Translating LPOD and CR-Prolog2 into Standard Answer Set Programs

Joohyung Lee and Zhun Yang

June 1, 2018

Translating LPOD and CR-Prolog$_2$ into Standard Answer Set Programs

Joohyung Lee and Zhun Yang
School of Computing, Informatics and Decision Systems Engineering
Arizona State University, Tempe, USA
(*e-mail:* {joolee, zyang90}@asu.edu)

submitted ; revised ; accepted

Abstract

Logic Programs with Ordered Disjunction (LPOD) is an extension of standard answer set programs to handle preference using the construct of ordered disjunction, and CR-Prolog$_2$ is an extension of standard answer set programs with consistency restoring rules and LPOD-like ordered disjunction. We present reductions of each of these languages into the standard ASP language, which gives us an alternative way to understand the extensions in terms of the standard ASP language.

1 Introduction

In answer set programming, each answer set encodes a solution to the problem that is being modeled. There is often a need to express that one solution is preferable to another, so several extensions of answer set programs were made to express a qualitative preference over answer sets. In Logic Programs with Ordered Disjunction (LPOD) (Brewka 2002), this is done by introducing the construct of ordered disjunction in the head of a rule: $A \times B \leftarrow \mathit{Body}$ intuitively means, when *Body* is true, if possible then A, but if A is not possible, then at least B. Proposition 2 from (Brewka 2002) states that there is no reduction of LPOD to disjunctive logic programs (Gelfond and Lifschitz 1991) based on the fact that the answer sets of disjunctive logic programs are subset-minimal whereas LPOD answer sets are not necessarily so. However, this justification is limited to translations that preserve the underlying signature, and it remained an open question if it is possible to turn LPOD into the language of standard ASP such as ASP-Core 2 (Calimeri et al. 2012) by using auxiliary atoms. In this paper, we provide a positive answer to this question.

We present a reduction of LPOD to standard answer set programs by compiling away ordered disjunctions. The translation gives us an alternative way to understand the semantics of LPOD in terms of the standard ASP language, and more generally, a method to express preference relations among answer sets. Instead of iterating the generator and the tester programs as in (Brewka et al. 2002), our reduction is one-pass: the preferred answer sets can be computed by calling an answer set solver one time.

It turns out that the translation idea is not restricted to LPOD but also applies to CR-Prolog$_2$ (Balduccini and Mellarkod 2004), which not only has a construct similar to ordered disjunction in LPOD but also inherits the construct of consistency-restoring rules—rules that can be added to make inconsistent programs to be consistent—from CR-Prolog (Balduccini and Gelfond 2003). With some modifications to the LPOD translation, we show that CR-Prolog$_2$ programs can also be turned into standard answer set programs by compiling away both ordered disjunctions and consistency-restoring rules.

The paper is organized as follows. Section 2 reviews LPOD and presents a translation that turns LPOD into standard answer set programs. Section 3 reviews CR-Prolog$_2$ and presents a translation that turns CR-Prolog$_2$ into standard answer set programs. The complete proofs are in the supplementary material at the TPLP archives (Lee and Yang 2018).

2 LPOD to ASP with Weak Constraints

2.1 Review: LPOD

We review the definition of LPOD by Brewka (2002). As in that paper, for simplicity, we assume the underlying signature is propositional.

Syntax: A (propositional) LPOD Π is $\Pi_{reg} \cup \Pi_{od}$, where its *regular part* Π_{reg} consists of usual ASP rules *Head* $\leftarrow$ *Body*, and its *ordered disjunction part* Π_{od} consists of *LPOD rules* of the form

$$C^1 \times \cdots \times C^n \leftarrow \mathit{Body} \tag{1}$$

in which C^i are atoms, n is at least 2, and *Body* is a conjunction of atoms possibly preceded by *not*.[1] Rule (1) intuitively says "when *Body* is true, if possible then C^1; if C^1 is not possible then C^2; . . . ; if all of $C^1, \dots, C^{n-1}$ are not possible then C^n."

Semantics: For an LPOD rule (1), its *i-th option* ($i = 1, \dots, n$) is defined as

$$C^i \leftarrow \mathit{Body}, \mathit{not}\ C^1, \dots, \mathit{not}\ C^{i-1}.$$

A *split program* of an LPOD Π is obtained from Π by replacing each rule in Π_{od} by one of its options. A set S of atoms is a *candidate answer set* of Π if it is an answer set of a split program of Π.

Example 1
(From (Brewka 2002)) The following LPOD Π_1,

$$\begin{array}{lcl} a \times b & \leftarrow & \mathit{not}\ c \\ b \times c & \leftarrow & \mathit{not}\ d, \end{array}$$

has four split programs:

$$\begin{array}{ll} a \leftarrow \mathit{not}\ c & a \leftarrow \mathit{not}\ c \\ b \leftarrow \mathit{not}\ d & c \leftarrow \mathit{not}\ d, \mathit{not}\ b \\ & \\ b \leftarrow \mathit{not}\ c, \mathit{not}\ a & b \leftarrow \mathit{not}\ c, \mathit{not}\ a \\ b \leftarrow \mathit{not}\ d & c \leftarrow \mathit{not}\ d, \mathit{not}\ b. \end{array} \tag{2}$$

Each of them has the following answer sets respectively, which are the candidate answer sets of Π_1.

$$\begin{array}{ll} \{a, b\} & \{c\} \\ & \\ \{b\} & \{b\}, \{c\}. \end{array}$$

[1] In (Brewka 2002), a usual ASP rule is viewed as a special case of a rule with ordered disjunction when $n = 1$ but in this paper, we distinguish them. This simplifies the presentation of the translation and also allows us to consider LPOD that are more general than the original definition by allowing modern ASP constructs such as aggregates.

A candidate answer set S of Π is said to *satisfy* rule (1)

- to degree 1 if S does not satisfy *Body*, and
- to degree j ($1 \leq j \leq n$) if S satisfies *Body* and $j = min\{k \mid C^k \in S\}$.

When Π_{od} contains m LPOD rules, the *satisfaction degree list* of a candidate answer set S of Π is $(d_1, \dots, d_m)$ where d_i is the degree to which S satisfies rule i in Π_{od}. For a candidate answer set S, let $S^i(\Pi)$ denote the set of rules in Π_{od} satisfied by S to degree i. For candidate answer sets S_1 and S_2 of Π, Brewka (2005) introduces the following four preference criteria.

1. **Cardinality-Preferred:** S_1 is *cardinality-preferred* to S_2 ($S_1 >^c S_2$) if there is a positive integer i such that $|S_1^i(\Pi)| > |S_2^i(\Pi)|$, and $|S_1^j(\Pi)| = |S_2^j(\Pi)|$ for all $j < i$.
2. **Inclusion-Preferred:** S_1 is *inclusion-preferred* to S_2 ($S_1 >^i S_2$) if there is a positive integer i such that $S_2^i(\Pi) \subset S_1^i(\Pi)$, and $S_1^j(\Pi) = S_2^j(\Pi)$ for all $j < i$.
3. **Pareto-Preferred:** S_1 is *Pareto-preferred* to S_2 ($S_1 >^p S_2$) if there is a rule that is satisfied to a lower degree in S_1 than in S_2, and there is no rule that is satisfied to a lower degree in S_2 than in S_1.
4. **Penalty-Sum-Preferred:** S_1 is *penalty-sum-preferred* to S_2 ($S_1 >^{ps} S_2$) if the sum of the satisfaction degrees of all rules is smaller in S_1 than in S_2.

A candidate answer set S of Π is a *k-preferred* ($k \in \{c, i, p, ps\}$) *answer set* if there is no candidate answer set S' of Π such that $S' >^k S$.

Example 1 (Continued)
Recall that Π_1 has three candidate answer sets: $\{a, b\}$, $\{b\}$, and $\{c\}$. Their satisfaction degree lists are (1,1), (2,1), and (1,2), respectively. One can check that $\{a, b\}$ is the only preferred answer set according to any of the four preference criteria.

