Static Analysis

Static analysis by abstract interpretation is able

- to prove automatically restricted properties
  - absence of runtime errors,
  - resource consumptions...
- on large programs

Example: Astrée successfully analyses ~1Mloc of a critical control-command software

What about soundness ?

Abstract interpretation gives
you a guideline to prove
soundness of a static analysis

What about soundness ?

Abstract interpretation gives you a guideline to prove soundness of a static analysis

...on paper

$$
\begin{array}{ll}
 & \ddot{\alpha}[\![P]\!](\mathrm{Post}[\![\texttt{if } B \texttt{ then } S_t \texttt{ else } S_f \texttt{ fi}]\!]) \\
= & \quad \wr \text{def. (110) of } \ddot{\alpha}[\![P]\!] \wr \\
 & \ddot{\alpha}[\![P]\!] \circ \mathrm{Post}[\![\texttt{if } B \texttt{ then } S_t \texttt{ else } S_f \texttt{ fi}]\!] \circ \ddot{\gamma}[\![P]\!] \\
= & \quad \wr \text{def. (103) of Post} \wr \\
 & \ddot{\alpha}[\![P]\!] \circ \mathrm{post}[\tau^\star[\![\texttt{if } B \texttt{ then } S_t \texttt{ else } S_f \texttt{ fi}]\!]] \circ \ddot{\gamma}[\![P]\!] \\
= & \quad \wr \text{big step operational semantics (93)} \wr \\
 & \ddot{\alpha}[\![P]\!] \circ \mathrm{post}\,[(1_{\Sigma[\![P]\!]} \cup \tau^B) \circ \tau^\star[\![S_t]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^t) \cup (1_{\Sigma[\![P]\!]} \cup \tau^{\bar{B}}) \circ \tau^\star[\![S_f]\!] \circ (1_{\Sigma[\![P]\!]} \cup \\
 & \tau^f)] \circ \ddot{\gamma}[\![P]\!] \\
= & \quad \wr \text{Galois connection (98) so that post preserves joins} \wr \\
 & \ddot{\alpha}[\![P]\!] \circ (\mathrm{post}\,[(1_{\Sigma[\![P]\!]} \cup \tau^B) \circ \tau^\star[\![S_t]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^t)] \dot{\cup} \\
 & \mathrm{post}[(1_{\Sigma[\![P]\!]} \cup \tau^{\bar{B}}) \circ \tau^\star[\![S_f]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^f)]) \circ \ddot{\gamma}[\![P]\!] \\
= & \quad \wr \text{Galois connection (106) so that } \ddot{\alpha}[\![P]\!] \text{ preserves joins} \wr \\
 & (\ddot{\alpha}[\![P]\!] \circ \mathrm{post}[(1_{\Sigma[\![P]\!]} \cup \tau^B) \circ \tau^\star[\![S_t]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^t)] \circ \ddot{\gamma}[\![P]\!]) \ddot{\cup} (\ddot{\alpha}[\![P]\!] \circ \\
 & \mathrm{post}[(1_{\Sigma[\![P]\!]} \cup \tau^{\bar{B}}) \circ \tau^\star[\![S_f]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^f)] \circ \ddot{\gamma}[\![P]\!]) \\
\dot{\sqsubseteq} & \quad \wr \text{lemma (5.3) and similar one for the } \texttt{else} \text{ branch} \wr \\
 & \lambda J^{t'} \cdot \text{let } J^{t'} = \lambda l \in \mathrm{in}_P[\![P]\!] \cdot (l = \mathrm{at}_P[\![S_t]\!] \ ? \ J_{\mathrm{at}_P[\![S_t]\!]} \dot{\cup} \mathrm{Abexp}[\![B]\!](J_\ell) \ ¿ \ J_l) \text{ in} \qquad (120) \\
 & \quad \text{let } J^{t''} = \mathrm{APost}[\![S_t]\!](J^{t'}) \text{ in} \\
 & \quad\quad \lambda l \in \mathrm{in}_P[\![P]\!] \cdot (l = \ell' \ ? \ J^{t''}_{\ell'} \dot{\cup} J^{t''}_{\mathrm{after}_P[\![S_t]\!]} \ ¿ \ J^{t''}_l) \\
 & \quad \ddot{\cup} \\
 & \quad \text{let } J^{f'} = \lambda l \in \mathrm{in}_P[\![P]\!] \cdot (l = \mathrm{at}_P[\![S_f]\!] \ ? \ J_{\mathrm{at}_P[\![S_f]\!]} \dot{\cup} \mathrm{Abexp}[\![T(\neg B)]\!](J_\ell) \ ¿ \ J_l) \text{ in} \\
 & \quad\quad \text{let } J^{f''} = \mathrm{APost}[\![S_f]\!](J^{f'}) \text{ in} \\
 & \quad\quad\quad \lambda l \in \mathrm{in}_P[\![P]\!] \cdot (l = \ell' \ ? \ J^{f''}_{\ell'} \dot{\cup} J^{f''}_{\mathrm{after}_P[\![S_f]\!]} \ ¿ \ J^{f''}_l) \\
= & \quad \wr \text{by grouping similar terms} \wr \\
 & \lambda J \cdot \text{let } J^{t'} = \lambda l \in \mathrm{in}_P[\![P]\!] \cdot (l = \mathrm{at}_P[\![S_t]\!] \ ? \ J_{\mathrm{at}_P[\![S_t]\!]} \dot{\cup} \mathrm{Abexp}[\![B]\!](J_\ell) \ ¿ \ J_l) \\
 & \quad \text{and } J^{f'} = \lambda l \in \mathrm{in}_P[\![P]\!] \cdot (l = \mathrm{at}_P[\![S_f]\!] \ ? \ J_{\mathrm{at}_P[\![S_f]\!]} \dot{\cup} \mathrm{Abexp}[\![T(\neg B)]\!](J_\ell) \ ¿ \ J_l) \text{ in} \\
 & \quad\quad \text{let } J^{t''} = \mathrm{APost}[\![S_t]\!](J^{t'}) \\
 & \quad\quad \text{and } J^{f''} = \mathrm{APost}[\![S_f]\!](J^{f'}) \text{ in} \\
 & \quad\quad\quad \lambda l \in \mathrm{in}_P[\![P]\!] \cdot (l = \ell' \ ? \ J^{t''}_{\ell'} \dot{\cup} J^{t''}_{\mathrm{after}_P[\![S_t]\!]} \dot{\cup} J^{f''}_{\ell'} \dot{\cup} J^{f''}_{\mathrm{after}_P[\![S_f]\!]} \ ¿ \ J^{t''}_l \dot{\cup} J^{f''}_l) \\
= & \quad \wr \text{by locality (113) and labelling scheme (59) so that in particular } J^{t''}_{\ell'} = J^{t'}_{\ell'} = J^t_{\ell'} = J^f_{\ell'} \\
 & \quad = J^{f'}_{\ell'} = J^{f''}_{\ell'} \text{ and } \mathrm{APost}[\![S_t]\!] \text{ and } \mathrm{APost}[\![S_f]\!] \text{ do not interfere} \wr
\end{array}
$$

