15-812: Programming Language Semantics

Lecture Notes on Uniform Substitution Understanding

André Platzer

Carnegie Mellon University

Lecture 8

1 Introduction

In the previous lectures we saw how useful formal proofs in dynamic logic can be to justify the correctness of programs. We also saw how subtle programs can be and that formal proofs help us identify crucial requirements that are needed for the programs to work correctly. Finally we learned how important it is for our reasoning principles to be sound, or else the analysis of the programs we are interested in could be flawed. We eventually traced the root cause to a confusion of syntax and semantics in our (incorrect!) soundness "proof" of the flawed induction axiom that was proposed in class:

$$[\alpha^*]\phi \leftrightarrow \phi \land (\phi \rightarrow [\alpha]\phi)$$

While it might have been momentarily lamentable that we wasted time with what turned out not to be correct in the end, this could, in fact, have a lasting impact on how carefully we guide the analysis of programs in the long run. Indeed, it turned out to be absolutely crucial to spot the problem in the "proof" that we carefully distinguish between formulas that are just formulas and formulas that are actually true, and where exactly in which states they are true. This care to the program semantics led us to understand that the correct axiom is:

$$\text{I}\ \ [\alpha^*]P \leftrightarrow P \land [\alpha^*](P \rightarrow [\alpha]P)$$

Armed with this newfound motivation to really carefully safeguard our reasoning about programs by justifying every step along the way from the very semantics of programs, we now look back at our adopted axioms in Fig. 1.

The axioms and proof rules of dynamic logic look mostly harmless. Of course, they still deserve careful soundness proofs! Before we get started, this is an excellent oppor-

$$
\begin{array}{rl}
[:=] & [x := \theta]p(x) \leftrightarrow p(\theta) \\[1em]
[?] & [?Q]P \leftrightarrow (Q \rightarrow P) \\[1em]
[;] & [\alpha;\beta]P \leftrightarrow [\alpha][\beta]P \\[1em]
[\mathrm{if}] & [\mathsf{if}(Q)\,\alpha\,\mathsf{else}\,\beta]P \leftrightarrow (Q \rightarrow [\alpha]P) \land (\neg Q \rightarrow [\beta]P) \\[1em]
[\mathrm{unwind}] & [\mathsf{while}(Q)\,\alpha]P \leftrightarrow [\mathsf{if}(Q)\,\{\alpha;\mathsf{while}(Q)\,\alpha\}]P \\[1em]
\mathrm{I} & [\alpha^*]P \leftrightarrow P \land [\alpha^*](P \rightarrow [\alpha]P) \\[1em]
\mathrm{G} & \dfrac{\vdash P}{\Gamma \vdash [\alpha]P, \Delta} \\[1em]
\mathrm{M}[\cdot] & \dfrac{P \vdash Q}{\Gamma, [\alpha]P \vdash [\alpha]Q, \Delta}
\end{array}
$$

Figure 1: Axioms and proof rules of dynamic logic

tunity for you to go ahead and prove the soundness of all the axioms and proof rules that are still in need of a proper soundness argument.

Today's lecture notes are based on the literature [Chu56, Pla15, Pla17, BRV$^+$17, Pla18].

2 What the Axioms Mean

The first subtlety we find among the axioms of Fig. 1 is right away in the assignment axiom [:=]. What exactly do $p(x)$ and $p(\theta)$ mean?

Clearly what this axiom is somehow trying to express is that after the assignment $x := \theta$ the formula $p(x)$ is true iff before the assignment the formula $p(\theta)$ is true, which is like $p(x)$ but has the term θ in place of the variable x. But where exactly is that and how does such an "in place of" replacement really happen? What could possibly go wrong?

3 Contextual Meaning

One way of understanding $p(x)$ and $p(\theta)$ in axiom [:=] is that they come from formula contexts, so formulas $p(\cdot)$ that contain a term-shaped hole, written $\cdot$, that is to be filled with x when forming $p(x)$ and to be filled with term θ when forming $p(\theta)$. Then the use of the axiom [:=] from left to right will fill the whole that used to be filled with x with the term θ, instead.