Example 2
To illustrate the difference among the four preference criteria, consider the following LPOD Π_2 about picking a hotel near the Grand Canyon. $hotel(1)$ is a 2 star hotel but is close to the Grand Canyon, $hotel(2)$ is a 3 star hotel and the distance is medium, and $hotel(3)$ is a 4 star hotel but is too far.

$$\begin{array}{ll}
close \times med \times far \times tooFar & \bot \leftarrow hotel(2),\ not\ med \\
star4 \times star3 \times star2 & \bot \leftarrow hotel(2),\ not\ star3 \\
1\{hotel(X) : X = 1..3\}1 & \bot \leftarrow hotel(3),\ not\ tooFar \\
\bot \leftarrow hotel(1),\ not\ close & \bot \leftarrow hotel(3),\ not\ star4 \\
\bot \leftarrow hotel(1),\ not\ star2 &
\end{array}$$

Π_2 has 4×3 split programs but only the following three programs are consistent (The regular part of Π_2 is not listed).

$$\begin{array}{ll}
close & med \leftarrow not\ close \\
star2 \leftarrow not\ star4, not\ star3 & star3 \leftarrow not\ star4
\end{array}$$

$$\begin{array}{l}
tooFar \leftarrow not\ close, not\ med, not\ far \\
star4
\end{array}$$

The candidate answer sets of Π_2 and their satisfaction degree lists are

$$\begin{array}{ll}
S_1 = \{hotel(1), close, star2, \dots\}, (1,3) & S_2 = \{hotel(2), med, star3, \dots\}, (2,2) \\
S_3 = \{hotel(3), tooFar, star4, \dots\}, (4,1) &
\end{array}$$

By definition, the cardinality-preferred answer set is S_1, the inclusion-preferred answer sets are S_1 and S_3, the Pareto-preferred answer sets are S_1, S_2 and S_3, while the penalty-sum-preferred answer sets are S_1 and S_2.

2.2 An Alternative Way to Generate Candidate Answer Sets: Assumption Programs

Before we describe the translation of LPOD into standard answer set programs, we consider an alternative way to generate candidate answer sets together with their "assumption degrees," which serves as a basis of our translation.

Let Π be an LPOD with m LPOD rules. For an LPOD rule i ($i \in \{1, \dots, m\}$)

$$C_i^1 \times \cdots \times C_i^{n_i} \leftarrow \mathit{Body}_i\ , \tag{3}$$

its *x-th assumption* ($x \in \{0, \dots, n_i\}$), denoted by $O_i(x)$, is defined as the set of ASP rules

$$\begin{array}{rcll} \mathit{body}_i & \leftarrow & \mathit{Body}_i & (4)\\ \bot & \leftarrow & x = 0,\ \mathit{body}_i & (5)\\ \bot & \leftarrow & x > 0,\ \mathit{not}\ \mathit{body}_i & (6)\\ C_i^j & \leftarrow & \mathit{body}_i,\ x = j \qquad\qquad (\text{for } 1 \le j \le n_i) & (7)\\ \bot & \leftarrow & \mathit{body}_i,\ x \neq j, \mathit{not}\ C_i^1, \dots, \mathit{not}\ C_i^{j-1},\ C_i^j \qquad (\text{for } 1 \le j \le n_i) & (8) \end{array}$$

where body_i is a new, distinct atom for each LPOD rule i. Rules (4)—(6) ensure that the body of (3) is false iff $x = 0$. Rule (7) represents that C_i^x is true under the x-th assumption, and rule (8) ensures that all atoms $C_i^1, \dots, C_i^{x-1}$ are false. The last two rules together tells us that the first atom in $C_i^1, \dots, C_i^{n_i}$ that is true is C_i^x. The reason we call rules (4)—(8) the x-th assumption is because they encode a certain assumption imposed on rule (3) in deriving each candidate answer set: $x = 0$ assumes Body_i is false, whereas $x > 0$ assumes Body_i is true and the x-th atom in the head is to be derived.

An *assumption program* of an LPOD Π is obtained from Π by replacing each rule in Π_{od} by one of its assumptions. If each LPOD rule i is replaced by its x_i-th assumption, we call $(x_1, \dots, x_m)$ the *assumption degree list* of the assumption program.

The following proposition asserts that the candidate answer sets can be obtained from assumption programs instead of split programs.

Proposition 1

For any LPOD Π of σ and any set S of atoms of σ, S is a candidate answer set of Π iff $S \cup \{\mathit{body}_i \mid S \text{ satisfies the body of rule } i \text{ in } \Pi_{od}\}$ is an answer set of some assumption program of Π.

Example 1 (Continued) The assumptions for rule $a \times b \leftarrow \mathit{not}\ c$, denoted by $O_1(X_1)$, and the assumptions for rule $b \times c \leftarrow \mathit{not}\ d$, denoted by $O_2(X_2)$ are as follows, where X_1 and X_2 range over $\{0, 1, 2\}$.

$$\begin{array}{rrcl} O_1(X_1): & \mathit{body}_1 & \leftarrow & \mathit{not}\ c\\ & \bot & \leftarrow & X_1 = 0, \mathit{body}_1\\ & \bot & \leftarrow & X_1 > 0, \mathit{not}\ \mathit{body}_1\\ & a & \leftarrow & \mathit{body}_1, X_1 = 1\\ & b & \leftarrow & \mathit{body}_1, X_1 = 2\\ & \bot & \leftarrow & \mathit{body}_1, X_1 \neq 1, a\\ & \bot & \leftarrow & \mathit{body}_1, X_1 \neq 2, \mathit{not}\ a, b \end{array} \qquad \begin{array}{rrcl} O_2(X_2): & \mathit{body}_2 & \leftarrow & \mathit{not}\ d\\ & \bot & \leftarrow & X_2 = 0, \mathit{body}_2\\ & \bot & \leftarrow & X_2 > 0, \mathit{not}\ \mathit{body}_2\\ & b & \leftarrow & \mathit{body}_2, X_2 = 1\\ & c & \leftarrow & \mathit{body}_2, X_2 = 2\\ & \bot & \leftarrow & \mathit{body}_2, X_2 \neq 1, b\\ & \bot & \leftarrow & \mathit{body}_2, X_2 \neq 2, \mathit{not}\ b, c \end{array}$$

Π_1 has 9 assumption programs,

$$\begin{array}{lll}
O_1(0) \cup O_2(0) & O_1(0) \cup O_2(1) & \boxed{O_1(0) \cup O_2(2)}, \{c\} \\
O_1(1) \cup O_2(0) & \boxed{O_1(1) \cup O_2(1)}, \{a,b\} & O_1(1) \cup O_2(2) \\
O_1(2) \cup O_2(0) & \boxed{O_1(2) \cup O_2(1)}, \{b\} & O_1(2) \cup O_2(2),
\end{array}$$

among which the three assumption programs in the boxes are consistent. Their answer sets are shown together.

An advantage of considering assumption programs over split programs is that the satisfaction degrees—a basis of comparing the candidate answer sets—can be obtained from the assumption degrees with a minor modification (Section 2.3.1). This is in part because each candidate answer set is obtained from only one assumption program whereas the same candidate answer set can be obtained from multiple split programs (e.g., $\{b\}$ in Example 1).

2.3 Turning LPOD into Standard Answer Set Programs

We define a translation $\mathsf{lpod2asp}(\Pi)$ that turns an LPOD Π into a standard answer set program.

Let Π be an LPOD of signature σ where Π_{od} contains m propositional rules with ordered disjunction:

$$\begin{array}{lrcl}
1: & C_1^1 \times \dots \times C_1^{n_1} & \leftarrow & Body_1 \\
 & & \dots & \\
m: & C_m^1 \times \dots \times C_m^{n_m} & \leftarrow & Body_m
\end{array} \tag{9}$$

where $1, \dots, m$ are rule indices, and $n_i \geq 2$ for $1 \leq i \leq m$.

The first-order signature σ' of $\mathsf{lpod2asp}(\Pi)$ contains m-ary predicate constant a/m for each propositional constant a of σ. Besides, σ' contains the following predicate constants not in σ: ap/m ("assumption program"), *degree*$/(m{+}1)$, $body_i/m$ ($i \in \{1, \dots, m\}$), *prf*$/2$ ("preferred"), and pAS/m ("preferred answer set"). Furthermore, σ' contains the following predicate constants according to each preference criterion:

- for cardinality-preferred: $card/3$, $equ2degree/3$, $prf2degree/3$
- for inclusion-preferred: $even/1$, $equ2degree/3$, $prf2degree/3$
- for Pareto-preferred: $equ/2$
- for penalty-sum-preferred: $sum/2$.

2.3.1 Generate Candidate Answer Sets

The first part of the translation $\mathsf{lpod2asp}(\Pi)$ is to generate all candidate answer sets of Π based on the notion of assumption programs. We use the assumption degree list as a "name space" for each candidate answer set, so that we can compare them in a single answer set program.

