Reasoning with movement based on qualitative representation

Przemysław Wałęga

2014,

Outline

Propositional Dynamic Logic

PDL language

- $\mathbb{V}$ – propositional variables,
- $\mathbb{RC}$ – relational constants (atomic programs),
- $\{\cup, ;, ?, ^*\}$ relational operations
  - $\cup$ nondeterministic choice,
  - $;$ sequential composition,
  - * iteration,
  - $?$ test performance,
- $\{\neg, \wedge, \vee, \rightarrow, [], \langle\rangle, \mathbf{0}\}$ propositional operators,

Propositional Dynamic Logic

PDL language

- $\mathbb{V}$ – propositional variables,
- $\mathbb{RC}$ – relational constants (atomic programs),
- $\{\cup, ;, ?, ^*\}$ relational operations
  - $\cup$ nondeterministic choice,
  - ; sequential composition,
  - * iteration,
  - ? test performance,
- $\{\neg, \wedge, \vee, \rightarrow, [], \langle\rangle, \mathbf{0}\}$ propositional operators,

Propositional Dynamic Logic

PDL language

- $\mathbb{V}$ – propositional variables,
- $\mathbb{RC}$ – relational constants (atomic programs),
- $\{\cup, ;, ?, ^*\}$ relational operations
  - $\cup$ nondeterministic choice,
  - $;$ sequential composition,
  - * iteration,
  - $?$ test performance,
- $\{\neg, \wedge, \vee, \rightarrow, [], \langle\rangle, \mathbf{0}\}$ propositional operators,

Propositional Dynamic Logic

PDL language

- $\mathbb{V}$ – propositional variables,
- $\mathbb{RC}$ – relational constants (atomic programs),
- $\{\cup, ;, ?, ^*\}$ relational operations
  - $\cup$ nondeterministic choice,
  - ; sequential composition,
  - * iteration,
  - ? test performance,
- $\{\neg, \wedge, \vee, \rightarrow, [], \langle\rangle, \mathbf{0}\}$ propositional operators,

Propositional Dynamic Logic

PDL language

- $\mathbb{V}$ – propositional variables,
- $\mathbb{RC}$ – relational constants (atomic programs),
- $\{\cup, ;, ?, ^*\}$ relational operations
  - $\cup$ nondeterministic choice,
  - $;$ sequential composition,
  - * iteration,
  - $?$ test performance,
- $\{\neg, \wedge, \vee, \rightarrow, [], \langle\rangle, \mathbf{0}\}$ propositional operators,

Propositional Dynamic Logic

PDL language

- $\mathbb{V}$ – propositional variables,
- $\mathbb{RC}$ – relational constants (atomic programs),
- $\{\cup, ;, ?, ^*\}$ relational operations
  - $\cup$ nondeterministic choice,
  - ; sequential composition,
  - * iteration,
  - ? test performance,
- $\{\neg, \wedge, \vee, \rightarrow, [], \langle\rangle, \mathbf{0}\}$ propositional operators,

Propositional Dynamic Logic

PDL language

- $\mathbb{V}$ – propositional variables,
- $\mathbb{RC}$ – relational constants (atomic programs),
- $\{\cup, ;, ?, ^*\}$ relational operations
  - $\cup$ nondeterministic choice,
  - ; sequential composition,
  - $*$ iteration,
  - ? test performance,
- $\{\neg, \wedge, \vee, \rightarrow, [], \langle\rangle, \mathbf{0}\}$ propositional operators,

Propositional Dynamic Logic

PDL language

- $\mathbb{V}$ – propositional variables,
- $\mathbb{RC}$ – relational constants (atomic programs),
- $\{\cup, ;, ?, ^*\}$ relational operations
  - $\cup$ nondeterministic choice,
  - $;$ sequential composition,
  - * iteration,
  - $?$ test performance,
- $\{\neg, \wedge, \vee, \rightarrow, [], \langle\rangle, \mathbf{0}\}$ propositional operators,

Propositional Dynamic Logic

PDL language

- $\mathbb{V}$ – propositional variables,
- $\mathbb{RC}$ – relational constants (atomic programs),
- $\{\cup, ;, ?, ^*\}$ relational operations
  - $\cup$ nondeterministic choice,
  - $;$ sequential composition,
  - * iteration,
  - $?$ test performance,
- $\{\neg, \wedge, \vee, \rightarrow, [], \langle\rangle, \mathbf{0}\}$ propositional operators,

Propositional Dynamic Logic

Intuitive meaning

- $[\alpha]\varphi$ – "It is necessary that φ after executing α"
- $\alpha \cup \beta$ – "Choose either α or β nondeterministically and execute it"
- $\alpha; \beta$ – "Execute α, then execute β "
- α^* – "Execute α a nondeterministically chosen finite number of times"
- $\varphi?$ – "Test φ proceed if true, fail if false"

Remark

$\langle\rangle$ is definable by $[]$:

- $\langle\alpha\rangle\varphi \stackrel{\text{df}}{=} \neg[\alpha]\neg\varphi$

Propositional Dynamic Logic

Intuitive meaning

- $[\alpha]\varphi$ – "It is necessary that φ after executing α"
- $\alpha \cup \beta$ – "Choose either α or β nondeterministically and execute it"
- $\alpha; \beta$ – "Execute α, then execute β "
- α^* – "Execute α a nondeterministically chosen finite number of times"
- $\varphi?$ – "Test φ proceed if true, fail if false"

Remark

$\langle\rangle$ is definable by $[]$:

- $\langle\alpha\rangle\varphi \stackrel{\text{df}}{=} \neg[\alpha]\neg\varphi$

Propositional Dynamic Logic

Intuitive meaning

- $[\alpha]\varphi$ – "It is necessary that φ after executing α"
- $\alpha \cup \beta$ – "Choose either α or β nondeterministically and execute it"
- $\alpha; \beta$ – "Execute α, then execute β "
- α^* – "Execute α a nondeterministically chosen finite number of times"
- $\varphi?$ – "Test φ proceed if true, fail if false"

Remark

$\langle\rangle$ is definable by $[]$:

- $\langle\alpha\rangle\varphi \stackrel{\text{df}}{=} \neg[\alpha]\neg\varphi$

Propositional Dynamic Logic

Intuitive meaning

- $[\alpha]\varphi$ – "It is necessary that φ after executing α"
- $\alpha \cup \beta$ – "Choose either α or β nondeterministically and execute it"
- $\alpha;\beta$ – "Execute α, then execute β "
- α^* – "Execute α a nondeterministically chosen finite number of times"
- $\varphi?$ – "Test φ proceed if true, fail if false"

Remark

$\langle\rangle$ is definable by $[]$:

- $\langle\alpha\rangle\varphi \overset{\text{df}}{=} \neg[\alpha]\neg\varphi$

Propositional Dynamic Logic

Intuitive meaning

- $[\alpha]\varphi$ – "It is necessary that φ after executing α"
- $\alpha \cup \beta$ – "Choose either α or β nondeterministically and execute it"
- $\alpha; \beta$ – "Execute α, then execute β "
- α^* – "Execute α a nondeterministically chosen finite number of times"
- $\varphi?$ – "Test φ proceed if true, fail if false"

Remark

$\langle\rangle$ is definable by $[]$:

