

# Interval Temporal Algebra

(Automated Theorem Prover for PITL)

Antonio Cau

Software Technology Research Laboratory  
De Montfort University

2008

(Html version of slides)

The logo for STRL, featuring the letters 'STRL' in a bold, blue, serif font with a slight 3D effect and a shadow.

This talk will introduce Interval Temporal Algebra

- 1 Introduction
- 2 Rationale
- 3 Propositional Interval Temporal Logic (PITL)
- 4 Algebraic semantics for PITL
- 5 Kleene and Omega Algebra
- 6 Interval Temporal Algebra
- 7 Automatic theorem prover for PITL
- 8 Conclusion and future work

An Automated Theorem Prover (ATP) for Interval Temporal Logic (ITL) has always been a desirable tool. There have been various attempts to implement such ATPs:

- Lite. Shinji Kono (1991 (Prolog), 2008 (Java)).  
Tableau-based, exponential in the number of the variables in the formula and also combinatorial w.r.t. to the nesting of temporal logic operators.
- ITL Library for PVS. Antonio Cau (1997).  
Interactive theorem prover for (in)finite first order ITL, requires expert proof knowledge and has little automation.
- DCVALID. Paritosh Pandya (2000).  
Quantified Discrete time Duration Calculus (QDDC).  
Translate QDDC into WS1S, use decision procedure of WS1S (MONA, Nils Klarlund and Anders Møller). WS1S has non-elementary complexity.

- PITL2Mona. Rodolfo Gomez (2004).  
Translate finite Propositional ITL (PITL) into WS1S, use decision procedure of WS1S (MONA). WS1S has non-elementary complexity.
- FL2CUDD. Ben Moszkowski (2005).  
Use a subset of (in)finite PITL called Fusion Logic (FL). The decision procedure of FL is built on top of Colorado University Decision Diagram (CUDD, Fabio Somenzi).

So mainly modelchecking or special purpose automated deduction.

There are off-the-shelf automated proof and counterexample search tools (ATP) for first-order and equational logic.

- PITL2Mona. Rodolfo Gomez (2004).  
Translate finite Propositional ITL (PITL) into WS1S, use decision procedure of WS1S (MONA). WS1S has non-elementary complexity.
- FL2CUDD. Ben Moszkowski (2005).  
Use a subset of (in)finite PITL called Fusion Logic (FL).  
The decision procedure of FL is built on top of Colorado University Decision Diagram (CUDD, Fabio Somenzi).

So mainly modelchecking or special purpose automated deduction.

There are off-the-shelf automated proof and counterexample search tools (ATP) for first-order and equational logic.

Inspired by the work of Peter Höffner, Georg Struth and Bernhard Möller we proceed as follows:

- Introduce Interval Temporal Algebra (ITA). An ITA is based on Kleene Algebra and Omega Algebra.
- Show that PITL is an ITA.
- Encode the axiom system for ITA in an off-the shelf ATP Prover9 (first-order and equational logic, William McCune).

Propositional Interval Temporal Logic (PITL) is a

- discrete,
- linear temporal logic
- for both finite and infinite time which includes
- a basic construct for sequential composition and
- an analog of Kleene star and Omega star

$$f ::= p \mid \neg f \mid f_1 \vee f_2 \mid \text{skip} \mid f_1 ; f_2 \mid f^*$$

where

- *skip* is an interval (sequence) of 2 states.
- $f_1 ; f_2$  is called ' $f_1$  chop  $f_2$ ' and denotes sequential composition of two intervals.
- $f^*$  is called ' $f$  chopstar' and denotes (in)finite iteration of an interval.



$\bigcirc f$	$\hat{=}$	$\text{skip} ; f$	next
$\bigcirc^w f$	$\hat{=}$	$\neg \bigcirc \neg f$	weak next
more	$\hat{=}$	$\bigcirc \text{true}$	interval with $\geq 2$ states
empty	$\hat{=}$	$\neg \text{more}$	one state interval
inf	$\hat{=}$	$\text{true} ; \text{false}$	infinite interval
finite	$\hat{=}$	$\neg \text{inf}$	finite interval
fmore	$\hat{=}$	$\text{more} \wedge \text{finite}$	finite with $\geq 2$ states
$\diamond f$	$\hat{=}$	$\text{finite} ; f$	sometimes
$\square f$	$\hat{=}$	$\neg \diamond \neg f$	always
$\diamond_i f$	$\hat{=}$	$f ; \text{true}$	some initial subinterval
$\square_i f$	$\hat{=}$	$\neg (\diamond_i \neg f)$	all initial subintervals
$\diamond_a f$	$\hat{=}$	$\text{finite} ; f ; \text{true}$	some subinterval
$\square_a f$	$\hat{=}$	$\neg (\diamond_a \neg f)$	all subintervals
etc.			