This is a possible way of understanding axiom [:=], but it will, of course, require a precise definition of what a formula context is, what its precise semantics is, how it relates to the semantics of ordinary formulas, and then all these meanings need to be respected during the soundness proof arguing why axiom [:=] is sound. In fact, this understanding with contexts also seems to impress itself upon us for making sense of what it means for us to use axioms in the context of a formula when applying axioms such as [:=] in the middle of a formula, which we found so useful in practice. Of course, an implementation of axiom [:=] in a verification tool will also have to implement the management of these contexts by an algorithm and will have to do so soundly or else we won't be able to trust its verification outcome.

We will eventually inspect this route for comparison, but first take a significantly easier approach.

4 Literal Meaning

The alternative way of understanding axiom [:=] is to take it literally and read $p(x)$ as an arity 1 predicate symbol p applied to the variable x while taking $p(\theta)$ as the same unary predicate symbol p applied to the term θ. That sounds pretty harmless, and, indeed, if we understand the axiom so literally, then it is quite obviously sound, because it simply represents a valid formula of dynamic logic, so is true in all states of all interpretations. We will get the same truth value whether we apply the predicate symbol p to the variable x after having assigned the value of term θ to x or whether we directly apply p to the term θ. The soundness of axiom [:=] then merely is an argument about the substitution of equals. And we did not even need to be bothered with understanding any contexts or holes or meta constructs at all to make sense of axiom [:=] literally!

Now the only problem is that the number of formulas that axiom [:=] will be able to prove for us with its literal verbatim reading is rather limited. What if we aren't even interested in the truth-value of predicate symbol p when we try to verify the correctness of our favorite squaring program?

The clou is that all that's missing is a way of taking a DL formula such as the one coming from the assignment axiom [:=] and substituting, e.g., another formula in for predicate symbol p. Of course, we should make darn sure to put in the same formula for the predicate symbol p uniformly in all places, which explains why the corresponding mechanism is called *uniform substitution*. Uniform substitutions have been originally invented for first-order logic [Chu56] but, after suitable generalizations, work equally well for dynamic logics [Pla15, Pla17].

5 Uniform Substitution

$$\text{US}\ \frac{\phi}{\sigma(\phi)}$$

The result of applying the uniform substitution σ to formula ϕ is denoted $\sigma(\phi)$ and defined in Fig. 2. Likewise, the result of applying the uniform substitution σ to term θ is denoted $\sigma(\theta)$ and the result of applying it to program α is denoted $\sigma(\alpha)$.

