A Temporally Expressive Planner Based on ASP with Constraints

Preliminary Design

Forrest Sheng Bao
Sandeep Chintabathina
Ricardo Morales
Nelson Rushton
Richard Watson
Yuanlin Zhang

KR Lab, CS, Texas Tech

12/08/2010
A Running Example: The Two-faucet World

Figure: The "two-faucet" world

Bao et al. (KR Lab, CS, Texas Tech)
Introduction

- Temporally expressive planning requires concurrency among actions.
- Previous approaches cannot handle temporal expressiveness or cannot handle well.
- We try to translate temporally expressive planning problems in PDDL2.1 into \textit{ACC} programs and find the plan using \textit{ACC} solvers.
- Our approach can solve drawbacks of existing other approaches.
Introduction

An Abstract PDDL Syntax

- Building Blocks
- Continuous Durative Actions
- PDDL Planning Problems

Translation from PDDL to ACC

- Building Blocks
- Continuous Durative Actions
Signatures

- **constants** – $N$
  - $f1$: faucet 1
  - $f2$: faucet 2

- **fluents** – $P$
  - $opened(F)$: faucet $F$ is opened.
  - $stuck(F)$: faucet $F$ is stuck.

- **variables** – $X$
  - $Sink$: amount of water in the sink
  - $Tank(F)$: amount of water in tank above faucet $F$
Signatures

- **constants** – $N$
  - $f1$: faucet 1
  - $f2$: faucet 2

- **fluents** – $P$
  - $opened(F)$: faucet $F$ is opened.
  - $stuck(F)$: faucet $F$ is stuck.

- **variables** – $X$
  - $Sink$: amount of water in the sink
  - $Tank(F)$: amount of water in tank above faucet $F$
Signatures

- constants – $N$
  - $f1$: faucet 1
  - $f2$: faucet 2

- fluents – $P$
  - $opened(F)$: faucet $F$ is opened.
  - $stuck(F)$: faucet $F$ is stuck.

- variables – $X$
  - $Sink$: amount of water in the sink
  - $Tank(F)$: amount of water in tank above faucet $F$
1. Introduction

2. An Abstract PDDL Syntax
   - Building Blocks
   - Continuous Durative Actions
   - PDDL Planning Problems

3. Translation from PDDL to ACC
   - Building Blocks
   - Continuous Durative Actions
Continuous Durative Actions

Example (a durative action)

\[
\langle \text{open}(F), \\
\langle \text{?duration} \leq \text{Tank}(F)/\text{flowRate}(F) \rangle, \\
\langle \neg \text{opened}(F) \land \neg \text{stuck}(F), \\
\quad \text{Tank}(F) > 0, \\
\quad \neg \text{stuck}(F) \land \text{opened}(F) \\
\rangle, \\
\langle \text{opened}(F), \\
\quad \text{Sink increase flowRate}(F) \ast \Delta t \land \\
\quad \text{Tank}(F) \text{ decrease flowRate}(F) \ast \Delta t, \\
\quad \neg \text{opened}(F) \\
\rangle
\]
Continuous Durative Actions

Example (a durative action)

\[
\langle \text{open}(F), \\
\langle \text{duration} \leq \text{Tank}(F)/\text{flowRate}(F) \rangle, \\
\langle \neg \text{opened}(F) \land \neg \text{stuck}(F), \\
\text{Tank}(F) > 0, \\
\neg \text{stuck}(F) \land \text{opened}(F) \rangle, \\
\langle \text{opened}(F), \\
\text{Sink increase flowRate}(F) \ast \Delta t \land \\
\text{Tank}(F) \text{ decrease flowRate}(F) \ast \Delta t, \\
\neg \text{opened}(F) \rangle
\]

Bao et al. (KR Lab, CS, Texas Tech)
Continuous Durative Actions

Example (a durative action)

\[
\langle \text{open}(F), \\
\langle \text{?duration} \leq \text{Tank}(F)/\text{flowRate}(F) \rangle, \\
\langle \neg \text{opened}(F) \land \neg \text{stuck}(F), \\
\text{Tank}(F) > 0, \\
\neg \text{stuck}(F) \land \text{opened}(F) \rangle, \\
\langle \text{opened}(F), \\
\text{Sink increase flowRate}(F) \ast \Delta t \land \\
\text{Tank}(F) \text{ decrease flowRate}(F) \ast \Delta t, \\
\neg \text{opened}(F) \rangle
\]
Continuous Durative Actions

Example (a durative action)

\[
\langle \text{open}(F),
\langle \text{？duration} \leq \text{Tank}(F)/\text{flowRate}(F) \rangle,
\langle \neg \text{opened}(F) \land \neg \text{stuck}(F),
\text{Tank}(F) > 0,
\neg \text{stuck}(F) \land \text{opened}(F)
\rangle,
\langle \text{opened}(F),
\text{Sink increase flowRate}(F) \ast \Delta t \land
\text{Tank}(F) \text{ decrease flowRate}(F) \ast \Delta t,
\neg \text{opened}(F)
\rangle
\rangle
\]
Continuous Durative Actions