The main semantic notion is interval which is a sequence of states.

Let

- $\Sigma$  denotes the set of states.
- $\Sigma^+$  denote the set of non-empty finite sequences of states.
- $\Sigma^\omega$  denote the set of infinite sequences of states.
- $\sigma$  denote an interval,  $\sigma \in \Sigma^+ \cup \Sigma^\omega$ .
- Let  $\llbracket \dots \rrbracket$  be the “meaning” (semantic) function from  $\Sigma^+ \cup \Sigma^\omega$  to  $\{\text{tt}, \text{ff}\}$ .

- $\llbracket p \rrbracket_\sigma = \text{tt}$  iff  $\sigma_0(p) = \text{tt}$
- $\llbracket \neg f \rrbracket_\sigma = \text{tt}$  iff not  $(\llbracket f \rrbracket_\sigma = \text{tt})$
- $\llbracket f_1 \vee f_2 \rrbracket_\sigma = \text{tt}$  iff  $\llbracket f_1 \rrbracket_\sigma = \text{tt}$  or  $\llbracket f_2 \rrbracket_\sigma = \text{tt}$
- $\llbracket \text{skip} \rrbracket_\sigma = \text{tt}$  iff  $|\sigma| = 1$  where  $|\sigma|$  denotes length of  $\sigma$  and is defined as number of states minus 1
- $\llbracket f_1 ; f_2 \rrbracket_\sigma = \text{tt}$  iff  
 (exists  $k$ , s.t.  $\llbracket f_1 \rrbracket_{\sigma_0 \dots \sigma_k} = \text{tt}$  and  $\llbracket f_2 \rrbracket_{\sigma_k \dots \sigma_{|\sigma|}} = \text{tt}$ )  
 or ( $\sigma$  is infinite and  $\llbracket f_1 \rrbracket_\sigma = \text{tt}$ )

- $\llbracket f^* \rrbracket_\sigma = \text{tt}$  iff  
 if  $\sigma$  is finite then  
 (exist  $l_0, \dots, l_n$  s.t.  $l_0 = 0$  and  $l_n = |\sigma|$  and  
 for all  $0 \leq i < n, l_i \leq l_{i+1}$  and  $\llbracket f \rrbracket_{\sigma_{l_i} \dots \sigma_{l_{i+1}}} = \text{tt}$ )  
 else  
 (exist  $l_0, \dots, l_n$  s.t.  $l_0 = 0$  and  
 $\llbracket f \rrbracket_{\sigma_{l_n} \dots \sigma_{|\sigma|}} = \text{tt}$  and  
 for all  $0 \leq i < n, l_i \leq l_{i+1}$  and  $\llbracket f \rrbracket_{\sigma_{l_i} \dots \sigma_{l_{i+1}}} = \text{tt}$ )  
 or  
 (exist an infinite number of  $l_i$  s.t.  $l_0 = 0$  and  
 for all  $0 \leq i, l_i \leq l_{i+1}$  and  $\llbracket f \rrbracket_{\sigma_{l_i} \dots \sigma_{l_{i+1}}} = \text{tt}$ )

PropAx	All axioms for propositional logic
ChopAssoc	$\vdash (f_0 ; f_1) ; f_2 \equiv f_0 ; (f_1 ; f_2)$
OrChopImp	$\vdash (f_0 \vee f_1) ; f_2 \supset (f_0 ; f_2) \vee (f_1 ; f_2)$
ChopOrImp	$\vdash f_0 ; (f_1 \vee f_2) \supset (f_0 ; f_1) \vee (f_0 ; f_2)$
EmptyChop	$\vdash \text{empty} ; f_1 \equiv f_1$
ChopEmpty	$\vdash f_1 ; \text{empty} \equiv f_1$
BiBoxChop	$\vdash \Box(f_0 \supset f_1) \wedge \Box(f_2 \supset f_3) \supset (f_0 ; f_2) \supset (f_1 ; f_3)$
StateImpBi	$\vdash p \supset \Box p$
NextImpWNext	$\vdash \bigcirc f_0 \supset \neg \bigcirc \neg f_0$
SkipAnd	$\vdash (\text{skip} \wedge f_0) ; \text{true} \supset \neg((\text{skip} \wedge \neg f_0) ; \text{true})$
BoxInduct	$\vdash f_0 \wedge \Box(f_0 \supset \neg \bigcirc \neg f_0) \supset \Box f_0$
ChopStarEqv	$\vdash f_0^* \equiv (\text{empty} \vee ((f_0 \wedge \text{more}) ; f_0^*))$
ChopStarInduct	$\vdash (\text{inf} \wedge f_0 \wedge \Box(f_0 \supset (f_1 \wedge \text{fmore}) ; f_0)) \supset f_1^*$
MP	$\vdash f_0 \supset f_1 \text{ and } \vdash f_0 \text{ implies } \vdash f_1$
BoxGen	$\vdash f_0 \text{ implies } \vdash \Box f_0$
BiGen	$\vdash f_0 \text{ implies } \vdash \Box f_0$

# STRL Algebraic semantics for PITL (13)

Can we give the semantic domain an algebraic structure?