1. We use atom $ap(x_1, \dots, x_m)$ to denote the assumption program whose assumption degree list is $(x_1, \dots, x_m)$. We consider all consistent assumption programs by generating a maximal set of $ap(\cdot)$ atoms: $ap(x_1, \dots, x_m)$ is included in an optimal answer set [2] iff the assumption program

[2] For programs containing weak constraints, an optimal answer set is defined by the penalty that comes from the weak constraints that are violated. (Calimeri et al. 2012)

denoted by $ap(x_1, \dots, x_m)$ is consistent.

$$\{ap(X_1, \dots, X_m): \ X_1 = 0..n_1, \ \dots \ , X_m = 0..n_m\}. \tag{10}$$

$$:\sim ap(X_1, \dots, X_m). \ \ [-1, X_1, \dots, X_m] \tag{11}$$

Rule (10) generates an arbitrary subset of $ap(\cdot)$ atoms, each of which records an assumption degree list. Rule (11) is a weak constraint that maximizes the number of $ap(\cdot)$ atoms by adding the penalty -1 for each true instance of $ap(X_1, \dots, X_m)$. Together with the rules below, these rules ensure that we consider all assumption programs that are consistent and that no candidate answer sets are missed in computing preference relationship in the second part of the translation.

2. We extend each atom to include the assumption degrees $X_1, \dots, X_m$, and append atom $ap(X_1, \dots, X_m)$ in the bodies of rules.

- For each rule *Head* $\leftarrow$ *Body* in Π_{reg}, $\mathsf{lpod2asp}(\Pi)$ contains

$$\mathit{Head}(X_1, \dots, X_m) \leftarrow \ ap(X_1, \dots, X_m), \mathit{Body}(X_1, \dots, X_m) \tag{12}$$

  where $\mathit{Head}(X_1, \dots, X_m)$ and $\mathit{Body}(X_1, \dots, X_m)$ are obtained from *Head* and *Body* by replacing each atom A in them with $A(X_1, \dots, X_m)$. Each schematic variable X_i ranges over $\{0, \dots, n_i\}$.
- For each rule

$$C_i^1 \times \cdots \times C_i^{n_i} \leftarrow \mathit{Body}_i$$

  in Π_{od}, where $n \geq 2$, $\mathsf{lpod2asp}(\Pi)$ contains

$$\mathit{body}_i(X_1, \dots, X_m) \leftarrow \ ap(X_1, \dots, X_m), \mathit{Body}_i(X_1, \dots, X_m) \tag{13}$$

$$\perp \leftarrow \ ap(X_1, \dots, X_m), \ X_i = 0, \ \mathit{body}_i(X_1, \dots, X_m) \tag{14}$$

$$\perp \leftarrow \ ap(X_1, \dots, X_m), \ X_i > 0, \ \mathit{not}\ \mathit{body}_i(X_1, \dots, X_m). \tag{15}$$

  And for $1 \leq j \leq n_i$, $\mathsf{lpod2asp}(\Pi)$ contains

$$C_i^j(X_1, \dots, X_m) \leftarrow \ \mathit{body}_i(X_1, \dots, X_m), X_i = j. \tag{16}$$

$$\begin{aligned} \perp \leftarrow \ & \mathit{body}_i(X_1, \dots, X_m), X_i \neq j, \\ & \mathit{not}\ C_i^1(X_1, \dots, X_m), \dots, \mathit{not}\ C_i^{j-1}(X_1, \dots, X_m), C_i^j(X_1, \dots, X_m). \end{aligned} \tag{17}$$

3. The satisfaction degree list can be obtained from the assumption degree list encoded in $ap(x_1, \dots, x_m)$ by changing x_i to 1 if it was 0. For this, $\mathsf{lpod2asp}(\Pi)$ contains

$$\begin{aligned} 1\{\mathit{degree}(ap(X_1, \dots, X_m), D_1, \dots, D_m): \ D_1 = 1..n_1, \dots, D_m = 1..n_m\}1 \\ \leftarrow ap(X_1, \dots, X_m). \end{aligned} \tag{18}$$

and for $1 \leq i \leq m$, $\mathsf{lpod2asp}(\Pi)$ contains

$$\perp \ \leftarrow \ \mathit{degree}(ap(X_1, \dots, X_m), D_1, \dots, D_m), \ X_i = 0, \ D_i \neq 1. \tag{19}$$

$$\perp \ \leftarrow \ \mathit{degree}(ap(X_1, \dots, X_m), D_1, \dots, D_m), \ X_i > 0, \ D_i \neq X_i. \tag{20}$$

Since all answer sets of the same assumption program are associated with the same satisfaction degree list, we say an assumption program *satisfies* LPOD rule i to degree d if its answer sets satisfy the rule to degree d. Rule (18) reads "for any assumption program $ap(x_1, \dots, x_m)$, it has exactly one assignment of satisfaction degrees $D_1, \dots, D_m$." Rules (19) and (20) say that the assumption program $ap(x_1, \dots, x_m)$ satisfies LPOD rule i to degree 1 if $x_i = 0$ (in which case *Body$_i$* is false) and to degree x_i if $x_i > 0$ (in which case *Body$_i$* is true).

Let us denote the set of rules (10)—(20) by $\mathsf{lpod2asp}(\Pi)_{base}$. Observe that the atoms $a(\mathbf{v})$ in the original signature σ are in the form of $a(\mathbf{v}, x_1, \dots, x_m)$ in the answer sets of $\mathsf{lpod2asp}(\Pi)_{base}$. We define a way to retrieve the candidate answer set of Π by removing $x_1, \dots, x_m$ as follows. Let S be an optimal answer set of $\mathsf{lpod2asp}(\Pi)_{base}$, and let

$$shrink(S, x_1, \dots, x_m) \text{ be } \{a(\mathbf{v}) \mid a(\mathbf{v}, x_1, \dots, x_m) \in S \text{ and } a(\mathbf{v}) \in \sigma\}.$$

If $S \models ap(x_1, \dots, x_m)$, we define the set $shrink(S, x_1, \dots, x_m)$ as a *candidate answer set on* σ of $\mathsf{lpod2asp}(\Pi)_{base}$.[3]

The following proposition asserts the soundness of the translation $\mathsf{lpod2asp}(\Pi)_{base}$.

Proposition 2
The candidate answer sets of an LPOD Π of signature σ are exactly the candidate answer sets on σ of $\mathsf{lpod2asp}(\Pi)_{base}$.

Example 1 Continued: The following is the encoding of $\mathsf{lpod2asp}(\Pi_1)_{base}$ in the input language of CLINGO.

```
%%%% 1 %%%%
{ap(X1,X2): X1=0..2, X2=0..2}.            :~ ap(X1,X2). [-1, X1, X2]

%%%% 2 %%%%
% a*b <- not c.
body_1(X1,X2) :- ap(X1,X2), not c(X1,X2).
:- ap(X1,X2), X1=0, body_1(X1,X2).        :- ap(X1,X2), X1>0, not body_1(X1,X2).

a(X1,X2) :- body_1(X1,X2), X1=1.          b(X1,X2) :- body_1(X1,X2), X1=2.

:- body_1(X1,X2), X1!=1, a(X1,X2).
:- body_1(X1,X2), X1!=2, not a(X1,X2), b(X1,X2).

% b*c <- not d.
body_2(X1,X2) :- ap(X1,X2), not d(X1,X2).
:- ap(X1,X2), X2=0, body_2(X1,X2).        :- ap(X1,X2), X2>0, not body_2(X1,X2).

b(X1,X2) :- body_2(X1,X2), X2=1.          c(X1,X2) :- body_2(X1,X2), X2=2.

:- body_2(X1,X2), X2!=1, b(X1,X2).
:- body_2(X1,X2), X2!=2, not b(X1,X2), c(X1,X2).

%%%% 3 %%%%
1{degree(ap(X1,X2), D1, D2): D1=1..2, D2=1..2}1 :- ap(X1,X2).

:- degree(ap(X1,X2), D1, D2), X1=0, D1!=1.
:- degree(ap(X1,X2), D1, D2), X1>0, D1!=X1.

:- degree(ap(X1,X2), D1, D2), X2=0, D2!=1.
:- degree(ap(X1,X2), D1, D2), X2>0, D2!=X2.
```

[3] We also apply this notation to the full translation $\mathsf{lpod2asp}(\Pi)$ and $\mathsf{crp2asp}(\Pi)$ below.

The optimal answer set S of $\mathsf{lpod2asp}(\Pi_1)_{base}$ is

$$\{ap(1,1), a(1,1), b(1,1), \dots, ap(2,1), b(2,1), \dots, ap(0,2), c(0,2), \dots\} \tag{21}$$

($body_i(\cdot)$ and $degree(\cdot)$ atoms are not listed). Since S satisfies $ap(1,1)$, $ap(2,1)$, and $ap(0,2)$, the candidate answer sets on σ of $\mathsf{lpod2asp}(\Pi_1)_{base}$ are

$$shrink(S,1,1) = \{a,b\}, \quad shrink(S,2,1) = \{b\}, \quad shrink(S,0,2) = \{c\}$$

which are exactly the candidate answer sets of Π_1.

