LTL Modellüberprüfung

Olaf Noppens

Ausarbeitung zum Hauptseminar
Modellüberprüfung
im
Wintersemester 2001/2002

Universität Ulm
Fakultät für Informatik
Abteilung Künstliche Intelligenz

Inhaltsverzeichnis

1 Einleitung 3
2 GrundlegendeBegriffe 4
3 LineareTemporallogik(*LTL*) 5
3.1 SyntaxundSemantik 6
3.2 VergangenheitsbezogeneOperatoren 7
3.3 StrikteOperator en 8
3.4 LTL,CTLundCTL* 9
4 Automaten 9
4.1 EndlicherAutomat(determinis tisch&nichtdeterministisch) 9
4.2 BüchiAutomat 11
4.3 AlternatierenderBüchi -Automat 12
5 LTLundAutomaten 14
5.1 Spezifikation 14
5.2 ÜberführungeinerLTL -FormelineinenAutomaten 15
5.3 Verifikation 16
6 Ausblick 17
7 Literaturnachweis 17

Zusammenfassung

IndieserArbeitwirdeineMöglichkeitderModellüberprüfungbetrachtet,beidereinschon fertiggestelltesProgrammdaraufhinuntersuchtwird,obesseinegefordertenSpezifikat ionen erfüllt.DabeiwerdendieSpezifikationenmithilfederlinearen Temporallogikform uliert.Die eigentlicheÜberprüfungerfolgtnacheinerUmformungvonProgrammundSpezif ikationen inBüchi -AutomatendurchAnwendungvonautomatentheoretischenErkenntnissen.Imfo l-gendenwerdendazuSyntaxundSemantikdervergangenheits -undzukunftbezogenenOper a-torenvonLTL,dienotwendigenGrundlagenderAutomatentheorie(insbesondereder Büchi -Automat)sowiedieTransformationvonProgrammundSpezifikationinAutomatenvorg e-stellt.

1 Einleitung

MitderzunehmendenVerbreitungvon ComputersystemeninallenBereichendesLebens werdenauchimmergrößereAnforderungenvonZuverlässigkeitundKorrektheitanbestim m-te,insbesonderekomplexeSystemegestelltwerden.EsmagaucheineVielzahlvonComp u-terprogrammengeben(obsieinHardwa refest -oderinSoftwarekodiertsind,spieltdabei keineRolle),diezwarmö glichstzuverlässiglaufensollen,derenÜberprüfungaufKorrektheit aberinkeinemVe rhältnismitdemfinanziellenAufwandundeinerpraktischenDurchführung steht.Sokannbeis pielsweiseeinineinemRasierapparateingebettetes,AmoklaufendesSy s-temnurbegrenztSchadenanrichten.DemgegenüberstehenaberSystemeinkritischenBere i-chen,indenennichtkorrektarbeitendeSystemeunmittelbaru nvorhersehbargroßeSchäden anMens chenzurFolgehabenkönnen.AlsklassischesBeispielseienhierSystemeinKer n-kraftwerkengenannt.HeutzutagewirdaberauchModellüberprüfung(*modelchecking*)inz u-nehmendenMaßinderModellierungvonHardware,insbesondereComputerchips,eingesetzt. DennindieserBranchemitsehrhohenInvestitionsausgabenkönnenfehlerhafteChipsein finanziellesFiaskobedeutenundbiszurInso lvenzdesgesamtenUnternehmensführen. SoistdieVerifizierungvonkorrektenProgrammenbzw.vonessentiellenProgrammte ilenein wichtigesZiel,insbesondereauchinUmgebungen,indeneneinSystemausmehrerennebe n-läufigenProgrammenbestehenmussunddieKorrektheitdesGesamtsystemsgewährleistet seinmuss.DennnichtnureinewachsendeKomplexitätinnerhalbgeschlossener Programme sondernauchdieZunahmevonoffenennebenläufigenProgrammen,beispielsweisedurchdie VerteilungvonRe ssourcenallerArt,bringtdieGefahrvonFehlersituationenmitsich.Man kannsichleichtvorstellen,dasseine rseitsdieeinzelnenFehlersi tuationenmiteinergeringen Eintrittswahrscheinlichkeitve rbundensind,andererseitsaberanvielenStellenimSystem dieseSituationenlauernundsichimschlimmstenFallsogargegenseitigverstärkenkö nnen. ImfolgendenhatBewerbenwollenwirnunbetr achten,wiemanzueinerModellüberpr üfung einesSy stemsmitLTLgelangenkann.TypischerweisebestehenSystemeausProgrammen. EinProgrammselbstistdabeieinAlgorithmus,der *Berechnungen*ausführtübertypische r-weisefestenDatenstrukturen.DurchdieA usführungdesAlgorithmus,d.h.durchdasFor t-schreitenvonB erechnungen,könnenDaten *variablen*verändertwerden.DieAusführungdes Algorithmusselbstgeschiehtdadurch,dassdasProgrammvoneinemZustand(repräsentiert durchdieVariablen)ineinenand erenZustandübergeht(vgl.Abbildung1).

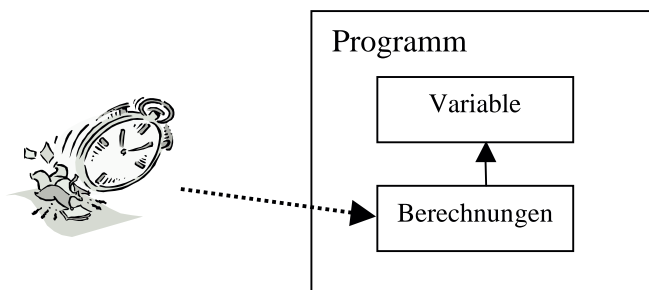

Abbildung 1: MitFortschreitenderZeitwerdenBerechnungenausgeführt,diedenZ ustand desProgrammsbzw.dieVaria blendirektverändern.

ImfolgendensollendieVariablenalseinzigeBestandteiledesProgrammsangesehenwerden, diezeitabhängiggeändertwerdenkönnen.DadieBefehleeinerBerechnungaufeinemRec h-nertypischerweiseindiskreterZeita blaufen,verändertsichderZustandeinesProgramms ebenfallsindi skretenAbständen.AufdieseWeisekanndieBerechnungalseinAblauf(*run*) vonverschiedenen(Programm -)Zuständenaufgefasstwerden,wiein Abbildung 2grafisch veranschaulichtist.

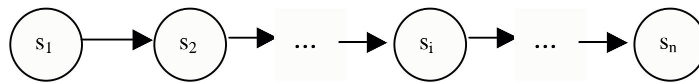

Abbildung 2: LineareFolgeaufeinanderfolgenderProgrammzuständen alsAblaufeinerBerec hnung.

BetrachtetmannuneinesolchelineareAbfolgevonZuständen,soistesleichtverständlich, zuversuchen,die SpezifikationenvonBerechnungenmithilfederlinearenTemporallogik (LTL – *LinearTemporalLogic*)zubeschreiben.SowerdenimerstenSchrittderModellübe r-prüfungmitLTLdiegefordertenEigenschaftendesProgramms,alsoseineSpezifikation,in Formelnd erLTL -Logiküberführt.ImnächstenSchrittwerdendieLTL -Formelnunddas ProgramminjeweilseinenAutomatenübersetzt,wiein Abbildung 3verdeutlichtist.MitHi l-federAutomatentheoriewirddanndasSystemver ifiziert,d.h.überprüft,obesdenSpezifik a-tioneng enügt.

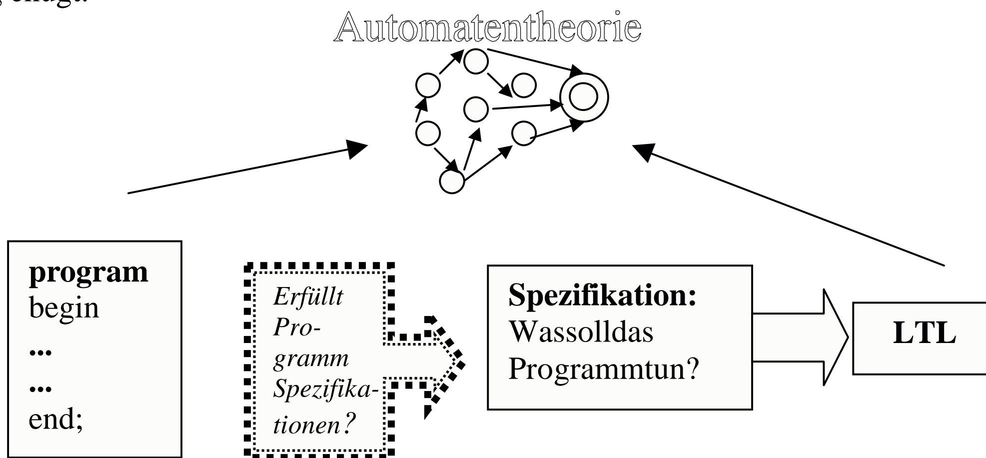

Abbildung 3: Prinzipder *LTL-Modellüberprüfung*:UmwandelndesProgrammsundderSp e-zifikationinjeweilseinenAutomatenundVerifikationmittelsderAut omaten-theorie.