©P.Cousot

What about soundness ?

Abstract interpretation gives you a guideline to prove soundness of a static analysis

But then comes the time to implement the analysis

...on paper

$$
\begin{array}{ll}
 & \ddot{\alpha}[\![P]\!](\mathrm{Post}[\![\texttt{if } B \texttt{ then } S_t \texttt{ else } S_f \texttt{ fi}]\!]) \\
= & \quad \text{〖def. (110) of } \ddot{\alpha}[\![P]\!]\text{〗} \\
 & \ddot{\alpha}[\![P]\!] \circ \mathrm{Post}[\![\texttt{if } B \texttt{ then } S_t \texttt{ else } S_f \texttt{ fi}]\!] \circ \ddot{\gamma}[\![P]\!] \\
= & \quad \text{〖def. (103) of Post〗} \\
 & \ddot{\alpha}[\![P]\!] \circ \mathrm{post}[\tau^\star[\![\texttt{if } B \texttt{ then } S_t \texttt{ else } S_f \texttt{ fi}]\!]] \circ \ddot{\gamma}[\![P]\!] \\
= & \quad \text{〖big step operational semantics (93)〗} \\
 & \ddot{\alpha}[\![P]\!] \circ \mathrm{post}\,[(1_{\Sigma[\![P]\!]} \cup \tau^B) \circ \tau^\star[\![S_t]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^t) \cup (1_{\Sigma[\![P]\!]} \cup \tau^{\bar{B}}) \circ \tau^\star[\![S_f]\!] \circ (1_{\Sigma[\![P]\!]} \cup \\
 & \tau^f)] \circ \ddot{\gamma}[\![P]\!] \\
= & \quad \text{〖Galois connection (98) so that post preserves joins〗} \\
 & \ddot{\alpha}[\![P]\!] \circ (\mathrm{post}\,[(1_{\Sigma[\![P]\!]} \cup \tau^B) \circ \tau^\star[\![S_t]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^t)] \dot{\cup} \\
 & \mathrm{post}[(1_{\Sigma[\![P]\!]} \cup \tau^{\bar{B}}) \circ \tau^\star[\![S_f]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^f)]) \circ \ddot{\gamma}[\![P]\!] \\
= & \quad \text{〖Galois connection (106) so that } \ddot{\alpha}[\![P]\!] \text{ preserves joins〗} \\
 & (\ddot{\alpha}[\![P]\!] \circ \mathrm{post}[(1_{\Sigma[\![P]\!]} \cup \tau^B) \circ \tau^\star[\![S_t]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^t)] \circ \ddot{\gamma}[\![P]\!]) \ddot{\cup} (\ddot{\alpha}[\![P]\!] \circ \\
 & \mathrm{post}[(1_{\Sigma[\![P]\!]} \cup \tau^{\bar{B}}) \circ \tau^\star[\![S_f]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^f)] \circ \ddot{\gamma}[\![P]\!]) \\
\dot{\sqsubseteq} & \quad \text{〖lemma (5.3) and similar one for the } \texttt{else} \text{ branch〗} \\
 & \lambda J^\bullet \cdot \text{let } J^{t'} = \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet \cdot (l = \mathrm{at}_P[\![S_t]\!] \ ? \ J_{\mathrm{at}_P[\![S_t]\!]} \dot{\cup} \mathrm{Abexp}[\![B]\!](J_\ell) \ ¿ \ J_l) \text{ in} \qquad (120) \\
 & \quad \text{let } J^{t''} = \mathrm{APost}[\![S_t]\!](J^{t'}) \text{ in} \\
 & \quad\quad \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet \cdot (l = \ell' \ ? \ J^{t''}_{\ell'} \dot{\cup} J^{t''}_{\mathrm{after}_P[\![S_t]\!]} \ ¿ \ J^{t''}_l) \\
 & \ddot{\cup} \\
 & \quad \text{let } J^{f'} = \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet \cdot (l = \mathrm{at}_P[\![S_f]\!] \ ? \ J_{\mathrm{at}_P[\![S_f]\!]} \dot{\cup} \mathrm{Abexp}[\![T(\neg B)]\!](J_\ell) \ ¿ \ J_l) \text{ in} \\
 & \quad\quad \text{let } J^{f''} = \mathrm{APost}[\![S_f]\!](J^{f'}) \text{ in} \\
 & \quad\quad\quad \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet \cdot (l = \ell' \ ? \ J^{f''}_{\ell'} \dot{\cup} J^{f''}_{\mathrm{after}_P[\![S_f]\!]} \ ¿ \ J^{f''}_l) \\
= & \quad \text{〖by grouping similar terms〗} \\
 & \lambda J^\bullet \cdot \text{let } J^{t'} = \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet \cdot (l = \mathrm{at}_P[\![S_t]\!] \ ? \ J_{\mathrm{at}_P[\![S_t]\!]} \dot{\cup} \mathrm{Abexp}[\![B]\!](J_\ell) \ ¿ \ J_l) \\
 & \quad \text{and } J^{f'} = \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet \cdot (l = \mathrm{at}_P[\![S_f]\!] \ ? \ J_{\mathrm{at}_P[\![S_f]\!]} \dot{\cup} \mathrm{Abexp}[\![T(\neg B)]\!](J_\ell) \ ¿ \ J_l) \text{ in} \\
 & \quad\quad \text{let } J^{t''} = \mathrm{APost}[\![S_t]\!](J^{t'}) \\
 & \quad\quad \text{and } J^{f''} = \mathrm{APost}[\![S_f]\!](J^{f'}) \text{ in} \\
 & \quad\quad\quad \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet \cdot (l = \ell' \ ? \ J^{t''}_{\ell'} \dot{\cup} J^{t''}_{\mathrm{after}_P[\![S_t]\!]} \dot{\cup} J^{f''}_{\ell'} \dot{\cup} J^{f''}_{\mathrm{after}_P[\![S_f]\!]} \ ¿ \ J^{t''}_l \dot{\cup} J^{f''}_l) \\
= & \quad \text{〖by locality (113) and labelling scheme (59) so that in particular } J^{t''}_{\ell'} = J^{t'}_{\ell'} = J^t_{\ell'} = J^f_{\ell'} \\
 & \quad = J^{f'}_{\ell'} = J^{f''}_{\ell'} \text{ and } \mathrm{APost}[\![S_t]\!] \text{ and } \mathrm{APost}[\![S_f]\!] \text{ do not interfere〗}
\end{array}
$$