Example (a durative action)

\[ \langle open(F), \langle ?duration \leq \text{Tank}(F)/\text{flowRate}(F) \rangle, \langle \neg opened(F) \land \neg stuck(F), \text{Tank}(F) > 0, \neg stuck(F) \land opened(F) \rangle, \langle opened(F), \text{Sink increase flowRate}(F) \ast \Delta t \land \text{Tank}(F) \text{ decrease flowRate}(F) \ast \Delta t, \neg opened(F) \rangle \rangle \]
Continuous Durative Actions

Example (a durative action)

\[
\langle \text{open}(F),
\langle ?\text{duration} \leq \text{Tank}(F)/\text{flowRate}(F) \rangle, \\
\langle \neg \text{opened}(F) \land \neg \text{stuck}(F),
\text{Tank}(F) > 0, \\
\neg \text{stuck}(F) \land \text{opened}(F) \rangle,
\rangle, \\
\langle \text{opened}(F), \\
\text{Sink increase flowRate}(F) \ast \Delta t \land \\
\text{Tank}(F) \text{ decrease flowRate}(F) \ast \Delta t, \\
\neg \text{opened}(F) \rangle
\]
Continuous Durative Actions

Example (a durative action)

\[
\langle \text{open}(F), \\
\langle \text{duration} \leq \text{Tank}(F)/\text{flowRate}(F) \rangle, \\
\langle \neg \text{opened}(F) \wedge \neg \text{stuck}(F), \\
\text{Tank}(F) > 0, \\
\neg \text{stuck}(F) \wedge \text{open}(F) \rangle, \\
\langle \text{opened}(F), \\
\text{Sink increase flowRate}(F) \ast \Delta t \wedge \\
\text{Tank}(F) \text{ decrease flowRate}(F) \ast \Delta t, \\
\neg \text{opened}(F) \rangle \rangle
\]
Example (a durative action)

\[
\langle \text{open}(F), \quad \\
\langle \ ?\text{duration} \leq \text{Tank}(F)/\text{flowRate}(F) \rangle, \quad \\
\langle \neg \text{opened}(F) \land \neg \text{stuck}(F), \quad \\
\text{Tank}(F) > 0, \quad \\
\neg \text{stuck}(F) \land \text{opened}(F) \rangle, \\
\langle \text{opened}(F), \quad \\
\text{Sink increase} \text{ flowRate}(F) \ast \Delta t \land \\
\text{Tank}(F) \text{ decrease} \text{ flowRate}(F) \ast \Delta t, \quad \\
\neg \text{opened}(F) \rangle \rangle
\]
1 Introduction

2 An Abstract PDDL Syntax
   - Building Blocks
   - Continuous Durative Actions
   - PDDL Planning Problems

3 Translation from PDDL to ACC
   - Building Blocks
   - Continuous Durative Actions
PDDL Planning Problems

A PDDL planning problem consists of:

- A set of actions.
- An initial state, a conjunction of ground fluents and numeric constraints.

Example (An initial state)

\{Sink = 0, \text{Tank}(f_1, 0) = 5, \text{Tank}(f, 0) = 5, \neg \text{opened}(f_1), \neg \text{opened}(f_2), \neg \text{stuck}(f_1), \neg \text{stuck}(f_2), \text{flowRate}(f_1) = 2, \text{flowRate}(f_2) = 3\}\n
- A goal state, a conjunction of ground fluents and numeric constraints.

Example (A goal state)

\{Sink = 5\}
PDDL Planning Problems

A PDDL planning problem consists of:

- A set of actions.
- An initial state, a conjunction of ground fluents and numeric constraints.

Example (An initial state)
\{Sink = 0, \text{Tank}(f_1, 0) = 5, \text{Tank}(f, 0) = 5, \neg \text{opened}(f_1), \neg \text{opened}(f_2), \neg \text{stuck}(f_1), \neg \text{stuck}(f_2), \text{flowRate}(f_1) = 2, \text{flowRate}(f_2) = 3\}

- A goal state, a conjunction of ground fluents and numeric constraints.

Example (A goal state)
\{Sink = 5\}
PDDL Planning Problems

A PDDL planning problem consists of:

- A set of actions.
- An initial state, a conjunction of ground fluents and numeric constraints

Example (An initial state)

\{ Sink = 0, Tank(f_1, 0) = 5, Tank(f, 0) = 5, \neg opened(f_1), \neg opened(f_2), \neg stuck(f_1), \neg stuck(f_2), flowRate(f_1) = 2, flowRate(f_2) = 3 \} 

- A goal state, a conjunction of ground fluents and numeric constraints.

