Horn Clauses in Hybrid-Dynamic First-Order Logic

Daniel Găină   Ionuț Țuțu

Institute of Mathematics for Industry, Kyushu University, Japan
Department of Mathematics and Statistics, La Trobe University, Australia

Simion Stoilow Institute of Mathematics of the Romanian Academy, Romania
Department of Computer Science, Royal Holloway University of London, UK

The 15th Theorem Proving and Provers meeting

Reconfiguration paradigm

- The present work is a part of a larger project: logical foundations of reconfiguration paradigm
- In many cases, the applications with reconfigurable features involve safety-critical areas. For example, the new generation of software-driven medical devices such as imaging machines, pill cameras, artificial pacemakers, the insulin infusion pump, etc.
- The safety requirements can be fulfilled only with formal methods
- The safety requirements can be fulfilled only by applying formal methods.
- *"One of the main issues is that there is no real formal method of implementing the reconfiguration of an application*" [Szepesi and Ciocarlie, Theory Appl. Math. Comput. Sci. 2011].

Hybrid-dynamic logics

Safe **reconfigurable systems**	**Medical devices**	**Driver applications**	**Intelligent transportation systems**	**Cloud systems**	
	They interact with users and the environment in which they coexist				

⇑

All developments above are powered by **formal methods**	**Formal methods** are mathematically-based techniques for the specification and verification of software systems, and the only way to ensure safety of applications

⇑

All developments above are powered by **math/logic**	The design of a formal method relies on	1. **model-theoretic studies** 2. **proof-theoretic support**

Reconfigurable systems can be regarded as transition systems in the following way:

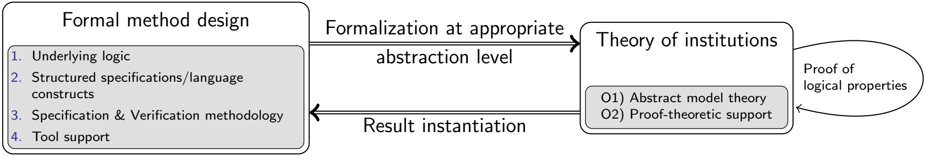
- the configurations are states, and
- switching from one configuration to another is a transition.

Hybrid dynamic logics are modal logics that can describe transitions systems and express the dynamics of (re)configurations:

- the configurations of the software in use today may be modeled with first-order logic, higher-order logic, rewriting logic, etc; these are considered **base logics**;
- the construction of a hybrid logic on top of a base logic is called **hybridization** [Diaconescu and Madeira, Math. Struct. Comput. Sci. 2016].

Formal method design

- **Institution** category-based formalization of the intuitive definition of logical system [Goguen and Burstall, Journal of the ACM 1992]
- **Hybrid institution** formalizes the notion of hybrid logic by supplementing the definition of **institution** with an additional structure to extract (a) nominals and modalities from signatures, and (b) frames from models.



- the design of a formal method consists of several steps depicted in the left node of the figure above
- each step relies on some model-theoretic or proof-theoretic properties which should be defined at an appropriate abstraction level in the framework of hybrid institutions
- the proof of model-theoretic and proof-theoretic properties should be performed within the framework of hybrid institutions as depicted in the right side of the figure above
- once proved the abstract results are instantiated to concrete logical systems

Preparation status

- Definition of hybrid institution and foundation of logic programming in hybrid institutions [Gaina, Theor. Comput. Sci. 2017]
- Abstract proof calculus for hybrid institutions whose sentences are Horn clauses [Gaina, Formal Asp. Comput. 2017]
- Proof calculus for the Horn clauses of hybrid-dynamic first-order logic [Gaina and Tutu, TABLEAUX 2019]
- Forcing in hybrid institutions [Gaina, Journal of the ACM accepted]

The following figure shows the advancements for this project and how this work will be completed in the present project.

