

制約論理プログラミング言語 CAL

ICOT

第 1 研究室

相 場 亮

制約論理プログラミングの特徴

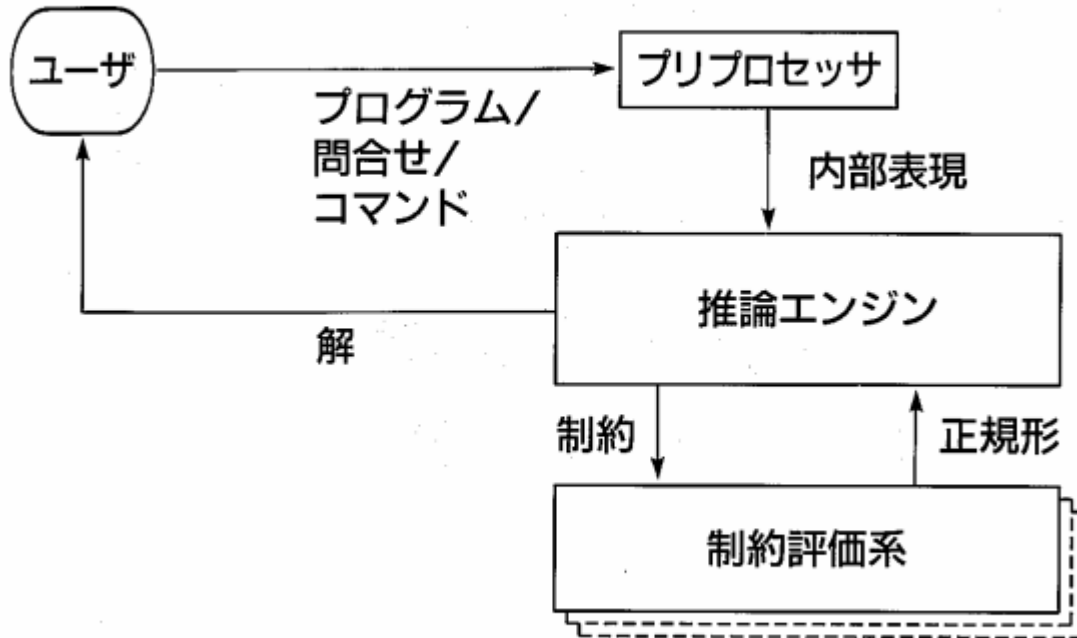
- 関係を記述する
- バックトラッキング
- 記号出力

$$x = y + 1$$

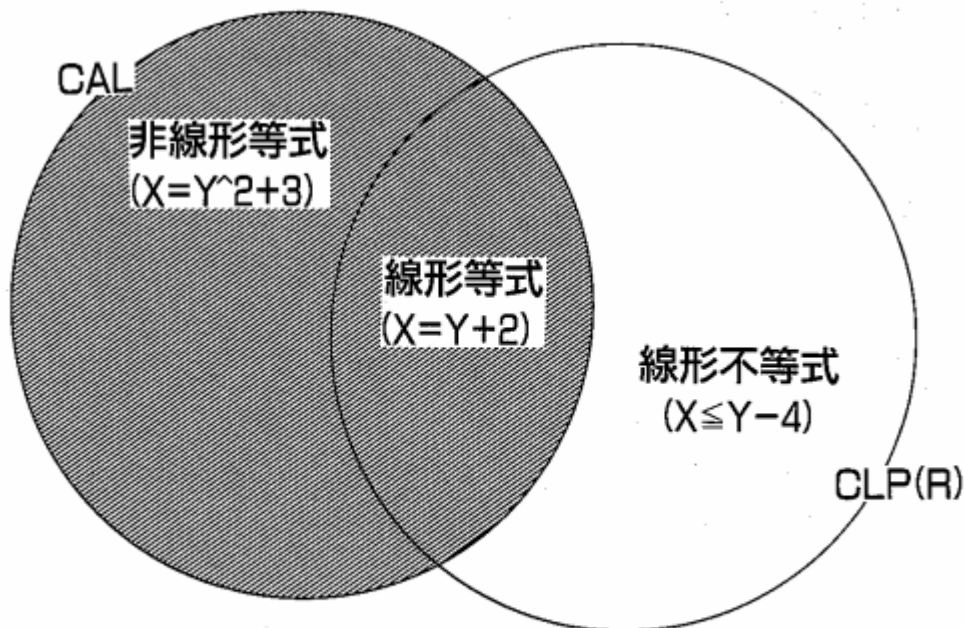
- 制約評価

$$\begin{cases} x + y = 3 \\ x - y = 1 \end{cases} \Rightarrow \begin{cases} x = 2 \\ y = 1 \end{cases}$$