$$
\begin{array}{rl}
\sigma(x) = x & \text{for variable } x \in \mathcal{V}\\
\sigma(f(\theta)) = (\sigma(f))(\sigma(\theta)) \stackrel{\text{def}}{=} \{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot)) & \text{for function symbol } f \in \sigma\\
\sigma(g(\theta)) = g(\sigma(\theta)) & \text{for function symbol } g \notin \sigma\\
\sigma(\theta + \eta) = \sigma(\theta) + \sigma(\eta) & \\
\sigma(\theta \cdot \eta) = \sigma(\theta) \cdot \sigma(\eta) & \\
\hline
\sigma(\theta \geq \eta) \equiv \sigma(\theta) \geq \sigma(\eta) & \\
\sigma(p(\theta)) \equiv (\sigma(p))(\sigma(\theta)) \stackrel{\text{def}}{\equiv} \{\cdot \mapsto \sigma(\theta)\}(\sigma p(\cdot)) & \text{for predicate symbol } p \in \sigma\\
\sigma(q(\theta)) \equiv q(\sigma(\theta)) & \text{for predicate symbol } q \notin \sigma\\
\sigma(C(\phi)) \equiv \sigma(C)(\sigma(\phi)) \stackrel{\text{def}}{\equiv} \{\_ \mapsto \sigma(\phi)\}(\sigma C(\_)) & \text{if } \sigma \text{ is } \mathcal{V}\text{-admissible for } \phi, C \in \sigma\\
\sigma(C(\phi)) \equiv C(\sigma(\phi)) & \text{if } \sigma \text{ is } \mathcal{V}\text{-admissible for } \phi, C \notin \sigma\\
\sigma(\neg\phi) \equiv \neg\sigma(\phi) & \\
\sigma(\phi \wedge \psi) \equiv \sigma(\phi) \wedge \sigma(\psi) & \\
\sigma(\forall x\, \phi) \equiv \forall x\, \sigma(\phi) & \text{if } \sigma \text{ is } \{x\}\text{-admissible for } \phi\\
\sigma(\exists x\, \phi) \equiv \exists x\, \sigma(\phi) & \text{if } \sigma \text{ is } \{x\}\text{-admissible for } \phi\\
\sigma([\alpha]\phi) \equiv [\sigma(\alpha)]\sigma(\phi) & \text{if } \sigma \text{ is } \mathrm{BV}(\sigma(\alpha))\text{-admissible for } \phi\\
\sigma(\langle\alpha\rangle\phi) \equiv \langle\sigma(\alpha)\rangle\sigma(\phi) & \text{if } \sigma \text{ is } \mathrm{BV}(\sigma(\alpha))\text{-admissible for } \phi\\
\hline
\sigma(a) \equiv \sigma a & \text{for program constant } a \in \sigma\\
\sigma(b) \equiv b & \text{for program constant } b \notin \sigma\\
\sigma(x := \theta) \equiv x := \sigma(\theta) & \\
\sigma(?Q) \equiv ?\sigma(Q) & \\
\sigma(\mathsf{if}(Q)\,\alpha\,\mathsf{else}\,\beta) \equiv \mathsf{if}(\sigma(Q))\,\sigma(\alpha)\,\mathsf{else}\,\sigma(\beta) & \\
\sigma(\alpha;\beta) \equiv \sigma(\alpha);\sigma(\beta) & \text{if } \sigma \text{ is } \mathrm{BV}(\sigma(\alpha))\text{-admissible for } \beta\\
\sigma(\alpha^*) \equiv (\sigma(\alpha))^* & \text{if } \sigma \text{ is } \mathrm{BV}(\sigma(\alpha))\text{-admissible for } \alpha
\end{array}
$$

Figure 2: Recursive application of uniform substitution σ

Definition 1 (Admissible uniform substitution)**.** A uniform substitution σ is *U-admissible for ϕ* (or θ or α, respectively) with respect to the variables $U \subseteq \mathcal{V}$ iff $\mathrm{FV}(\sigma|_{\Sigma(\phi)}) \cap U = \emptyset$, where $\sigma|_{\Sigma(\phi)}$ is the restriction of σ that only replaces symbols that occur in ϕ, and $\mathrm{FV}(\sigma) = \bigcup_{f\in\sigma} \mathrm{FV}(\sigma f(\cdot)) \cup \bigcup_{p\in\sigma} \mathrm{FV}(\sigma p(\cdot))$ are the *free variables* that σ introduces. A uniform substitution σ is *admissible for ϕ* (or θ or α, respectively) iff the bound variables U of each operator of ϕ are not free in the substitution on its arguments, i.e. σ is U-admissible. These admissibility conditions are listed explicitly in Fig. 2, which defines the result $\sigma(\phi)$ of applying σ to ϕ.