- $\langle\alpha\rangle\varphi \stackrel{\text{df}}{=} \neg[\alpha]\neg\varphi$

Propositional Dynamic Logic

Intuitive meaning

- $[\alpha]\varphi$ – "It is necessary that φ after executing α"
- $\alpha \cup \beta$ – "Choose either α or β nondeterministically and execute it"
- $\alpha; \beta$ – "Execute α, then execute β "
- α^* – "Execute α a nondeterministically chosen finite number of times"
- $\varphi?$ – "Test φ proceed if true, fail if false"

Remark

$\langle\rangle$ is definable by $[]$:

- $\langle\alpha\rangle\varphi \stackrel{\mathrm{df}}{=} \neg[\alpha]\neg\varphi$

Propositional Dynamic Logic

Intuitive meaning

- $[\alpha]\varphi$ – "It is necessary that φ after executing α"
- $\alpha \cup \beta$ – "Choose either α or β nondeterministically and execute it"
- $\alpha; \beta$ – "Execute α, then execute β "
- α^* – "Execute α a nondeterministically chosen finite number of times"
- $\varphi?$ – "Test φ proceed if true, fail if false"

Remark

$\langle\rangle$ is definable by $[]$:

- $\langle\alpha\rangle\varphi \stackrel{\text{df}}{=} \neg[\alpha]\neg\varphi$

Propositional Dynamic Logic

Intuitive meaning

- $[\alpha]\varphi$ – "It is necessary that φ after executing α"
- $\alpha \cup \beta$ – "Choose either α or β nondeterministically and execute it"
- $\alpha; \beta$ – "Execute α, then execute β "
- α^* – "Execute α a nondeterministically chosen finite number of times"
- $\varphi?$ – "Test φ proceed if true, fail if false"

Remark

$\langle\rangle$ is definable by $[]$:

- $\langle\alpha\rangle\varphi \stackrel{\text{df}}{=} \neg[\alpha]\neg\varphi$

Propositional Dynamic Logic

Intuitive meaning

- $[\alpha]\varphi$ – "It is necessary that φ after executing α"
- $\alpha \cup \beta$ – "Choose either α or β nondeterministically and execute it"
- $\alpha; \beta$ – "Execute α, then execute β "
- α^* – "Execute α a nondeterministically chosen finite number of times"
- $\varphi?$ – "Test φ proceed if true, fail if false"

Remark

$\langle\rangle$ is definable by $[]$:

- $\langle\alpha\rangle\varphi \stackrel{\text{df}}{=} \neg[\alpha]\neg\varphi$

Propositional Dynamic Logic

Syntax

for formulas φ , ψ and set of atomic formulas Φ_0
programs α , β and set of atomic programs Π_0
set of formulas Φ and set of programs Π are the smallest sets such that:

- $\Phi_0 \subseteq \Phi$
- $\Pi_0 \subseteq \Pi$
- if $\varphi, \psi \in \Phi$, then $\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi, \varphi \rightarrow \psi \in \Phi$
- if $\alpha, \beta \in \Pi$, then $\alpha \cup \beta, \alpha; \beta, \alpha^*, \in \Pi$
- if $\alpha \in \Pi$ and $\varphi \in \Phi$, then $[\alpha]\varphi \in \Phi$
- if $\varphi \in \Phi$, then $\varphi? \in \Pi$

Propositional Dynamic Logic

Syntax

for formulas φ , ψ and set of atomic formulas Φ_0
programs α , β and set of atomic programs Π_0
set of formulas Φ and set of programs Π are the smallest sets such that:

- $\Phi_0 \subseteq \Phi$
- $\Pi_0 \subseteq \Pi$
- if $\varphi, \psi \in \Phi$, then $\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi, \varphi \rightarrow \psi \in \Phi$
- if $\alpha, \beta \in \Pi$, then $\alpha \cup \beta, \alpha; \beta, \alpha^*, \in \Pi$
- if $\alpha \in \Pi$ and $\varphi \in \Phi$, then $[\alpha]\varphi \in \Phi$
- if $\varphi \in \Phi$, then $\varphi? \in \Pi$

Propositional Dynamic Logic

Syntax

for formulas φ , ψ and set of atomic formulas Φ_0
programs α , β and set of atomic programs Π_0
set of formulas Φ and set of programs Π are the smallest sets such that:

- $\Phi_0 \subseteq \Phi$
- $\Pi_0 \subseteq \Pi$
- if $\varphi, \psi \in \Phi$, then $\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi, \varphi \rightarrow \psi \in \Phi$
- if $\alpha, \beta \in \Pi$, then $\alpha \cup \beta, \alpha; \beta, \alpha^*, \in \Pi$
- if $\alpha \in \Pi$ and $\varphi \in \Phi$, then $[\alpha]\varphi \in \Phi$
- if $\varphi \in \Phi$, then $\varphi? \in \Pi$

Propositional Dynamic Logic

Syntax

for formulas φ , ψ and set of atomic formulas Φ_0
programs α , β and set of atomic programs Π_0
set of formulas Φ and set of programs Π are the smallest sets such that:

- $\Phi_0 \subseteq \Phi$
- $\Pi_0 \subseteq \Pi$
- if $\varphi, \psi \in \Phi$, then $\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi, \varphi \rightarrow \psi \in \Phi$
- if $\alpha, \beta \in \Pi$, then $\alpha \cup \beta, \alpha; \beta, \alpha^*, \in \Pi$
- if $\alpha \in \Pi$ and $\varphi \in \Phi$, then $[\alpha]\varphi \in \Phi$
- if $\varphi \in \Phi$, then $\varphi? \in \Pi$

Propositional Dynamic Logic

Syntax

for formulas φ , ψ and set of atomic formulas Φ_0
programs α , β and set of atomic programs Π_0
set of formulas Φ and set of programs Π are the smallest sets such that:

- $\Phi_0 \subseteq \Phi$
- $\Pi_0 \subseteq \Pi$
- if $\varphi, \psi \in \Phi$, then $\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi, \varphi \rightarrow \psi \in \Phi$
- if $\alpha, \beta \in \Pi$, then $\alpha \cup \beta, \alpha; \beta, \alpha^*, \in \Pi$
- if $\alpha \in \Pi$ and $\varphi \in \Phi$, then $[\alpha]\varphi \in \Phi$
- if $\varphi \in \Phi$, then $\varphi? \in \Pi$

Propositional Dynamic Logic

Syntax

for formulas φ , ψ and set of atomic formulas Φ_0
programs α , β and set of atomic programs Π_0
set of formulas Φ and set of programs Π are the smallest sets such that:

- $\Phi_0 \subseteq \Phi$
- $\Pi_0 \subseteq \Pi$
- if $\varphi, \psi \in \Phi$, then $\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi, \varphi \rightarrow \psi \in \Phi$
- if $\alpha, \beta \in \Pi$, then $\alpha \cup \beta, \alpha; \beta, \alpha^*, \in \Pi$
- if $\alpha \in \Pi$ and $\varphi \in \Phi$, then $[\alpha]\varphi \in \Phi$
- if $\varphi \in \Phi$, then $\varphi? \in \Pi$

Propositional Dynamic Logic

Syntax

for formulas φ , ψ and set of atomic formulas Φ_0
programs α , β and set of atomic programs Π_0
set of formulas Φ and set of programs Π are the smallest sets such that:

- $\Phi_0 \subseteq \Phi$
- $\Pi_0 \subseteq \Pi$
- if $\varphi, \psi \in \Phi$, then $\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi, \varphi \rightarrow \psi \in \Phi$
- if $\alpha, \beta \in \Pi$, then $\alpha \cup \beta, \alpha; \beta, \alpha^*, \in \Pi$
- if $\alpha \in \Pi$ and $\varphi \in \Phi$, then $[\alpha]\varphi \in \Phi$
- if $\varphi \in \Phi$, then $\varphi? \in \Pi$

Propositional Dynamic Logic

Syntax

for formulas φ , ψ and set of atomic formulas Φ_0
programs α , β and set of atomic programs Π_0
set of formulas Φ and set of programs Π are the smallest sets such that:

- $\Phi_0 \subseteq \Phi$
- $\Pi_0 \subseteq \Pi$
- if $\varphi, \psi \in \Phi$, then $\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi, \varphi \to \psi \in \Phi$
- if $\alpha, \beta \in \Pi$, then $\alpha \cup \beta, \alpha; \beta, \alpha^*, \in \Pi$
- if $\alpha \in \Pi$ and $\varphi \in \Phi$, then $[\alpha]\varphi \in \Phi$
- if $\varphi \in \Phi$, then $\varphi? \in \Pi$

Propositional Dynamic Logic

Remarks

Set of formulas Φ and set of programs Π
cannot be defined separetely, because:

- $[\alpha]\varphi \in \Phi$
- $\varphi? \in \Pi$

Propositional Dynamic Logic

Remarks

Set of formulas Φ and set of programs Π
cannot be defined separetely, because:

- $[\alpha]\varphi \in \Phi$
- $\varphi? \in \Pi$

Propositional Dynamic Logic

Remarks

Set of formulas Φ and set of programs Π
cannot be defined separetely, because:

- $[\alpha]\varphi \in \Phi$
- $\varphi? \in \Pi$

Propositional Dynamic Logic

Remarks

Set of formulas Φ and set of programs Π
cannot be defined separetely, because:

- $[\alpha]\varphi \in \Phi$
- $\varphi? \in \Pi$

Propositional Dynamic Logic

Semantics

Kripke Model

$$\mathfrak{K} = (K, m_{\mathfrak{K}})$$

- K – not-empty set of states: k,u,v, ...
- $m_{\mathfrak{K}}$ – meaning function:
  - $m_{\mathfrak{K}}(\varphi) \subseteq K, \qquad \varphi \in \Phi$
  - $m_{\mathfrak{K}}(\alpha) \subseteq K \times K, \quad \alpha \in \Pi$

Propositional Dynamic Logic

Semantics

Kripke Model

$$\mathfrak{K} = (K, m_{\mathfrak{K}})$$

- K – not-empty set of states: k,u,v, ...
- $m_{\mathfrak{K}}$ – meaning function:
  - $m_{\mathfrak{K}}(\varphi) \subseteq K, \qquad \varphi \in \Phi$
  - $m_{\mathfrak{K}}(\alpha) \subseteq K \times K, \quad \alpha \in \Pi$

Propositional Dynamic Logic

Semantics

Kripke Model

$$\mathfrak{K} = (K, m_{\mathfrak{K}})$$

- K – not-empty set of states: k,u,v, ...
- $m_{\mathfrak{K}}$ – meaning function:
  - $m_{\mathfrak{K}}(\varphi) \subseteq K, \quad \varphi \in \Phi$
  - $m_{\mathfrak{K}}(\alpha) \subseteq K \times K, \quad \alpha \in \Pi$

Propositional Dynamic Logic

Semantics

Kripke Model

$$\mathfrak{K} = (K, m_{\mathfrak{K}})$$

- K – not-empty set of states: k,u,v, ...
- $m_{\mathfrak{K}}$ – meaning function:
  - $m_{\mathfrak{K}}(\varphi) \subseteq K, \quad \varphi \in \Phi$
  - $m_{\mathfrak{K}}(\alpha) \subseteq K \times K, \quad \alpha \in \Pi$

Propositional Dynamic Logic

Semantics

Kripke Model

$$\mathfrak{K} = (K, m_{\mathfrak{K}})$$

- K – not-empty set of states: k,u,v, ...
- $m_{\mathfrak{K}}$ – meaning function:
  - $m_{\mathfrak{K}}(\varphi) \subseteq K, \qquad \varphi \in \Phi$
  - $m_{\mathfrak{K}}(\alpha) \subseteq K \times K, \quad \alpha \in \Pi$

Propositional Dynamic Logic

Semantics

Kripke Model

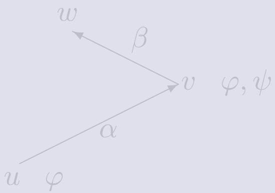
$$\mathfrak{K} = (K, m_{\mathfrak{K}})$$

- K – not-empty set of states: k,u,v, . . .
- $m_{\mathfrak{K}}$ – meaning function:
  - $m_{\mathfrak{K}}(\varphi) \subseteq K, \qquad \varphi \in \Phi$
  - $m_{\mathfrak{K}}(\alpha) \subseteq K \times K, \quad \alpha \in \Pi$

Propositional Dynamic Logic

Model example $\mathfrak{K} = (K, m_{\mathfrak{K}})$

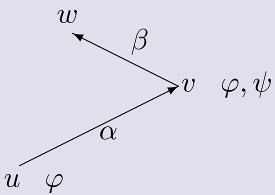
- $K = \{u, v, w\}$
- $m_{\mathfrak{K}}(\varphi) = \{u, v\}$
- $m_{\mathfrak{K}}(\psi) = \{v\}$
- $m_{\mathfrak{K}}(\alpha) = \{(u, v)\}$
- $m_{\mathfrak{K}}(\beta) = \{(v, w)\}$



Propositional Dynamic Logic

Model example $\mathfrak{K} = (K, m_{\mathfrak{K}})$

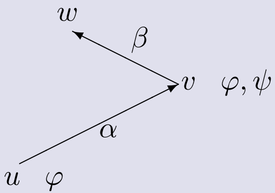
- $K = \{u, v, w\}$
- $m_{\mathfrak{K}}(\varphi) = \{u, v\}$
- $m_{\mathfrak{K}}(\psi) = \{v\}$
- $m_{\mathfrak{K}}(\alpha) = \{(u, v)\}$
- $m_{\mathfrak{K}}(\beta) = \{(v, w)\}$



Propositional Dynamic Logic

Model example $\mathfrak{K} = (K, m_{\mathfrak{K}})$

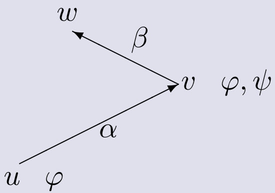
- $K = \{u, v, w\}$
- $m_{\mathfrak{K}}(\varphi) = \{u, v\}$
- $m_{\mathfrak{K}}(\psi) = \{v\}$
- $m_{\mathfrak{K}}(\alpha) = \{(u, v)\}$
- $m_{\mathfrak{K}}(\beta) = \{(v, w)\}$