Example (A goal state)

\{ Sink = 5 \}
A **plan** is a set of timed pairs of either the form \((t, a[t'])\) or \((t, a)\) where \(t\) is a time, \(a\) an action name and \(t'\) a time specifying the duration of a durative action.

A valid plan for “two-faucet” example is \(\langle(0, open(f1)[1]), (0, open(f2)[1])\rangle\).
 Plans

- A plan is a set of timed pairs of either the form \((t, a[t'])\) or \((t, a)\) where \(t\) is a time, \(a\) an action name and \(t'\) a time specifying the duration of a durative action.

- A valid plan for “two-faucet” example is \(\langle (0, \text{open}(f1)[1]), (0, \text{open}(f2)[1]) \rangle\).
1 Introduction

2 An Abstract PDDL Syntax
   - Building Blocks
   - Continuous Durative Actions
   - PDDL Planning Problems

3 Translation from PDDL to ACC
   - Building Blocks
   - Continuous Durative Actions
Example (Predicates for the “two-faucet” world)

\[
\begin{align*}
\text{opened}(F) & \implies \text{opened}(F, S) \\
\text{stuck}(F) & \implies \text{stuck}(F, S) \\
\text{(introduced)} & \implies \text{occurs}(A, S)
\end{align*}
\]

More will be added later.
Numeric Variables and Functions

Example (Functions for the “two-faucet” world)

Sink \implies Sink(S) \in \mathcal{R}

Tank(F) \implies Tank(F, S) \in \mathcal{R}

(introduced) \implies at(S) \in \text{the domain of time}

More will be added later.
Initial States

Example (A set of facts representing an initial state)

\[ \neg \text{opened}(f_1, 0). \]
\[ \neg \text{opened}(f_2, 0). \]
\[ \neg \text{stuck}(f_1, 0). \]
\[ \neg \text{stuck}(f_2, 0). \]
\[ \text{Sink}(0) = 0. \]
\[ \text{Tank}(f_1, 0) = 5. \]
\[ \text{Tank}(f_2, 0) = 5. \]
Example (A set of rules representing a goal state)

\[
\text{goal} \leftarrow \text{Sink}(S) = 5.
\]

\[
\leftarrow \text{not goal}.
\]
1 Introduction

2 An Abstract PDDL Syntax
   - Building Blocks
   - Continuous Durative Actions
   - PDDL Planning Problems

3 Translation from PDDL to ACC
   - Building Blocks
   - Continuous Durative Actions
Divide and Conquer

Original action

\[
\langle open(F),
\langle \text{?duration} \leq \text{tank}(F)/\text{flowRate}(F) \rangle, \\
\langle \neg opened(F) \land \neg stuck(F), \\
\hspace{1cm} \text{Tank}(F) > 0, \\
\hspace{1.5cm} \neg stuck(F) \land opened(F) \\
\rangle, \\
\langle opened(F), \\
\hspace{0.5cm} \text{Sink increase} \text{ flowRate}(F) \ast \Delta t \land \\
\hspace{1cm} \text{Tank}(F) \text{ decrease} \text{ flowRate}(F) \ast \Delta t, \\
\hspace{1.5cm} \neg opened(F) \\
\rangle
\]

Now it becomes

A new action for “at start”:

\[
\langle open_s(F), \\
\neg opened(F) \land \neg stuck(F), \\
\neg stuck(F) \land opened(F) \\
\rangle
\]

and a new action for “at end”:

\[
\langle open_e(F), \\
\neg stuck(F) \land \neg opened(F) \\
\rangle
\]

Later we will discuss about invariant monitoring and continuous assignment.
Divide and Conquer

Original action

\[
\langle \text{open}(F), \\
\langle ?\text{duration} \leq \text{tank}(F) / \text{flowRate}(F) \rangle, \\
\langle \neg \text{opened}(F) \land \neg \text{stuck}(F), \\
\text{Tank}(F) > 0, \\
\neg \text{stuck}(F) \land \text{opened}(F) \rangle \\
\langle \text{opened}(F), \\
\text{Sink increase} \ \text{flowRate}(F) \ast \Delta t \land \\
\text{Tank}(F) \text{ decrease} \ \text{flowRate}(F) \ast \Delta t, \\
\neg \text{opened}(F) \rangle
\]

Now it becomes

A new action for “at start”:

\[
\langle \text{open}_s(F), \\
\neg \text{opened}(F) \land \neg \text{stuck}(F), \\
\text{opened}(F) \rangle
\]

and a new action for “at end”:

\[
\langle \text{open}_e(F), \\
\neg \text{stuck}(F) \land \text{opened}(F), \\
\neg \text{opened}(F) \rangle
\]