Let  $\llbracket f \rrbracket$  denote the set of intervals for which  $\llbracket f \rrbracket_\sigma = \text{tt}$ , i.e.,

$$\llbracket f \rrbracket \hat{=} \{\sigma \mid \llbracket f \rrbracket_\sigma = \text{tt}\}$$

The  $\vee$  of two PITL formula is then

$$\begin{aligned} \llbracket f_1 \vee f_2 \rrbracket &= \\ &\text{— definition of } \llbracket \cdot \rrbracket \\ &\{\sigma \mid \llbracket f_1 \vee f_2 \rrbracket_\sigma = \text{tt}\} \\ &\text{— definition of } \llbracket f_1 \vee f_2 \rrbracket_\sigma \\ &\{\sigma \mid \llbracket f_1 \rrbracket_\sigma = \text{tt} \text{ or } \llbracket f_2 \rrbracket_\sigma = \text{tt}\} \\ &\text{— set theory, let } \cup \text{ denote union} \\ &\{\sigma \mid \llbracket f_1 \rrbracket_\sigma = \text{tt}\} \cup \{\sigma \mid \llbracket f_2 \rrbracket_\sigma = \text{tt}\} \\ &\text{— definition of } \llbracket \cdot \rrbracket \\ &\llbracket f_1 \rrbracket \cup \llbracket f_2 \rrbracket \end{aligned}$$

Can we give the semantic domain an algebraic structure?

Let  $\llbracket f \rrbracket$  denote the set of intervals for which  $\llbracket f \rrbracket_\sigma = \text{tt}$ , i.e.,

$$\llbracket f \rrbracket \hat{=} \{\sigma \mid \llbracket f \rrbracket_\sigma = \text{tt}\}$$

The  $\vee$  of two PITL formula is then

$$\begin{aligned} \llbracket f_1 \vee f_2 \rrbracket &= \\ &\text{— definition of } \llbracket \cdot \rrbracket \\ &\{\sigma \mid \llbracket f_1 \vee f_2 \rrbracket_\sigma = \text{tt}\} \\ &\text{— definition of } \llbracket f_1 \vee f_2 \rrbracket_\sigma \\ &\{\sigma \mid \llbracket f_1 \rrbracket_\sigma = \text{tt} \text{ or } \llbracket f_2 \rrbracket_\sigma = \text{tt}\} \\ &\text{— set theory, let } \cup \text{ denote union} \\ &\{\sigma \mid \llbracket f_1 \rrbracket_\sigma = \text{tt}\} \cup \{\sigma \mid \llbracket f_2 \rrbracket_\sigma = \text{tt}\} \\ &\text{— definition of } \llbracket \cdot \rrbracket \\ &\llbracket f_1 \rrbracket \cup \llbracket f_2 \rrbracket \end{aligned}$$

So we need algebraic operators that correspond to  $\neg$ ,  $\vee$ , skip, ; and \*

- $\vee$  corresponds to union ( $\cup$ ) of sets of intervals
- $\neg$  corresponds to complement, i.e.,

$$\llbracket \neg f \rrbracket =$$

— definition of  $\llbracket \cdot \rrbracket$

$$\{\sigma \mid \llbracket \neg f \rrbracket_{\sigma} = \text{tt}\}$$

— definition of  $\llbracket \neg f \rrbracket_{\sigma}$

$$\{\sigma \mid \text{not } (\llbracket f \rrbracket_{\sigma} = \text{tt})\}$$

— set theory, let  $\overline{\phantom{x}}$  denote set complement

$$\overline{\{\sigma \mid \llbracket f \rrbracket_{\sigma} = \text{tt}\}}$$

— definition of  $\llbracket \cdot \rrbracket$

$$\llbracket f \rrbracket$$



What about chop (':')?