Propositional Dynamic Logic

Model example $\mathfrak{K} = (K, m_{\mathfrak{K}})$

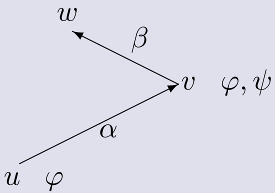
- $K = \{u, v, w\}$
- $m_{\mathfrak{K}}(\varphi) = \{u, v\}$
- $m_{\mathfrak{K}}(\psi) = \{v\}$
- $m_{\mathfrak{K}}(\alpha) = \{(u, v)\}$
- $m_{\mathfrak{K}}(\beta) = \{(v, w)\}$



Propositional Dynamic Logic

Model example $\mathfrak{K} = (K, m_{\mathfrak{K}})$

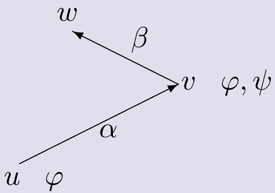
- $K = \{u, v, w\}$
- $m_{\mathfrak{K}}(\varphi) = \{u, v\}$
- $m_{\mathfrak{K}}(\psi) = \{v\}$
- $m_{\mathfrak{K}}(\alpha) = \{(u, v)\}$
- $m_{\mathfrak{K}}(\beta) = \{(v, w)\}$



Propositional Dynamic Logic

Model example $\mathfrak{K} = (K, m_{\mathfrak{K}})$

- $K = \{u, v, w\}$
- $m_{\mathfrak{K}}(\varphi) = \{u, v\}$
- $m_{\mathfrak{K}}(\psi) = \{v\}$
- $m_{\mathfrak{K}}(\alpha) = \{(u, v)\}$
- $m_{\mathfrak{K}}(\beta) = \{(v, w)\}$



Propositional Dynamic Logic

Programs construction

$$\textbf{skip} \stackrel{df}{=} 1?$$
$$\textbf{fail} \stackrel{df}{=} 0?$$
$$\textbf{if } \varphi \textbf{ than } \alpha \textbf{ else } \beta \stackrel{df}{=} (\varphi?;\alpha) \cup (\neg\varphi?;\beta)$$
$$\textbf{while } \varphi \textbf{ do } \alpha \stackrel{df}{=} (\varphi?;\alpha)^*;\neg\varphi?$$
$$\textbf{repeat } \alpha \textbf{ until } \varphi \stackrel{df}{=} \alpha;(\neg\varphi?;\alpha)^*;\varphi?$$

Propositional Dynamic Logic

Programs construction

$$\textbf{skip} \stackrel{\text{df}}{=} \mathbf{1}?$$

$$\textbf{fail} \stackrel{\text{df}}{=} \mathbf{0}?$$

$$\textbf{if } \varphi \textbf{ than } \alpha \textbf{ else } \beta \stackrel{\text{df}}{=} (\varphi?;\alpha) \cup (\neg\varphi?;\beta)$$

$$\textbf{while } \varphi \textbf{ do } \alpha \stackrel{\text{df}}{=} (\varphi?;\alpha)^*;\neg\varphi?$$

$$\textbf{repeat } \alpha \textbf{ until } \varphi \stackrel{\text{df}}{=} \alpha;(\neg\varphi?;\alpha)^*;\varphi?$$

Propositional Dynamic Logic

Programs construction

$$\begin{aligned}
\textbf{skip} &\stackrel{\text{df}}{=} \mathbf{1}? \\
\textbf{fail} &\stackrel{\text{df}}{=} \mathbf{0}? \\
\textbf{if } \varphi \textbf{ than } \alpha \textbf{ else } \beta &\stackrel{\text{df}}{=} (\varphi?;\alpha) \cup (\neg\varphi?;\beta) \\
\textbf{while } \varphi \textbf{ do } \alpha &\stackrel{\text{df}}{=} (\varphi?;\alpha)^*;\neg\varphi? \\
\textbf{repeat } \alpha \textbf{ until } \varphi &\stackrel{\text{df}}{=} \alpha;(\neg\varphi?;\alpha)^*;\varphi?
\end{aligned}$$

Propositional Dynamic Logic

Programs construction

$$\begin{aligned}
\textbf{skip} &\overset{\text{df}}{=} \mathbf{1}? \\
\textbf{fail} &\overset{\text{df}}{=} \mathbf{0}? \\
\textbf{if } \varphi \textbf{ than } \alpha \textbf{ else } \beta &\overset{\text{df}}{=} (\varphi?;\alpha) \cup (\neg\varphi?;\beta) \\
\textbf{while } \varphi \textbf{ do } \alpha &\overset{\text{df}}{=} (\varphi?;\alpha)^*;\neg\varphi? \\
\textbf{repeat } \alpha \textbf{ until } \varphi &\overset{\text{df}}{=} \alpha;(\neg\varphi?;\alpha)^*;\varphi?
\end{aligned}$$

Propositional Dynamic Logic

Programs construction

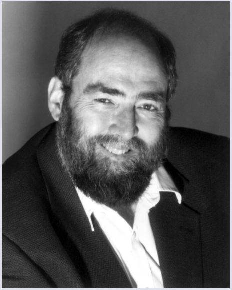
$$\begin{aligned} \textbf{skip} &\stackrel{\text{df}}{=} \mathbf{1}? \\ \textbf{fail} &\stackrel{\text{df}}{=} \mathbf{0}? \\ \textbf{if } \varphi \textbf{ than } \alpha \textbf{ else } \beta &\stackrel{\text{df}}{=} (\varphi?;\alpha) \cup (\neg\varphi?;\beta) \\ \textbf{while } \varphi \textbf{ do } \alpha &\stackrel{\text{df}}{=} (\varphi?;\alpha)^*;\neg\varphi? \\ \textbf{repeat } \alpha \textbf{ until } \varphi &\stackrel{\text{df}}{=} \alpha;(\neg\varphi?;\alpha)^*;\varphi? \end{aligned}$$

Propositional Dynamic Logic

Programs construction

$$
\begin{aligned}
\textbf{skip} &\overset{\text{df}}{=} \mathbf{1}? \\
\textbf{fail} &\overset{\text{df}}{=} \mathbf{0}? \\
\textbf{if } \varphi \textbf{ than } \alpha \textbf{ else } \beta &\overset{\text{df}}{=} (\varphi?;\alpha) \cup (\neg\varphi?;\beta) \\
\textbf{while } \varphi \textbf{ do } \alpha &\overset{\text{df}}{=} (\varphi?;\alpha)^*;\neg\varphi? \\
\textbf{repeat } \alpha \textbf{ until } \varphi &\overset{\text{df}}{=} \alpha;(\neg\varphi?;\alpha)^*;\varphi?
\end{aligned}
$$

Qualitative approach



,,People who have never heard of differential equations successfully reason about the common sense world of quantities, motion, space, and time."

Kenneth D. Forbus

Qualitative approach

Qualitative approach

- qualitative representation,
- qualitative reasoning.

Qualitative approach

Qualitative approach

- qualitative representation,
- qualitative reasoning.

Qualitative approach

Qualitative approach

- qualitative representation,
- qualitative reasoning.

Logic

Qualitative reasoning method PDL_M^F:

MUNOZ-VELASCO, E,; BURRIEZA, A.; OJEDA-ACIEGO, M.:
A logic framework for reasoning with movement based on fuzzy qualitative representation.
Fuzzy Sets and Systems.