システム構成



CALとCLP(R)との比較



プログラム例-1 (鶴亀算)

trkm (Tsuru, Kame, Atama, Ashi) :-
Atama = Tsuru + Kame,
Ashi = 2*Tsuru + 4*Kame.

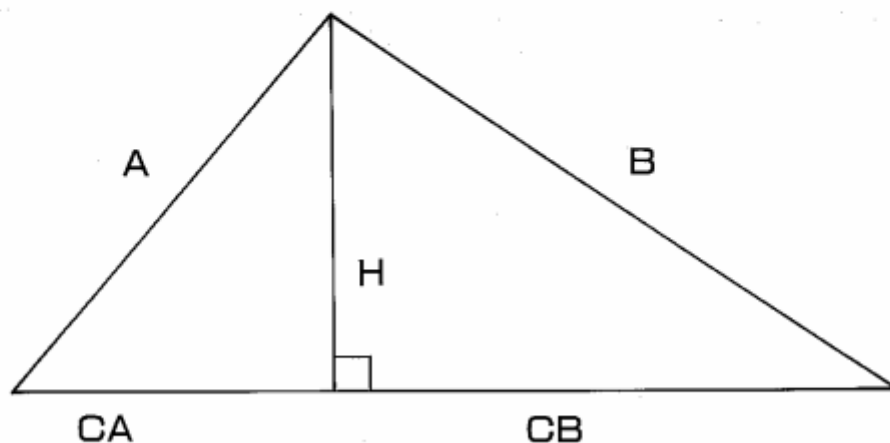
? - trkm (tsuru, kame, 4, 10).

→

tsuru = 3
kame = 1

プログラム例-2 (ヘロンの公式)

sur(H, A, S) :- A*H = 2*S.
right(A, B, C) :- A² + B² = C².
tri(A, B, C, S) :- C = CA + CB,
right(CA, H, A), right(CB, H, B), sur(H, C, S).



$$S = \sqrt{(r(r-a)(r-b)(r-c))} \quad \text{ただし } r = (a+b+c)/2$$

tri(a, b, c, s).

→

$$s^2 = (-c^4 + -1*a^4 + 2*(2*b^2*a^2) + -1*b^4 + 2*(2*c^2*a^2) + 2*(2*c^2*b^2))/16$$

tri(3, 4, 5, S).

→

$$S^2 = 36$$

プログラム例-3(未定乗数法)

ex(F, Constraint, Vars) :-

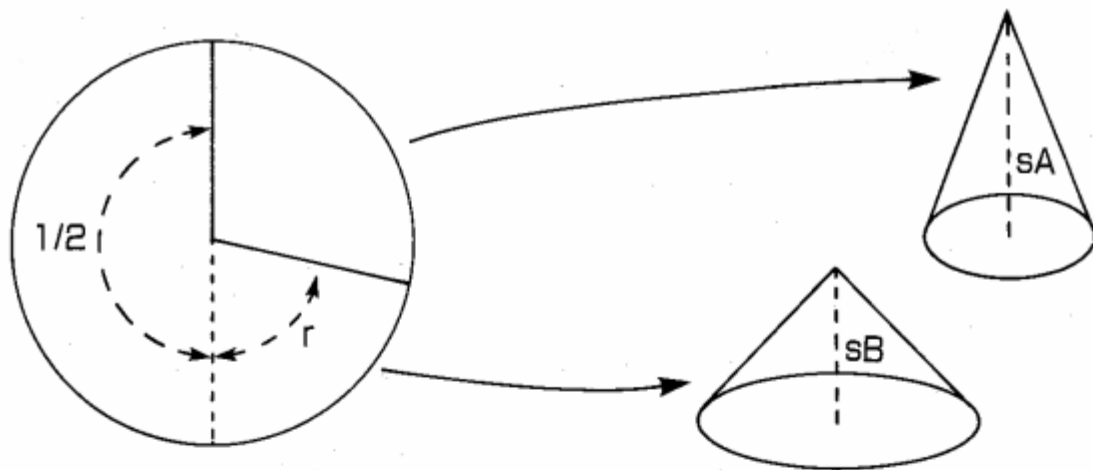
lag(Constraint, Lag), difs(Vars, F, Lag).

