Решение задач удовлетворения ограничений с помощью их сведения к задаче *SAT*. Программное средство *Sugar*

Курс «Программные средства для решения задачи удовлетворения ограничений»

НИУ ИТМО, кафедра «Компьютерные технологии»

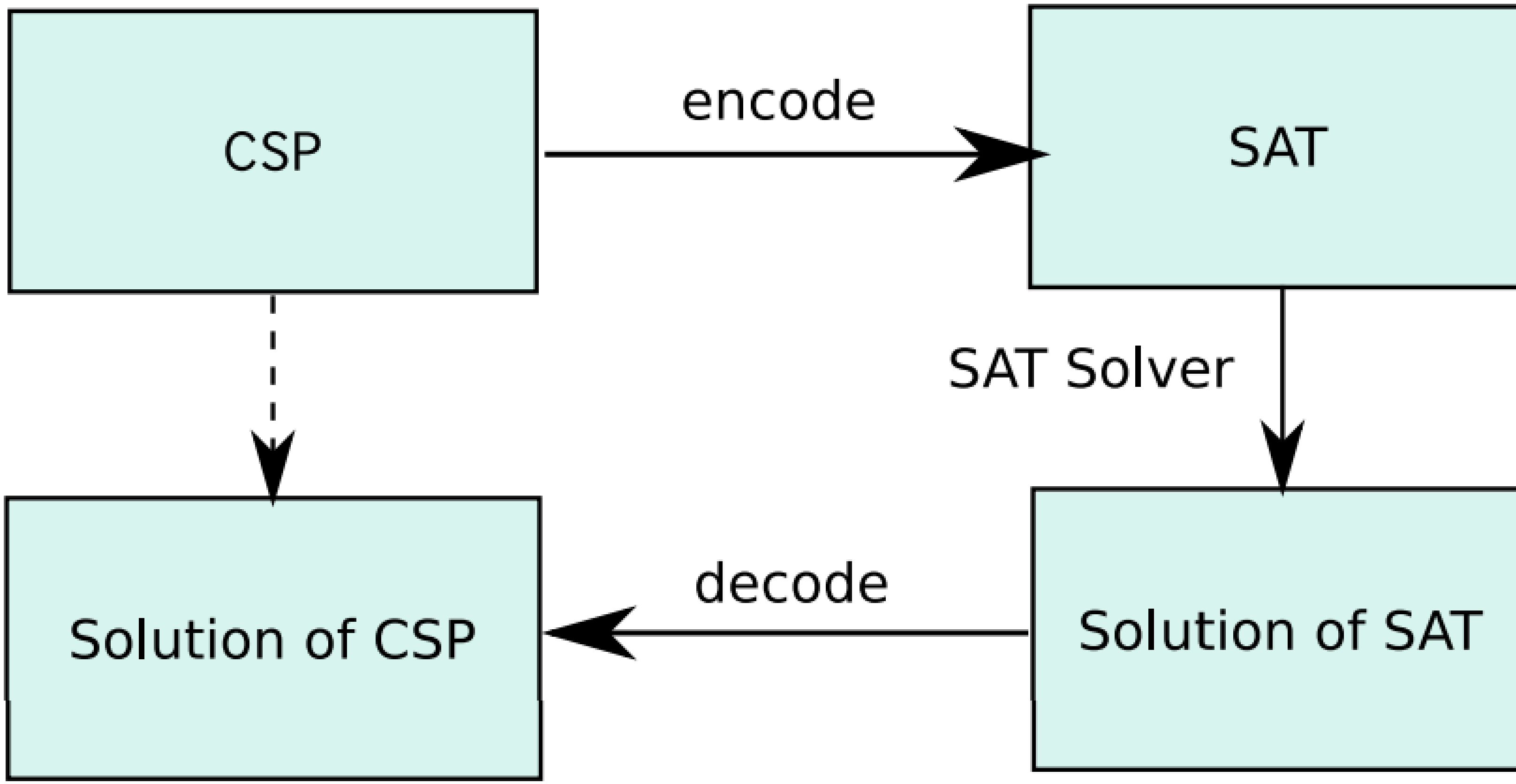
Задача *SAT*

- Задача выполнимости булевой формулы
- Одна из центральных задач информатики
- *NP*-полна
- Имеется множество высокопроизводительных программных средств, находящих выполняющую подстановку эвристически

Общая схема сведения задачи удовлетворения ограничений (*CSP*) к задаче *SAT* (1)

- Кодирование переменных при помощи булевых, преобразование ограничений
- Запуск стороннего программного средства, решающего задачу *SAT*
- Обратный переход к задаче *CSP*

Общая схема сведения задачи удовлетворения ограничений (*CSP*) к задаче *SAT* (2)



