellt einen potentiellen Kandidaten für $< E >^{\uparrow(\text{WSC})}$ dar.

Proposition 2.3.1. Zu einer Spezifikation $E \subseteq \Sigma^*$ betrachte man die durch Iteration (2.13) definierte Folge $(K_i)_{i \in \mathbb{N}_0}$ mit dem Grenzwert $K_\infty := \cap_{i \in \mathbb{N}_0} K_i$. Die supremale schwach sensorkonsistente Teilsprache $< E >^{\uparrow(\text{WSC})}$ der Spezifikation E ist durch den Grenzwert K_∞ der Folge $(K_i)_{i \in \mathbb{N}_0}$ genau dann gegeben, falls K_∞ ein Fixpunkt des Operators Ω_{WSC} ist, in

Zeichen

$$K_\infty =< E >^{\uparrow(\mathrm{WSC})} \quad \Leftrightarrow \quad K_\infty = \Omega_{\mathrm{WSC}} K_\infty .$$

□

Beweis. Zunächst zeige man durch Induktion über die Folgenglieder K_i, dass für alle $i \in \mathbb{N}_0$

$$< E >^{\uparrow(\mathrm{WSC})} \subseteq K_i \tag{2.14}$$

gilt. Für $i = 0$ ist $K_0 = E$, sodass die Behauptung aus der Definition von $< E >^{\uparrow(\mathrm{WSC})}$ als supremale schwach sensorkonsistente Teilsprache von E folgt. Man nehme nun an, es gäbe ein $i \in \mathbb{N}_0$, für das $< E >^{\uparrow(\mathrm{WSC})} \subseteq K_i$ gilt, und man zeige durch Widerspruch, dass daraus auch $< E >^{\uparrow(\mathrm{WSC})} \subseteq K_{i+1}$ folgt. Dazu nehme man an, es gelte sowohl $< E >^{\uparrow(\mathrm{WSC})} \subseteq K_i$ als auch $< E >^{\uparrow(\mathrm{WSC})} \not\subseteq K_{i+1} = \Omega_{\mathrm{WSC}} K_\mathrm{i}$, d. h. durch den Operator Ω_{WSC} werden Zeichenketten aus dem Argument entfernt, die nicht mit schwacher Sensorkonsistenz konfligieren. Man wähle $t \in \Sigma^*$ mit $t \in < E >^{\uparrow(\mathrm{WSC})} \subseteq K_\mathrm{i}$ und $t \notin \Omega K_\mathrm{i}$. Gemäß (A1) ist $< E >^{\uparrow(\mathrm{WSC})}$ selbst schwach sensorkonsistent, sodass für alle Vorsilben $s \leq t$ mit $s\Sigma_{\mathrm{UCON,V}} \cap (\mathrm{pre}\, \mathrm{p}_{\mathrm{P,V}}^{-1} L_\mathrm{V}) \neq \emptyset$ auch $s(\Sigma - \Sigma_{\mathrm{C,V}})^* \Sigma_{\mathrm{UCON,V}} \cap \mathrm{pre} < E >^{\uparrow(\mathrm{WSC})} = \emptyset$ gilt. Aus der Monotonie des Vorsilbenperators pre folgt wegen $< E >^{\uparrow(\mathrm{WSC})} \subseteq K_\mathrm{i}$ auch $\mathrm{pre} < E >^{\uparrow(\mathrm{WSC})} \subseteq \mathrm{pre}\, K_\mathrm{i}$. Insbesondere gilt auch $s(\Sigma - \Sigma_{\mathrm{C,V}})^* \Sigma_{\mathrm{UCON,V}} \cap \mathrm{pre}\, K_i = \emptyset$. Gemäß der Definition des Operators Ω_{WSC} ist $t \in \Omega_{\mathrm{WSC}} K_\mathrm{i}$. Dies ist ein Widerspruch zur Voraussetzung $t \in < E >^{\uparrow(\mathrm{WSC})} \not\subseteq \Omega K_\mathrm{i}$.

Um die Aussage des obigen Satzes zu beweisen, nehme man an, es gelte $K_\infty =< E >^{\uparrow(\mathrm{WSC})}$. Dann ist mit $< E >^{\uparrow(\mathrm{WSC})}$ auch K_∞ schwach sensor-konsistent. Zusammen mit der Definition des Operators Ω_{WSC} folgt sofort $K_\infty = \Omega_{\mathrm{WSC}} K_\infty$. Umgekehrt nehme man an, es gelte $K_\infty = \Omega_{\mathrm{WSC}} K_\infty$. Bereits gezeigt wurde $< E >^{\uparrow(\mathrm{WSC})} \in K_\mathrm{i}$ für alle $i \in \mathbb{N}_0$. Zusammen mit der Monotonie der Folge $(K_i)_{i \in \mathbb{N}_0}$ ergibt sich $< E >^{\uparrow(\mathrm{WSC})} \subseteq \cap_{i \in \mathbb{N}_0} K_i = K_\infty$. Es ist K_∞ ein Fixpunkt des Operators Ω_{WSC}, sodass K_∞ selbst schwach sensorkonsistent ist. Folglich gilt $K_\infty \subseteq < E >^{\uparrow(\mathrm{WSC})}$. q.e.d.

Endliche Konvergenz für reguläre Parameter

Falls E und L_V regulär sind, lässt sich aus der folgenden Proposition die endliche Konvergenz der Folge $(K_i)_{i \in \mathbb{N}_0}$, induziert durch die Iteration (2.13), gegen $K_\infty =< E >^{\uparrow(\mathrm{WSC})}$ schließen.

Proposition 2.3.2. Gegeben sei ein Rekonfigurationsproblem $(\Sigma, L_\mathrm{N}, E_\mathrm{N}, L_\mathrm{F}, E_\mathrm{F}, E_\mathrm{R})$. Zu einer Spezifikation $E \subseteq \Sigma^*$ betrachte man die durch Iteration (2.13) definierte Folge

$(K_i)_{i\in\mathbb{N}_0}$ mit dem Grenzwert $K_\infty := \cap_{i\in\mathbb{N}_0} K_i$. Falls $L_\mathrm{V} := \mathrm{h} L_\mathrm{N}$ und E regulär sind, dann erhält Ω_WSC die Regularität. Zudem existiert $k \in \mathbb{N}_0$ so dass $K_\infty = K_k$.

Beweis. Kann eine uniforme untere Schranke für die Mächtigkeit eines Zustandsraums angegeben werden, auf dem die Iterate K_i, $i \in \mathbb{N}_0$ dargestellt werden können, folgt aus der Monotonieeigenschaft der Folge $(K_i)_{i\in\mathbb{N}_0}$ auch ihre Konvergenz nach endlich vielen Iterationen und die Regularität ihrer Folgenglieder.

Zunächst wird dazu für alle s', $s'' \in \mathrm{pre}\, K_i$

$$s' \equiv_E s'' \quad \wedge \quad s' \equiv_{\mathrm{pre}\, \mathrm{p}_{\mathrm{P,V}} L_\mathrm{V}} s'' \quad \Rightarrow \quad s' \equiv_{K_i} s'' \tag{2.15}$$

gezeigt. Für $i = 0$ gilt $K_0 = E$ und die obige Annahme ist offensichtlich wahr. Für einen Beweis durch Induktion nehme man an, dass die Implikation für ein $i \in \mathbb{N}_0$ gilt und man betrachte beliebige s', $s'' \in \mathrm{pre}\, K_{\mathrm{i}+1}$ mit $s' \equiv_E s''$ und $s' \equiv_{\mathrm{pre}\, \mathrm{p}^{-1}_{\mathrm{P,V}} L_\mathrm{V}} s''$. Gemäß der Induktionsannahme gilt $K_{\mathrm{i}+1} \subseteq K_i$ und man erhält $s' \equiv_{K_i} s''$. Nun wähle man t, sodass $s't \in K_{\mathrm{i}+1}$ ist. Insbesondere gilt $s't \in K_i$ und wegen $s' \equiv_{\mathrm{K_i}} s''$ gilt auch $s''t \in K_i$. Um $s''t \in K_{\mathrm{i}+1}$ zu zeigen, wähle man $r \leq s''t$ mit $r\Sigma_{\mathrm{UCON,V}} \cap (\mathrm{pre}\, \mathrm{p}^{-1}_{\mathrm{P,V}} L_\mathrm{V}) \neq \emptyset$. Es werden zwei Fälle unterschieden:

Fall 1. Falls $r \leq s''$ gilt, beziehe man sich auf die Voraussetzung $s'' \in \mathrm{pre}\, K_{\mathrm{i}+1} = \Omega_\mathrm{WSC} K_\mathrm{i}$ um zusammen mit der Definition des Operators Ω_WSC auf $r(\Sigma - \Sigma_{\mathrm{C,V}})^* \Sigma_{\mathrm{UCON,V}} \cap \mathrm{pre}\, K_i \neq \emptyset$ zu schließen.

Fall 2. Für den komplementären Fall $s'' < r$, schreibe man $s''v = r$ und folgere aus der Invarianz einer Rechtskongruenz gegenüber Konkatenation zunächst $s'v \equiv_{\mathrm{pre}\, \mathrm{p}^{-1}_{\mathrm{P,V}} L_\mathrm{V}} s''v$ und daraus $s'v \equiv_{\mathrm{pre}\, \mathrm{p}^{-1}_{\mathrm{P,V}} L_\mathrm{V}} r$. Letzeres impliziert $s'v\Sigma_{\mathrm{UCON,V}} \cap (\mathrm{pre}\, \mathrm{p}^{-1}_{\mathrm{P,V}} L_\mathrm{V}) \neq \emptyset$ und wegen $s'v \leq s't \in K_{\mathrm{i}+1}$ ist auch $s'v \in \mathrm{pre}\, K_{\mathrm{i}+1}$, sodass ein $u \in (\Sigma - \Sigma_{\mathrm{C,V}})^* \Sigma_{\mathrm{UCON,V}}$ mit $s'vu \in \mathrm{pre}\, K_i$ gewählt werden kann. Aus $s' \equiv_{K_i} s''$ folgt $s''vu \in \mathrm{pre}\, K_i$, sodass auch für den zweiten Fall $r(\Sigma - \Sigma_{\mathrm{C,V}})^* \Sigma_{\mathrm{UCON,V}} \cap \mathrm{pre}\, K_i \neq \emptyset$ gilt. Dies schließt den Beweis von $s''t \in K_{\mathrm{i}+1}$ ab. Durch Variation von t erhält man $s't \in K_{\mathrm{i}+1}$, was $s''t \in K_{\mathrm{i}+1}$ impliziert. Uniforme Substitution liefert die Umkehrung. Deshalb gilt $s' \equiv_{\mathrm{K_{i+1}}} s''$ und die Implikation (2.15) folgt durch Induktion.

Mit Implikation (2.15) als Voraussetzung folgt aus Lemma 3.1 aus [WR87] eine obere Schranke für die Anzahl der Neurode-Äquivalenzklassen von $\equiv_{K_i}$ in Σ^* in Termen der entsprechenden Anzahl von $\equiv_E$ und $\equiv_{\mathrm{pre}\, \mathrm{p}^{-1}_{\mathrm{P,V}} L_\mathrm{V}}$. Insbesondere ist, falls E und $\mathrm{pre}\, \mathrm{p}^{-1}_{\mathrm{P,V}} L_\mathrm{V}$ regulär sind, diese Schranke endlich und deshalb sind die Iterate K_i regulär.

Weiter ist die Schranke uniform in i und gemäß [WR87], Lemma 3.2, gibt es nur endlich viele unterschiedliche Iterate K_i. Wegen der Monotonie von $(K_i)_{i\in\mathbb{N}_0}$ muss ein Fixpunkt in einer endlichen Anzahl von Schritten erreicht werden, d. h. es existiert ein $k \in \mathbb{N}_0$,

sodass $K_i = K_k$ für alle $i \geq k$ gilt; siehe [WR87], Lemma 3.3. , was den Beweis endlicher Konvergenz abschließt.

q.e.d.

Realisierung im Zustandsraum

Für die weitere Diskussion werden $\mathrm{pre}\,\mathrm{p}_{\mathrm{P,V}}^{-1} L_{\mathrm{V}}$ und E als regulär angenommen mit den trimmen Realisierungen $G = (Q, \Sigma, \delta, q_{\mathrm{o}}, Q_{\mathrm{m}})$ und $H = (X, \Sigma, \xi, x_{\mathrm{o}}, X_{\mathrm{m}})$, d. h. $\mathrm{L}(G) = \mathrm{pre}\,\mathrm{p}_{\mathrm{P,V}}^{-1} L_{\mathrm{V}}$ und $\mathrm{L_m}(H) = E$ bzw. $\mathrm{L}(H) = \mathrm{pre}\,E$.

Gemäß Proposition 2.3.1 liefert eine beliebige Implementierung von Ω_{WSC} zusammen mit Iteration (2.13) eine Möglichkeit zur Berechnung der supremalen schwach sensorkonsistenten Teilsprache $< E >^{\uparrow(\mathrm{WSC})}$ einer gegebenen Spezifikation $E \subseteq \Sigma^*$. Bezieht man sich dazu allerdings auf den Rahmen in [MBY$^+$12], dann kann die Berechnung von $< E >^{\uparrow(\mathrm{WSC})}$ mit Hilfe der Iteration

$$\begin{aligned} &N_0 := E\,, \\ &N_{\mathrm{i}+1} := < \mathrm{pre}\,N_{\mathrm{i}} >^{\uparrow(\mathrm{WSC})} \cap E\,. \end{aligned} \tag{2.16}$$

durchgeführt werden. Hier wird zwar $< \mathrm{pre}\,N_{\mathrm{i}} >^{\uparrow(\mathrm{WSC})}$ durch die Iteration (2.13) ausgewertet, allerdings ist es ausreichend, den Operator Ω_{WSC} für abgeschlossene Sprachen umzusetzen. Der Schnitt mit E kann dann durch Produktkomposition realisiert werden. Die Charakterisierung von $< E >^{\uparrow(\mathrm{WSC})}$ als Fixpunkt der Iteration (2.16) ist dann eine Konsequenz des Satzes 5 in [MBY$^+$12]. Kann zudem

(A3) Für eine beliebige Realisierung H der Spezifikation E existiert ein Generator G_{WSC} mit $\mathrm{L}(G_{\mathrm{WSC}}) = \Sigma^*$, sodass

$$(\forall F \sqsubseteq G_{\mathrm{WSC}} \times H \; \exists F^{\uparrow} \sqsubseteq G_{\mathrm{WSC}} \times H)[\mathrm{L}(F^{\uparrow}) = < \mathrm{L}(F) >^{\uparrow(\mathrm{WSC})}]$$

gilt.

verifiziert werden, dann ist die endliche Konvergenz der Iteration 2.16 eine Konsequenz des Satzes 7 in [MBY$^+$12].

Für die Wahl des Generators G_{WSC} erinnere man Implikation (2.15), die zur Realisierung der Iterate K_{i} gemäß Iteration (2.13) einen Zustandsraum fordert, der wenigstens so fein ist, dass entschieden werden kann, ob eine Zeichenkette sowohl in der Spezifikation E als auch in den Vorsilben der virtuellen Strecke $\mathrm{pre}\,\mathrm{p}_{\mathrm{P,V}}^{-1} L_{\mathrm{V}}$ liegt. Dies legt die Realisierung

G von $\mathrm{pre}\,\mathrm{p}_{\mathrm{P,V}}^{-1} L_{\mathrm{V}}$ als Wahl für G_{WSC} nahe. Um $\mathrm{L}(G_{\mathrm{WSC}}) = \Sigma^*$ sicherzustellen, konstruiere man $G_{\mathrm{WSC}} = (Q', \Sigma, \delta', q_{\mathrm{o}}, Q_{\mathrm{m}})$ durch Erweiterung von G mit einem Zustand $q_{\mathrm{dmp}} \notin Q$ und zusätzlichen Transitionen $\delta'(q_{\mathrm{o}}, s) = q_{\mathrm{dmp}}$ für alle $s \in \Sigma^* - \mathrm{L}(G)$. Man bemerke einerseits $G \sqsubseteq G_{\mathrm{WSC}}$ und andererseits $\mathrm{L}(G_{\mathrm{WSC}} \times H) = \mathrm{L}(H)$.

Neben der Wahl eines geeigneten Zustandsraumes muss eine Umsetzung des Operators Ω_{WSC} für abgeschlossene Sprachen gefunden werden, sodass für beliebiges $i \in \mathbb{N}_0$ die Iterate K_{i} und $K_{\mathrm{i+1}}$ durch Realisierungen $F \sqsubseteq G_{\mathrm{WSC}} \times H$ und $F' \sqsubseteq F$ dargestellt werden können. Anforderung (A3) ist dann eine direkte Konsequenz von Proposition 2.3.1.

Gegeben sei $F = (Z, \Sigma, \zeta, z_{\mathrm{o}}, Z_{\mathrm{m}})$, $F \sqsubseteq G_{\mathrm{WSC}} \times H$, es wird $F' \sqsubseteq F$ so konstruiert, dass $\mathrm{L}(F') = \Omega_{\mathrm{WSC}}\,\mathrm{L}(F)$. Betrachtet werden die Zustandsmengen

$$Z_{\mathrm{LUCON}} := \{(q,x) \in Q \times X \mid (\exists \sigma \in \Sigma_{\mathrm{UCON,V}})[\delta(q,\sigma)!]\} \tag{2.17}$$

$$Z_{\mathrm{KUCON}} := \{(q,x) \in Q' \times X \mid (\exists r \in (\Sigma - \Sigma_{\mathrm{C,V}})^* \Sigma_{\mathrm{UCON,V}})[\zeta((q,x),r)!]\}. \tag{2.18}$$

In jedem Zustand $z \in Z_{\mathrm{LUCON}}$ kann die virtuelle Strecke ein virtuelles nicht-steuerbares Ereignis ausführen. Die Zustandsmenge Z_{KUCON} enthält alle Zustände $(q,x) \in Q \times X$, in denen F eine Zeichenkette generieren kann, die in einem virtuellen nicht-steuerbaren Ereignis endet.

Um die Implikation in der Definition Ω_{WSC} zu gefährden, muss eine Zeichenkette $s \in \Sigma^*$ in einem Zustand $(q,x) \in Z_{\mathrm{LUCON}} - Z_{\mathrm{KUCON}}$ enden. Man definiere deshalb $F' = (Z', \Sigma, \zeta', z_{\mathrm{o}}, Z'_{\mathrm{m}}) := F|_{Z'}$ als die Restriktion von F auf eine Zustandsmenge $Z' := \{z \in Z \,|\, z \in Z_{\mathrm{LUCON}} \;\Rightarrow\; z \in Z_{\mathrm{KUCON}}\}$. Im Besonderen ist F' der leere Generator $G_\emptyset$ genau dann, wenn F der leere Generator ist oder $z_{\mathrm{o}} \in Z_{\mathrm{LUCON}} - Z_{\mathrm{KUCON}}$. Mit der nachfolgenden Proposition wird die Verifikation von (A3) abgeschlossen.

Proposition 2.3.3. In dem oben aufgespannten Rahmen, in dem man $F \sqsubseteq G_{\mathrm{WSC}} \times H$ annimmt, resultiert die Konstruktion $F' \sqsubseteq F$ in $\mathrm{L}(F') = \Omega_{\mathrm{WSC}}\,\mathrm{L}(F)$. □

Beweis. Falls F der leere Automat ist, dann gilt das auch für F' und es gilt $\mathrm{L}(F') = \Omega_{\mathrm{WSC}}\,\mathrm{L}(F) = \emptyset$. Im Folgenden sei F nicht der leere Generator, und es bezeichnet $K := \mathrm{L}(F) \neq \emptyset$ die von F generierte Sprache.

Um $\Omega_{\mathrm{WSC}} K \subseteq \mathrm{L}(F')$ zu zeigen, wähle man $t \in \Omega_{\mathrm{WSC}} K \subseteq K$ beliebig. Aus der Definition des Operators Ω_{WSC}, siehe Gl. (2.12), folgt für beliebiges $s \leq t$ mit $s\sigma \in \mathrm{pre}\,\mathrm{p}_{\mathrm{P,V}}^{-1} L_{\mathrm{V}}$ für ein beliebiges $\sigma \in \Sigma_{\mathrm{UCON,V}}$ die Existenz einer Fortsetzung $r \in (\Sigma - \Sigma_{\mathrm{C,V}})^* \Sigma_{\mathrm{UCON,V}}$ mit $sr \in K$. Das schließt auch den Spezialfall $s = \varepsilon$ ein. Es ist $s\sigma \in \mathrm{pre}\,\mathrm{p}_{\mathrm{P,V}}^{-1} L_{\mathrm{V}} = \mathrm{L}(G)$, sodass $\delta(q_0, s\sigma)!$. Daraus ergibt sich für $z_{\mathrm{s}} \in Z$ mit $\zeta(z_0, s) = z_{\mathrm{s}}$ aus Gl. (2.17) auch $z_{\mathrm{s}} \in Z_{\mathrm{LUCON}}$.

Weiter ist $sr \in K$, sodass auch $\zeta(z_0, sr)!$. Gemäß Gl. (2.18) ist dann $z_s \in Z_{\text{KUCON}}$. Insbesondere ist damit $z \in Z'$, sodass $s \in \mathrm{L}(F')$ gilt. Zudem wurde s als Vorsilbe von t beliebig gewählt, woraus schließlich $t \in \mathrm{L}(F')$ folgt.

