Linguaggi Moderni per la Definizione di Domini di Pianificazione

Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Brescia

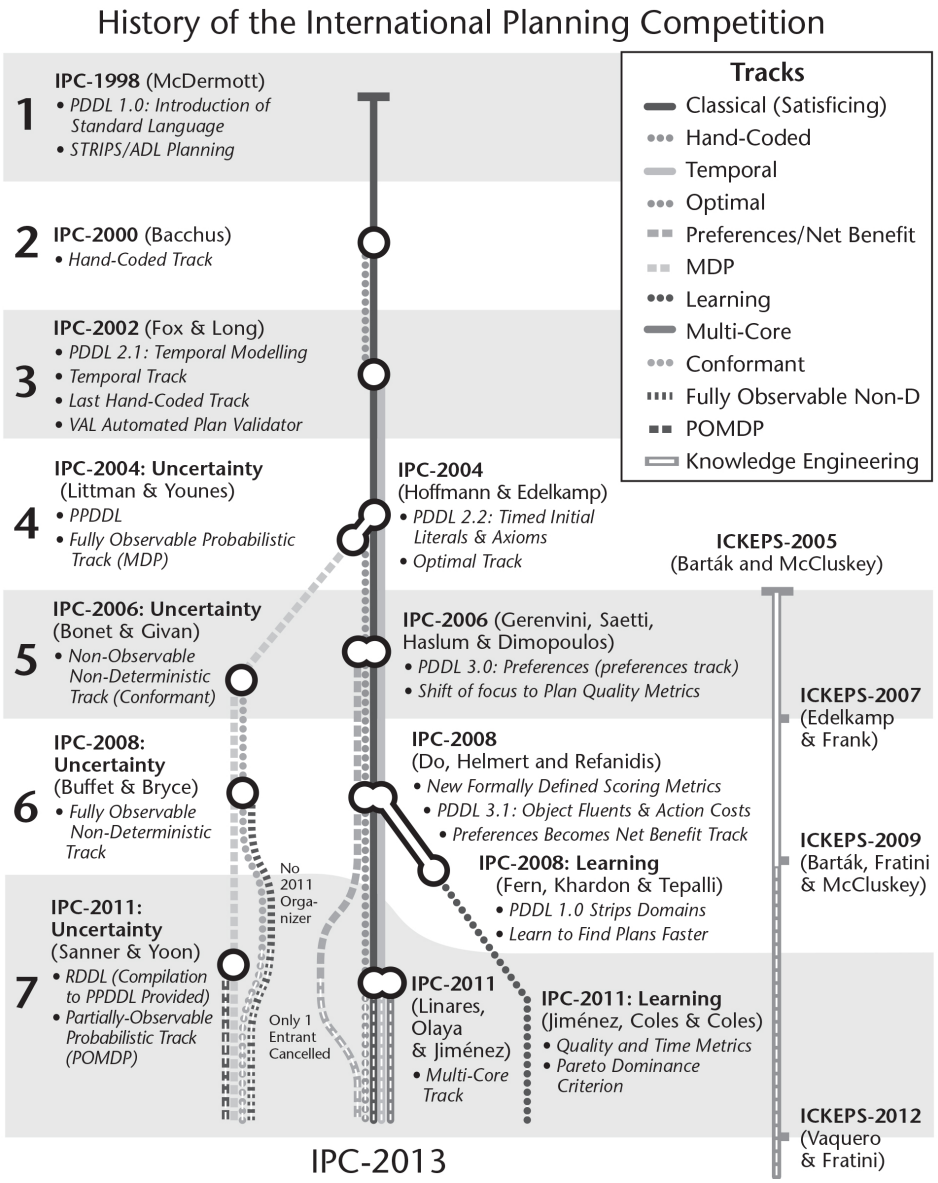
Alfonso Gerevini

alfonso.gerevini@unibs.it

Materiale per il corso di Intelligenza Artificiale

(preparato con il Prof. Alessandro Saetti)