Let  $\cdot$  denote the fusion of two intervals  $\sigma_1, \sigma_2 \in \Sigma^* \cup \Sigma^\omega$ , i.e.,  
 Let  $\mathbf{a}, \mathbf{b} \in \Sigma$  ( $\mathbf{a}$  and  $\mathbf{b}$  are not the same),  $\mathbf{v}, \mathbf{w} \in \Sigma^*$  and  
 $\mathbf{s}, \mathbf{t} \in \Sigma^\omega$

$$\sigma_1 \cdot \sigma_2 \hat{=} \begin{cases} \mathbf{vaw} & \text{if } \sigma_1 = \mathbf{va}, \sigma_2 = \mathbf{aw} \\ \emptyset & \text{if } \sigma_1 = \mathbf{va}, \sigma_2 = \mathbf{bw} \\ \mathbf{vas} & \text{if } \sigma_1 = \mathbf{va}, \sigma_2 = \mathbf{as} \\ \emptyset & \text{if } \sigma_1 = \mathbf{va}, \sigma_2 = \mathbf{bs} \\ \mathbf{s} & \text{if } \sigma_1 = \mathbf{s}, \sigma_2 = \mathbf{aw} \\ \mathbf{s} & \text{if } \sigma_1 = \mathbf{s}, \sigma_2 = \mathbf{t} \end{cases}$$

Let  $\mathbf{S}, \mathbf{T} \subseteq \Sigma^* \cup \Sigma^\omega$  then

$$\mathbf{S} \cdot \mathbf{T} \hat{=} \{\sigma_1 \cdot \sigma_2 \mid \sigma_1 \in \mathbf{S} \text{ and } \sigma_2 \in \mathbf{T}\}$$

What about chop (':')?

Let  $\cdot$  denote the fusion of two intervals  $\sigma_1, \sigma_2 \in \Sigma^* \cup \Sigma^\omega$ , i.e.,  
 Let  $\mathbf{a}, \mathbf{b} \in \Sigma$  ( $\mathbf{a}$  and  $\mathbf{b}$  are not the same),  $\mathbf{v}, \mathbf{w} \in \Sigma^*$  and  
 $\mathbf{s}, \mathbf{t} \in \Sigma^\omega$

$$\sigma_1 \cdot \sigma_2 \hat{=} \begin{cases} \mathbf{vaw} & \text{if } \sigma_1 = \mathbf{va}, \quad \sigma_2 = \mathbf{aw} \\ \emptyset & \text{if } \sigma_1 = \mathbf{va}, \quad \sigma_2 = \mathbf{bw} \\ \mathbf{vas} & \text{if } \sigma_1 = \mathbf{va}, \quad \sigma_2 = \mathbf{as} \\ \emptyset & \text{if } \sigma_1 = \mathbf{va}, \quad \sigma_2 = \mathbf{bs} \\ \mathbf{s} & \text{if } \sigma_1 = \mathbf{s}, \quad \sigma_2 = \mathbf{aw} \\ \mathbf{s} & \text{if } \sigma_1 = \mathbf{s}, \quad \sigma_2 = \mathbf{t} \end{cases}$$

Let  $\mathbf{S}, \mathbf{T} \subseteq \Sigma^* \cup \Sigma^\omega$  then

$$\mathbf{S} \cdot \mathbf{T} \hat{=} \{\sigma_1 \cdot \sigma_2 \mid \sigma_1 \in \mathbf{S} \text{ and } \sigma_2 \in \mathbf{T}\}$$

- ‘;’ corresponds to fusion ‘ $\cdot$ ’, i.e.,

$$\llbracket f_1 ; f_2 \rrbracket =$$

— definition of  $\llbracket \cdot \rrbracket$

$$\{\sigma \mid \llbracket f_1 ; f_2 \rrbracket_\sigma = \text{tt}\}$$

— definition of  $\llbracket f_1 ; f_2 \rrbracket_\sigma$

$$\{\sigma \mid (\text{exists } k, \text{ s.t. } \llbracket f_1 \rrbracket_{\sigma_0 \dots \sigma_k} = \text{tt and } \llbracket f_2 \rrbracket_{\sigma_k \dots \sigma_{|\sigma|}} = \text{tt}) \\ \text{or } (\sigma \text{ is infinite and } \llbracket f_1 \rrbracket_\sigma = \text{tt})\}$$

— definition of  $\cdot$

$$\{\sigma \mid \llbracket f_1 \rrbracket_\sigma = \text{tt}\} \cdot \{\sigma \mid \llbracket f_2 \rrbracket_\sigma = \text{tt}\}$$

— definition of  $\llbracket \cdot \rrbracket$

$$\llbracket f_1 \rrbracket \cdot \llbracket f_2 \rrbracket$$

What about **empty**?

$\llbracket \text{empty} \rrbracket =$   
— definition of  $\llbracket \cdot \rrbracket$   
 $\{\sigma \mid \llbracket \text{empty} \rrbracket_\sigma = \text{tt}\}$   
— definition of  $\llbracket \text{empty} \rrbracket_\sigma$   
 $\{\sigma \mid \sigma \text{ is a } 1 \text{ state interval}\}$   
— definition of  $\Sigma$   
 $\Sigma$

What about **empty**?