IndenfolgendenAbschnittenwirdnacheinemkurzenAbschnittüberBe griffsklärungzuerst aufdielineareTemporallogik,dannaufdiezuverwendeteAutomatentheorieundaufdieV e-rifikationdesProgrammsanhandderSpezifikationeingegangen.

2 GrundlegendeBegriffe

EsfolgteinekurzeZusammenstellunggrundlegenderBegriffevonLogikundAutomatenth e-orie.

Aussagenlogik(***propositional logic*****,PL)**
DieAussagenlogikkennt(elementare)Aussagen,dieentwederwahroderfalschsind.Fo r-melnkönnend urchdiebinärenKonnektoren $\wedge$ (log. *und*)und $\vee$ (log. *oder*)sowiedenu n-ärenOper ator $\neg$ (log. *nicht*)aufgebautwerden.DaeineAussagenurwahroderfalschsein kann,lässtsichimmerentsch eiden,obeineFormel –aufgebautausbeliebigvielenverknüp f-tenAussagen –wahroderfalschist.DieAussagenl ogikistalso *entscheidbar*.
Belegung(Interpretation)
EineInterpretationisteineAbbildungderMengederSymboleindieMengederWahrheit s-werte{W,F}={wahr,falsch}={true,false}.
Erfüllbarkeit(***satisfiability)***
EineFormelisterfüllbargenaudannwennesmindestenseineBelegungfürdieFormelgibt, sodassdieFormelwahrist.

Modell(***model*****)**
ZulässigeBelegungfüreineFormel,unterderdie sewahrwird.
Tautologie
FormeldiefürjedemöglicheBelegungimmererfülltist.
Leerheitsproblem
Fragestellung,obeineMengeleerist;insbesonderevonBedeutunginderAutomate ntheorie: isteinevoneinenAutomatenakzeptierteSpracheleer?
Transitionssystem
EinTransitionssystemisteineStruktur $T = (S, \rightarrow)$,wobeiSeineMengevonZuständenist mit $s \in S$ und $\rightarrow \subseteq S \times S$ eineTransitionsrelation.EinTrans itionssystemlässtsichgrafisch alsGraphzeichnen,wob eidiePfeilevone inemZustandzueinemanderenZustanddurchdie Transitionsrelation(*Überführungsrelation*)gegebensind.

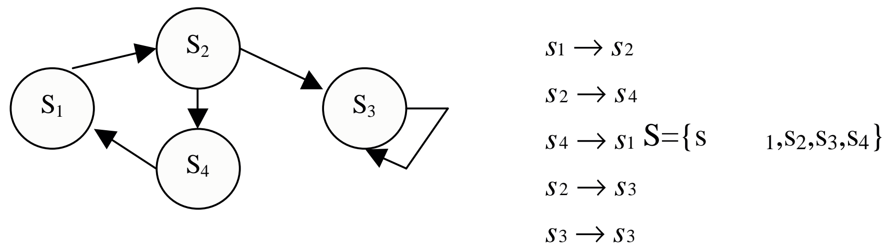

Abbildung 4: GrafischeDarstellungeineseinfachenTransition ssystemsundDarstellung deren tsprechendenRelation(inInfix -Notation).

3 LineareTemporallogik(*LTL*)

DaunserZieldarinbesteht,mithilfevonLTLnachzuweisen,dasseinProgrammseineng e-fordertenEigenschaftengenügt,müssenzuerstdiezuüberprüfendenEigenschaftenintemp o-rallogischeFormelnüberführtwerden,wobeidieTemporallogikunsdabeidieMöglichkeit gibt,zeitlicheAbhängigkeitendarzustellen.
Wirwollenunszunächstinformellder *linearenTemporallogiküberderStrukturderAuss a-genlogik*(*lineareaussagenlogischeTemporallogik* , *linear-timepropositionaltemporall ogic, LTL*)nähern.
Inder *Aussagenlogik*lassensicheinzelnebzw.verknüpfteAussagenausdrücken.Seienbe i-spielsweisediefo lgendenumgangssprachlichenAussagengegeben:

(a) Esregnet.
(b) Esregnet,alsobraucheicheinenSchirm

DannkannmandieseinaussagenlogischeFormelnpacken:DabeisolldasSymbolRfürdie Aussage„esregnet“unddasSymbolSfürdieAussage„ichbrauche einenSchirm“stehen. SomitergebensichdielogischenFormeln:

(a) R
(b) R=>S(„ *RimpliziertS* “„ *ausRfolgtS* “)

ImfolgendensolldieZeitals **diskret**betrachtetwerden,d.h.demZeitpunkt$t_j$folgtunmitte l-barderZeitpunktt_{j+1}.DieFormelnkönnenzueinem bestimmtenZeitpunkt$t_i$wahroderfalsch sein:wahrwennesin$t_i$regnet,falschzueinemZeitpunktt_i,andemesnicht(mehr)regnet. DaabertypischerweisederRegenauchirgendwannaufhörtundesZeitabschni ttegibt,an denenesnichtregnet,möchteman auchAussagentreffenkönnenübereinenbestimmten, möglicherweiseoffenenZeitabschnitt.VielleichtmöchtemanauchVorsorgetreffenunde i-nenSchirmkaufen,wennmanweiß,dassesirgendwanninderZukunfteinenZeitabschnitt gibt,währenddessenesre gnet.

AlleinmitderAussagenlogikkannmanindiesenFällenaberkeineAussagemehrtreffen.Ihr fehltdie *zeitliche*DimensionderAussagen.IneinerdiskretenTemporallogikwiederline aren TemporallogikwirdeineAussageunterderBedingungeinesZeitpunk tesbetrachtet:esha n-deltsichnichtmehrumeineabsoluteAussage,sonderninder *linearen*Temporallogikum einelineareFolge(*sequence*)derAussagezudenjeweiligenZeitpunkten.In Abbildung 5ist diesanhanddesobigenBei spielsve rdeutlicht.

... R[i]R[i+1]R[i+2]R[i+3] **...** R[i+m]R[i+m+1]**...**
wwwfff

Abbildung 5: BeispielfürdieatomareAussageR(„ *esregnet“)* über eine(dis krete)Zeit:zudenZei tpunkteni,i+1,i+2istdie Aussagewahr(„w“).

3.1 SyntaxundSemantik

LTLbringtdiezeitlicheDimensioninaussagenlogischeFormeln:diederLTLzugrundeli e-gendenDatenstrukturistdieBoolescheAlgebra.Jedeau ssagenlogischeFormel isteineLTL -Formel.DarüberhinauswerdenOperatorenzurVerfügunggestellt,mitderenHilfesichAu s-sagenüberdieZeitformulierenlassen.DieNotation,insbesonderefürtemporaleOperatoren erfolgthiernach[HR].
SeipeinebeliebigeLTL -Formel, σ eineFolgevonBelegungen $\sigma := (\sigma_0, \sigma_1, \sigma_2, ...)$fürdie Formelp,danngilt:

$(\sigma, i) \models p$ genaudann,wenn $\sigma_i \models p$ (füri $\in$ MengedernatürlichenZahlen)

Definition 1: Modell

EswirdalsodieFormelpzumZeitpunktierfüllt,wenndieBelegungzumZeitpunkti(d.h. σ_i)dieFo rmelerfüll t.ImFall(σ,0)|=pschreibtmanauchkurz $\sigma \models p$ undsprichtdavon, dassdasModell σ dieFormelperfüllt.
FürdiebooleschenOperatoren $\neg$ und $\vee$ gilt:

$(\sigma, j) \models \neg p$ genaudann,wenn $(\sigma, j) \not\models p$
$(\sigma, j) \models p \vee q$ genaudann,wenn $(\sigma, j) \models p$ oder $(\sigma, j) \models q$.