2.3.2 Find Preferred Answer Sets

The second part of the translation $\mathsf{lpod2asp}(\Pi)$ is to compare the candidate answer sets to find the preferred answer sets. For each preference criterion, $\mathsf{lpod2asp}(\Pi)$ contains the following rules respectively. Below $maxdegree$ is $max\{n_i \mid i \in \{1,\dots,m\}\}$.

(a) **Cardinality-Preferred:** For this criterion, $\mathsf{lpod2asp}(\Pi)$ contains the following rules.

$$\begin{aligned}
card(P,X,N) &\leftarrow degree(P,D_1,\dots,D_m), X = 1..maxdegree,\\
&\qquad N = \{D_1 = X; \dots; D_m = X\}. &(22)\\
equ2degree(P_1,P_2,X) &\leftarrow card(P_1,X,N), card(P_2,X,N), P_1 \neq P_2. &(23)\\
prf2degree(P_1,P_2,X) &\leftarrow card(P_1,X,N_1), card(P_2,X,N_2), N_1 > N_2. &(24)\\
prf(P_1,P_2) &\leftarrow X = 0..maxdegree-1, prf2degree(P_1,P_2,X+1),\\
&\qquad X\{equ2degree(P_1,P_2,Y) : Y = 1..X\}. &(25)\\
pAS(X_1,\dots,X_m) &\leftarrow ap(X_1,\dots,X_m), \{prf(P, ap(X_1,\dots,X_m))\}0. &(26)
\end{aligned}$$

P, P_1, and P_2 denote assumption programs in the form of $ap(X_1,\dots,X_m)$. $card(P,X,N)$ is true if P satisfies N rules in Π_{od} to degree X. $equ2degree(P_1,P_2,X)$ is true if P_1 and P_2 have the same number of rules that are satisfied to degree X. $prf2degree(P_1,P_2,X)$ is true if P_1 satisfies more rules to degree X than P_2 does. $prf(P_1,P_2)$ is true if P_1 is cardinality-preferred to P_2: P_1 satisfies more rules to degree $X+1$ than P_2 does whereas they satisfy the same number of rules up to degree X. Rule (26) reads as: given an assumption program represented by $ap(X_1,\dots,X_m)$, if we cannot find an assumption program P that is more preferable, then the answer sets of $ap(X_1,\dots,X_m)$ are all preferred answer sets of Π. Note that P in rule (26) is a local variable that ranges over all $ap(\cdot)$ atoms.

(b) **Inclusion-Preferred:** For this criterion, $\mathsf{lpod2asp}(\Pi)$ contains the following rules.

$$\begin{aligned}
&even(0;2). &(27)\\
equ2degree(P_1,P_2,X) &\leftarrow P_1 \neq P_2, X = 1..maxdegree,\\
&\qquad degree(P_1,D_{11},\dots,D_{1m}), degree(P_2,D_{21},\dots,D_{2m}),\\
&\qquad C_1 = \{D_{11} = X; D_{21} = X\}, \dots, C_m = \{D_{1m} = X; D_{2m} = X\},\\
&\qquad even(C_1), \dots, even(C_m). &(28)\\
prf2degree(P_1,P_2,X) &\leftarrow P_1 \neq P_2, X = 1..maxdegree,\\
&\qquad not\ equ2degree(P_1,P_2,X),\\
&\qquad degree(P_1,D_{11},\dots,D_{1m}), degree(P_2,D_{21},\dots,D_{2m}),\\
&\qquad \{D_{11} \neq X; D_{21} = X\}1, \dots, \{D_{1m} \neq X; D_{2m} = X\}1. &(29)\\
prf(P_1,P_2) &\leftarrow X = 0..maxdegree-1, prf2degree(P_1,P_2,X+1),\\
&\qquad X\{equ2degree(P_1,P_2,Y) : Y = 1..X\}. &(30)\\
pAS(X_1,\dots,X_m) &\leftarrow ap(X_1,\dots,X_m), \{prf(P, ap(X_1,\dots,X_m))\}0. &(31)
\end{aligned}$$

where $\{D_{11} = X; D_{21} = X\}$ counts the number of true atoms in this set, so it equals to 0 (or 2) when none (or both) of $D_{11} = X$ and $D_{21} = X$ are true; $\{D_{11} \neq X; D_{21} = X\}1$ means that the number of true atoms in this set must be smaller or equal to 1, which means that $D_{11} \neq X$ and $D_{21} = X$ cannot be true at the same time – in other words, $D_{21} = X$ implies $D_{11} = X$.

(c) **Pareto-Preferred:** For this criterion, $\mathsf{lpod2asp}(\Pi)$ contains the following rules.

$$equ(P_1, P_2) \leftarrow \mathit{degree}(P_1, D_1, \dots, D_m), \mathit{degree}(P_2, D_1, \dots, D_m). \tag{32}$$

$$\begin{aligned} \mathit{prf}(P_1, P_2) \leftarrow\ & \mathit{degree}(P_1, D_{11}, \dots, D_{1m}), \mathit{degree}(P_2, D_{21}, \dots, D_{2m}), \\ & \mathit{not}\ equ(P_1, P_2), D_{11} \leq D_{21}, \dots, D_{1m} \leq D_{2m}. \end{aligned} \tag{33}$$

$$pAS(X_1, \dots, X_m) \leftarrow \mathit{ap}(X_1, \dots, X_m), \{\mathit{prf}(P, \mathit{ap}(X_1, \dots, X_m))\}0. \tag{34}$$

where $equ(P_1, P_2)$ means that P_1 is equivalent to P_2 at all degrees.

(d) **Penalty-Sum-Preferred:** For this criterion, $\mathsf{lpod2asp}(\Pi)$ contains the following rules.

$$sum(P, N) \leftarrow \mathit{degree}(P, D_1, \dots, D_m), N = D_1 + \cdots + D_m. \tag{35}$$

$$\mathit{prf}(P_1, P_2) \leftarrow sum(P_1, N_1), sum(P_2, N_2), N_1 < N_2. \tag{36}$$

$$pAS(X_1, \dots, X_m) \leftarrow \mathit{ap}(X_1, \dots, X_m), \{\mathit{prf}(P, \mathit{ap}(X_1, \dots, X_m))\}0. \tag{37}$$

where $sum(P, N)$ means that the sum of P's satisfaction degrees of all rules is N.

If $S \models pAS(x_1, \dots, x_m)$, we define the set $shrink(S, x_1, \dots, x_m)$ to be a *preferred answer set on σ* of $\mathsf{lpod2asp}(\Pi)$.

a

The following theorem assert the soundness of the translation $\mathsf{lpod2asp}(\Pi)$.

Theorem 1

Under any of the four preference criteria, the candidate (preferred, respectively) answer sets of an LPOD Π of signature σ are exactly the candidate (preferred, respectively) answer sets on σ of $\mathsf{lpod2asp}(\Pi)$.

Example 2 Continued: The first part of $\mathsf{lpod2asp}(\Pi_2)$ contains the following rules.

```
#const maxdegree = 4.

%%%% 1 %%%%

{ap(X1,X2): X1=0..4, X2=0..3}.             :~ ap(X1,X2). [-1, X1, X2]

%%%% 2 %%%%

1{hotel(H,X1,X2): H=1..3}1 :- ap(X1,X2).
:- ap(X1,X2), hotel(1,X1,X2), not close(X1,X2).
:- ap(X1,X2), hotel(1,X1,X2), not star2(X1,X2).
:- ap(X1,X2), hotel(2,X1,X2), not med(X1,X2).
:- ap(X1,X2), hotel(2,X1,X2), not star3(X1,X2).
:- ap(X1,X2), hotel(3,X1,X2), not tooFar(X1,X2).
:- ap(X1,X2), hotel(3,X1,X2), not star4(X1,X2).

% close * med * far * tooFar.

body_1(X1,X2) :- ap(X1,X2).
```

```
:- ap(X1,X2), X1=0, body_1(X1,X2).         :- ap(X1,X2), X1>0, not body_1(X1,X2).

close(X1,X2) :- body_1(X1,X2), X1=1.        med(X1,X2) :- body_1(X1,X2), X1=2.
far(X1,X2) :- body_1(X1,X2), X1=3.          tooFar(X1,X2) :- body_1(X1,X2), X1=4.

:- body_1(X1,X2), X1!=1, close(X1,X2).
:- body_1(X1,X2), X1!=2, not close(X1,X2), med(X1,X2).
:- body_1(X1,X2), X1!=3, not close(X1,X2), not med(X1,X2), far(X1,X2).
:- body_1(X1,X2), X1!=4, not close(X1,X2), not med(X1,X2), not far(X1,X2),
   tooFar(X1,X2).

% star4 * star3 * star2.

body_2(X1,X2) :- ap(X1,X2).

:- ap(X1,X2), X2=0, body_2(X1,X2).          :- ap(X1,X2), X2>0, not body_2(X1,X2).

star4(X1,X2) :- body_2(X1,X2), X2=1.         star3(X1,X2) :- body_2(X1,X2), X2=2.
star2(X1,X2) :- body_2(X1,X2), X2=3.

:- body_2(X1,X2), X2!=1, star4(X1,X2).
:- body_2(X1,X2), X2!=2, not star4(X1,X2), star3(X1,X2).
:- body_2(X1,X2), X2!=3, not star4(X1,X2), not star3(X1,X2), star2(X1,X2).

%%%% 3 %%%%

1{degree(ap(X1,X2), D1, D2): D1=1..4, D2=1..3}1 :- ap(X1,X2).

:- degree(ap(X1,X2), D1, D2), X1=0, D1!=1.
:- degree(ap(X1,X2), D1, D2), X1>0, D1!=X1.

:- degree(ap(X1,X2), D1, D2), X2=0, D2!=1.
:- degree(ap(X1,X2), D1, D2), X2>0, D2!=X2.
```