©P.Cousot

What about soundness ?

Abstract interpretation gives you a guideline to prove soundness of a static analysis

...on paper

$$
\begin{array}{ll}
 & \ddot{\alpha}[\![P]\!](\mathrm{Post}[\![\texttt{if } B \texttt{ then } S_t \texttt{ else } S_f \texttt{ fi}]\!]) \\
= & \quad \text{〖def. (110) of } \ddot{\alpha}[\![P]\!]\text{〗} \\
 & \ddot{\alpha}[\![P]\!] \circ \mathrm{Post}[\![\texttt{if } B \texttt{ then } S_t \texttt{ else } S_f \texttt{ fi}]\!] \circ \ddot{\gamma}[\![P]\!] \\
= & \quad \text{〖def. (103) of Post〗} \\
 & \ddot{\alpha}[\![P]\!] \circ \mathrm{post}[\tau^\star[\![\texttt{if } B \texttt{ then } S_t \texttt{ else } S_f \texttt{ fi}]\!]] \circ \ddot{\gamma}[\![P]\!] \\
= & \quad \text{〖big step operational semantics (93)〗} \\
 & \ddot{\alpha}[\![P]\!] \circ \mathrm{post}\,[(1_{\Sigma[\![P]\!]} \cup \tau^B) \circ \tau^\star[\![S_t]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^t) \cup (1_{\Sigma[\![P]\!]} \cup \tau^{\bar{B}}) \circ \tau^\star[\![S_f]\!] \circ (1_{\Sigma[\![P]\!]} \cup \\
 & \tau^f)] \circ \ddot{\gamma}[\![P]\!] \\
= & \quad \text{〖Galois connection (98) so that post preserves joins〗} \\
 & \ddot{\alpha}[\![P]\!] \circ (\mathrm{post}\,[(1_{\Sigma[\![P]\!]} \cup \tau^B) \circ \tau^\star[\![S_t]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^t)] \dot{\cup} \\
 & \mathrm{post}[(1_{\Sigma[\![P]\!]} \cup \tau^{\bar{B}}) \circ \tau^\star[\![S_f]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^f)]) \circ \ddot{\gamma}[\![P]\!] \\
= & \quad \text{〖Galois connection (106) so that } \ddot{\alpha}[\![P]\!] \text{ preserves joins〗} \\
 & (\ddot{\alpha}[\![P]\!] \circ \mathrm{post}[(1_{\Sigma[\![P]\!]} \cup \tau^B) \circ \tau^\star[\![S_t]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^t)] \circ \ddot{\gamma}[\![P]\!]) \dot{\cup} (\ddot{\alpha}[\![P]\!] \circ \\
 & \mathrm{post}[(1_{\Sigma[\![P]\!]} \cup \tau^{\bar{B}}) \circ \tau^\star[\![S_f]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^f)] \circ \ddot{\gamma}[\![P]\!]) \\
\dot{\sqsubseteq} & \quad \text{〖lemma (5.3) and similar one for the } \texttt{else} \text{ branch〗} \\
 & \lambda J^\bullet \cdot \text{let } J^{t'} = \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet\cdot(l = \mathrm{at}_P[\![S_t]\!] \;?\; J_{\mathrm{at}_P[\![S_t]\!]} \dot{\cup} \mathrm{Abexp}[\![B]\!](J_\ell) \;¿\; J_l) \text{ in} \qquad (120) \\
 & \quad \text{let } J^{t''} = \mathrm{APost}[\![S_t]\!](J^{t'}) \text{ in} \\
 & \quad\quad \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet\cdot(l = \ell' \;?\; J^{t''}_{\ell'} \dot{\cup} J^{t''}_{\mathrm{after}_P[\![S_t]\!]} \;¿\; J^{t''}_l) \\
 & \quad \dot{\cup} \\
 & \quad \text{let } J^{f'} = \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet\cdot(l = \mathrm{at}_P[\![S_f]\!] \;?\; J_{\mathrm{at}_P[\![S_f]\!]} \dot{\cup} \mathrm{Abexp}[\![T(\neg B)]\!](J_\ell) \;¿\; J_l) \text{ in} \\
 & \quad\quad \text{let } J^{f''} = \mathrm{APost}[\![S_f]\!](J^{f'}) \text{ in} \\
 & \quad\quad\quad \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet\cdot(l = \ell' \;?\; J^{f''}_{\ell'} \dot{\cup} J^{f''}_{\mathrm{after}_P[\![S_f]\!]} \;¿\; J^{f''}_l) \\
= & \quad \text{〖by grouping similar terms〗} \\
 & \lambda J^\bullet \cdot \text{let } J^{t'} = \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet\cdot(l = \mathrm{at}_P[\![S_t]\!] \;?\; J_{\mathrm{at}_P[\![S_t]\!]} \dot{\cup} \mathrm{Abexp}[\![B]\!](J_\ell) \;¿\; J_l) \\
 & \quad \text{and } J^{f'} = \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet\cdot(l = \mathrm{at}_P[\![S_f]\!] \;?\; J_{\mathrm{at}_P[\![S_f]\!]} \dot{\cup} \mathrm{Abexp}[\![T(\neg B)]\!](J_\ell) \;¿\; J_l) \text{ in} \\
 & \quad\quad \text{let } J^{t''} = \mathrm{APost}[\![S_t]\!](J^{t'}) \\
 & \quad\quad \text{and } J^{f''} = \mathrm{APost}[\![S_f]\!](J^{f'}) \text{ in} \\
 & \quad\quad\quad \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet\cdot(l = \ell' \;?\; J^{t''}_{\ell'} \dot{\cup} J^{t''}_{\mathrm{after}_P[\![S_t]\!]} \dot{\cup} J^{f''}_{\ell'} \dot{\cup} J^{f''}_{\mathrm{after}_P[\![S_f]\!]} \;¿\; J^{t''}_l \dot{\cup} J^{f''}_l) \\
= & \quad \text{〖by locality (113) and labelling scheme (59) so that in particular } J^{t''}_{\ell'} = J^{t'}_{\ell'} = J^t_{\ell'} = J^f_{\ell'} \\
 & \quad = J^{f'}_{\ell'} = J^{f''}_{\ell'} \text{ and } \mathrm{APost}[\![S_t]\!] \text{ and } \mathrm{APost}[\![S_f]\!] \text{ do not interfere〗}
\end{array}
$$