Definition 2: NegationundDisjunktion

DieOperatorenfürdaslogischeUnd,fürdielogischeImplikationundÄqu ivalenzlassensich ausden *deMorganschen*Regelnleichtableiten.

	σ_0	σ_1	σ_2	σ_3
P	**w**	**w**	**f**	**f**
Q	**w**	**f**	**w**	**f**
$p \wedge q$	**w**	**f**	**f**	**f**
$p \vee q$	**w**	**w**	**w**	**f**

Abbildung 6: BeispielfürdieErfüllbarkeitvonFo r-melnbei5Zeitschri ttenvon σ.

WirwollennungrundlegendeOperatorenderlinearenTemporallogikbe trachten.Dazugilt, dasspeinebeliebigeLTL -Formelsei.
UmAussagenüberdenZustandimnächstenZeitpunktzutreffen,gibtesden **NEXT** -Oper a-tor ❍. ❍pwirderfüllt,fallsimnächstenZustand(zumZeitpunktj+1,fallszumZeitpunktj betrachtet)perfülltist.

$$(\sigma, j) \models \bigcirc p \text{ genau dann, wenn } (\sigma, j+1) \models p$$

Definition 3: NEXT -Operator

Betrachtenwirnundennächstenwichtigen Operator,den **UNTIL -** OperatorU.A nschaulich giltdieFormelpUq(pundqlinear -temporallogischeFormeln)bezüglichdesZeitpun ktesj, wenndieFormelpfüralleZeitpunktegilt,bisaussc hließlichzueinemZeitpunktk,andemq gilt.EswirdkeineAuss agedarübergetroffen,obqauchnachdemZeitpunktknochgiltoder nicht.DieSemantikistfolgendermaßendefiniert:

$$(\sigma, j) \models p\,U\,q \text{ genau dann, wenn } \exists k \geq j \text{ mit } (\sigma, k) \models q \text{ und } \forall j \leq i < k : (\sigma, i) \models p$$

Definition 4: UNTIL -Operator

DiefolgendeAbbildungveran schaulichtdiesebeidenOperatorenanhandeinesBeispiels.

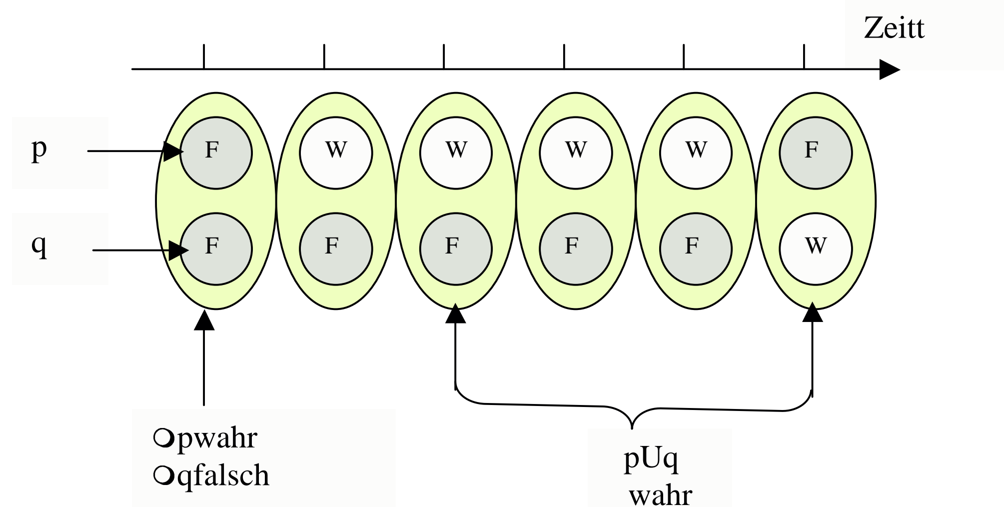

Abbildung 7: BeispielfürdietemporalenOperatoren ❍(NEXT)undU(UNTIL).

Häufigmöchtemanauchausdrücken,dassfürallefolgendenZeitpunkteeineFormelwahrist bzw.abgeschwächt,dassesabeinembestimmtenZei tpunktmindestenseinenZeitpunktgibt, andemdieFormelwahrist(natü rlichohneexplizit diesenZeitpunktzunennen).Aufgrund derHäufigkeitderB enutzungdieserMöglichkeiten,gibteshierfürjeeineneigenenOperator: **EVENTUALLY**(mitdemSymbol ◊)und **ALWAYS**(mitdemSymbol □).BeideOperat oren lassensichaberdirekt mithilfedes UNTIL -Oper atorsausdrücken.Somitergibtsich:

$$\Diamond p := true\,U\,p \quad (\textbf{EVENTUALLY})$$
$$\Box p := \neg(\Diamond \neg p) \quad (\textbf{ALWAYS})$$

Definition 5: ALWAYSund EVENTUALLY,aus UNTILabgeleitet

DamitergebensichfürdieDefinitionenvon ◊ und □:

$$(\sigma, j) \models \Diamond p \text{ genau dann, wenn } (\sigma, k) \models p \text{ für mindestens ein } k \geq j$$
$$(\sigma, j) \models \Box p \text{ genau dann, wenn } (\sigma, k) \models p \quad \forall k \geq j$$

Definition 6: ALWAYSund EVENTUALLY: direkteDefinition

3.2 VergangenheitsbezogeneOperat oren

DiebisherbetrachtetenOperatorenbetrachtenZeitpunkte,diebezüglichderaktuellenPosit i-oninderZukunftliegen,anschaulichalsobezüglicheinerweiterrechtsliegendenPositionauf derZeitachse.SeidieaktuellePositionj,dannbezie htsicheinzukunftsbezogenerOperator

aufZuständej,j+1,j+2,j+3,...;demgegenüberbetrachteteinvergangenheitsbezogenerOper a-torZuständej,j -1,j -2,...0.
AnalogderOperatorendesvorigenAbschnittssollenhierkurzihreäquivalentenvergange n-heitsbezogenenOperatorenvorgestelltwerden.Seifürdasfolgendewiederangenommen, dasspundqlinear -temporallogischeFo rmelnseien.
Umauszudrücken,dassdieAussagepzumvorigenZeitpunkwahrwar,b enutztmanden **PREVIOUS**-Operator ⊙:

$$(\sigma, j) \models \odot p \text{ genau dann, wenn } (\sigma, j-1) \models p \ (j>0)$$

Definition 7: Der PREVIOUS-Operator

Damanvoneinerdiskreten,mitdemZeitpunkt0anfangendenZeitausgeht,mussj>0sein.
Analogzum UNTIL-Operatordrücktder **SINCE-**Operatoraus,dasseineFormelqzue inem vergangenenZeitpunktwahrunddieFormelpseitdemZeitpunkt,andemqdasletzteMal wahrwar,biszumgegenwärtigenZeitpunktwahrist.DiesenSachverhaltdefiniertmanfo r-malwiefolgt:

$$(\sigma, j) \models p\,S\,q \text{ genau dann, wenn } \exists k, 0 \le k \le j \text{ mit } (\sigma, k) \models q$$
$$\text{und } \forall i: k<i \le j \text{ mit } (\sigma, i) \models p$$

Definition 8: SINCE -Operator

MitHilfedesSINCE -Operatorslassensich –analogderDefinitionderzukunftsbez ogenen Operatoren ALWAYSund EVENTUALLYmithilf evon UNTIL –dieOperatoren **HAS-ALWAYS-BEEN** ⊡und **ONCE** ⟐definieren. WirwollensiehieraufgrundeinererhöhtenVerständlic h-keitüberihreinte ndierteSemantikdefinieren:

ONCE: $(\sigma, j) \models ⟐ p$ genau dann, wenn $(\sigma, k) \models p$ für mindestens ein $0 \le k \le j$
HAS-ALWAYS-BEEN: $(\sigma, j) \models ⊡ p$ genau dann, wenn $(\sigma, k) \models p \quad \forall\, 0 \le k \le j$