PDDL


History of the International Planning Competition
Tracks
Classical (Satisficing)
Hand-Coded
Temporal
Optimal
Preferences/Net Benefit
MDP
Learning
Multi-Core
Conformant
Fully Observable Non-D
POMDP
Knowledge Engineering
1
IPC-1998 (McDermott)
• PDDL 1.0: Introduction of Standard Language
• STRIPS/ADL Planning
2
IPC-2000 (Bacchus)
• Hand-Coded Track
3
IPC-2002 (Fox & Long)
• PDDL 2.1: Temporal Modelling
• Temporal Track
• Last Hand-Coded Track
• VAL Automated Plan Validator
4
IPC-2004: Uncertainty (Littman & Younes)
• PPDDL
• Fully Observable Probabilistic Track (MDP)
IPC-2004 (Hoffmann & Edelkamp)
• PDDL 2.2: Timed Initial Literals & Axioms
• Optimal Track
ICKEPS-2005 (Barták and McCluskey)
5
IPC-2006: Uncertainty (Bonet & Givan)
• Non-Observable Non-Deterministic Track (Conformant)
IPC-2006 (Gerenvini, Saetti, Haslum & Dimopoulos)
• PDDL 3.0: Preferences (preferences track)
• Shift of focus to Plan Quality Metrics
ICKEPS-2007 (Edelkamp & Frank)
6
IPC-2008: Uncertainty (Buffet & Bryce)
• Fully Observable Non-Deterministic Track
IPC-2008 (Do, Helmert and Refanidis)
• New Formally Defined Scoring Metrics
• PDDL 3.1: Object Fluents & Action Costs
• Preferences Becomes Net Benefit Track
IPC-2008: Learning (Fern, Khardon & Tepalli)
• PDDL 1.0 Strips Domains
• Learn to Find Plans Faster
ICKEPS-2009 (Barták, Fratini & McCluskey)
No 2011 Organizer
7
IPC-2011: Uncertainty (Sanner & Yoon)
• RDDL (Compilation to PPDDL Provided)
• Partially-Observable Probabilistic Track (POMDP)
Only 1 Entrant Cancelled
IPC-2011 (Linares, Olaya & Jiménez)
• Multi-Core Track
IPC-2011: Learning (Jiménez, Coles & Coles)
• Quality and Time Metrics
• Pareto Dominance Criterion
ICKEPS-2012 (Vaquero & Fratini)
IPC-2013

PDDL

- *Classico* problema di pianificazione: $\langle F, A, I, G \rangle$
- linguaggio centrato sulle azioni
- ispirato alla formulazione STRIPS
- standardizzazione della sintassi per esprimere la semantica delle azioni (precondizioni e effetti positivi e negativi)

Introduzione (cont.)

- Esprime la fisica del dominio
- Non fornisce consigli sulla risoluzione del problema
- sintassi ispirata al LISP (liste di exp)
- separare la descrizione degli operatori da oggetti, condizioni iniziali e goals
- permette precondizioni e effetti negativi, effetti condizionali e la quantificazione nelle precondizioni e negli effetti (ADL)

PDDL (cont.)

- il PDDL permette di definire il modello di azioni e fatti
- l'instanziazione è eseguita sostituendo gli oggetti del problema alle variabili che compaiono nel modello di azioni e fatti
- il PDDL permette la tipizzazioni delle variabili che appaiono nei modelli di azioni e fatti
- i fatti istanziati sono proposizioni logiche costruite da predicati e termini (oggetti del problema)
- speciale predicato "=" per indicare il fatto che due termini si riferiscono allo stesso oggetto
- precondizioni ed effetti delle azioni istanziate sono formule logiche costruite da proposizioni logiche e connettivi logici

Esempio di Dominio PDDL

```
(define (domain zeno-travel)
(:requirements :typing)
(:types aircraft person city flevel - object)
(:predicates (at ?x - (either person aircraft) ?c - city)
             (in ?p - person ?a - aircraft)
             (fuel-level ?a - aircraft ?l - flevel)
             (next ?l1 ?l2 - flevel))

(:action board ...)
(:action debark ...)
(:action fly ...)
(:action zoom ...)
(:action refuel ...)
)
```

Esempio di Problema PDDL

```
(define (problem ZTRAVEL-1-2)
(:domain zeno-travel)
(:objects
        plane1 - aircraft
        person1 - person
        person2 - person
        city0 - city
        city1 - city
        city2 - city
        fl0 - flevel
        fl1 - flevel
        fl2 - flevel
        fl3 - flevel
        fl4 - flevel
        fl5 - flevel
        fl6 - flevel )
```

Esempio di Problema PDDL (cont.)