©P.Cousot

But then comes the time to implement the analysis

... in C !

```
matrix_t* _matrix_alloc_int(const int mr, const int nc)
{
  matrix_t* mat = (matrix_t*)malloc(sizeof(matrix_t));
  mat->nbrows = mat->_maxrows = mr;
  mat->nbcolumns = nc;
  mat->_sorted = s;
  if (mr*nc>0){
    int i;
    pkint_t* q;
    mat->_pinit = _vector_alloc_int(mr*nc);
    mat->p = (pkint_t**)malloc(mr * sizeof(pkint_t*));
    q = mat->_pinit;
    for (i=0;i<mr;i++){
      mat->p[i]=q;
      q=q+nc;
    }}
  return mat;
}


void backsubstitute(matrix_t* con, int rank)
{
  int i,j,k;
  for (k=rank-1; k>=0; k--) {
    j = pk_cherni_intp[k];
    for (i=0; i<k; i++) {
      if (pkint_sgn(con->p[i][j]))
        matrix_combine_rows(con,i,k,i,j);
    }
    for (i=k+1; i<con->nbrows; i++) {
      if (pkint_sgn(con->p[i][j]))
        matrix_combine_rows(con,i,k,i,j);
    }}
}
```

©B.Jeannet

What about soundness ?

Abstract interpretation gives you a guideline to prove soundness of a static analysis