Definition 9: ONCE- und HAS-ALWAYS-BEEN-Operator

3.3 StrikteOperatoren

DiebisheraufgeführtenOperatoren,diesichnichtnurdirektaufdennäch stenZeitpunktj+1 wiebeispielsweise ❍ und ⊙beziehen,benutzeninihrerDefinitionZeitpunkteiundk,diein schwacher Relation(≤bzw. ≥)zumReferenzzeitpunktjstehen.Dasbedeutetaber,dassder Zeitpunktj,alsoderaktuelleZeitpunktbzw.dergegenwärtigeZeitpunkt,sowohlTeilder ZukunftalsauchderVergangenheitist:bet rachtemandazudieOperatoren ◊ und ⟐sowie dieelementarlogischeAussagep.SeifernerideraktuelleZeitpunkt,und σ weisejedem Zeitpunktjmitj ≠ idenWahrheitswertwahrzu.Sogilt,dass(σ,j)|= ◊ pund(σ,j)|= ⟐p.
UmdieDoppelzugehörigkeitdesgegenwärtigenZeitpunktszuvermeiden,könnenanalogder obigenDefinitionendieOperatorenineinerstriktenVe rsiondefiniertwerde,indemstattder Relationen ≤bzw. ≥dieRelationen<bzw.>verwendetwerden.Andererseitsreichtdasbi s-herigeOperatorenrepertoire(insbesonderedurchVerwendungvon ❍bzw. ⊙)aus,umdiese striktenOperatorenmitihrerHilfezudefinieren.EinigeBeispiele sindinfolgenderAufzä h-lunggegeben:

$$p\,U^{strikt}\,q = ❍(p\,U\,q)$$
$$◊^{strikt}p = ❍◊p$$
$$□^{strikt}p = ❍□p$$
$$⟐^{strikt}p = \odot⟐p$$

Abbildung 8: Beispiele,wiediestrikteVersionderOperat o-renmitschonbekan ntenOperat orenausgedrücktwerdenkann

3.4 LTL,CTLundCTL*

FürModellüberprüfungundinsbesonderefürdieImplementierungvonModellüberprüfern benutztmanheutenebenderTemporallogikLTLauchdieTemporallogikenCTLundCTL*. BeiCTL(*ComputationTree Logic*)könnenFormelnnichtnurEigenschafteneinerSequenz beschreiben,sondernesbestehtdieMöglichkeitAussagenüberverschiedeneFolgezustände bzw.Folgepfadezutreffen.DieZeitwirdnichtwiebeiLTLalslinearangesehensondernals verzweigteZeit,wobeiinnerhalbeineseinzelnenPfadesdieZeitlinearist.Dafürliegenin CTLentsprechendePfadquantorenvor,aberdietemporallogischenOperatorenkönnennicht beliebiggeschachteltwerden.CTL*isteineKombinationvonLTLundCTL:esgibtPfad-quantorenundTemporaloperatoren.DabeikönnenPfadquantoren,Temporaloperatoren *und* boolescheVerknüpfungen *beliebig*geschachteltwerden. Abbildung 9zeigtdenZusammen-hangzwischenCTL*undLTLbzw.CTL.DabeikönnendieAussagenvonCTLbzw.CTL* folgendermaßen„übersetzt“werden:

- AGEFp:injedemZustandentlangallerPfadegibteseinenFolgezustand, indempwahrist.
- AG(p → AFq):jedempfolgtirgendwanneinq.
- E(GFp)esgibteinenPfad, aufdempunendlichoftwahrwird.

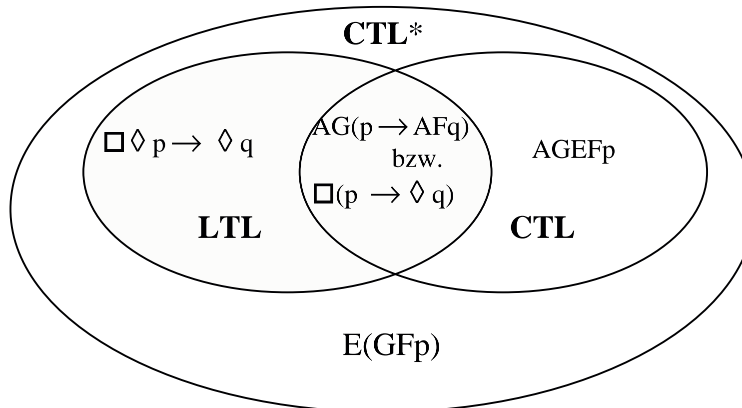

Abbildung 9:CTL*bildetObermengevonLTLundCTL.EsgibtAussagen,dieinLTL abernichtinCTLundAussagen,dieinCTL abernichtinLTLdargestelltwerdenkönnen.

4 Automaten

ImletztenAbschnittwurdeLTLvorgestellt,mitderenHilfeesmöglichist,dieSpezifikation einesProgrammsmittemporallogischeFormelnauszudrücken.DadiehiervorgestellteVer-ifikationdesProgrammsmitGrundlagenderAutomatentheoriedurchgeführtwird,mussdiein LTLformulierteSpezifikationineinenAutomatenüberführtwerden.ImfolgendenAbschnitt sollendierelevantenBereichederAutomatentheorievorgestelltwerden.

4.1 EndlicherAutomat(deterministisch&nichtdeterministisch)

Vorderformalen DefinitioneinesAutomatenseizunächsteinAutomatinformellbetrachtet. Betrachtetmandenin Abbildung 10alsgerichtetenGraphendargestelltenAutomaten,sogilt: DieKnotendesGraphenbezeichnendieZustände(s_1,s_2,s_3,s_4),davonsinds_1 bzw.s_3 Anfangs-bzw.Endzustand.DiegerichtetenPfeilesindmitSymbolenmarkiertundgebendie Möglichkeitan,voneinemZustand$s_i$ineinenFolgezustands_j(wobeinichtnotwendigerweise $s_i \neq s_j$seinmuss,vgl.s_3)zu gelangen.DerAutomatakzeptiertgenaudieWorte,diesichdurch KonkatenationderSymbole$a_i$ergeben,diesichaufeinemPfadvoneinemAnfangs-zueinem

Endzustandbefinden,wobeisichdieReihenfolgederSymboledurchdieReihe nfolgeder SymboleimPfa dergibt,wiebeispielsweisedieWo rteac,abbac,acccc.

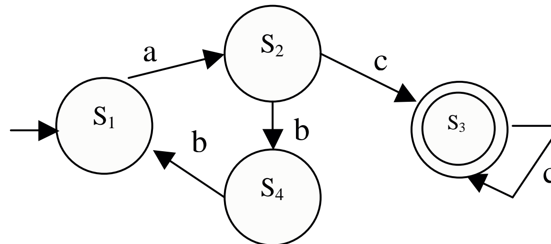

Abbildung 10: GerichteterGraphals(endlicher,deterministischer)A u-tomat

Formalistein(deterministischerundnichtdeterministischer) *endlicher Automat*Aein5er TupelA=(Σ,S,S_0, δ,F),wobei Σ dasendlichenichtleereAlphabet,Sdieendlichenichtle e-reMengevonZuständen,$S_0 \subseteq$ SdieMengederStartzustände, δ:S $\times \Sigma \rightarrow 2^S$eine TransitionsfunktionundF $\subseteq$ SdieMengederEndzuständeist.

Einendlicher *deterministischer*Automatunterscheidetsichvoneinem(endlichen)nich tde-terministischenAutomatendahingehend,dassesnureinenAnfangszustand(d.h.$|S_0|=1$) undesfürjedesSymbolinjedemZustandmaximaleinenFolgezustandgibt(d.h.$|\delta(s,a)| \leq 1$, $\forall s \in S$und $\forall a \in \Sigma$). DerGrapheinessolchenAutomatenistin Abbildung 10gegebenmit $\Sigma=\{a,b,c\}$,$S=\{s_1,s_2,s_3,s_4\}$,$S_0=\{s_1\}$,$F=\{s_3\}$, δ: $\delta(s_1,a)=\{s_2\}$, $\delta(s_2,b)=\{s_4\}$, $\delta(s_4,b)=\{s_1\}$, $\delta(s_2,c)=\{s_3\}$, $\delta(s_3,c\}=\{s_3\}$.
Damitgiltfüre inen(endlichen) *nichtdeterministischen*Automaten,dassermehrereStartz u-ständehabenkannunddieTransitionsfunktion δ mehreremöglichenFolgezuständefürjedes Symbolerlaubenkann.
DerLaufr(engl. *run*)einesWortesw=$a_0 a_1 a_2 a_3 a_4 ... a_{n-1}$ ($a_i \in \Sigma$)isteineFolgevonnZ u-ständen$s_0,s_1,s_2,...s_n$,wobei$s_0$einStartzustandist($s_0 \in S_0$)und$s_{i+1} \in \delta(s_i,a_i)$(für$0 \leq i<n$).Der Laufwirdakzeptiert,falls$s_n \in F$undeinWortwwirdvonAakzeptiert,fallsese inenakze p-tierendenLaufrauf Agibt.FüreinendeterministischenAutomatenkannesnureinenakze p-tierendenLauffüreinbestimmtesWo rtwgeben,da$|S_0|=1$und$|\delta(s,a)| \leq 1$.DieSpracheL vonA(L(A))istdieMengeallerendl ichenWorte,dievonAakzeptiertwerden.
ImfolgendensindeinigewichtigeEigenschaftenderendlichenAutomatenbez üglichderA b-geschlossenheitunterBooleschenOperationenaufgeführt.DieelementarenBeweisesindbe i-spielsweisein[Vardi]zufinden.