Falls F' der leere Generator ist, dann ist $\mathrm{L}(F') \subseteq \Omega_{\text{WSC}} K$ trivial erfüllt. Im Folgenden wird von $\mathrm{L}(F') \neq \emptyset$ ausgegangen, insbesondere ist $F' \neq G_\emptyset$. Man wähle ein beliebiges $t \in \mathrm{L}(F')$. Aus $F' \sqsubseteq F$ folgere man $t \in K$. Es bleibt zu zeigen, dass für $s \leq t$ die Implikation in Ω_{WSC}, siehe Gl. (2.12), erfüllt ist. Falls $s\Sigma_{\text{UCON,V}} \cap (\mathrm{pre}\, \mathrm{p}_{\text{P,V}}^{-1} L_{\text{V}}) = \emptyset$ ist das geforderte Konditional trivial erfüllt. Im anderen Fall $s\Sigma_{\text{UCON,V}} \cap (\mathrm{pre}\, \mathrm{p}_{\text{P,V}}^{-1} L_{\text{V}}) \neq \emptyset$, ist $\delta(q_0, s\sigma) = q$ für ein $\sigma \in \Sigma_{\text{UCON,V}}$ und $q \in Q$. Es ist $s \in \mathrm{L}(F) \subseteq \mathrm{L}(G_{\text{WSC}} \times H)$, sodass für ein $(q', x) \in Q' \times X$ auch $\zeta((q_0, x_0), s) = (q', x)$ gilt. Wegen $F \sqsubseteq G_{\text{WSC}} \times H$ und $G \sqsubset G_{\text{WSC}}$ gilt $\delta(q_0, s) = \delta'(q_0, s) = q'$, d. h. die Zustände q und q' in (q', x) sind identisch. Deshalb ist $\delta(q', \sigma)$ für ein $\sigma \in \Sigma_{\text{UCON,V}}$ definiert und demnach ist $(q', x) \in Z_{\text{LUCON}}$. Gemäß Konstruktion ist $F' = F|_{Z'}$ mit $Z' := \{z \in Z \,|\, z \in Z_{\text{LUCON}} \;\Rightarrow\; z \in Z_{\text{KUCON}}\}$ und deshalb ist $(q', x) \in Z_{\text{KUCON}}$. Deshalb existiert ein $r \in (\Sigma - \Sigma_{\text{C,V}})^* \Sigma_{\text{UCON,V}}$ mit $\zeta((q, x), r)!$. Daraus folgt $\zeta(z_0, sr)!$. Deshalb ist $sr \in \mathrm{pre}\, K$. Insgesamt ist $s(\Sigma - \Sigma_{\text{C,V}})^* \Sigma_{\text{UCON,V}} \cap \mathrm{pre}\, K \neq \emptyset$ und demnach $t \in \Omega_{\text{WSC}} K$. q.e.d.

Eine Rechnerumsetzung der Berechnung von $< \mathrm{pre}\, E >^{\uparrow(\text{WSC})}$ startet mit $F_0 := G_{\text{WSC}} \times H$ und $\mathrm{L}(F_0) = \mathrm{pre}\, E = K_0$. Durch Iteration der obigen Teilautomatenkonstruktion erhält man eine fallende Folge $(F_i)_{i \in \mathbb{N}_0}$, $F_{i+1} \sqsubseteq F_i \sqsubseteq F_0$, und gemäß Proposition 2.3.3 ergibt sich $\mathrm{L}(F_i) = K_i$ für alle $i \in \mathbb{N}_0$. Da es nur endlich viele Teilautomaten von F_0 gibt, folgt aus der Monotonie, dass ein Fixpunkt $F_k = F_{k+1}$ nach einer endlichen Anzahl von Schritten $k \in \mathbb{N}_0$ erreicht wird. Proposition 2.3.1 impliziert dann $\mathrm{L}(F_k) = K_\infty =< \mathrm{pre}\, E >^{\uparrow(\text{WSC})}$.

Damit ist die Diskussion der supremalen schwach sensorkonsistenten Teilsprache abgeschlossen. Die nachfolgenden Betrachtungen der supremalen Σ_{o}-vollständigen Teilsprache sind stark an die bisherigen Ausführungen angelehnt.

2.3.2 Supremale Σ_{o}-vollständige Teilsprache

Eine Sprache $K \subseteq \Sigma^*$ heißt Σ_{o}-vollständig, falls

$$(\forall s \in \mathrm{pre}\, K \, \exists t \in (\Sigma - \Sigma_{\text{o}})^*, \sigma \in \Sigma_{\text{o}})[st\sigma \in \mathrm{pre}\, K]$$

gilt, vgl. (M4) in Abschnitt 2.1 mit $\Sigma_{\text{o}} = \Sigma - \Sigma_{\text{HI}}$, und es bezeichnet

$$\mathcal{A}_{\text{CMP}} := \{K \subseteq \Sigma^* \mid K \text{ ist } \Sigma_{\text{o}}\text{-vollständig}\} \tag{2.19}$$

die Menge aller Σ_{o}-vollständigen Sprachen.

Σ_o-Vollständigkeit supremaler Elemente

Das Supremum einer beliebigen Teilmenge $\mathcal{B} \in \mathcal{A}_{\text{CMP}}$ ist durch die Vereinigung all ihrer Elemente $K_\cup := \cup_{K\in\mathcal{B}} K$ gegeben. Die Eigenschaft

(B1) $\quad \mathcal{B} \subseteq \mathcal{A}_{\text{CMP}} \quad \Rightarrow \quad \cup_{K\in\mathcal{B}} K \in \mathcal{A}_{\text{CMP}}$,

fordert, dass mit allen Elementen $K \in \mathcal{A}_{\text{CMP}}$ auch ihre Vereinigung $K_\cup$ selbst Σ_o-vollständig ist.

Zur Verifikation von (B1) betrachte man eine beliebige Teilmenge $\mathcal{B} \subseteq \mathcal{A}_{\text{CMP}}$. Um zu zeigen, dass $K_\cup$ selbst Σ_o-vollständig ist, wähle man $s \in \text{pre}\, K_\cup$ beliebig. Insbesondere gilt $s \in \text{pre}\, K$ für ein $K \in \mathcal{B}$ und deshalb existiert ein $r\sigma \in (\Sigma - \Sigma_o)^*\Sigma_o$, sodass $sr\sigma \in \text{pre}\, K \subseteq \text{pre}\, K_\cup$. Folglich ist $K_\cup$ selbst Σ_o-vollständig.

Das Supremum einer beliebigen Teilmenge $\mathcal{B} \in \mathcal{A}_{\text{CMP}}$ ist also selbst Σ_o-vollständig. Damit formt $(\mathcal{A}_{\text{CMP}}, \cup)$ einen vollständigen oberen Halbverband. Folglich ist die Existenz der supremalen Σ_o-vollständigen Teilsprache einer gegebenen Spezifikation $E \subseteq \Sigma^*$, in Zeichen

$$< E >^{\uparrow(\text{CMP})} := \sup\{K \subseteq E \,|\, K \in \mathcal{A}_{\text{CMP}}\},$$

gesichert.

Vorsilbencharakterisierung

Die Eigenschaft Σ_o-Vollständigkeit ist lediglich anhand des Vorsilbenabschlusses $\text{pre}\, K$ einer Sprache K festgemacht. Zur Verifikation der Eigenschaft

(B2) $\quad K \in \mathcal{A}_{\text{CMP}} \quad \Leftrightarrow \quad \text{pre}\, K \in \mathcal{A}_{\text{CMP}}$.

folge man der Argumentation zur Verifikation der Eigenschaft (A1) im Abschnitt 2.3.1.

Fixpunktcharakterisierung supremaler Elemente

Zu einer Sprache $K \subseteq \Sigma^*$ definiere man den Operator Ω_{CMP} gemäß

$$\Omega_{\text{CMP}} K := \{ t \in K \,|\, (\forall s \leq t)[s(\Sigma - \Sigma_o)^*\Sigma_o \cap \text{pre}\, K \neq \emptyset] \}. \tag{2.20}$$

Die Aufgabe des Operators Ω_{CMP} ist die Entfernung aller Zeichenketten, die mit der Σ_o-Vollständigkeit der Sprache K konfligieren.

Aus der Definition des Operators Ω_{CMP}, Gl. (2.20), folgt, dass das Bild einer Sprache K selbst vollständig in K enthalten ist, in Zeichen $\Omega_{\mathrm{CMP}} K \subseteq K$. Dabei gilt Gleichheit genau dann, falls K bereits Σ_{o}-vollständig ist, d. h. falls $K \in \mathcal{A}_{\mathrm{CMP}}$.

Die Sprachenfolgen $(K_i)_{i\in\mathbb{N}_0}$, gegeben durch die Iteration

$$K_0 := E\,, \quad K_{\mathrm{i}+1} := \Omega_{\mathrm{CMP}} K_{\mathrm{i}}, \tag{2.21}$$

ist wegen $\Omega_{\mathrm{CMP}} K_{\mathrm{i}} \subseteq K_{\mathrm{i}}$, für alle $i \in \mathbb{N}_0$, monoton fallend und strebt definitionsgemäß gegen den Grenzwert $K_\infty := \cap_{i\in\mathbb{N}_0} K_i$. Die nachfolgende Proposition charakterisiert $< E >^{\uparrow(\mathrm{CMP})}$ als Fixpunkt der Folge $(K_i)_{i\in\mathbb{N}_0}$.

Proposition 2.3.4. Zu einer Spezifikation $E \subseteq \Sigma^*$ betrachte man die durch Iteration (2.13) definierte Folge $(K_i)_{i\in\mathbb{N}_0}$ mit dem Grenzwert $K_\infty := \cap_{i\in\mathbb{N}_0} K_i$. Die supremale Σ_{o}-vollständige Teilsprache $< E >^{\uparrow(\mathrm{CMP})}$ der Spezifikation E ist durch den Grenzwert K_∞ der Folge $(K_i)_{i\in\mathbb{N}_0}$ genau dann gegeben, falls K_∞ ein Fixpunkt des Operators Ω_{CMP} ist, in Zeichen

$$K_\infty = < E >^{\uparrow(\mathrm{CMP})} \quad \Leftrightarrow \quad K_\infty = \Omega_{\mathrm{CMP}} K_\infty.$$

□

Beweis. Man zeige zunächst durch Induktion über die Folgenglieder K_{i}, dass für alle $i \in \mathbb{N}_0$

$$< E >^{\uparrow(\mathrm{CMP})} \subseteq K_i\,. \tag{2.22}$$

gilt. Zuvorderst bemerke man, dass wegen (B1) $< E >^{\uparrow(\mathrm{CMP})}$ tatsächlich Σ_{o}-vollständig ist. Für $i = 0$ ist $K_0 = E$. Es ist $< E >^{\uparrow(\mathrm{CMP})}$ die supremale Teilsprache von E, sodass die Behauptung offensichtlich erfüllt ist. Man nehme an, für ein beliebiges $i \in \mathbb{N}_0$ gelte die Annahme $< E >^{\uparrow(\mathrm{CMP})} \subseteq K_i$ und man zeige, dass daraus auch $< E >^{\uparrow(\mathrm{CMP})} \subseteq K_{i+1}$ folgt. Dazu wähle man $s \in \mathrm{pre} < E >^{\uparrow(\mathrm{CMP})}$ beliebig. Es ist $< E >^{\uparrow(\mathrm{CMP})}$ die supremale Σ_{o}-vollständige Teilsprache der Spezifikation E. Wegen (B2) sind mit $< E >^{\uparrow(\mathrm{CMP})}$ auch alle ihre Vorsilben $\mathrm{pre} < E >^{\uparrow(\mathrm{CMP})}$ Σ_{o}-vollständig. Es existiert also ein $t \in (\Sigma - \Sigma_{\mathrm{o}})^*\Sigma_{\mathrm{o}}$ mit $st \in \mathrm{pre} < E >^{\uparrow(\mathrm{CMP})}$ und folglich ist $s(\Sigma - \Sigma_{\mathrm{o}})^*\Sigma_{\mathrm{o}} \cap (\mathrm{pre} < E >^{\uparrow(\mathrm{CMP})}) \neq \emptyset$. Gemäß der Induktionsannahme $< E >^{\uparrow(\mathrm{CMP})} \subseteq K_i$ und der Monotonie des Vorsilbenoperators gilt auch $\mathrm{pre} < E >^{\uparrow(\mathrm{CMP})} \subseteq \mathrm{pre}\, K_{\mathrm{i}}$. Man erhält $s(\Sigma - \Sigma_{\mathrm{o}})^*\Sigma_{\mathrm{o}} \cap \mathrm{pre}\, K_i \neq \emptyset$. Wegen $s(\Sigma - \Sigma_{\mathrm{o}})^*\Sigma_{\mathrm{o}} \cap (\mathrm{pre}\, K_{\mathrm{i}}) \neq \emptyset$ ist $s \in \Omega_{\mathrm{CMP}} K_{\mathrm{i}} = K_{\mathrm{i}+1}$. Es kann s' als Vorsilbe von st beliebig gewählt werden, sodass st selbst die Implikation in Definition (2.20) erfüllt. Folglich gilt $st \in \Omega_{\mathrm{CMP}} K_{\mathrm{i}} = K_{\mathrm{i}+1}$, was den Beweis von Gl. (2.22) abschließt.

Um zu zeigen, dass aus $< E >^{\uparrow(\mathrm{CMP})} = K_\infty$ auch $\Omega_{\mathrm{CMP}} K_\infty = K_\infty$ folgt, erinnere man, dass $< E >^{\uparrow(\mathrm{CMP})}$ selbst Σ_o-vollständig ist. Folglich ist auch K_∞ Σ_o-vollständig und aus der Definition des Operators Ω_{CMP} folgt $\Omega_{\mathrm{CMP}} K_\infty = K_\infty$. Umgekehrt nehme man an, K_∞ sei ein Fixpunkt von Ω_{CMP} und zeige $K_\infty =< E >^{\uparrow(\mathrm{CMP})}$. Dann ist K_∞ selbst Σ_o-vollständig und eine Teilmenge von E, also gilt $K_\infty \subseteq < E >^{\uparrow(\mathrm{CMP})}$. Andererseits gilt $< E >^{\uparrow(\mathrm{CMP})} \subseteq K_i$ für alle $i \in \mathbb{N}_0$, sodass auch für den Durchschnitt aller Folgenglieder $K_\infty = \cap_{i\in\mathbb{N}_0} K_i$ selbst $< E >^{\uparrow(\mathrm{CMP})} \subseteq K_\infty$ gilt. q.e.d.

Endliche Konvergenz für reguläre Parameter

Falls E regulär ist, folgt aus der nächsten Proposition die endliche Konvergenz der Folge $(K_i)_{i\in\mathbb{N}_0}$, gegeben durch die Iteration 2.21 gegen ihren Grenzwert K_∞.

Proposition 2.3.5. Zu einer Spezifikation $E \subseteq \Sigma^*$ betrachte man die durch Iteration (2.21) definierte Folge $(K_i)_{i\in\mathbb{N}_0}$ mit dem Grenzwert $K_\infty := \cap_{i\in\mathbb{N}_0} K_i$. Falls E regulär ist, dann erhält Ω_{CMP} die Regularität. Zudem existiert $k \in \mathbb{N}_0$, sodass $K_\infty = K_k$ gilt.

Beweis. Die folgende Beziehung zeigt, dass eine uniforme Schranke für den kleinsten Zustandsraum zur Realisierung der Iterate K_i, $i \in \mathbb{N}_0$ existiert: für beliebiges $i \in \mathbb{N}_0$ und beliebige s', $s'' \in K_i$ wird die Implikation

$$s',\ s'' \in \mathrm{pre}\, K_i,\ s' \equiv_E s'' \quad \Rightarrow \quad s' \equiv_{K_i} s''. \tag{2.23}$$

gezeigt. Für $i = 0$ folgt $K_0 = E$ und die obige Implikation ist offensichtlich wahr. Für einen Induktionsbeweis nehme man an, die Implikation gilt für $i \in \mathbb{N}_0$ und betrachte beliebige s', $s'' \in \mathrm{pre}\, K_{i+1}$ mit $s' \equiv_E s''$. Gemäß der Induktionsannahme und $K_{i+1} \subseteq K_i$ folgt $s' \equiv_{K_i} s''$. Man wähle ein beliebiges t mit $s't \in K_{i+1}$. Insbesondere gilt $s't \in K_i$ und deshalb gilt $s''t \in K_i$. Daraus folgt $s't(\Sigma - \Sigma_o)^*\Sigma_o \cap \mathrm{pre}\, K_i \neq \emptyset$ und daraus die Existenz eines $r\sigma \in (\Sigma - \Sigma_o)^*\Sigma_o$ mit $s'tr\sigma \in \mathrm{pre}\, K_i$. Mit $s' \equiv_{K_i} s''$, erhält man $s''tr\sigma \in \mathrm{pre}\, K_i$ und damit auch $s''t(\Sigma - \Sigma_o)^*\Sigma_o \cap \mathrm{pre}\, K_i \neq \emptyset$. Dies schließt den Beweis von $s''t \in K_{i+1}$ ab. Durch Variation von t erhält man die Implikation $s't \in K_{i+1} \Rightarrow s''t \in K_{i+1}$, und uniforme Substitution liefert die umgekehrte Implikation. Also ist $s' \equiv_{K_{i+1}} s''$, und die Implikation (2.23) folgt durch Induktion.

Mit Implikation (2.23) können die Behauptungen wörtlich wie in dem Beweis von Proposition 2.3.1 abgeleitet werden. q.e.d.

Realisierung im Zustandsraum

Für die folgende Diskussion nehme man an, dass E regulär ist und die trimme Realisierung $H = (X, \Sigma, \xi, x_o, X_m)$ besitzt, d. h., $L_m(H) = E$ und $L(H) = \mathrm{pre}\, E$. Gemäß Proposition 2.3.4 ergibt jede Implementierung von Ω_{CMP} in Kombination mit Iteration (2.21) eine praktische Prozedur zur Berechnung der supremalen Σ_o-vollständigen Teilsprache $< E >^{\uparrow(CMP)}$ der Spezifikation E. Bezieht man sich allerdings auf den Rahmen in [MBY$^+$12], dann kann die Berechnung von $< E >^{\uparrow(CMP)}$ mit Hilfe der Iteration

$$\begin{aligned} &N_0 := E, \\ &N_{i+1} :=< \mathrm{pre}\, N_i >^{\uparrow(CMP)} \cap E \end{aligned} \tag{2.24}$$

durchgeführt werden. Hier wird zwar $< \mathrm{pre}\, N_i >$ durch die Iteration (2.21) ausgewertet, allerdings ist es ausreichend, den Operator Ω_{CMP} für abgeschlossene Sprachen umzusetzen. Der Schnitt mit E kann dann durch Produktkomposition realisiert werden. Die Charakterisierung von $< E >^{\uparrow(CMP)}$ als Fixpunkt der Iteration (2.24) ist dann eine Konsequenz des Satzes 5 in [MBY$^+$12]. Kann zudem

(B3) Für einen beliebigen Automaten H, existiert ein G_{CMPL} mit $L(G_{CMPL}) = \Sigma^*$, sodass

$$(\forall F \sqsubseteq G_{CMPL} \times H \exists F^{\uparrow} \sqsubseteq G_{CMPL} \times H)[L(F^{\uparrow}) =< L(F) >^{\uparrow(CMPL)}]$$

gilt.

nachgewiesen werden, dann ist die endliche Konvergenz der Iteration (2.24) eine Konsequenz des Satzes 7 in [MBY$^+$12].

Wir zeigen (B3) für einen beliebigen Generator $G_{CMPL} = (Q, \Sigma, \delta, q_o, Q_m)$ mit $L(G_{CMPL}) = \Sigma^*$. Gegeben sei $F \sqsubseteq G_{CMP} \times H$; man konstruiere $F' \sqsubseteq F$ mit $L(F') = \Omega_{CMP} L(F)$ und folgere (B3) als Konsequenz von Proposition 2.3.4.

Für einen beliebigen Teilautomaten $F = (Z, \Sigma, \zeta, z_o, Z_m)$, $F \sqsubseteq G_{WSC} \times H$, betrachte man die Zustandsmengen Z_{LIVE}, in denen eine Zeichenkette, die mit einem Σ_o-Ereignis endet, ausgeführt werden kann, d. h.

$$Z_{LIVE} := \{(q,x) \in Q \times X \mid (\exists r \in (\Sigma - \Sigma_o)^* \Sigma_o)[\zeta((q,x), r)!]\}. \tag{2.25}$$

Deshalb definiere man $F' = (Z', \Sigma, \zeta', z_o, Z'_m) := F|_{Z_{LIVE}}$ als die Einschränkung von F auf Z_{LIVE}. Man bemerke, dass F' genau dann der leere Generator $G_\emptyset$ ist, wenn F der leere Generator ist oder $z_o \notin Z_{LIVE}$ gilt.

Proposition 2.3.6. In dem obigen Rahmen, in dem angenommen wird, dass $F \sqsubseteq G_{\text{CMP}} \times H$ gilt, resultiert die Konstruktion von $F' \sqsubseteq F$ in $\mathrm{L}(F') = \Omega_{\text{CMP}}\mathrm{L}(F)$. □

Beweis. Falls F der leere Generator ist, dann gilt das auch für F' und die Behauptung ist wahr. Von jetzt an wird angenommen, dass F nicht der leere Generator ist und setzen $K := \mathrm{L}(F) \neq \emptyset$.

Man wähle ein beliebiges $s \in \Omega_{\text{CMP}}K$ und zeige $s \in \mathrm{L}(F')$. Gemäß der Definition des Operators Ω_{CMP} existiert ein $r \in (\Sigma - \Sigma_{\text{o}})^*\Sigma_{\text{o}}$ mit $sr \in \mathrm{pre}\,K = \mathrm{L}(F)$. Insbesondere ist für $(q,x) = \zeta(z_0, sr)$ auch $\zeta((q,x), r)!$. Daraus folgt $(q,x) \in Z_{\text{LIVE}}$ und weiter $s \in \mathrm{L}(F|_{Z_{\text{LIVE}}}) = \mathrm{L}(F')$.

Für die rückwärtige Einschließung wähle man ein beliebiges $s \in \mathrm{L}(F')$ und zeige $s \in \Omega_{\text{CMP}}K$. Wegen $F' = F|_{Z_{\text{LIVE}}}$ gilt für ein $(q,x) \in Z_{\text{LIVE}}$ auch $\zeta(z_0, s) = (q,x)$. Wegen $(q,x) \in Z_{\text{LIVE}}$ existiert ein $r \in (\Sigma - \Sigma_{\text{o}})^*\Sigma_{\text{o}}$ mit $\zeta((q,x), r)!$. Insbesondere ist $sr \in \mathrm{pre}\,K = \mathrm{L}(F)$, weshalb auch $s \in \Omega_{\text{CMP}}K$ gilt. q.e.d.

Dies schließt die Verifikation von (B3) ab. Eine Prozedur zur Berechnung von $< E >^{\uparrow(\text{CMP})}$ $\mathrm{pre}\,E$ kann aus der obigen Iteration über Automaten aufgebaut werden. Man folge dazu der Argumentation wie bei der schwachen Sensorkonsistenz.