lag([], 0) :- !.

lag([L=R|Cs], Mult*(L-R)+Lag) :- L=R, !, lag(Cs, Lag).

difs([], _, _) :- !.

difs([Var|Vars], F, Lag) :- dif(F, Var)=dif(Lag, Var), !, difs(Vars, F, Lag).



$$sA^2 + (1/2 + r)^2 - 1 = 0 \quad (= \psi)$$

$$sB^2 + (1/2 - r)^2 - 1 = 0 \quad (= \phi)$$

$$F_r = \lambda_1 \psi_r + \lambda_2 \phi_r$$

$$F_{sA} = \lambda_1 \psi_{sA} + \lambda_2 \phi_{sA}$$

$$F_{sB} = \lambda_1 \psi_{sB} + \lambda_2 \phi_{sB} \quad \text{ただし, } F = (1/2 + r)^2 \cdot sA + (1/2 - r)^2 \cdot sB$$

$$\text{ex}((1/2 + r)^2 \cdot sA + (1/2 - r)^2 \cdot sB,$$

$$[sA^2 + (1/2 + r)^2 = 1, sB^2 + (1/2 - r)^2 = 1],$$

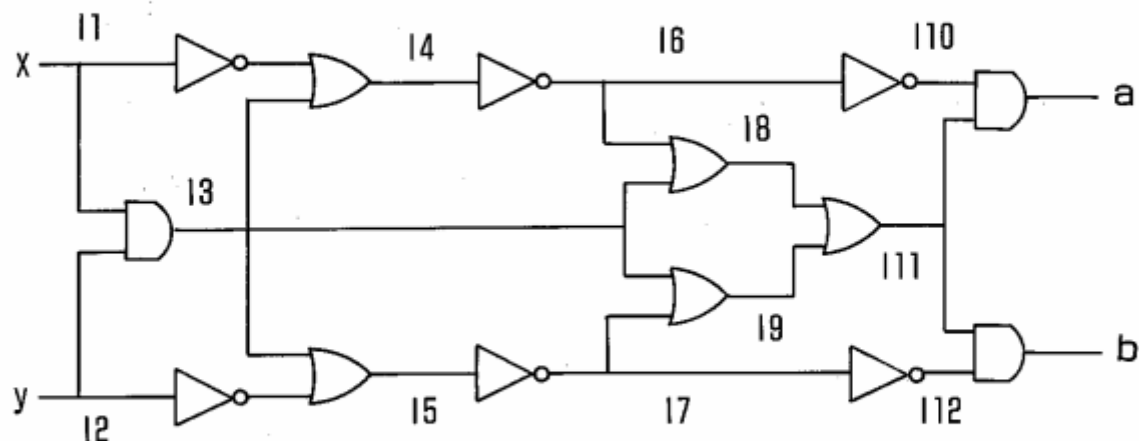
$$[sA, sB, r]).$$

→

Time Used: 91080 msec = 91.080 sec

$$18: r^7 = (29/12) \cdot r^5 + (-17/48) \cdot r^3 + (5/576) \cdot r$$

ブール代数のためのシステム(Boole CAL)



cross(X, Y, A, B) :-

13 = X & Y, 14 = ~X v 13, 15 = ~Y v 13,

18 = ~14 v 13, 19 = ~15 v 13,

A = 14 & 111, 111 = 18 v 19, B = 15 & 111.

cross(x, y, a, b).

→

Time Used : 930msec = .930sec

x = b

y = a

今後の課題

- 制約評価系の充実と多元化
 限量子除去アルゴリズムの実装と型の導入
- ある程度大規模なアプリケーション
 PSI - II マシン上での展開
- 並列化への対応
 並列制約論理プログラミング言語に向けて