	Hybrid institutions	Hybrid-dynamic institutions
Proof calculi (for all sentences)	[Gaina, Journal of the ACM accepted]	future work
Birkhoff proof calculi (for Horn clauses only)	[Gaina, Formal Asp. Comput. 2017]	[Gaina and Tutu, TABLEAUX 2019] + future work
Structured specifications (initiality + imports)	[Gaina, Theor. Comput. Sci. 2017]	[Gaina and Tutu, TABLEAUX 2019] + future work

Semantics first: Kripke models

- mathematical structures used to model the behaviour of a system
- underlie the semantics of modal, hybrid, dynamic and temporal logics

- typically, a Kripke structure consists in a (hyper)graph where:
  - each node (possible world) represents a state of the system
  - each edge (part of an accessibility relation) represents a transition

- the original definition has been extended in various ways:
  - complex algebraic structures as labels of the states
  - structured actions as labels of transitions
  - model constraints

Hybrid-Dynamic First-Order Logic with user-defined sharing

A logic for specifying and reasoning about Kripke structures

... obtained by enriching first-order logic
with features that are characteristic to
hybrid logics (nominals and local-satisfaction operators) and
dynamic logics (structured actions over modalities)

... and with a number of distinctive attributes:

- a first-order structure on possible worlds
- supports sharing between possible worlds / rigidity constraints
- hybrid terms (annotated with nominals)

Signatures

Definition. An HDFOL-signature is a tuple $\Delta = (\Sigma^n, \Sigma^r \subseteq \Sigma)$, where:

- $\Sigma^n = (F^n, P^n)$ is a single-sorted first-order signature of nominals, where F^n_i is a set of nominal operations of arity $i \in \mathbb{N}$
- $\Sigma^r = (S^r, F^r, P^r)$ is a many-sorted signature of rigid symbols, where $F^r_{ar \to s}$ is a set of operations of arity $ar \in S^{r*}$ and sort $s \in S^r$
- $\Sigma = (S, F, P)$ is a many-sorted first-order signature of both rigid and flexible symbols

An HDFOL-signature morphism $\varphi\colon \Delta \to \Delta'$ consists of a pair of first-order signature morphisms $\varphi^n\colon \Sigma^n \to \Sigma'^n$ and $\varphi\colon \Sigma \to \Sigma'$ such that $\varphi(\Sigma^r) \subseteq \Sigma'^r$.

Semantics

Definition. A Kripke model of $\Delta = (\Sigma^n, \Sigma^r \subseteq \Sigma)$ is a pair $\langle W, M \rangle$:

- W is a Σ^n-model, whose carrier set we denote by $|W|$
- $M = (M_w)_{w \in |W|}$ is a family of Σ-models, indexed by worlds, such that $M_{w_1,\varsigma} = M_{w_2,\varsigma}$ for all $w_1, w_2 \in |W|$ and symbols ς in Σ^r

(rigid symbols have the same interpretation across possible worlds)

A homomorphism $h \colon \langle V, N \rangle \to \langle W, M \rangle$ is also a pair, consisting of first-order homomorphisms $h \colon V \to W$ and $h_v \colon N_v \to M_{h(v)}$, for every world $v \in |V|$, such that $h_{v_1,s} = h_{v_2,s}$ for all $v_1, v_2 \in |V|$ and $s \in S^r$.

Syntax

The set actions over Δ is defined in an inductive fashion, according to the following grammar:

$$\mathfrak{a} ::= \lambda \in P_2^n \mid \mathfrak{a} \,\mathrm{;}\, \mathfrak{a} \mid \mathfrak{a} \cup \mathfrak{a} \mid \mathfrak{a}^*$$

The atomic sentences defined over a signature Δ are given by:

$$\rho ::= \overbrace{k_1 = k_2 \mid \lambda(k')}^{\text{nominal equations and relations}} \mid \underbrace{t_1 =_{k,s} t_2 \mid \varpi(t) \mid \pi(k;t)}_{\text{hybrid equations and relations}}$$

Full sentences over Δ are built from atomic sentences according to the following grammar:

$$\gamma ::= \rho \mid \mathfrak{a}(k_1, k_2) \mid @_k\, \gamma \mid \neg\gamma \mid \bigwedge \Gamma \mid {\downarrow} z \cdot \gamma' \mid \forall X \cdot \gamma''$$

where z is a nominal variable, X is a set of variables, and γ', γ'' are sentences over the extended signatures $\Delta[z]$ and $\Delta[X]$, respectively.