2.3.3 Supremaler rekonfigurierender Regelkreis

Auf Basis der Diskussion supremaler schwach sensorkonsistenter bzw. Σ_{o}-vollständiger Teilsprachen in den vorhergehenden Abschnitten 2.3.1 und 2.3.2 wird nachfolgend $K^{\uparrow}$ als Fixpunkt der Verknüpfung von Operatoren zur Berechnung der supremalen Teilsprache bzgl. der gewünschten Eigenschaften (M1) bis (M5) dargestellt. Die verbleibenden Eigenschaften (M6) und (M7) führen auf die Spezifikation $K \subseteq E := E_{\text{F}} \parallel E_{\text{R}} \parallel L_{\text{V}}$.

Für eine formale Diskussion bezeichnen

$$\mathcal{A}_{\text{CNTR}} := \{K \subseteq \Sigma^* \,|\, K \text{ erfüllt die Steuerbarkeitsbedingung (M1)}\}, \tag{2.26}$$

$$\mathcal{A}_{\text{PNRM}} := \{K \subseteq \Sigma^* \,|\, K \text{ erfüllt Vorsilbennormalität (M2)}\} \tag{2.27}$$

die Mengen aller steuerbaren bzw. vorsilbennormalen Sprachen. Wie $\mathcal{A}_{\text{CMP}}$ und $\mathcal{A}_{\text{WSC}}$ aus den Abschnitten 2.3.1 und 2.3.2 formen die obigen Mengen vollständige obere Halbverbände. Insbesondere weist das individuelle, supremale Element einer gegebenen Spezifi-

kation E die entsprechenden Eigenschaften auf, d. h.

$$< E >^{\uparrow(\mathrm{CNTR})} := \sup\{K \subseteq E \mid K \in \mathcal{A}_{\mathrm{CNTR}}\} \in \mathcal{A}_{\mathrm{CNTR}}\,, \tag{2.28}$$

$$< E >^{\uparrow(\mathrm{PNRM})} := \sup\{K \subseteq E \mid K \in \mathcal{A}_{\mathrm{PNRM}}\} \in \mathcal{A}_{\mathrm{PNRM}}\,. \tag{2.29}$$

Zudem bildet die Schnittmenge

$$\mathcal{A} := \mathcal{A}_{\mathrm{CNTR}} \cap \mathcal{A}_{\mathrm{PNRM}} \cap \mathcal{A}_{\mathrm{CMP}} \cap \mathcal{A}_{\mathrm{WSC}} \tag{2.30}$$

ebenfalls einen vollständigen oberen Halbverband. Sie repräsentiert das minimal-restriktive geschlossene Regelkreisverhalten mit den Eigenschaften (M1) bis (M5). Deshalb erfüllt das minimal-restriktive Regelkreisverhalten

$$K^{\uparrow} := \sup\{K \subseteq E \,|\, K \in \mathcal{A}\} \tag{2.31}$$

selbst die Eigenschaften (M1) bis (M7). Zur Berechnung von $K^{\uparrow}$ definiere man für ein $K \subseteq \Sigma^*$ den Operator

$$\Omega(K) := [<>^{(\uparrow\mathrm{CNTR})} \circ <>^{(\uparrow\mathrm{PNRM})} \circ <>^{(\uparrow\mathrm{WSC})} \circ <>^{(\uparrow\mathrm{WSC})}](\mathrm{pre}\,K) \cap E$$

und bemerke, dass mit den beteiligten Operatoren $<>^{(\uparrow\mathrm{CNTR})}$, $<>^{(\uparrow\mathrm{PNRM})}$, $<>^{(\uparrow\mathrm{WSC})}$ und $<>^{(\uparrow\mathrm{WSC})}$, auch Ω selbst monoton ist, in Zeichen $\Omega K \subseteq K$.

Aufgrund der Monotonieeigenschaft des Operators Ω konvergiert die Sprachenfolge $(K_i)_{i\in\mathbb{N}_0}$, definiert durch die Iteration

$$\begin{aligned} K_0 &:= E \\ K_{i+1} &= \Omega K_i, \end{aligned} \tag{2.32}$$

gegen den Grenzwert $K_\infty := \cap_{i\in\mathbb{N}_0} K_i$. Der Grenzwert K_∞ stellt einen Kandidaten für den gesuchten optimalen Regelkreis $K^{\uparrow}$ dar.

Satz 2.3.1. Zu einem Rekonfigurationsproblem $(\Sigma, L_N, E_N, L_F, E_F, E_R)$ setze man $E := E_F \parallel E_R \parallel L_V$ und betrachte die Sprachenfolge $(K_i)_{i\in\mathbb{N}_0}$, definiert durch die Iteration (2.32), mit dem Grenzwert $K_\infty := \cap_{i\in\mathbb{N}_0} K_i$. Weiter bezeichnet $K^{\uparrow} \subseteq \Sigma^*$ die supremale Teilsprache der Spezifikation $E_F \parallel E_R$ bzgl. der Eigenschaften (M1) bis (M7). Es ist $K^{\uparrow} = K_\infty$, falls K_∞ ein Fixpunkt des Operators Ω ist, in Zeichen

$$K_\infty = K^{\uparrow} \quad \Leftrightarrow \quad K_\infty = \Omega K_\infty.$$

Sind zudem alle Parameter des Rekonfigurationsproblems regulär, dann konvergiert die Iteration (2.32) nach endlich vielen Iterationen, d. h. es existiert ein $k \in \mathbb{N}_0$ mit $k < \infty$, sodass $K_k = K_{k+1} = K_\infty$ gilt. □

Beweis. Man bemerke, dass die Iteration (2.32) ein Spezialfall der Iteration 2 in [MBY$^+$12] ist. Darin etabliert Satz 7 endliche Konvergenz gegen $K^\uparrow$. Um Satz 7 anwenden zu können, ist zu zeigen, dass ein Zustandsraum $G_\mathcal{A}$ existiert, auf dem simultan die supremalen Teilsprachen bzgl. der Eigenschaften (M1), (M2), (M4) und (M5) realisiert werden können, d. h. es ist die Eigenschaft (A4) erfüllt. Dies ist gleichbedeutend mit der Wahl eines Zustandsraums, sodass simultan die Eigenschaft (A3) für die Eigenschaften (M1), (M2), (M3) und (M5) erfüllt ist. Da Vorsilbennormalität die einzige Eigenschaft ist, bei der die Wahl des Zustandsraums in (A3) von H abhängt, konstruiere man $G_\mathcal{A}$ uniform wie folgt:

1. Es seien G_{CTRL}, G_{CMPL} und G_{WSC} Generatoren, die (A3) für die entsprechenden Eigenschaften (M1), (M4) und (M5) individuell erfüllt sind;
2. Man setze $G' = G_{\text{CTRL}} \times G_{\text{CMPL}} \times G_{\text{WSC}}$ und bemerke, dass G' die Implikation in (A3) uniform für (M1), (M4) und (M5) erfüllt;
3. Es sei R_{obs} ein Beobachter für $G' \times H$ gemäß [CM89], erweitert um einen Dumpstate und entsprechende Transitionen, sodass $\mathrm{L}(R_{\text{obs}}) = \Sigma^*$ gilt;
4. letztlich sei $G_\mathcal{A} = G' \times R_{\text{obs}}$.

Gemäß den Ergebnissen in [CM89], erfüllt $G_\mathcal{A}$ die Implikation in (A3) für Vorsilbennormalität. Für die verbleibenden Eigenschaften schreibe man die Spezifikation als $H' = H \times R_{\text{obs}}$, um die Implikation in (A3) zu erfüllen. Dies schließt den Beweis der endlichen Konvergenz gegen $K^\uparrow$ ab. q.e.d.

Ergebnisstand. Der Entwurf universeller Rekonfiguratoren wurde zunächst für Systeme mit trivialen Letztendlichkeitseigenschaften diskutiert. Es wurden mit der Steuerbarkeit, relativen Abgeschlossenheit, Vorsilbennormalität, Vollständigkeit, schwacher Sensorkonsistenz, Nominalstrecken- und Spezifikationseinschluss eine Reihe von Eigenschaften an einen Regelkreiskandidaten gestellt, aus dem ein universeller Rekonfigurator durch Projektion und Vorsilbenbildung gewonnen werden kann. Weist die fehlerverträgliche Strecke nicht-triviale Letztendlichkeitseigenschaften auf, kann mit Hilfe einer geeigneten Rückwärtserreichbarkeitsanalyse die Konfliktfreiheit des rekonfigurierenden Regelkreises verifiziert werden. Abschließend wurde der Rekonfiguratorentwurf im Zustandsraum diskutiert. Als Regelkreiskandidat diente die supremale Teilsprache der Spezifikation bzgl. der bereits genannten Eigenschaften.

Kapitel 3

Praktische Umsetzung

In diesem Kapitel wird die praktische Umsetzung der fehlerverträglichen Regelung und der fehlerverdeckenden Steuerungsrekonfiguration im Rahmen einer Machbarkeitsstudie vorgestellt. Für Experimente stand mit der *Smart*Automation-Anlage (SmA) eine Testumgebung bereit, in der der Industriepartner unter realistischen Bedingungen die Anwendbarkeit neuartiger Methoden testet. Können die fehlerverträgliche Regelung und die fehlerverdeckende Steuerungsrekonfiguration an der SmA erfolgreich umgesetzt werden, darf daraus geschlossen werden, dass auch die Anwendung in der industriellen Praxis konzeptionell möglich ist.

Als Anwendungsszenario wurde mit der *Kollisionsfreien Zusammenführung zweier Materialflüsse* ein Beispiel mit einfacher ereignisdiskreter Dynamik gewählt. Trotzdem hat der Reglerentwurf gezeigt, dass der Steuerbarkeitsbegriff (R3), siehe Abschnitt 1.5.1, für die vorliegende Anwendung zu streng ist. Im ersten Teil dieses Kapitels wird deshalb der bestehende durch einen allgemeineren und weniger restriktiven Steuerbarkeitsbegriff ersetzt. Anschließend wird auf die Integration fehlertoleranter Regler in bestehende Steuerungsarchitekturen eingegangen. Den wesentlichen Teil dieses Kapitels stellt der Reglerentwurf am Beispiel dar.

3.1 Pragmatischer Rekonfiguratorentwurf

Für die vorliegende Anwendung hat sich gezeigt, dass der Steuerbarkeitsbegriff (R3) in Abschnitt 1.5.1 zu streng ist.

Beispiel 3.1.1. Eine einfache Maschine führt nacheinander zwei verschiedene Prozesse aus und meldet deren Ende mit den nicht-steuerbaren Ereignissen a und b. Mit dem Auf-

treten eines Fehlers fällt der zweite Prozess aus. Der minimal-restriktive Nominalregler schränkt die Strecke in ihrem Verhalten nicht ein. Abbildung 3.1.1 zeigt das fehlerverträgliche Streckenmodell L_F (links) und den virtualisierten Nominalregler H_V (rechts).

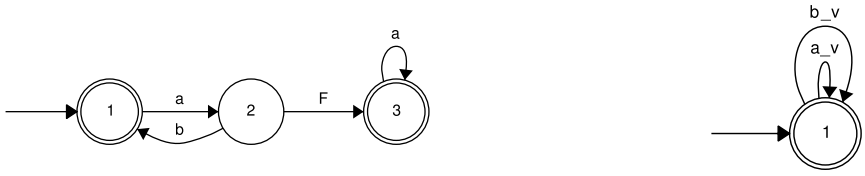


Abbildung 3.1.1: Fehlerverträgliche Strecke L_F und virtualisierter Nominalregler H_V

Die formale Strecke $L_F \parallel H_V$ soll nun den Inaktivitätsbedingungen E_R, gemäß Gl. 2.1, siehe Abbildung 3.1.2, genügen.

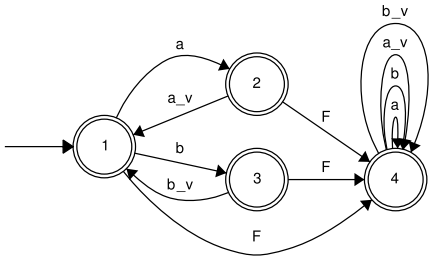


Abbildung 3.1.2: Inakativitätsbedingungen

Generiert die Maschine das Ereignis *a*, befindet sich sowohl die Streckenkomponente L_F als auch die Spezifikation E_R in Zustand 2. In diesem Zustand verbietet die Spezifikation das Auftreten des Ereignisses *b*, während es die Strecke erlaubt. Die supremale Teilsprache der Spezifikation bzgl. des Steuerbarkeitsbegriffs (M1) ist dann die leere Menge.

In einer praktischen Anwendung werden die virtuellen Ereignisse in Software ausgeführt, sodass das System quasi ohne Zeitverlust einen Zustandswechsel ausführt. Demnach verstreicht keine Zeit zwischen den Ereignissen *a* und *a_v*. In einer realen Anwendung wird das Ereignis *b* nur für eine vernachlässigbar kleine Spanne verboten. Der Steuerbarkeitsbegriff (M1) erweist sich daher als zu streng. □

Unter Berücksichtigung des Verstreichens physikalischer Zeit kann ein, im Vergleich zu [RW87, RW89], allgemeinerer Steuerbarkeitsbegriff entwickelt werden, siehe [BW94]. Er führt auf eine Steuerungstechnologie, in der das Verbieten eines Ereignisses an das Verstreichen von Zeit und die Möglichkeit, das Auftreten eines alternativen Ereignisses erzwingen zu können, gebunden ist.

In [BW94] wird das Gesamtalphabet $\Sigma = \Sigma_c \dot\cup \Sigma_{uc}$ um ein *„tick"-Ereignis* τ zu

$$\hat{\Sigma} := \Sigma_c \dot\cup \Sigma_{uc} \dot\cup \{\tau\} \tag{3.1}$$

erweitert. Das Ereignis τ modelliert das Ticken einer globalen Uhr. Zudem werden *erzwingbare Ereignisse* Σ_{for} ausgezeichnet, die anstelle eines τ-Ereignisses erzwungen werden können. Außerdem darf eine zeitbewertete Strecke $L \subseteq \hat{\Sigma}^*$ keine zyklischen Ereignisketten aufweisen, die kein τ-Ereignis enthalten, d. h. es muss

$$(\forall s \in \text{pre}\, L, t \in \hat{\Sigma}^*)[st \equiv_{\text{L}} s \Rightarrow t \notin \Sigma^*] \tag{3.2}$$

gelten.

Steuerbarkeit bei Zeitbewertung. Gegeben seien zwei Sprachen $L, K \subseteq \Sigma^*$. Es heißt K *bei Zeitbewertung steuerbar bzgl.* $(L, \Sigma_{\text{for}}, \Sigma_{\text{uc}})$, falls für alle $s \in \text{pre}\, K$

$$\begin{aligned} &s\Sigma_{\text{for}} \cap \text{pre}\, K = \emptyset \Rightarrow s(\Sigma_{\text{uc}} \dot{\cup} \{\tau\}) \cap L \subseteq \text{pre}\, K \quad \vee \\ &s\Sigma_{\text{for}} \cap \text{pre}\, K \neq \emptyset \Rightarrow s\Sigma_{\text{uc}} \cap L \subseteq \text{pre}\, K \end{aligned} \tag{3.3}$$

gilt.

Für den Entwurf fehlerverdeckender Rekonfiguratoren wird davon ausgegangen, dass allen nicht-steuerbaren Ereignissen Σ_{UCON} und virtuellen steuerbaren Ereignissen $\Sigma_{\text{CON,V}}$ ein τ-Ereignis vorausgeht. Zudem sind alle steuerbaren Ereignisse Σ_{CON} und virtuellen nichtsteuerbaren Ereignisse erzwingbar. Die verbleibenden Ereignisse, Σ_{HI} und Σ_{LO}, werden als nicht-steuerbar angenommen. Entsprechend ergeben sich

$$\begin{aligned} \Sigma_{\text{for}} &= \Sigma_{\text{CON}} \dot{\cup} \Sigma_{\text{UCON,V}} \\ \Sigma_{\text{uc}} &= \Sigma_{\text{LO}} \dot{\cup} \Sigma_{\text{HI}} \end{aligned}$$

und der Steuerbarkeitsbegriff (R3) in der Definition des Rekonfigurationsproblems, Definition 1.5.1 Abschnitt 1.5.1, wird durch

(R3') Steuerbarkeit bei Zeitbewertung bzgl. $(L_{\text{F}} \parallel H_{\text{V}}, \Sigma_{\text{for}}, \Sigma_{\text{uc}})$, d. h.
$(\forall s \in (\text{pre}\, L_{\text{F}} \parallel R \parallel H_{\text{V}})[$
$s\Sigma_{\text{for}} \cap (\text{pre}\, L_{\text{F}}) \parallel R \parallel H_{\text{V}} = \emptyset \Rightarrow s(\Sigma_{\text{uc}} \dot{\cup} \{\tau\}) \cap (\text{pre}\, L_{\text{F}}) \parallel H_{\text{V}} \subseteq (\text{pre}\, L_{\text{F}}) \parallel R \parallel H_{\text{V}} \vee$
$s\Sigma_{\text{for}} \cap (\text{pre}\, L_{\text{F}}) \parallel R \parallel H_{\text{V}} \neq \emptyset \Rightarrow s\Sigma_{\text{uc}} \cap (\text{pre}\, L_{\text{F}}) \parallel H_{\text{V}} \subseteq (\text{pre}\, L_{\text{F}}) \parallel R \parallel H_{\text{V}}$
$]$

ersetzt.

Ist ein Kandidat K mit (M1) und (M2) sowie

(M3') Steuerbarkeit bei Zeitbewertung bzgl. $(L_{\text{F}} \parallel H_{\text{V}}^{\uparrow}, \Sigma_{\text{for}}, \Sigma_{\text{uc}})$, d. h.
$(\forall s \in \text{pre}\, K[$
$s\Sigma_{\text{for}} \cap \text{pre}\, K = \emptyset \Rightarrow s(\Sigma_{\text{uc}} \dot{\cup} \{\tau\}) \cap (\text{pre}\, L_{\text{F}}) \parallel H_{\text{V}}^{\uparrow} \subseteq \text{pre}\, K \vee$
$s\Sigma_{\text{for}} \cap \text{pre}\, K \neq \emptyset \Rightarrow s\Sigma_{\text{uc}} \cap (\text{pre}\, L_{\text{F}}) \parallel H_{\text{V}}^{\uparrow} \subseteq \text{pre}\, K$
$]$

gegeben, so kann man zeigen, dass für beliebige virtualisierte Lösungen $H_V \subseteq \Sigma^*_{C,V}$ des Nominalentwurfsproblems (Σ_N, L_N, E_N) die Steuerbarkeitsbedingung (R3') des resultierenden selbst-rekonfigurierenden Regelkreises $L_F \parallel R \parallel H_V$ erfüllt ist. Weiter kann nachgewiesen werden, dass (M3') unter beliebiger Vereinigung erhalten bleibt.

Auf Grund der strukturellen Ähnlichkeit der Aussagen (M3) und (M3') wird erwartet, dass die Berechnung der supremalen Teilsprache einer gegebenen Spezifikation E bzgl. der Eigenschaften (M1)-(M3') sowie (M4)-(M7) gemäß [MBY+12] erfolgen kann. Ein entsprechendes Iterationsschema wurde in verschiedenen Beispielen getestet. Seine Softwareumsetzung terminierte nach endlich vielen Schritten und die praktische Anwendung der entworfenen Rekonfiguratoren war erfolgreich.

3.2 Umsetzung fehlertoleranter Steuerungsstrategien

Sowohl fehlerverträgliche Regler als auch fehlerverdeckende Rekonfiguratoren sind für die Eingliederung in eine bestehende Steuerungsarchitektur gedacht. Durch die Bedienereignisse werden abstrakte Schnittstellen zwischen der bestehenden Nominalsteuerung und der bestehenden Steuerungsarchitektur bereitgestellt. Für eine praktische Umsetzung müssen softwaretechnisch entsprechende Schnittstellen zwischen dem bestehenden Steuerungsprogramm und einem modellbasierten Regler geschaffen werden. Außerdem erfordert die praktische Umsetzung modellbasierter Regler die Ableitung geeigneten Steuerungscodes aus einem gegebenen Reglermodell.

3.2.1 Umsetzung ereignisdiskreter Reglerdynamik

Ein formales Reglermodell legt lediglich die erlaubte Abfolge abstrakter Ereignisse fest. Insbesondere fehlt Information über die genaue zeitliche Ereignisabfolge, über die physikalische Interpretation des Auftretens eines abstrakten Ereignisses sowie über die Auflösung von eventuellen Mehrdeutigkeiten und gleichzeitig erkannter Ereignisse. Das Fehlen dieser Information wird durch das Bereitstellen eines Regelsatzes, nach dem die Ereignisausführung zu erfolgen hat, ausgeglichen. In der Literatur werden dazu verschiedene Ansätze vorgestellt, siehe z.B. [FH98, BC07, LW, TMP10]. In dieser Arbeit wird auf den Regelsatz in [TMP10] zurückgegriffen.

In [TMP10] werden Anordnungen betrachtet, die einer Regelung ausschließlich über Sensoren und Aktuatoren zugänglich sind. Ereignisse, die sich auf die Ansteuerung von

Aktuatoren beziehen, werden *Ausgangsereignisse* genannt. Dagegen heißen Ereignisse, die sich auf die Auswertung von Sensoren beziehen, *Eingangsereignisse*. Ist ein Regler $H \subseteq \Sigma_C^*$ gegeben, wird das Alphabet Σ_C in Eingangsereignisse Σ_{IN} und Ausgangsereignisse Σ_{OUT} partitioniert. Jedem Eingangsereignis wird zusätzlich ein Sensortyp (steigende oder fallende Flanke) zugeordnet, während jedem Ausgangsereignis ein Kommandotyp (Setzen oder Rücksetzen des Prozessabbildeintrags) zugewiesen wird. Der Regelsatz in [TMP10] ist durch die Angabe eines unter Vorsilbenbildung abgeschlossenen Reglermodells und der Partitionierung des Regleralphabets in Eingangs- und Ausgangsereignisse sowie den entsprechenden Flanken- bzw. Kommandotypen vollständig parametriert. Optional kann die Auflösung von Mehrdeutigkeiten durch eine Priorisierung der Ereignisse gezielt beeinflusst werden.