For the second part of the translation, $\mathsf{lpod2asp}(\Pi_2)$ contains one of the following sets of rules.

```
%%%% a. Cardinality %%%%
card(P,X,N) :- degree(P,D1,D2), X=1..maxdegree, N={D1=X; D2=X}.
equ2degree(P1,P2,X) :- card(P1,X,N), card(P2,X,N), P1!=P2.
prf2degree(P1,P2,X) :- card(P1,X,N1), card(P2,X,N2), N1>N2.
prf(P1,P2) :- X=0..maxdegree-1, prf2degree(P1,P2,X+1), X{equ2degree(P1,P2,Y): Y=1..X}.
pAS(X1,X2) :- ap(X1,X2), {prf(P, ap(X1,X2))}0.
```

```
%%%% b. Inclusion %%%%
even(0;2).
equ2degree(P1,P2,X) :- P1!=P2, X=1..maxdegree, degree(P1,D11,D12), degree(P2,D21,D22),
                         C1 = {D11=X; D21=X}, C2={D12=X; D22=X}, even(C1), even(C2).
prf2degree(P1,P2,X) :- P1!=P2, X=1..maxdegree, not equ2degree(P1,P2,X),
                        degree(P1,D11,D12), degree(P2,D21,D22),
                        {D11!=X; D21=X}1, {D12!=X; D22=X}1.
prf(P1,P2) :- X=0..maxdegree-1, prf2degree(P1,P2,X+1), X{equ2degree(P1,P2,Y): Y=1..X}.
pAS(X1,X2) :- ap(X1,X2), {prf(P, ap(X1,X2))}0.
```

```
%%%% c. Pareto %%%%
equ(P1,P2) :- degree(P1,D1,D2), degree(P2,D1,D2).
prf(P1,P2) :- degree(P1,D11,D12), degree(P2,D21,D22), not equ(P1,P2),
              D11<=D21, D12<=D22.
pAS(X1,X2) :- ap(X1,X2), {prf(P, ap(X1,X2))}0.
```

```
%%%% d. Penalty-Sum %%%%
sum(P,N) :- degree(P,D1,D2), N=D1+D2.
prf(P1,P2) :- sum(P1,N1), sum(P2,N2), N1<N2.
pAS(X1,X2) :- ap(X1,X2), {prf(P, ap(X1,X2))}0.
```

Note that each set of rules in the second part conservatively extends the answer set of the base program. For example, the optimal answer set of $\mathsf{lpod2asp}(\Pi_1)$ under Penalty-Sum preference is the union of (21) and $\{sum(ap(0,2),3), sum(ap(1,1),2), sum(ap(2,1),3), prf(ap(1,1),ap(0,2)), prf(ap(1,1),ap(2,1)), pAS(1,1)\}$, which indicates that $\{a,b\}$ is the preferred answer set.

The optimal answer set S of $\mathsf{lpod2asp}(\Pi_2)$ under the cardinality preference is

$$\begin{array}{llllll}\{pAS(1,3), & ap(1,3), & hotel(1,1,3), & close(1,3), & star2(1,3), \\ & ap(2,2), & hotel(2,2,2), & med(2,2), & star3(2,2), \\ & ap(4,1), & hotel(3,4,1), & tooFar(4,1), & star4(4,1),\dots\}\end{array}$$

Since S satisfies $ap(1,3)$, $ap(2,2)$, and $ap(4,1)$, the candidate answer sets on σ of $\mathsf{lpod2asp}(\Pi_2)$ are

$$\begin{array}{l}shrink(S,1,3) = \{hotel(1), close, star2\},\\ shrink(S,2,2) = \{hotel(2), med, star3\},\\ shrink(S,4,1) = \{hotel(3), tooFar, star4\},\end{array}$$

which are exactly the candidate answer sets of Π_2. Since S satisfies $pAS(1,3)$, the preferred answer sets on σ of $\mathsf{lpod2asp}(\Pi_2)$ is $shrink(S,1,3) = \{hotel(1), close, star2\}$ which is exactly the cardinality-preferred answer set of Π_2. Let

$$\begin{array}{l}pAS_1 = \{pAS(1,3), hotel(1,1,3), close(1,3), star2(1,3)\},\\ pAS_2 = \{pAS(2,2), hotel(2,2,2), med(2,2), star3(2,2)\},\\ pAS_3 = \{pAS(4,1), hotel(3,4,1), tooFar(4,1), star4(4,1)\}.\end{array}$$

The optimal answer sets of $\mathsf{lpod2asp}(\Pi_2)$ under 4 criteria contain

cardinality-preferred:	pAS_1	inclusion-preferred:	$pAS_1 \cup pAS_3$
Pareto-preferred:	$pAS_1 \cup pAS_2 \cup pAS_3$	penalty-sum-preferred:	$pAS_1 \cup pAS_2$

which are in a 1-1 correspondence with the preferred answer sets of Π_2 under each of the four criteria respectively.

3 CR-Prolog$_2$ to ASP with Weak Constraints

3.1 Review: CR-Prolog$_2$

We review the definition of CR-Prolog$_2$ from (Balduccini et al. 2003).

Syntax: A (propositional) CR-Prolog$_2$ program Π consists of four kinds of rules:

$$\textit{regular rule} \qquad \textit{Head} \leftarrow \textit{Body} \tag{38}$$

$$\textit{ordered rule} \qquad i: \; C^1 \times \cdots \times C^{n_i} \leftarrow \textit{Body} \tag{39}$$

$$\textit{cr-rule} \qquad i: \; \textit{Head} \overset{+}{\leftarrow} \textit{Body} \tag{40}$$

$$\textit{ordered cr-rule} \qquad i: \; C^1 \times \cdots \times C^{n_i} \overset{+}{\leftarrow} \textit{Body} \tag{41}$$

where *Head* $\leftarrow$ *Body* is a standard ASP rule, i is the index of the rule, C^j are atoms, and $n_i \geq 2$. The intuitive meaning of an ordered disjunction $C^1 \times \cdots \times C^{n_i}$ is similar to the one for LPOD. A cr-rule (40) or an ordered cr-rule (41) is *applied* in Π if it is treated as a usual ASP rule in Π (by replacing $\overset{+}{\leftarrow}$ with $\leftarrow$); it is not applied if it is omitted in Π. A cr-rule (40) or an ordered cr-rule (41) is applied only if the agent has no way to obtain a consistent set of beliefs using regular rules or ordered rules only. By *Head*(i) and *Body*(i), we denote the head and the body of rule i.

Semantics: The semantics of CR-Prolog$_2$ is based on the transformation from a CR-Prolog$_2$ program Π of signature σ into an answer set program H_Π, which is constructed as follows. The first-order signature of H_Π is $\sigma \cup \{\textit{choice}/2, \textit{appl}/1, \textit{fired}/1, \textit{isPreferred}/2\}$, where *choice* is a function constant, *appl*, *fired*, *isPreferred* are predicate constants not in σ.