Logic

Qualitative reasoning method PDL_M^F:

MUNOZ-VELASCO, E,; BURRIEZA, A.; OJEDA-ACIEGO, M.:
A logic framework for reasoning with movement based on fuzzy qualitative representation.
Fuzzy Sets and Systems.

Idea

Movement representation

Movement is represented by a tuple L:

$$L = L_1 \times L_2 \times L_3 \times L_4 \times L_5 \times L_6 \times L_7$$

- $L_1 = A \times A$, for $A = \{A_1, \ldots, A_k\}$, $k \in \mathcal{N}$,
- $L_2 = 2^{\{v_0, v_1, v_2, v_3\}} \setminus \varnothing$,
- $L_3 = 2^{\{o_0, o_1, o_2, o_3, o_4\}} \setminus \varnothing$,
- $L_4 = (2^{\{0, -, +\}} \setminus \varnothing) \times (2^{\{0, -, +\}} \setminus \varnothing)$,
- $L_5 = 2^{\{o_0, o_1, o_2, o_3, o_4\}} \setminus \varnothing$,
- $L_6 = (2^{\{o_1, o_2\}} \setminus \varnothing) \times (2^{\{d_0, d_1, d_2, d_3\}} \setminus \varnothing)$,
- $L_7 = (2^{\{o_3, o_4\}} \setminus \varnothing) \times (2^{\{d_0, d_1, d_2, d_3\}} \setminus \varnothing)$.

Idea

Movement representation

Movement is represented by a tuple L:

$$L = L_1 \times L_2 \times L_3 \times L_4 \times L_5 \times L_6 \times L_7$$

- $L_1 = A \times A$, for $A = \{A_1, \ldots, A_k\}$, $k \in \mathcal{N}$,
- $L_2 = 2^{\{v_0, v_1, v_2, v_3\}} \setminus \varnothing$,
- $L_3 = 2^{\{o_0, o_1, o_2, o_3, o_4\}} \setminus \varnothing$,
- $L_4 = (2^{\{0, -, +\}} \setminus \varnothing) \times (2^{\{0, -, +\}} \setminus \varnothing)$,
- $L_5 = 2^{\{o_0, o_1, o_2, o_3, o_4\}} \setminus \varnothing$,
- $L_6 = (2^{\{o_1, o_2\}} \setminus \varnothing) \times (2^{\{d_0, d_1, d_2, d_3\}} \setminus \varnothing)$,
- $L_7 = (2^{\{o_3, o_4\}} \setminus \varnothing) \times (2^{\{d_0, d_1, d_2, d_3\}} \setminus \varnothing)$.

Idea

Movement representation

Movement is represented by a tuple L:

$$L = L_1 \times L_2 \times L_3 \times L_4 \times L_5 \times L_6 \times L_7$$

- $L_1 = A \times A$, for $A = \{A_1, \ldots, A_k\}$, $k \in \mathcal{N}$,
- $L_2 = 2^{\{v_0, v_1, v_2, v_3\}} \setminus \varnothing$,
- $L_3 = 2^{\{o_0, o_1, o_2, o_3, o_4\}} \setminus \varnothing$,
- $L_4 = (2^{\{0, -, +\}} \setminus \varnothing) \times (2^{\{0, -, +\}} \setminus \varnothing)$,
- $L_5 = 2^{\{o_0, o_1, o_2, o_3, o_4\}} \setminus \varnothing$,
- $L_6 = (2^{\{o_1, o_2\}} \setminus \varnothing) \times (2^{\{d_0, d_1, d_2, d_3\}} \setminus \varnothing)$,
- $L_7 = (2^{\{o_3, o_4\}} \setminus \varnothing) \times (2^{\{d_0, d_1, d_2, d_3\}} \setminus \varnothing)$.

Idea

Movement representation

Movement is represented by a tuple L:

$$L = L_1 \times L_2 \times L_3 \times L_4 \times L_5 \times L_6 \times L_7$$

- $L_1 = A \times A$, for $A = \{A_1, \ldots, A_k\}$, $k \in \mathcal{N}$,
- $L_2 = 2^{\{v_0, v_1, v_2, v_3\}} \setminus \varnothing$,
- $L_3 = 2^{\{o_0, o_1, o_2, o_3, o_4\}} \setminus \varnothing$,
- $L_4 = (2^{\{0,-,+\}} \setminus \varnothing) \times (2^{\{0,-,+\}} \setminus \varnothing)$,
- $L_5 = 2^{\{o_0, o_1, o_2, o_3, o_4\}} \setminus \varnothing$,
- $L_6 = (2^{\{o_1, o_2\}} \setminus \varnothing) \times (2^{\{d_0, d_1, d_2, d_3\}} \setminus \varnothing)$,
- $L_7 = (2^{\{o_3, o_4\}} \setminus \varnothing) \times (2^{\{d_0, d_1, d_2, d_3\}} \setminus \varnothing)$.

Idea

Movement representation

Movement is represented by a tuple L:

$$L = L_1 \times L_2 \times L_3 \times L_4 \times L_5 \times L_6 \times L_7$$

- $L_1 = A \times A$, for $A = \{A_1, \ldots, A_k\}$, $k \in \mathcal{N}$,
- $L_2 = 2^{\{v_0, v_1, v_2, v_3\}} \setminus \varnothing$,
- $L_3 = 2^{\{o_0, o_1, o_2, o_3, o_4\}} \setminus \varnothing$,
- $L_4 = (2^{\{0, -, +\}} \setminus \varnothing) \times (2^{\{0, -, +\}} \setminus \varnothing)$,
- $L_5 = 2^{\{o_0, o_1, o_2, o_3, o_4\}} \setminus \varnothing$,
- $L_6 = (2^{\{o_1, o_2\}} \setminus \varnothing) \times (2^{\{d_0, d_1, d_2, d_3\}} \setminus \varnothing)$,
- $L_7 = (2^{\{o_3, o_4\}} \setminus \varnothing) \times (2^{\{d_0, d_1, d_2, d_3\}} \setminus \varnothing)$.

Idea

Movement representation

Movement is represented by a tuple L:

$$L = L_1 \times L_2 \times L_3 \times L_4 \times L_5 \times L_6 \times L_7$$

- $L_1 = A \times A$, for $A = \{A_1, \ldots, A_k\}$, $k \in \mathcal{N}$,
- $L_2 = 2^{\{v_0, v_1, v_2, v_3\}} \setminus \varnothing$,
- $L_3 = 2^{\{o_0, o_1, o_2, o_3, o_4\}} \setminus \varnothing$,
- $L_4 = (2^{\{0, -, +\}} \setminus \varnothing) \times (2^{\{0, -, +\}} \setminus \varnothing)$,
- $L_5 = 2^{\{o_0, o_1, o_2, o_3, o_4\}} \setminus \varnothing$,
- $L_6 = (2^{\{o_1, o_2\}} \setminus \varnothing) \times (2^{\{d_0, d_1, d_2, d_3\}} \setminus \varnothing)$,
- $L_7 = (2^{\{o_3, o_4\}} \setminus \varnothing) \times (2^{\{d_0, d_1, d_2, d_3\}} \setminus \varnothing)$.