But then comes the time to implement the analysis

...on paper

$$
\begin{array}{ll}
 & \ddot{\alpha}[\![P]\!](\mathrm{Post}[\![\texttt{if } B \texttt{ then } S_t \texttt{ else } S_f \texttt{ fi}]\!]) \\
= & \quad \text{〖def. (110) of } \ddot{\alpha}[\![P]\!]\text{〗} \\
 & \ddot{\alpha}[\![P]\!] \circ \mathrm{Post}[\![\texttt{if } B \texttt{ then } S_t \texttt{ else } S_f \texttt{ fi}]\!] \circ \ddot{\gamma}[\![P]\!] \\
= & \quad \text{〖def. (103) of Post〗} \\
 & \ddot{\alpha}[\![P]\!] \circ \mathrm{post}[\tau^\star[\![\texttt{if } B \texttt{ then } S_t \texttt{ else } S_f \texttt{ fi}]\!]] \circ \ddot{\gamma}[\![P]\!] \\
= & \quad \text{〖big step operational semantics (93)〗} \\
 & \ddot{\alpha}[\![P]\!] \circ \mathrm{post}\,[(1_{\Sigma[\![P]\!]} \cup \tau^B) \circ \tau^\star[\![S_t]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^t) \cup (1_{\Sigma[\![P]\!]} \cup \tau^{\bar{B}}) \circ \tau^\star[\![S_f]\!] \circ (1_{\Sigma[\![P]\!]} \cup \\
 & \tau^f)] \circ \ddot{\gamma}[\![P]\!] \\
= & \quad \text{〖Galois connection (98) so that post preserves joins〗} \\
 & \ddot{\alpha}[\![P]\!] \circ (\mathrm{post}\,[(1_{\Sigma[\![P]\!]} \cup \tau^B) \circ \tau^\star[\![S_t]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^t)] \dot{\cup} \\
 & \mathrm{post}[(1_{\Sigma[\![P]\!]} \cup \tau^{\bar{B}}) \circ \tau^\star[\![S_f]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^f)]) \circ \ddot{\gamma}[\![P]\!] \\
= & \quad \text{〖Galois connection (106) so that } \ddot{\alpha}[\![P]\!] \text{ preserves joins〗} \\
 & (\ddot{\alpha}[\![P]\!] \circ \mathrm{post}[(1_{\Sigma[\![P]\!]} \cup \tau^B) \circ \tau^\star[\![S_t]\!] \circ (1_{\Sigma[\![P]\!]} \cup \tau^t)] \circ \ddot{\gamma}[\![P]\!]) \dot{\cup} (\ddot{\alpha}[\![P]\!] \circ \\
 & \mathrm{post}[(1_{\Sigma[\![P]\!]} \cup \tau^{\bar{B}}) \circ \tau^\star[\![S_f \\
\dot{\sqsubseteq} & \quad \text{〖lemma (5.3) and simil} \\
 & \lambda J^\bullet \cdot \mathrm{let}\ J^{t'} = \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet \\
 & \quad \mathrm{let}\ J^{t''} = \mathrm{APost}[\![S_t]\!](\\
 & \quad\quad \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet (l = \ell' \,?\, J^{t''}_{\ell'} \dot{\cup} J^{t''}_{\mathrm{after}_P[\![S_t]\!]} \,¿\, J^{t''}_l) \\
 & \quad \dot{\cup} \\
 & \quad \mathrm{let}\ J^{f'} = \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet (l = \mathrm{at}_P[\![S_f]\!] \,?\, J_{\mathrm{at}_P[\![S_f]\!]} \dot{\cup} \mathrm{Abexp}[\![T(\neg B)]\!](J_\ell) \,¿\, J_l) \ \mathrm{in} \\
 & \quad\quad \mathrm{let}\ J^{f''} = \mathrm{APost}[\![S_f]\!](J^{f'})\ \mathrm{in} \\
 & \quad\quad\quad \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet (l = \ell' \,?\, J^{f''}_{\ell'} \dot{\cup} J^{f''}_{\mathrm{after}_P[\![S_f]\!]} \,¿\, J^{f''}_l) \\
= & \quad \text{〖by grouping similar terms〗} \\
 & \lambda J^\bullet \cdot \mathrm{let}\ J^{t'} = \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet (l = \mathrm{at}_P[\![S_t]\!] \,?\, J_{\mathrm{at}_P[\![S_t]\!]} \dot{\cup} \mathrm{Abexp}[\![B]\!](J_\ell) \,¿\, J_l) \\
 & \quad \mathrm{and}\ J^{f'} = \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet (l = \mathrm{at}_P[\![S_f]\!] \,?\, J_{\mathrm{at}_P[\![S_f]\!]} \dot{\cup} \mathrm{Abexp}[\![T(\neg B)]\!](J_\ell) \,¿\, J_l)\ \mathrm{in} \\
 & \quad\quad \mathrm{let}\ J^{t''} = \mathrm{APost}[\![S_t]\!](J^{t'}) \\
 & \quad\quad \mathrm{and}\ J^{f''} = \mathrm{APost}[\![S_f]\!](J^{f'})\ \mathrm{in} \\
 & \quad\quad\quad \lambda l \in \mathrm{in}_P[\![P]\!]^\bullet (l = \ell' \,?\, J^{t''}_{\ell'} \dot{\cup} J^{t''}_{\mathrm{after}_P[\![S_t]\!]} \dot{\cup} J^{f''}_{\ell'} \dot{\cup} J^{f''}_{\mathrm{after}_P[\![S_f]\!]} \,¿\, J^{t''}_l \dot{\cup} J^{f''}_l) \\
= & \quad \text{〖by locality (113) and labelling scheme (59) so that in particular } J^{t''}_{\ell'} = J^{t'}_{\ell'} = J^t_{\ell'} = J^f_{\ell'} \\
 & \quad = J^{f'}_{\ell'} = J^{f''}_{\ell'} \text{ and APost}[\![S_t]\!] \text{ and APost}[\![S_f]\!] \text{ do not interfere〗}
\end{array}
$$

©P.Cousot

... in C !

```
matrix_t* _matrix_alloc_int(const int mr, const int nc)
{
  matrix_t* mat = (matrix_t*)malloc(sizeof(matrix_t));
  mat->nbrows = mat->_maxrows = mr;
  mat->nbcolumns = nc;
  mat->_sorted = s;
  if (mr*nc>0){
    int i;
    pkint_t* q;
    mat->_pinit = _vector_alloc_int(mr*nc);
    mat->p = (pkint_t**)malloc(mr * sizeof(pkint_t*));
    q = mat->_pinit;
    for (i=0;i<mr;i++){
      mat->p[i]=q;
      q=q+nc;
```

```
void backsubstitute(matrix_t* con, int rank)
{
  int i,j,k;
  for (k=rank-1; k>=0; k--) {
    j = pk_cherni_intp[k];
    for (i=0; i<k; i++) {
      if (pkint_sgn(con->p[i][j]))
        matrix_combine_rows(con,i,k,i,j);
    }
    for (i=k+1; i<con->nbrows; i++) {
      if (pkint_sgn(con->p[i][j]))
        matrix_combine_rows(con,i,k,i,j);
    }}
}
```

©B.Jeannet

Do both parts talk about the same ?

Certified Static Analysis

A simple idea:

Program and prove your analysis in the same language !

Which language ?

Certified Static Analysis

A simple idea:

Program and prove your analysis in the same language !

Which language ?

Coq : an animal with two faces

First face:

- a proof assistant that allows to interactively build proof in constructive logic.

Second face:

- a functional programming language with a very rich type system example :

- with an extraction mechanism to Ocaml

Hence, Coq allows to program, to prove correct and to efficiently execute programs.

Coq : an animal with two faces

First face:

- a proof assistant that allows to interactively build proof in constructive logic.

Second face:

- a functional programming language with a very rich type system example :

  $\texttt{sort} : \forall l : \texttt{list int}, \{ l' : \texttt{list int} \mid \texttt{Sorted}(l') \wedge \texttt{Permutation}(l, l') \}$

- with an extraction mechanism to Ocaml

  $\texttt{sort} : \texttt{list int} \to \texttt{list int}$

Hence, Coq allows to program, to prove correct and to efficiently execute programs.