1. Let R_Π be the set of rules obtained from Π by replacing every cr-rule and ordered cr-rule of index i with a rule:
$$i: \;\; \textit{Head}(i) \leftarrow \textit{Body}(i), \textit{appl}(i)$$
where *appl*(i) means rule i is applied. Notice that R_Π contains only regular rules and ordered rules.
H_Π is then obtained from R_Π by replacing every ordered rule of index r, where *Head*$(r) = C^1 \times \cdots \times C^{n_i}$, with the following rules (for $1 \leq j \leq n_i$):
$$\begin{array}{ll} C^j \leftarrow \textit{Body}(r), \textit{appl}(\textit{choice}(r,j)) & \\ \textit{fired}(r) \leftarrow \textit{appl}(\textit{choice}(r,j)) & \\ \textit{prefer}(\textit{choice}(r,j), \textit{choice}(r,j+1)) & (j < n_i) \\ \bot \leftarrow \textit{Body}(r), \textit{not fired}(r) & \end{array} \tag{42}$$
where *appl*$(\textit{choice}(r,j))$ means that the j-th atom in the ordered disjunction *Head*(r) is chosen, i.e., C^j is true if *Head*(r) is true.
2. H_Π also contains the following set of rules:
$$\begin{array}{l} \textit{isPreferred}(R1,R2) \leftarrow \textit{prefer}(R1,R2). \\ \textit{isPreferred}(R1,R3) \leftarrow \textit{prefer}(R1,R2), \textit{isPreferred}(R2,R3). \\ \bot \leftarrow \textit{isPreferred}(R,R). \\ \bot \leftarrow \textit{appl}(R1), \textit{appl}(R2), \textit{isPreferred}(R1,R2). \end{array}$$
where $R1, R2, R3$ are schematic variables ranging over indices of cr-rules and ordered cr-rules in Π as well as terms of the form *choice*$(\cdot)$.

By $atoms(H_\Pi, \{\textit{appl}\})$, we denote the set of atoms in H_Π in the form of *appl*$(\cdot)$. A *generalized answer set* of Π is an answer set of $H_\Pi \cup A$ where $A \subseteq atoms(H_\Pi, \{\textit{appl}\})$.

Let S_1, S_2 be generalized answer sets of Π. S_1 *dominates* S_2 if there exist r_1 and r_2 such that *appl*$(r_1) \in S_1$, *appl*$(r_2) \in S_2$, and *isPreferred*$(r_1, r_2) \in S_1 \cap S_2$. Further, we say this domination is *rule-wise* if r_1 and r_2 are indices of two cr-rules; *atom-wise* if r_1 and r_2 are two terms of the form *choice*$(\cdot)$. S_1 is a *candidate answer set* of Π if there is no other generalized answer set that dominates S_1.

The projection of S_1 onto σ is a *preferred answer set* of Π if S_1 is a candidate answer set of Π and there is no other candidate answer set S_2 such that $S_2 \cap atoms(H_\Pi, \{appl\}) \subset S_1$.

Example 3
(From (Balduccini et al. 2003)) Consider the following CR-Prolog$_2$ program Π_3:

$$\begin{array}{lll} q \leftarrow t. & p \leftarrow not\ q. & 1: \quad t \overset{+}{\leftarrow}. \\ s \leftarrow t. & r \leftarrow not\ s. & 2: \quad q \times s \overset{+}{\leftarrow}. \\ & \leftarrow p, r. & \end{array}$$

which has 5 generalized answer sets (the atoms formed by *isPreferred* or *fired* are omitted)

$$\begin{array}{l} S_1 = \{q, s, t, appl(1),\ \ prefer(choice(2,1), choice(2,2))\} \\ S_2 = \{q, r, appl(2), appl(choice(2,1)),\ \ prefer(choice(2,1), choice(2,2))\} \\ S_3 = \{p, s, appl(2), appl(choice(2,2)),\ \ prefer(choice(2,1), choice(2,2))\} \\ S_4 = \{q, s, t, appl(1), appl(2), appl(choice(2,1)),\ \ prefer(choice(2,1), choice(2,2))\} \\ S_5 = \{q, s, t, appl(1), appl(2), appl(choice(2,2)),\ \ prefer(choice(2,1), choice(2,2))\}. \end{array}$$

Since S_2 (atom-wise) dominates S_3 and S_5, the candidate answer sets are S_1, S_2, and S_4. Since $S_1 \cap atoms(H_{\Pi_3}, \{appl\}) \subset S_4$, the preferred answer sets of Π_3 are the projections from S_1 or S_2 onto σ.

3.2 Turning CR-Prolog$_2$ into ASP with Weak Constraints

We define a translation crp2asp(Π) that turns a CR-Prolog$_2$ program Π into an answer set program with weak constraints.

Let Π be a CR-Prolog$_2$ program of signature σ, where its rules are rearranged such that the cr-rules are of indices $1, \dots, k$, the ordered cr-rules are of indices $k+1, \dots, l$, and the ordered rules are of indices $l+1, \dots, m$.

For an ordered rule (39) or an ordered cr-rule (41), its i-th *assumption*, where $i \in \{1, \dots, n_i\}$, is defined as $C^i \leftarrow$ *Body*. An *assumption program* $AP(x_1, \dots, x_m)$ of Π whose *assumption degree list* is $(x_1, \dots, x_m)$ is obtained from Π as follows ($x_i \in \{0,1\}$ if $i = 1, \dots k$; $x_i \in \{0, \dots, n_i\}$ if $i = k{+}1, \dots, l$; $x_i \in \{1, \dots, n_i\}$ if $i = l{+}1, \dots, m$, where n_i is the number of atoms in the head of rule i).

- every regular rule (38) is in $AP(x_1, \dots, x_m)$;
- a cr-rule (40) is omitted if $x_i = 0$, and is replaced by *Head* $\leftarrow$ *Body* if $x_i = 1$;
- an ordered cr-rule (41) is omitted if $x_i = 0$, and is replaced by its x_i-th assumption if $x_i > 0$;
- an ordered rule (39) is replaced by its x_i-th assumption.

Besides, each assumption program $AP(x_1, \dots, x_m)$ contains

$$\begin{array}{l} isPreferred(R1, R2) \leftarrow prefer(R1, R2). \\ isPreferred(R1, R3) \leftarrow prefer(R1, R2), isPreferred(R2, R3). \\ \leftarrow isPreferred(R, R). \\ \leftarrow x_{r_1} > 0, x_{r_2} > 0, isPreferred(r_1, r_2). \qquad (1 \le r_1, r_2 \le l) \end{array}$$

The generalized answer sets of Π can be obtained from the answer sets of all the assumption programs of Π.

Proposition 3
For any CR-Prolog$_2$ program Π of signature σ, a set X of atoms is the projection of a generalized answer set of Π onto σ iff X is the projection of an answer set of an assumption program of Π onto σ.

Let Π_1 and Π_2 be two assumption programs of Π. We say an answer set S_1 of Π_1 *dominates* an answer set S_2 of Π_2 if (i) there exists a rule i in Π that is replaced by its j_1-th assumption in Π_1, is replaced by its j_2-th assumption in Π_2, and $j_1 < j_2$; or (ii) there exist 2 rules r_1, r_2 in Π such that r_1 is applied in Π_1, r_2 is applied in Π_2, and $prefer(r_1, r_2) \in S_1 \cap S_2$. Indeed, by Proposition 3, S_1 dominates S_2 iff the corresponding generalized answer set of the former dominates that of the latter.

An answer set program with weak constraints $\mathsf{crp2asp}(\Pi)$ is obtained from Π based on the notion of assumption programs as follows. The first-order signature σ' of $\mathsf{crp2asp}(\Pi)$ contains m-ary predicate constant a/m for each propositional constant a of σ. Besides, σ' contains the following predicate constants not in σ: ap/m, $dominate/2$, $\mathit{isPreferred}/(m{+}2)$, $candidate/m$, $lessCrRulesApplied/2$, and pAS/m.

1. To consider a maximal set of consistent assumption programs, $\mathsf{crp2asp}(\Pi)$ contains

$$\{ap(X_1, \dots, X_m) : X_1 = 0..1,\ \dots\ , X_k = 0..1,\ \ X_{k+1} = 0..n_{k+1}, \dots, X_l = 0..n_l,$$
$$X_{l+1} = 1..n_{l+1}, \dots, X_m = 1..n_m\}. \tag{43}$$
$$:\sim\ ap(X_1, \dots, X_m).\ [-1, X_1, \dots, X_m] \tag{44}$$

where n_i is the number of atoms in $\mathit{Head}(i)$, $ap(X_1, \dots, X_p)$ denotes an assumption program obtained from Π.