Idea

Movement representation

Movement is represented by a tuple L:

$$L = L_1 \times L_2 \times L_3 \times L_4 \times L_5 \times L_6 \times L_7$$

- $L_1 = A \times A$, for $A = \{A_1, \ldots, A_k\}$, $k \in \mathcal{N}$,
- $L_2 = 2^{\{v_0, v_1, v_2, v_3\}} \setminus \varnothing$,
- $L_3 = 2^{\{o_0, o_1, o_2, o_3, o_4\}} \setminus \varnothing$,
- $L_4 = (2^{\{0,-,+\}} \setminus \varnothing) \times (2^{\{0,-,+\}} \setminus \varnothing)$,
- $L_5 = 2^{\{o_0, o_1, o_2, o_3, o_4\}} \setminus \varnothing$,
- $L_6 = (2^{\{o_1, o_2\}} \setminus \varnothing) \times (2^{\{d_0, d_1, d_2, d_3\}} \setminus \varnothing)$,
- $L_7 = (2^{\{o_3, o_4\}} \setminus \varnothing) \times (2^{\{d_0, d_1, d_2, d_3\}} \setminus \varnothing)$.

Idea

Movement representation

Movement is represented by a tuple L:

$$L = L_1 \times L_2 \times L_3 \times L_4 \times L_5 \times L_6 \times L_7$$

- $L_1 = A \times A$, for $A = \{A_1, \ldots, A_k\}$, $k \in \mathcal{N}$,
- $L_2 = 2^{\{v_0, v_1, v_2, v_3\}} \setminus \varnothing$,
- $L_3 = 2^{\{o_0, o_1, o_2, o_3, o_4\}} \setminus \varnothing$,
- $L_4 = (2^{\{0, -, +\}} \setminus \varnothing) \times (2^{\{0, -, +\}} \setminus \varnothing)$,
- $L_5 = 2^{\{o_0, o_1, o_2, o_3, o_4\}} \setminus \varnothing$,
- $L_6 = (2^{\{o_1, o_2\}} \setminus \varnothing) \times (2^{\{d_0, d_1, d_2, d_3\}} \setminus \varnothing)$,
- $L_7 = (2^{\{o_3, o_4\}} \setminus \varnothing) \times (2^{\{d_0, d_1, d_2, d_3\}} \setminus \varnothing)$.

Idea

Example

$$(A_i, A_j; v_2v_3; o_3; +, -; o_1o_2o_3; o_1, d_1d_2; o_3, d_0)$$

Idea

Composition tables

A_iA_j \ A_jA_k	$*0$	$*-$	$*+$
$0*$	00	$0\pm$	$0\pm$
$-*$	-0	$-\pm$	$-\pm$
$+*$	-0	$+\pm$	$+\pm$

A_iA_j \ A_jA_k	o_rd_0	o_rd_u	o_td_u
o_rd_0	o_rd_0	o_rd_u	o_td_u
o_rd_s	o_rd_s	$o_r(d_s+d_u)$	$\begin{cases} o_t(d_u-d_s) & \text{if } s \leq u \\ o_r(d_s-d_u) & \text{if } s \geq u \end{cases}$

Idea

Composition tables

A_iA_j \ A_jA_k	$*0$	$*-$	$*+$
$0*$	00	$0\pm$	$0\pm$
$-*$	-0	$-\pm$	$-\pm$
$+*$	-0	$+\pm$	$+\pm$

A_iA_j \ A_jA_k	o_rd_0	o_rd_u	o_td_u
o_rd_0	o_rd_0	o_rd_u	o_td_u
o_rd_s	o_rd_s	$o_r(d_s + d_u)$	$\begin{cases} o_t(d_u - d_s) & \text{if } s \leq u \\ o_r(d_s - d_u) & \text{if } s \geq u \end{cases}$

Syntax

Syntax

- $\Phi_0 = \mathbb{V} \cup L$,
  - $\mathbb{V}$ – set of propositional variables,
  - $L = L_1 \times L_2 \times L_3 \times L_4 \times L_5 \times L_6 \times L_7$,
- Φ defined as in PDL,
- $\Pi_0 = \{rev_x | x \in L\} \cup \{\otimes_{x,y} | x, y \in L\} \cup \{Dec_x^s, Man_x^s, Inc_x^s$ $| s \in \{0, 1, 2, 3, 4\}, x \in L\}$,
  - rev_x – movement representation of A_j relative to A_i if x represents move of A_i relative to A_j,
  - $\otimes_{x,y}$ – composition of x and y,
  - $Dec_x^s, Man_x^s, Inc_x^s$ – decreasing, maintaining, increasing velocity x in direction s,
- Π defined as in PDL.

Syntax

Syntax

- $\Phi_0 = \mathbb{V} \cup L$,
  - $\mathbb{V}$ – set of propositional variables,
  - $L = L_1 \times L_2 \times L_3 \times L_4 \times L_5 \times L_6 \times L_7$,
- Φ defined as in PDL,
- $\Pi_0 = \{rev_x | x \in L\} \cup \{\otimes_{x,y} | x, y \in L\} \cup \{Dec_x^s, Man_x^s, Inc_x^s$ $| s \in \{0, 1, 2, 3, 4\}, x \in L\}$,
  - rev_x – movement representation of A_j relative to A_i if x represents move of A_i relative to A_j,
  - $\otimes_{x,y}$ – composition of x and y,
  - $Dec_x^s, Man_x^s, Inc_x^s$ – decreasing, maintaining, increasing velocity x in direction s,
- Π defined as in PDL.

Syntax

Syntax

- $\Phi_0 = \mathbb{V} \cup L$,
  - $\mathbb{V}$ – set of propositional variables,
  - $L = L_1 \times L_2 \times L_3 \times L_4 \times L_5 \times L_6 \times L_7$,
- Φ defined as in PDL,
- $\Pi_0 = \{rev_x | x \in L\} \cup \{\otimes_{x,y} | x, y \in L\} \cup \{Dec_x^s, Man_x^s, Inc_x^s$ $| s \in \{0, 1, 2, 3, 4\}, x \in L\}$,
  - rev_x – movement representation of A_j relative to A_i if x represents move of A_i relative to A_j,
  - $\otimes_{x,y}$ – composition of x and y,
  - $Dec_x^s, Man_x^s, Inc_x^s$ – decreasing, maintaining, increasing velocity x in direction s,
- Π defined as in PDL.

Syntax

Syntax

- $\Phi_0 = \mathbb{V} \cup L$,
  - $\mathbb{V}$ – set of propositional variables,
  - $L = L_1 \times L_2 \times L_3 \times L_4 \times L_5 \times L_6 \times L_7$,
- Φ defined as in PDL,
- $\Pi_0 = \{rev_x | x \in L\} \cup \{\otimes_{x,y} | x, y \in L\} \cup \{Dec^s_x, Man^s_x, Inc^s_x$ $| s \in \{0, 1, 2, 3, 4\}, x \in L\}$,
  - rev_x – movement representation of A_j relative to A_i if x represents move of A_i relative to A_j,
  - $\otimes_{x,y}$ – composition of x and y,
  - $Dec^s_x, Man^s_x, Inc^s_x$ – decreasing, maintaining, increasing velocity x in direction s,
- Π defined as in PDL.