$\llbracket \text{empty} \rrbracket =$   
— definition of  $\llbracket \ ]$   
 $\{\sigma \mid \llbracket \text{empty} \rrbracket_\sigma = \text{tt}\}$   
— definition of  $\llbracket \text{empty} \rrbracket_\sigma$   
 $\{\sigma \mid \sigma \text{ is a } \mathbf{1} \text{ state interval}\}$   
— definition of  $\Sigma$   
 $\Sigma$

What about **skip**?

**skip** can be defined as  $\overline{\Sigma \cup \overline{\Sigma} \cdot \overline{\Sigma}}$

$$\overline{\Sigma \cup \overline{\Sigma} \cdot \overline{\Sigma}} =$$

— De Morgan for set theory

$$\overline{\Sigma} \cap \overline{\overline{\Sigma} \cdot \overline{\Sigma}}$$

$\overline{\Sigma}$  is the set of intervals containing  $\geq 2$  states

$\overline{\Sigma} \cdot \overline{\Sigma}$  is the set of intervals containing  $\geq 3$  states

$\overline{\Sigma \cdot \Sigma}$  is the set of intervals containing  $\leq 2$  states

$\overline{\Sigma} \cap \overline{\overline{\Sigma} \cdot \overline{\Sigma}}$  is the set of intervals containing exactly 2 states

What about **skip**?

**skip** can be defined as  $\overline{\Sigma \cup \overline{\Sigma} \cdot \overline{\Sigma}}$

$$\overline{\Sigma \cup \overline{\Sigma} \cdot \overline{\Sigma}} =$$

— De Morgan for set theory

$$\overline{\Sigma} \cap \overline{\overline{\Sigma} \cdot \overline{\Sigma}}$$

$\overline{\Sigma}$  is the set of intervals containing  $\geq 2$  states

$\overline{\Sigma} \cdot \overline{\Sigma}$  is the set of intervals containing  $\geq 3$  states

$\overline{\Sigma \cdot \overline{\Sigma}}$  is the set of intervals containing  $\leq 2$  states

$\overline{\Sigma} \cap \overline{\overline{\Sigma} \cdot \overline{\Sigma}}$  is the set of intervals containing exactly 2 states

What about a state formula, i.e., a formula without temporal operators?

A state formula only constrains the first state of an interval.  
Let  $p$  be a state formula. Then the following holds

$$\llbracket p \rrbracket = (\llbracket p \rrbracket \cap \Sigma) \cdot T$$

where

$$\begin{aligned} T &\hat{=} \llbracket \text{true} \rrbracket = \Sigma^* \cup \Sigma^\omega \\ \emptyset &\hat{=} \llbracket \text{false} \rrbracket = \emptyset \end{aligned}$$



What about a state formula, i.e., a formula without temporal operators?

A state formula only constrains the first state of an interval.  
Let  $p$  be a state formula. Then the following holds

$$\llbracket p \rrbracket = (\llbracket p \rrbracket \cap \Sigma) \cdot T$$

where

$$\begin{aligned} T &\hat{=} \llbracket \text{true} \rrbracket = \Sigma^* \cup \Sigma^\omega \\ \emptyset &\hat{=} \llbracket \text{false} \rrbracket = \emptyset \end{aligned}$$

What about chopstar ‘\*’?

In the semantics of ‘\*’ both finite and infinite iteration are considered simultaneously. Let’s define separate algebraic operators for them.

Let  $S^*$  and  $S^\omega$  denote respectively finite and infinite iteration of a set  $S \subseteq \Sigma^+ \cup \Sigma^\omega$  and can be defined as follows

$$\begin{aligned} S_0 &\hat{=} \Sigma \\ S_i &\hat{=} \{\sigma_1 \cdot \sigma_2 \mid \sigma_1 \in S_{i-1} \text{ and } \sigma_2 \in S\} \\ S^* &\hat{=} \bigcup_{i \in \mathbb{N}} S_i \\ S^\omega &\hat{=} \bigcup_i S_i \end{aligned}$$

Then we have

$$\llbracket f^* \rrbracket = \dots = \llbracket f \rrbracket^* \cup \llbracket f \rrbracket^\omega$$

What about chopstar ‘\*’?

In the semantics of ‘\*’ both finite and infinite iteration are considered simultaneously. Let’s define separate algebraic operators for them.