SeienA_1undA_2endlicheAutomaten.

(1) Abgeschlossenheitunter **Vereinigung**:
DannexistierteinendlicherAutomatAmitL (A)=L(A_1) $\cup$ mitL(A_2)
(2) Abgeschlossenheitunter **Durchschnitt**
EsexistierteinAutomatAmit:L(A)=L(A_1) $\cap$ L(A_2)
(3) Abgeschlossenheitunter **Komplementbildung**
EsexistierteinAutomat $\overline{A}$ mitL($\overline{A}$)= Σ^*-L(A).

EinAutomatgiltalsnichttrivial,wenndiedurchihnerzeugteSpracheL(A)wederleerist nochallemöglichenWorteüberseinemAlphabet Σ akzeptiert.EinefundamentaleFrages tel-lunginderAutomatentheorieistdieFragenachderNichtleerheitundnachderNichtunivers a-lität.

DasNichtleerheitsproblemstelltdieFrage,obdieSprachevonAleerist,dasNichtuniversa l-problemstelltdieFrage,obL(A) $\neq \Sigma^*$ist.DieBeweisefürdiefolgendenAntwo rtenkönnen in[Vardi]gefundenwerdenundwerdenhieralsErgebnissepräsentiert.

(1) **Nichtleerheitsproblem**:
   (a) Das Nichtleerheitsproblem ist in *linearer* Zeit *entscheidbar*
   (b) Das Nichtleerheitsproblem ist NLOGSPACE -vollständig
(2) **Universalitätsproblem**
   (a) Das Universalitätsproblem ist für Automaten in *exponentieller* Zeit *entscheidbar*
   (b) Das Universalitätsproblem für Automaten ist PSPACE –vollständig

Die Bildung von Vereinigung, Durchschnitt und Komplement sowie das Nichtleerheitsproblem werden bei der Überprüfung, ob ein Programm seine Spezifikation erfüllt, wichtig sein.

4.2 Büchi Automat

Der bisher betrachtete Typ eines (deterministischen bzw. nichtdeterministischen) Automaten akzeptiert nur endliche Worte $w = a_1 a_2 a_3 a_4 \ldots a_{n-1}$ ($a_i \in \Sigma$): nach dem vollständigen Abarbeiten von w erreicht der Automat einen Endzustand (w akzeptiert) oder nicht (w nicht akzeptiert). In einer LTL -Formel können Aussagen über einen zukünftigen, aber unbestimmten Zeitpunkt (z.B. durch den Operator ◊) getroffen werden. Ebenfalls müssen typischerweise bei Programmen Eigenschaften während der gesamten, aber in vielen Fällen unbestimmten Laufzeit eingehalten werden. Deshalb betrachten wir nun einen Automaten über unendlichen Worten (einen sog. *Büchi-Automaten*). Dabei muss eine neue Akzeptanzbedingung gefunden werden: da w unendlich ist, ist es nicht möglich, den letzten Zustand, den der Automat erreicht, zu betrachten und zu entscheiden, ob er ein Endzustand ist.
Formal betrachtet ist auch der Büchi -Automat ein 5er Tupel $A = (\Sigma, S, S_0, \delta, F)$, wobei die einzelnen Tupelkomponenten dieselben wie beim endlichen Automaten (siehe Abschnitt 4.1) sind. Der Lauf r des Wortes w ist aber eine (unendliche) Sequenz von Zuständen $s_0, s_1, s_2, \ldots$ mit $s_0 \in S_0$ und $s_{i+1} \in \delta(s_i, a_i)$ für alle $i \geq 0$. Ferner betrachten wir das Limit des Laufes r: $\lim(r) = \{s \mid s = s_i$ für unendliche viele i's$\}$. Also ist $\lim(r)$ die Menge der Zustände, die sich in r unendlich oft wiederholen. Da die Zustandsmenge S endlich ist und w unendlich, kann diese Menge nicht leer sein. Ein Lauf r wird vom Automaten akzeptiert, wenn es einen Zustand aus der Menge F gibt, der sich in r unendlich oft wiederholt (d.h. $\lim(r) \cap F \neq \emptyset$). Damit gilt wiederum: ein (unendliches) Wort w wird von A akzeptiert, wenn es einen Lauf r gibt, welcher von A akzeptiert wird. Die Menge aller unendlichen Worte w, die von A akzeptiert werden, bildet die von Automaten akzeptierte Sprache $L(A)$.
Betrachten wir nun den Büchi -Automaten A_1 von Abbildung 11: Bei der Repräsentation des Automaten als Graph kann dieser ebenso als Graph für einen endlichen Automaten aufgefasst werden, doch beachte man den Unterschied, dass A_1 nur *unendliche* Worte akzeptiert, so wird der Unterschied in der *Semantik* klar: er akzeptiert nur (unendliche) Worte über dem Alphabet $\Sigma = \{a, b\}$, die unendlich oft das Symbol „a“ enthalten. Dabei stellt der doppelt umrandete Zustand s_1 auch keinen Endzustand im konventionellen Sinne dar, vielmehr muss dieser Zustand *unendlich oft* durchlaufen werden.

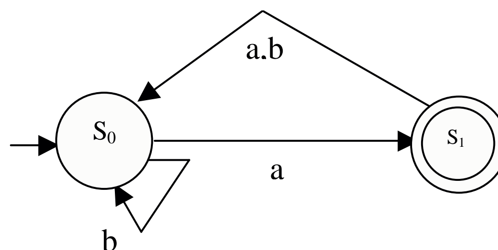

Abbildung 11: Beispiel eines (deterministischen) Büchi–Automaten.

DerBüchi–AutomatA_2aus Abbildung12akzeptiertunendlicheWorte,die abirgendeiner Positionunendlichoftausdem Symbol„b“bestehen.DaalleZustandsübergängeauchalleine durchdasSymbol„b“ausgelöstwerdenkönnen,istA_2nichtdeterministisch.

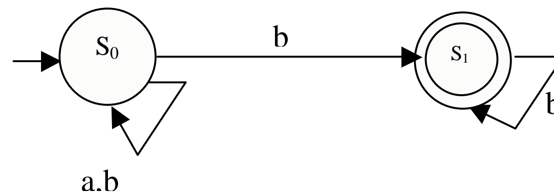

Abbildung 12: nichtdeterministischerBüchi-Automat:er beschreibtWorte,dieirgendwannunendlichoftnurnochausdemSymbol„b“bestehen.

FernerlässtsichandiesemAutomatendieMöglichkeiteinerÜberführungeinerLTL-Formel ineinensolchähnlichenAutomatenerahnen:erhateineVerwandtschaftmiteinerlineartemporalenFormel,beiderabeinemzukünftigenunspezifiziertenZeitpunktdieAussage„b“ gilt.

AuchdieBüchi-AutomatensindunterdenbooleschenOperationenDurchschnittsbildung, KomplementbildungundVereinigungabgeschlossen.DienotwendigenBeweisedazusindein wenigaufwendigundimfolgendennichtaufgeführt.DerinteressierteLeserkanndieBeweise zurKomplementärbildungin[Büchi],zurDurchschnittsbildungundVereinigungin[Choueka]studieren.
AuchfürdieAntworten aufdasNichtleerheitsproblemunddasUniversalitätsproblemgilt AnalogeswiefürdieAutomatenüberendlichenWorten.