Syntax

Syntax

- $\Phi_0 = \mathbb{V} \cup L$,
  - $\mathbb{V}$ – set of propositional variables,
  - $L = L_1 \times L_2 \times L_3 \times L_4 \times L_5 \times L_6 \times L_7$,
- Φ defined as in PDL,
- $\Pi_0 = \{rev_x | x \in L\} \cup \{\otimes_{x,y} | x, y \in L\} \cup \{Dec_x^s, Man_x^s, Inc_x^s$ $| s \in \{0, 1, 2, 3, 4\}, x \in L\}$,
  - rev_x – movement representation of A_j relative to A_i if x represents move of A_i relative to A_j,
  - $\otimes_{x,y}$ – composition of x and y,
  - $Dec_x^s, Man_x^s, Inc_x^s$ – decreasing, maintaining, increasing velocity x in direction s,
- Π defined as in PDL.

Syntax

Syntax

- $\Phi_0 = \mathbb{V} \cup L$,
  - $\mathbb{V}$ – set of propositional variables,
  - $L = L_1 \times L_2 \times L_3 \times L_4 \times L_5 \times L_6 \times L_7$,
- Φ defined as in PDL,
- $\Pi_0 = \{rev_x | x \in L\} \cup \{\otimes_{x,y} | x, y \in L\} \cup \{Dec_x^s, Man_x^s, Inc_x^s$ $| s \in \{0, 1, 2, 3, 4\}, x \in L\}$,
  - rev_x – movement representation of A_j relative to A_i if x represents move of A_i relative to A_j,
  - $\otimes_{x,y}$ – composition of x and y,
  - $Dec_x^s, Man_x^s, Inc_x^s$ – decreasing, maintaining, increasing velocity x in direction s,
- Π defined as in PDL.

Syntax

Syntax

- $\Phi_0 = \mathbb{V} \cup L$,
  - $\mathbb{V}$ – set of propositional variables,
  - $L = L_1 \times L_2 \times L_3 \times L_4 \times L_5 \times L_6 \times L_7$,
- Φ defined as in PDL,
- $\Pi_0 = \{rev_x | x \in L\} \cup \{\otimes_{x,y} | x, y \in L\} \cup \{Dec^s_x, Man^s_x, Inc^s_x \, | s \in \{0, 1, 2, 3, 4\}, x \in L\}$,
  - rev_x – movement representation of A_j relative to A_i if x represents move of A_i relative to A_j,
  - $\otimes_{x,y}$ – composition of x and y,
  - $Dec^s_x, Man^s_x, Inc^s_x$ – decreasing, maintaining, increasing velocity x in direction s,
- Π defined as in PDL.

Syntax

Syntax

- $\Phi_0 = \mathbb{V} \cup L$,
  - $\mathbb{V}$ – set of propositional variables,
  - $L = L_1 \times L_2 \times L_3 \times L_4 \times L_5 \times L_6 \times L_7$,
- Φ defined as in PDL,
- $\Pi_0 = \{rev_x | x \in L\} \cup \{\otimes_{x,y} | x, y \in L\} \cup \{Dec^s_x, Man^s_x, Inc^s_x$ $| s \in \{0, 1, 2, 3, 4\}, x \in L\}$,
  - rev_x – movement representation of A_j relative to A_i if x represents move of A_i relative to A_j,
  - $\otimes_{x,y}$ – composition of x and y,
  - $Dec^s_x, Man^s_x, Inc^s_x$ – decreasing, maintaining, increasing velocity x in direction s,
- Π defined as in PDL.

Syntax

Syntax

- $\Phi_0 = \mathbb{V} \cup L$,
  - $\mathbb{V}$ – set of propositional variables,
  - $L = L_1 \times L_2 \times L_3 \times L_4 \times L_5 \times L_6 \times L_7$,
- Φ defined as in PDL,
- $\Pi_0 = \{rev_x | x \in L\} \cup \{\otimes_{x,y} | x, y \in L\} \cup \{Dec_x^s, Man_x^s, Inc_x^s$ $| s \in \{0, 1, 2, 3, 4\}, x \in L\}$,
  - rev_x – movement representation of A_j relative to A_i if x represents move of A_i relative to A_j,
  - $\otimes_{x,y}$ – composition of x and y,
  - $Dec_x^s, Man_x^s, Inc_x^s$ – decreasing, maintaining, increasing velocity x in direction s,
- Π defined as in PDL.

Syntax

Syntax

- $\Phi_0 = \mathbb{V} \cup L$,
  - $\mathbb{V}$ – set of propositional variables,
  - $L = L_1 \times L_2 \times L_3 \times L_4 \times L_5 \times L_6 \times L_7$,
- Φ defined as in PDL,
- $\Pi_0 = \{rev_x | x \in L\} \cup \{\otimes_{x,y} | x, y \in L\} \cup \{Dec_x^s, Man_x^s, Inc_x^s$ $| s \in \{0, 1, 2, 3, 4\}, x \in L\}$,
  - rev_x – movement representation of A_j relative to A_i if x represents move of A_i relative to A_j,
  - $\otimes_{x,y}$ – composition of x and y,
  - $Dec_x^s, Man_x^s, Inc_x^s$ – decreasing, maintaining, increasing velocity x in direction s,
- Π defined as in PDL.

Semantics

Model $\mathcal{M}$

$$\mathcal{M} = (W, m)$$

- W – not empty set of states denoted by tuples L,
- m – meaning function:
  - $m(\varphi) \subseteq W$
  - $m(\alpha) \subseteq W \times W$

Semantics

Model $\mathcal{M}$

$$\mathcal{M} = (W, m)$$

- W – not empty set of states denoted by tuples L,
- m – meaning function:
  - $m(\varphi) \subseteq W$
  - $m(\alpha) \subseteq W \times W$

Semantics

Model $\mathcal{M}$

$$\mathcal{M} = (W, m)$$

- W – not empty set of states denoted by tuples L,
- m – meaning function:
  - $m(\varphi) \subseteq W$
  - $m(\alpha) \subseteq W \times W$

Semantics

Model $\mathcal{M}$

$$\mathcal{M} = (W, m)$$

- W – not empty set of states denoted by tuples L,
- m – meaning function:
  - $m(\varphi) \subseteq W$
  - $m(\alpha) \subseteq W \times W$

Semantics

Model $\mathcal{M}$

$$\mathcal{M} = (W, m)$$

- W – not empty set of states denoted by tuples L,
- m – meaning function:
  - $m(\varphi) \subseteq W$
  - $m(\alpha) \subseteq W \times W$

Semantics

Model $\mathcal{M}$

$$\mathcal{M} = (W, m)$$

- W – not empty set of states denoted by tuples L,
- m – meaning function:
  - $m(\varphi) \subseteq W$
  - $m(\alpha) \subseteq W \times W$

Properies

PDL_M^F logic is:

- sound,
- complete,
- decidable.

Properies

PDL_M^F logic is:

- sound,
- complete,
- decidable.

Properies

PDL_M^F logic is:

- sound,
- complete,
- decidable.

Properies

PDL^F_M logic is:

- sound,
- complete,
- decidable.