Im Falle eines fehlerverträglichen Reglers liegt ein unter Vorsilbenbildung abgeschlossenes Reglermodell vor, siehe Def. 1.2.1. Das Regleralphabet $\Sigma_C = \Sigma_{HI} \dot\cup \Sigma_{CON} \dot\cup \Sigma_{UCON}$ wird in die Ausgangsereignisse $\Sigma_{OUT} := \Sigma_{CON}$ und die Eingangsereignisse $\Sigma_{IN} := \Sigma_{UCON}$ partitioniert. Die Bedienerereignisse Σ_{HI} nehmen eine Sonderrolle ein. Sie werden je nach Ereignissemantik den Ausgangs- oder Eingangsereignissen zugeschlagen.

Gemäß Def. 1.5.1 ist eine (universelle) Lösung eines Rekonfigurationsproblems eine unter Vorsilbenbildung abgeschlossene formale Sprache $R \subseteq \Sigma_R^*$. Für die Umsetzung eines Rekonfigurators wird wieder die Perspektive aus Kapitel 2 eingenommen; der Rekonfigurator dient als Koordinationsregler für den Nominalregler und einer fehlerbehafteten Strecke. Von diesem Standpunkt aus sind alle steuerbaren Ereignisse Σ_{CON} und alle virtuellen nicht-steuerbaren Ereignisse $\Sigma_{CON,V}$ Ausgangsereignisse, während alle nicht-steuerbaren Ereignisse Σ_{UCON} und alle virtuellen steuerbaren Ereignisse $\Sigma_{CON,V}$ Eingangsereignisse sind. Dementsprechend wird das Alphabet $\Sigma_R = \Sigma_{CON} \dot\cup \Sigma_{UCON} \dot\cup \Sigma_{CON,V} \dot\cup \Sigma_{UCON,V}$ gemäß $\Sigma_R = \Sigma_{IN} \dot\cup \Sigma_{OUT}$ mit $\Sigma_{IN} := \Sigma_{UCON} \dot\cup \Sigma_{CON,V}$ und $\Sigma_{OUT} := \Sigma_{CON} \dot\cup \Sigma_{UCON,V}$ partitioniert.

Obwohl denkbar, scheidet eine händische Umsetzung eines gegebenen Regelsatzes in Form eines Steuerungsprogramms auf Grund der hohen Komplexität der Reglermodelle aus. Stattdessen wird die automatisierte Erzeugung von Steuerungscode angestrebt. In Abschnitt 2.3 wurde gezeigt, dass die Entwurfsmethodik in Kapitel 2 auf reguläre Reglermodelle führen, solange Strecke und Spezifikation regulär sind. Am Lehrstuhl für Regelungstechnik der FAU Erlangen-Nürnberg wurden Quelltextgeneratoren zur Umsetzung des in [TMP10] festgelegten Regelsatzes auf Basis von endlichen Automatenrepräsentationen der Reglermodelle entwickelt, siehe [Sch11, Lac12].

3.2.2 Einbettung fehlertoleranter Regler

Konzeptionell ersetzen fehlerverträgliche Regler den Nominalregler, während die übrige Steuerungshierarchie unberührt bleibt. In einer praktischen Umsetzung kann das durch einfaches Überschreiben der nominellen Steuerungskommandos geschehen. Dagegen besitzen Rekonfiguratoren mit virtuellen Ereignissen eine definierte Schnittstelle zu dem Nominalregler. Um diese Schnittstelle zu realisieren, wird ein Speicherbereich mit der Größe des physischen Prozessabbildes, nachfolgend mit *virtuelles Prozessabbild* bezeichnet, ausgewiesen. Die Umbenennungsfunktion h gemäß Kapitel 1 kann dann durch das Ersetzen von Referenzen auf das physikalische Prozessabbild durch Referenzen auf das virtuelle Prozessabbild realisiert werden. Aus diesem Blickwinkel ist die Aufgabe des Rekonfigurators die Umrechnung von Änderungen im physischen Prozessabbild auf Änderungen im virtuellen Prozessabbild und umgekehrt.

Um die Funktionalität der bestehenden Steuerungshierarchie zu gewährleisten, müssen die Schnittstellen zu den umgebenden Hierarchieebenen genau nachgebildet werden. Insbesondere muss die Zuordnung von Speichereinträgen und abstrakten Bedienereignissen der angestrebten Schnittstellensemantik entsprechen. Von daher setzt die Einbettung fehlertoleranter Regler detailliertes Wissen über den Aufbau und die Aufgaben der beteiligten Steuerungsebenen im Einzelnen und der gesamten Steuerungsarchitektur im Ganzen voraus.

3.3 Fehlertolerante Zusammenführung zweier Materialflüsse

Im Rahmen der Industriekooperation stand die Versuchsanlage „*Smart*Automation“ (SmA) zu Testzwecken zur Verfügung. Die SmA stellt ein Modell eines Flaschenabfüllprozesses dar. Sie besteht aus verschiedenen Stationen, an denen die Flaschen befüllt, auf korrekte Befüllung getestet und wieder entleert werden, siehe Abbildung 3.3.1.

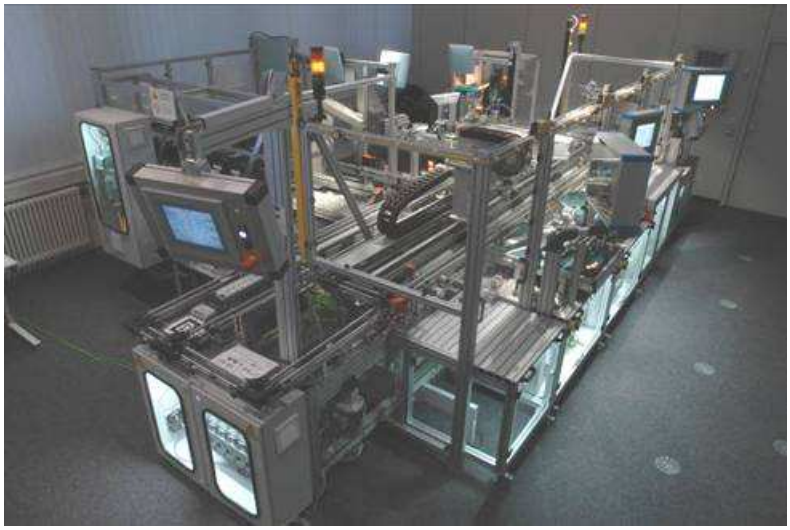


Abbildung 3.3.1: *Smart*Automation Anlage

Alle Stationen sind über ein Transportband verbunden. Dabei tritt eine Situation auf, in der Flaschen von verschiedenen Transportbandsegmenten zusammengeführt werden, siehe Abbildung 3.3.2.

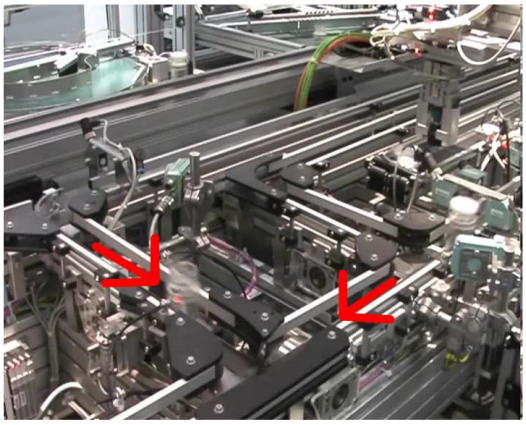


Abbildung 3.3.2: Prozessskizze: Materialflusszusammenführung

Abbildung 3.3.3 zeigt ein Prinzipbild der Situation in Abildung 3.3.2.

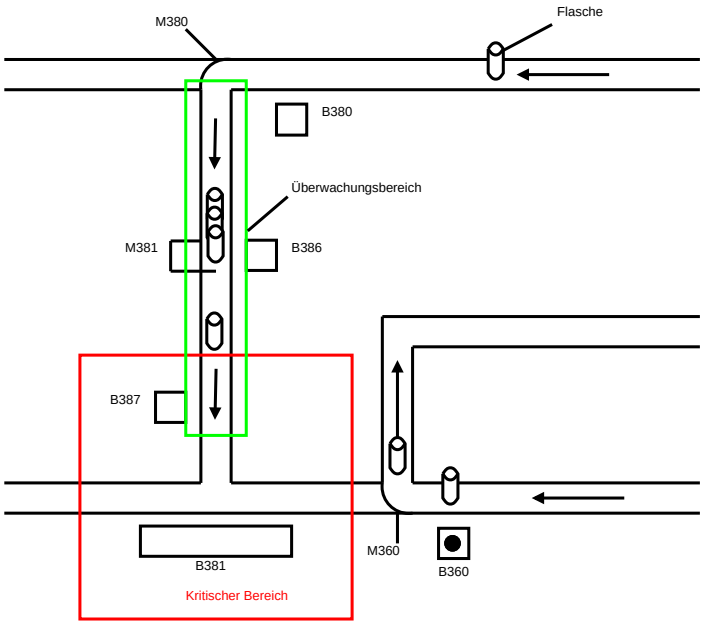


Abbildung 3.3.3: Prozessskizze: Materialflusszusammenführung

Information über den Systemzustand kann über die Sensoren B380, B386, B387 und B381 gewonnen werden. Eine gezielte Beeinflussung des Systems ist über die Weichen M380 und M360 und den Vereinzeler M381 möglich. Im Nominalbetrieb wird der Vereinzeler M381 gegen den Sensor B381 verriegelt, sodass keine Flasche den Vereinzeler passiert, solange der Sensor B381 belegt ist. Kollisionen im *kritischen Bereich* werden auf diese Weise verhindert.

Im Fehlerfall versagt die Verriegelung des Vereinzelers M381 gegen den Sensor B381. Flaschen können dann den Vereinzeler passieren, obwohl der kritische Bereich belegt ist; Kollisionen werden dann nicht mehr verhindert. Eine mögliche fehlertolerante Steuerungsstrategie besteht in der Verriegelung der Weiche M360, solange sich eine Flasche auf dem Querförderer (*Überwachungsbereich*) befindet.

In diesem Beispiel werden die Komponenten M360, M380 und M381 dezentral gesteuert und von einem übergeordneten Prozessleitsystem (FMS) koordiniert. Die Eingaben des FMS werden als Bedienerereignisse aufgefasst.

3.3.1 Fehlerverträgliche Regelung

Gegenstand des Entwurfs ist ein Regler für die Weiche M360. Dazu wird der gesamte Prozess in drei Komponenten unterteilt und modelliert: die Weiche M360 selbst, den kritischen Bereich und den Überwachungsbereich.

Modellbildung

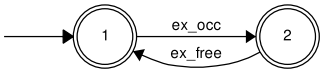
Die Auswirkungen eines Fehlers äußern sich im Verhalten des kritischen Bereichs, während die Weiche M360 und der Überwachungsbericht nicht beeinflusst werden.

Weiche M360. Die Weiche M360 kennt zwei Positionen. Anfänglich befindet sie sich in Grundposition. Ihre Dynamik ist durch den Wechsel von Grundposition zur Arbeitsposition (*M360_WP_SET*) und zurück (*M360_HP_SET*) bestimmt. Die Integration des fehlerverträglichen Reglers erfolgt nicht auf Prozessabbildebene. Stattdessen werden im bestehenden Quellcode die Variablen identifiziert, die für das Schalten der Weiche benützt werden. Sowohl für das Öffnen als auch für das Verriegeln werden unterschiedliche Variablen eingesetzt, an denen nur positive Flanken eine Wirkung zeigen. Entsprechend müssen diese Variablen, nach einer positiven Flanke, zurückgesetzt werden. Dazu werden die Ereignisse *M360_WP_CLR* und *M360_WP_CLR* eingeführt, siehe Abb. 3.3.4.

Ereignis	Interpretation	Alphabet
M360_HP_SET	Pos. Flanke an M360 Grundstellung	CON
M360_HP_CLR	Neg. Flanke an M360 Grundstellung	CON
M360_WP_SET	Pos. Flanke an M360 Arbeitsstellung	CON
M360_WP_CLR	Neg. Flanke an M360 Arbeitsstellung	CON

Abbildung 3.3.4: Nominalmodell der Weiche M360

Überwachungsbereich. Für den Reglerentwurf ist lediglich relevant, ob der erweiterte kritische Bereich belegt ist (*ex_occ*) oder nicht (*ex_free*). Die tatsächlich vorhandene Flaschenzahl spielt dabei keine Rolle. Anfänglich ist der erweiterte kritische Bereich frei. Nachfolgend wechselt sein Status von „frei“ nach „belegt“ und umgekehrt. Eine Automatenrepräsentation der Dynamik des kritischen Bereichs ist in Abb. 3.3.5 dargestellt.



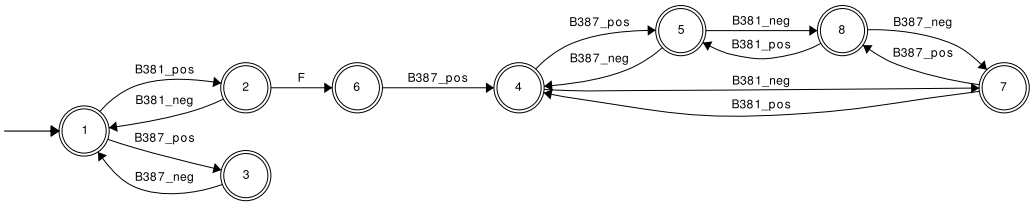
Ereignis	Interpretation	Alphabet
ex_occ	Überwachungsbreich belegt	UCON
ex_free	Überwachungsbereich frei	UCON

Abbildung 3.3.5: Nominalmodell des erweiterten kritischen Bereichs

Kritischer Bereich. Im Nominalbetrieb stellt das FMS durch Verriegelung des Vereinzelers M381 gegen den Sensor B381 Kollisionsfreiheit sicher. Es kann dann nur jeweils

einer der beiden Sensoren B381 und B387 belegt sein kann. Anfänglich befindet sich keine Flasche im kritischen Bereich. Im Nominalbetrieb ist die Dynamik des kritischen Bereichs durch Abfolgen positiver und negativer Flanken an einem der beiden Sensoren B381 und B387 gekennzeichnet.

Im Fehlerfall greift die Verriegelung des Vereinzelers M381 gegen den Sensor B381 nicht mehr, sodass nach einem Fehler die Sensoren B381 und B387 gleichzeitig belegt sein können. Ein Fehler tritt auf, falls der Sensor B381 belegt ist (Zustand 2) und der Sensor B387 eine positive Flanke zeigt, siehe Abb. 3.3.6.



Ereignis	Interpretation	Alphabet
B381_pos	Pos. Flanke an B381	UCON
B381_neg	Neg. Flanke an B381	UCON
B387_pos	Pos. Flanke an B387	UCON
B387_neg	Neg. Flanke an B387	UCON
F	Fehler	LO

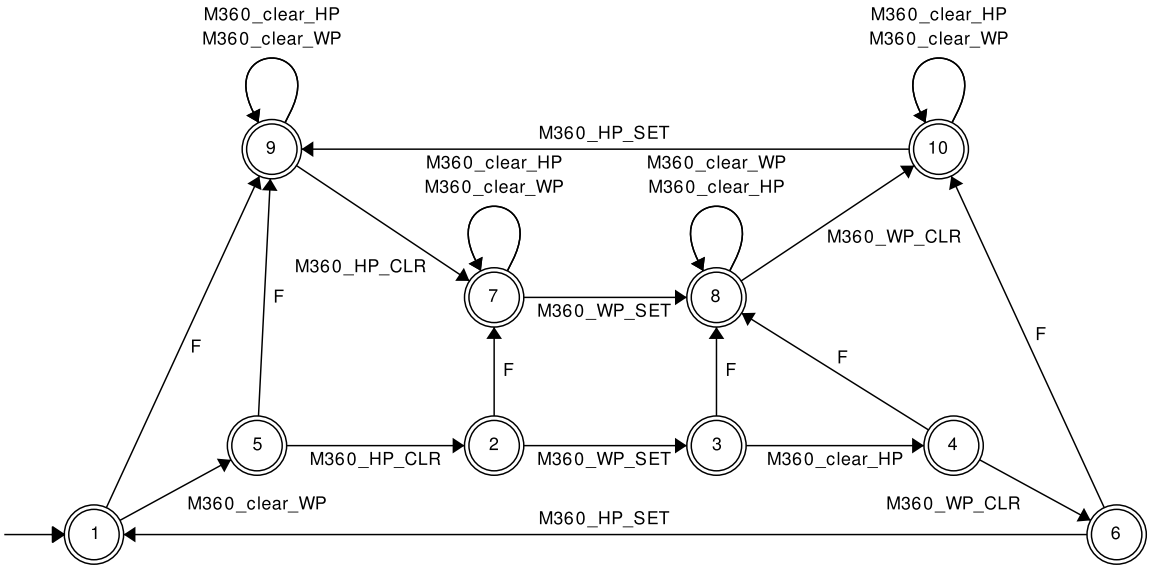
Abbildung 3.3.6: Fehlerverträgliches Modell des kritischen Bereichs

Bemerkung 3.3.1. Das Modell des kritischen Bereichs beschreibt im engeren Sinn nicht das ungesteuerte Systemverhalten, sondern Spezifikationsvorgaben für den Steuerungsentwurf, speziell die Kollisionsfreiheit. Hier wird das gesteuerte Verhalten als Streckenmodell aufgefasst. □

Entwurf eines fehlerverträglichen Reglers

Weiche M360, Spezifikation. Um im Falle eines Fehlers zulässiges Verhalten im geschlossenen Regelkreis zu sichern, wird die Bindung der Positionswechselkommandos *M360_WP_SET, M360_HP_SET, M360_WP_CLR, M360_HP_CLR* an Bedienerereignisse *M360_clear_WP* und *M360_clear_HP* aufgebrochen, siehe Abb. 3.3.7 Zustände 7, 8, 9, 10.

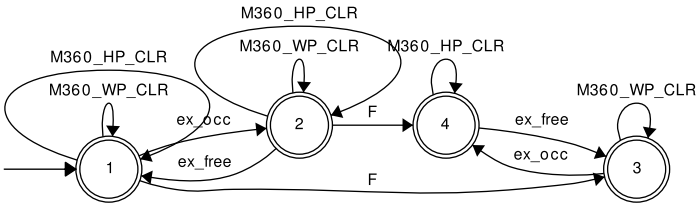
Überwachungsbereich, Spezifikation. Eine Spezifikation zur Verrieglung der Weiche gegen den erweiterten kritischen Bereich ist in Abb. 3.3.8 gezeigt. In den Zuständen 1 und 2 arbeitet der Prozess im Nominalbetrieb; das Schließen und Öffnen der Weiche wird über die Bedienerkommandos durch das FMS festgelegt, siehe Abb. 3.3.10. Nach einem Fehler ziehen die Bedienerkommandos keine Konsequenz mehr nach sich. Die dadurch



Ereignis	Interpretation	Alphabet
M360_HP_SET	Pos. Flanke an M360 Grundstellung	CON
M360_HP_CLR	Neg. Flanke an M360 Grundstellung	CON
M360_WP_SET	Pos. Flanke an M360 Arbeitsstellung	CON
M360_WP_CLR	Neg. Flanke an M360 Arbeitsstellung	CON
M360_clear_WP	Pos. Flanke an Freigabe M360 zur Arbeitsposition	HI
M360_clear_HP	Pos. Flanke an Freigabe M360 zur Grundposition	HI
F	Fehler	LO

Abbildung 3.3.7: Fehlerverträgliche Spezifikation für die Weiche M360

gewonnenen Freiheiten werden zur Verriegelung der Weiche M360 gegen den erweiterten kritischen Bereich genutzt, siehe Zustände 3 und 4. In Zustand 3 ist der Überwachungsbereich frei; das Verriegeln der Weiche wird verboten. Dagegen ist in Zustand 4 der kritische Bereich belegt, und das Öffnen der Weiche wird verboten.



Ereignis	Interpretation	Alphabet
M360_HP_CLR	Neg. Flanke an M360 Grundstellung	CON
M360_WP_CLR	Neg. Flanke an M360 Grundstellung	CON
ex_occ	Erweiterter krit. Bereich belegt	UCON
ex_free	Erweiterter krit. Bereich frei	UCON
F	Fehler	LO

Abbildung 3.3.8: Fehlerverträgliche Spezifikation des Überwachungsbereichs

Bemerkung 3.3.2. Konzeptionell kann ein ereignisdiskreter Regler nur das Auftreten bestimmter Ereignisse verhindern, aber nicht erzwingen. In der verbalen Formulierung der Spezifikation des Überwachungsbereichs wird aber eine aktive Handlung eingefordert.

Die Methodik liefert hier keine Begründung, warum die in Abbildung 3.3.8 dargestellte formale Spezifikation die verbale Spezifikation umsetzt. Gemäß des gewählten Regelsatzes zur Umsetzung ereignisdiskreter Reglerdynamik, siehe Abschnitt 3.2.1, wird ein Ausgangsereignis sofort ausgeführt, falls kein Eingangsereignis erkannt wurde. Erlaubt das Reglermodell das Schließen oder Öffnen der Weiche und wird kein Sensorereignis erkannt, darf davon ausgegegangen werden, dass die Stelleinrichtung das Öffnen bzw. Schließen der Weiche sofort ausführt und damit die Spezifikation im Wortlaut einlöst.

Tatsächlich sind die Positionswechselkommandos steuerbar und können deshalb verboten werden. Regler und Strecke sind abgeschlossen und besitzen daher nur triviale Letztendlichkeitsbedingungen. Es kann deshalb Lösungen des Entwurfsproblems geben, die Kollisionen durch das permanente Verriegeln der Weiche verhindern. Obwohl zulässig, ist dieses Verhalten nicht erwünscht. Letztlich muss im Experiment das Verhalten im geschlossenen Regelkreis geprüft und für akzeptabel befunden werden. □

Gemäß Definition 1.4.2 liegt ein Standardentwurfsproblem vor, sodass der Reglerentwurf mit in [lib13] implementierten Algorithmen durchgeführt werden kann. An der SmA wurden, neben den bereits erwähnten, weitere Entwurfsvorgaben umgesetzt, die aber keinen Einfluss auf die fehlertolerante Steuerungsstrategie haben. Das Modell des realisierten Reglers wies 136 Zustände, 14 Ereignisse und 754 Zustände auf.