(1) **Nichtleerheitsproblem**:
   (a)DasNichtleerheitsproblemistin *linearer*Zeit *entscheidbar*
   (b)DasNichtleerheitsproblemistNLOGSPACE-vollständig

(2) **Universalitätsproblem**
   (a) DasUniversalitätsproblemistfürAutomatenin *exponentieller*Zeit *entscheidbar*
   (b) DasUniversalitätsproblemfürAutomatenistPSPACE-vollständig

4.3 AlternatierenderBüchi-Automat

IndiesemAbschnittwerdenzurVorbereitungderÜberführungeinerLTL-Formelineinen Automaten *alternierendeBüchi-Automaten*(*alternatingBüchi*)vorgestellt.DieUmformung ineinenalternierendenBüchi-Automatenwirdhiergewählt,weilerdieTransformationeiner LTL-Formelerleichtert.
Umdas PrinzipeinesalternierendenBüchi-Automatenzuverstehen,seiimfolgendendies anhandeinesAutomatenüber *endlichen*WortenalleinedesVerständniswegengenauerbetrachtet.DanachübertragenwirdasErgebnisaufBüchi-Automaten.
SeiXeineMengevonFormeln,dannbezeichne$B^+(X)$dieMengeallerbooleschenFormeln, dieausElementenvonXsowiedenVerknüpfungen $\wedge$und $\vee$sowiedenFormeln *true*und *false* gebildetwerdenkönnen.SoistbeispielsweisedieFormel $a \wedge b \vee c \in B+(\{a,b,c\})$,aber dieFormel $\neg a \wedge b \wedge \neg c \notin B^+(\{a,b,c\})$,daderunäreOperator $\neg$nichterlaubtist.Aus diesemGrundnenntmanübrigensdieElementevon$B^+(X)$auch *positive Formeln.*Nach [Vardi]erfüllteineTeilmengeYvonXeineFormel$a \in B^+(X)$,wenndenElementenausXder

Wahrheitswert *true*unddenElementenderDifferenzmengeX -YderWert *false*zuge wiesen wirdunddieElementevonXdieFormelerfüllen.
BetrachtenwirdiesenSachverhaltanhandeinesBeispiels:dazuseiX={s1,s2,s3,s4}und dieUntermengenvonXY1={s1,s3},Y2={s1,s4}undY3={s1,s2}s owiedieFormela= $(s1 \vee s2) \wedge (s3 \vee s4)$gegeben.BetrachtenwirzuerstY1:dannwirdjedemElementausY1 *true*zugewiesen,jedemElementausderDifferenzme ngeX -Y *false*(d.h.s1=true,s3=true,s2 =false,s4=false).DieFormelawirdde mnachvonY1erfüllt.Dasgle ichegiltfürY2aber nichtfürY3.

MitpositivenFormelnsollnundieTransitionsfunktion δ einesnichtdeterministischenAut o-matenausgedrücktwerden.BeieinemsolchenAutomatenistesjamöglich,dasses verschie-deneZustandsübergängefüreinSymbola $\in \Sigma$ auseinemZustands $\in$ SineinenFolgezustand gibt.Sokannman $\delta(s,a)=\{s1,s2,s3\}$auchfolgendermaßenschreiben $\delta(s,a)=s1 \vee s2 \vee s3$.
Beialte rnierendenAutomatengehtmanabernocheinenSchrittweiter:aufderrechtenSeite derTransitionsfunktionkann *jedebeliebige* Formelaus $B^+(X)$stehen.EinAusdruckwi ebe i-spielsweise $\delta(s,a)=(s1 \wedge s2) \vee (s3 \wedge s4)$besagt,dassderAutomatdasWortaw(a $\in \Sigma$, w$\in \Sigma^*$)akzeptiert,fallsermomentanimZustandsistundsowohldasWortwvons1unds2 odervons3unds4akzeptiert.

DieBerechnungeinesWorteswdurcheinenalternierendenAutomatenistnichtmehreinl i-neareSequenzsonderneineBaumstruktur.EinB aum(*tree*)tisteingerichteterGraphmit einemausgezeichnetenWurzelknoten ε undKnoten,fürdiegilt,dassjederNichtwurzelkn o-tengenaueinenVaterknotenhabenmuss(ε hatalsokeinenVaterknoten).Inder GraphdarstellunghateinKnotenk $_1$einenVaterknotenk $_2$,fallseinegerichteteVerbindung vonk $_2$zuk $_1$vorhandenist.EinZweig(*branch*)bisteineFolgevonKnotenx $_0$ $(=\varepsilon)$,x $_1$,x $_2$, x$_3$,....,wobeix $_i$derVaterknotenvonx $_{i+1}$ fürallei ≥ 0ist.Ein Σ-markierterBaum(Σ endlich) isteinPaar(t,r)mitBaumtundreineFunktion,diejedemKnotenvonteinSymbolaus Σ zuweist.Sod efinierteinZweigx=x $_0$,x $_1$,x $_2$,...mitreinunendlichesWortr(w)=r(x $_0$),r(x $_1$), r(x$_2$),...

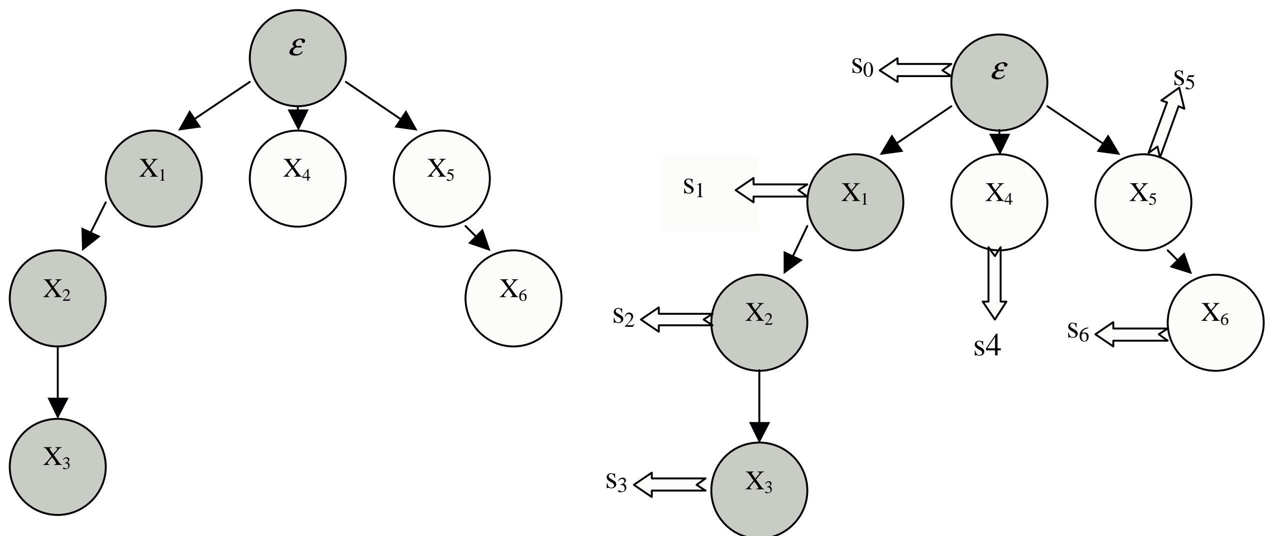

Abbildung 13: BeispieleinesBaumesmitausgewähltenZweig(grauschattierteKnoten)sowiedieser Baumals Σ-markierter Baummit $\Sigma=\{s_1,s_2,s_3,s_4,s_5,s_6\}$.

SeinundasBeispielausAbbildung13betrachtet.DerBaumbestehtausKnotenx $_i$.Der Zweigz(grauunterlegteKnoteninderAbbildung)bestehtausdenKnoten ε,x $_1$,x $_2$undx $_3$. x$_4$g ehörtnichtindieseFolge,dadieBedingung,dassx $_3$Vaterknotenvonx $_4$ist,verletztist.

DerrechtsinderAbbildungzusehende Σ–markierteBaum($\Sigma=\{s_i \mid i \geq 0\}$istdurchdieFun k-tionrmitr(x_i)=$s_i$entstanden.DamitdefiniertrfürdenZweigzdasWortr(z)=r(x_0)r(x_1) r(x_2)r(x_3)=$s_0 s_1 s_2 s_3$.
FormalkommenwirdamitzusammenfassendzufolgenderDefinition.DieBerechnungeines Wortesw=$a_0 a_1 a_2 a_3 ... a_{n-1}$isteinendlicherS -markierterBaumrmitr(ε)=$s_0$undesgiltfür einenKnotenx:wenn|x|=i<nundr(x)=sund δ(s,a_i)= ξ,dannhatderKnotenxk -viele Kinderx $x_1,...,x_k$ k $\leq$|S|und{r(x_1),r(x_2),..,r(x_k)}erfülltdieFormel ξ.(Mit|x|istdieDistanz desKnotensxvomWurzelknoten εgemeintmitr(ε):=0.)