Application

Scenario

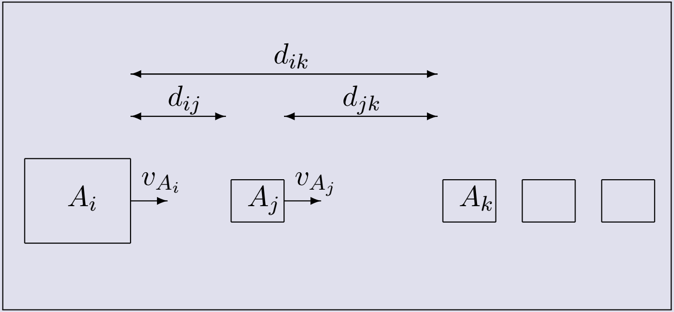


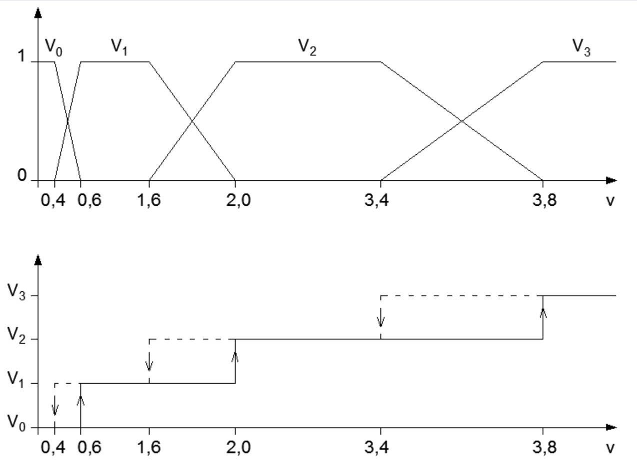
Figure : The scenario setup.

Application

Scenario

Application

Qualitativness



Composition table

A_iA_j \ A_jA_k	o_rd_0	o_rd_u	o_td_u
o_rd_0	o_rd_0	o_rd_u	o_td_u
o_rd_s	o_rd_s	$o_r(d_s + d_u)$	$\begin{cases} o_t(d_u - d_s) & \text{if } s \leq u \\ o_r(d_s - d_u) & \text{if } s \geq u \end{cases}$

A_iA_j \ A_jA_k	o_rd_0	o_rd_u
o_rd_0	o_rd_0	o_rd_u
o_rd_s	o_rd_s	$o_r(d_s + d_u) = o_rd_{max\{s,u\}}$

Composition table

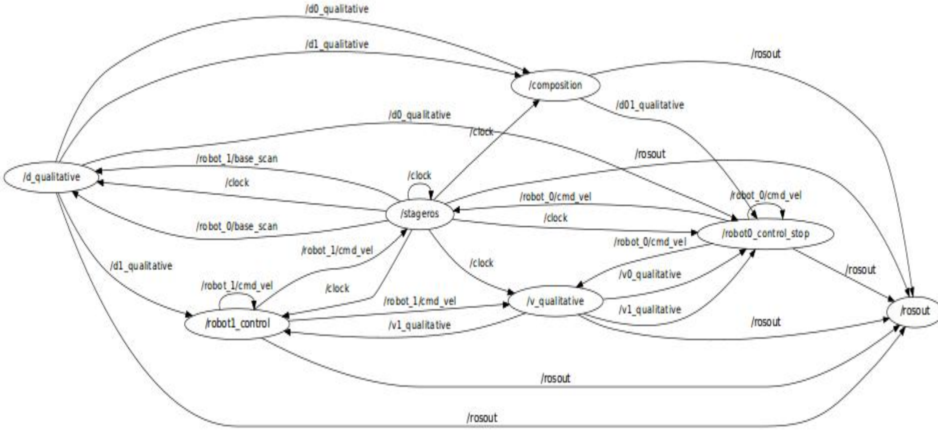
Table : All possible cases of adding qualitative distances for $d_s + d_u = d_{max\{s,u\}}$.

$+$	d_0	d_1	d_2	d_3
d_0	d_0	d_1	d_2	d_3
d_1	d_1	$\mathbf{d_1} d_2$	$\mathbf{d_2} d_3$	d_3
d_2	d_2	$\mathbf{d_2} d_3$	$\mathbf{d_2} d_3$	d_3
d_3	d_3	d_3	d_3	d_3

Velocity control

d \ dv	v_{-3}	v_{-2}	v_{-1}	v_0	v_1	v_2	v_3
d_0	Man_x^0	Man_x^0	$\boldsymbol{Dec_x^0}$	$\boldsymbol{Dec_x^0}$	$\boldsymbol{Dec_x^0}$	$\boldsymbol{Dec_x^0}$	$\boldsymbol{Dec_x^0}$
d_1	$\boldsymbol{Inc_x^0}$	Man_x^0	Man_x^0	$\boldsymbol{Dec_x^0}$	$\boldsymbol{Dec_x^0}$	$\boldsymbol{Dec_x^0}$	$\boldsymbol{Dec_x^0}$
d_2	$\boldsymbol{Inc_x^0}$	$\boldsymbol{Inc_x^0}$	Man_x^0	Man_x^0	Man_x^0	$\boldsymbol{Dec_x^0}$	$\boldsymbol{Dec_x^0}$
d_3	$\boldsymbol{Inc_x^0}$	$\boldsymbol{Inc_x^0}$	$\boldsymbol{Inc_x^0}$	$\boldsymbol{Inc_x^0}$	$\boldsymbol{Inc_x^0}$	$\boldsymbol{Inc_x^0}$	$\boldsymbol{Inc_x^0}$

Application



Simulation

Following

Simulation

Crash avoidance

Simulation

Crash avoidance

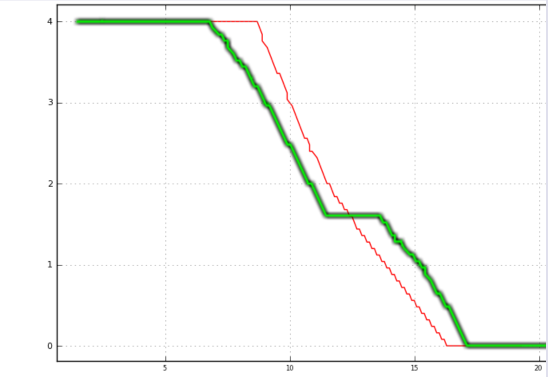


Figure : The A_i velocity change in the performed tests.

Summarize

Our application:

- solves the collision avoidance problem,
- shows the example of PDL_M^F framework usage,
- may be used in further, more complex PDL_M^F applications (in simulations or in real robots).

Summarize

Our application:

- solves the collision avoidance problem,
- shows the example of PDL_M^F framework usage,
- may be used in further, more complex PDL_M^F applications (in simulations or in real robots).

Summarize

Our application:

- solves the collision avoidance problem,
- shows the example of PDL^F_M framework usage,
- may be used in further, more complex PDL^F_M applications (in simulations or in real robots).

Summarize

Our application:

- solves the collision avoidance problem,
- shows the example of PDL_M^F framework usage,
- may be used in further, more complex PDL_M^F applications (in simulations or in real robots).

Thank you for your attention

przemek.walega@wp.pl

The project is partially supported by
the NCN grant DEC-2011/02/A/HS1/00395.