6 Soundness of Uniform Substitution

Lemma 2 (Uniform substitution for formulas)**.** *The uniform substitution σ and its adjoint interpretation $\sigma^*_\omega I, \omega$ for I, ω have the same semantics for all* formulas *ϕ:*

$$\omega \in I[\![\sigma(\phi)]\!] \textit{ iff } \omega \in \sigma^*_\omega I[\![\phi]\!]$$

Theorem 3 (Uniform substitution). *The proof rule US is sound:*

$$\text{US} \ \frac{\phi}{\sigma(\phi)}$$

Proof. The proof [Pla17] uses that truth of the substituted formula is equivalent to truth of the original formula in the adjoint interpretation to conclude that validity of the premise in all interpretations implies validity in the adjoint interpretation so validity of the conclusion. Let the premise ϕ of rule US be valid, i.e., $\omega \in I[\![\phi]\!]$ for all states ω and for all interpretations I of the program, predicate, and function symbols. To show that the conclusion is valid, consider any state ω and any interpretation I and show that $\omega \in I[\![\sigma(\phi)]\!]$. By Lemma 2, the uniformly substituted formula $\sigma(\phi)$ is true in state ω of interpretation I iff the original formula ϕ is true in state ω of the adjoint interpretation $\sigma^*_\omega I$ that has already been modified according to the substitution σ, that is $\omega \in I[\![\sigma(\phi)]\!]$ iff $\omega \in \sigma^*_\omega I[\![\phi]\!]$. Now $\omega \in \sigma^*_\omega I[\![\phi]\!]$ holds, because $\omega \in I[\![\phi]\!]$ for all states ω and interpretations I, including for state ω and interpretation $\sigma^*_\omega I$, by premise. □

The proof of the uniform substitution lemma Lemma 2 that is used crucially in the proof of Theorem 3 crucially relies on correctness properties of the free and bound variable definitions, which we will have to investigate first.

7 Uniform Substitution Lemmas

A crucial element for the proof of Lemma 2 is the following fact about adjoint interpretations.

Corollary 4 (Admissible adjoints). *If $\omega = \nu$ on $FV(\sigma)$, then $\sigma^*_\omega I = \sigma^*_\nu I$. If σ is U-admissible for θ (or ϕ or α, respectively) and $\omega = \nu$ on $U^\complement$, then*

$$\sigma^*_\omega I[\![\theta]\!] = \sigma^*_\nu I[\![\theta]\!] \text{ i.e. } \sigma^*_\omega I\mu[\![\theta]\!] = \sigma^*_\nu I\mu[\![\theta]\!] \text{ for all states } \mu$$
$$\sigma^*_\omega I[\![\phi]\!] = \sigma^*_\nu I[\![\phi]\!]$$
$$\sigma^*_\omega I[\![\alpha]\!] = \sigma^*_\nu I[\![\alpha]\!]$$

Lemma 5 (Uniform substitution for terms). *The uniform substitution σ and its adjoint interpretation $\sigma^*_\omega I, \omega$ for I, ω have the same semantics for all* terms *θ:*

$$I\omega[\![\sigma(\theta)]\!] = \sigma^*_\omega I\omega[\![\theta]\!]$$

Proof of Lemma 2. The proof is by structural induction on ϕ and the structure on σ, simultaneously with Lemma 6.

1. $\omega \in I[\![\sigma(\theta \geq \eta)]\!]$ iff $\omega \in I[\![\sigma(\theta) \geq \sigma(\eta)]\!]$ iff $I\omega[\![\sigma(\theta)]\!] \geq I\omega[\![\sigma(\eta)]\!]$, by Lemma 5, iff $\sigma^*_\omega I\omega[\![\theta]\!] \geq \sigma^*_\omega I\omega[\![\eta]\!]$ iff $\sigma^*_\omega I\omega[\![\theta \geq \eta]\!]$.