Static Analysis and PCC

In the context of the Mobius project, static analysis is used as a specific instance of PCC

Abstract Carrying Code

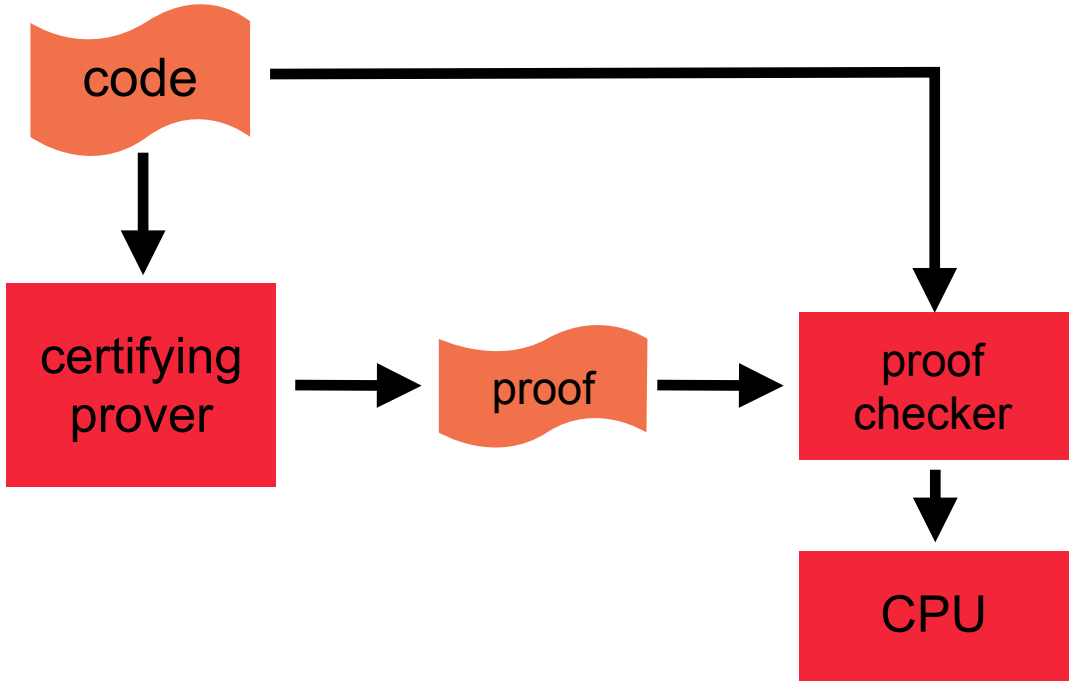
- A static analysis is specified as the solution of fixpoint system
- Any solution of the system is a valid result
  - proof = fixpoint
  - proof checking = fixpoint checking (simpler than fixpoint inference)

In ACC, checker are **specialized** in a safety property and a static analysis

Accumulation of ACC checker may compromise TCB.

Trusted Computing Base (TCB)

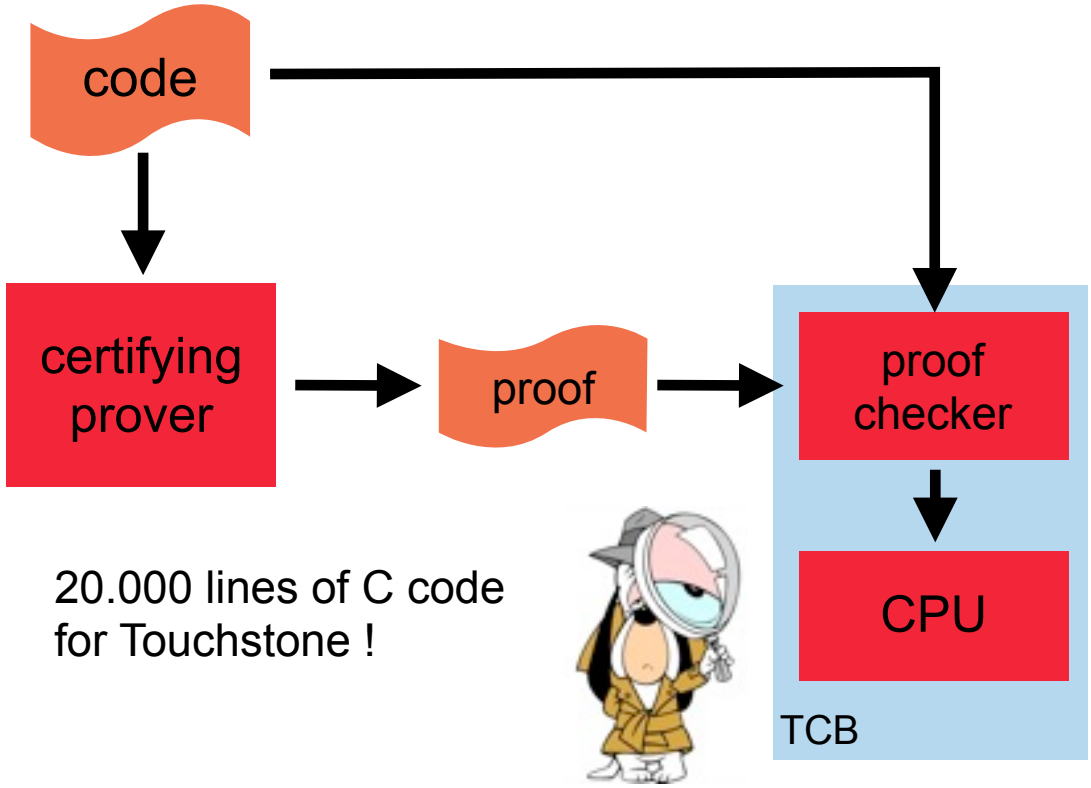
The TCB of a verification system is the set of components that must be trusted to ensure its soundness. Any bug in the others components will never affect the soundness.



Trusted Computing Base (TCB)

The TCB of a verification system is the set of components that must be trusted to ensure its soundness. Any bug in the others components will never affect the soundness.

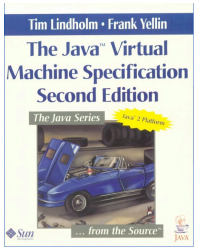
- In Traditional PCC, the TCB is formed of the checker and the computation unit.



Trusted Computing Base (TCB)

The TCB of a verification system is the set of components that must be trusted to ensure its soundness. Any bug in the others components will never affect the soundness.

