

ILLUSTRATION: Alloy expression to SAT formula

Boolean variables: Man: m_0, m_1

Woman: w_0, w_1

parents: $p_{00}, p_{01}, p_{10}, p_{11}$

spouse: $s_{00}, s_{01}, s_{10}, s_{11}$

Constraints from signatures: $m_0 \oplus w_0 \wedge m_1 \oplus w_1$

1) Constraint for Alloy expression Man.spouse in Woman:

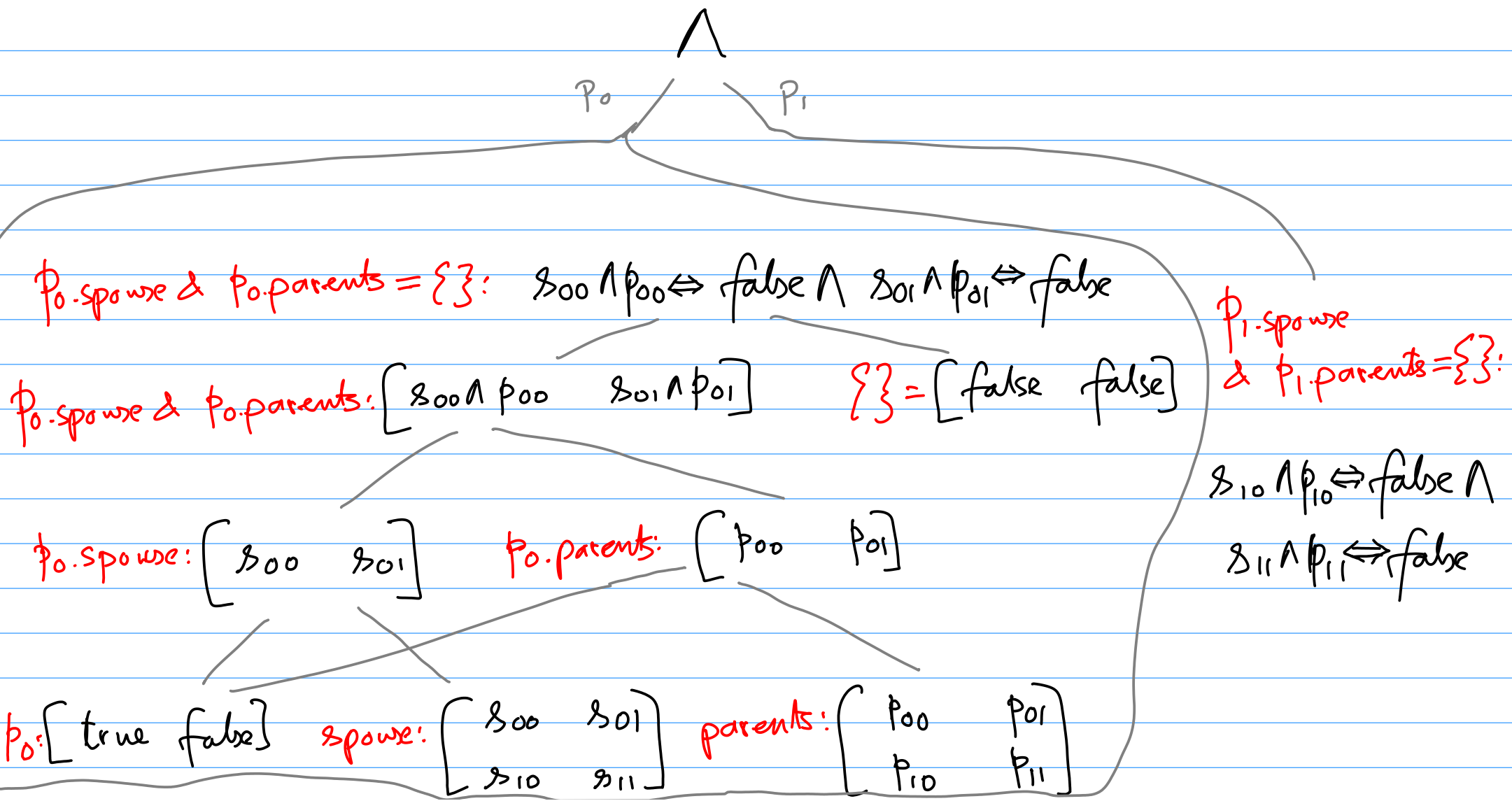
$$((m_0 \wedge s_{00}) \vee (m_1 \wedge s_{10}) \Rightarrow w_0) \wedge ((m_0 \wedge s_{01}) \vee (m_1 \wedge s_{11}) \Rightarrow w_1)$$