Let  $\mathbf{S}^*$  and  $\mathbf{S}^\omega$  denote respectively finite and infinite iteration of a set  $\mathbf{S} \subseteq \Sigma^+ \cup \Sigma^\omega$  and can be defined as follows

$$\begin{aligned}\mathbf{S}_0 &\hat{=} \Sigma \\ \mathbf{S}_i &\hat{=} \{\sigma_1 \cdot \sigma_2 \mid \sigma_1 \in \mathbf{S}_{i-1} \text{ and } \sigma_2 \in \mathbf{S}\} \\ \mathbf{S}^* &\hat{=} \bigcup_{i \in \mathbb{N}} \mathbf{S}_i \\ \mathbf{S}^\omega &\hat{=} \bigcup_i \mathbf{S}_i\end{aligned}$$

Then we have

$$\llbracket f^* \rrbracket = \dots = \llbracket f \rrbracket^* \cup \llbracket f \rrbracket^\omega$$

Let  $K$  denote  $\mathcal{P}(\Sigma^+ \cup \Sigma^\omega)$ ,

$(K, \cup, \emptyset, \cdot, \Sigma)$  is an idempotent left semiring iff

(for  $a, b, c \in K$ )

- $(K, \cup, \emptyset)$  is a commutative monoid, i.e.,  
 $a \cup b = b \cup a$   
 $a \cup \emptyset = a$   
 $a \cup (b \cup c) = (a \cup b) \cup c$
- $(K, \cdot, \Sigma)$  is a monoid, i.e.,  
 $a \cdot \Sigma = a$   
 $\Sigma \cdot a = a$   
 $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
- $(a \cup b) \cdot c = a \cdot c \cup b \cdot c$
- $\emptyset \cdot a = \emptyset$
- $a \cup a = a$
- $b \subseteq c$  implies  $a \cdot b \subseteq a \cdot c$  where  $a \subseteq b$  iff  $a \cup b = b$

Let  $K$  denote  $\mathcal{P}(\Sigma^+ \cup \Sigma^\omega)$ ,

$(K, \cup, \emptyset, \cdot, \Sigma)$  is an idempotent left semiring iff  
(for  $a, b, c \in K$ )

- $(K, \cup, \emptyset)$  is a commutative monoid, i.e.,  
 $a \cup b = b \cup a$   
 $a \cup \emptyset = a$   
 $a \cup (b \cup c) = (a \cup b) \cup c$
- $(K, \cdot, \Sigma)$  is a monoid, i.e.,  
 $a \cdot \Sigma = a$   
 $\Sigma \cdot a = a$   
 $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
- $(a \cup b) \cdot c = a \cdot c \cup b \cdot c$
- $\emptyset \cdot a = \emptyset$
- $a \cup a = a$
- $b \subseteq c$  implies  $a \cdot b \subseteq a \cdot c$  where  $a \subseteq b$  iff  $a \cup b = b$

Let  $K$  denote  $\mathcal{P}(\Sigma^+ \cup \Sigma^\omega)$ ,

$(K, \cup, \emptyset, \cdot, \Sigma)$  is an idempotent left semiring iff  
(for  $a, b, c \in K$ )

- $(K, \cup, \emptyset)$  is a commutative monoid, i.e.,  
 $a \cup b = b \cup a$   
 $a \cup \emptyset = a$   
 $a \cup (b \cup c) = (a \cup b) \cup c$
- $(K, \cdot, \Sigma)$  is a monoid, i.e.,  
 $a \cdot \Sigma = a$   
 $\Sigma \cdot a = a$   
 $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
- $(a \cup b) \cdot c = a \cdot c \cup b \cdot c$
- $\emptyset \cdot a = \emptyset$
- $a \cup a = a$
- $b \subseteq c$  implies  $a \cdot b \subseteq a \cdot c$  where  $a \subseteq b$  iff  $a \cup b = b$

Let  $K$  denote  $\mathcal{P}(\Sigma^+ \cup \Sigma^\omega)$ ,

$(K, \cup, \emptyset, \cdot, \Sigma)$  is an idempotent left semiring iff  
(for  $a, b, c \in K$ )

- $(K, \cup, \emptyset)$  is a commutative monoid, i.e.,  
 $a \cup b = b \cup a$   
 $a \cup \emptyset = a$   
 $a \cup (b \cup c) = (a \cup b) \cup c$
- $(K, \cdot, \Sigma)$  is a monoid, i.e.,  
 $a \cdot \Sigma = a$   
 $\Sigma \cdot a = a$   
 $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
- $(a \cup b) \cdot c = a \cdot c \cup b \cdot c$
- $\emptyset \cdot a = \emptyset$
- $a \cup a = a$
- $b \subseteq c$  implies  $a \cdot b \subseteq a \cdot c$  where  $a \subseteq b$  iff  $a \cup b = b$