2. Let $p \in \sigma$. Then $\omega \in I\llbracket\sigma(p(\theta))\rrbracket$ iff $\omega \in I\llbracket(\sigma(p))\big(\sigma(\theta)\big)\rrbracket$ iff $\omega \in I\llbracket\{\cdot \mapsto \sigma(\theta)\}(\sigma p(\cdot))\rrbracket$ iff, by IH, $\omega \in I_{\cdot}^{d}\llbracket\sigma p(\cdot)\rrbracket$ iff $d \in \sigma^*_\omega I(p)$ iff $(\sigma^*_\omega I\omega\llbracket\theta\rrbracket) \in \sigma^*_\omega I(p)$ iff $\omega \in \sigma^*_\omega I\llbracket p(\theta)\rrbracket$ with $d \overset{\text{def}}{=} I\omega\llbracket\sigma(\theta)\rrbracket = \sigma^*_\omega I\omega\llbracket\theta\rrbracket$ by using Lemma 5 for $\sigma(\theta)$ and by using the induction hypothesis for $\{\cdot \mapsto \sigma(\theta)\}(\sigma p(\cdot))$ on the possibly bigger formula $\sigma p(\cdot)$ but the structurally simpler uniform substitution $\{\cdot \mapsto \sigma(\theta)\}(\dots)$ that is a mere substitution on function symbol $\cdot$ of arity zero, not a substitution of predicates.

3. Let $q \notin \sigma$. Then $\omega \in I\llbracket\sigma(q(\theta))\rrbracket$ iff $\omega \in I\llbracket q(\sigma(\theta))\rrbracket$ iff $\big(I\omega\llbracket\sigma(\theta)\rrbracket\big) \in I(q)$ so, by Lemma 5, that holds iff $\big(\sigma^*_\omega I\omega\llbracket\theta\rrbracket\big) \in I(q)$ iff $\big(\sigma^*_\omega I\omega\llbracket\theta\rrbracket\big) \in \sigma^*_\omega I(q)$ iff $\omega \in \sigma^*_\omega I\llbracket q(\theta)\rrbracket$ since $I(q) = \sigma^*_\omega I(q)$ as the interpretation of q does not change in $\sigma^*_\omega I$ when $q \notin \sigma$.

4. $\omega \in I\llbracket\sigma(\neg\phi)\rrbracket$ iff $\omega \in I\llbracket\neg\sigma(\phi)\rrbracket$ iff $\omega \notin I\llbracket\sigma(\phi)\rrbracket$, so by IH, iff $\omega \notin \sigma^*_\omega I\llbracket\phi\rrbracket$ iff $\omega \in \sigma^*_\omega I\llbracket\neg\phi\rrbracket$

5. $\omega \in I\llbracket\sigma(\phi \land \psi)\rrbracket$ iff $\omega \in I\llbracket\sigma(\phi) \land \sigma(\psi)\rrbracket$ iff $\omega \in I\llbracket\sigma(\phi)\rrbracket$ and $\omega \in I\llbracket\sigma(\psi)\rrbracket$, by induction hypothesis, iff $\omega \in \sigma^*_\omega I\llbracket\phi\rrbracket$ and $\omega \in \sigma^*_\omega I\llbracket\psi\rrbracket$ iff $\omega \in \sigma^*_\omega I\llbracket\phi \land \psi\rrbracket$

6. $\omega \in I\llbracket\sigma(\exists x\, \phi)\rrbracket$ iff $\omega \in I\llbracket\exists x\, \sigma(\phi)\rrbracket$ (provided that σ is $\{x\}$-admissible for ϕ) iff $\omega_x^d \in I\llbracket\sigma(\phi)\rrbracket$ for some d, so, by induction hypothesis, iff $\omega_x^d \in \sigma^*_{\omega_x^d} I\llbracket\phi\rrbracket$ for some d, which is equivalent to $\omega_x^d \in \sigma^*_\omega I\llbracket\phi\rrbracket$ by Corollary 4 as σ is $\{x\}$-admissible for ϕ and $\omega = \omega_x^d$ on $\{x\}^\complement$. Thus, this is equivalent to $\omega \in \sigma^*_\omega I\llbracket\exists x\, \phi\rrbracket$.