3.3.2 Fehlerverdeckende Steuerungsrekonfiguration

Kernstück der Modellbildung für den Rekonfiguratorentwurf ist die Erstellung eines minimal-restriktiven Reglermodells. Auf die Notwendigkeit einer hinreichend genauen Nachbildung der Bedienerschnittstellen wurde bereits mehrfach verwiesen. Daneben stellt der Detailreichtum der Nominalmodelle eine weitere wichtige Variable dar. Im Gegensatz zur fehlerverträglichen Regelung werden zusätzlich Sensoren zur Konsistenzprüfung berücksichtigt und die Modellbildung auf Prozessabbildebene durchgeführt.

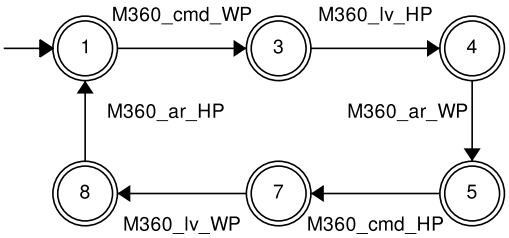
Nominalreglerentwurf

Eine Modellbildung auf Prozessabbildebene verlangt eine genaue Nachbildung möglicher Abfolgen von steigenden und fallenden Signalflanken. Das betrifft sowohl Aktuatorik als auch Sensorik. Mit Sensoren assoziierte virtuelle Ereignisse bedingen die Erzeugung der entsprechenden steigenden und fallenden Flanken durch die Rekonfiguratorimplementierung.

Virtuelle, mit Aktuatoren assoziierte Ereignisse sind aus Sicht der Rekonfiguratorumsetzung Eingangsereignisse. Setzt ein Steuerungsprogramm Rekonfiguratordynamik gemäß [TMP10] um, so reagiert es lediglich auf das Auftreten von nicht-steuerbaren und virtuellen steuerbaren Ereignissen mit der Erzeugung von entsprechenden Ausgangsereignissen. Die korrekte Nachbildung der Abfolge virtueller steuerbarer Ereignisse ist deshalb für die erfolgreiche Umsetzung der fehlerverdeckenden Steuerungsrekonfiguration von großer Bedeutung. Wurde die Nominalreglerimplementierung nicht modellbasiert entworfen, ist die Erstellung entsprechender Nominalmodelle und -spezifikationen, auf Grund fehlender Information über die umgesetzte Steuerungsstrategie, insbesondere der genauen Aktuatoransteuerung, eine große Herausforderung.

Nominalreglerentwurf

Weiche M360. Im Gegensatz zur fehlerverträglichen Regelung wird die Modellbildung für den Rekonfiguratorentwurf auf Prozessabbildebene durchgeführt. Hier steht zur Ansteuerung der Weiche nur ein Eintrag zur Verfügung. Eine pos. Flanke zeigt dann einen Wechsel zur Arbeitsposition und eine neg. Flanke einen Wechsel zur Grundstellung an. Für das Nominalmodell der Weiche müssen neben den Stellereignissen *M360_cmd_WP* und *M360_cmd_WP* zusätzlich Überwachungssensoren für das Erreichen/Verlassen der Arbeits- bzw. Grundposition, *M360_ar_WP*, *M360_lv_WP*, *M360_ar_HP* und *M360_lv_HP*, berücksichtigt werden, siehe Abb. 3.3.9.

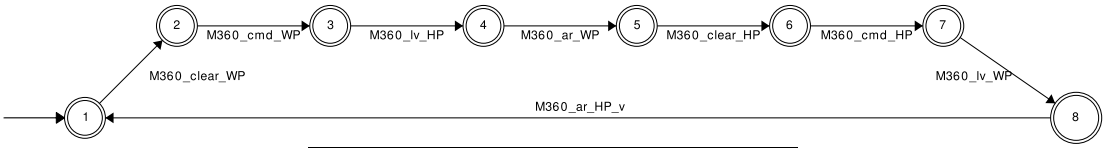


Ereignis	Interpretation	Alphabet
M360_cmd_WP	M360 zur Arbeitsposition	CON
M360_cmd_HP	M360 zur Grundposition	CON
M360_lv_WP	M360 verlässt Arbeitsposition	UCON
M360_lv_HP	M360 verlässt Grundposition	UCON
M360_ar_WP	M360 erreicht Arbeitsposition	UCON
M360_ar_HP	M360 erreicht Grundposition	UCON

Abbildung 3.3.9: Nominalmodell der Weiche M360 für den Rekonfiguratorentwurf

Weiche M360, Spezifikation. Die Entwurfsvorgaben für die Weiche M360 betreffen die Bedienerschnittstellen. Freigabekommandos (*M360_clear_WP* bzw. *M360_clear_HP*) müssen einem Positionswechsel vorausgehen, siehe Abb. 3.3.10.

Der minimal-restriktive Regler $H_{\mathrm{N}}^{\uparrow}$ ist in diesem Fall identisch mit der Spezifikation.

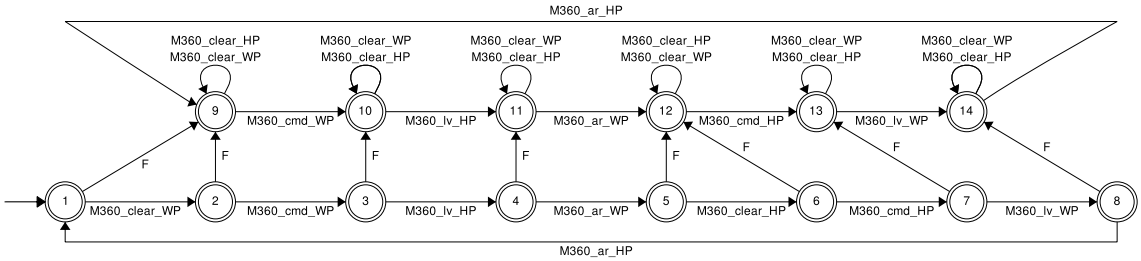


Ereignis	Interpretation	Alphabet
M360_WP	M360 zur Arbeitsposition	CON
M360_HP	M360 zur Grundposition	CON
M360_clear_WP	Freigabe M360 zur Arbeitsposition	HI
M360_clear_HP	Freigabe M360 zur Grundposition	HI

Abbildung 3.3.10: Nominalspezifikation für die Weiche M360

Rekonfiguratorentwurf

Weiche M360, fehlerverträgliche Spezifikation. Die wörtliche Spezifikation für die Weiche M360 für den Rekonfiguratorentwurf gleicht ihrem Pendant in der fehlerverträglichen Regelung. Um im Falle eines Fehlers zulässiges Verhalten im geschlossenen Regelkreis zu sichern, wird die harte Bindung der Positionswechselkommandos *M360_WP* und *M360_HP* an Bedienerereignisse *M360_clear_WP* und *M360_clear_HP* aufgebrochen, siehe Abb. 3.3.11 Zustände 9-14.



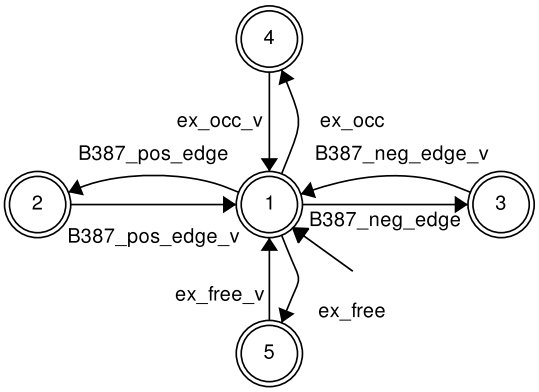
Ereignis	Interpretation	Alphabet
M360_cmd_WP	M360 zur Arbeitsposition	CON
M360_cmd_HP	M360 zur Grundposition	CON
M360_lv_WP	M360 verlässt Arbeitsposition	UCON
M360_lv_HP	M360 verlässt Grundposition	UCON
M360_ar_WP	M360 erreicht Arbeitsposition	UCON
M360_ar_HP	M360 erreicht Grundposition	UCON

Abbildung 3.3.11: Fehlerverträgliche Spezifikation des Überwachungsbereichs

Bemerkung 3.3.3. Die Bedeutung der Bedienerereignisse im Standardregelungsproblem und dem Rekonfiguratorentwurf ist eine grundlegend andere. Während im Standardentwurfsproblem die Semantik der Bedienerereignisse durch die Abfolge der steuerbaren, nicht-steuerbaren und logischen Ereignisse festgelegt wird, verkehrt sich dieser Zusammenhang im Rekonfiguratorentwurf ins Gegenteil. Während die Abfolge der Bedienerereignisse durch den Nominalregler fest vorgegeben ist, legt die fehlerverträgliche Spezifikation fest, welche Abfolgen von steuerbaren, nicht-steuerbaren und logischen Ereignis-

sen mit der Abfolge der Bedienerkommandos konsistent sind. Formal sind im Standardentwurfsproblem die Bedienerkommandos steuerbar, während sie im Entwurf fehlerverträglicher Regler weder steuerbar noch beobachtbar sind.

Weiche M360, Inaktivitätsbedingungen Zusätzlich werden die Inaktivitätsbedingungen gemäß Gl. (2.1) eingefordert sowie eine fehlerunabhängige 1:1 Umsetzung für die Ereignisse siehe Abb. 3.3.12



Ereignis	Interpretation	Alphabet
M360_cmd_WP	M360 zur Arbeitsposition	CON
M360_cmd_HP	M360 zur Grundposition	CON
M360_lv_WP	M360 verlässt Arbeitsposition	UCON
M360_lv_HP	M360 verlässt Grundposition	UCON
M360_ar_WP	M360 erreicht Arbeitsposition	UCON
M360_ar_HP	M360 erreicht Grundposition	UCON

Abbildung 3.3.12: Fehlerunabhängige 1:1 – Umsetzung

Die fehlertolerante Steuerungsstrategie in der fehlerverträglichen Regelung und der fehlerverdeckenden Steuerungsrekonfiguration bleibt gleich. Die Spezifikationsmodelle werden nur leicht angepasst und werden deshalb an dieser Stelle nicht mehr diskutiert.

Man bemerke, dass der in Abbildung 3.3.10 gezeigte Nominalregler nur die minimalrestriktive, sondern überhaupt die einzige Lösung des Nominalentwurfsproblems darstellt. Damit ist ein Modell des Nominalreglers bekannt, und das Rekonfigurationsproblem kann auf ein Standardentwurfsproblem zurückgeführt werden, siehe Korollar 2.0.1. An der SmA wurde ein Rekonfigurator mit 942 Zuständen, 25 Ereignissen und 7388 Transitionen umgesetzt.

3.3.3 Modifikationen des Nominalsteuerungsprogramms

Streckenerweiterungen. Anhand der im Überwachungsbereich vorhandenen Flaschen kann entschieden werden, ob der betreffende Bereich belegt ist oder nicht. Eine Modellbildung mit Automaten ist auf Basis einer oberen Schranke für eine Anzahl vorhandener Flaschen möglich, führt allerdings auf große Zustandsräume. Die Bereichsüberwachung wird deshalb in Software realisiert und das Prozessabbild um einen Eintrag ergänzt. Ist die Flaschenzahl in dem betreffenden Bereich gleich Null, wird in den Prozessabbildeintrag

eine Null geschrieben, anderenfalls eine Eins. Dementsprechend werden steigende Flanken mit dem Ereignis *ex_occ* und fallende Flanken mit dem Ereignis *ex_free* identifiziert.

Virtualisierung des Nominalreglers. Zur Umsetzung der Umbenennungsfunktion *h* gemäß Kap. 1 wird der Nominalsteuerungscode um separate Datenbausteine mit je einer booleschen Variable pro Prozessabbildeintrag ergänzt. Mit Hilfe der Entwicklungsumgebung Simatik Manager konnten alle Stellen im Nominalcode, an denen Referenzen auf relevante Teile des physischen Prozessabbildes vorkommen, aufgelistet werden. Die Umbenennungsfunktion *h* wird dann durch einfaches Ersetzen der Referenzen auf Einträge im physischen Prozessabbild durch die entsprechenden Variablen in den neu angelegten Datenbausteinen umgesetzt.

Ergebnisstand und Bewertung. Es wurden die Ergebnisse einer Machbarkeitsstudie zur fehlerverträglichen Regelung und der fehlerverdeckenden Steuerungsrekonfiguration vorgestellt. Die Ergebnisse lassen auf eine erfolgreiche Anwendung in der industriellen Praxis schließen.

Der Steuerbarkeitsbegriff nach [RW87, RW89] ist ein geeigneter Begriff zur Berücksichtigung der Eingriffsmöglichkeiten ereignisdiskreter Regler. Für die praktische Anwendung im Rahmen dieser Arbeit erweist er sich aber als zu restriktiv und wird durch einen allgemeineren Steuerbarkeitsbegriff nach [BW94] ersetzt. Während die Vollständigkeit als natürliche Eigenschaft von SPS-Programmen gelten darf, bestand, auf Grund der lediglich trivialen Letztendlichkeitseigenschaften der betrachteten Systeme, keine Notwendigkeit für einen expliziten Konfliktfreiheitsnachweis.

Auf Grund der Reglerkomplexität wurde auf automatisierte Verfahren zur Erzeugung von Steuerungscode zurückgegriffen. Abhängig von der Speicherkapazität der verwendeten speicherprogrammierbaren Steuerung gibt es aber auch Grenzen für die Anzahl der Zustände und Transitionen eines Reglermodells. Diese hängen sowohl von dem Streckenmodell als auch von der Spezifikation ab. Um all zu große Zustandsräume zu vermeiden wurden Entwurfsvorgaben, die irgendeine Art von „Zählen“ beinhalten, durch softwaretechnische Streckenerweiterungen unterstützt.

Ein Eingabeparameter des Rekonfiguratorentwurfs ist ein Modell des Nominalreglers. Im Beispiel der Machbarkeitsstudie lag lediglich ein von Hand und gemäß einer informalen Spezifikation programmiertes Steuerungsprogramm vor. Eine Ableitung formaler Spezifikationsmodelle aus der gegebenen Spezifikationsschrift war nicht möglich. Der wesentliche Teil der Spezifikation, nämlich die Bedienereereignissemantik, wurde durch Sichtung des Quellcodes gewonnen und die erlaubte Abfolge physischer Ereignisse durch Beobachten des gesteuerten Prozesses. Seine Vorteile spielt der vorgestellte Ansatz zur fehlerverdeckenden Steuerungsrekonfiguration also erst bei einem vorliegenden Nominalreglermodell aus.

Der hohe Modellierungsaufwand motiviert den Wunsch nach wiederverwendbaren Modellen. Während atomare Systeme wie Weichen und Vereinzeler durchaus komponentenweise modelliert und einer Datenbank zugeführt werden können, gilt das nicht für die Spezifikationen. Letztere müssen von Situation zu Situation angepasst und der Reglerentwurf muss neu durchgeführt werden. Neuere Arbeiten zur hierarchischen Regelung [BM12] adressieren dieses Problem. Eine Diskussion der fehlerverdeckender Steuerungsrekonfiguration in diesen Rahmen erscheint deshalb wünschenswert.

Schlussbemerkung

Im Rahmen einer Industriekooperation wurde auf Basis des Konzepts der fehlerverdeckenden Steuerungsrekonfiguration eine Methodik für den modellbasierten Entwurf fehlertoleranter Steuerungsalgorithmen für Systeme mit ereignisdiskreter Dynamik geschaffen. Als besonderes Merkmal weisen diese Steuerungsalgorithmen die Fähigkeit auf, nachträglich in eine bestehende Steuerungsarchitektur eingefügt werden zu können, ohne die bereits validierten Steuerungsprogramme ersetzen zu müssen.

Mit der Supervisory Control Theory wurde dabei auf einen regelungstechnischen Rahmen zurückgegriffen, der den Entwurf formal verifizierter Softwaresysteme erlaubt. Die Entwurfsziele wurden in Termen des rekonfigurierenden Regelkreises formuliert und bilden den Kern eines Rekonfigurationsproblems. Als dessen Lösung erzielt ein Rekonfigurator per Definition die Entwurfsziele. Er ist in diesem Sinn „correct-by-design“.

Für den praktischen Rekonfiguratorentwurf wurden Entwurfsalgorithmen angegeben, die zunächst zu einem Modell des Rekonfigurators führen. Mit Hilfe entsprechender Codegeneratoren wurde der Steuerungsquellcode automatisiert generiert. Gerade im Kontext der Verlässlichkeit technischer Systeme erscheint dieser Aspekt in einem besonderen Licht, konnten doch so Umsetzungsfehler praktisch ausgeschlossen werden.

Konzeptbedingt sind modellbasierte Regler lediglich korrekt bzgl. der Entwurfsparameter. Fehler in der Modellbildung wirken sich deshalb massiv auf die Funktionstüchtigkeit des Reglers, gerechnet gegen den physikalischen Prozess, aus. In dieser Arbeit kommt beim Entwurf universeller Rekonfiguratoren ein Modell des minimal-restriktiven Nominalreglers zum Tragen. Falls dieses Modell keine Übermenge des tatsächlich implementierten Nominalreglers bildet, kann deshalb nicht erwartet werden, dass im rekonfigurierenden Regelkreis die Entwurfsziele erreicht werden.

Die durchgeführte Machbarkeitsstudie zeigte, dass die fehlerverdeckende Steuerungsrekonfiguration in der industriellen Vorfeldentwicklung erfolgreich umgesetzt werden kann. Mit zunehmender Erfahrung und Sorgfalt bei der Modellbildung nahmen Modellfehler ab.

Verbleibende Modellfehler konnten relativ leicht, z. B. mit geeigneten Softwarewerkzeugen oder Experimenten, gefunden werden. Nach Meinung des Autors werden die Nachteile eines modellbasierten Reglerentwurfs, gerade vor dem Hintergrund verlässlicher Systeme, durch dessen Vorteile, in Form formal korrekter Steuerungsalgorithmen und dem Ausschluss menschlichen Versagens bei der Umsetzung, bei weitem überwogen.

Anhang A

Beweise

A.1 Zum Standardreglerentwurf

Proposition A.1.1. Gegeben sei ein Standardentwurfsproblem (Σ, L, E) und ein Regelkreiskandidat $K \subseteq \Sigma^*$ mit den Eigenschaften

(K1) Steuerbarkeit bzgl. $(L, \Sigma_{\mathrm{UCON}} \dot{\cup} \Sigma_{\mathrm{LO}})$, d. h. $(\mathrm{pre}\, K)(\Sigma_{\mathrm{UCON}} \dot{\cup} \Sigma_{\mathrm{LO}}) \parallel (\mathrm{pre}\, L) \subseteq \mathrm{pre}\, K$

(K2) Vorsilbennormalität bzgl. (L, Σ_{C}), d. h. $\mathrm{pre}\, K = (\mathrm{p}_{\mathrm{P}}^{-1}\, \mathrm{pre}\, L) \cap (\mathrm{p}_{\mathrm{C}}^{-1}\, \mathrm{p}_{\mathrm{C}}\, \mathrm{pre}\, K)$

(K3) Relative Abgeschlossenheit bzgl. L, d. h. $K = (\mathrm{pre}\, K) \cap \mathrm{p}_{\mathrm{P}}^{-1} L$

(K4) Vollständigkeit, d. h. $(\forall s \in \mathrm{pre}\, K\, \exists \sigma \in \Sigma)[s\sigma \in \mathrm{pre}\, K]$

(K5) Spezifikationseinschluss, d. h. $K \subseteq E$,

dann ist $H := \mathrm{p}_{\mathrm{C}}\, \mathrm{pre}\, K$ eine Lösung des gegebenen Standardentwurfsproblems. Im Umkehrschluss erzielt eine Lösung $H \subseteq \Sigma_{\mathrm{C}}^*$ des gegebenen Standardentwurfsproblems im geschlossenen Regelkreis $K := L \parallel H$ die Eigenschaften (K1) bis (K5). □

Beweis. Es sind nachfolgende zwei Aussagen zu zeigen:

(1) Erfüllt $K \subseteq \Sigma^*$ die Eigenschaften (K1) bis (K5), dann löst $H := \mathrm{p}_{\mathrm{C}}\, \mathrm{pre}\, K$ das gegebene Standardentwurfsproblem.

(2) Ist H eine Lösung des Standardentwurfsproblems, dann weist der geschlossene Regelkreis $K := L \parallel H$ die Eigenschaften (K1) bis (K5) auf.

Zu Aussage (1): Man wähle $K \subseteq \Sigma^*$ mit den Eigenschaften (K1) bis (K5) und setze $H := \mathrm{p}_{\mathrm{C}}\, \mathrm{pre}\, K$. Zunächst zeige man die nachfolgenden Hilfsaussagen:

$$(\mathrm{pre}\, L) \parallel H = \mathrm{pre}\, K \tag{A.1}$$

$$L \parallel H = K \tag{A.2}$$

Bezüglich Gl. (A.1) folge man der Ableitung

$$\begin{aligned}(\mathrm{pre}\,L) \parallel H &= (\mathrm{p}_{\mathrm{P}}^{-1}\,\mathrm{pre}\,L) \cap (\mathrm{p}_{\mathrm{C}}^{-1} H) \\ &= (\mathrm{p}_{\mathrm{P}}^{-1}\,\mathrm{pre}\,L) \cap (\mathrm{p}_{\mathrm{C}}^{-1}\,\mathrm{p}_{\mathrm{C}}\,\mathrm{pre}\,K) && \text{[gemäß der Def. von } H\text{]} \\ &= \mathrm{pre}\,K, && \text{[gemäß (K2)]}\end{aligned}$$

während Gl. (A.2) aus

$$\begin{aligned}L \parallel H &= (\mathrm{p}_{\mathrm{P}}^{-1} L) \cap (\mathrm{p}_{\mathrm{C}}^{-1} H) \\ &= (\mathrm{p}_{\mathrm{P}}^{-1} L) \cap (\mathrm{p}_{\mathrm{P}}^{-1}\,\mathrm{pre}\,L) \cap (\mathrm{p}_{\mathrm{C}}^{-1}\,\mathrm{p}_{\mathrm{C}}\,\mathrm{pre}\,K) && \text{[gemäß der Def. von } H\text{]} \\ &= (\mathrm{p}_{\mathrm{P}}^{-1} L) \cap (\mathrm{pre}\,K) && \text{[gemäß (K2)]} \\ &= K, && \text{[gemäß (K3)]}\end{aligned}$$

folgt. Die Gln. (A.1) und (A.2) zeigen, dass der Regler H tatsächlich das Verhalten $L \parallel H = K$ im geschlossenen Regelkreis erzielt. Nachfolgend werden die Eigenschaften (SR1) bis (SR4) aus den Eigenschaften von K und den Gln. (A.1) und (A.2) abgeleitet.