$(K, \overline{\phantom{x}})$  is a Boolean idempotent left semiring iff

- $(K, \cup, \emptyset, \cdot, \Sigma)$  is an idempotent left semiring
- $a = \overline{(\overline{a} \cup \overline{b})} \cup (\overline{a} \cup \overline{b})$  (Huntington equation)

As usual we have the following

- $a \cap b = \overline{\overline{a} \cup \overline{b}}$
- $\top = a \cup \overline{a}$  (greatest element w.r.t.  $\subseteq$ , i.e.  $\Sigma^+ \cup \Sigma^\omega$ )
- $\emptyset = a \cap \overline{a}$  (smallest element w.r.t.  $\subseteq$ , i.e.,  $\emptyset$ )



$(K, \overline{\phantom{x}})$  is a Boolean idempotent left semiring iff

- $(K, \cup, \emptyset, \cdot, \Sigma)$  is an idempotent left semiring
- $\mathbf{a} = \overline{(\overline{\mathbf{a}} \cup \overline{\mathbf{b}})} \cup \overline{(\overline{\mathbf{a}} \cup \mathbf{b})}$  (Huntington equation)

As usual we have the following

- $\mathbf{a} \cap \mathbf{b} = \overline{\overline{\mathbf{a}} \cup \overline{\mathbf{b}}}$
- $\top = \mathbf{a} \cup \overline{\mathbf{a}}$  (greatest element w.r.t.  $\subseteq$ , i.e.  $\Sigma^+ \cup \Sigma^\omega$ )
- $\emptyset = \mathbf{a} \cap \overline{\mathbf{a}}$  (smallest element w.r.t.  $\subseteq$ , i.e.,  $\emptyset$ )

$(K, \overline{\phantom{x}})$  is a Boolean idempotent left semiring iff

- $(K, \cup, \emptyset, \cdot, \Sigma)$  is an idempotent left semiring
- $a = \overline{\overline{a} \cup \overline{b}} \cup \overline{\overline{a} \cup b}$  (Huntington equation)

As usual we have the following

- $a \cap b = \overline{\overline{a} \cup \overline{b}}$
- $\top = a \cup \overline{a}$  (greatest element w.r.t.  $\subseteq$ , i.e.  $\Sigma^+ \cup \Sigma^\omega$ )
- $\emptyset = a \cap \overline{a}$  (smallest element w.r.t.  $\subseteq$ , i.e.,  $\emptyset$ )

$(K, *)$  is a Boolean strong left (lazy) Kleene algebra iff

- $(K, \neg)$  is a Boolean idempotent left semiring
- $\Sigma \cup a \cdot a^* \subseteq a^*$
- $b \cup a \cdot c \subseteq c$  implies  $a^* \cdot b \subseteq c$
- $b \cup c \cdot a \subseteq c$  implies  $b \cdot a^* \subseteq c$

$(K, {}^\omega)$  is a Boolean left (lazy) omega algebra iff

- $(K, *)$  is a Boolean strong left (lazy) Kleene algebra
- $a^\omega = a \cdot a^\omega$
- $c \subseteq b \cup a \cdot c$  implies  $c \subseteq a^\omega \cup a^* \cdot b$

$(K, *)$  is a Boolean strong left (lazy) Kleene algebra iff

- $(K, \neg)$  is a Boolean idempotent left semiring
- $\Sigma \cup a \cdot a^* \subseteq a^*$
- $b \cup a \cdot c \subseteq c$  implies  $a^* \cdot b \subseteq c$
- $b \cup c \cdot a \subseteq c$  implies  $b \cdot a^* \subseteq c$

$(K, {}^\omega)$  is a Boolean left (lazy) omega algebra iff

- $(K, *)$  is a Boolean strong left (lazy) Kleene algebra
- $a^\omega = a \cdot a^\omega$
- $c \subseteq b \cup a \cdot c$  implies  $c \subseteq a^\omega \cup a^* \cdot b$

$(K, *)$  is a Boolean strong left (lazy) Kleene algebra iff

- $(K, \neg)$  is a Boolean idempotent left semiring
- $\Sigma \cup a \cdot a^* \subseteq a^*$
- $b \cup a \cdot c \subseteq c$  implies  $a^* \cdot b \subseteq c$
- $b \cup c \cdot a \subseteq c$  implies  $b \cdot a^* \subseteq c$

$(K, {}^\omega)$  is a Boolean left (lazy) omega algebra iff

- $(K, *)$  is a Boolean strong left (lazy) Kleene algebra
- $a^\omega = a \cdot a^\omega$
- $c \subseteq b \cup a \cdot c$  implies  $c \subseteq a^\omega \cup a^* \cdot b$