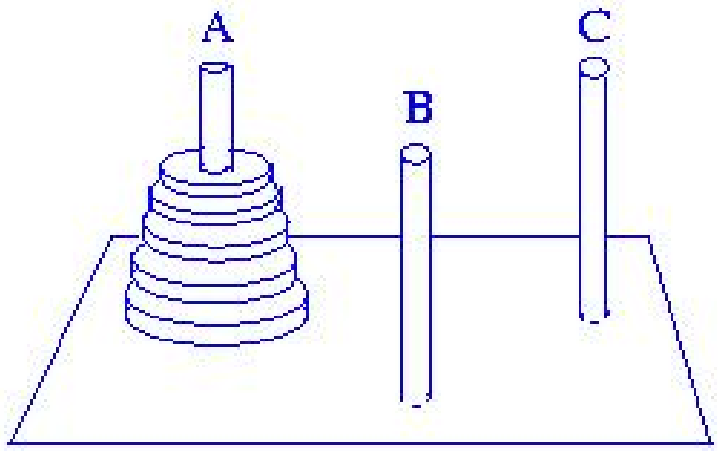
```
(:init
        (at plane1 city0)
        (fuel-level plane1 fl1)
        (at person1 city0)
        (at person2 city2)
        (next fl0 fl1)
        (next fl1 fl2)
        (next fl2 fl3)
        (next fl3 fl4)
        (next fl4 fl5)
        (next fl5 fl6) )
(:goal (and
        (at plane1 city1)
        (at person1 city0)
        (at person2 city2) ) ) )
```

Esempio di Azione PDDL

```
(:action fly
 :parameters (?a - aircraft ?c1 ?c2 - city ?l1 ?l2 - flevel)
 :precondition (and (at ?a ?c1)
                    (fuel-level ?a ?l1)
                    (next ?l2 ?l1))
 :effect (and (not (at ?a ?c1))
              (at ?a ?c2)
              (not (fuel-level ?a ?l1))
              (fuel-level ?a ?l2)))
```

Esercizio 1: Torre di hanoi

Modellizzare l'azione di spostamento per il problema della **Torre di Hanoi**



- si può spostare un disco sse sopra non c'è niente
- si può spostare il disco sopra ad un disco più grande

PDDL con Costrutti ADL

ADL (Action Definition Language):

- Introdotto nel 1989 da Edwin Pednault
- Era un linguaggio (proprietario) di un sistema di pianificazione sviluppato per robots
- È un'estensione del linguaggio STRIPS
- È più facile e coinciso codificare un problema
- Integrato fin dalla versione 1.7 del PDDL

PDDL con Precondizioni ADL

- Congiuntive (come in STRIPS)
- Negative: `(not (pred1 ?x))`
- Disgiuntivi: `(or (Pred1 ?x) (Pred2 ?x))`,

  `(imply (Pred1 ?x) (Pred2 ?x))`
- Quantificate esistenzialmente: `(exists (?x - type1) (Pred ?x))`
- Quantificate universalmente: `(forall (?x - type1) (Pred ?x))`

PDDL con Effetti ADL

- Effetti condizionali: `(when (cond) (cond-eff))`

```
(:action stack
 :parameters (?sob ?sunderob)
 :precondition (and (holding ?sob) (clear ?sunderob))
 :effect (and (not (holding ?sob)) (not (clear ?sunderob))
              (clear ?sob) (arm-empty) (on ?sob ?sunderob)
              (when (fragile ?sunderob) (broken ?sunderob))))
```

- Simple-ADL = ADL - effetti condizionali

PDDL: Tradurre ADL in STRIPS

Tradurre precondizioni ADL in precondizioni STRIPS:

1. Variabili quantificate vanno istanziate

2. La ridondanza va eliminata (ad esempio `(and A (and B C))` va sostituito con `(and A B C)`)

3. `(imply (A) (B))` va sostituito con `(or (not (A)) (B))`

4. La risultante formula va tradotta in Negation Normal Form

5. La risultante formula va tradotta in Disjunctive Normal Form

6. Ciascun disgiunto diventa la "precondition formula" di una nuova azione

PDDL: Tradurre ADL in STRIPS

Tradurre effetti condizionali `(when (cond) (eff-cond))` in STRIPS:

1–6. Ripetere i passi 1–6 per la traduzione di `cond`

7–8. Ripetere i passi 1–2 per `eff-cond`

9. Le restanti variabili quantificate vanno istanziate

10. `(when (cond) (eff-cond))` va tradotto in due operatori
    - Il primo operatore ha `cond` come precondizione aggiuntiva ed `eff-cond` come effetto aggiuntivo
    - Il secondo operatore ha `(not (cond))` come precondizione aggiuntiva

$\Rightarrow$ Tradurre precondizioni o effetti ADL in STRIPS può comportare un incremento esponenziale della descrizione del problema