Zur Abgeschlossenheit des Reglers (H0): Aus der Definition von H folgt sofort (H0).

Zur Konfliktfreiheit (SR1). Es gilt

$$\begin{aligned}(\mathrm{pre}\,L) \parallel H &= (\mathrm{p}_{\mathrm{P}}^{-1}\,\mathrm{pre}\,L) \cap (\mathrm{p}_{\mathrm{C}}^{-1} H) \\ &= (\mathrm{p}_{\mathrm{P}}^{-1}\,\mathrm{pre}\,L) \cap (\mathrm{p}_{\mathrm{C}}^{-1}\,\mathrm{p}_{\mathrm{C}}\,\mathrm{pre}\,K) && \text{[gemäß der Def. von } H\text{]} \\ &= \mathrm{pre}\,K && \text{[gemäß (K2)]} \\ &= \mathrm{pre}\,(L \parallel H), && \text{[by Eq. (A.2)]}\end{aligned}$$

woraus (SR1) folgt.

Zur Vollständigkeit (SR2): Gleichung (A.1) besagt $\mathrm{pre}\,K = (\mathrm{pre}\,L) \parallel H$, weshalb aus (K4) die Eigenschaft (SR2) folgt.

Zur Steuerbarkeit (SR3): Gleichung (A.1) besagt $\mathrm{pre}\,K = (\mathrm{pre}\,L) \parallel H$, sodass aus (K1) die Eigenschaft (SR3) folgt.

Zur Entwurfszielsicherung (SR4): Aus Gleichung A.2 and (K5) folgt $L \parallel H = K \subseteq E$, sodass (SR4) gilt. Damit ist der Beweis von Aussage (1) abgeschlossen.

Zu Aussage (2): Man wähle eine beliebige Lösung $H \subseteq \Sigma_{\mathrm{C}}^{*}$ des gegebenen Entwurfsproblems und zeige, dass für $K := L \parallel H$ die Eigenschaften (K1) bis (K5) gelten. Es ist H eine Lösung des Standardentwurfsproblems (Σ, L, E), deshalb besitzt der geschlossene Regelkreis $L \parallel H$ die Eigenschaften (SR1) bis (SR4).

Zur Steuerbarkeit (K1): Per Definition gilt $K := L \parallel H$, sodass aus (SR1) die Eigenschaft (K1) folgt.

Zur Vorsilbennormalität (K2): Um zu zeigen, dass $\operatorname{pre} K \subseteq (\mathrm{p}_{\mathrm{P}}^{-1} \operatorname{pre} L) \cap (\mathrm{p}_{\mathrm{C}}^{-1} \mathrm{p}_{\mathrm{C}} \operatorname{pre} K)$ gilt, bemerke man zunächst, dass aus allgemeinen Rechenregeln $\operatorname{pre} K \subseteq \mathrm{p}_{\mathrm{C}}^{-1} \mathrm{p}_{\mathrm{C}} \operatorname{pre} K$ folgt. Weiter bedeutet $K := L \parallel H$ gleichzeitig auch $\operatorname{pre} K \subseteq \mathrm{p}_{\mathrm{P}}^{-1} \operatorname{pre} L$. Deshalb gilt $\operatorname{pre} K \subseteq (\mathrm{p}_{\mathrm{P}}^{-1} \operatorname{pre} L) \cap (\mathrm{p}_{\mathrm{C}}^{-1} \mathrm{p}_{\mathrm{C}} \operatorname{pre} K)$. Die rückwärtige Einschließung folgt aus

$$
\begin{aligned}
&(\mathrm{p}_{\mathrm{P}}^{-1} \operatorname{pre} L) \cap (\mathrm{p}_{\mathrm{C}}^{-1} \mathrm{p}_{\mathrm{C}} \operatorname{pre} K) \subseteq (\mathrm{p}_{\mathrm{P}}^{-1} \operatorname{pre} L) \cap (\mathrm{p}_{\mathrm{C}}^{-1} \mathrm{p}_{\mathrm{C}} \mathrm{p}_{\mathrm{C}}^{-1} H) && [\text{aus } \operatorname{pre} K = (\operatorname{pre} L) \parallel H \subseteq \mathrm{p}_{\mathrm{C}}^{-1} H] \\
&= (\mathrm{p}_{\mathrm{P}}^{-1} \operatorname{pre} L) \cap (\mathrm{p}_{\mathrm{C}}^{-1} H) && [\text{aus allg. Rechenregeln}] \\
&= (\operatorname{pre} L) \parallel H, \\
&= \operatorname{pre} (L \parallel H) && [\text{gemäß (SR1)}] \\
&= \operatorname{pre} K, && [\text{gemäß der Def. von } K]
\end{aligned}
$$

was den Beweis von (K2) abschließt.

Zur relativen Abgeschlossenheit (K3): Man folge der Ableitung

$$
\begin{aligned}
K = L \parallel H &= (\mathrm{p}_{\mathrm{P}}^{-1} L) \cap \mathrm{p}_{\mathrm{C}}^{-1} H && [\text{gemäß Def. von } K] \\
&= (\mathrm{p}_{\mathrm{P}}^{-1} \operatorname{pre} L) \cap (\mathrm{p}_{\mathrm{P}}^{-1} L) \cap \mathrm{p}_{\mathrm{C}}^{-1} H \\
&= ((\operatorname{pre} L) \parallel H) \cap \mathrm{p}_{\mathrm{P}}^{-1} L \\
&= (\operatorname{pre} (L \parallel H)) \cap \mathrm{p}_{\mathrm{P}}^{-1} L && [\text{gemäß (SR1)}] \\
&= (\operatorname{pre} K) \cap \mathrm{p}_{\mathrm{P}}^{-1} L, && [\text{gemäß der Def. von } K]
\end{aligned}
$$

und bemerke, dass (K3) gilt.

Zur Vollständigkeit (K4): Per Definition gilt $K := L \parallel H$, sodass mit (SR2) auch (K4) gilt.

Zur Spezifikationsentsprechung (K5): Per Definition gilt $K := L \parallel H \subseteq E$, sodass aus (SR4) die Eigenschaft (K5) folgt, was den Beweis von Aussage (2) abschließt. q.e.d.

Proposition A.1.2. Gegeben sei ein Standardentwurfsproblem (Σ, L, E). Es bezeichnet $K^{\uparrow} \subseteq \Sigma^*$ die supremale Teilsprache (von Σ^*) bzgl. der Eigenschaften (K1) bis (K5). Weiter sei mit $H^{\Uparrow} \subseteq \Sigma_{\mathrm{C}}^*$ die supremale Lösung des Standardentwurfsproblems (Σ, L, E) gegeben. Dann sind die durch $H^{\uparrow} := \mathrm{p}_{\mathrm{C}} \operatorname{pre} K^{\uparrow}$ und $H^{\Uparrow}$ erzielten Verhalten im geschlossenen Regelkreis identisch, in Zeichen $L \parallel H^{\uparrow} = L \parallel H^{\Uparrow}$. □

Beweis. Gemäß Prop. A.1.1 ist $H^{\uparrow} := \mathrm{p}_{\mathrm{C}} K_{\mathrm{N}}^{\uparrow}$ eine Lösung des Standardentwurfsproblems. Es ist $H^{\Uparrow}$ die supremale Lösung des gegebenen Entwurfsproblems, weshalb $H^{\uparrow} \subseteq H_{\mathrm{N}}^{\Uparrow}$ gilt.

Aus der Monotonie des parallelen Produkts erhält man $L \parallel H^{\uparrow} \subseteq L \parallel H^{\Uparrow}$. Um die rückwärtige Einschließung zu zeigen, wähle man eine beliebige Lösung $H \subseteq \Sigma^*$ des Standardentwurfsproblems. Gemäß Prop. A.1.1, erfüllt der geschlossene Regelkreis $L \parallel H$ die Eigenschaften (K1) bis (K5). Insbesondere gilt dieser Zusammenhang auch für die supremale Lösung $H^{\Uparrow}$ des Standardentwurfsproblems. Es ist $K^{\uparrow}$ die supremale Teilsprache der Spezifikation E bzgl. der Eigenschaften (K1) bis (K4), sodass $L \parallel H^{\Uparrow} \subseteq K^{\uparrow} = L \parallel H^{\uparrow}$ gilt. q.e.d.

A.2 Hilfssätze

Proposition A.2.1. Gegeben seien die Alphabete $\Sigma_1 \subseteq \Sigma_2 \subseteq \Sigma$, die Projektionen $\mathrm{p}_1 : \Sigma^* \to \Sigma_1^*$, $\mathrm{p}_2 : \Sigma^* \to \Sigma_2^*$ sowie $\mathrm{p}_1^2 : \Sigma_2^* \to \Sigma_1^*$ zusammen mit den entsprechenden inversen Projektionen p_1^{-1}, p_2^{-1} und p_1^{-2}. Für eine beliebige Sprache $L \subseteq \Sigma_1^*$ gilt:

$$\mathrm{p}_2\,\mathrm{p}_1^{-1}L = \mathrm{p}_1^{-2}L.$$

□

Beweis. Man bemerke zunächst, dass

$$\mathrm{p}_2\,\mathrm{p}_1^{-1}L = \{s \in \Sigma_2^* \mid (\exists t \in \Sigma^*)[\mathrm{p}_1 t \in L \quad \wedge \quad \mathrm{p}_2 t = s]\} \text{ und} \tag{A.3}$$

$$\mathrm{p}_1^{-2}L = \{s \in \Sigma_2^* \mid \mathrm{p}_1 s \in L\} \tag{A.4}$$

gilt.

Beginnend mit der rückwärtigen Einschließung $\mathrm{p}_1^{-2}L \subseteq \mathrm{p}_2\,\mathrm{p}_1^{-1}L$ wähle man ein beliebiges $s \in \mathrm{p}_1^{-2}L$ und zeige $s \in \mathrm{p}_2\,\mathrm{p}_1^{-1}L$. Offensichtlich ist $s \in \Sigma_2^*$, und es bleibt die Existenz eines $t \in \Sigma^*$ mit $\mathrm{p}_1 t \in L$ und $\mathrm{p}_2 t = s$ zu zeigen. Man setze $t := s$ und bemerke, dass wegen $\Sigma_2 \subseteq \Sigma$ mit $s \in \Sigma^*$ auch $t \in \Sigma^*$ gilt. Aus (A.4) folgt sofort $\mathrm{p}_1 t = \mathrm{p}_1 s \in L$. Weiter ist $s \in \Sigma_2^*$, sodass durch die Projektion p_2 keine Zeichen aus s enfernt werden, d. h. $\mathrm{p}_2 s = s$. Abschließend bemerke man, dass t für s die Anforderungen in (A.3) erfüllt. Folglich ist $s \in \mathrm{p}_2\,\mathrm{p}_1^{-1}L$.

Um nun $\mathrm{p}_2\,\mathrm{p}_1^{-1}L \subseteq \mathrm{p}_1^{-2}L$ zu zeigen, wähle man $s \in \mathrm{p}_2\,\mathrm{p}_1^{-1}L$ und man zeige $s \in \mathrm{p}_1^{-2}L$. Es ist Σ_2^* das Bild der Projektion p_2, woraus sofort $s \in \Sigma_2^*$ folgt. Es bleibt $\mathrm{p}_1 s \in L$ zu zeigen. Gemäß (A.3) existiert ein $t \in \Sigma^*$ mit $\mathrm{p}_1 t \in L$ und $\mathrm{p}_2 t = s$, sodass $\mathrm{p}_1 s = \mathrm{p}_1\,\mathrm{p}_2 t$ gilt. Um den Beweis abzuschließen ist $\mathrm{p}_1\,\mathrm{p}_2 t = \mathrm{p}_1 t$ nachzuweisen. Für einen Beweis durch Induktion über die Länge $n \in \mathbb{N}_0$ von t betrachte man zu Beginn den Fall $n = 0$. Dann gilt $\mathrm{p}_1\,\mathrm{p}_2 t^0 =$

$p_1 \varepsilon = \varepsilon = p_1 t^0$. Für ein beliebiges i, mit $0 < i < n$, nehme man an, es gelte $p_1 p_2 t^i = p_1 t^i$. Im Induktionsschritt $i \to i+1$ betrachte man $t^{i+1} = t^i \sigma$ und man zeige $p_1 p_2 t^{i+1} = p_1 t^{i+1}$. Dazu sind die folgenden Fälle zu unterscheiden: $\sigma \notin \Sigma_2$, $\sigma \in \Sigma_2 - \Sigma_1$ und $\sigma \in \Sigma_1$. Falls $\sigma \notin \Sigma_2$ gilt, hat man $p_2 t^{i+1} = p_2 t^i$ und aus $\Sigma_1 \subseteq \Sigma_2$ folgt $p_1 t^i = p_1 t^{i+1}$. Aus dem letzen Argument und der Induktionshypothese folgt: $p_1 p_2 t^i = p_1 t^i = p_1 t^{i+1}$. Falls $\sigma \in \Sigma_2 - \Sigma_1$ gilt, dann erhält man $p_2 t^i \sigma = (p_2 t^i)\sigma$. Weiter gilt $p_1 p_2 t^i \sigma = p_1 (p_2 t^i)\sigma = p_1 p_2 t^i$. Aus der Induktionshypothese folgt $p_1 p_2 t^i = p_1 t^i$ und wegen $\sigma \notin \Sigma_1$ gilt dann auch $p_1 t^i = p_1 t^{i+1}$. Falls $\sigma \in \Sigma_1$ gilt, dann folgt $p_2 t^i \sigma = (p_2 t^i)\sigma$ und $p_1 t^{i+1} = (p_1 t^i)\sigma$. Zudem gilt wegen $\Sigma_1 \subseteq \Sigma_2$ auch $p_1 p_2 t^i \sigma = (p_1 p_2 t^i)\sigma$. Aus der Induktionshypothese folgt $(p_1 p_2 t^i)\sigma = (p_1 t^i)\sigma = p_1 t^{i+1}$ q.e.d.

Anhang B

Fehlerverträgliche Modelle für Systeme mit Aktuatoren und Sensoren

In Abschnitt 1.4.1 wurde eine Klassifikation fehlerverträglicher Modelle anhand verschiedener Ausfallmuster vorgeschlagen. Ausgehend von einem Streckennominalmodell $L_N \subseteq \Sigma^*_{P,N}$ kann dann für jede Klasse ein Konstruktionsschema zur Erstellung des entsprechenden Streckenfehlermodells $L_D \subseteq \Sigma^*_{P,F}$ angegeben werden. Es wird dazu von regulären Modellen, repräsentiert durch endliche Automaten, ausgegegangen. Weiter wird angenommen, dass Fehler durch möglicherweise ausbleibende nominelle Ereignisse, nachfolgend mit *kritische Ereignisse* $\Sigma_{krit} \subseteq \Sigma_N$ bezeichnet, angezeigt werden. Das Fehlerereignis F wird in jedem Zustand erlaubt, in dem auch das betreffende kritische Ereignis möglich ist. Ist das kritische Ereignis ein Aktuatorereignis, dann endet mit dem Fehlerereignis die bisher aufgelaufene Ereignisfolge, falls das kritische Ereignis unterbleibt. Ist dagegen das kritische Ereignis ein Sensorereignis, dann kann auf Grund der fehlenden Information nicht entschieden werden, in welchem Zustand sich das Gesamtsystem befindet. Die bisher aufgelaufene Ereigniskette wird aber nicht unterbrochen. Im Detail werden die folgenden Ausfallmuster betrachtet:

- *Vollständiges Versagen.* Ein Aktuator versagt ohne sich regenerieren zu können. Das entsprechende kritische Ereignis tritt dann nicht mehr auf.
- *Versagen mit Regeneration.* Nach einem Ausfall kann sich der Aktuator wieder erholen. Je nach Zustand des Aktuators ist das Auftreten des entsprechenden kritischen Ereignisses möglich oder nicht.
- *Uneingeschränkter Betrieb.* Ein Aktuator ist immer in Betrieb. Das Auftreten des entsprechenden kritischen Ereignisses ist nicht zu unterbinden.

- *Vollständiger Informationsverlust.* Ein Sensor liefert keine Information mehr. Das entsprechende kritische Ereignis tritt dann nicht mehr auf.
- *Informationsverlust mit Regeneration.* Nach einem Ausfall kann sich der Sensor wieder erholen. Je nach Zustand des Sensors wird ein entsprechendes kritisches Ereignis generiert oder nicht.
- *Zufälliges Auslösen.* Ein Sensor liefert Information, ohne einem Muster zu folgen. Ein entsprechendes kritisches Ereignis kann ohne Einschränkung generiert werden.

Die angegebenen Konstruktionsschemata werden gegen drei prototypische Streckenmodelle auf ihre Schlüssigkeit geprüft: Systeme ohne Redundanz, Abbildung B.0.1, Systeme mit Redundanz, Abbildung B.0.2, und Systeme mit mehreren fehlerbehafteten Komponenten, Abbildung B.0.3. Darin bezeichnen *ACT, ACT1, ACT2* Aktuatorereignisse und *SNS, SNS1, SNS2* entsprechend Sensorereignisse. In den genannten Abbildungen zeigt die jeweils linke Spalte ein Aktorikbeispiel und die jeweils rechte Spalte ein Sensorikbeispiel.

Abbildung B.0.1: Systeme ohne Redundanz

Abbildung B.0.2: Systeme mit Redundanz

Abbildung B.0.3: Systeme mit mehreren fehlerbehafteten Komponenten

B.1 Aktuatoren

B.1.1 Vollständiges Versagen

Unterliegt ein Aktuator dem Fehlermuster „Vollständiges Versagen", dann endet eine aufgelaufene Ereigniskette, statt mit dem entsprechenden kritischen Ereignis, mit dem Feh-

lerereignis. Im Modell verbleiben dann lediglich alternative Ereignisfolgen, wie sie z. B. durch redundante Aktorik in das System eingebracht werden.

Ausgehend von einem Streckennominalmodell und einer Menge kritischer Ereignisse, stellt Algorithmus 2 ein mögliches Konstruktionsverfahren für ein entsprechendes Streckenfehlermodell vor. Darin wird zunächst die Nominaltransitionenstruktur, inklusive der Markierungen, dupliziert. Anschließend werden Fehlertransitionen eingefügt. Schließlich werden aus den duplizierten Transitionen diejenigen mit kritischen Ereignissen entfernt. Letztlich werden alle nicht erreichbaren und nicht co-erreichbaren Zustände entfernt.

Algorithmus 2 L_D-Konstruktion für das Ausfallmuster "Vollständiges Versagen"

$G := (Q, \Sigma, \delta, q_0, Q_m), L(G) = L_N$
$\Sigma_{krit} \subseteq \Sigma$
$H := (X, \Lambda, \xi, x_0, X_m), X := Q, \Lambda := \Sigma, \xi := \delta, x_0 := q_0, X_m := \emptyset$
$\Lambda_{krit} := \Sigma_{krit}$,
$X_0 := X, \xi_0 := \xi$
$X_1 := hX_0, h : X_0 \to X_1$ bijektiv
// Füge Fehlerereignis ein
$\Lambda \leftarrow \Lambda \dot{\cup} \{F\}$
// Dupliziere Nominaltransitionenstruktur
$X \leftarrow X \cup X_1$
for all $(x_1, \alpha, x_2) \in \xi_0$ **do**
  $\xi \leftarrow \xi \cup (hx_1, \alpha, hx_2)$
end for
for all $x \in X_0$ **do**
  if $x \in X_m$ **then** $X_m \leftarrow X_m \cup \{hx\}$ **end if**
end for
// Füge Fehlertransitionen ein
for all $(x_1, \alpha, x_2) \in \xi_0$ **do**
  if $\alpha \in \Lambda_{krit}$ **then** $\xi \leftarrow \xi \cup (x_1, F, hx_2)$ **end if**
end for
// Entferne Transitionen mit kritischen Ereignissen
for all $(x_1, \alpha, x_2) \in \xi - \xi_0$ **do**
  if $\alpha \in \Lambda_{krit}$ **then** $\xi \leftarrow \xi - (x_1, \alpha, x_2)$ **end if**
end for
// Entferne nicht erreichbare und co-erreichbare Zustände
Trimm(H)
return H

Beispiel B.1.1. Betrachtet wird ein System mit redundanter Aktorik, vgl. Abbildung B.0.2 links, mit dem kritischen Ereignis *ACT1*. Das Berechnungsschema 2 liefert den in Abbildung B.1.1 dargestellten Automaten.

Der Teilautomat $G|_{\{1,2,3\}}$ spiegelt das Nominalverhalten wider. Das Fehlerereignis F ist an das kritische Ereignis *ACT1* gebunden und ist deshalb nur in Zustand 1 erlaubt. Dem Ausfallmuster "Vollständiges Versagen" entsprechend, enden alle Ereignisketten, die nach einem Fehler das Ereignis *ACT1* beinhalten, mit dem Fehlerereignis. Der Modellbildung

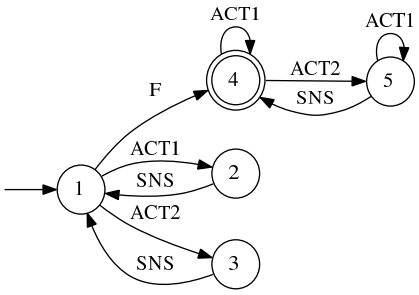


Abbildung B.1.1: Streckenfehlermodell bei redundanter Aktorik und vollständigem Versagen

auf Prozessabbildebene ist geschuldet, dass das Ereignis *ACT1* zwar auftreten kann, aber keinen Einfluss mehr auf den Prozess hat. Es wird als Eigenschleife in den Zuständen 4 und 5 aufgenommen. □

Tabelle B.1.1 listet Streckennominal- und die entsprechenden Streckenfehlermodelle für Systeme mit und ohne Redundanz sowie für Systeme mit mehreren defekten Komponenten auf.