7. The case $\omega \in I\llbracket\sigma(\forall x\, \phi)\rrbracket$ follows by duality $\forall x\, \phi \equiv \neg\exists x\, \neg\phi$, which is respected in the definition of uniform substitutions.

8. $\omega \in I\llbracket\sigma(\langle\alpha\rangle\phi)\rrbracket$ iff $\omega \in I\llbracket\langle\sigma(\alpha)\rangle\sigma(\phi)\rrbracket$ (provided σ is $\text{BV}(\sigma(\alpha))$-admissible for ϕ) iff there is a ν such that $(\omega, \nu) \in I\llbracket\sigma(\alpha)\rrbracket$ and $\nu \in I\llbracket\sigma(\phi)\rrbracket$, which, by Lemma 6 and induction hypothesis, respectively, is equivalent to: there is a ν such that $(\omega, \nu) \in \sigma^*_\omega I\llbracket\alpha\rrbracket$ and $\nu \in \sigma^*_\nu I\llbracket\phi\rrbracket$, which is equivalent to $\omega \in \sigma^*_\omega I\llbracket\langle\alpha\rangle\phi\rrbracket$, because $\nu \in \sigma^*_\nu I\llbracket\phi\rrbracket$ is equivalent to $\omega \in \sigma^*_\omega I\llbracket\phi\rrbracket$ by Corollary 4 as σ is $\text{BV}(\sigma(\alpha))$-admissible for ϕ and $\omega = \omega$ on $\text{BV}(\sigma(\alpha))^\complement$ by Lemma **??** since $(\omega, \nu) \in I\llbracket\sigma(\alpha)\rrbracket$.

9. The case $\omega \in I\llbracket\sigma([\alpha]\phi)\rrbracket$ follows by duality $[\alpha]\phi \equiv \neg\langle\alpha\rangle\neg\phi$, which is respected in the definition of uniform substitutions. □

Lemma 6 (Uniform substitution for programs). *The uniform substitution σ and its adjoint interpretation $\sigma^*_\omega I, \omega$ for I, ω have the same semantics for all* programs α*:*

$$(\omega, \nu) \in I\llbracket\sigma(\alpha)\rrbracket \textit{ iff } (\omega, \nu) \in \sigma^*_\omega I\llbracket\alpha\rrbracket$$

Proof. The proof is by structural induction on α, simultaneously with Lemma 2.

1. $(\omega, \nu) \in I\llbracket\sigma(a)\rrbracket = I\llbracket\sigma a\rrbracket = \sigma^*_\omega I(a) = \sigma^*_\omega I\llbracket a\rrbracket$ for program constant $a \in \sigma$ (the proof is accordingly for $a \notin \sigma$).

2. $(\omega, \nu) \in I\llbracket\sigma(x := \theta)\rrbracket = I\llbracket x := \sigma(\theta)\rrbracket$ iff $\nu = \omega_x^{I\omega\llbracket\sigma(\theta)\rrbracket} = \omega_x^{\sigma^*_\omega I\omega\llbracket\theta\rrbracket}$ by Lemma 5, which is, thus, equivalent to $(\omega, \nu) \in \sigma^*_\omega I\llbracket x := \theta\rrbracket$.

3. $(\omega,\nu) \in I[\![\sigma(?Q)]\!] = I[\![?\sigma(Q)]\!]$ iff $\nu = \omega$ and $\omega \in I[\![\sigma(Q)]\!]$, iff, by Lemma 2, $\nu = \omega$ and $\omega \in \sigma^*_\omega I[\![Q]\!]$, which is equivalent to $(\omega,\nu) \in \sigma^*_\omega I[\![?Q]\!]$.