Local-satisfaction relation

Definition. Given a Kripke model $\langle W, M\rangle$ of a signature Δ and a possible world $w \in |W|$, we have, for atomic sentences:

- $\langle W, M\rangle \models^w k_1 = k_2$ iff $W_{k_1} = W_{k_2}$
- $\langle W, M\rangle \models^w \lambda(k)$ iff $W_k \in W_\lambda$
- $\langle W, M\rangle \models^w t_1 =_k t_2$ iff $\langle W, M\rangle_{t_1} = \langle W, M\rangle_{t_2}$
- $\langle W, M\rangle \models^w \varpi(t)$ iff $\langle W, M\rangle_t \in M_{w,\varpi}$
- $\langle W, M\rangle \models^w \pi(k; t)$ iff $\langle W, M\rangle_t \in M_{w',\pi}$, where $w' = W_k$

Local-satisfaction relation

Definition. Given a Kripke model $\langle W, M\rangle$ of a signature Δ and a possible world $w \in |W|$, we have, for full sentences:

- $\langle W, M\rangle \models^w \mathfrak{a}(k_1, k_2)$ iff $(W_{k_1}, W_{k_2}) \in W_{\mathfrak{a}}$
- $\langle W, M\rangle \models^w @_k\,\gamma$ iff $\langle W, M\rangle \models^{w'} \gamma$, where $w' = W_k$
- $\langle W, M\rangle \models^w \neg\gamma$ iff $\langle W, M\rangle \not\models^w \gamma$
- $\langle W, M\rangle \models^w \bigwedge\Gamma$ iff $\langle W, M\rangle \models^w \gamma$ for all $\gamma \in \Gamma$
- $\langle W, M\rangle \models^w {\downarrow} z \cdot \gamma$ iff $\langle W, M\rangle^{z\leftarrow w} \models^w \gamma$
- $\langle W, M\rangle \models^w \forall X \cdot \gamma$ iff $(W', M') \models^w \gamma$ for all $\Delta[X]$-expansions (W', M') of $\langle W, M\rangle$

Expressivity and relationship to other modal logics

Support for conventional modal operators

- $[\mathfrak{a}]\gamma \triangleq {\downarrow} z \cdot \forall z' \cdot \mathfrak{a}(z, z') \Rightarrow @_{z'}\, \gamma$
- $\langle\mathfrak{a}\rangle\gamma \triangleq {\downarrow} z \cdot \exists z' \cdot \mathfrak{a}(z, z') \wedge @_{z'}\, \gamma$

Support for (linear) temporal operators

- $\bigcirc\, \gamma \triangleq {\downarrow} z \cdot @_{\mathsf{next}(z)}\, \gamma$
- $\rho \,\mathsf{Until}\, \gamma \triangleq \exists z \cdot \Diamond(z \wedge \gamma) \wedge \Box(\Diamond z \Rightarrow \rho)$

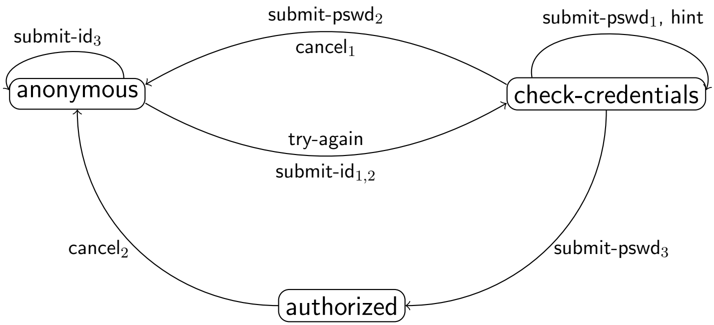
Hybrid (annotated) terms vs ordinary terms

- $c_1(k) = \sigma(k; c_2(k))$ is equivalent to $@_k\, c_1 = \sigma(c_2)$
- $c_3(k) = \sigma(k; c_4(k_0))$ is equivalent to $\exists x \cdot @_{k_0}\, x = c_4 \wedge @_k\, c_3 = \sigma(x)$

Implicit vs explicit dependence on possible worlds

- the nominal sentence k is equivalent to ${\downarrow} z \cdot z = k$
- the store sentence ${\downarrow} z \cdot \gamma$ is equivalent to $\exists z \cdot (z \wedge \gamma)$

Event-based transition system



Events:

- submit-id, submit-pswd, cancel
- try-again, hint

Attributes:

- $\underbrace{\text{status, uid, attpts,}}_{\text{observations}}$
- $\underbrace{\text{type-id, type-pswd}}_{\text{random values}}$

Setting the stage for Birkhoff completeness

Goal: syntactic characterization of the satisfiability relations $\Gamma \models_\Delta \gamma$

$$\langle W, M\rangle \models \gamma \text{ for all } \Delta\text{-models } \langle W, M\rangle \text{ such that } \langle W, M\rangle \models \Gamma$$