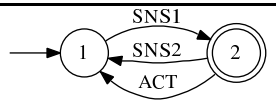
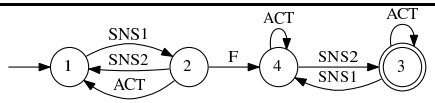
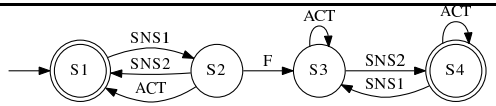
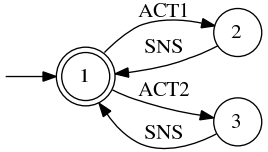
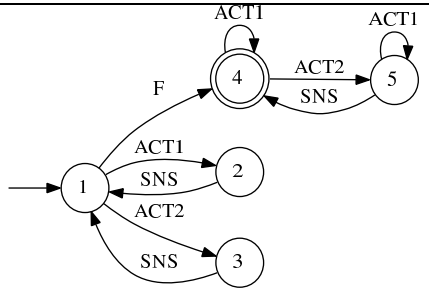
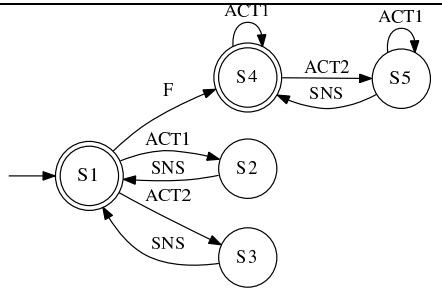
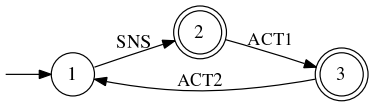
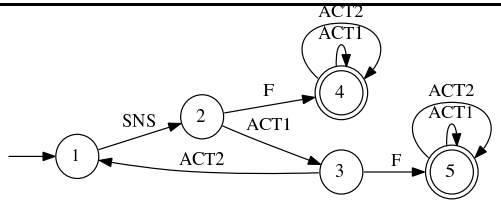
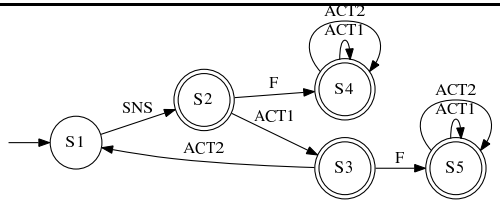
Σ_{krit}	L_N	L_D	L_F
ACT			
ACT1			
$\{ACT1,$ $ACT2\}$			

Tabelle B.1.1: Konstruktion fehlerverträglicher Verhalten für das Ausfallmuster “Vollständiges Versagen”

B.1.2 Versagen mit Regeneration

Unterliegt ein Aktuator dem Ausfallmuster “Versagen mit Regeneration”, tritt das betreffende kritische Ereignis möglicherweise zeitlich verzögert oder überhaupt nicht mehr auf. In der vorgestellten Form enthalten ereignisdiskrete Modelle allerdings keine Information über das zeitliche Auftreten der Ereignisse, sodass sich die Ereignisreihenfolge vor und nach dem Fehler nicht unterscheidet. Der Fall, in dem der Aktuator dauerhaft ausfällt, wird durch die Aufnahme zusätzlicher Letztendlichkeitsbedingungen in das Modell berücksichtigt.

Zu einem gegebenen Streckennominalmodell und einer Menge kritischer Ereignisse kann ein Streckenfehlermodell durch Duplizieren der Nominaltransitionenstruktur inklusive Markierungen und Einfügen geeigneter Fehlertransition sowie dem Markieren von Fehlersenken generiert werden, siehe Algorithmus 3.

Beispiel B.1.2. Betrachtet wird ein System ohne redundante Aktorik, vgl. Abbildung B.0.1 links, mit dem kritischen Ereignis *ACT*. Berechnungsschema 3 liefert den Automaten in Abbildung B.1.2.

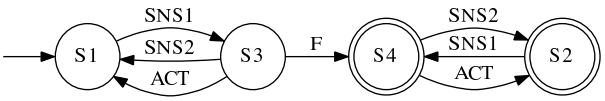


Abbildung B.1.2: Streckenfehlermodell ohne redundate Aktorik und Versagen mit Regeneration

Der Teilautomat $G|_{\{1,2\}}$ spiegelt das Nominalstreckenverhalten wider, während die verbleibenden Zustände das Streckenverhalten nach einem Fehler beschreiben. Das lokale Verhalten vor und nach dem Fehler ist gleich. Durch den Fehler werden jedoch die Letztendlichkeitsbedingungen verändert, sodass ein Auftreten des Ereignisses *ACT* nicht mehr zwingend erforderlich ist, bzw. das System durchgehend in Zustand 2 verweilen kann. □

Tabelle B.1.2 listet Streckennominalmodell und Streckenfehlermodell sowie das entsprechende fehlerverträgliche Verhalten für Systeme mit und ohne Redundanz sowie mit mehreren fehlerhaften Komponenten auf.

Algorithmus 3 L_{D}-Konstruktion für das Ausfallmuster “Versagen mit Regeneration”

$G := (Q, \Sigma, \delta, q_0, Q_{\mathrm{m}}), L(G) = L_{\mathrm{N}}$

$\Sigma_{\mathrm{krit}} \subseteq \Sigma$

$H := (X, \Lambda, \xi, x_0, X_{\mathrm{m}}), X := Q, \Lambda := \Sigma, \xi := \delta, x_0 := q_0, X_{\mathrm{m}} := \emptyset$

$\Lambda_{\mathrm{krit}} := \Sigma_{\mathrm{krit}}$,

$X_0 := X$, $\xi_0 := \xi$

$X_1 := \mathrm{h}X_0$, $\mathrm{h} : X_0 \to X_1$ bijektiv

// Füge Fehlerereignis ein

$\Lambda \leftarrow \Lambda \dot{\cup} \{F\}$

// Dupliziere Nominaltransitionenstruktur

$X \leftarrow X \cup X_1$

for all $(x_1, \alpha, x_2) \in \xi_0$ **do**

    $\xi \leftarrow \xi \cup (\mathrm{h}x_1, \alpha, \mathrm{h}x_2)$

end for

for all $x \in X_0$ **do**

    if $x \in X_{\mathrm{m}}$ **then** $X_{\mathrm{m}} \leftarrow X_{\mathrm{m}} \cup \{\mathrm{h}x\}$ **end if**

end for

// Füge Fehlertransitionen ein

for all $(x_1, \alpha, x_2) \in \xi_0$ **do**

    if $\alpha \in \Lambda_{\mathrm{krit}}$ **then** $\xi \leftarrow \xi \cup (x_1, F, \mathrm{h}x_2)$ **end if**

end for

// Füge zusätzliche Markierungen ein

for all $(x_1, \alpha, x_2) \in \xi - \xi_0$ **do**

    if $\alpha \in \Lambda_{\mathrm{krit}}$ & $\neg x_1 \in X_{\mathrm{m}}$ **then** $X_{\mathrm{m}} \leftarrow X_{\mathrm{m}} \cup \{x_1\}$ **end if**

end for

return H

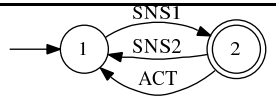
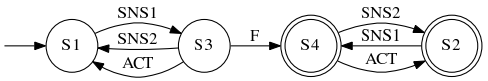
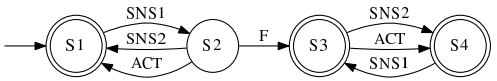
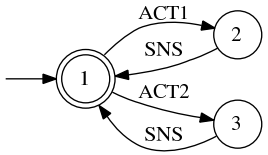
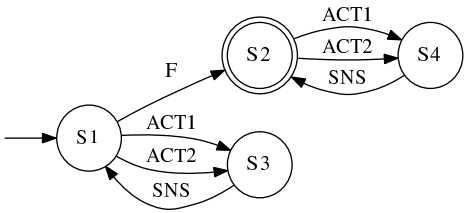
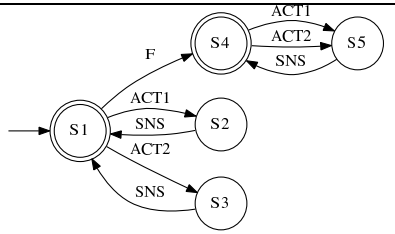
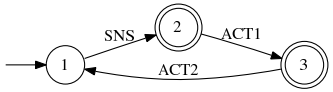
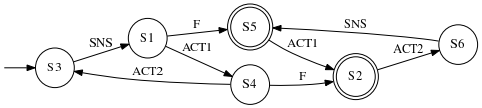
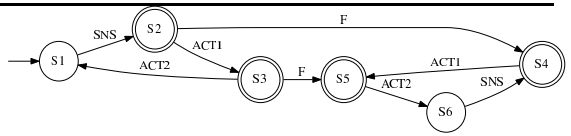
Σ_{krit}	L_N	L_D	L_F
ACT			
ACT1			
$\{ACT1, ACT2\}$			

Tabelle B.1.2: Konstruktion fehlerverträglicher Verhalten für das Ausfallmuster “Versagen mit Regeneration”

B.1.3 Uneingeschränkter Betrieb

Befindet sich ein Aktuator im „uneingeschränkten Betrieb“, tritt das entsprechende Ereignis sofort ein und das System wechselt seinen Zustand. Das entsprechende kritische Ereignis durch Regelung zu unterdrücken, ist also nicht mehr möglich. Im Modell wird dies durch das Ersetzen des entsprechenden Ereignisses durch eine ε-Transition berücksichtigt.

Bemerkung B.1.1. Effektiv ändert sich nach dem Auftreten eines Fehlers das Steuerbarkeitsattribut des betreffenden kritischen Ereignisses. Es ist daher auch denkbar, das kritische Ereignis durch ein neu eingeführtes nicht-steuerbares, aber beobachtbares Ereignis zu ersetzen. Ein solches Ereignis zu generieren setzt allerdings eine geeignete Diagnoseeinrichtung voraus, worauf in dieser Arbeit verzichtet werden soll. □

Berechnungsschema 4 liefert, bei gegebenen Streckennominalmodell und einer Menge kritischer Ereignisse, das entsprechende Streckenfehlermodell. Dazu werden die Nominaltransitionenstruktur dupliziert und geeignete Fehlertransitionen eingeführt. Anschließend werden in der duplizierten Transitionenstruktur alle kritischen Ereignisse umbenannt. Abschließend wird eine Projektion auf die Nominalereignisse und das Fehlerereignis durchgeführt.

Beispiel B.1.3. Betrachtet wird ein System mit Aktuatorredundanz, vgl. Abbildung B.0.2 links, mit dem kritischen Ereignis *ACT1*. Das Berechnungsschema 4 liefert für diese Eingabeparameter den Automaten in Abbildung B.1.3.

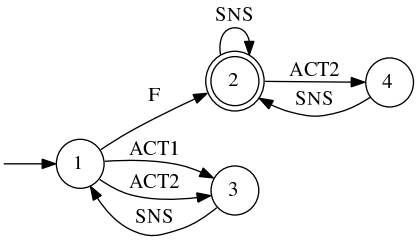


Abbildung B.1.3: Streckenfehlermodell bei Aktuatorredundanz und uneingeschränktem Betrieb

Die Zustände 1,3 spiegeln das Streckennominalverhalten wider, während die Zustände 2,4 das Streckenfehlerverhalten wiedergeben. Gemäß dem Ausfallmuster “Uneingeschränkter Betrieb” wird in Zustand 2 das Ereignis *ACT1* „übersprungen“. Es besteht keine Möglichkeit mehr das Ereignis *SNS* durch Regelung zu verhindern. Zudem kann das Ereignis *SNS* beliebig oft hintereinander auftreten. □

Tabelle B.1.3 listet Streckennominalmodell und Streckenfehlermodell sowie das entsprechende fehlerverträgliche Verhalten für Systeme mit und ohne Redundanz sowie mit mehreren fehlerhaften Komponenten auf.

Algorithmus 4 L_{D}-Konstruktion für das Ausfallmuster “Uneingeschränkter Betrieb”

$G := (Q, \Sigma, \delta, q_0, Q_{\mathrm{m}}), L(G) = L_{\mathrm{N}}$
$\Sigma_{\mathrm{krit}} \subseteq \Sigma$
$H := (X, \Lambda, \xi, x_0, X_{\mathrm{m}}), X := Q, \Lambda := \Sigma, \xi := \delta, x_0 := q_0, X_{\mathrm{m}} := \emptyset$
$\Lambda_{\mathrm{krit}} := \Sigma_{\mathrm{krit}}$,
$X_0 := X$, $\xi_0 := \xi$
$X_1 := \mathrm{h}X_0$, $\mathrm{h} : X_0 \to X_1$ bijektiv
// Füge Fehlerereignis ein
$\Lambda \leftarrow \Lambda \dot{\cup} \{F\}$
// Dupliziere Nominaltransitionenstruktur
$X \leftarrow X \cup X_1$
for all $(x_1, \alpha, x_2) \in \xi_0$ **do**
    $\xi \leftarrow \xi \cup (\mathrm{h}x_1, \alpha, \mathrm{h}x_2)$
end for
for all $x \in X_0$ **do**
    if $x \in X_{\mathrm{m}}$ **then** $X_{\mathrm{m}} \leftarrow X_{\mathrm{m}} \cup \{\mathrm{h}x\}$ **end if**
end for
// Füge Fehlertransitionen ein
for all $(x_1, \alpha, x_2) \in \xi_0$ **do**
    if $\alpha \in \Lambda_{\mathrm{krit}}$ **then** $\xi \leftarrow \xi \cup (x_1, F, \mathrm{h}x_2)$ **end if**
end for
// Ersetze Transitionen mit kritischen Ereignissen durch ε-Transition
for all $(x_1, \alpha, x_2) \in \xi - \xi_0$ **do**
    if $\alpha \in \Lambda_{\mathrm{krit}}$ **then**
        $\xi \leftarrow \xi \cup (x_1, \varepsilon, x_2)$
        $\xi \leftarrow \xi - (x_1, \alpha, x_2)$
    end if
end for
// Determinisierung
Deterministic(H)
return H

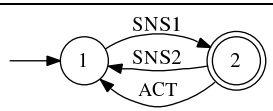
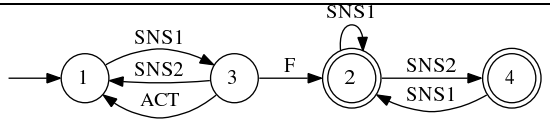
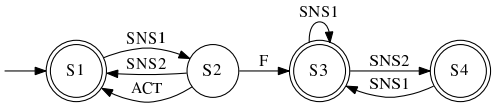
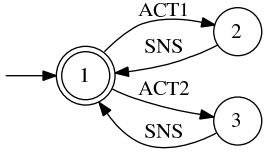
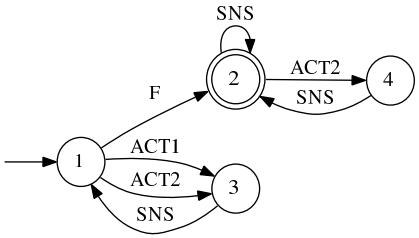
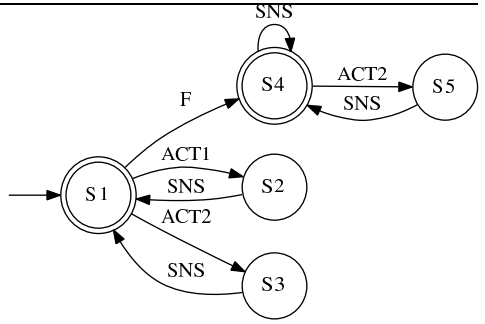
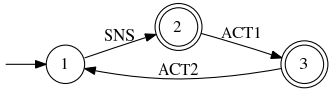
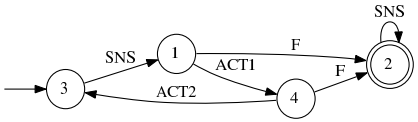
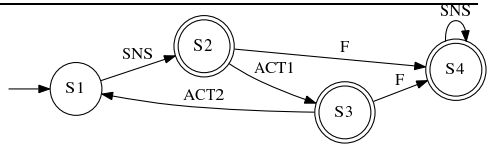
Σ_{krit}	L_N	L_D	L_F
ACT			
ACT1			
$\{ACT1,$ $ACT2\}$			

Tabelle B.1.3: Konstruktion des fehlerverträglichen Verhaltens für das Ausfallmuster “Uneingeschränkter Betrieb”

Die Ausfallmuster “Vollständiges Versagen” und “Versagen mit Regeneration” führen auf Modelle mit Ereignisfolgen, die sich vor und nach dem Fehlerereignis gleichen. Das ist weniger der Modellkonstruktion zuzuschreiben, als der konzeptbedingten fehlenden Zeitinformation in ereignisdiskreten Modellen. Bedingt eine Spezifikation eine fehlerbedingte Verhaltensänderung, ist ein erfolgreicher normalitätsbasierter Reglerentwurf mit den hier vorgestellten Methoden nur dann zu erwarten, wenn der Fehler beobachtbar ist.

B.2 Sensoren

B.2.1 Vollständiger Informationsverlust

Unterliegt ein Sensor dem Ausfallmuster “Vollständiger Informationsverlust”, steht nicht mehr genügend Information zur Generierung eines entsprechenden Ereignisses zur Verfügung. Der exakte Systemzustand ist damit nicht mehr bekannt. Um diesen Umstand zu berücksichtigen, werden die entsprechenden kritischen Ereignisse nach dem Auftreten eines Fehlers durch $\varepsilon-$Transitionen ersetzt und eine Determinisierung durchgeführt.

Berechnungsschema 7 liefert bei gegebenen Streckennominalmodell und einer Menge kritischer Ereignisse das entsprechende Streckenfehlermodell. Dazu wird die Nominaltransitionenstruktur dupliziert, und es werden geeignete Fehlertransitionen eingeführt. Anschließend werden in der duplizierten Transitionenstruktur alle kritischen Ereignisse umbenannt und eine Projektion auf die Nominalereignisse und das Fehlerereignis durchgeführt.

Beispiel B.2.1. Betrachtet wird ein System mit Sensorredundanz, vgl. Abbildung B.0.2 rechts, mit dem kritischen Ereignis *SNS1*. Das Berechnungsschema 7 liefert für diese Eingabeparameter den Automaten in Abbildung B.2.1.

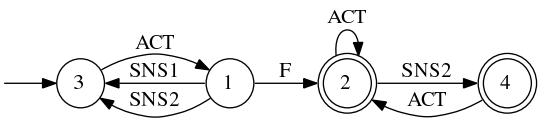


Abbildung B.2.1: Streckenfehlermodell bei Sensorredundanz und uneingeschränktem Informationsverlust

Die Zustände 1,3 spiegeln das Streckennominalverhalten wider, während die Zustände 2,4 das Streckenfehlerverhalten wiedergeben. Gemäß dem Ausfallmuster “Vollständiger Informationsverlust” fehlt in Zustand 2 die Information, um zu entscheiden, ob ein Zustandswechsel stattgefunden hat oder nicht. Dementsprechend ist in Zustand 2, d. h. nach

Algorithmus 5 L_D-Konstruktion für das Ausfallmuster "Vollständiger Informationsverlust"

$G := (Q, \Sigma, \delta, q_0, Q_m), L(G) = L_N$
$\Sigma_{krit} \subseteq \Sigma$
$H := (X, \Lambda, \xi, x_0, X_m), X := Q, \Lambda := \Sigma, \xi := \delta, x_0 := q_0, X_m := \emptyset$
$\Lambda_{krit} := \Sigma_{krit}$,
$X_0 := X$, $\xi_0 := \xi$
$X_1 := hX_0$, $h : X_0 \to X_1$ bijektiv
// Füge Fehlerereignis ein
$\Lambda \leftarrow \Lambda \dot{\cup} \{F\}$
// Dupliziere Nominaltransitionenstruktur
$X \leftarrow X \cup X_1$
for all $(x_1, \alpha, x_2) \in \xi_0$ **do**
    $\xi \leftarrow \xi \cup (hx_1, \alpha, hx_2)$
end for
for all $x \in X_0$ **do**
    if $x \in X_m$ **then** $X_m \leftarrow X_m \cup \{hx\}$ **end if**
end for
// Füge Fehlertransitionen ein
for all $(x_1, \alpha, x_2) \in \xi_0$ **do**
    if $\alpha \in \Lambda_{krit}$ **then** $\xi \leftarrow \xi \cup (x_1, F, hx_2)$ **end if**
end for
// Ersetze Transitionen mit kritischen Ereignissen durch ε-Transition
for all $(x_1, \alpha, x_2) \in \xi - \xi_0$ **do**
    if $\alpha \in \Lambda_{krit}$ **then**
        $\xi \leftarrow \xi \cup (x_1, \varepsilon, x_2)$
        $\xi \leftarrow \xi - (x_1, \alpha, x_2)$
    end if
end for
// Determinisierung
Deterministic(H)
return H

einem gedachten *SNS1*-Ereignis sofort *ACT* erlaubt. Nominelle Ereignisketten, die *SNS2* beinhalten, verbleiben auch nach dem Fehler im Modell. □

Tabelle B.2.1 listet Streckennominalmodell und Streckenfehlermodell sowie das entsprechende fehlerverträgliche Verhalten für Systeme mit und ohne Redundanz sowie mit mehreren fehlerhaften Komponenten auf.