- In Traditional PCC, the TCB is formed of the checker and the computation unit.
- In Certified PCC, the TCB is formed of Coq, the formal specification of the PCC soundness and the computation unit.



5.000 lines of Coq for the JVM

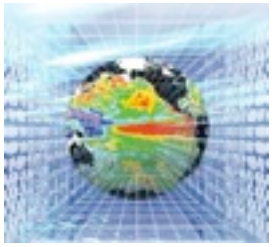
Certified PCC

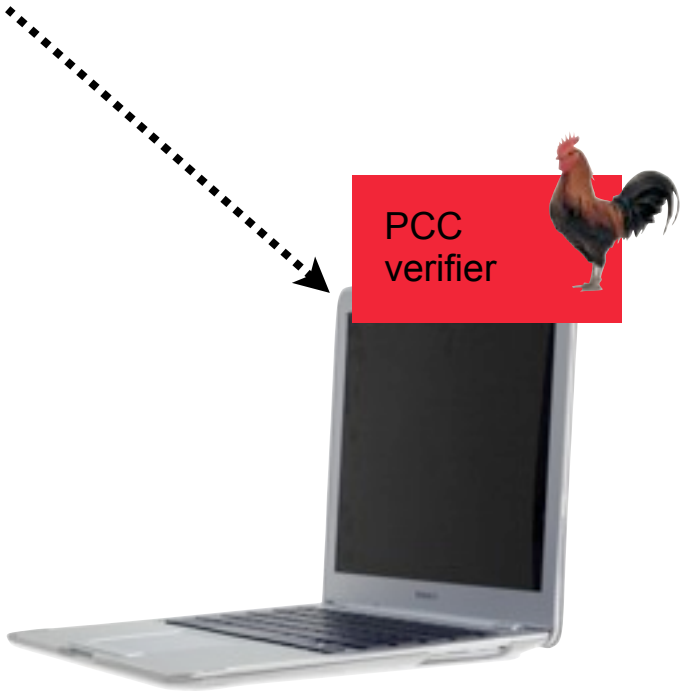
Certified PCC

PCC verifier

Contains a Coq correctness proof of the PCC verifier

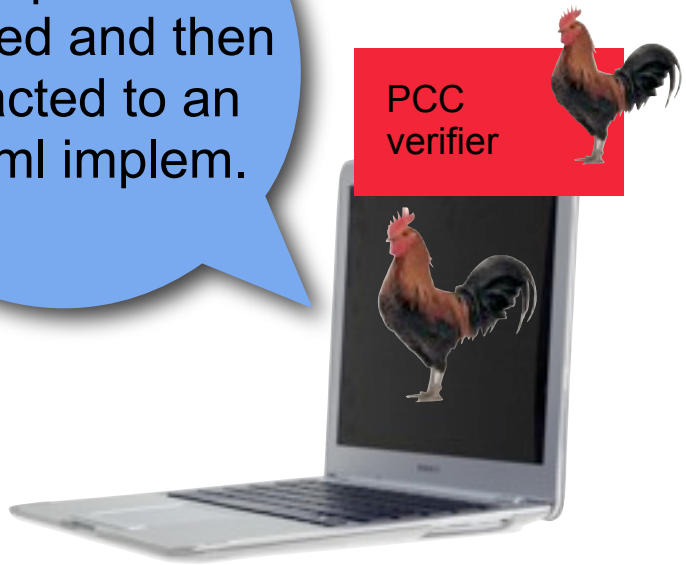
Certified PCC





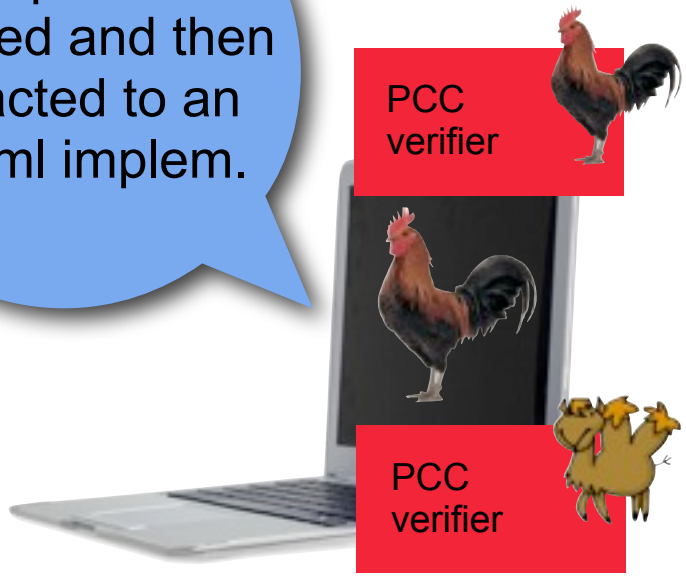
Certified PCC

The proof is checked and then extracted to an OCaml implem.