ÜbertragenwirdieseDefinitionaufBüchi -Automaten,alsoaufAutomaten,w elcheunendl i-cheWorteakzeptieren,sogilt:
SeiA=(Σ,S,S_0, δ,F)einalternierenderBüchi -Automat.EineBerechungvonAübereinem unendlichenWortw=$a_0 a_1 a_2 a_3$,...istein(möglicherweiseunendlicher)S -markierterBaumr mitr(ε)=$s_0$undesgilt:wenn|x| $\leq$i,r(x)=sund δ(s,a_i)=dannhatderKnotenxk -viele Kinderx $x_1,...,x_k$k $\leq$|S|und{r(x_1),r(x_2),..,r(x_k)}erfülltdi eFormel ξ.
DieBerechnungwirdakzeptiert,wennjederunendlicheZweiginrunendlichvieleBene n-nungenausFenthält.

AlternierendeBüchi-AutomatensindvollständigäquivalentzunichtdeterministischenB üchi-Automaten:Jedernich tdeterministischeBüchiAutomatkannineinenalternierendenBüchi -Automatenüberführtwerden(denBeweiskannderinteressierteLeserin[MS87]nac hlesen), undfürjedenalternierendenBüchi –AutomatengibteseinennichtdeterministischenB üchi-Automaten(Beweissiehehierzuin[Miyano]).

5 LTLundAutomaten

IndiesemAbschnittwirdgezeigt,wieeineModellüberprüfungdurchgeführtwerdenkann, d.h.wieestechnischmöglichist,mitdenindenvorigenAbschnitteneingeführtenMethoden undGrundlagenzuzeigen ,dasseinProgrammseineSpezifikationenerfüllt.DahierModel l-überprüfungtechnischmitGrundlagenderAutomatentheoriedurchgeführtwirdunddieSp e-zifikationeneinesProgrammsmitLTLfestgelegtwurden,seialsweiteresHilfsmitteldie Ü-berführungein erLTL -FormelineinenBüchi -Automatenbetrachtet.

5.1 Spezifikation

EineSpezifikationbeschreibtdieEigenschaften,dienotwendigerweisevondenjeweiligen Programmenerfülltwerdenmüssen.TypischerweisewerdenzuerstdieSpezifikationenin umgangssprachlicheBedingungengefasst,bevorsieinFormelnderLTLüberführtwe rden.

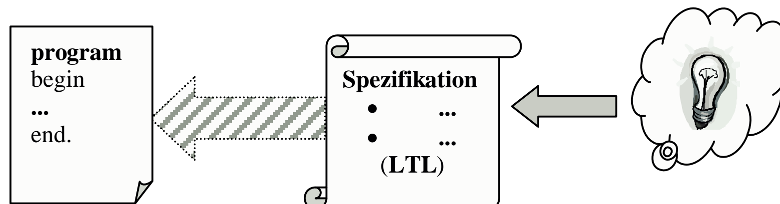

Abbildung 14: DieSpezifikationbeschreibtdievomProgrammiereran dasProgrammgestel ltenEigenschaften.

SeialseinfachesBeispieldergegenseitig eAusschluss(*mutualexclusion*)vonzweiProzessen betrachtet,ohnedassBlockierenausgeschlo ssenwird:

(1) IstProzessP1(bzw.P2)imkritischenAbschnitt,darfProzessP2(bzw. P1)nichtimkritischenA bschnittsein

(2) WillP1(bzw.P2)indenkritischenAbs chnitt,kommtirgendwannP1 (bzw.P2)auchindenkrit ischenAbschnitt

ÜberführunginLTL:

(1) $\Box$(P1.critical $\rightarrow$ $\neg$ P2.critical)
$\Box$(P2.critical $\rightarrow$ $\neg$ P1.critical)

(2) $\Box$(P1.request $\rightarrow$ $\Diamond$ P1.critical)und
$\Box$(P2.request $\rightarrow$ $\Diamond$ P2.critical)

5.2 ÜberführungeinerLTL -FormelineinenAutomaten

ImfolgendensolldieÜberführungeinerLTL -FormelineinenBüchi -Automatenbetrachtet werden.DiedurchdenBüchi -AutomatenakzeptierteSprachesolldabeiexaktdie Mengeder Berechungensein,welchedieLTL -Formelerfüllen.Dabeiwerdenfolgendezwei„Einschrä n-kungen“vorgenommen:

(a) ÜberführungderFormelineinenalternierendenBüchi -Automaten
(b) DieLTL -FormelbestehtnurauszukunftsbezogenenOperatoren.

DieAussage(a)kanninsoweitunberücksichtigtbleiben,dazuBeginndesletztenA bschnitt erwähntwurde,dasseszujedemalternierendenBüchi -AutomateneinenBüchi -Automaten gibt.DieEinschränkung(b)besagt,dassnurdiezukunftsbezogenenOperatoren UNTILund NEXTv orkommen.InAbschnitt 3.1wurdeabergezeigt,dassdiezukunftsbezogenenOperat o-renmitdiesenbeidenOperatorenausgedrücktwerdenkönnenunddassauchdiestriktenOp e-ratorenhierausabgeleitetwerdenkönnen.

Sei φ eineLTL -Formel.DannexistierteinalternierenderBüchi -Automat$A_\varphi=(\Sigma,S,s_0,\delta,F)$mit $\Sigma=2^{Prop}$(alsodiePotenzmengeallerin φ vorkommendenaussagenlogischenFo rmeln), $|S|=O(|\varphi|)$,sodass$L(A_\varphi)$exaktdieMengederBerechnungenist,welchedieFo rmel φ erfüllen.

EinBeweishierfürkannin[Vardi]vominteressiertenLesernäh erstudiertwerden.Imfo l-gendenbeschränkeichmichaufdieKonstruktionvon Σ,S,s_0, δ undF.

(1)Das *Alphabet* Σ istdiePotenzmengeallerin φ vorkommendenaussagenlog i-schenFo rmeln.

(2)DieZustandsmenge *S*bestehtausallenTeilformelnvon φ sowiederenNegati o-nen,d.h.$S=\{a, \neg a | a$ istTeilformelvon $\varphi\}$.

(3) $s_0 \in$ SistdieForm el φ

(4)dieMengeder *akzeptierendenZustände* $F=\{\phi = \neg(aUb) | \phi \in S\}$

(5)die *Transitonsfunktion* δ istfolgendermaßendefi niert:

$\delta(s,a)$=true,fallss $\in$ a

$\delta(s,a)$=false,fallss $\notin$ a

$$\delta(p \wedge q, a) = \delta(p,a) \wedge \delta(q,a)$$

$$\delta(\neg p, a) = \overline{\delta(p,a)}$$

$$\delta(\bigcirc p, a) = p$$

$$\delta(pUq, a) = \delta(q,a) \vee (\delta(p,a) \wedge (pUq))$$

wobeifürdenOperator $\overline{}$ gilt:

$\overline{a} = \neg a$,füra $\in$ S

$\overline{true}$ =false

$\overline{false}$ =true

$\overline{a \wedge b} = \overline{a} \vee \overline{b}$

$$\overline{a \vee b} = \overline{a} \wedge \overline{b}$$

NunseizumbesserenVerständniseinBeispielbetrachtet:SeidieFormel φ=($\bigcirc\neg$p)Uq gegeben.Danngiltfür $\Sigma=2^{\{p,q\}}$={{p,q},{p},{q},{}}.DieZustandsmengeSbestehtaus allenTei lformelnvon φundderenNegation,alsoS={ φ, $\neg\varphi$, $\bigcirc\neg$p, $\neg\bigcirc\neg$p, $\neg$p, ($\neg\neg$p=)p,q, $\neg$q},derStartzustands $_0$istdieFormel φ.DieMengeFsolljaausallen FormelnderBauart $\neg$(aUb) $\in$ Sbestehen.DahiernureineFormeldieseBauarthat,nä m-lich φselber,giltfürF={($\bigcirc\neg$p)Uq}={ $\neg\varphi$}.DieTransitionsfunktion δwirddurch folgendeTabellebeschrieben:

S| δ	δ(s,{p,q})	δ (s,{p})	δ(s,{q})	δ(s,{})
φ	True	$\neg$p $\wedge\varphi$	True	$\neg$p $\wedge\varphi$
$\neg\varphi$	False	p $\vee\neg\varphi$	False	p $\vee\neg\varphi$
$\bigcirc\neg$p	$\neg$p	$\neg$p	$\neg$p	$\neg$p
$\neg\bigcirc\neg$p	p	p	p	p
$\neg$p	False	False	True	True
p	True	True	False	False
q	True	False	True	False
$\neg$q	false	True	False	True