$(K, \omega)$  is an interval temporal algebra iff

Let  $\text{skip} \triangleq \overline{\Sigma \cup \overline{\Sigma} \cdot \overline{\Sigma}}$ ,  $\text{finite} \triangleq \overline{T \cdot \emptyset}$  and  $\Box a \triangleq \overline{\text{finite} \cdot a}$

P1  $(K, \omega)$  is a Boolean left (lazy) omega algebra

P2  $a \cdot (b \cup c) \subseteq a \cdot b \cup a \cdot c$

P3  $(\text{skip} \cap a) \cdot c \cap (\text{skip} \cap b) \cdot d = (\text{skip} \cap a \cap b) \cdot (c \cap d)$

P4  $c \cdot (\text{skip} \cap a) \cap d \cdot (\text{skip} \cap b) = (c \cap d) \cdot (\text{skip} \cap a \cap b)$

P5  $(\Sigma \cap a) \cdot c \cap (\Sigma \cap b) \cdot d = (\Sigma \cap a \cap b) \cdot (c \cap d)$

P6  $c \cdot (\Sigma \cap a) \cap d \cdot (\Sigma \cap b) = (c \cap d) \cdot (\Sigma \cap a \cap b)$

P7  $\text{finite} \subseteq \text{skip}^*$

P8  $\overline{\text{skip} \cdot a} = \Sigma \cup \text{skip} \cdot a$

P9  $\overline{a \cdot \text{skip}} = \Sigma \cup a \cdot \text{skip}$

PropAx	All axioms for propositional logic
ChopAssoc	$\vdash (f_0 ; f_1) ; f_2 \equiv f_0 ; (f_1 ; f_2)$
OrChopImp	$\vdash (f_0 \vee f_1) ; f_2 \supset (f_0 ; f_2) \vee (f_1 ; f_2)$
ChopOrImp	$\vdash f_0 ; (f_1 \vee f_2) \supset (f_0 ; f_1) \vee (f_0 ; f_2)$
EmptyChop	$\vdash \text{empty} ; f_1 \equiv f_1$
ChopEmpty	$\vdash f_1 ; \text{empty} \equiv f_1$
BiBoxChop	$\vdash \Box(f_0 \supset f_1) \wedge \Box(f_2 \supset f_3) \supset (f_0 ; f_2) \supset (f_1 ; f_3)$
StateImpBi	$\vdash p \supset \Box p$
NextImpWNext	$\vdash \bigcirc f_0 \supset \neg \bigcirc \neg f_0$
SkipAnd	$\vdash (\text{skip} \wedge f_0) ; \text{true} \supset \neg((\text{skip} \wedge \neg f_0) ; \text{true})$
BoxInduct	$\vdash f_0 \wedge \Box(f_0 \supset \neg \bigcirc \neg f_0) \supset \Box f_0$
ChopStarEqv	$\vdash f_0^* \equiv (\text{empty} \vee ((f_0 \wedge \text{more}) ; f_0^*))$
ChopStarInduct	$\vdash (\text{inf} \wedge f_0 \wedge \Box(f_0 \supset (f_1 \wedge \text{fmore}) ; f_0)) \supset f_1^*$
MP	$\vdash f_0 \supset f_1 \text{ and } \vdash f_0 \text{ implies } \vdash f_1$
BoxGen	$\vdash f_0 \text{ implies } \vdash \Box f_0$
BiGen	$\vdash f_0 \text{ implies } \vdash \Box f_0$

## Example

$$\begin{aligned} & \text{finite} \cdot \Sigma \\ & \text{--- } a \cdot \Sigma = a \\ & = \text{finite} \end{aligned}$$

## Example

$$\begin{aligned} & (T \cdot \emptyset) \cdot \emptyset \\ & \text{--- } a \cdot (b \cdot c) = (a \cdot b) \cdot c \\ = & T \cdot (\emptyset \cdot \emptyset) \\ & \text{--- } \emptyset \cdot a = \emptyset \\ = & T \cdot \emptyset \end{aligned}$$



Theorem

*PITL is an interval temporal algebra*

Theorem

*PITL's axiom/proof system can be derived from ITA's axiom system*

## *STRL* Automatic theorem prover for PITL (28)

- Take an off-the-shelf automatic theorem prover (ATP)
- encode the algebraic rules for an interval temporal algebra within it

We used Prover9 as ATP and encoded the interval temporal algebraic rules.

- Proved more than 350 PITL theorems so far
- Derived all PITL axioms/proof rules from ITA's axioms

Demo

- An interval temporal algebraic encoding of PITL in Prover9 results in quite a useful tool
- need to reduce the axioms of ITA
- replace some of the more complex axioms of ITA by simpler ones,
- Use a different ATP
- Integration of Description Logic axiom system with Interval Temporal Algebra axiom system