Прямое кодирование целочисленных переменных

- Пусть x – переменная
- Тогда булева переменная $p(x = i)$ задается тогда и только тогда, когда i – возможное значение x
- Это наиболее часто используемый подход
- Пример: для $x \in \{2, 3, 4, 5, 6\}$ будут заданы переменные $p(\mathrm{x} = 2)$, $p(\mathrm{x} = 3)$, $p(\mathrm{x} = 4)$, $p(\mathrm{x} = 5)$, $p(\mathrm{x} = 6)$

Пример преобразования ограничений при прямом кодировании (1)

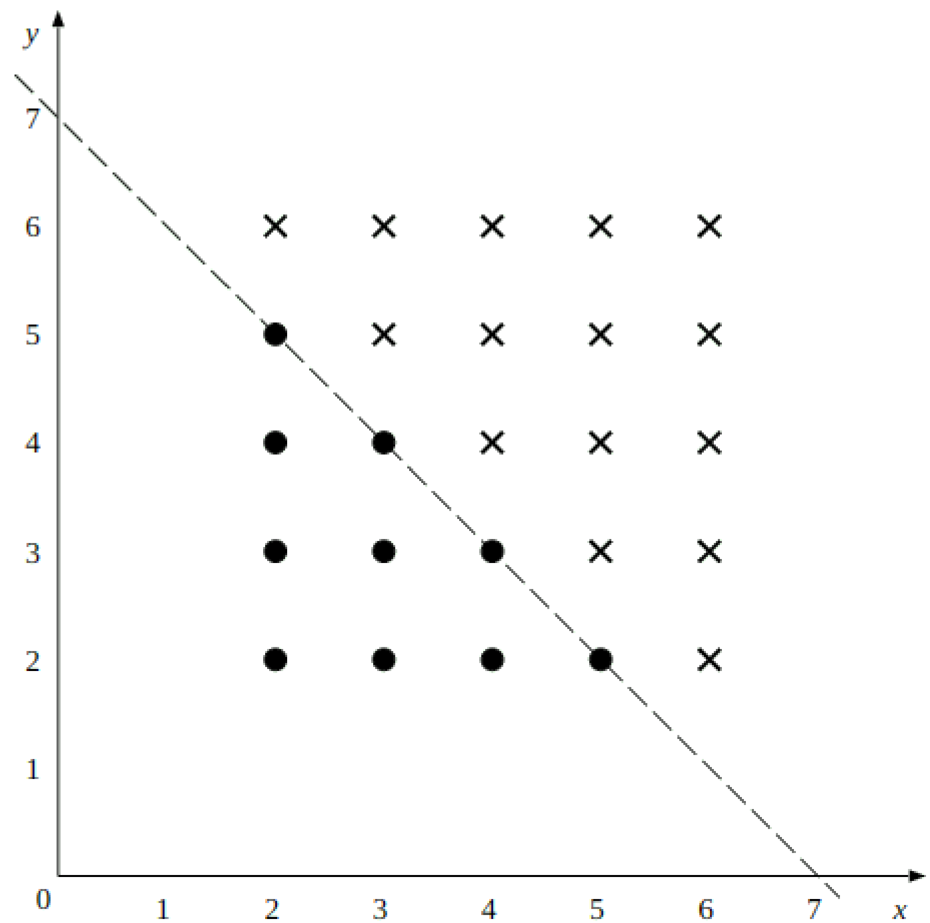
- $x + y \leq 7$
- Преобразуется в 15 ограничений:

$\neg p(x = 2) \vee \neg p(y = 6)$
$\neg p(x = 3) \vee \neg p(y = 5)$
$\neg p(x = 3) \vee \neg p(y = 6)$
$\neg p(x = 4) \vee \neg p(y = 4)$
$\neg p(x = 4) \vee \neg p(y = 5)$
$\neg p(x = 4) \vee \neg p(y = 6)$
$\neg p(x = 5) \vee \neg p(y = 3)$
$\neg p(x = 5) \vee \neg p(y = 4)$

$\neg p(x = 5) \vee \neg p(y = 5)$
$\neg p(x = 5) \vee \neg p(y = 6)$
$\neg p(x = 6) \vee \neg p(y = 2)$
$\neg p(x = 6) \vee \neg p(y = 3)$
$\neg p(x = 6) \vee \neg p(y = 4)$
$\neg p(x = 6) \vee \neg p(y = 5)$
$\neg p(x = 6) \vee \neg p(y = 6)$

Пример преобразования ограничений при прямом кодировании (2)

$x + y \leq 7$



Кодирование порядка (order encoding)

- Используются переменные вида $p(x \le i)$
- Пример: $x \in \{2, 3, 4, 5, 6\}$
- Получаем переменные $p(x \le 2)$, $p(x \le 3)$, $p(x \le 4)$, $p(x \le 5)$
- Ограничение $p(x \le 6)$ не нужно, поскольку выполнено всегда

Пример преобразования ограничений при кодировании порядка

- Рассмотрим ту же переменную $x \in \{2, 3, 4, 5, 6\}$
- Нужно три ограничения:

$\neg p(x \leq 2) \vee p(x \leq 3)$

$\neg p(x \leq 3) \vee p(x \leq 4)$

$\neg p(x \leq 4) \vee p(x \leq 5)$

Кодирование порядка: свойства

- Пусть $d = |D(x)|$
- Для одной переменной требуется $O(d)$ булевых переменных и $O(d)$ SAT-дизъюнктов
- Для ограничения в виде линейного неравенства, содержащего m переменных , требуется $O(d^{n-1})$ ограничений SAT-дизъюнктов

Другие способы кодирования переменных

- Multivalued encoding (Selman, 1992)
- Support encoding (Kasif, 1990)
- Log encoding (Iwama, 1994)
- Log-support encoding (Gavanelli, 2007)

Программное средство *Sugar*

- http://bach.istc.kobe-u.ac.jp/sugar/
- Основано на сведении задачи удовлетворения ограничений к задаче SAT
- В качестве программного средства для решения задачи *SAT* могут использоваться *zChaff*, *minisat*, *CryptoMiniSat*…
- Используется кодирование порядка

Sugar: задание областей значений

```
DomainDefinition ::=
    (domain DomainName LowerBound UpperBound) |
    (domain DomainName (Range+)) |
    (domain DomainName Value)
DomainName ::=
    Symbol
LowerBound ::=
    Integer
UpperBound ::=
    Integer
Value ::=
    Integer
Range ::=
    Integer | (Integer Integer)
```

Sugar: задание областей значений – пример

```
; это комментарий
; d0 = { x | 1000 ≤ x ≤ 2000 }
(domain d0 1000 2000)
; d1 = { 1, 2, 3, 5, 10, 11, 12 }
(domain d1 ((1 3) 5 (10 12)))
; d2 = { 1000 }
(domain d2 1000)
```

Sugar: задание целочисленных переменных

IntegerVariableDefinition ::=
  (int IntegerVariableName DomainName) |
  (int IntegerVariableName LowerBound UpperBound) |
  (int IntegerVariableName (Range+)) |
  (int IntegerVariableName Value)

IntegerVariableName ::=
  Symbol

Пример:

```
(int x0 d0)                ; x0 ∈ d0
(int x1 1000 2000)         ; x1 ∈ { x | 1000 ≤ x ≤ 2000 }
(int x2 ((1 3) 5 (10 12))) ; x2 ∈ { 1, 2, 3, 5, 10, 11, 12 }
(int x3 1000)              ; x3 ∈ { 1000 }
```

Sugar: выражения

- Префиксная нотация

Term ::=
  Integer |
  IntegerVariableName |
  (abs Term) |
  (neg Term) | (- Term) |
  (add Term*) | (+ Term*) |
  (sub Term Term+) | (- Term Term+) |
  (mul Term Term) | (* Term Term) |
  (div Term Term) | (/ Term Term) |
  (mod Term Term) | (% Term Term) |
  (pow Term Term) |
  (min Term Term) |
  (max Term Term) |
  (if LogicalFormula Term Term)

- **Пример:** (- x y z) ; x – y – z

Sugar: отношения

RelationDefinition ::=
  (relation RelationName Arity RelationBody)
RelationName ::=
  Symbol
Arity ::=
  Integer
RelationBody ::=
  (RelationType Tuple*)
RelationType ::=
  supports | conflicts
Tuple ::=
  (Integer+)

Пример: (relation r0 2 (conflicts (0 0) (1 1) (2 2)))

Sugar: определение предикатов

PredicateDefinition ::=
   (predicate PredicateHead PredicateBody)
PredicateHead ::=
   (PredicateName Parameter*)
PredicateName ::=
   Symbol
Parameter ::=
   IntegerVariableName
PredicateBody ::=
   LogicalFormula

Пример: (predicate (p0 x1 x2) (<= x1 (+ x2 1)) ; (p0 x1 x2) ⇔ $x1 \leq x2+1$

Sugar: ограничения

- Приведем только примеры
- (or p0 (r0 x4 x5) (>= x0 (+ x0 1)))
- (alldifferent (+ x1 1) (+ x2 2) (+ x3 3) (+ x4 4))
- ; 1*v0 + 2*v1 - 3*v3 > 12

  (weightedsum ((1 v0) (2 v1) (-3 v2)) gt 12)
- (cumulative ((s0 2 nil 1) (s1 2 nil 1) (s2 2 nil 1) (s3 2 nil 1)) 2)
- (element i (x1 x2 x3 x4) xi)

Позиции *Sugar* в соревнованиях 2009 CSP Solver Competition

Category	Sugar v1.14.6+minisat	Sugar v1.14.6+picosat
2-ARY-EXT	8 / 13	9 / 13
2-ARY-INT	7 / 13	6 / 13
N-ARY-EXT	12 / 14	13 / 14
N-ARY-INT	9 / 13	10 / 13
Alldiff	3 / 11	**1** / 11
Alldiff+Elt+Wsum	2 / 8	**1** / 8
Alldiff+Cumul+Elt+Wsum	**1** / 6	2 / 6

Спасибо за внимание!
Вопросы?