5.3 Verifikation

FürdieFragestellung,obeinProgrammPseineSpezif ikationenerfüllt,istesnotwendig,P folgendermaßeneinzuschränken:essseiennurProgrammebetrachtet,dieeineendlicheAnzahl vonVariablenüberendlichenWertebereichendurchlaufen.DieseArtderProg rammewerden auchals *finite-stateprograms* bezeichnet.DieseEi nschränkungistschwach,dadieMehrheit derProgramme,beideneneineModellüberprüfungmehrheitlichalswic htigundnotwendig betrachtetwird(d.h.insbesondereNetzwerkkommunikationsprotoko lleundSynchronisat i-onsmechanismenbzw. –protokolle),inderTatfinite -statePr ogrammesind.
DadieVariablenendlicheWertebereichebesitzen,kanneinProgrammzustanddurcheine VerknüpfungvonAussagenbeschriebenwerden.DieAussagenwerdendabeimith ilfeder Aussagenlogikformuliert.ZurFormalisierungeinesProgrammsbenötigtmaneineMengeW vonallenZuständendesProgramms,einenAnfangszustandw $_0\in$W.DieZustandstransitio n-relationR $\subseteq$W$\times$WbeschreibtdiemöglichenÜbergängevonZuständen,wobeiNichtdeterm i-nismusmöglichist.DieZuordnungsfunktionV:W $\rightarrow 2^{Prop}$weistinAbhängigkeitvomZ u-standdenAussagen(*Prop*seidieMengederverwendetenelementaraussagenlogis chenFo r-meln)Wahrheitswertezu,gibtalsoan,welcheAuss ageninwelchemZustandgelten.Also lässtsichdasProgrammdurcheinen4erTupelP=(W,w$_0$,R,V)beschreiben.Die ZustandstransitionsrelationRmussdabeijedenZustandineinenNachfolgezustand überfü h-ren.Dasbedeutetaberauch,dasseskeinenEndzustanddesProgrammsimengerenSinne gibt.StattdessenwirdmanEndzuständesocharakterisieren,dasseinEndz ustandsichselbst alsNachfolgezustandhat.EinZustandw $\in$Wsollte dabeitypischerweiseauchSpeicherinhalt, Registerinhalt,Programmzähler,Pufferinhaltetc.enthalten.
WennueineunendlicheSequenzvonZuständenu $_0$,u $_1$,u$_2$,u$_3$,...istundu $_0$=w$_0$,u $_i$Ru$_{i+1}$ fürallei, dannbeschreibtV(u $_0$),V(u$_1$),...einemöglicheBerechnung desProgramms.Dawirwissen wollen,obdasProgrammdieSpezifikationenerfüllt,müssenwirüberprüfen,oballemögl i-chenBerechnungendesProgrammsdieSpezifikationen(dargestelltmittelsLTL)erfüllen:Sei

φ eineinLTLformulierte geforderteSpezifikationundPdasProgramm,danngehenwirfo l-gendermaßenbeiderÜberprüfungvor:

(1) UmformendesProgrammsP=(W,w $_0$,R,V)zueinemBüchi -AutomatenA $_p$=(Σ,W,{w $_0$}, δ,W),wobeigilt: Σ=2 ^{Prop},s’ $\in$ δ(s, a)genaudann,wenn(s,s’) $\in$ Runda=V(s).
(2) AufstelleneinesBüchi -AutomatenA $_\varphi$,dergenaudieBerechnungen von φ akzeptiert.

DieFrageistnun,oballevonA $_p$akzeptiertenBerechnungenauchvonA $_\varphi$akzeptiertwe rden, d.h.obL(A $_p$) $\subseteq$L(A $_\varphi$).Äquivalentdazukanngeprüftwerden,obL(A $_p$) $\cap$L($\overline{A_\varphi}$)= $\emptyset$ist. EntsprechenddenErkenntnissenderAutomatentheorieausAbschnitt 4,könnennundas KomplementundderDurchschnittgebildetwe rdenundaufLeerheitgetestetwerden. Damitkan nleichtüberdenWegderAutomatentheorieüberprüftwerden,obeinProgramm seineSpezifikationenerfüllt,wieabschließendzusammenfassendin Abbildung 15dargestellt ist.

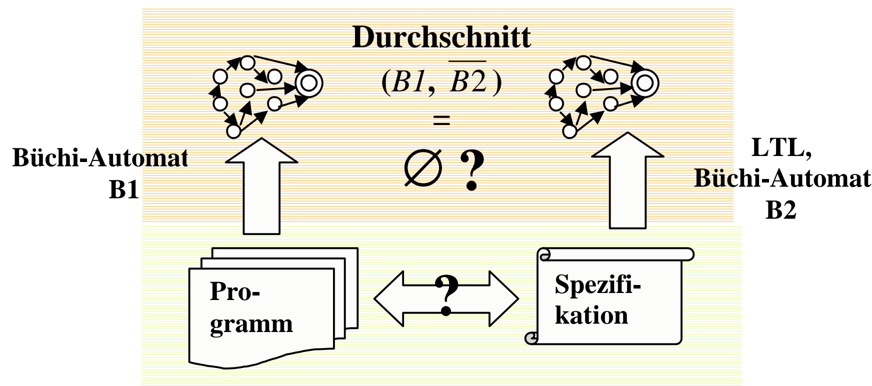

Abbildung 15: Mo delüberprüfung:DiemitLTLformulierteSpezifikationwirdineinen Automatenüberführt.DieserwirdmitdemausdemProgrammgewonn e-nenAutomatengeschni ttenundaufLeerheitüberprüft.

6 Ausblick

IndieserArbeitwurdeeineMöglichkeitderModellüberprüfu ngbetrachtet,beidereinschon fertiggestelltesProgrammuntersuchtwird,obdiesesauchdiegefordertenSp ezifikationen erfüllt.DieSpezifikationenwerdendabeiinLTLformuliert.VieleModellüberprüferbenu t-zeninderPraxisheutzutagejedochCTLoder eineCTL -Variante,weildabeinichtnureine SequenzvonZustandsfolgensonderneinBaumvonmöglichenZustandsfolgenbetrachtet wird.Damitlassensichinsb esonderenichtdeterministischeAbläufemodellieren. DadieAusgangssituationdervorgestelltenMod ellüberprüfungdasfertiggestellteProgramm istundvielArbeitschonindiesesinvestiertwurde,bedeuteteinefehlgeschlageneModel l-überprüfungdasBereinigenvonFehlern.Demgegenüberwäreesbeiweitembesser,nicht eineVerifikationdesbestehendenPr ogramms,sonderneineSyntheseeinesProgrammsaus denformuliertenSpezifikationenundEigenschaftendurchzuführen.EinsolcherAnsatzwird beispielsweiseauchim[Vardi]mithilfesog.Rabin -Automatenvorgestellt.

7 Literaturnachweis

[Vardi] VARDI,M OSHE Y.: *AnAutomata -TheoreticApproachtoLinearTemporal Logic*.In:ProceedingsoftheVIIIBanffHigherOrderWorkshop,Banff1994.

[HR] HUTH,M ICHAEL;R YAN,M ARK: *LogicinComputerScience:Modellingand reasoningaboutsystems* ,S.179 –280,CambridgeUniv ersityPress.

[MSS88] MULLER,D.E.;SAOUDI,A;SCHUPP,P.E.: *Weakalternatingautomatagivea simpleexplanationofwhymosttemporalanddynamiclogicsaredecidablein exponentialtime* .In:Proceedingsofthe3rd IEEESymposiumonLogicin ComputerScience,S.422-427,Edinburgh,July1988.

[Miyano] MIYANO,S;HAYASHI,T .: *Alternatingfiniteautomataonw-words* .Theoretical ComputerScience32,S.321-330,1984.

[MS87] MULLER,D.E;SCHUPP,P.E.: *Alternatingautomataoninfinitetrees* .Theoretical ComputerScience54,S.267-276,1987.

[Choueka] CHOUEKA. Y : *Theoriesofautomataonw-Tapes:Asimplifiedapproach.* JournalofComputerandSystemSciences8,S.117-141,1974.

[Büchi] BÜCHI,J.R.: *Onadecisionmethodinrestrictedsecondorderarithmetic*.In: ProceedingsoftheInternational.CongressofLogic,MethodandPhilos. Science1960,S.1-12,Stanford,1962,StanfordUniversityPress.