4. $(\omega,\nu) \in I[\![\sigma(\alpha;\beta)]\!] = I[\![\sigma(\alpha);\sigma(\beta)]\!] = I[\![\sigma(\alpha)]\!] \circ I[\![\sigma(\beta)]\!]$ (provided σ is $BV(\sigma(\alpha))$-admissible for β) iff there is a μ such that $(\omega,\mu) \in I[\![\sigma(\alpha)]\!]$ and $(\mu,\nu) \in I[\![\sigma(\beta)]\!]$, which, by induction hypothesis, is equivalent to $(\omega,\mu) \in \sigma^*_\omega I[\![\alpha]\!]$ and $(\mu,\nu) \in \sigma^*_\mu I[\![\beta]\!]$. Yet, $\sigma^*_\mu I[\![\beta]\!] = \sigma^*_\omega I[\![\beta]\!]$ by Corollary 4, because σ is $BV(\sigma(\alpha))$-admissible for β and $\omega = \nu$ on $BV(\sigma(\alpha))^\complement$ by Lemma **??** since $(\omega,\mu) \in I[\![\sigma(\alpha)]\!]$. Finally, $(\omega,\mu) \in \sigma^*_\omega I[\![\alpha]\!]$ and $(\mu,\nu) \in \sigma^*_\omega I[\![\beta]\!]$ for some μ is equivalent to $(\omega,\nu) \in \sigma^*_\omega I[\![\alpha;\beta]\!]$.

5. $(\omega,\nu) \in I[\![\sigma(\alpha^*)]\!] = I[\![(\sigma(\alpha))^*]\!] = \big(I[\![\sigma(\alpha)]\!]\big)^* = \bigcup_{n\in\mathbb{N}}(I[\![\sigma(\alpha)]\!])^n$ (provided that σ is $BV(\sigma(\alpha))$-admissible for α) iff there are $n \in \mathbb{N}$ and $\nu_0 = \omega, \nu_1, \dots, \nu_n = \nu$ such that $(\nu_i,\nu_{i+1}) \in I[\![\sigma(\alpha)]\!]$ for all $i < n$. By n uses of the induction hypothesis, this is equivalent to $(\nu_i,\nu_{i+1}) \in \sigma^*_{\nu_i} I[\![\alpha]\!]$ for all $i < n$, which is equivalent to $(\nu_i,\nu_{i+1}) \in \sigma^*_\omega I[\![\alpha]\!]$ by Corollary 4 since σ is $BV(\sigma(\alpha))$-admissible for α and $\nu_{i+1} = \nu_i$ on $BV(\sigma(\alpha))^\complement$ by Lemma **??** as $(\nu_i,\nu_{i+1}) \in I[\![\sigma(\alpha)]\!]$ for all $i < n$. Thus, this is equivalent to $(\omega,\nu) \in \sigma^*_\omega I[\![\alpha^*]\!] = \big(\sigma^*_\omega I[\![\alpha]\!]\big)^*$. □

References

[BRV+17] Brandon Bohrer, Vincent Rahli, Ivana Vukotic, Marcus Völp, and André Platzer. Formally verified differential dynamic logic. In Yves Bertot and Viktor Vafeiadis, editors, *Certified Programs and Proofs - 6th ACM SIGPLAN Conference, CPP 2017, Paris, France, January 16-17, 2017*, pages 208–221, New York, 2017. ACM. doi:10.1145/3018610.3018616.

[Chu56] Alonzo Church. *Introduction to Mathematical Logic*. Princeton University Press, Princeton, 1956.

[Pla15] André Platzer. A uniform substitution calculus for differential dynamic logic. In Amy Felty and Aart Middeldorp, editors, *CADE*, volume 9195 of *LNCS*, pages 467–481, Berlin, 2015. Springer. doi:10.1007/978-3-319-21401-6_32.

[Pla17] André Platzer. A complete uniform substitution calculus for differential dynamic logic. *J. Autom. Reas.*, 59(2):219–265, 2017. doi:10.1007/s10817-016-9385-1.

[Pla18] André Platzer. *Logical Foundations of Cyber-Physical Systems*. Springer, Switzerland, 2018. URL: http://www.springer.com/978-3-319-63587-3.