- both Γ and γ belong to the Horn-clause fragment of HDFOL

Definition. By Horn clause, we mean a sentence obtained from atomic sentences by repeated applications of the following sentence-building operators, in any order:

- *retrieve*
- *implication* (hypothesis: only atoms or action relations)
- *store*
- *universal quantification*
- *necessity*, *next*

Setting the stage for Birkhoff completeness

Goal: syntactic characterization of the satisfiability relations $\Gamma \models_\Delta \gamma$

$$\langle W, M\rangle \models \gamma \text{ for all } \Delta\text{-models } \langle W, M\rangle \text{ such that } \langle W, M\rangle \models \Gamma$$

- both Γ and γ belong to the Horn-clause fragment of HDFOL

Definition. By Horn clause, we mean a sentence obtained from atomic sentences by repeated applications of the following sentence-building operators, in any order:

- *retrieve*
- *implication* (hypothesis: only atoms or action relations)
- *store*
- *universal quantification*
- *necessity*, *next*

A layered approach

- develop progressively a series of syntactic entailment relations

$$\Gamma \vdash \gamma$$

- where each layer builds on the previous one to allow for more general antecedents or consequents
- three major steps / entailment relations

  Atomic completeness: both Γ and γ are atomic
  Quasi-completeness: Γ is arbitrary, but γ is atomic
  Horn-clause completeness: both Γ and γ are arbitrary

- all are sound and complete; only the first two are also compact
- atomic completeness is much more difficult to establish than it may seem at first sight...

A layered approach

- develop progressively a series of syntactic entailment relations

$$\Gamma \vdash \gamma$$

- where each layer builds on the previous one to allow for more general antecedents or consequents
- three major steps / entailment relations

  Atomic completeness: both Γ and γ are atomic
  Quasi-completeness: Γ is arbitrary, but γ is atomic
  Horn-clause completeness: both Γ and γ are arbitrary

- all are sound and complete; only the first two are also compact
- atomic completeness is much more difficult to establish than it may seem at first sight...

A layered approach

- develop progressively a series of syntactic entailment relations

$$\Gamma \vdash \gamma$$

- where each layer builds on the previous one to allow for more general antecedents or consequents
- three major steps / entailment relations

  Atomic completeness: both Γ and γ are atomic
  Quasi-completeness: Γ is arbitrary, but γ is atomic
  Horn-clause completeness: both Γ and γ are arbitrary

- all are sound and complete; only the first two are also compact
- atomic completeness is much more difficult to establish than it may seem at first sight...

Atomic completeness

Proof rules

Lemma. For every set Γ of nominal equations over a signature Δ, there exists a reachable initial model (W^Γ, M^Γ) such that

$$\Gamma \models \rho \quad \text{iff} \quad (W^\Gamma, M^\Gamma) \models \rho \quad \text{iff} \quad \Gamma \vdash \rho$$

for all nominal or hybrid equations ρ over Δ.

Theorem (Atomic completeness). Every set Γ of atomic sentences over a signature Δ has a reachable initial model (W^Γ, M^Γ) such that

$$\Gamma \models \rho \quad \text{iff} \quad (W^\Gamma, M^\Gamma) \models \rho \quad \text{iff} \quad \Gamma \vdash \rho$$

for all atomic sentences ρ over Δ.

Quasi-completeness

Proof rules

Theorem (Quasi-completeness). Let:

- Γ be a set of clauses over a signature Δ,
- $\Gamma_0 = \{\rho \in \mathrm{Sen}^{\mathsf{HDCLS}}(\Delta) \mid \Gamma \vdash \rho \;\&\; \rho$ is atomic$\}$,
- $(W^{\Gamma_0}, M^{\Gamma_0})$ a reachable initial model of Γ_0 as before.

Then the following statements are equivalent:

1. $\Gamma \models \rho$
2. $(W^{\Gamma_0}, M^{\Gamma_0}) \models \rho$
3. $\Gamma \vdash \rho$

Horn-clause completeness

Proof rules

Theorem (Birkhoff completeness). The entailment relation generated by the rules presented thus far is sound and complete.

$$\Gamma \models \gamma \qquad \text{if and only if} \qquad \Gamma \vdash \gamma$$

Moreover, in the absence of $(\mathsf{Star}_\mathsf{I})$, it is also compact.

Proposition (Lack of compactness). HDCLS does not admit an entailment relation that is sound, complete, and also compact.