2. $\mathsf{crp2asp}(\Pi)$ contains the following rules to construct all assumption programs $AP(x_1, \dots, x_m)$:

- for each regular rule $\quad \mathit{Head} \leftarrow \mathit{Body} \quad$ in Π, $\mathsf{crp2asp}(\Pi)$ contains

$$\mathit{Head}(X_1, \dots, X_m) \leftarrow ap(X_1, \dots, X_m), Body(X_1, \dots, X_m) \tag{45}$$

- for each cr-rule $\quad i:\ \mathit{Head}_i \overset{+}{\leftarrow} \mathit{Body}_i \quad$ in Π, $\mathsf{crp2asp}(\Pi)$ contains

$$\mathit{Head}_i(X_1, \dots, X_m) \leftarrow ap(X_1, \dots, X_m), \mathit{Body}_i(X_1, \dots, X_m), X_i = 1 \tag{46}$$

- for each ordered rule or ordered cr-rule $\quad i:\ \ C_i^1 \times \dots \times C_i^n \overset{(+)}{\leftarrow} \mathit{Body}_i \quad$ in Π, for $1 \le j \le n_i$, $\mathsf{crp2asp}(\Pi)$ contains

$$C_i^j(X_1, \dots, X_m) \leftarrow ap(X_1, \dots, X_m), \mathit{Body}_i(X_1, \dots, X_m), X_i = j \tag{47}$$

3. To define $dominate$ in the semantics of CR-Prolog$_2$, $\mathsf{crp2asp}(\Pi)$ contains the following rules.

Atom-wise dominance: Instead of using $\mathit{choice}(\cdot)$ terms and $\mathit{appl}(\mathit{choice}(\cdot))$ atoms in (42), we represent the atom wise dominance by comparing the assumption degrees. For ordered cr-rules and ordered rules $i \in \{k+1, \dots m\}$, we include

$$dominate(ap(X_1, \dots, X_m), ap(Y_1, \dots, Y_m)) \leftarrow$$
$$ap(X_1, \dots, X_m), ap(Y_1, \dots, Y_m), 0 < X_i, X_i < Y_i \tag{48}$$

rule-wise dominance: The following rules are included only when Π contains an atom $\mathit{prefer}(\cdot)$.

r_1 and r_2 ranges over $\{1, \dots, l\}$.

$$isPreferred(R_1, R_2, X_1, \dots, X_m) \leftarrow prefer(R_1, R_2, X_1, \dots, X_m) \quad (49)$$

$$isPreferred(R_1, R_3, X_1, \dots, X_m) \leftarrow prefer(R_1, R_2, X_1, \dots, X_m), isPreferred(R_2, R_3, X_1, \dots, X_m) \quad (50)$$

$$\leftarrow isPreferred(R, R, X_1, \dots, X_m) \quad (51)$$

$$\leftarrow isPreferred(r_1, r_2, X_1, \dots, X_m), X_{r_1} > 0, X_{r_2} > 0 \quad (52)$$

$$dominate(ap(X_1, \dots, X_m), ap(Y_1, \dots, Y_m)) \leftarrow ap(X_1, \dots, X_m), ap(Y_1, \dots, Y_m), isPreferred(r_1, r_2, X_1, \dots, X_m), isPreferred(r_1, r_2, Y_1, \dots, Y_m), X_{r_1} > 0, Y_{r_2} > 0 \quad (53)$$

We say an assumption program Π_1 *dominates* an assumption program Π_2 if an answer set of Π_1 dominates an answer set of Π_2. Indeed, our translation guarantees that if Π_1 dominates Π_2, all answer sets of Π_1 dominates any answer sets of Π_2. Rule (48) says that the assumption program $AP(x_1, \dots, x_m)$ dominates the assumption program $AP(y_1, \dots, y_m)$ if there exists a rule i in Π that is replaced by its x_i-th assumption in $AP(x_1, \dots, x_m)$, by its y_i-th assumption in $AP(y_1, \dots, y_m)$, and $x_i < y_i$. Rules (49), (50), (51), (52) are the set of rules in the semantics of CR-Prolog$_2$ with the extended signature σ'. Rule (53) says that $AP(x_1, \dots, x_m)$ dominates $AP(y_1, \dots, y_m)$ if *isPreferred*(r_1, r_2) is true in both assumption programs while r_1 is applied in $AP(x_1, \dots, x_m)$ and r_2 is applied in $AP(y_1, \dots, y_m)$.

4. To define candidate answer sets in the semantics of CR-Prolog$_2$, crp2asp(Π) contains

$$candidate(X_1, \dots, X_m) \leftarrow ap(X_1, \dots, X_m), \{dominate(P, ap(X_1, \dots, X_m))\}0 \quad (54)$$

Rule (54) says that the answer sets of $AP(x_1, \dots, x_m)$ are candidate answer sets if there does not exist an assumption program P that dominates $AP(x_1, \dots, x_m)$.

5. To define the preference between two candidate answer sets and find preferred answer sets, crp2asp(Π) contains

$$lessCrRulesApplied(ap(X_1, \dots, X_m), ap(Y_1, \dots, Y_m)) \leftarrow candidate(X_1, \dots, X_m), candidate(Y_1, \dots, Y_m), 1\{X_1 \neq Y_1; \dots; X_m \neq Y_m\}, X_1 \leq Y_1, \dots, X_m \leq Y_m \quad (55)$$

$$pAS(X_1, \dots, X_m) \leftarrow candidate(X_1, \dots, X_m), \{lessCrRulesApplied(P, ap(X_1, \dots, X_m))\}0 \quad (56)$$

Rule (55) says that for any different assumption programs $AP(x_1, \dots, x_m)$ and $AP(y_1, \dots, y_m)$ whose answer sets are candidate answer sets, if all the choices in $AP(x_1, \dots, x_m)$ is not worse than [4] those in $AP(y_1, \dots, y_m)$, then the former must apply less cr-rules or ordered cr-rules than the latter. Rule (56) says that the answer sets of $AP(x_1, \dots, x_m)$ are preferred answer sets if these answer sets are candidate answer sets and there does not exist an assumption program P that applies less cr-rules than $AP(x_1, \dots, x_m)$.

Let S be an optimal answer set of crp2asp(Π); $x_1, \dots, x_m$ be a list of integers. If $S \models ap(x_1, \dots, x_m)$, we define the set $shrink(S, x_1, \dots, x_m)$ as a *generalized answer set on* σ of crp2asp(Π); if $S \models candidate(x_1, \dots, x_m)$, we define the set $shrink(S, x_1, \dots, x_m)$ as a *candidate answer set on* σ of crp2asp(Π); if $S \models pAS(x_1, \dots, x_p)$, we define the set $shrink(S, x_1, \dots, x_p)$ as a *preferred answer set on* σ of crp2asp(Π).

[4] i.e., for any rule i in Π, if it is applied in $AP(x_1, \dots, x_m)$, it must be applied in $AP(y_1, \dots, y_m)$; if it is replaced by its x_i-th assumption in $AP(x_1, \dots, x_m)$, it must be replaced by its y_i-th assumption in $AP(y_1, \dots, y_m)$ and $x_i \leq y_i$

Theorem 2
For any CR-Prolog$_2$ program Π of signature σ, (a) the projections of the generalized answer sets of Π onto σ are exactly the generalized answer sets on σ of $\mathsf{crp2asp}(\Pi)$. (b) the projections of the candidate answer sets of Π onto σ are exactly the candidate answer sets on σ of $\mathsf{crp2asp}(\Pi)$. (c) the preferred answer sets of Π are exactly the preferred answer sets on σ of $\mathsf{crp2asp}(\Pi)$.