Later we will discuss about invariant monitoring and continuous assignment.
Divide and Conquer

Original action

\[
\langle \text{open}(F), \\
\langle \text{duration} \leq \text{tank}(F)/\text{flowRate}(F) \rangle, \\
\langle \neg \text{opened}(F) \land \neg \text{stuck}(F), \\
\text{Tank}(F) > 0, \\
\neg \text{stuck}(F) \land \text{opened}(F) \rangle, \\
\langle \text{opened}(F), \\
\text{Sink increase flowRate}(F) \ast \Delta t \land \\
\text{Tank}(F) \text{ decrease flowRate}(F) \ast \Delta t, \\
\neg \text{opened}(F) \rangle \rangle
\]

Now it becomes

A new action for “at start”:

\[
\langle \text{open}_s(F), \\
\neg \text{opened}(F) \land \neg \text{stuck}(F), \\
\text{opened}(F) \rangle
\]

and a new action for “at end”:

\[
\langle \text{open}_e(F), \\
\neg \text{stuck}(F) \land \text{opened}(F), \\
\neg \text{opened}(F) \rangle
\]

Later we will discuss about invariant monitoring and continuous assignment.
Divide and Conquer

Original action

\[
\langle \text{open}(F), \\
\langle \text{duration} \leq \text{tank}(F)/\text{flowRate}(F) \rangle, \\
\langle \neg\text{opened}(F) \land \neg\text{stuck}(F), \\
\text{Tank}(F) > 0, \\
\neg\text{stuck}(F) \land \text{opened}(F) \rangle, \\
\langle \text{opened}(F), \\
\text{Sink increase flowRate}(F) \times \Delta t \land \\
\text{Tank}(F) \text{ decrease flowRate}(F) \times \Delta t, \\
\neg\text{opened}(F) \rangle
\]

Now it becomes

A new action for “at start”:
\[
\langle \text{open}_s(F), \\
\neg\text{opened}(F) \land \neg\text{stuck}(F), \\
\text{opened}(F) \rangle
\]

and a new action for “at end”:
\[
\langle \text{open}_e(F), \\
\neg\text{stuck}(F) \land \text{opened}(F), \\
\neg\text{opened}(F) \rangle
\]

Later we will discuss about invariant monitoring and continuous assignment.
Example (Encoding preconditions for $open_s(F)$)

$\leftarrow open_s(F, S), \neg stuck(F, S)$.  
$\leftarrow open_s(F, S), \neg opened(F, S)$.  

Basic idea: If a precondition is not known as satisfied, then the action cannot execute.
Effects on Fluents of Simple Actions

Example (Encoding Effects on Fluents for $\text{open}_e(F)$)

\[
\begin{align*}
\neg \text{opened}(F, S + 1) & \iff \text{occurs}(\text{open}_e(F), S) \\
\text{opened}(F, S + 1) & \iff \text{opened}(F, S), \neg \neg \text{opened}(F, S + 1) \\
\neg \text{opened}(F, S + 1) & \iff \neg \text{opened}(F, S), \neg \text{opened}(F, S + 1)
\end{align*}
\]
Continuous Variable Assignment I

1. Introduce a new predicate \textit{action\_on}(A, S) to mark whether action \textit{S} is on at step \textit{S}.

2. We then introduce a function \textit{delta}.

Example (Contribution of each action to Sink(S))

\[
\begin{align*}
delta(\text{Sink}(S), \text{open}(F), S, T) &= \text{flowRate}(F) \times T \\
&\quad \leftarrow \text{action\_on}(\text{open}(F), S). \\
delta(\text{Sink}(S), \text{open}(F), S, T) &= 0 \\
&\quad \leftarrow \text{not action\_on}(\text{open}(F), S).
\end{align*}
\]
Finally, we introduce a function $f$ to sum up the contribution of actions on variables.

**Example (Summing up contribution on Sink(S))**

$$f(Sink(S), T) = Sink(S) + \delta(Sink(S), open(f1), S, T) + \delta(Sink(S), open(f2), S, T)$$

$$\leftarrow at(S + 1) = T_1, at(S) = T_0, 0 \leq T \leq T_1 - T_0.$$
Example (Invariant Monitoring for open(S) action)

\[\text{violated}(\text{open}(F), S_s, S_e) \quad \leftarrow \quad \text{duration}(\text{open}(F), S_s, S_e), \]
\[\quad \text{at}(S) = T_1, \text{at}(S + 1) = T_2, \]
\[\quad S_s \leq S \leq S_e, T_1 \leq T \leq T_2, \]
\[\quad f(\text{Tank}(F), S, T - T_1) = X, \]
\[\quad \text{not} \ (X > 0). \]
\[\quad \leftarrow \quad \text{violated}(\text{open}(F), S_s, S_e). \]
Potluck

What time?
What food?