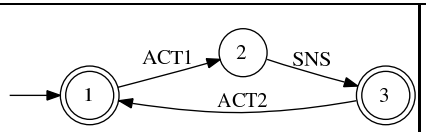
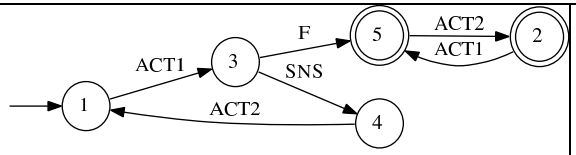
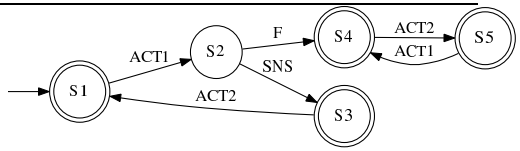
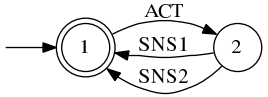
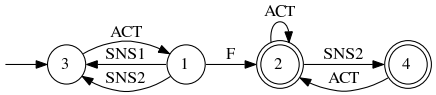
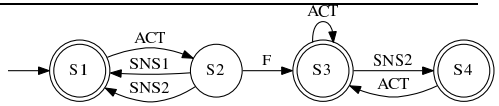
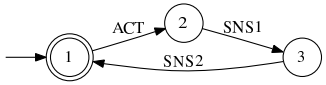
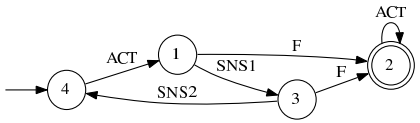
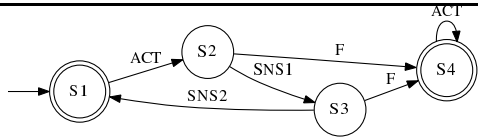
Σ_{krit}	L_N	L_D	L_F
ACT			
ACT1			
$\{ACT1,$ $ACT2\}$			

Tabelle B.2.1: Konstruktion fehlerverträglicher Verhalten für das Ausfallmuster “Vollständiger Informationsverlust”

B.2.2 Informationsverlust mit Regeneration

Unterliegt ein Sensor dem Ausfallmuster “Informationsverlust mit Regeneration”, so muss nicht zwingend genügend Information zur Generierung eines kritischen Ereignisses vorhanden sein. Der Fehler führt zu einem oder mehreren möglichen Systemzuständen. Entsprechend wird zu jeder Transition mit dem betreffenden Ereignis eine parallele $\varepsilon-$Transition eingeführt.

Berechnungsschema 6 liefert bei gegebenen Streckennominalmodell und einer Menge kritischer Ereignisse das entsprechende Streckenfehlermodell. Dazu wird die Nominaltransitionenstruktur dupliziert und es werden geeignete Fehlertransitionen eingeführt. Anschließend werden in der duplizierten Transitionenstruktur parallel zu Transitionen mit kritischen Ereignissen entsprechende Transitionen mit umbenannten Ereignissen eingeführt. Abschließend wird eine Projektion bzgl. der Nominalereignisse und des Fehlerereignisses durchgeführt.

Beispiel B.2.2. Betrachtet wird ein System ohne Sensorredundanz, vgl. Abbildung B.0.1 rechts, mit dem kritischen Ereignis *SNS*. Das Berechnungsschema 7 liefert für diese Eingabeparameter den Automaten in Abbildung B.2.2.

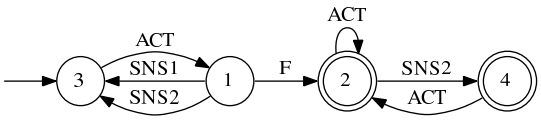


Abbildung B.2.2: Streckenfehlermodell bei Sensorredundanz und wiederkehrendem Informationsverlust

Die Zustände 1,3 spiegeln das Streckennominalverhalten wider, während die Zustände 2,4 das Streckenfehlerverhalten wiedergeben. Gemäß dem Ausfallmuster “Informationsverlust mit Regeneration” fehlt in Zustand 2 möglicherweise die Information, um zu entscheiden, ob ein Zustandswechsel stattgefunden hat oder nicht. Dementsprechend ist in Zustand 6 sowohl *ACT* als auch *SNS2* erlaubt. □

Tabelle B.2.2 listet Streckennominalmodell und Streckenfehlermodell sowie das entsprechende fehlerverträgliche Verhalten für Systeme mit und ohne Redundanz sowie mit mehreren fehlerhaften Komponenten auf.

Algorithmus 6 L_{D}-Konstruktion für das Ausfallmuster “Informationsverlust mit Regeneration”

$G := (Q, \Sigma, \delta, q_0, Q_{\mathrm{m}}), L(G) = L_{\mathrm{N}}$
$\Sigma_{\mathrm{krit}} \subseteq \Sigma$
$H := (X, \Lambda, \xi, x_0, X_{\mathrm{m}}), X := Q, \Lambda := \Sigma, \xi := \delta, x_0 := q_0, X_{\mathrm{m}} := \emptyset$
$\Lambda_{\mathrm{krit}} := \Sigma_{\mathrm{krit}}$,
$X_0 := X$, $\xi_0 := \xi$
$X_1 := \mathrm{h}X_0$, $\mathrm{h} : X_0 \to X_1$ bijektiv
// Füge Fehlerereignis ein
$\Lambda \leftarrow \Lambda \dot{\cup} \{F\}$
// Dupliziere Nominaltransitionenstruktur
$X \leftarrow X \cup X_1$
for all $(x_1, \alpha, x_2) \in \xi_0$ **do**
    $\xi \leftarrow \xi \cup (\mathrm{h}x_1, \alpha, \mathrm{h}x_2)$
end for
for all $x \in X_0$ **do**
    if $x \in X_{\mathrm{m}}$ **then** $X_{\mathrm{m}} \leftarrow X_{\mathrm{m}} \cup \{\mathrm{h}x\}$ **end if**
end for
// Füge Fehlertransitionen ein
for all $(x_1, \alpha, x_2) \in \xi_0$ **do**
    if $\alpha \in \Lambda_{\mathrm{krit}}$ **then** $\xi \leftarrow \xi \cup (x_1, F, \mathrm{h}x_2)$ **end if**
end for
// Einfügen von $\varepsilon-$Transitionen
for all $(x_1, \alpha, x_2) \in \xi - \xi_0$ **do**
    if $\alpha \in \Lambda_{\mathrm{krit}}$ & $\neg x_1 \in X_{\mathrm{m}}$ **then**
        $\xi \leftarrow \xi \cup (x_1, \varepsilon, x_2)$
    end if
end for
// Determinisierung
Deterministic(H)
return H

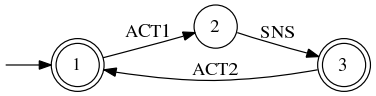
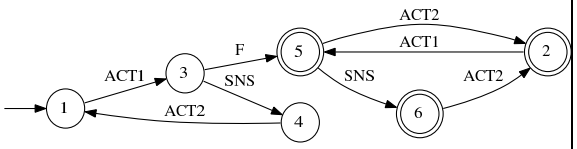
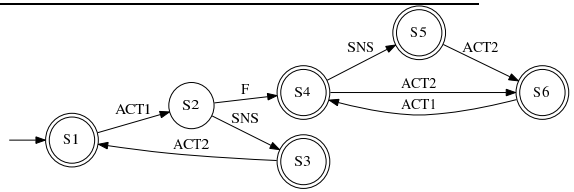
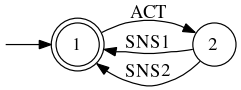
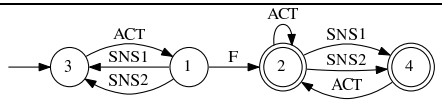
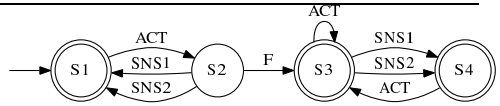
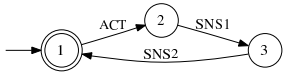
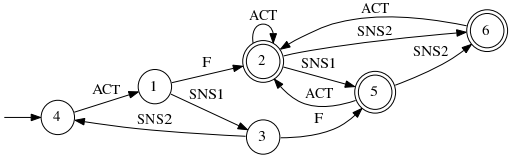
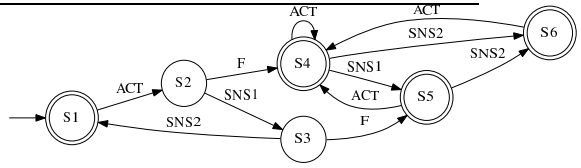
Σ_{krit}	L_N	L_D	L_F
SNS			
SNS1			
$\{SNS1, SNS2\}$			

Tabelle B.2.2: Konstruktion fehlerverträglicher Verhalten für das Ausfallmuster “Informationsverlust mit Regeneration”

B.2.3 Zufälliges Auslösen

Unterliegt ein Sensor dem Ausfallmuster “Zufälliges Auslösen”, kann zu jedem Zählzeitpunkt ein entsprechendes Ereignis generiert werden. Dementsprechend wird nach dem Fehler zu jedem betreffenden Nominalereignis eine parallele $\varepsilon-$Transition eingeführt. Zudem werden die betreffenden Nominalereignisse als Eigenschleifen im Modell vorgesehen. Ein entsprechendes Konstruktionsschema ist mit Algorithmus 7 gegeben.

Beispiel B.2.3. Betrachtet wird ein System ohne Sensorredundanz, vgl. Abbildung B.0.1 rechts, mit dem kritischen Ereignis *SNS*. Das Berechnungsschema 7 liefert für diese Eingabeparameter den Automaten in Abbildung B.2.3.

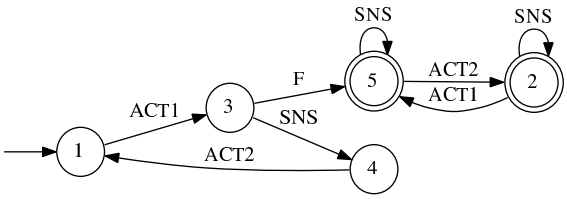


Abbildung B.2.3: Streckenfehlermodell bei Sensorredundanz und wiederkehrendem Informationsverlust

Die Zustände 1,3,4 spiegeln das Streckennominalverhalten wider, während die Zustände 5,2 das Streckenfehlerverhalten wiedergeben. Gemäß dem Ausfallmuster “Zufälliges Auslösen” fehlt in Zustand 2 die Information, um zu entscheiden, ob ein Zustandswechsel stattgefunden hat oder nicht. Dementsprechend ist in Zustand 5 das Ereignis *ACT2* erlaubt. Außerdem ist das Ereignis *SNS* in dem Zustand 2 als Eigenschleife vorgesehen und spiegelt das zufällige Auslösen des entsprechenden Sensors wider. □

Tabelle B.2.3 listet Streckennominalmodell und Streckenfehlermodell sowie das entsprechende fehlerverträgliche Verhalten für Systeme mit und ohne Redundanz sowie mit mehreren fehlerhaften Komponenten auf.

Algorithmus 7 L_D-Konstruktion für das Ausfallmuster "Dauerhafter Informationsverlust"

$G := (Q, \Sigma, \delta, q_0, Q_m), L(G) = L_N$

$\Sigma_{krit} \subseteq \Sigma$

$H := (X, \Lambda, \xi, x_0, X_m), X := Q, \Lambda := \Sigma, \xi := \delta, x_0 := q_0, X_m := \emptyset$

$\Lambda_{krit} := \Sigma_{krit}$,

$X_0 := X$, $\xi_0 := \xi$

$X_1 := hX_0, h : X_0 \to X_1$ bijektiv

// Füge Fehlerereignis ein

$\Lambda \leftarrow \Lambda \dot{\cup} \{F\}$

// Dupliziere Nominaltransitionenstruktur

$X \leftarrow X \cup X_1$

for all $(x_1, \alpha, x_2) \in \xi_0$ **do**

  $\xi \leftarrow \xi \cup (hx_1, \alpha, hx_2)$

end for

for all $x \in X_0$ **do**

  if $x \in X_m$ **then** $X_m \leftarrow X_m \cup \{hx\}$ **end if**

end for

// Füge Fehlertransitionen ein

for all $(x_1, \alpha, x_2) \in \xi_0$ **do**

  if $\alpha \in \Lambda_{krit}$ **then** $\xi \leftarrow \xi \cup (x_1, F, hx_2)$ **end if**

end for

// Ersetze Transitionen mit kritischen Ereignissen durch ε-Transition

for all $(x_1, \alpha, x_2) \in \xi - \xi_0$ **do**

  if $\alpha \in \Lambda_{krit}$ **then**

    $\xi \leftarrow \xi \cup (x_1, \varepsilon, x_2)$

    $\xi \leftarrow \xi - (x_1, \alpha, x_2)$

  end if

end for

for all $\sigma \in \Sigma_{krit}$ **do**

  for all $x \in X - X_0$ **do**

    $\xi \leftarrow \xi \cup (x, \sigma, x)$

  end for

end for

// Determinisierung

Deterministic(H)

return H

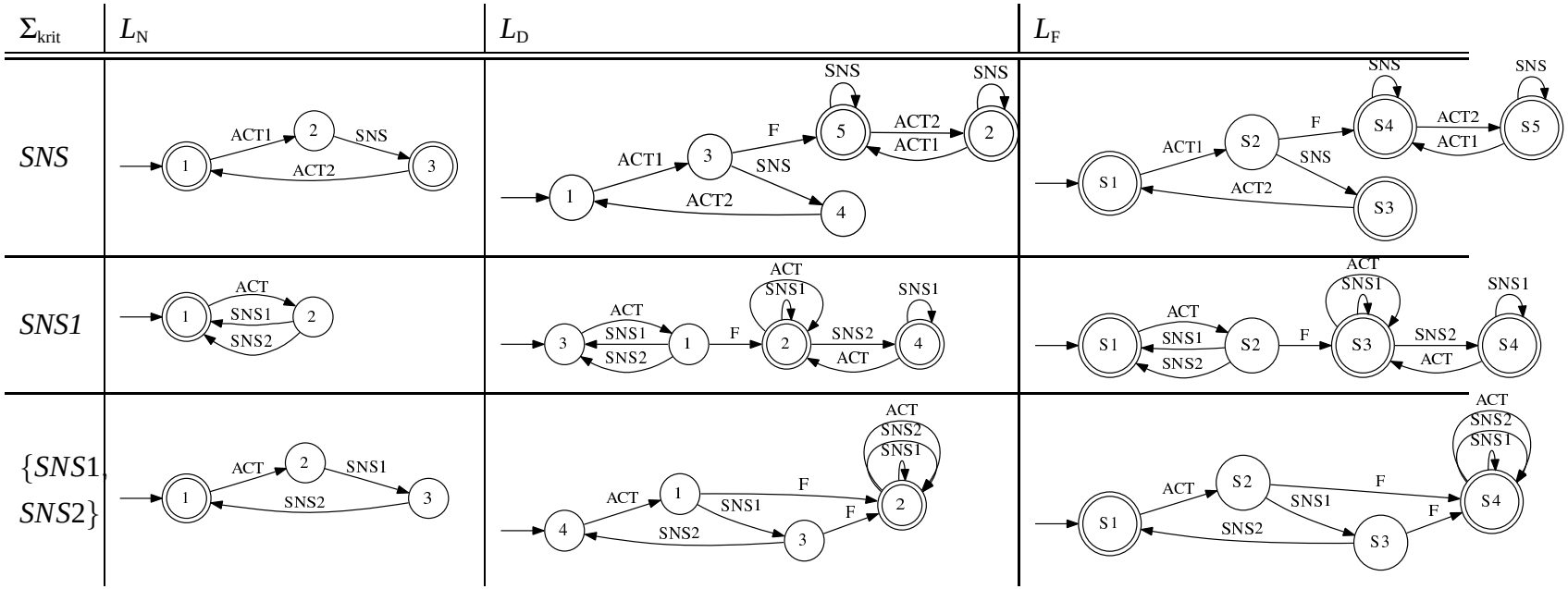


Tabelle B.2.3: Konstruktion fehlerverträglicher Verhalten für das Ausfallmuster “Zufälliges Auslösen”

Literaturverzeichnis

[BC07] F. Basile and P. Chiacchio. On the implementation of supervised control of discrete event systems. *IEEE Transactions on Control Systems Technology*, (15):725–739, 2007.

[Bir14] A. Birolini. *Reliability Engineering*. Springer, 2014.

[BKL+06] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, and J. Schröder. *Diagnosis and Fault-Tolerant Control*. Springer, 2006.

[BLW05] S. E. Bourdon, M. Lawford, and W.M. Wonham. Robust nonblocking supervisory control of discrete-event systems. *IEEE Transactions of Automatic Control*, 50(12):2015–2021, 2005.

[BM12] C. Baier and T. Moor. A hierarchical control architecture for sequential behaviours. *Workshop on Discrete Event Systems 2012*, pages 259–264, 2012.

[BW94] B. A. Brandin and W. M. Wonham. Supervisory control of timed discrete-event systems. *Automatic Control, IEEE Transactions on*, 39(2):329–342, 1994.

[CK99] J. E. R. Cury and B.H. Krogh. Robustness of supervisors for discrete-event systems. *IEEE Transactions of Automatic Control*, 44(2):376–379, 1999.

[CL98] K.-H. Cho and J.-T. Lim. Synthesis of fault-tolerant supvervisor for automated manufacturing systems: A case study on photolithographic process. *IEEE Trans. Robotics and Automation*, 14(2):348–351, 4 1998.

[CL08] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Springer, second edition, 2008.

[CM89] H. Cho and S. I. Marcus. On supremal languages of classes of sublanguages that arise in supervisor synthesis problems with partial observation. *Mathematics of Control, Signals and Systems*, 2(1):47–69, 1989.

[FH98] M. Fabian and A. Hellgren. PLC-based Implementation of Supervisory Control for Discrete Event Systems. In *Proceedings of the 37th IEEE Conference on Decisions and Control*, pages 3305–3310, 1998.

[HU94] John E. Hopcroft and Jeffrey D. Ullman. *Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie*. Addison-Wesley Longman Publishing Co., Inc., 3 edition, 1994.

[KGM92] R. Kumar, V. Garg, and S. I. Marcus. On supervisory control of sequential behaviors. *IEEE Trans. Automatic Control*, 37:1978–85, 1992.

[KGM93] R. Kumar, V. Garg, and S. I. Marcus. Language stability and stabilizability of discrete event dynamical systems. *SIAM Journal of Control and Optimization*, 31:132–0, 1993.

[KT12] R. Kumar and S. Takai. A framework for control-reconfiguration following fault-detection in discrete event systems. *Proc. 8th IFAC SAFEPROCESS*, 2012.

[Lac12] J. Lachky. Interpretation synthetischer ereignisdiskreter Regler auf speicherprogrammierbaren Steuerungen. Masterarbeit, Friedrich-Alexander Universität Erlangen-Nürnberg, 2012.

[lib13] libFAUDES. A software library for discrete event systems, 2006-2013.

[Lin93] F. Lin. Robust and adaptive supervisory control of discrete event systems. *IEEE Transactions of Automatic Control*, 38(12):1848–1852, 1993.

[LW] R. J. Leduc and Y. Wang. Sampled-data supervisory control.

[LW88] F. Lin and W. M. Wonham. On observability of discrete-event systems. *Information Sciences*, 44:173–198, 1988.

[MBY+12] T. Moor, Ch. Baier, T. S. Yoo, F. Lin, and S. Lafortune. On the computation of supremal sublanguages relevant to supervisory control. *Proc. 11th Workshop on Discrete Event Systems (WODES)*, 2012.

[Nke13] Y. S. Nke. *Fault-Tolerant Control of Nondeterministic Input/Ouput Automata*. Logos Verlag Berlin, 2013.

[PL99] S. J. Park and J. T. Lim. Fault-tolerant robust supervisor for discrete event systems with model uncertainty and its application to a workcell. *IEEE Trans. Robotics and Automation*, 15(2):386–391, 1999.

[PSL11] A. Paoli, M. Sartini, and S. Lafortune. Active fault tolerant control of discrete event systems using online diagnostics. *Automatica*, 47(4):639–649, 2011.

[Ric11] J. H. Richter. *Reconfigurable control of nonlinear dynamical systems: fault hiding approach*, volume 408 of *LNCIS*. Springer, 2011.

[RW87] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM J. Control and Optimization*, 25:206–230, 1987.

[RW89] P. J. Ramadge and W. M. Wonham. The control of discrete event systems. *Proc. IEEE*, 77:81–98, 1989.

[Sch11] F. Schieber. Implementierung von ereignisdiskreten Reglern auf speicherprogrammierbaren Steuerungen. Bachelorarbeit, Friedrich-Alexander Universität Erlangen-Nürnberg, 2011.

[Sch12] K. Schmidt. Computation of supervisors for reconfigurable machine tools. *Proc. 11th Workshop on Discrete Event Systems*, pages 227–232, 2012.

[SMP08] K. Schmidt, Th. Moor, and S. Perk. Nonblocking hierachical control of decentralized discrete event systems. *IEEE Trans. Automatic Control*, 53:2252–65, 2008.

[SS13] A. N. Sulek and K. Schmidt. Computation of fault-tolerant supervisors for discrete event systems. *Proc. of the 4th IFAC Workshop on Dependable Control of Discrete Systems*, 2013.

[SSL95] M. Sampath, R. Sengupta, and S. Lafortune. Diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, 1995.

[Ste05] Th. Steffen. *Control Reconfiguration of Dynamical Systems: Linear Approaches and Structural Tests*, volume 320 of *LNCIS*. Springer, 2005.

[Tak00] S. Takai. Robust supervisory control of a class of timed discrete event systems under partial observation. *Systems and Control Letters*, 39(4):267 – 273, 2000.

[TM] Th. Wittmann T. Moor, C. Baier. Consistent abstractions for the purpose of supervisory control. Accepted for presentation at the 52th IEEE Conference on Decision and Control.

[TMP10] K. Schmidt T. Moor and S. Perk. Applied Supervisory Control for a Flexible Manufacturing System. In *Workshop on Discrete Event Systems (WODES)*, pages 725–739, 2010.

[WHK08] Q. Wen, J. Huang, and R. Kumar. Synthesis of optimal fault-tolerant supervisor for discrete event systems. *American Control Conference*, pages 1172–1177, 2008.

[WKHL08] Q. Wen, R. Kumar, J. Huang, and H. Liu. A framework for fault-tolerant control for discrete event systems. *IEEE Trans. Automatic Control*, 53(8):1839–49, 2008.

[WR87] W. M. Wonham and P. J. Ramadge. On the supremal controllable sublanguage of a given language. *SIAM J. Control and Optimization*, 25:637–659, 1987.

[WRM12] Th. Wittmann, J. H. Richter, and T. Moor. Fault-tolerant control of discrete event systems based on fault-accommodating models. *Proc. 8th IFAC SAFEPROCESS*, pages 854–859, 2012.

[WRM13] Th. Wittmann, J. H. Richter, and T. Moor. Fault-hiding control reconfiguration for a class of diskrete event systems. *Proc. of the 4th IFAC Workshop on Dependable Control of Discrete Systems*, 2013.

[WW96] K. C. Wong and W. M. Wonham. Hierarchical control of discrete-event systems. *Discrete Event Dynamic Systems*, 6(3):241–273, 1996.