Example 3 Continued: The translated ASP program $\mathsf{crp2asp}(\Pi_3)$ is

```
%%%% 1 %%%%
{ap(X1,X2): X1=0..1, X2=0..2}.            :~ ap(X1,X2). [-1,X1,X2]

%%%% 2 %%%%
q(X1,X2) :- ap(X1,X2), t(X1,X2).          s(X1,X2) :- ap(X1,X2), t(X1,X2).
p(X1,X2) :- ap(X1,X2), not q(X1,X2).      r(X1,X2) :- ap(X1,X2), not s(X1,X2).
:- ap(X1,X2), p(X1,X2), r(X1,X2).

% 1: t <+-.
t(X1,X2) :- ap(X1,X2), X1=1.

% 2: q*s <+-.
q(X1,X2) :- ap(X1,X2), X2=1.              s(X1,X2) :- ap(X1,X2), X2=2.

%%%% 3 %%%%
dominate(ap(X1,X2), ap(Y1,Y2)) :- ap(X1,X2), ap(Y1,Y2), 0<X1, X1<Y1.
dominate(ap(X1,X2), ap(Y1,Y2)) :- ap(X1,X2), ap(Y1,Y2), 0<X2, X2<Y2.

%%%% 4 %%%%
candidate(X1,X2) :- ap(X1,X2), {dominate(P,ap(X1,X2))}0.

%%%% 5 %%%%
lessCrRulesApplied(ap(X1,X2), ap(Y1,Y2)) :- candidate(X1,X2), candidate(Y1,Y2),
       1{X1!=Y1;X2!=Y2}, X1<=Y1, X2<=Y2.
pAS(X1,X2) :- candidate(X1,X2), {lessCrRulesApplied(P,ap(X1,X2))}0.
```

The optimal answer set S of $\mathsf{crp2asp}(\Pi_3)$ is

$$\begin{array}{rrll}
\{pAS(1,0), & candidate(1,0), & ap(1,0), & t(1,0), q(1,0), s(1,0), \\
pAS(0,1), & candidate(0,1), & ap(0,1), & q(0,1), r(0,1), \\
& & ap(0,2), & p(0,2), s(0,2), \\
& candidate(1,1), & ap(1,1), & t(1,1), q(1,1), s(1,1), \\
& & ap(1,2), & t(1,2), q(1,2), s(1,2), \dots\}.
\end{array}$$

Since S satisfies $ap(1,0)$, $ap(0,1)$, $ap(0,2)$, $ap(1,1)$, $ap(1,2)$, the generalized answer sets on σ of $\mathsf{crp2asp}(\Pi_3)$ are

$$\begin{array}{ll}
shrink(S,1,0) = \{t,q,s\} & \\
shrink(S,0,1) = \{q,r\} & shrink(S,1,1) = \{t,q,s\} \\
shrink(S,0,2) = \{p,s\} & shrink(S,1,2) = \{t,q,s\}
\end{array}$$

which are exactly the projections of the generalized answer sets of Π_3 onto σ. Similarly, we observe that the candidate (preferred, respectively) answer sets on σ of $\mathsf{crp2asp}(\Pi_3)$ are exactly the projections of the candidate (preferred, respectively) answer sets of Π_3 onto σ.

Furthermore, let $\Pi_3' = \Pi_3 \cup \{\mathit{prefer}(2,1).\}$. The translation $\mathsf{crp2asp}(\Pi_3')$ is $\mathsf{crp2asp}(\Pi_3) \cup R$, where R is the set of the following rules:

```
%%%% 2 %%%%
prefer(2,1,X1,X2) :- ap(X1,X2).

%%%% 3 %%%%
isPreferred(R1,R2,X1,X2) :- prefer(R1,R2,X1,X2).
isPreferred(R1,R3,X1,X2) :- prefer(R1,R2,X1,X2), isPreferred(R2,R3,X1,X2).
:- isPreferred(R,R,X1,X2).
:- isPreferred(2,1,X1,X2), X2>0, X1>0.

dominate(ap(X1,X2), ap(Y1,Y2)) :- ap(X1,X2), ap(Y1,Y2),
        isPreferred(2,1,X1,X2), isPreferred(2,1,Y1,Y2), X2>0, Y1>0.
```

The optimal answer set S of $\mathsf{crp2asp}(\Pi_3')$ is

$$\begin{array}{llll} \{ & & ap(1,0), & t(1,0), q(1,0), s(1,0), \\ pAS(0,1), & candidate(0,1), & ap(0,1), & q(0,1), r(0,1), \\ & & ap(0,2), & p(0,2), s(0,2), \dots \} \end{array}$$

and it is easy to check that the generalized (/candidate/preferred) answer sets on σ of $\mathsf{crp2asp}(\Pi_3')$ are exactly the projections of the generalized (/candidate/preferred) answer sets of Π_3' onto σ.

4 Related Work and Conclusion

We presented reductions of LPOD and CR-Prolog$_2$ into the standard ASP language, which explains the new constructs for preference handling in terms of the standard ASP language. The one-pass translations are theoretically interesting. They may be a useful tool for studying the mathematical properties of LPOD and CR-Prolog$_2$ programs by reducing them to more well-known properties of standard answer set programs. Both translations are "almost" modular in the sense that the translations are rule-by-rule but the argument of each atom representing the assumption degrees may need to be expanded when new rules are added.

However, the direct implementations may not lead to effective implementations. The size of $\mathsf{lpod2asp}(\Pi)$ and $\mathsf{crp2asp}(\Pi)$ after grounding could be exponential to the size of the non-regular rules in Π. This is because these translations compare all possible assumption programs whose number is exponential to the size of non-regular rules. One may consider parallelizing the computation of assumption programs since they are disjoint from each other according to the translations.

In a sense, our translations are similar to the meta-programming approach to handle preference in ASP (e.g., (Delgrande et al. 2003)) in that we turn LPOD and CR-Prolog$_2$ into answer set programs that do not have the built-in notion of preference.

In (Brewka et al. 2002), LPOD is implemented using SMODELS. The implementation interleaves the execution of two programs–a generator which produces candidate answer sets and a tester which checks whether a given candidate answer set is maximally preferred or produces a more preferred candidate if it is not. An implementation of CR-Prolog reported in (Balduccini 2007) uses a similar algorithm. In contrast, the reductions shown in this paper can be computed by calling an answer set solver one time without the need for iterating the generator and the tester. This feature may be useful for debugging LPOD and CR-Prolog$_2$ programs because it allows us to compare all candidate and preferred answer sets globally.

Asprin (Brewka et al. 2015) provides a flexible way to express various preference relations over answer sets and is implemented in CLINGO. Similar to the existing LPOD solvers, CLINGO makes iterative calls to find preferred answer sets, unlike the one-shot execution as we do.

Asuncion *et al.* (2014) presents a first-order semantics of logic programs with ordered disjunction by translation into second-order logic whereas our translation is into the standard answer set programs.

Acknowledgements: We are grateful to Yi Wang and the anonymous referees for their useful comments. This work was partially supported by the National Science Foundation under Grant IIS-1526301.

References

ASUNCION, V., ZHANG, Y., AND ZHANG, H. 2014. Logic programs with ordered disjunction: first-order semantics and expressiveness. In *Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning*. AAAI Press, 2–11.

BALDUCCINI, M., , BALDUCCINI, M., AND MELLARKOD, V. 2003. CR-Prolog with ordered disjunction. In *In ASP03 Answer Set Programming: Advances in Theory and Implementation, volume 78 of CEUR Workshop proceedings*.

BALDUCCINI, M. 2007. CR-MODELS: an inference engine for CR-Prolog. In *Proceedings of the 9th international conference on Logic programming and nonmonotonic reasoning*. Springer-Verlag, 18–30.

BALDUCCINI, M. AND GELFOND, M. 2003. Logic programs with consistency-restoring rules. In *International Symposium on Logical Formalization of Commonsense Reasoning, AAAI 2003 Spring Symposium Series*. 9–18.

BALDUCCINI, M. AND MELLARKOD, V. 2004. A-Prolog with CR-rules and ordered disjunction. In *Intelligent Sensing and Information Processing, 2004. Proceedings of International Conference on*. IEEE, 1–6.

BREWKA, G. 2002. Logic programming with ordered disjunction. In *AAAI/IAAI*. 100–105.

BREWKA, G. 2005. Preferences in answer set programming. In *CAEPIA*. Vol. 4177. Springer, 1–10.

BREWKA, G., DELGRANDE, J. P., ROMERO, J., AND SCHAUB, T. 2015. asprin: Customizing answer set preferences without a headache. In *AAAI*. 1467–1474.

BREWKA, G., NIEMELÄ, I., AND SYRJÄNEN, T. 2002. Implementing ordered disjunction using answer set solvers for normal programs. In *European Workshop on Logics in Artificial Intelligence*. Springer, 444–456.

CALIMERI, F., FABER, W., GEBSER, M., IANNI, G., KAMINSKI, R., KRENNWALLNER, T., LEONE, N., RICCA, F., AND SCHAUB, T. 2012. ASP-Core-2: Input language format. *ASP Standardization Working Group, Tech. Rep.*

DELGRANDE, J. P., SCHAUB, T., AND TOMPITS, H. 2003. A framework for compiling preferences in logic programs. *Theory and Practice of Logic Programming 3,* 2, 129–187.

FERRARIS, P. 2011. Logic programs with propositional connectives and aggregates. *ACM Transactions on Computational Logic (TOCL) 12,* 4, 25.

FERRARIS, P., LEE, J., LIFSCHITZ, V., AND PALLA, R. 2009. Symmetric splitting in the general theory of stable models. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*. 797–803.

GELFOND, M. AND LIFSCHITZ, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing 9*, 365–385.

LEE, J. AND YANG, Z. 2018. Online appendix for the paper "Translating LPOD and CR-Prolog2 into standard answer set programs".