Man.spouse: $[(m_0 \wedge s_{00}) \vee (m_1 \wedge s_{10}) \quad (m_0 \wedge s_{01}) \vee (m_1 \wedge s_{11})]$ Woman: $[w_0 \quad w_1]$

Man: $[m_0 \quad m_1]$ spouse: $\begin{bmatrix} s_{00} & s_{01} \\ s_{10} & s_{11} \end{bmatrix}$

2)

Constraint for all $p: \text{Person} \mid (p.\text{spouse} \ \& \ p.\text{parents}) = \{\}$



ALGORITHM: Alloy expression to SAT formula

Terminology:

Global variable: relation, or non-top-level signature

TSignature: top-level signature

DSignature: derived signature

Variable: Global variable | Quantified variable

R-expr: Alloy relational expression

B-expr: Alloy formula

Formula: Propositional formula

Matrix: Matrix/vector of formulas

n_s : Given bound for top-level signature S

Type declarations:

Env: Variable $\rightarrow$ Type

Type: TSignature | TSignature $\times$ TSignature
| DSignature

// Assume all relations are
// binary relations, from
// top level sigs to top
// level sigs.

Procedures

Formula $\text{main}(\text{B-expr } \text{exp}) \{$

env = a mapping where each non-top-level signature is mapped to its corresponding top-level signature,
& each global relation r is mapped to a term (S_1, S_2) where S_1 is the top-level signature corresponding to r 's domain & S_2 is the top-level signature corresponding to r 's range.

return $\text{Translate-B-expr}(\text{exp}, \text{env});$

}

Matrix Translate-R-Expr (R-expr exp, Env env) {

case exp is π , where π is a relation:

$$(S_1, S_2) = \text{env}(\pi)$$

m = a new matrix of size $n_{S_1} \times n_{S_2}$

for all $0 \leq i \leq n_{S_1}, 0 \leq j \leq n_{S_2}$

$m[i, j] = \pi_{ij}$, where π_{ij} is the (i, j) th boolean variable corresponding to π

return m ;

case exp is ϑ , where ϑ is a non-top-level signature:

$$S = \text{env}(\vartheta)$$

return $[\vartheta_0, \vartheta_1, \dots, \vartheta_{n_S-1}]$

case exp is a top-level signature S :

return $[\text{true}, \text{true}, \dots, \text{true}]$ // size of vector is n_S

case exp is w_i , an instance of a quantified variable w :
 $\vartheta = \text{env}(w_i)$. S = top-level sig on which ϑ is based.

m = a new row vector of size n_S

if $(S = \vartheta)$ $m[i] = \text{true}$ else $m[i] = \vartheta_i$ // ϑ_i is i^{th} boolean
// corresponding to

for all j in $0 \dots (n_S - 1), j \neq i : m[j] = \text{false}$ // The signature ϑ

return m ;

case exp is $e_1 \cdot e_2$:

$m_1 = \text{Translate-R-expr}(e_1, \text{env})$;

$m_2 = \text{Translate-R-expr}(e_2, \text{env})$;

assert ($\# \text{ cols in } m_1 = \# \text{ rows in } m_2$);

$m_3 = m_1 \times m_2$ // use '&&' instead of 'x' & '||'

return m_3 ; // instead of '+'

.... similarly, handle all other relational operators

Formula $\text{Translate-B-expr}(B\text{-expr } exp, Env \text{ env}) \{$

case exp is "all $v:T \mid f$ ": // we assume T is a sig.
// and not a general expression
let s be top-level signature corresponding to T

for i an $0 \dots (n_s - 1)$ {

$f_i = f$, with all occurrences of v replaced with v_i

$res_i = \text{Translate-B-expr}(f_i, env ++ (v_i \rightarrow T))$

}

return $\bigwedge_{0 \dots n_s} res_i$;

case exp is " e_1 in e_2 ":

$m_1 = \text{Translate-R-expr}(e_1, \text{env});$

$m_2 = \text{Translate-R-expr}(e_2, \text{env});$

assert (dimensions of m_1 = dimensions of m_2);

$p = \text{num_rows}(m_1); \quad q = \text{num_cols}(m_1);$

return $\bigwedge (m_1[i,j] \Rightarrow m_2[i,j])$

$0 \leq i < p,$

$0 \leq j < q$

case exp is "Some e_1 ":

$m = \text{Translate-R-expr}(e_1, \text{env});$

$p = \text{num_rows}(m); \quad q = \text{num_cols}(m);$

return $\bigvee m[i, j]$

$0 \leq i < p,$

$0 \leq j < q$

... similarly, all other cases of boolean operators...

}