Conclusions

To sum up

- we have introduced a hybrid-dynamic first-order logic
- seen how it relates to modal/temporal/hybrid logics
- presented a sound and complete calculus for its Horn fragment
- touched upon the limits of compactness

Ongoing work

- decidability properties and support for executable specifications
- case studies & a prototype implementation

Conclusions

To sum up

- we have introduced a hybrid-dynamic first-order logic
- seen how it relates to modal/temporal/hybrid logics
- presented a sound and complete calculus for its Horn fragment
- touched upon the limits of compactness

Ongoing work

- decidability properties and support for executable specifications
- case studies & a prototype implementation

Thank you!

Atomic completeness

Rules for nominal terms

$$(\mathsf{R^n}) \quad \frac{}{\Gamma \vdash k = k} \qquad\qquad (\mathsf{S^n}) \quad \frac{\Gamma \vdash k_1 = k_2}{\Gamma \vdash k_2 = k_1}$$

$$(\mathsf{T^n}) \quad \frac{\Gamma \vdash k_1 = k_2 \qquad \Gamma \vdash k_2 = k_3}{\Gamma \vdash k_1 = k_3}$$

$$(\mathsf{F^n}) \quad \frac{\Gamma \vdash k_1 = k_2}{\Gamma \vdash o(k_1) = o(k_2)} \qquad\qquad (\mathsf{P^n})$$

$$\frac{\Gamma \vdash \lambda(k_1) \qquad \Gamma \vdash k_1 = k_2}{\Gamma \vdash \lambda(k_2)}$$

Atomic completeness

Rules for sharing

$$(\mathsf{W}^{\mathsf{h}}) \quad \frac{\Gamma \vdash k = k'}{\Gamma \vdash t =_{k,s} \delta_{k'/k}(t)} \quad \text{where } s \in S^r$$

$$(\mathsf{W}^{\mathsf{r}}) \quad \frac{\Gamma \vdash t_1 =_{k_1,s} t_2}{\Gamma \vdash t_1 =_{k_2,s} t_2} \quad \text{where } s \in S^r$$

$$(\mathsf{W}^{\mathsf{f}}) \quad \frac{\Gamma \vdash k = k' \qquad \Gamma \vdash t_1 =_k t_2}{\Gamma \vdash \delta_{k/k'}(t_1) =_{k'} \delta_{k/k'}(t_2)}$$

Atomic completeness

Rules for hybrid terms

$$(\mathsf{F}^{\mathsf{r}})\quad \frac{\Gamma \vdash t_1 =_{k,ar} t_2}{\Gamma \vdash \sigma(t_1) =_{k,s} \sigma(t_2)} \qquad (\mathsf{F}^{\mathsf{f}})\quad \frac{\Gamma \vdash t_1 =_{k,ar} t_2}{\Gamma \vdash \sigma(k;t_1) =_{k,s} \sigma(k;t_2)}$$

$$(\mathsf{P}^{\mathsf{r}})\quad \frac{\Gamma \vdash t_1 =_k t_2 \quad \Gamma \vdash \pi(t_1)}{\Gamma \vdash \pi(t_2)} \qquad (\mathsf{P}^{\mathsf{f}})\quad \frac{\Gamma \vdash t_1 =_k t_2 \qquad \Gamma \vdash \pi(k;t_1)}{\Gamma \vdash \pi(k;t_2)}$$

$$(\mathsf{P}^{\mathsf{h}})\quad \frac{\Gamma \vdash k_1 = k_2 \quad \Gamma \vdash \pi(k_1;t_1)}{\Gamma \vdash \pi(k_2;\delta_{k_1/k_2}(t_1))} \qquad (\mathsf{Ret_0})\quad \frac{\Gamma \vdash @_k\, \rho}{\Gamma \vdash \rho}$$

Quasi-completeness

Rules for action relations

$$(\mathrm{Comp})\quad \frac{\Gamma \vdash \mathfrak{a}_1(k_1,k_2) \qquad \Gamma \vdash \mathfrak{a}_2(k_2,k_3)}{\Gamma \vdash (\mathfrak{a}_1 \mathbin{;} \mathfrak{a}_2)(k_1,k_3)}$$

$$(\mathrm{Union})\quad \frac{\Gamma \vdash \mathfrak{a}_i(k_1,k_2)}{\Gamma \vdash (\mathfrak{a}_1 \cup \mathfrak{a}_2)(k_1,k_2)}$$

$$(\mathrm{Refl})\quad \frac{\Gamma \vdash k_1 = k_2}{\Gamma \vdash \mathfrak{a}^*(k_1,k_2)} \qquad (\mathrm{Star})\quad \frac{\Gamma \vdash \mathfrak{a}(k_i,k_{i+1}) \text{ for } 0 \leq i < n}{\Gamma \vdash \mathfrak{a}^*(k_0,k_n)}$$