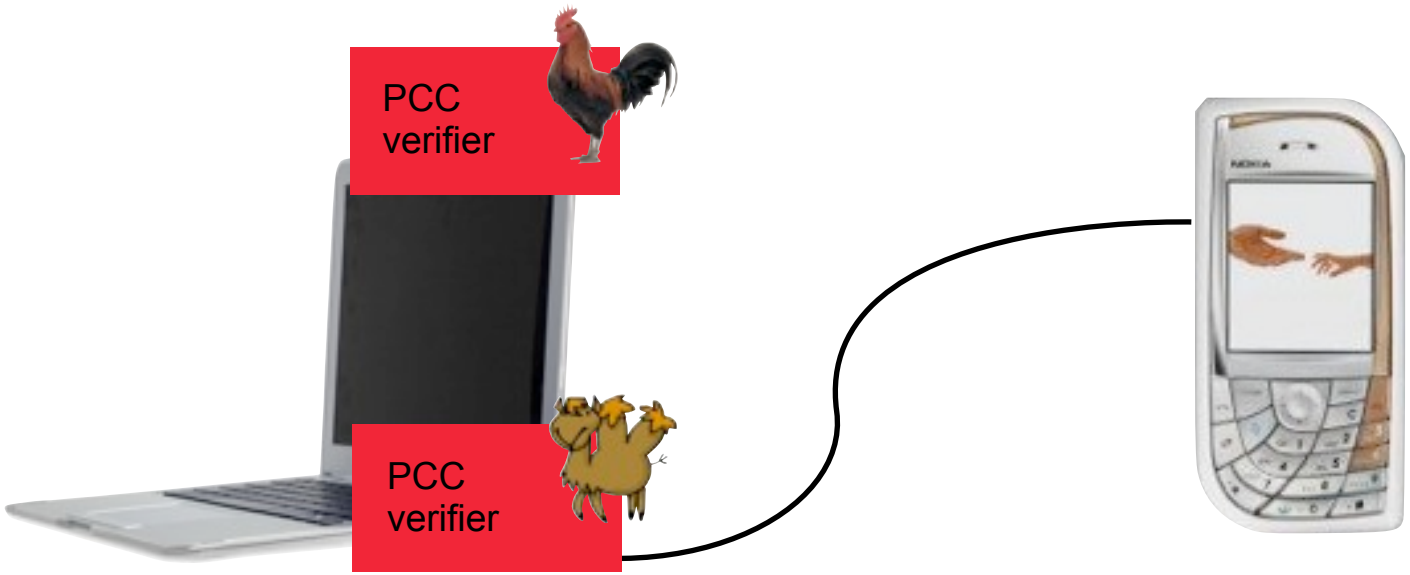


Certified PCC

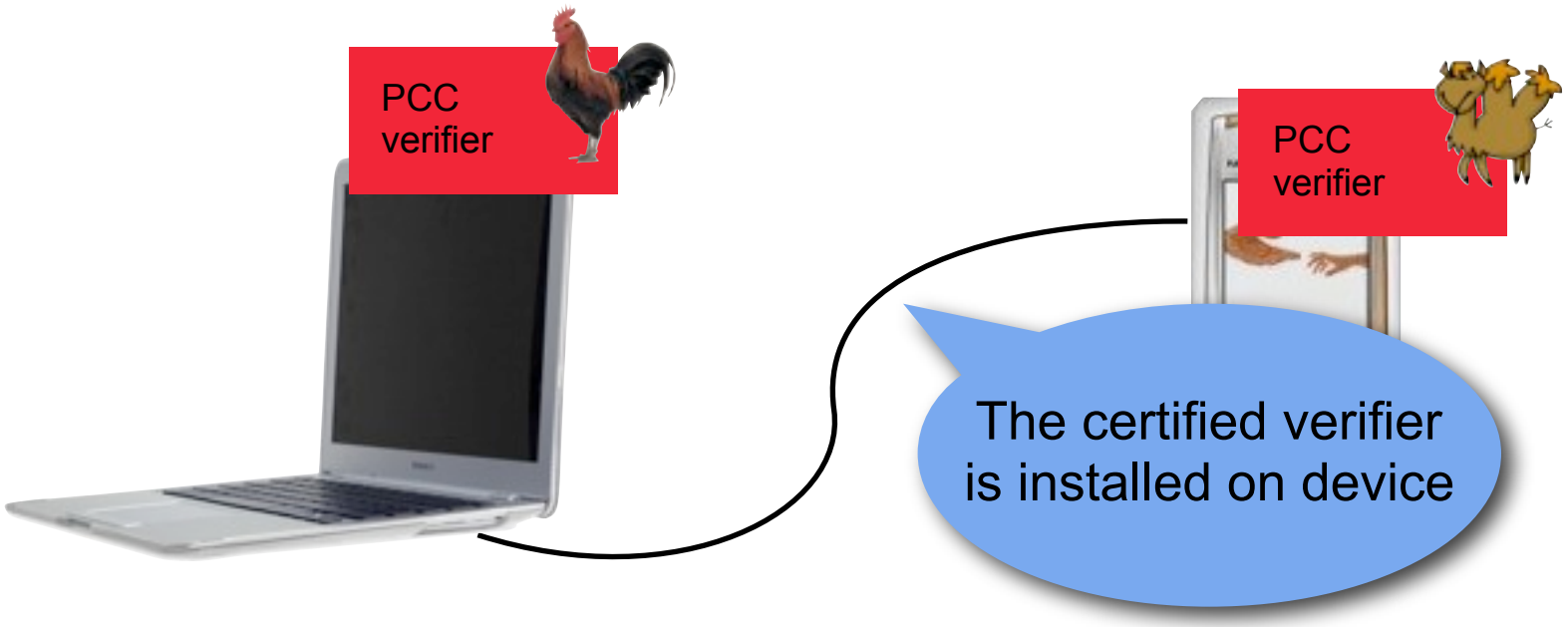
The proof is checked and then extracted to an OCaml implem.



Certified PCC



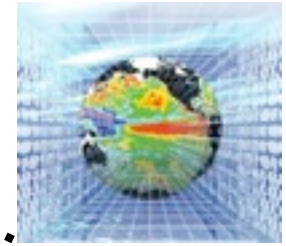
Certified PCC



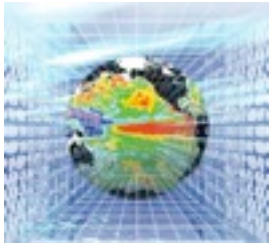
Certified PCC

Certified PCC

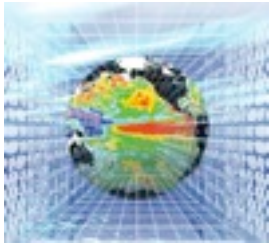
Certified PCC



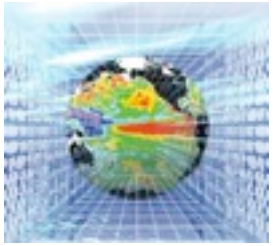
Certified PCC



Certified PCC



Certified PCC



Certified PCC

One verifier for several mobile codes