Quasi-completeness

Rules for Horn clauses

$$(\mathsf{Ret}_@)\quad \frac{\Gamma \vdash @_{k_1} @_{k_2}\, \gamma}{\Gamma \vdash @_{k_2}\, \gamma} \qquad\qquad (\mathsf{Ret}_\mathsf{I})\quad \frac{\Gamma \vdash \gamma}{\Gamma \vdash @_k\, \gamma}$$

$$(\mathsf{Imp}_\mathsf{E})\quad \frac{\Gamma \vdash @_k\, (\bigwedge H \Rightarrow \gamma)}{\Gamma \cup H \vdash @_k\, \gamma}$$

$$(\mathsf{Store}_\mathsf{E})\quad \frac{\Gamma \vdash @_k \downarrow z \cdot \gamma}{\Gamma \vdash @_k\, \theta_{z \leftarrow k}(\gamma)} \qquad\qquad (\mathsf{Subst}_\mathsf{q})\quad \frac{\Gamma \vdash @_k\, \forall X \cdot \gamma}{\Gamma \vdash @_k\, \theta(\gamma)}$$

Horn-clause completeness

Additional rules for Horn clauses

$$(\mathsf{Ret_E})\quad \frac{\Gamma \vdash_{\Delta[z]} @_z\, \gamma}{\Gamma \vdash_{\Delta} \gamma} \qquad\qquad (\mathsf{Imp_I})\quad \frac{\Gamma \cup H \vdash @_k\, \gamma}{\Gamma \vdash @_k\, (\bigwedge H \Rightarrow \gamma)}$$

$$(\mathsf{Store_I})\quad \frac{\Gamma \vdash @_k\, \theta_{z\leftarrow k}(\gamma)}{@_k \downarrow z \cdot \gamma} \qquad\qquad (\mathsf{Quant_I})\quad \frac{\Gamma \vdash_{\Delta[X]} @_k\, \gamma}{\Gamma \vdash_{\Delta} @_k\, \forall X \cdot \gamma}$$

Horn-clause completeness

Additional rules for action relations

$$(\mathsf{Comp_I})\quad \frac{E \cup \{\mathfrak{a}_1(k_1, z), \mathfrak{a}_2(z, k_2)\} \vdash_{\Delta[z]} e}{E \cup \{(\mathfrak{a}_1 \fatsemi \mathfrak{a}_2)(k_1, k_2)\} \vdash_{\Delta} e}$$

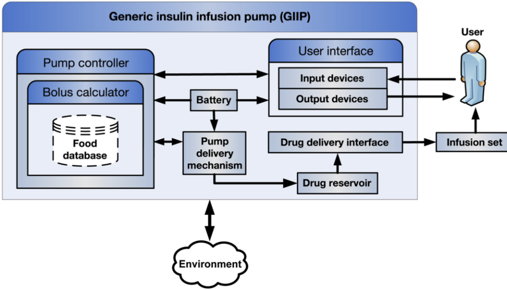
$$(\mathsf{Union_I})\quad \frac{E \cup \{\mathfrak{a}_i(k_1, k_2)\} \vdash e \text{ for } i \in \{1, 2\}}{E \cup \{(\mathfrak{a}_1 \cup \mathfrak{a}_2)(k_1, k_2)\} \vdash e}$$

$$(\mathsf{Star_I})\quad \frac{E \cup \{\mathfrak{a}^n(k_1, k_2)\} \vdash e \text{ for all } n \in \mathbb{N}}{E \cup \{\mathfrak{a}^*(k_1, k_2)\} \vdash e}$$

Generic insulin infusion pump

Pump controller - an abstract representation of generic insulin pump software

- Main functionality: command the pump delivery mechanism to propel insulin stored in the drug reservoir to the patient through the drug delivery interface and the infusion set
- Overall responsibility: ensure correct operation of the model



More about pump controller functionality:

- interacts with the patient through a user interface
- it alerts the patient when abnormal conditions arise
- recommends appropriate bolus dosages with the help of a bolus calculator and a food database
- manages and checks parameters and programs related to insulin administration;
- logs important data and events during pump use to facilitate clinical use analysis and problem diagnosis